

Understanding Molecular Simulation

From Algorithms to Applications

Understanding Molecular Simulation

From Algorithms to Applications

Third Edition

Daan Frenkel

Yusuf Hamied Department of Chemistry
University of Cambridge
Cambridge, United Kingdom

Berend Smit

Laboratory of molecular simulation (LSMO)
Institut des Sciences et Ingénierie Chimiques
École Polytechnique Fédérale de Lausanne (EPFL)
Sion, Switzerland

Department of Chemical and Biomolecular Engineering
Department of Chemistry
University of California at Berkeley
Berkeley, CA, United States



ACADEMIC PRESS

An imprint of Elsevier

Academic Press is an imprint of Elsevier
125 London Wall, London EC2Y 5AS, United Kingdom
525 B Street, Suite 1650, San Diego, CA 92101, United States
50 Hampshire Street, 5th Floor, Cambridge, MA 02139, United States
The Boulevard, Langford Lane, Kidlington, Oxford OX5 1GB, United Kingdom

Copyright © 2023 Elsevier Inc. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or any information storage and retrieval system, without permission in writing from the publisher. Details on how to seek permission, further information about the Publisher's permissions policies and our arrangements with organizations such as the Copyright Clearance Center and the Copyright Licensing Agency, can be found at our website: www.elsevier.com/permissions.

This book and the individual contributions contained in it are protected under copyright by the Publisher (other than as may be noted herein).

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods, professional practices, or medical treatment may become necessary.

Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information, methods, compounds, or experiments described herein. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

ISBN: 978-0-323-90292-2

For information on all Academic Press publications
visit our website at <https://www.elsevier.com/books-and-journals>

Publisher: Candice Janco

Acquisitions Editor: Charles Bath

Editorial Project Manager: Aera Gariguez

Production Project Manager:

Bharatwaj Varatharajan

Cover Designer: Victoria Pearson

Typeset by VTeX



Working together
to grow libraries in
developing countries

www.elsevier.com • www.bookaid.org

Contents

Preface to the third edition	xv
Preface to the second edition	xix
Preface to first edition	xxi

1. Introduction

Part I Basics

2. Thermodynamics and statistical mechanics

2.1 Classical thermodynamics	12
2.1.1 Auxiliary functions	18
2.1.2 Chemical potential and equilibrium	21
2.1.3 Energy, pressure, and chemical potential	23
2.2 Statistical thermodynamics	25
2.2.1 Basic assumption	25
2.2.2 Systems at constant temperature	27
2.2.3 Towards classical statistical mechanics	28
2.3 Ensembles	31
2.3.1 Micro-canonical (constant-NVE) ensemble	31
2.3.2 Canonical (constant-NVT) ensemble	32
2.3.3 Isobaric-isothermal (constant-NPT) ensemble	33
2.3.4 Grand-canonical (constant- μ VT) ensemble	34
2.4 Ergodicity	36
2.5 Linear response theory	38
2.5.1 Static response	39
2.5.2 Dynamic response	41
2.6 Questions and exercises	45

3. Monte Carlo simulations

3.1 Preamble: molecular simulations	53
3.2 The Monte Carlo method	54
3.2.1 Metropolis method	56
3.2.2 Parsimonious Metropolis algorithm	62
3.3 A basic Monte Carlo algorithm	62
3.3.1 The algorithm	62

3.3.2	Technical details	63
3.3.2.1	Boundary conditions	65
3.3.2.2	Truncation of interactions	67
3.3.2.3	Whenever possible, use potentials that need no truncation	72
3.3.2.4	Initialization	72
3.3.2.5	Reduced units	73
3.3.3	Detailed balance versus balance	78
3.4	Trial moves	81
3.4.1	Translational moves	81
3.4.2	Orientational moves	87
3.4.2.1	Rigid, linear molecules	88
3.4.2.2	Rigid, nonlinear molecules	88
3.4.2.3	Non-rigid molecules	89
3.5	Questions and exercises	92
4.	Molecular Dynamics simulations	
4.1	Molecular Dynamics: the idea	97
4.2	Molecular Dynamics: a program	98
4.2.1	Initialization	99
4.2.2	The force calculation	102
4.2.3	Integrating the equations of motion	104
4.3	Equations of motion	106
4.3.1	Accuracy of trajectories and the Lyapunov instability	106
4.3.2	Other desirable features of an algorithm	110
4.3.3	Other versions of the Verlet algorithm	112
4.3.4	Liouville formulation of time-reversible algorithms	118
4.3.5	One more way to look at the Verlet algorithm...	122
4.4	Questions and exercises	123
5.	Computer experiments	
5.1	Static properties	127
5.1.1	Temperature	127
5.1.2	Internal energy	128
5.1.3	Partial molar quantities	128
5.1.4	Heat capacity	129
5.1.5	Pressure	130
5.1.5.1	Pressure by thermodynamic integration	133
5.1.5.2	Local pressure and method of planes	133
5.1.5.3	Virtual volume changes	134
5.1.5.4	Compressibility	135
5.1.6	Surface tension	135
5.1.7	Structural properties	139
5.1.7.1	Structure factor	139
5.1.7.2	Radial distribution function	140
5.2	Dynamical properties	147
5.2.1	Diffusion	148

5.2.2	Order-n algorithm to measure correlations	155
5.2.3	Comments on the Green-Kubo relations	160
5.3	Statistical errors	167
5.3.1	Static properties: system size	167
5.3.2	Correlation functions	169
5.3.3	Block averages	171
5.4	Questions and exercises	174
Part II		
Ensembles		
6.	Monte Carlo simulations in various ensembles	
6.1	General approach	182
6.2	Canonical ensemble	182
6.2.1	Monte Carlo simulations	183
6.2.2	Justification of the algorithm	183
6.3	Isobaric-isothermal ensemble	184
6.3.1	Statistical mechanical basis	184
6.3.2	Monte Carlo simulations	188
6.3.3	Applications	191
6.4	Isotension-isothermal ensemble	193
6.5	Grand-canonical ensemble	195
6.5.1	Statistical mechanical basis	196
6.5.2	Monte Carlo simulations	199
6.5.3	Molecular case	203
6.5.4	Semigrand ensemble	206
6.5.4.1	Phase coexistence in the semigrand ensemble	209
6.5.4.2	Chemical equilibria	211
6.6	Phase coexistence without boundaries	216
6.6.1	The Gibbs-ensemble technique	217
6.6.2	The partition function	218
6.6.3	Monte Carlo simulations	219
6.6.4	Applications	226
6.7	Questions and exercises	228
7.	Molecular Dynamics in various ensembles	
7.1	Molecular Dynamics at constant temperature	234
7.1.1	Stochastic thermostats	237
7.1.1.1	Andersen thermostat	237
7.1.1.2	Local, momentum-conserving stochastic thermostat	239
7.1.1.3	Langevin dynamics	242
7.1.2	Global kinetic-energy rescaling	244
7.1.2.1	Extended Lagrangian approach	245
7.1.2.2	Application	253
7.1.3	Stochastic global energy rescaling	256

7.1.4	Choose your thermostat carefully	257
7.2	Molecular Dynamics at constant pressure	258
7.3	Questions and exercises	259
Part III		
Free-energy calculations		
8.	Free-energy calculations	
8.1	Introduction	263
8.1.1	Importance sampling may miss important states	263
8.1.2	Why is free energy special?	264
8.2	General note on free energies	267
8.3	Free energies and first-order phase transitions	267
8.3.1	Cases where free-energy calculations are not needed	268
8.3.1.1	Direct coexistence calculations	268
8.3.1.2	Coexistence without interfaces	270
8.3.1.3	Tracing coexistence curves	270
8.4	Methods to compute free energies	274
8.4.1	Thermodynamic integration	274
8.4.2	Hamiltonian thermodynamic integration	277
8.5	Chemical potentials	279
8.5.1	The particle insertion method	280
8.5.2	Particle-insertion method: other ensembles	284
8.5.3	Chemical potential differences	287
8.6	Histogram methods	288
8.6.1	Overlapping-distribution method	289
8.6.2	Perturbation expression	292
8.6.3	Acceptance-ratio method	293
8.6.4	Order parameters and Landau free energies	296
8.6.5	Biased sampling of free-energy profiles	299
8.6.6	Umbrella sampling	301
8.6.7	Density-of-states sampling	303
8.6.8	Wang-Landau sampling	304
8.6.9	Metadynamics	309
8.6.10	Piecing free-energy profiles together: general aspects	311
8.6.11	Piecing free-energy profiles together: MBAR	312
8.7	Non-equilibrium free energy methods	317
8.8	Questions and exercises	320
9.	Free energies of solids	
9.1	Thermodynamic integration	324
9.2	Computation of free energies of solids	326
9.2.1	Atomic solids with continuous potentials	326
9.2.2	Atomic solids with discontinuous potentials	329
9.2.3	Molecular and multi-component crystals	330
9.2.4	Einstein-crystal implementation issues	332

9.2.5	Constraints and finite-size effects	340
9.3	Vacancies and interstitials	346
9.3.1	Defect free energies	346
9.3.1.1	Vacancies	348
9.3.1.2	Interstitials	350
10.	Free energy of chain molecules	
10.1	Chemical potential as reversible work	351
10.2	Rosenbluth sampling	352
10.2.1	Macromolecules with discrete conformations	353
10.2.2	Extension to continuously deformable molecules	357
10.2.3	Overlapping-distribution Rosenbluth method	363
10.2.4	Recursive sampling	365
10.2.5	Pruned-enriched Rosenbluth method	366
Part IV		
Advanced techniques		
11.	Long-ranged interactions	
11.1	Introduction	371
11.2	Ewald method	373
11.2.1	Dipolar particles	382
11.2.2	Boundary conditions	385
11.2.3	Accuracy and computational complexity	386
11.3	Particle-mesh approaches	388
11.4	Damped truncation	393
11.5	Fast-multipole methods	394
11.6	Methods that are suited for Monte Carlo simulations	398
11.6.1	Maxwell equations on a lattice	399
11.6.2	Event-driven Monte Carlo approach	402
11.7	Hyper-sphere approach	402
12.	Configurational-bias Monte Carlo	
12.1	Biased sampling techniques	406
12.1.1	Beyond Metropolis	406
12.1.2	Orientational bias	407
12.2	Chain molecules	413
12.2.1	Configurational-bias Monte Carlo	414
12.2.2	Lattice models	414
12.2.3	Off-lattice case	417
12.3	Generation of trial orientations	424
12.3.1	Strong intramolecular interactions	424
12.4	Fixed endpoints	432
12.4.1	Lattice models	432
12.4.2	Fully flexible chain	434

12.4.3 Strong intramolecular interactions	436
12.5 Beyond polymers	436
12.6 Other ensembles	440
12.6.1 Grand-canonical ensemble	440
12.7 Recoil growth	445
12.7.1 Algorithm	446
12.8 Questions and exercises	450
13. Accelerating Monte Carlo sampling	
13.1 Sampling intensive variables	455
13.1.1 Parallel tempering	457
13.1.2 Expanded ensembles	464
13.2 Noise on noise	467
13.3 Rejection-free Monte Carlo	468
13.3.1 Hybrid Monte Carlo	468
13.3.2 Kinetic Monte Carlo	469
13.3.3 Sampling rejected moves	471
13.4 Enhanced sampling by mapping	473
13.4.1 Machine learning and the rebirth of static Monte Carlo sampling	475
13.4.2 Cluster moves	479
13.4.2.1 Cluster moves on lattices	480
13.4.2.2 Off-lattice cluster moves	482
13.4.3 Early rejection method	484
13.4.4 Beyond detailed-balance	486
14. Time-scale-separation problems in MD	
14.1 Constraints	494
14.1.1 Constrained and unconstrained averages	499
14.1.2 Beyond bond constraints	505
14.2 On-the-fly optimization	506
14.3 Multiple time-step approach	509
15. Rare events	
15.1 Theoretical background	516
15.2 Bennett-Chandler approach	520
15.2.1 Dealing with holonomic constraints (Blue-Moon ensemble)	522
15.3 Diffusive barrier crossing	527
15.4 Path-sampling techniques	534
15.4.1 Transition-path sampling	535
15.4.1.1 Path ensemble	536
15.4.1.2 Computing rates	538
15.4.2 Path sampling Monte Carlo	543
15.4.3 Beyond transition-path sampling	546
15.4.4 Transition-interface sampling	546
15.5 Forward-flux sampling	547

15.5.1	Jump forward-flux sampling	548
15.5.2	Transition-path theory	549
15.5.3	Mean first-passage times	552
15.6	Searching for the saddle point	556
15.7	Epilogue	558
16.	Mesoscopic fluid models	
16.1	Dissipative-particle dynamics	561
16.1.1	DPD implementation	562
16.1.2	Smoothed dissipative-particle dynamics	566
16.2	Multi-particle collision dynamics	567
16.3	Lattice-Boltzmann method	569
 Part V		
Appendices		
A.	Lagrangian and Hamiltonian equations of motion	573
A.1	Action	573
A.2	Lagrangian	575
A.3	Hamiltonian	577
A.4	Hamilton dynamics and statistical mechanics	580
A.4.1	Canonical transformation	580
A.4.2	Symplectic condition	581
A.4.3	Statistical mechanics	583
B.	Non-Hamiltonian dynamics	587
C.	Kirkwood-Buff relations	591
C.1	Structure factor for mixtures	591
C.2	Kirkwood-Buff in simulations	593
D.	Non-equilibrium thermodynamics	595
D.1	Entropy production	595
D.1.1	Enthalpy fluxes	597
D.2	Fluctuations	597
D.3	Onsager reciprocal relations	600
E.	Non-equilibrium work and detailed balance	603
F.	Linear response: examples	607
F.1	Dissipation	607
F.2	Electrical conductivity	610
F.3	Viscosity	611
F.4	Elastic constants	612

G. Committor for 1d diffusive barrier crossing	617
G.1 1d diffusive barrier crossing	617
G.2 Computing the committor	618
H. Smoothed dissipative particle dynamics	621
H.1 Navier-Stokes equation and Fourier's law	621
H.2 Discretized SDPD equations	622
I. Saving CPU time	625
I.1 Verlet list	625
I.2 Cell lists	629
I.3 Combining the Verlet and cell lists	630
I.4 Efficiency	631
J. Some general purpose algorithms	637
J.1 Gaussian distribution	637
J.2 Selection of trial orientations	638
J.3 Generate random vector on a sphere	638
J.4 Generate bond length	639
J.5 Generate bond angle	640
J.6 Generate bond and torsion angle	641
 Part VI	
Repository¹	
K. Errata	645
L. Miscellaneous methods	647
M. Miscellaneous examples	649
N. Supporting information for case studies	651
O. Small research projects	653
P. Hints for programming	655
 Bibliography	657
Acronyms	695
Glossary	697
Index	701
Author index	715

¹ Repository is available in its entirety online at <https://www.elsevier.com/books-and-journals/book-companion/9780323902922>.

Online appendices

K. Errata	e1
L. Miscellaneous methods	e3
L.1 Higher-order integration schemes	e3
L.2 Surface tension via the pressure tensor	e4
L.3 Micro-canonical Monte Carlo	e5
L.4 Details of the Gibbs “ensemble”	e6
L.4.1 Free energy of the Gibbs ensemble	e6
L.4.1.1 Basic definitions and results for the canonical ensemble	e6
L.4.1.2 The free energy density in the Gibbs ensemble	e8
L.4.2 Graphical analysis of simulation results	e13
L.4.3 Chemical potential in the Gibbs ensemble	e17
L.4.4 Algorithms of the Gibbs ensemble	e18
L.5 Multi-canonical ensemble method	e18
L.6 Nosé-Hoover dynamics	e22
L.6.1 Nosé-Hoover dynamics equations of motion	e22
L.6.1.1 The Nosé-Hoover algorithm	e22
L.6.1.2 Nosé-Hoover chains	e24
L.6.1.3 The NPT ensemble	e30
L.6.2 Nosé-Hoover algorithms	e32
L.6.2.1 Canonical ensemble	e33
L.6.2.2 The isothermal-isobaric ensemble	e37
L.7 Ewald summation in a slab geometry	e41
L.8 Special configurational-bias Monte Carlo cases	e45
L.8.1 Generation of branched molecules	e45
L.8.2 Rebridging Monte Carlo	e48
L.8.3 Gibbs-ensemble simulations	e50
L.9 Recoil growth: justification of the method	e54
L.10 Overlapping distribution for polymers	e57
L.11 Hybrid Monte Carlo	e61
L.12 General cluster moves	e62
L.13 Boltzmann-sampling with dissipative particle dynamics	e65
L.14 Reference states	e67
L.14.1 Grand-canonical ensemble simulation	e67
L.14.1.1 Preliminaries	e67
L.14.1.2 Ideal gas	e68
L.14.1.3 Grand-canonical simulations	e68
M. Miscellaneous examples	e71
M.1 Gibbs ensemble for dense liquids	e71
M.2 Free energy of a nitrogen crystal	e71
M.3 Zeolite structure solution	e74

N. Supporting information for case studies	e75
N.1 Equation of state of the Lennard-Jones fluid-I	e75
N.2 Importance of detailed balance	e77
N.3 Why count the old configuration again?	e79
N.4 Static properties of the Lennard-Jones fluid	e79
N.5 Dynamic properties of the Lennard-Jones fluid	e82
N.6 Algorithms to calculate the mean-squared displacement	e83
N.7 Equation of state of the Lennard-Jones fluid	e85
N.8 Phase equilibria from constant-pressure simulations	e86
N.9 Equation of state of the Lennard-Jones fluid - II	e87
N.10 Phase equilibria of the Lennard-Jones fluid	e87
N.11 Use of Andersen thermostat	e89
N.12 Use of Nosé-Hoover thermostat	e90
N.13 Harmonic oscillator (I)	e92
N.14 Nosé-Hoover chain for harmonic oscillator	e93
N.15 Chemical potential: particle-insertion method	e93
N.16 Chemical potential: overlapping distributions	e94
N.17 Solid-liquid equilibrium of hard spheres	e96
N.18 Equation of state of Lennard-Jones chains	e101
N.19 Generation of trial configurations of ideal chains	e101
N.20 Recoil growth simulation of Lennard-Jones chains	e104
N.21 Multiple time step versus constraints	e107
N.22 Ideal gas particle over a barrier	e109
N.23 Single particle in a two-dimensional potential well	e112
N.24 Dissipative particle dynamics	e114
N.25 Comparison of schemes for the Lennard-Jones fluid	e116
O. Small research projects	e119
O.1 Adsorption in porous media	e119
O.2 Transport properties of liquids	e120
O.3 Diffusion in a porous medium	e120
O.4 Multiple-time-step integrators	e121
O.5 Thermodynamic integration	e122
P. Hints for programming	e125

Preface to the third edition

The third edition of Understanding Molecular Simulation is rather different from the previous one. The main reason why we opted for a different approach is that there have been massive changes in the way in which simulations are used. However, before discussing these changes, we first emphasize what has remained the same: this book is still about *understanding* molecular simulation. As we wrote in the preface to the first edition, “This book is *not* a molecular-simulation cookbook”, and it still isn’t.

It has been more than 20 years since the second edition was published, and it was only due to the lock-downs resulting from the Covid-19 pandemic that we found time to undertake the rather massive task of updating the book.

Twenty years is a long time and it is reasonable to ask: what, in the book, has changed? First and foremost, the community of people using molecular simulations has grown tremendously —and for many of them simulations are not the central focus of their research. Whereas at the time of the first edition of our book, many simulators were writing their own code, this group —whilst still very active— has become a minority. An appreciable fraction of all recent molecular simulations are reported in papers that report also, or even primarily, experimental work. This influx of new users went hand-in-hand with the increasing prevalence of a number of powerful simulation packages. It is hard to overstate the importance of this development, because it has removed an important barrier to the widespread use of molecular simulations.

At first sight, one might think that the use of simulation packages has decreased the need for “understanding” molecular simulation. However, we argue that the opposite is true: there is now a much larger need for information about the techniques that are used inside these packages, and about the choices that have to be made between different options in such packages. In this sense, understanding molecular simulations has become uncoupled from writing your own code.

But there is another factor that required a major rethink of our book, and it is directly related to the growth in computing power, or more specifically, to the staggering growth in the size of solid-state memories. In the early days of simulations, the available memory of computers could be measured in kilobytes. As a consequence, simulations were performed and the output was written on

tapes, or later on disks, which were subsequently read by separate codes for analyzing the data. Then computer memories grew, and much of the analysis could be (and was) carried out on the fly, because storing the data on an external device and analyzing them afterwards, would slow down the process. However, with the subsequent explosion in the capacity of cheap solid-state memories, it is now, once again, attractive to separate simulations from analysis. Hence, even if the simulations are carried out using a standard simulation package, there is a good chance that the users will write their own analysis code —and even if not, there is a good reason for separating the process of running the simulations from the data analysis. In the current edition of our book, we have reorganized the material to reflect this change in the way simulations are used.

Finally, there is an obvious reason for changing the structure of this third edition: there has been an explosion in the number of new algorithms. This is an important development, but it also created a challenge: this is a book to help *understanding* molecular simulation, not to review the many hundreds of algorithms that have been published. So, we don't. Rather, we try to explain the approach of certain classes of algorithms by choosing as an example an algorithm that is among the simplest in its category: but the simplest algorithm is rarely the fastest, nor the one that is most widely used. We stress that our choice of examples does not imply a “choice” against the more popular algorithms —just a way to be succinct. Succinctness is important, because we do not want this book to grow without bounds.

For this reason, we have moved a fair amount of older material that is of less general interest to a website¹ containing **Supplementary Information (SI)**, where it can still be consulted. The same website will also be used to maintain a list of the inevitable errata that keep on emerging more or less from the very moment the manuscript has been submitted.

The carbon footprint of computing

There are wildly varying estimates of the energy consumption associated with the global computing infrastructure. However, one thing is clear: the energy spent on computing is a substantial fraction of the total. At present, most electricity is generated using fossil fuels. What does that mean? According to Wikipedia, the energy consumption of a typical supercomputer in 2022 is in the Mega-Watt range, which would require the equivalent of several metric tonnes of fossil fuel per day. Moreover, the total amount of energy spent on computing is rising. Clearly, computing has to be made sustainable. On the “supply side”, this means that computers should be powered with electricity that is generated sustainably. But users can also contribute, by computing more efficiently. This is where algorithms can make a large difference, provided that the increased efficiency is not used to run ever larger simulations. Also, in computing, it is often true that “small is beautiful”.

¹ <https://www.elsevier.com/books-and-journals/book-companion/9780323902922>.

Acknowledgments

Special thanks are due to Rosalind Allen, Dick Bedeaux, Peter Bolhuis, Giovanni Bussi, Bingqing Cheng, Samuel Coles, Stephen Cox, Alex Cumberworth, John Chodera, Giovanni Cicotti, Christoph Dellago, Oded Farago, Susana Garcia, Bjørn Hafskjold, Kevin Jablonka, Signe Kjelstrup, Werner Krauth, Alessandro Laio, Ben Leimkuhler, Andrea Liu, Erik Luijten, Tony Maggs, Sauradeep Majumdar, Elias Moubarak, Beatriz Mouríño, Frank Noe, Miriam Pougin, Benjamin Rotenberg, David Shalloway, Michiel Sprik, Eric VandenEijnden, Joren Van Herck, Fred Verhoeckx, Patrick Warren, Peter Wirnsberger, and Xiaoqi Zhang for their suggestions for improvements in the text. Special thanks go to Jacobus van Meel, who provided the raw material for the image on the cover of the book, which shows the nucleation of a crystallite in cavity in a solid surface.

Some of our online exercises are based on Python code written by Manav Kumar and David Coker. We gratefully acknowledge them. However, they bear no responsibility for the mistakes that we introduced by changing their code

In addition, we thank everyone who pointed out the numerous mistakes and typos in the previous edition, in particular Giovanni Cicotti, Clemens Föerst, Viktor Ivanov, Brian Laird, Ting Li, Erik Luijten, Mat Mansell, Bortolo Mognetti, Nicy Nicy, Gerardo Odriozola, Arno Proeme, Mikhail Stukan, Petr Sulc, Krzysztof Szalewicz, David Toneian, Patrick Varilly, Patrick Warren, and Martijn Wehrens.

We have tried to resolve the issues that were raised. We stress that all remaining errors and misconceptions in the text are ours, and ours alone.

Daan Frenkel & Berend Smit, January 2023

Preface to the second edition

Why did we write a second edition? A minor revision of the first edition would have been adequate to correct the (admittedly many) typographical mistakes. However, many of the nice comments that we received from students and colleagues alike, ended with a remark of the type: “unfortunately, you don’t discuss topic x ”. And indeed, we feel that, after only five years, the simulation world has changed so much that the title of the book was no longer covered by the contents.

The first edition was written in 1995 and since then several new techniques have appeared or matured. Most (but not all) of the major changes in the second edition deal with these new developments. In particular, we have included a section on:

- Transition path sampling and diffusive barrier crossing to simulate rare events
- Dissipative particle dynamic as a coarse-grained simulation technique
- Novel schemes to compute the long-ranged forces
- Discussion on Hamiltonian and non-Hamiltonian dynamics in the context of constant-temperature and constant-pressure Molecular Dynamics simulations
- Multiple-time-step algorithms as an alternative for constraints
- Defects in solids
- The pruned-enriched Rosenbluth sampling, recoil growth, and concerted rotations for complex molecules
- Parallel tempering for glassy Hamiltonians

We have updated some of the examples to include also recent work. Several new Examples have been added to illustrate recent applications.

We have taught several courses on Molecular Simulation, based on the first edition of this book. As part of these courses, Dr. Thijs Vlugt prepared many *Questions*, *Exercises*, and *Case Studies*, most of which have been included in the present edition. Some additional exercises can be found on the Web. We are very grateful to Thijs Vlugt for the permission to reproduce this material.

Many of the advanced Molecular Dynamics techniques described in this book are derived using the Lagrangian or Hamilton formulations of classical mechanics. However, many chemistry and chemical engineering students are not familiar with these formalisms. While a full description of classical mechanics

is clearly beyond the scope of the present book, we have added an Appendix that summarizes the necessary essentials of Lagrangian and Hamiltonian mechanics.

Special thanks are due to Giovanni Ciccotti, Rob Groot, Gavin Crooks, Thijs Vlugt, and Peter Bolhuis for their comments on parts of the text. In addition, we thank everyone who pointed out mistakes and typos, in particular Drs. J.B. Freund, R. Akkermans, and D. Moroni.

Daan Frenkel & Berend Smit, 2001

Preface to first edition

This book is not a computer simulation cookbook. Our aim is to explain the physics that is behind the “recipes” of molecular simulation. Of course, we also give the recipes themselves, because otherwise the book would be too abstract to be of much practical use. The scope of this book is necessarily limited: we do not aim to discuss all aspects of computer simulation. Rather, we intend to give a unified presentation of those computational tools that are currently used to study the equilibrium properties and, in particular, the phase behavior of molecular and supramolecular substances. Moreover, we intentionally restrict the discussion to simulations of *classical* many-body systems, even though some of the techniques mentioned can be applied to quantum systems as well. And, within the context of classical many-body systems, we restrict our discussion to the modeling of systems at, or near, equilibrium.

The book is aimed at readers who are active in computer simulation or are planning to become so. Computer simulators are continuously confronted with questions concerning the choice of technique, because a bewildering variety of computational tools is available. We believe that, to make a rational choice, a good understanding of the physics behind each technique is essential. Our aim is to provide the reader with this background.

We should state at the outset that we consider some techniques to be more useful than others, and therefore our presentation is biased. In fact, we believe that the reader is well served by the fact that we do not present all techniques as equivalent. However, whenever we express our personal preference, we try to back it up with arguments based in physics, applied mathematics, or simply experience. In fact, we mix our presentation with practical examples that serve a twofold purpose: first, to show how a given technique works in practice, and second, to give the reader a flavor of the kind of phenomena that can be studied by numerical simulation.

The reader will also notice that two topics are discussed in great detail, namely simulation techniques to study first-order phase transitions, and various aspects of the configurational-bias Monte Carlo method. The reason why we devote so much space to these topics is not that we consider them to be more important than other subjects that get less coverage, but rather because we feel that, at present, the discussion of both topics in the literature is rather fragmented.

The present introduction is written for the nonexpert. We have done so on purpose. The community of people who perform computer simulations is rapidly expanding as computer experiments become a general research tool. Many of the new simulators will use computer simulation as a tool and will not be primarily interested in techniques. Yet, we hope to convince those readers who consider a computer simulation program a *black box*, that the inside of the black box is interesting and, more importantly, that a better understanding of the working of a simulation program may greatly improve the efficiency with which the black box is used.

In addition to the theoretical framework, we discuss some of the practical tricks and rules of thumb that have become “common” knowledge in the simulation community and are routinely used in a simulation. Often, it is difficult to trace back the original motivation behind these rules. As a result, some “tricks” can be very useful in one case yet result in inefficient programs in others. In this book, we discuss the rationale behind the various tricks, in order to place them in a proper context. In the main text of the book we describe the theoretical framework of the various techniques. To illustrate how these ideas are used in practice we provide Algorithms, Case Studies and Examples.

Algorithms

The description of an algorithm forms an essential part of this book. Such a description, however, does not provide much information on how to implement the algorithm efficiently. Of course, details about the implementation of an algorithm can be obtained from a listing of the complete program. However, even in a well-structured program, the code contains many lines that, although necessary to obtain a working program, tend to obscure the essentials of the algorithm that they express. As a compromise solution, we provide a pseudo-code for each algorithm. These pseudo-codes contain only those aspects of the implementation directly related to the particular algorithm under discussion. This implies that some aspects that are essential for using this pseudo-code in an actual program have to be added. For example, the pseudo-codes consider only the x directions; similar lines have to be added for the y and z direction if the code is going to be used in a simulation. Furthermore, we have omitted the initialization of most variables.

Case Studies

In the Case Studies, the algorithms discussed in the main text are combined in a complete program. These programs are used to illustrate some elementary aspects of simulations. Some Case Studies focus on the problems that can occur in a simulation or on the errors that are sometimes made. The complete listing of the FORTRAN codes that we have used for the Case Studies is accessible to the reader through the Internet.¹

¹ The original FORTRAN code of the Case Studies can be found at: <https://doi.org/10.5281/zenodo.7503798> For any updates, we refer to our GitHub site: <https://github.com/UnderstandingMolecularSimulation>.

Examples

In the Examples, we demonstrate how the techniques discussed in the main text are used in an application. We have tried to refer as much as possible to research topics of current interest. In this way, the reader may get some feeling for the type of systems that can be studied with simulations. In addition, we have tried to illustrate in these examples how simulations can contribute to the solution of “real” experimental or theoretical problems.

Many of the topics that we discuss in this book have appeared previously in the open literature. However, the Examples and Case Studies were prepared specifically for this book. In writing this material, we could not resist including a few computational tricks that, to our knowledge, have not been reported in the literature.

In computer science it is generally assumed that any source code over 200 lines contains at least one error. The source codes of the Case Studies contain over 25,000 lines of code. Assuming we are no worse than the average programmer this implies that we have made at least 125 errors in the source code. If you spot these errors and send them to us, we will try to correct them (we can not promise this!). It also implies that, before you use part of the code yourself, you should convince yourself that the code is doing what you expect it to do.

In the light of the previous paragraph, we must add the following disclaimer:

We make no warranties, express or implied, that the programs contained in this work are free of error, or that they will meet your requirements for any particular application. They should not be relied on for solving problems whose incorrect solution could result in injury, damage, or loss of property. The authors and publishers disclaim all liability for direct or consequential damages resulting from your use of the programs.

Although this book and the included programs are copyrighted, we authorize the readers of this book to use parts of the programs for their own use, provided that proper acknowledgment is made.

Finally, we gratefully acknowledge the help and collaboration of many of our colleagues. In fact, many dozens of our colleagues collaborated with us on topics described in the text. Rather than listing them all here, we mention their names at the appropriate place in the text. Yet, we do wish to express our gratitude for their input. Moreover, Daan Frenkel should like to acknowledge numerous stimulating discussions with colleagues at the FOM Institute for Atomic and Molecular Physics in Amsterdam and at the van ’t Hoff Laboratory of Utrecht University, while Berend Smit gratefully acknowledges discussions with colleagues at the University of Amsterdam and Shell. In addition, several colleagues helped us directly with the preparation of the manuscript, by reading the text or part thereof. They are Giovanni Ciccotti, Mike Deem, Simon de Leeuw, Toine Schlijper, Stefano Ruffo, Maria-Jose Ruiz, Guy Verbist and Thijs Vlugt. In addition, we thank Klaas Esselink and Sami Karaborni for the cover

figure. We thank them all for their efforts. In addition we thank the many readers who have drawn our attention to errors and omissions in the first print. But we stress that the responsibility for the remainder of errors in the text is ours alone.

Daan Frenkel & Berend Smit, 1996

Chapter 1

Introduction

(Pre)history of computer simulation

It usually takes decades rather than years before a fundamentally new invention finds widespread application. For computer simulation, the story is rather different. Computer simulation started as a tool to exploit the electronic computing machines that had been developed during and after the Second World War. These machines had been built to perform the very heavy computation involved in the development of nuclear weapons and code-breaking. In the early 1950s, electronic computers became partly available for non-military use and this was the beginning of the discipline of computer simulation. W.W. Wood [1] recalls: “When the Los Alamos MANIAC became operational in March 1952, Metropolis was interested in having as broad a spectrum of problems as possible tried on the machine, in order to evaluate its logical structure and demonstrate the capabilities of the machine.”

The strange thing about computer simulation is that it is also a discovery, albeit a delayed discovery that grew slowly after the introduction of the technique. In fact, *discovery* is probably not the right word, because it does not refer to a new insight into the working of the natural world but into our description of nature. Working with computers has provided us with a new metaphor for the laws of nature: they carry as much (and as little) information as algorithms. For any nontrivial algorithm (i.e., loosely speaking, one that cannot be solved analytically), you cannot predict the outcome of a computation simply by looking at the program, although it often is possible to make precise statements about the general nature (e.g., the symmetry) of the result of the computation. Similarly, the basic laws of nature, as we know them, have the unpleasant feature that they are expressed in terms of equations that we cannot solve exactly, except in a few very special cases. If we wish to study the motion of more than two interacting bodies, even the relatively simple laws of Newtonian mechanics become essentially unsolvable. That is to say, they cannot be solved analytically, using only a pencil and the back of the proverbial envelope. However, using a computer, we can get the answer to any desired accuracy. Most of materials science deals with the properties of systems of many atoms or molecules. *Many* almost always means more than two; usually, very much more. So if we wish to compute the properties of a liquid (to take a particularly nasty example), there is no hope of finding the answer exactly using only pencil and paper.

Before computer simulation appeared on the scene, there was only one way to predict the properties of a molecular substance, namely by making use of

a theory that provided an approximate description of that material. Such approximations are inevitable precisely because there are very few systems for which the equilibrium properties can be computed exactly (examples are the ideal gas, the harmonic crystal, and a number of lattice models, such as the two-dimensional Ising model for ferromagnets). As a result, most properties of real materials were predicted on the basis of approximate theories (examples are the van der Waals equation for dense gases, the Debye-Hückel theory for electrolytes, and the Boltzmann equation to describe the transport properties of dilute gases). Given sufficient information about the intermolecular interactions, these theories will provide us with an estimate of the properties of interest. Unfortunately, our knowledge of the intermolecular interactions of all but the simplest molecules is also quite limited. This leads to a problem if we wish to test the validity of a particular theory by comparing directly to experiment. If we find that theory and experiment disagree, it may mean that our theory is wrong, or that we have an incorrect estimate of the intermolecular interactions, or both.

Clearly, it would be nice if we could obtain essentially exact results for a given model system without having to rely on approximate theories. Computer simulations allow us to do precisely that. On the one hand, we can now compare the calculated properties of a model system with those of an experimental system: if the two disagree, our model is inadequate; that is, we have to improve on our estimate of the intermolecular interactions. On the other hand, we can compare the result of a simulation of a given model system with the predictions of an approximate analytical theory applied to the same model. If we now find that theory and simulation disagree, we know that the *theory* is flawed. So, in this case, the computer simulation plays the role of the experiment designed to test the theory. This method of screening theories before we apply them to the real world is called a *computer experiment*. This application of computer simulation is of tremendous importance. It has led to the revision of some very respectable theories, some of them dating back to Boltzmann. And it has changed the way in which we construct new theories. Nowadays it is becoming increasingly rare that a theory is applied to the real world before being tested by computer simulation. The simulation then serves a twofold purpose: it gives the theoretician a feeling for the physics of the problem, and it generates some “exact” results that can be used to test the quality of the theory to be constructed. Computer experiments have become standard practice, to the extent that they now provide the first (and often the last) test of a new theoretical result.

But note that the computer as such offers us no understanding, only numbers. And, as in a real experiment, these numbers have statistical errors. So what we get out of a simulation is never directly a theoretical relation. As in a real experiment, we still have to extract the useful information. To take a not very realistic example, suppose we were to use the computer to measure the pressure of an ideal gas as a function of density. This example is unrealistic because the volume dependence of the ideal-gas pressure has, in fact, been well-known since the work of Boyle and Gay-Lussac. The Boyle-Gay-Lussac law states that the

TABLE 1.1 Simulated equation of state of an ideal gas.

$\rho k_B T$	P
1	1.03 ± 0.04
2	1.99 ± 0.03
3	2.98 ± 0.05
4	4.04 ± 0.03
5	5.01 ± 0.04

product of volume and pressure of an ideal gas is constant. Now suppose we were to measure this product by computer simulation. We might, for instance, find the set of experimental results in Table 1.1. The data suggest that P equals $\rho k_B T$, but no more than that. It is left to us to infer the conclusions.

The early history of computer simulation (see ref. [2]) illustrates this role of computer simulation. Some areas of physics appeared to have little need for simulation because very good analytical theories were available, e.g., to predict the properties of dilute gases or of nearly harmonic crystalline solids. However, in other areas, few if any exact theoretical results were known, and progress was much hindered by the lack of unambiguous tests to assess the quality of approximate theories. A case in point was the theory of dense liquids. Before the advent of computer simulations, the only way to model liquids was by mechanical simulation [3–5] of large assemblies of macroscopic spheres (e.g., ball bearings). Then the main problem becomes how to arrange these balls in the same way as atoms in a liquid. Much work on this topic was done by the famous British scientist J.D. Bernal, who built and analyzed such mechanical models for liquids. Actually, it would be fair to say that the really tedious work of analyzing the resulting three-dimensional structures was done by his research students, such as the unfortunate Miss Wilkinson whose research assignment was to identify all distinct local packing geometries of plastic foam spheres: she found that there were at least 197.

It is instructive to see how Bernal built some of his models. The following quote from the 1962 Bakerian lecture describes Bernal's attempt to build a ball-and-spoke model of a liquid [5]:

...I took a number of rubber balls and stuck them together with rods of a selection of different lengths ranging from 2.75 to 4 inch. I tried to do this in the first place as casually as possible, working in my own office, being interrupted every five minutes or so and not remembering what I had done before the interruption. However,...

Subsequent models were made, for instance, by pouring thousands of steel balls from ball bearings into a balloon. It should be stressed that these mechanical models for liquids were, in some respects, quite realistic. However, the analysis

of the structures generated by mechanical simulation was very laborious and, in the end, had to be performed by computer anyway.

In view of the preceding, it is hardly surprising that, when electronic computers were, for the first time, made available for unclassified research, numerical simulation of dense liquids was one of the first problems to be tackled. In fact, the first simulation of a liquid was carried out by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller on the MANIAC computer at Los Alamos [6], using (or, more properly, introducing) the Metropolis Monte Carlo (MC) method. The name *Monte Carlo simulation* had been coined earlier by Metropolis and Ulam (see Ref. [7]), because the method makes heavy use of computer-generated random numbers. Almost at the same time, Fermi, Pasta, Ulam and Tsingou [8,9] performed their famous numerical study of the dynamics of an anharmonic, one-dimensional crystal. The first proper Molecular Dynamics (MD) simulations were reported in 1956 by Alder and Wainwright [10] at Livermore, who studied the dynamics of an assembly of hard spheres. The first MD simulation of a model for a “real” material was reported in 1959 (and published in 1960) by the group led by Vineyard at Brookhaven [11], who simulated radiation damage in crystalline Cu (for a historical account, see [12]). The first MD simulation of a real liquid (argon) was reported in 1964 by Rahman at Argonne [13]. After that, computers were increasingly becoming available to scientists outside the US government labs, and the practice of simulation started spreading to other continents [14–17]. Much of the methodology of computer simulations has been developed since then, although it is fair to say that the basic algorithms for MC and MD have hardly changed since the 1950s.

The most common application of computer simulations is to predict the properties of materials. The need for such simulations may not be immediately obvious. After all, it is much easier to measure the freezing point of water than to extract it from a computer simulation. The point is, of course, that it is easy to measure the freezing point of water at 1 atmosphere but often very difficult and, therefore, expensive to measure the properties of real materials at very high pressures or temperatures. The computer does not care: it does not go up in smoke when you ask it to simulate a system at 10,000 K. In addition, we can use computer simulation to predict the properties of materials that have not yet been made. And finally, computer simulations are increasingly used in data analysis. For instance, a very efficient technique for obtaining structural information about macromolecules from 2d-NMR is to feed the experimental data into a Molecular Dynamics simulation and let the computer find the structures that are both energetically favorable and compatible with the available NMR data.

Initially, such simulations were received with a certain amount of skepticism, and understandably so. Simulation did not fit into the existing idea that whatever was not experiment had to be theory. In fact, many scientists much preferred to keep things the way they were: theory for the theoreticians and experiments for the experimentalists and no computers to confuse the issue. However, this

position became untenable, as is demonstrated by the following autobiographical quote of George Vineyard [12], who was the first to study the dynamics of radiation damage by numerical simulation:

...In the summer of 1957 at the Gordon Conference on Chemistry and Physics of Metals, I gave a talk on radiation damage in metals.... After the talk, there was a lively discussion.... Somewhere the idea came up that a computer might be used to follow in more detail what actually goes on in radiation damage cascades. We got into quite an argument, some maintaining that it wasn't possible to do this on a computer, others that it wasn't necessary. John Fisher insisted that the job could be done well enough by hand, and was then goaded into promising to demonstrate. He went off to his room to work. Next morning he asked for a little more time, promising to send me the results soon after he got home. After about two weeks, not having heard from him, I called and he admitted that he had given up. This stimulated me to think further about how to get a high-speed computer into the game in place of John Fisher....

Finally, computer simulation can be used as a purely exploratory tool. This sounds strange. One would be inclined to say that one cannot “discover” anything by simulation because you can never get out what you have not put in. Computer discoveries, in this respect, are not unlike mathematical discoveries. In fact, before computers were actually available this kind of numerical charting of unknown territory was never considered.

The best way to explain it is to give an explicit example. In the mid-1950s, one of the burning questions in statistical mechanics was this: can crystals form in a system of spherical particles that have a harsh short-range repulsion, but no mutual attraction whatsoever? In a very famous computer simulation, Alder and Wainwright [18] and Wood and Jacobson [19] showed that such a system does indeed have a first-order freezing transition. This is now accepted wisdom, but at the time it was greeted with skepticism. For instance, at a meeting in New Jersey in 1957, a group of some 15 very distinguished scientists (among whom were 2 Nobel laureates) discussed the issue. When a vote was taken as to whether hard spheres can form a stable crystal, it appeared that half the audience simply could not believe this result. However, the work of the past 30 years has shown that harsh repulsive forces really determine the structural properties of a simple liquid and that attractive forces are in a sense of secondary importance.

More about the early history of molecular simulations can be found in the book by Battimelli and Ciccotti [2].

Machine Learning

The power of computers has grown by more than 15 orders of magnitude since the early 1950s. This unprecedented growth has driven fundamental, qualitative changes in the way in which computers are used. One of these changes is the unstoppable growth of **Machine Learning (ML)** in almost every field, including Molecular Simulations, and we are only at the beginning. In this book we had

to make a choice: we stress the growing importance of ML, but the focus of the book cannot be on the technical aspects of ML: others are much better qualified to describe this than we are.

Moreover, many of the current applications of ML in simulations are in the construction of cheaper force fields. This is a topic of great practical importance, but the focus of this book is on algorithms, not force fields. However, in some cases (and there will be many more), ML is used to construct novel algorithms or used to facilitate the analysis of high-dimensional data sets. In section 13.4.1, we will touch upon algorithm design using ML. Yet, in view of the very rapid developments in the field, the reader should be aware that the examples that we discuss are just snapshots: suggestive, but patchy.

We only hint at the use of ML in the context of data analysis (see section 15.7), not because it is too small, but because it is too big for this book. Yet, a few words are in place. The essence of constructing models is data reduction: as humans, we are continually confronted with sensory overload, and our brains have developed efficient analysis tools to reduce the myriad of data into a highly simplified (and occasionally oversimplified) representation of reality. In Science, a “model” can be viewed as such a low-dimensional representation of the system that we try to understand. For instance, the ideal gas law collapses a large number of measurements of the pressure as a function of density and temperature onto a plane ($P = N\rho T$). That is dimensionality reduction. In a simulation, as in many other fields, we are increasingly confronted with high-dimensional data sets that we would like to represent with a low-dimensional function. Finding such a representation (if it exists) is often too hard for our brains. This is where ML, in particular the version that makes use of auto-encoders, comes in. It allows us to identify what combination of variables, e.g., specific structures, correlate with a particular observation. ML does not replace model-building, but it is a tool that helps us construct new models, just as the observed linear relation between P and ρT suggested the ideal gas law. Clearly, in this context, there is a huge variety of ML techniques that can be used, and the list is growing rapidly. For more details on this fast-moving field, we refer the reader to the relevant literature: a snapshot that was almost up-to-date in 2022 can be found in ref. [20].

Suggested reading

As stated at the outset, the present book does not cover all aspects of computer simulation. Readers who are interested in aspects of computer simulation not covered in this book are referred to one of the many books on the subject, some old, some more recent. We list only a few:

- Allen and Tildesley, *Computer Simulation of Liquids* [21]
- Haile, *Molecular Dynamics Simulations: Elementary Methods* [22]
- Leimkuhler and Matthews, *Molecular Dynamics* [23]
- Tuckerman, *Statistical mechanics: theory and molecular simulation* [24]

- Landau and Binder, *A Guide to Monte Carlo Simulations in Statistical Physics* [25]
- Rapaport, *The Art of Molecular Dynamics Simulation* [26]
- Newman and Barkema, *Monte Carlo Methods in Statistical Physics* [27]

Also of interest in this context are the books by Hockney and Eastwood [28], Hoover [29,30], Vesely [31], and Heermann [32] and the book by Evans and Morriss [33] for the theory and simulation of transport phenomena. The latter book is out of print and has been made available in electronic form.¹ A book by Peters [34] deals specifically with reaction rates and rare events.

A general discussion of Monte Carlo sampling can be found in Koonin's *Computational Physics* [35]. As the title indicates, this is a textbook on computational physics in general, as is the book by Gould and Tobochnik [36]. In contrast, the book by Kalos and Whitlock [37] focuses specifically on the Monte Carlo method. A good discussion of (quasi) random-number generators can be found in *Numerical Recipes* [38], while Ref. [37] gives a detailed discussion of tests for random-number generators. An early discussion of Monte Carlo simulations with emphasis on techniques relevant to atomic and molecular systems may be found in two articles by Valleau and Whittington in *Modern Theoretical Chemistry* [39,40]. The books by Binder [41,42] and Mouritsen [43] emphasize the application of MC simulations to discrete systems, phase transitions and critical phenomena. In addition, there exist several very useful proceedings of summer schools [44–49] on computer simulation.

How to use this book

The main part of this book is a description of the theoretical framework of the various simulation techniques. To illustrate how these ideas are used in practice we provide Algorithms, Illustration, Examples, and Case Studies.

Algorithms

Throughout the text, we use *pseudo-codes* to explain algorithms. In earlier editions of this book, the pseudo-codes looked very much like FORTRAN, and, although FORTRAN is a powerful language to write code that has to run fast, many readers will not be familiar with it. We considered switching to a widely used language, such as Python. In the end, we chose not to, because one of the reasons why Python is popular is that it is very compact: compactness facilitates *writing* Python, but not *reading* it. In our pseudo-codes, we want to spell out what many languages hide: we sacrifice compactness to show the inner workings of a piece of coding. Our pseudo-codes will make skilled programmers cringe. However, for the sake of clarity, we follow Boltzmann's advice that “Elegance should be left to shoemakers and tailors”.

Even so, in our pseudo-codes, we need some shorthand notation. As much as possible, we have used standard notation: For comparisons, we use $>$, $<$, \geq ,

¹ See <http://doi.org/10.22459/SMNL.08.2007>.

\leq , and \neq . To test an equality, we use `==`, and the logical *and* is written as `&`. We use `%` to denote the *modulo* operation. To round a floating point number x down *in magnitude* to an integer, we use `int(x)`. Finally, we use the symbol `R` to denote random numbers, uniform between 0 and 1. The most common mathematical functions (`sin`, `cos`, `exp`, `ln`) should be self-explanatory. For the typical programming instructions we use instructions like `for 1 ≤ x ≤ y do ... enddo` or `if xy then ... endif`.

Many of our examples use **functions** that may or may not return information back to the main program. Our function codes are really fragments because we leave out all the bits that set the values of parameters and that specify how variables/arrays (e.g., the particle coordinates, velocities, and forces) are shared between different parts of a program. However, in certain cases, in particular, when the same function is called more than once, but with different inputs, we show the control parameters explicitly. In all other cases, the default assumption of the reader should be that whatever parameters and variables are used in a **functions**, are shared with those parts of the program that generate the input or need the result.

For the sake of compactness, most sample algorithms consider a one-dimensional system. The extension to higher dimensions is usually straightforward, and is occasionally shown in the pseudo-code.

Illustrations

In the Illustrations, we demonstrate how the techniques discussed in the main text are used in an application. We have tried to refer as much as possible to research topics that optimally illustrate the technical aspect. In this way, the reader may get some feeling for the type of systems that can be studied with simulations. In addition, these Illustrations should give the readers some insights into the diversity of topics for which simulations are used.

Examples

In the Examples, the algorithms discussed in the main text are combined in a complete program. These programs are used to illustrate some elementary aspects of simulations. Some Examples focus on the problems that can occur in a simulation or on the errors that are sometimes made. For those of you that are interested in running these codes, the original FORTRAN code of the Case Studies can be found at: <https://doi.org/10.5281/zenodo.7503798>. For any updates, we refer to our GitHub site: <https://github.com/UnderstandingMolecularSimulation>.

Case Studies

In the SI all Case Studies that have been included in the previous editions, can be found together with the codes that were used to generate the data. The current edition contains a selection of these Case Studies which are presented as Examples.

Chapter 2

Thermodynamics and statistical mechanics

Molecular simulations can be used to predict the structural, dynamical, and thermodynamic properties of a classical many-body system. The number of properties that can be computed once we know the natural time evolution of an N -particle system is effectively unlimited, but not all quantities that can be computed in a simulation correspond to properties that are experimentally accessible. To give a specific example: in a **Molecular Dynamics** simulation of liquid water, we could compute the time evolution of the positions and velocities of all molecules in the liquid. However, such information cannot be compared with experiments, because no known experimental technique can provide us with such detailed information. Rather, typical experiments sample properties that are averaged over a large number of particles and usually also averaged over the time of the measurement. This feature of experiments is no accident: measurements should yield outcomes that are reproducible by repeating the experiment many times, or by running it for a long time. If we wish to use computer simulation as the numerical counterpart of experiments,¹ we must know what kind of computational measurements yield reproducible outcomes.

As in experiments, the reproducible properties of a model system depend on a small number of macroscopic control variables that characterize the macroscopic state in which the system was prepared: e.g., its energy, volume, and number of particles. The theoretical framework of thermodynamics provides us with rules that relate the variation of macroscopic observables with the change in the “thermodynamic” control parameters such as volume or applied external fields. Different sets of thermodynamic parameters correspond to different experiments, e.g., we can either consider a system at fixed density or at fixed pressure. Hence, before we start our discussion of simulation methods, we need to introduce the language of thermodynamics. We also need the language of statistical mechanics, which provides the link between the microscopic state of a system and its macroscopic properties.

As this is not a book about either of these subjects, our introduction to thermodynamics and statistical mechanics will be oversimplified and should be seen

¹ Simulations can also be used to test theoretical predictions, but such predictions also tend to focus on reproducible properties, even though they may not always correspond to experimentally accessible quantities.

either as a refresher, or as an incentive to learn more about the subject from the many excellent textbooks.

In what follows, we present a brief overview of classical thermodynamics and a quick (and slightly dirty) derivation of the basic expressions of statistical mechanics. The aim of these derivations is only to show that there is nothing mysterious about concepts such as phase space, temperature, entropy, and many of the other statistical mechanical objects that will appear time and again in the remainder of this book.

If you are familiar with thermodynamics, you may wish to skip the next section. But remember the quote attributed to Sommerfeld:

“Thermodynamics is a funny subject. The first time you go through it, you don’t understand it at all. The second time you go through it, you think you understand it, except for one or two points. The third time you go through it, you know you don’t understand it, but by that time you are so used to the subject, it doesn’t bother you anymore.”

2.1 Classical thermodynamics

Thermodynamics is difficult because it seems so abstract. However, we should always bear in mind that thermodynamics is based on experimental observations. For instance, the First Law of thermodynamics expresses the empirical observation that energy is conserved, even though it can be converted into various forms. The internal energy of a system can be changed by performing work w on the system or by transferring an amount of heat q . It is meaningless to speak about the total amount of heat in a system, or the total amount of work. This is not mysterious: it is just as meaningless to speak about the number of train-travelers and the number of pedestrians in a train-station: people enter the station as pedestrians, and exit as train-travelers or vice versa. However, if we add the sum of the changes in the number of train-travelers and pedestrians, then we obtain the change in the number of *people* in the station. And this quantity is well-defined. Similarly, the sum of q and w is equal to the change in the internal energy E of the system

$$dE = q + w. \quad (2.1.1)$$

This is the First Law of thermodynamics.

The Second Law seems more abstract, but it is not. The Second Law is based on the *experimental* observation that it is impossible to make an engine that works by converting heat from a single heat bath (i.e., a large reservoir in equilibrium) into work. This observation is equivalent to another —equally empirical— observation, namely that heat can *never* flow spontaneously (i.e., without performing work) from a cold reservoir to a warmer reservoir. This statement is a bit more subtle than it seems because, before we have defined temperature, we can only distinguish *hotter* and *colder* by looking at the direction of heat flow. What the Second Law says is that it is *never* possible to

sustain a spontaneous heat flow in the “wrong” direction (e.g., if heat can flow spontaneously from system A to system B , and from system B to system C , then heat cannot flow spontaneously from C to A). How do we get from such a seemingly trivial statement to something as abstract as entropy? This is most easily achieved by introducing the concept of a *reversible* heat engine, i.e., an engine that is completely free of internal dissipative energy losses.

A reversible engine is, as the word suggests, an engine that can be operated in reverse. During one cycle (a sequence of steps that is completed when the engine is returned to its original state) this engine takes in an amount of heat q_1 from a hot reservoir, converts part of it into work w , and delivers a remaining amount of heat q_2 to a cold reservoir. The reverse process is that, by performing an amount of work w , we can take an amount of heat q_2 from the cold reservoir and deliver an amount of heat q_1 to the hot reservoir. Reversible engines are an idealization because in any real engine, there will be additional frictional losses. However, the ideal reversible engine can be approximated arbitrarily closely by a real engine if, at every stage, the real engine is sufficiently close to equilibrium. As the engine is returned to its original state at the end of one cycle, its internal energy E has not changed. Hence, the First Law tells us that

$$\Delta E = q_1 - (w + q_2) = 0, \quad (2.1.2)$$

or

$$q_1 = w + q_2. \quad (2.1.3)$$

Now consider the “efficiency”, η , of the engine $\eta \equiv w/q_1$ —i.e., the amount of work delivered per amount of heat taken in. At first, one might think that η depends on the precise design of our reversible engine. However, this is not true. η is the same for *all* reversible engines operating between the same two reservoirs. To demonstrate this, we show that if different engines could have different values for η then we would contradict the Second Law in its form “heat can never spontaneously flow from a cold to a hot reservoir”. Suppose, therefore, that we have another reversible engine that takes in an amount of heat q'_1 from the hot reservoir, delivers the same amount of work w , and then delivers an amount of heat q'_2 to the cold reservoir. Let us denote the efficiency of this engine by η' . Now, we use the work generated by the engine with the highest efficiency (say η) to drive the second engine in reverse. The amount of heat delivered to the hot reservoir by the second engine is

$$q'_1 = w/\eta' = q_1(\eta/\eta'), \quad (2.1.4)$$

where we have used $w = q_1\eta$. As, by assumption, $\eta' < \eta$ it follows that $q'_1 > q_1$. Hence *there would be a net heat flow from the cold reservoir into the hot reservoir*. But this contradicts the Second Law of thermodynamics. Therefore we must conclude that the efficiency of all reversible heat engines operating between the same reservoirs is *identical*. The efficiency only depends on the

temperatures t_1 and t_2 of the reservoirs (the temperatures t could be measured in any scale, e.g., in Fahrenheit or Réaumur, as long as it is single-valued. As $\eta(t_1, t_2)$ depends only on the temperature in the reservoirs, then so does the ratio $q_2/q_1 = 1 - \eta$. Let us call this ratio $R(t_2, t_1)$. Now suppose that we have a reversible engine that consists of two stages: one working between reservoir 1 and 2, and the other between 2 and 3. In addition, we have another reversible engine that works directly between 1 and 3. As both engines must be equally efficient, it follows that

$$R(t_3, t_1) = R(t_3, t_2)R(t_2, t_1). \quad (2.1.5)$$

This can only be true in general if $R(t_1, t_2)$ is of the form

$$R(t_2, t_1) = \frac{f(t_2)}{f(t_1)}, \quad (2.1.6)$$

where $f(t)$ is an, as yet unknown function of our measured temperature. What we do now is to introduce an “absolute” or thermodynamic temperature T given by

$$T \equiv f(t). \quad (2.1.7)$$

Then, it immediately follows that

$$\frac{q_2}{q_1} = R(t_2, t_1) = \frac{T_2}{T_1}. \quad (2.1.8)$$

Note that the thermodynamic temperature could just as well have been defined as $c \times f(t)$. In practice, c has been fixed such that, around room temperature, 1 degree in the absolute (Kelvin) scale is equal to 1 degree Celsius. But that choice is, of course, purely historical and—as it will turn out later—a bit unfortunate.

Why do we need all this? We need it to introduce entropy, the most mysterious of all thermodynamic quantities. To do so, note that Eq. (2.1.8) can be written as

$$\frac{q_1}{T_1} = \frac{q_2}{T_2}, \quad (2.1.9)$$

where q_1 is the heat that flows *in* reversibly at the high temperature T_1 , and q_2 is the heat that flows *out* reversibly at the low temperature T_2 . We see therefore that, during a complete cycle, the difference between q_1/T_1 and q_2/T_2 is zero. Recall that, at the end of a cycle, the internal energy of the system has not changed. Now Eq. (2.1.9) tells us that there is also another quantity, denoted by S , which is unchanged when we restore the system to its original state. Following Clausius, we use the name “entropy” for S .

In Thermodynamics, quantities such as S that are unchanged if we return a system to its original state, are called *state functions*. We do not know what S is,

but we do know how to compute its change. In the above example, the change in S was given by $\Delta S = (q_1/T_1) - (q_2/T_2) = 0$. In general, the change in entropy of a system due to the reversible addition of an infinitesimal amount of heat δq_{rev} from a reservoir at temperature T is

$$dS = \frac{\delta q_{\text{rev}}}{T}. \quad (2.1.10)$$

We also note that S is *extensive*. That means that the total entropy of two non-interacting systems, is equal to the sum of the entropies of the individual systems. Consider a system with a fixed number of particles² N and a fixed volume V . If we transfer an infinitesimal amount of heat δq to this system, then the change in the internal energy of the system, dE is equal to δq . Hence,

$$\left(\frac{\partial S}{\partial E}\right)_{V,N} = \frac{1}{T}. \quad (2.1.11)$$

The most famous, though not the most intuitively obvious, statement of the Second Law of Thermodynamics is that *A spontaneous change in a closed system, i.e., a system that exchanges neither energy nor particles with its environment, can never lead to a decrease of the entropy*. Hence, in equilibrium, the entropy of a closed system is at a maximum. The argument behind this sweeping statement is simple: consider a system with energy E , volume V and number of particles N that *is* in equilibrium. We denote the entropy of this system by $S_0(E, V, N)$. In equilibrium, all spontaneous changes that *can* happen, *have* happened. Now suppose that we want to change something in this system —for instance, we increase the density in one half of the system and decrease it in the other half. As the system was in equilibrium, this change does not occur spontaneously. Hence, in order to realize this change, we must perform a certain amount of *work*, w (for instance, by placing a piston in the system and moving it). We assume that this work is performed reversibly in such a way that E , the total energy of the system, stays constant (and also V and N). The First Law tells us that we can only keep E constant if, while doing the work, we allow an amount of heat q , to flow out of the system, such that $q = w$. Eq. (2.1.10) tells us that when an amount of heat q flows reversibly out of the system, the entropy S of the system must decrease. Let us denote the entropy of this constrained state by $S_1(E, V, N) < S_0(E, V, N)$. Having completed the change in the system, we insulate the system thermally from the rest of the world, and we remove the constraint that kept the system in its special state (taking the example of the piston: we make an opening in the piston). Now the system goes back *spontaneously* (and irreversibly) to equilibrium. However, no work is done, and no

² Thermodynamics does not assume anything about atoms or molecules. In thermodynamics, we would specify the total mass (a macroscopic quantity) of a given species in the system, rather than the number of particles. But when we talk about Statistical Mechanics or simulation techniques, we will always specify the amount of matter in a system by the number of molecules. By doing the same while discussing thermodynamics, we can keep the same notation throughout.

heat is transferred. Hence the final energy E is equal to the original energy (and V and N are also constant). This means that the system is now back in its *original* equilibrium state and its entropy is once more equal to $S_0(E, V, N)$. The entropy change during this spontaneous change is equal to $\Delta S = S_0 - S_1$. But, as $S_1 < S_0$, it follows that $\Delta S > 0$. As this argument is quite general, we have indeed shown that any spontaneous change in a closed system leads to an increase in the entropy. Hence, in equilibrium, the entropy of a closed system is at a maximum.

We can now combine the First Law and the Second Law to arrive at an expression for the energy change of a thermodynamic system. We consider an infinitesimal reversible change in the system due to heat transfer and work. The First Law states:

$$dE = q + w.$$

For a reversible change, we can write $q = TdS$. In the case of w , there are many ways in which work can be performed on a system, e.g., compression, electrical polarization, magnetic polarization, elastic deformation, etc. Here we will focus on one such form of work, namely work due to a volume change against an external pressure P . In that case, the work performed *on* the system during an infinitesimal volume change dV is $w = -PdV$, and the First Law can be written as

$$dE = TdS - PdV.$$

However, there is another, important way in which we can change the energy of a system, namely by transferring matter into or out of the system. For convenience, we consider a system containing only one species. As before, the number of molecules of this species is denoted by N . As we (reversibly) change the number of molecules in the system at constant V and S , the energy of the system changes: $dE = \mu dN$. This expression defines the constant of proportionality, μ , the “chemical potential”

$$\mu \equiv \left(\frac{\partial E}{\partial N} \right)_{S,V}. \quad (2.1.12)$$

We note that, for consistency with the rest of the book, we have defined the chemical potential in terms of the number of molecules. In classical thermodynamics, the chemical potential is defined as

$$\mu^{\text{thermo}} \equiv \left(\frac{\partial E}{\partial n} \right)_{S,V}, \quad (2.1.13)$$

where n denotes the number of moles. The relation between the thermodynamic μ^{thermo} and the molecular μ is simple:

$$\mu^{\text{thermo}} = N_A \mu,$$

where N_A denotes Avogadro's number. The generalization of Eq. (2.1.12) to multi-component systems is straightforward and will be encountered later. We can now write down the most frequently used form of the First Law of Thermodynamics

$$dE = TdS - PdV + \mu dN . \quad (2.1.14)$$

Often we use Eq. (2.1.14) in the form

$$dS = \frac{1}{T}dE + \frac{P}{T}dV - \frac{\mu}{T}dN , \quad (2.1.15)$$

which implies that:

$$\left(\frac{\partial S}{\partial V} \right)_{E,N} = \frac{P}{T}$$

and

$$\left(\frac{\partial S}{\partial N} \right)_{E,V} = -\frac{\mu}{T}.$$

We already knew that

$$\left(\frac{\partial S}{\partial E} \right)_{V,N} = \frac{1}{T}.$$

It is important to make a distinction between thermodynamic properties that are *extensive* and those that are *intensive*. Intensive properties do not depend on the size of the system under consideration. Examples are temperature, pressure, and chemical potential. If we combine two identical systems in the same thermodynamic state, then the temperature, pressure, and chemical potential of the resulting system are the same as that of the constituent systems. In contrast, energy, entropy, volume, and number of particles are extensive. This means that they scale with the system size. Now assume that we construct a thermodynamic system by combining a large number of infinitesimal systems. Then the extensivity of E , S , V , and N implies that, for the resulting system, we have

$$E = TS - PV + \mu N . \quad (2.1.16)$$

Now consider a small variation in E :

$$dE = dTS - dPV + d\mu N = TdS + SdT - PdV - VdP + \mu dN + Nd\mu .$$

If we combine this with the First Law of thermodynamics, we find:

$$0 = SdT - VdP + Nd\mu . \quad (2.1.17)$$

This is an important relation because it shows that T , P , and μ are dependent variables. Two of them suffice to specify the thermodynamic state of the system.

However, in addition, we always need (at least) one extensive thermodynamic variable to specify the size of the system: T , P , and μ are intensive and therefore they do not contain that information.

From this point on, we can derive all of thermodynamics, except one law: the so-called Third Law of Thermodynamics. The Third Law can be formulated in a number of ways. The shortest version states that the entropy of the equilibrium state of a pure substance at $T = 0$ is equal to zero. However, the fact that the value of the entropy at $T = 0$ must be zero follows from considerations outside thermodynamics. The Third Law is not as “basic” as the First and the Second, and, anyway, we shall soon get a more direct interpretation of its meaning.

2.1.1 Auxiliary functions

Eq. (2.1.14) expresses the First Law of thermodynamics as a relation between the variations in E with those in S , V , and N . Sometimes, it is more convenient to use other independent variables, e.g., the temperature instead of the entropy, the pressure instead of the volume, or the chemical potential instead of the number of particles. “Convenient” variables are those that can be controlled in a given experiment. There is a simple procedure to recast the First Law in terms of these other variables.

Enthalpy

For instance, if we use S , P , and N as the independent variables we can carry out a so-called *Legendre* transform, which allows us to replace the energy with a new state function that is a function of S , P , and N . This function is called the Enthalpy (H), defined as $H \equiv E + PV$. Clearly

$$dH = dE + dPV = TdS - PdV + \mu dN + PdV + VdP = TdS + VdP + \mu dN, \quad (2.1.18)$$

showing that the independent variables controlling the enthalpy are S , P , and N .

Helmholtz free energy

Similarly, we can introduce a function F , called the Helmholtz free energy, defined as $F \equiv E - TS$. As in the case of the enthalpy, it is easy to show that

$$dF = -SdT - PdV + \mu dN. \quad (2.1.19)$$

Gibbs free energy

The Gibbs free energy G is defined as $F + PV$, and it satisfies

$$dG = -SdT + VdP + \mu dN. \quad (2.1.20)$$

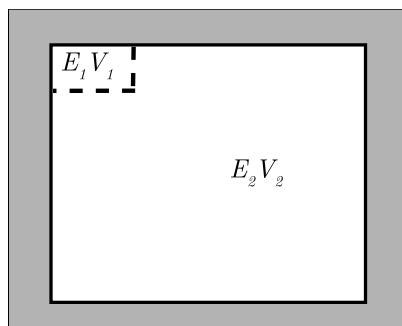


FIGURE 2.1 An isolated system consisting of two boxes 1 and 2 that each have a fixed volume, the two subsystems can exchange heat but the subsystem 2 is much larger than system 1 and therefore acts as a heat bath.

Grand Potential

Finally, we can introduce the Grand Potential, Φ , which is $\Phi \equiv F - \mu N$, satisfying

$$d\Phi = -SdT - PdV - Nd\mu. \quad (2.1.21)$$

However, for homogeneous systems, we rarely use the symbol Φ for the Grand Potential, because, if the pressure of a system is well-defined, $F - \mu N = -PV$, we can replace Φ by $-PV$.³

Auxiliary functions and the Second Law

For a closed N -particle system with a given internal energy E and volume V , equilibrium is reached when the entropy S is at a maximum. To describe experiments at conditions other than constant N , V , E , we must reformulate the Second Law of thermodynamics, because, in general, S will not be at a maximum in equilibrium if we hold P or T constant. Fortunately, we can use the original Second Law of thermodynamics to derive the condition for thermodynamic equilibrium under conditions other than constant E , V , and N .

Let us consider the system shown in Fig. 2.1. The total system is isolated and has a fixed volume. In the system, we have a subsystem 1, which is much smaller than subsystem 2 (we will refer to the larger system as the “heat bath”). If we allow subsystems 1 and 2 to exchange energy, the total energy of the combined system is still conserved. Hence, we can apply the Second Law of thermodynamics to the combined system, i.e., the total entropy of the combined system must be at a maximum in equilibrium. As long as there is a net energy flux between systems 1 and 2, the total system is not yet in equilibrium and we

³ The pressure is not a convenient variable for describing the state of confined or porous systems. For such systems, it is best to use $\Phi \equiv F - \mu N$.

must have

$$\Delta S_{\text{tot}} = \Delta S_1 + \Delta S_2 \geq 0. \quad (2.1.22)$$

As the heat bath (subsystem 2) is much larger than subsystem 1, its temperature T will not change when a small amount of energy is exchanged with subsystem 1. Using $dS = \delta q_{\text{rev}}/T$ (Eq. (2.1.10)), we can then write:

$$\Delta S_2 = \frac{\Delta E_2}{T}. \quad (2.1.23)$$

As the total energy is conserved, (i.e., $\Delta E_2 = -\Delta E_1$), we can then write:

$$\Delta S_1 + \Delta S_2 = \frac{1}{T} (T \Delta S_1 - \Delta E_1) \geq 0. \quad (2.1.24)$$

This equation expresses the condition for equilibrium in terms of the properties of subsystem 1. The only effect of the heat bath is that it imposes the temperature T . Note that the quantity that is varied in Eq. (2.1.24) is nothing else than the Helmholtz free energy (see Eq. (2.1.19))

$$F_1(N, V, T) \equiv E_1 - TS_1.$$

Then the Second Law, in the form given by Eq. (2.1.24), implies that, for a system in contact with a heat bath,

$$-\frac{1}{T} \Delta F_1 \geq 0. \quad (2.1.25)$$

In other words: when an N -particle system in a volume V is in contact with a heat bath at a (positive) temperature T , then a spontaneous change can never increase its Helmholtz free energy:

$$dF \leq 0. \quad (2.1.26)$$

Similarly, we can define two systems that can not only exchange of energy, but also change their volumes in such a way that the total volume remains constant (see Fig. 2.2). As before, the combined system is isolated and its total volume is fixed. Then we can again apply the Second Law of thermodynamics to the combined system. As system 2 is again assumed to be much larger than system 1, its temperature and pressure do not change when the energy and volume of system 1 change. The entropy change of system 2 is therefore given by

$$\Delta S_2 = \frac{\Delta E_2}{T} + \frac{P \Delta V_2}{T}. \quad (2.1.27)$$

As the total energy is conserved ($\Delta E_2 = -\Delta E_1$) and the total volume remains constant ($\Delta V_2 = -\Delta V_1$), we can write for the total entropy change:

$$\Delta S_1 + \Delta S_2 = \frac{1}{T} (T \Delta S_1 - \Delta E_1 - P \Delta V_1) \geq 0, \quad (2.1.28)$$

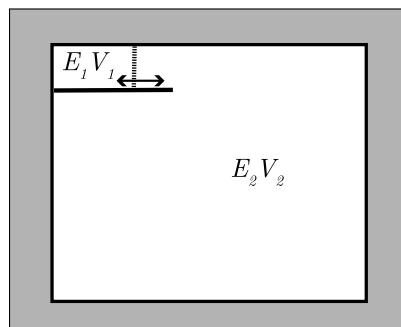


FIGURE 2.2 An isolated system consisting of two boxes 1 and 2 that can exchange heat and change their volume in such a way that the total volume and energy remains constant. System 2 is much larger compared to system 1 so it can act as a heat bath that exerts a constant pressure on system 1.

or $(dG)/T \leq 0$, where dG is the variation in the Gibbs free energy (Eq. (2.1.20)). In this equation, we have again expressed the inequality in terms of the properties of system 1 only. A spontaneous change in a system at constant temperature and pressure can never increase its Gibbs free energy.

2.1.2 Chemical potential and equilibrium

Thus far, we have considered systems in contact with a reservoir at constant temperature and constant pressure. Let us now consider what happens if we bring a system in contact with a “particle reservoir,” i.e., a system at constant chemical potential. Clearly, as T , P , and μ are linearly dependent, we cannot consider a system in contact with a reservoir at constant T , P , μ , because these variables are not enough to fix the size of the system. Hence, when considering a system in contact with a particle reservoir, we should fix at least one extensive variable. The most convenient choice is to fix the volume V . So, we will consider a system (1) of volume V in contact with a reservoir (system 2) at constant T and μ . As before, we can use the Second Law of thermodynamics as applied to the combined system, to derive the equilibrium condition for system 1 under conditions where this system can exchange heat and particles with the reservoir (system 2):

$$\Delta S_{\text{tot}} = \Delta S_1 + \Delta S_2 = \Delta S_1 - \frac{\Delta E_1}{T} + \frac{\mu \Delta N_1}{T} \geq 0 \quad (2.1.29)$$

or

$$\Delta S_{\text{tot}} = \frac{\Delta(TS_1 - E_1 + \mu N_1)}{T} \geq 0, \quad (2.1.30)$$

or, using Eq. (2.1.21), $\Delta\Phi \leq 0$. Hence, at constant T , V , and μ , Φ is at a minimum in equilibrium.

We have now obtained the conditions for equilibrium for some of the conditions of the greatest practical importance, namely:

Equilibrium at constant N, V, E :	S is maximal
Equilibrium at constant N, V, T :	F is minimal
Equilibrium at constant N, P, T :	G is minimal
Equilibrium at constant μ , V, T :	Φ is minimal

(2.1.31)

Using the definitions of F , G , and Φ , we can write down expressions for an infinitesimal variation in each of these quantities. For instance:

$$dF = dE - dTS.$$

Using the First Law, we can rewrite this as

$$dF = TdS - PdV + \mu dN - TdS - SdT = -SdT - PdV + \mu dN.$$

Similarly, we can write:

$$\begin{aligned} dS &= \frac{1}{T}dE + \frac{P}{T}dV - \frac{\mu}{T}dN \\ dF &= -SdT - PdV + \mu dN \\ dG &= -SdT + VdP + \mu dN \\ d\Phi &= -SdT - PdV - Nd\mu . \end{aligned} \quad (2.1.32)$$

In what follows, we will replace Φ by $-PV$, unless explicitly stated otherwise. Eq. (2.1.32), in combination with the equilibrium conditions (2.1.31), are extremely important because they specify the conditions under which a system is in thermodynamic (chemical or phase) equilibrium. Later, we shall make extensive use of the conditions for phase equilibrium.

Next, consider a closed system containing two subsystems. The total volume of the system $V = V_1 + V_2$ is fixed. Similarly, $N_1 + N_2$ and $E_1 + E_2$ are fixed. These conditions imply that $dV_1 = -dV_2$, $dN_1 = -dN_2$ and $dE_1 = -dE_2$. The Second Law tells us that, in equilibrium, the total entropy of the system $S_{\text{tot}} = S_1 + S_2$, must be an extremum (note that S_{tot} is not fixed). Hence, the derivatives of S_{tot} with respect to E_1 , N_1 and V_1 must vanish, i.e.:

$$\begin{aligned} \left(\frac{\partial (S_1 + S_2)}{\partial E_1} \right) &= \left(\frac{\partial S_1}{\partial E_1} \right) - \left(\frac{\partial S_2}{\partial E_2} \right) = 0 \\ \left(\frac{\partial (S_1 + S_2)}{\partial V_1} \right) &= \left(\frac{\partial S_1}{\partial V_1} \right) - \left(\frac{\partial S_2}{\partial V_2} \right) = 0 \\ \left(\frac{\partial (S_1 + S_2)}{\partial N_1} \right) &= \left(\frac{\partial S_1}{\partial N_1} \right) - \left(\frac{\partial S_2}{\partial N_2} \right) = 0 . \end{aligned} \quad (2.1.33)$$

If we combine Eq. (2.1.33) with the expression for dS , Eq. (2.1.32), we obtain

$$\begin{aligned}\frac{1}{T_1} &= \frac{1}{T_2} \\ \frac{P_1}{T_1} &= \frac{P_2}{T_2} \\ \frac{\mu_1}{T_1} &= \frac{\mu_2}{T_2}.\end{aligned}\quad (2.1.34)$$

The first condition implies thermal equilibrium between the two systems, i.e., $T_1 = T_2 \equiv T$. Then the second condition simply implies $P_1 = P_2$, and the third is $\mu_1 = \mu_2$.⁴ Eq. (2.1.34) is the starting point for all free-energy-based calculations to locate the point where two systems (two phases) are in equilibrium (see Chapter 8).

2.1.3 Energy, pressure, and chemical potential

One of the aims of molecular simulations is to compute the thermodynamic properties of a system based on our knowledge of the interactions between constituent molecules. In subsequent chapters, we discuss how the relevant expressions can be derived from Statistical Mechanics. Here we focus on one general feature: in all cases, we will start from the thermodynamic definitions of the quantities to be computed. For instance, the starting point for computing the pressure, P , is a thermodynamic relation of the type:

$$P = -\left(\frac{\partial F}{\partial V}\right)_{N,T}.\quad (2.1.35)$$

This is the expression that we would use if we consider a system at fixed N , V , and T . However, if we would consider a system at constant N , V and E , we would use:

$$\frac{P}{T} = \left(\frac{\partial S}{\partial V}\right)_{N,E}.$$

Analogous expressions can be written down for other thermodynamic variables and other conditions. For instance, for a system at constant N , V and T , the energy is given by the following thermodynamic relation:

$$\begin{aligned}E &= F + TS = F - T\left(\frac{\partial F}{\partial T}\right)_{V,N} \\ &= \left(\frac{\partial F/T}{\partial 1/T}\right)_{V,N}.\end{aligned}\quad (2.1.36)$$

⁴ For a multicomponent mixture, the chemical potential μ^α of each component α in the mixture must be equal in the two subsystems: $\mu_1^\alpha = \mu_2^\alpha$.

We can also obtain the chemical potential, μ , from F using:

$$\mu = \left(\frac{\partial F}{\partial N} \right)_{T,V} . \quad (2.1.37)$$

As most Monte Carlo simulations are carried out at constant N , V , and T , we will use these relations extensively.

Relations between partial molar derivatives

All extensive thermodynamic variables X of a multi-component system can be written as:

$$X = \sum_i N_i x_i ,$$

where x_i is the partial derivative of the quantity X with respect to N_i , the number of particles of species i , at constant P , T , and N_j :

$$x_i \equiv \left(\frac{\partial X}{\partial N_i} \right)_{P,T,\{N_j\}} .$$

For instance, in the case that $X = S$. Then

$$S = \sum_i N_i s_i ,$$

where, we have for the molar entropy of component i , s_i

$$s_i \equiv \left(\frac{\partial S}{\partial N_i} \right)_{P,T,\{N_j\}} .$$

It is important to note that it makes a difference what thermodynamic variables we keep constant. Clearly,

$$s_i \neq \left(\frac{\partial S}{\partial N_i} \right)_{E,V,\{N_j\}} = \mu_i / T .$$

But there is no inconsistency: when we add a particle of species i at constant P and T , the internal energy changes by an amount e_i and the volume by an amount v_i . To reduce the system to constant energy and volume, we should therefore compute

$$\begin{aligned} \left(\frac{\partial S}{\partial N_i} \right)_{E,V,\{N_j\}} &= \left(\frac{\partial S}{\partial N_i} \right)_{P,T,\{N_j\}} - \left(\frac{\partial S}{\partial V} \right)_{E,\{N\}} v_i - \left(\frac{\partial S}{\partial E} \right)_{V,\{N\}} e_i \\ &= s_i - \frac{P v_i}{T} - \frac{e_i}{T} \end{aligned}$$

$$= \frac{Ts_i - Pv_i - e_i}{T} = -\frac{\mu_i}{T} = -\frac{1}{T} \left(\frac{\partial G}{\partial N_i} \right)_{P,T,\{N_j\}}.$$

2.2 Statistical thermodynamics

In the previous section, we introduced the framework of thermodynamics. Thermodynamics is a phenomenological theory: it provides relations between experimentally observable quantities. However, it does not predict these quantities on the basis of a microscopic model. Statistical Mechanics provides the link between the microscopic description of a system of interacting atoms or molecules and the prediction of thermodynamical observables such as pressure or chemical potential. For all but the simplest systems, the statistical mechanical expressions for the thermodynamical observables are too complex to be evaluated analytically. However, in many cases, numerical simulations will allow us to obtain accurate estimates of the quantities of interest.

2.2.1 Basic assumption

Most of the computer simulations that we discuss are based on the assumption that classical mechanics can be used to describe the motions of atoms and molecules. This assumption leads to a great simplification in almost all calculations, and it is therefore most fortunate that it is justified in many cases of practical interest. Surprisingly, it turns out to be easier to derive the basic laws of statistical mechanics using the language of quantum mechanics. We will follow this route of least resistance. In fact, for our derivation, we need only little quantum mechanics. Specifically, we need the fact that a quantum mechanical system can be found in different states. For the time being, we limit ourselves to quantum states that are eigenvectors of the Hamiltonian \mathcal{H} of the system (i.e., energy eigenstates). For any such state $|i\rangle$, we have that $\mathcal{H}|i\rangle = E_i|i\rangle$, where E_i is the energy of state $|i\rangle$. Most examples discussed in quantum mechanics textbooks concern systems with only a few degrees of freedom (e.g., the one-dimensional harmonic oscillator or a particle in a box). For such systems, the degeneracy of energy levels will be small. However, for the systems that are of interest to statistical mechanics (i.e., systems with $\mathcal{O}(10^{23})$ particles), the degeneracy of energy levels is super-astronomically large. In what follows, we denote by Ω the number of eigenstates with energy E of a system of N particles in a volume V , $\Omega = \Omega(E, V, N)$. We now express the basic assumption of statistical mechanics as follows: a system with fixed N , V , and E is equally likely to be found in any of its $\Omega(E)$ eigenstates. Much of statistical mechanics follows from this simple (but nontrivial) assumption.

To see this, let us again consider a system with total energy E that consists of two weakly interacting subsystems. In this context, *weakly interacting* means that the subsystems can exchange energy but that we can write the total energy of the system as the sum of the energies E_1 and E_2 of the subsystems. There are

many ways in which we can distribute the total energy over the two subsystems such that $E_1 + E_2 = E$. For a given choice of E_1 , the total number of degenerate states of the system is $\Omega_1(E_1) \times \Omega_2(E_2)$. Note that the total number of states is the product of the number of states in the individual systems. In what follows, it is convenient to have a measure of the degeneracy of the subsystems that is additive. A logical choice is to take the (natural) logarithm of the degeneracy. Hence:

$$\ln \Omega(E_1, E - E_1) = \ln \Omega_1(E_1) + \ln \Omega_2(E - E_1). \quad (2.2.1)$$

We assume that subsystems 1 and 2 can exchange energy. What is the most likely distribution of the energy? We know that *every* energy state of the total system is equally likely. But the number of eigenstates corresponding to a given distribution of the energy over the subsystems depends very strongly on the value of E_1 . We wish to know the most likely value of E_1 , that is, the one that maximizes $\ln \Omega(E_1, E - E_1)$. The condition for this maximum is that

$$\left(\frac{\partial \ln \Omega(E_1, E - E_1)}{\partial E_1} \right)_{N, V, E} = 0 \quad (2.2.2)$$

or, in other words,

$$\left(\frac{\partial \ln \Omega_1(E_1)}{\partial E_1} \right)_{N_1, V_1} = \left(\frac{\partial \ln \Omega_2(E_2)}{\partial E_2} \right)_{N_2, V_2}. \quad (2.2.3)$$

We introduce the shorthand notation

$$\beta(E, V, N) \equiv \left(\frac{\partial \ln \Omega(E, V, N)}{\partial E} \right)_{N, V}. \quad (2.2.4)$$

With this definition, we can write Eq. (2.2.3) as

$$\beta(E_1, V_1, N_1) = \beta(E_2, V_2, N_2). \quad (2.2.5)$$

Clearly, if initially, we put all energy in system 1 (say), there will be energy transfer from system 1 to system 2 until Eq. (2.2.3) is satisfied. From that moment on, no net energy flows from one subsystem to the other, and we say that the two subsystems are in (thermal) equilibrium. When this equilibrium is reached, $\ln \Omega$ of the total system is at a maximum. This suggests that $\ln \Omega$ is somehow related to the thermodynamic entropy S of the system. As we have seen in the previous section, the Second Law of thermodynamics states that the entropy of a system N , V , and E is at its maximum when the system has reached thermal equilibrium.

To establish the relation between $\ln \Omega$ and the entropy we could simply start by assuming that the entropy is equal to $\ln \Omega$ and then check whether predictions based on this assumption agree with experiment. If we do so, we find that the answer is “not quite”: for (unfortunate) historical reasons (entropy already had

units before statistical mechanics had been created), entropy is not simply equal to $\ln \Omega$; rather we have

$$S(N, V, E) \equiv k_B \ln \Omega(N, V, E), \quad (2.2.6)$$

where k_B is Boltzmann's constant, which in S.I. units has the value $1.380649 \cdot 10^{-23}$ J/K. With this identification, we see that our assumption that all degenerate eigenstates of a quantum system are equally likely immediately implies that, in thermal equilibrium, the entropy of a composite system is at a maximum. It would be a bit premature to refer to this statement as the Second Law of thermodynamics, as we have not yet demonstrated that the present definition of entropy is, indeed, equivalent to the thermodynamic definition. We simply take an advance on this result.

The next thing to note is that thermal equilibrium between subsystems 1 and 2 implies that $\beta_1 = \beta_2$. In everyday life, we have another way to express the same thing: we say that two bodies brought into thermal contact are in equilibrium if their temperatures are the same. This suggests that β must be related to the absolute temperature. The thermodynamic definition of temperature is given by Eq. (2.1.11), or

$$1/T = \left(\frac{\partial S}{\partial E} \right)_{V,N}. \quad (2.2.7)$$

If we use the same definition here, we find that

$$\beta = 1/(k_B T). \quad (2.2.8)$$

2.2.2 Systems at constant temperature

Now that we have a statistical mechanical definition of temperature, we can consider what happens if, as in section 2.1.1, we have a small system (denoted by system 1) in thermal equilibrium with a large heat bath (system 2) (see Fig. 2.1). The total system is closed; that is, the total energy $E = E_1 + E_2$ is fixed. Suppose that the system 1 is prepared in one specific quantum state i with energy E_i . The bath then has an energy $E_2 = E - E_i$ and the degeneracy of the bath is given by $\Omega_2(E - E_i)$. Clearly, the degeneracy of the bath determines the probability P_i to find system 1 in state i :

$$P_i = \frac{\Omega_2(E - E_i)}{\sum_j \Omega_2(E - E_j)}. \quad (2.2.9)$$

To compute $\Omega_2(E - E_i)$, we assume that our bath, system 2, is much larger than system 1, which allows us to expand $\ln \Omega_2(E - E_i)$ around $E_i = 0$:

$$\ln \Omega_2(E - E_i) = \ln \Omega_2(E) - E_i \frac{\partial \ln \Omega_2(E)}{\partial E} + \mathcal{O}(1/E), \quad (2.2.10)$$

and, using Eqs. (2.2.6) and (2.2.7),

$$\ln \Omega_2(E - E_i) = \ln \Omega_2(E) - E_i/k_B T + \mathcal{O}(1/E). \quad (2.2.11)$$

If we insert this result in Eq. (2.2.9), and take the limit $E \rightarrow \infty$, we get

$$P_i = \frac{\exp(-E_i/k_B T)}{\sum_j \exp(-E_j/k_B T)}. \quad (2.2.12)$$

This is the well-known Boltzmann distribution for a system at temperature T . Knowledge of the energy distribution allows us to compute the average energy $\langle E \rangle$ of the system at the given temperature T :

$$\begin{aligned} \langle E \rangle &= \sum_i E_i P_i \\ &= \frac{\sum_i E_i \exp(-E_i/k_B T)}{\sum_j \exp(-E_j/k_B T)} \\ &= -\frac{\partial \ln \sum_i \exp(-E_i/k_B T)}{\partial 1/k_B T} \\ &= -\frac{\partial \ln Q}{\partial 1/k_B T}, \end{aligned} \quad (2.2.13)$$

where, in the last line, we have defined the partition function, $Q \equiv Q(N, V, T)$. If we compare Eq. (2.2.13) with the thermodynamic relation Eq. (2.1.36)

$$E = \frac{\partial F/T}{\partial 1/T},$$

where F is the Helmholtz free energy, we see that F is related to the partition function Q :

$$F = -k_B T \ln Q = -k_B T \ln \left(\sum_i \exp(-E_i/k_B T) \right). \quad (2.2.14)$$

Strictly speaking, F is fixed only up to a constant. Or, what amounts to the same thing, the reference point of the energy can be chosen arbitrarily. In what follows, we can use Eq. (2.2.14) without loss of generality. The relation between the Helmholtz free energy and the partition function is often more convenient to use than the relation between $\ln \Omega$ and the entropy. As a consequence, Eq. (2.2.14) is the workhorse of equilibrium statistical mechanics.

2.2.3 Towards classical statistical mechanics

Thus far, we have formulated statistical mechanics in purely quantum mechanical terms. The entropy is related to the density-of-states of a system with energy

E , volume V , and number of particles N . Similarly, the Helmholtz free energy is related to the partition function \mathcal{Q} , a sum over all quantum states i of the Boltzmann factor $\exp(-E_i/k_B T)$. To be specific, let us consider the average value of some observable A . We know the probability that a system at temperature T will be found in an energy eigenstate with energy E_i and we can therefore compute the thermal average of A as

$$\langle A \rangle = \frac{\sum_i \exp(-E_i/k_B T) \langle i | A | i \rangle}{\sum_j \exp(-E_j/k_B T)}, \quad (2.2.15)$$

where $\langle i | A | i \rangle$ denotes the expectation value of the operator A in quantum state i . This equation suggests how we should go about computing thermal averages: first we solve the Schrödinger equation for the (many-body) system of interest, and next we compute the expectation value of the operator A for all those quantum states that have a non-negligible statistical weight. Unfortunately, this approach is doomed for all but the simplest systems. First of all, we cannot hope to solve the Schrödinger equation for an arbitrary many-body system. And second, even if we could, the number of quantum states that contribute to the average in Eq. (2.2.15) would be so huge ($\mathcal{O}(10^{10^{25}})$) that a numerical evaluation of all expectation values would be inconceivable. Fortunately, Eq. (2.2.15) can be simplified to a more workable expression in the classical limit. To this end, we first rewrite Eq. (2.2.15) in a form that is independent of the specific basis set. We note that $\exp(-E_i/k_B T) = \langle i | \exp(-\mathcal{H}/k_B T) | i \rangle$, where \mathcal{H} is the Hamiltonian of the system. Using this relation, we can write

$$\begin{aligned} \langle A \rangle &= \frac{\sum_i \langle i | \exp(-\mathcal{H}/k_B T) A | i \rangle}{\sum_j \langle j | \exp(-\mathcal{H}/k_B T) | j \rangle} \\ &= \frac{\text{Tr} \exp(-\mathcal{H}/k_B T) A}{\text{Tr} \exp(-\mathcal{H}/k_B T)}, \end{aligned} \quad (2.2.16)$$

where Tr denotes the trace of the operator. As the value of the trace of an operator does not depend on the choice of the basis set, we can compute thermal averages using any basis set we like. Preferably, we use simple basis sets, such as the set of eigenfunctions of the position or the momentum operator. Next, we use the fact that the Hamiltonian \mathcal{H} is the sum of a kinetic part \mathcal{K} and a potential part \mathcal{U} . The kinetic energy operator is a quadratic function of the momenta of all particles. As a consequence, momentum eigenstates are also eigenfunctions of the kinetic energy operator. Similarly, the potential energy operator is a function of the particle coordinates. Matrix elements of \mathcal{U} , therefore, are most conveniently computed in a basis set of position eigenfunctions. However, $\mathcal{H} = \mathcal{K} + \mathcal{U}$ itself is not diagonal in either basis set nor is $\exp[-\beta(\mathcal{K} + \mathcal{U})]$. However, if we could replace $\exp(-\beta\mathcal{H})$ by $\exp(-\beta\mathcal{K})\exp(-\beta\mathcal{U})$, then we could simplify Eq. (2.2.16) considerably. In general, we cannot make this replacement because

$$\exp(-\beta\mathcal{K})\exp(-\beta\mathcal{U}) = \exp\{-\beta[\mathcal{K} + \mathcal{U} + \mathcal{O}([\mathcal{K}, \mathcal{U}])]\},$$

where $[\mathcal{K}, \mathcal{U}]$ is the commutator of the kinetic and potential energy operators: $\mathcal{O}([\mathcal{K}, \mathcal{U}])$ stands for all terms containing commutators and higher-order commutators of \mathcal{K} and \mathcal{U} . It is easy to verify that the commutator $[\mathcal{K}, \mathcal{U}]$ is of order \hbar ($\hbar \equiv h/(2\pi)$, where h is Planck's constant). Hence, in the limit $\hbar \rightarrow 0$, we may ignore the terms of order $\mathcal{O}([\mathcal{K}, \mathcal{U}])$. In that case, we can write

$$\text{Tr} \exp(-\beta \mathcal{H}) \approx \text{Tr} \exp(-\beta \mathcal{U}) \exp(-\beta \mathcal{K}). \quad (2.2.17)$$

If we use the notation $|r\rangle$ for eigenvectors of the position operator and $|k\rangle$ for eigenvectors of the momentum operator, we can express Eq. (2.2.17) as

$$\text{Tr} \exp(-\beta \mathcal{H}) = \sum_{r,k} \langle r | e^{-\beta \mathcal{U}} | r \rangle \langle r | k \rangle \langle k | e^{-\beta \mathcal{K}} | k \rangle \langle k | r \rangle. \quad (2.2.18)$$

All matrix elements can be evaluated directly:

$$\langle r | \exp(-\beta \mathcal{U}) | r \rangle = \exp\left[-\beta \mathcal{U}(\mathbf{r}^N)\right],$$

where $\mathcal{U}(\mathbf{r}^N)$ on the right-hand side is no longer an operator but a function of the coordinates of all N particles. Here, and in what follows, we denote this set of coordinates by \mathbf{r}^N . Similarly,

$$\langle k | \exp(-\beta \mathcal{K}) | k \rangle = \exp\left[-\beta \sum_{i=1}^N p_i^2 / (2m_i)\right],$$

where $p_i = \hbar k_i$, and

$$\langle r | k \rangle \langle k | r \rangle = 1/V^N,$$

where V is the volume of the system and N the number of particles. Finally, we can replace the sum over states by an integration over all coordinates and momenta. The final result is

$$\begin{aligned} \text{Tr} \exp(-\beta \mathcal{H}) &\approx \frac{1}{h^{dN} N!} \int d\mathbf{p}^N d\mathbf{r}^N \exp\left\{-\beta \left[\sum_i p_i^2 / (2m_i) + \mathcal{U}(\mathbf{r}^N)\right]\right\} \\ &\equiv Q_{\text{classical}}, \end{aligned} \quad (2.2.19)$$

where d is the dimensionality of the system and the last line defines the classical partition function. The factor $1/N!$ corrects for the fact that any permutation of indistinguishable particles corresponds to the same macroscopic state.⁵

⁵ The factor $1/N!$ is often justified by invoking the quantum-mechanical indistinguishability of identical particles. However, even in systems of non-identical particles that are so similar that they cannot be separated, the same factor is necessary to ensure extensivity of the Helmholtz free energy [50–52].

Similarly, we can derive the classical limit for $\text{Tr} \exp(-\beta \mathcal{H}) A$, and finally, we can write the classical expression for the thermal average of the observable A as

$$\langle A \rangle = \frac{\int d\mathbf{p}^N d\mathbf{r}^N \exp\left\{-\beta\left[\sum_i p_i^2/(2m_i) + \mathcal{U}(\mathbf{r}^N)\right]\right\} A(\mathbf{p}^N, \mathbf{r}^N)}{\int d\mathbf{p}^N d\mathbf{r}^N \exp\left\{-\beta\left[\sum_j p_j^2/(2m_j) + \mathcal{U}(\mathbf{r}^N)\right]\right\}}. \quad (2.2.20)$$

Eqs. (2.2.19) and (2.2.20) constitute the starting points for a wide range of simulations of classical many-body systems. Eqs. (2.2.19) and (2.2.20) are expressed in terms of high-dimensional integrals over the dN momenta and dN coordinates of all N particles, where d denotes the dimensionality of the system. The $2dN$ -dimensional space spanned by all momenta and coordinates is called *phase space*.

2.3 Ensembles

In statistical mechanics as in thermodynamics, the state of the system is determined by a number of control parameters, some of them extensive (e.g., N , the number of particles), some of them intensive (e.g., the pressure P or the temperature T). For historical reasons we denote the collection of all realizations of a system, which are compatible with a set of control parameters by the name “ensemble”. There are different ensembles for different sets of control parameters. The historical names of these ensembles (“micro-canonical,” “canonical,” “grand-canonical,” etc.) are not particularly illuminating. Below, we will list these names when we describe the most frequently used ensembles. However, in what follows, we will often denote ensembles by the control variables that are kept constant, e.g., the “constant- NVE ensemble” or the “constant- μVT ensemble”. In the following sections, we assume for convenience that the system consists of particles with no internal degrees of freedom (i.e., no rotations, vibrations, or electronic excitations). That assumption simplifies the notation, but for molecular systems, we will of course have to take internal degrees of freedom into account.

2.3.1 Micro-canonical (constant-NVE) ensemble

In the micro-canonical ensemble the energy, volume, and the number of particles of every species are kept constant.⁶ In a classical system, the total energy is given by the Hamiltonian, \mathcal{H} , which is the sum of the kinetic and potential

⁶ In simulations we can fix different thermodynamic parameters $X, Y, Z, \dots; N, V, E$ is just one example. For the sake of compactness, we will often refer to an ensemble with X, Y, Z constant as the XYZ -ensemble. We refer to the corresponding simulations as XYZ -MC or XYZ -MD.

energy:

$$\mathcal{H} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m} + U(\mathbf{r}^N), \quad (2.3.1)$$

in which we have assumed that the potential energy does not depend on the momenta \mathbf{p} . The classical partition function in the micro-canonical ensemble is the phase-space integral over a hyper-surface where the value of the Hamiltonian is equal to the imposed value of the energy E .

The constraint that the system has to be on the hyper-surface $\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) = E$ can be imposed via a δ -function, and hence for a three-dimensional system ($d = 3$):

$$\Omega_{E,V,N} \equiv \frac{1}{h^{3N} N!} \int d\mathbf{p}^N d\mathbf{r}^N \delta(\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) - E). \quad (2.3.2)$$

2.3.2 Canonical (constant-NVT) ensemble

The ensemble of states at constant N , V and T is called the “canonical ensemble.” As described in the previous section, the classical partition function, \mathcal{Q} , for a system of atoms at constant N , V and T is given by:

$$\mathcal{Q}_{N,V,T} = \frac{1}{h^{3N} N!} \int d\mathbf{p}^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]. \quad (2.3.3)$$

As the potential energy does not depend on the momenta of the system, the integration over the momenta can be done analytically⁷

$$\int d\mathbf{p}^N \exp\left[-\beta \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m}\right] = \left[\int dp \exp\left[-\beta \frac{p^2}{2m}\right] \right]^{3N} = \left(\frac{2\pi m}{\beta} \right)^{3N/2}. \quad (2.3.4)$$

If we define the thermal Broglie wavelength Λ as

$$\Lambda = \sqrt{\frac{h^2}{2\pi mk_B T}}, \quad (2.3.5)$$

we can write the canonical partition function as:

$$\mathcal{Q}(N, V, T) = \frac{1}{\Lambda^{3N} N!} \int d\mathbf{r}^N \exp\left[-\beta U(\mathbf{r}^N)\right] \equiv \frac{1}{\Lambda^{3N} N!} Z(N, V, T), \quad (2.3.6)$$

⁷ For molecular systems, in particular for systems of flexible molecules where bond lengths are fixed by holonomic constraints (see section 14.1), the integration over momenta may result in a Jacobian that depends on the coordinates of the nuclei in the molecule.

which defines the configurational integral, $Z \equiv Z(N, V, T)$:

$$Z(N, V, T) = \int d\mathbf{r}^N \exp\left[-\beta U(\mathbf{r}^N)\right]. \quad (2.3.7)$$

Unlike the integral over momenta, the configurational integral can almost never be computed analytically. The canonical partition function $\Omega(N, V, T)$ is related to the Helmholtz free energy, F , through

$$\beta F = -\ln \Omega(N, V, T). \quad (2.3.8)$$

Having defined the configurational integral $Z(N, V, T)$, we can write the ensemble average of a quantity $A(\mathbf{r}^N)$ that depends on the coordinates only as

$$\langle A \rangle = \frac{1}{Z(N, V, T)} \int d\mathbf{r}^N A(\mathbf{r}^N) \exp\left[-\beta U(\mathbf{r}^N)\right]. \quad (2.3.9)$$

The probability (\mathcal{N}) of finding our system in a particular configuration \mathbf{r}^N , is given by

$$\begin{aligned} \mathcal{N}(\mathbf{r}^N) &= \frac{1}{Z(N, V, T)} \int d\mathbf{r}'^N \delta(\mathbf{r}^N - \mathbf{r}'^N) \exp\left[-\beta U(\mathbf{r}'^N)\right] \\ &\propto \exp\left[-\beta U(\mathbf{r}^N)\right]. \end{aligned} \quad (2.3.10)$$

2.3.3 Isobaric-isothermal (constant-NPT) ensemble

The canonical ensemble describes a system at constant temperature and volume. In experiments, it is more common to fix the pressure P than the volume V . As with the constant- NVT ensemble, we can derive the probability distribution function for the constant- NPT ensemble by considering a closed system that consists of the system of interest (system 1), that is in contact with a reservoir (system 2), which acts as both a thermostat and barostat (see Fig. 2.2). The two sub-systems can exchange energy and change their volume in such a way that the total volume is constant.

For simplicity, we start with the quantum expression for the total entropy of the system, Eq. (2.2.6):

$$S = S_1 + S_2 = k_B \ln \Omega_1(E_1, V_1, N_1) + k_B \ln \Omega_2(E_2, V_2, N_2). \quad (2.3.11)$$

As system 2 is much larger than system 1, we can make an expansion of Ω around V and E :

$$\begin{aligned} \ln \Omega(E_2, V_2, N_2) &= \ln \Omega(E, V, N_2) + \left(\frac{\partial \ln \Omega(E, V, N_2)}{\partial E} \right)_{N,V} (E - E_1) \\ &\quad + \left(\frac{\partial \ln \Omega(E, V, N_2)}{\partial V} \right)_{N,E} (V - V_1) + \dots \end{aligned}$$

$$= \ln \Omega(E, V, N_2) + \frac{E - E_1}{k_B T} + \frac{P(V - V_1)}{k_B T} + \dots , \quad (2.3.12)$$

where we have used Eq. (2.1.15), which relates the derivative of the entropy with respect to energy and volume to $1/T$ and P/T , respectively. We can then write the probability to find system 1 with an energy E_1 and volume V_1 as:

$$\begin{aligned} P(E_1, V_1, N_1) &= \frac{\Omega(E - E_1, V - V_1, N_2)}{\sum_j \int dV' \Omega(E - E_j, V - V', N_2)} \\ &\propto \exp \left[-\frac{E_1}{k_B T} - \frac{PV_1}{k_B T} \right]. \end{aligned} \quad (2.3.13)$$

Taking the classical limit, we obtain an expression for the N, P, T -partition function, $\mathcal{Q} \equiv Q(N, P, T)$, which is an integral over particle coordinates *and* over the volume V :

$$\mathcal{Q}(N, P, T) \equiv \frac{\beta P}{\Lambda^{3N} N!} \int dV \exp(-\beta PV) \int d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)], \quad (2.3.14)$$

where the factor βP has been included to make $\mathcal{Q}(N, P, T)$ dimensionless. From Eq. (2.3.14) we obtain the probability to find our system in a particular configuration \mathbf{r}^N and volume V :

$$\mathcal{N}(\mathbf{r}^N) \propto \exp[-\beta PV - \beta U(\mathbf{r}^N)]. \quad (2.3.15)$$

$\mathcal{Q}(N, P, T)$ is related to the Gibbs free energy, G , via

$$\beta G = -\ln \mathcal{Q}(N, P, T). \quad (2.3.16)$$

The above relation follows from the fact that, in the thermodynamic limit, the integral in Eq. (2.3.14) is completely dominated by the maximum value of the integrand $\sim \exp\{-\beta[PV^* + F(N, V^*, T)]\}$, where V^* is the volume for which the integrand is maximal. This *maximum term* method is used to establish the relation between thermodynamic variables and statistical mechanical partition functions for other ensembles.⁸

2.3.4 Grand-canonical (constant- μ VT) ensemble

Thus far, we have considered ensembles in which the total number of particles remains constant. It is often convenient to consider open systems, in which the

⁸ The maximum-term or *saddle-point* approximation relies on the observation that we can approximate the (one-dimensional) integral \mathcal{I} of a function $e^{R(x)}$, which is sharply peaked at x^* , by replacing $R(x)$ close to x^* by $R(x^*) - (c/2)(x - x^*)^2$, where c equals (minus) the second derivative of $R(x)$ at x^* . The resulting Gaussian integral $\int dx e^{R(x^*) - (c/2)(x - x^*)^2}$ yields $\mathcal{I} \approx e^{R(x^*)} \sqrt{2\pi/c}$. In statistical mechanics, $\ln \mathcal{I}$ is related to the appropriate thermodynamic potential, and the contribution of $\ln \sqrt{2\pi/c}$ is negligible in the thermodynamic limit.

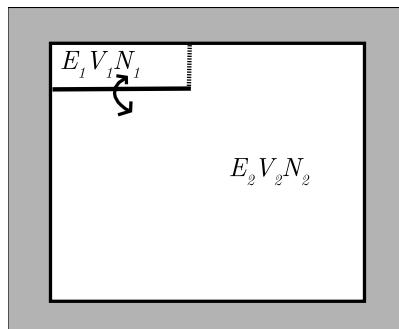


FIGURE 2.3 An isolated system consisting of two boxes 1 and 2 that can exchange heat and exchange particles in such a way that the total energy and number of particles remain constant. System 2 is much larger compared to system 1, so it can act as a heat bath and buffer on system 1.

number of particles is allowed to change. We consider again a system containing two subsystems (Fig. 2.3). The volume of system 1 is fixed, but it is allowed to exchange energy and particles with the reservoir 2. As before, the entire system is closed, and the entropy of the entire system is given by Eq. (2.3.11). As system 2 is much larger than system 1, we can expand $\ln \Omega$ around E and N :

$$\begin{aligned} \ln \Omega_2(E_2, V_2, N_2) &= \ln \Omega(E, V_2, N) + \left(\frac{\partial \ln \Omega(E, V, N_2)}{\partial E} \right)_{N, V} (E - E_1) \\ &\quad + \left(\frac{\partial \ln \Omega(E, V_2, N)}{\partial N} \right)_{E, V} (N - N_1) + \dots \\ &= \ln \Omega(E, V_2, N) + \frac{E - E_1}{k_B T} - \frac{\mu(N - N_1)}{k_B T} + \dots \end{aligned} \quad (2.3.17)$$

where we have used Eq. (2.1.15) to relate the derivative of the entropy with respect to the number of particles to the chemical potential. It then follows that the probability to find system 1 with an energy E_1 and number of particles N_1 is:

$$\begin{aligned} P(E - E_1, V_2, N - N_1) &= \frac{\Omega(E - E_1, V_2, N - N_1)}{\sum_j \sum_{M=0}^N \Omega(E - E_j, V_2, N - M)} \\ &\propto \exp \left[-\frac{E_1}{k_B T} + \frac{\mu N_1}{k_B T} \right]. \end{aligned} \quad (2.3.18)$$

The classical partition function now involves a summation of over all particles in system 1. As the reservoir is much larger than system 1, we can replace the upper limit of this summation by ∞ :

$$\begin{aligned}\Xi(\mu, V, T) &\equiv \sum_{N=0}^{\infty} \frac{\exp(\beta\mu N)}{\Lambda^{3N} N!} \int d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)] \\ &= \sum_{N=0}^{\infty} \exp(\beta\mu N) e^{-\beta F(N, V, T)},\end{aligned}\quad (2.3.19)$$

where we have defined the grand-canonical partition function, $\Xi \equiv \Xi(\mu, V, T)$. From Eq. (2.3.19) we obtain the probability to find N particles in system 1, in configuration \mathbf{r}^N :

$$\mathcal{N}(\mathbf{r}^N) \propto \exp[\beta\mu N - \beta U(\mathbf{r}^N)]. \quad (2.3.20)$$

From Eqs. (2.3.19) and (2.3.8) it follows, using the maximum term method, that

$$-k_B T \ln \Xi = F - N\mu = \Phi, \quad (2.3.21)$$

where Φ is the grand potential defined in section 2.1.2. For homogeneous systems, we can replace Φ by $-PV$.

2.4 Ergodicity

Thus far, we have discussed the average behavior of many-body systems in a purely static sense: we introduced only the assumption that every quantum state of a many-body system with energy E is equally likely to be occupied. Such an average over all possible quantum states of a system is called an *ensemble* average. However, this is not how we usually think about the average behavior of a system. In most experiments, we perform a series of measurements during a certain time interval and then determine the average of these measurements. In fact, the idea behind **Molecular Dynamics** simulations is precisely that we can study the average behavior of a many-particle system simply by computing the natural time evolution of that system numerically and then average the quantity of interest over a sufficiently long time. To take a specific example, let us consider a fluid consisting of atoms. Suppose that we wish to compute the average density of the fluid at a distance r from a given atom i , $\rho_i(r)$. Clearly, the instantaneous density depends on the coordinates \mathbf{r}_j of all particles j in the system. As time progresses, the atomic coordinates will change (according to Newton's equations of motion), and hence the density around atom i will change. Provided that we have specified the initial coordinates and momenta of all atoms $(\mathbf{r}^N(0), \mathbf{p}^N(0))$, we know, at least in principle, the time evolution of $\rho_i(r; \mathbf{r}^N(0), \mathbf{p}^N(0), t)$.

In a standard **Molecular Dynamics** simulation, we measure the time-averaged density $\overline{\rho_i(r)}$ of a system of N atoms, in a volume V , at a constant total energy E :

$$\overline{\rho_i(r)} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t dt' \rho_i(r; t'). \quad (2.4.1)$$

Note that, in writing down this equation, we have implicitly assumed that, for t sufficiently long, the time average does not depend on the initial conditions. This is, in fact, a subtle assumption that is not true in general (see e.g., [53]). However, we shall disregard subtleties and simply assume that, once we have specified N , V , and E , time averages do not depend on the initial coordinates and momenta. If that is so, then we would not change our result for $\overline{\rho_i(r)}$ if we average over many different initial conditions; that is, we consider the hypothetical situation where we run a large number of **Molecular Dynamics** simulations at the same values for N , V , and E , but with different initial coordinates and momenta,

$$\overline{\rho_i(r)} = \frac{\sum_{\text{initial conditions}} \left(\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t dt' \rho_i(r; \mathbf{r}^N(0), \mathbf{p}^N(0), t') \right)}{\text{number of initial conditions}}. \quad (2.4.2)$$

We now consider the limiting case where we average over all initial conditions compatible with the imposed values of N , V , and E . In that case, we can replace the sum over initial conditions with an integral:

$$\frac{\sum_{\text{initial conditions}} f(\mathbf{r}^N(0), \mathbf{p}^N(0))}{\text{number of initial conditions}} \rightarrow \frac{\int_E d\mathbf{r}^N d\mathbf{p}^N f(\mathbf{r}^N(0), \mathbf{p}^N(0))}{\Omega(N, V, E)}, \quad (2.4.3)$$

where f denotes an arbitrary function of the initial conditions $\mathbf{r}^N(0)$, $\mathbf{p}^N(0)$, while $\Omega(N, V, E) = \int_E d\mathbf{r}^N d\mathbf{p}^N$, where we have ignored a constant factor. Note that the second line in Eq. (2.4.3) is nothing else than the micro-canonical (constant- NVE) average of f . In what follows, we denote an ensemble average by $\langle \dots \rangle$ to distinguish it from a time average, denoted by a bar. If we switch the order of the time averaging and the averaging over initial conditions, we find

$$\overline{\rho_i(r)} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t dt' \left\langle \rho_i(r; \mathbf{r}^N(0), \mathbf{p}^N(0), t') \right\rangle_{NVE}. \quad (2.4.4)$$

However, the ensemble average in this equation does not depend on the time t' . This is so because there is a one-to-one correspondence between the initial phase-space coordinates of a system and those that specify the state of the system at a later time t' (see e.g., [53,54]). Hence, averaging over all initial phase-space coordinates is equivalent to averaging over the time-evolved phase-space coordinates. For this reason, we can leave out the time averaging in Eq. (2.4.4), and we find

$$\overline{\rho_i(r)} = \langle \rho_i(r) \rangle_{NVE}. \quad (2.4.5)$$

This equation states that, if we wish to compute the average of a function of the coordinates and momenta of a many-particle system, we can *either* compute that quantity by time averaging (the “MD” approach) *or* by ensemble averaging (the

“MC” approach). It should be stressed that the preceding paragraphs are meant only to make Eq. (2.4.5) *plausible*, not as a proof. In fact, that would have been quite impossible because Eq. (2.4.5) is not true in general. However, in what follows, we shall simply assume that the “ergodic hypothesis”, as Eq. (2.4.5) is usually referred to, applies to the systems that we study in computer simulations. The reader should be aware that there are many examples of systems that are not ergodic *in practice*, such as glasses and metastable phases, or even *in principle*, such as nearly harmonic solids.

2.5 Linear response theory

Until now, we focused on ensemble averages (Eq. (2.2.20)) or time averages (Eq. (2.4.1)) of quantities that are constant in time (once fluctuations have been averaged out). One of the great advantages of the **Molecular Dynamics** method is that it also allows us to predict the time-dependent response of a system to an external perturbation. Examples of such responses are the heat current due to a temperature gradient, or the electrical current induced by an electric field (we shall see more examples in Chapter 5).

It would seem that computing such currents would require a non-equilibrium simulation, where we impose the external perturbation of interest. This is indeed possible, but the disadvantage is that for every different perturbation, we would have to carry out a separate simulation. Fortunately, for systems that are only weakly perturbed (such that the response is linear in the applied perturbation), we can predict the response to an applied perturbation by studying the decay of fluctuations in the corresponding current *in equilibrium*.

Onsager [55,56] was the first to suggest that a response (e.g., a current) induced in a system by a weak external perturbation, decays in the same way as a spontaneous fluctuation of that current in equilibrium. Onsager formulated his “regression” hypothesis in the language of non-equilibrium thermodynamics; in fact, Onsager’s 1931 papers [55,56], although partly inspired by earlier work, created the field (see [57]). The theory of non-equilibrium thermodynamics provides a phenomenological description of the relation between applied perturbations and resulting fluxes/currents. In particular, it defines the linear transport coefficients that relate a small perturbation to the resulting fluxes. However, non-equilibrium thermodynamics is not an atomistic theory and hence it is not always straightforward to make a link between the macroscopic forces and fluxes and the atomistic description of the same quantities. Moreover, for simulations, we need expressions for the transport coefficients that are computable in terms of molecular coordinates and momenta. Such expressions are provided by linear response theory.

Below we give a simple introduction to *classical* linear response theory, to illustrate the mechanical basis of Onsager’s regression hypothesis. For a more detailed discussion, the reader is referred to textbooks on advanced statistical mechanics, such as [53]. A simple introduction (similar to the one presented

here) is given in the book by Chandler [58], while an extensive discussion of linear response theory in the context of the theory of liquids is given in [59].

2.5.1 Static response

Before discussing transport, we consider the static response of a system to a weak applied field. The field could be an electric field, for instance, and the response might be the electric current or, for a nonconducting material, the electric polarization. Suppose that we are interested in the response to a property that can be expressed as the ensemble average of a dynamical variable A . In the presence of an external perturbation, the average of A changes from its equilibrium value $\langle A \rangle_0$ to $\langle A \rangle_0 + \langle \Delta A \rangle$. Next, we must specify the perturbation. We assume that the perturbation also can be written as an explicit function of the coordinates (and, possibly, momenta) of the particles in the system. The effect of the perturbation is to change the Hamiltonian \mathcal{H}_0 of the system, to $\mathcal{H}_0 - \lambda B(\mathbf{p}^N, \mathbf{r}^N)$. For instance, in the case of an electric field along the x direction, the change in \mathcal{H} would be $\Delta \mathcal{H} = -E_x M_x(\mathbf{r}^N)$, where M_x is the x component of the total dipole moment of the system. The electric field E_x corresponds to the parameter λ . We can immediately write down the general expression for $\langle \Delta A \rangle$:

$$\langle A \rangle_0 + \langle \Delta A \rangle = \frac{\int d\Gamma A \exp[-\beta(\mathcal{H}_0 - \lambda B)]}{\int d\Gamma \exp[-\beta(\mathcal{H}_0 - \lambda B)]}, \quad (2.5.1)$$

where we have used the symbol Γ to denote $\{\mathbf{p}^N, \mathbf{r}^N\}$, the phase-space coordinates of the system. Let us now compute the part of $\langle \Delta A \rangle$ that varies linearly with λ . To this end, we compute

$$\left(\frac{\partial \langle \Delta A \rangle}{\partial \lambda} \right)_{\lambda=0}. \quad (2.5.2)$$

Straightforward differentiation shows that

$$\left(\frac{\partial \langle \Delta A \rangle}{\partial \lambda} \right)_{\lambda=0} = \beta \{ \langle AB \rangle_0 - \langle A \rangle_0 \langle B \rangle_0 \}. \quad (2.5.3)$$

Taking again the example of the electric polarization, we can compute the change in dipole moment of a system due to an applied field E_x :

$$\langle \Delta M_x \rangle = E_x \left(\frac{\partial M_x}{\partial E_x} \right)_{E_x=0} = \beta E_x \left\{ \langle M_x^2 \rangle - \langle M_x \rangle^2 \right\}. \quad (2.5.4)$$

Suppose that we wish to compute the electric susceptibility of an ideal gas of nonpolarizable dipolar molecules with dipole moment μ . In that case,

$$\left\{ \langle M_x^2 \rangle - \langle M_x \rangle^2 \right\} = \sum_{i,j=1}^N \langle \mu_x^i \mu_x^j \rangle$$

$$= N \left\langle (\mu_x^i)^2 \right\rangle \\ = \frac{N\mu^2}{3},$$

and hence,

$$P_x \equiv \frac{M_x}{V} = \frac{\mu^2 \rho}{3k_B T} E_x. \quad (2.5.5)$$

Of course, this example is special because it can be evaluated exactly. But, in general, we can compute the expression (2.5.3) for the susceptibility, only numerically. It should also be noted that, actually, the computation of the dielectric susceptibility is quite a bit more subtle than suggested in the preceding example. The subtleties are discussed in the book by Allen and Tildesley [21] and the contribution of McDonald in [44].

Hamiltonian thermodynamic integration

The above discussion of static linear response is but a special case of the effect of a change in the Hamiltonian of a system on its free energy (see section 8.4.2) —an approach pioneered by Kirkwood [60], partly together with Monroe-Boggs [61].⁹

The essence of Hamiltonian integration can be expressed in a few lines. Consider a Hamiltonian $\mathcal{H}(\lambda)$, such that $\mathcal{H}(\lambda = 0) = \mathcal{H}_0$ and $\mathcal{H}(\lambda = 1) = \mathcal{H}_1$. Typically, \mathcal{H}_0 corresponds to a reference state for which we know the free energy. We need not assume that $\mathcal{H}(\lambda)$ is a linear function of λ . In what follows, we use the notation

$$\mathcal{H}'(\lambda) \equiv \left(\frac{\partial \mathcal{H}(\lambda)}{\partial \lambda} \right).$$

For the case discussed above $\mathcal{H}(\lambda) = \mathcal{H}_0 - \lambda B$, and hence $\mathcal{H}'(\lambda) = -B$.

In general, we have

$$\left(\frac{\partial F(\lambda)}{\partial \lambda} \right) = \langle \mathcal{H}'(\lambda) \rangle,$$

and hence

$$F(\lambda = 1) = \int_0^1 d\lambda \langle \mathcal{H}'(\lambda) \rangle_\lambda. \quad (2.5.6)$$

In Eq. (2.5.6), the subscript in $\langle \dots \rangle_\lambda$ means that the Boltzmann average is computed using a Boltzmann weight corresponding to the Hamiltonian $\mathcal{H}(\lambda)$. Eq. (2.5.6) is the starting point for most “Hamiltonian” integration schemes in simulations that will be discussed in Section 8.4.2. Such schemes aim to compute the unknown free energy of a system at $\lambda = 1$ from knowledge of the free

⁹ The same Elisabeth Monroe who, together with Kirkwood, laid the foundations for the theory of entropic freezing.

energy of a reference state ($\lambda = 0$) using the average values of $\mathcal{H}'(\lambda)$ evaluated during the simulations.

2.5.2 Dynamic response

Next, we consider a simple time-dependent perturbation. We begin by preparing the system in the presence of a very weak, constant perturbation (λB). The static response of A to this perturbation is given by Eq. (2.5.3). At time $t = 0$, we switch off the external perturbation. The response ΔA will now decay to 0. We can write an expression for the average of ΔA at time t :

$$\langle \Delta A(t) \rangle = \frac{\int d\Gamma \exp[-\beta(\mathcal{H}_0 - \lambda B)] A(t)}{\int d\Gamma \exp[-\beta(\mathcal{H}_0 - \lambda B)]}, \quad (2.5.7)$$

where $A(t)$ is the value of A at time t if the system started at point Γ in phase space and then evolved according to the natural time evolution of the *unperturbed* system. For convenience, we have assumed that the average of A in the unperturbed system vanishes. In the limit $\lambda \rightarrow 0$, we can write

$$\begin{aligned} \langle \Delta A(t) \rangle &= \beta \lambda \frac{\int d\Gamma \exp[-\beta \mathcal{H}_0] B A(t)}{\int d\Gamma \exp[-\beta \mathcal{H}_0]} \\ &= \beta \lambda \langle B(0) A(t) \rangle. \end{aligned} \quad (2.5.8)$$

The quantity $\langle B(0) A(t) \rangle$ is called a *time correlation function* (if $B = A$, it is called an *autocorrelation* function). The time correlation function $\langle B(0) A(t) \rangle$ is the time average of the product of the value of A at time τ and B at time $t + \tau$:

$$\langle B(0) A(t) \rangle \equiv \lim_{t_0 \rightarrow \infty} \frac{1}{t_0} \int_0^{t_0} d\tau A(\mathbf{r}^N(\tau), \mathbf{p}^N(\tau)) B(\mathbf{r}^N(t + \tau), \mathbf{p}^N(t + \tau)), \quad (2.5.9)$$

where $\{\mathbf{r}^N(x), \mathbf{p}^N(x)\}$ denotes the phase-space coordinates at time x . Note that the time evolution of $\{\mathbf{r}^N(x), \mathbf{p}^N(x)\}$ is determined by the *unperturbed* Hamiltonian \mathcal{H}_0 .

To give a specific example, consider a gas of dipolar molecules again in the presence of a weak electric field E_x . The perturbation is equal to $-E_x M_x$. At time $t = 0$, we switch off the electric field. When the field was still on, the system had a net dipole moment. When the field is switched off, this dipole moment decays:

$$\langle \Delta M_x(t) \rangle = E_x \beta \langle M_x(0) M_x(t) \rangle. \quad (2.5.10)$$

In words, the decay of the macroscopic dipole moment of the system is determined by the dipole autocorrelation function, which describes the decay of spontaneous fluctuations of the dipole moment in equilibrium. This relation between the decay of the response to an external perturbation and the decay of fluctuations in equilibrium is an example of Onsager's regression hypothesis.

It might seem that the preceding example of a constant perturbation that is suddenly switched off is of little practical use, because we are interested in the effect of an arbitrary time-dependent perturbation. Fortunately, in the linear regime that we are considering, the relation given by Eq. (2.5.8) is enough to derive the general response.

To see this, let us consider a time-dependent external field $f(t)$ that couples to a mechanical property B ; that is,

$$\mathcal{H}(t) = \mathcal{H}_0 - f(t)B. \quad (2.5.11)$$

To linear order in $f(t)$, the most general form of the response of a mechanical property A to this perturbation is

$$\langle \Delta A(t) \rangle = \int_{-\infty}^{\infty} dt' \chi_{AB}(t, t') f(t'), \quad (2.5.12)$$

where χ_{AB} , the “after-effect function”, describes the linear response. We know several things about the response of the system that allow us to simplify Eq. (2.5.12). First of all, the response must be *causal*; that is, there can be no response *before* the perturbation is applied. As a consequence,

$$\chi_{AB}(t, t') = 0 \text{ for } t < t'. \quad (2.5.13)$$

Secondly, the response at time t to a perturbation at time t' depends only on the time difference $t - t'$. Hence,

$$\langle \Delta A(t) \rangle = \int_{-\infty}^t dt' \chi_{AB}(t - t') f(t'). \quad (2.5.14)$$

Once we know χ , we can compute the linear response of the system to an arbitrary time-dependent perturbing field $f(t)$. To find an expression for χ_{AB} , let us consider the situation described in Eq. (2.5.8), namely, an external perturbation that has a constant value λ until $t = 0$ and 0 from then on. From Eq. (2.5.14), it follows that the response to such a perturbation is

$$\begin{aligned} \langle \Delta A(t) \rangle &= \lambda \int_{-\infty}^0 dt' \chi_{AB}(t - t') \\ &= \lambda \int_t^{\infty} d\tau \chi_{AB}(\tau). \end{aligned} \quad (2.5.15)$$

If we compare this expression with the result of Eq. (2.5.8), we see immediately that

$$\lambda \int_t^{\infty} d\tau \chi_{AB}(\tau) = \beta \lambda \langle B(0)A(t) \rangle \quad (2.5.16)$$

or

$$\chi_{AB}(t) = \begin{cases} -\beta \langle B(0)\dot{A}(t) \rangle & \text{for } t > 0 \\ 0 & \text{for } t \leq 0. \end{cases} \quad (2.5.17)$$

To give a specific example, consider the mobility of a molecule in an external field F_x . The Hamiltonian in the presence of this field is

$$\mathcal{H} = \mathcal{H}_0 - F_x x. \quad (2.5.18)$$

The phenomenological expression for the steady-state velocity of a molecule in an external field is

$$\langle \Delta v_x(t) \rangle = mF_x, \quad (2.5.19)$$

where m is the mobility of the molecule under consideration. We can now derive a microscopic expression for the mobility in terms of a time-correlation function. From Eqs. (2.5.14) through (2.5.17), we have

$$\begin{aligned} \langle \Delta v_x(t) \rangle &= F_x \int_{-\infty}^t dt' \chi_{v_x x}(t-t') \\ &= F_x \int_0^\infty d\tau \chi_{v_x x}(\tau) \\ &= -\beta F_x \int_0^\infty d\tau \langle x(0)\dot{v}_x(\tau) \rangle \\ &= +\beta F_x \int_0^\infty d\tau \langle v_x(0)v_x(\tau) \rangle. \end{aligned} \quad (2.5.20)$$

In the last line of Eq. (2.5.20), we used the stationarity property of time-correlation functions:

$$\frac{d}{dt} \langle A(t)B(t+t') \rangle = 0. \quad (2.5.21)$$

Carrying out the differentiation, we find that

$$\langle \dot{A}(t)B(t+t') \rangle = -\langle A(t)\dot{B}(t+t') \rangle. \quad (2.5.22)$$

Combining Eqs. (2.5.19) and (2.5.20), we find that

$$m = \beta \int_0^\infty dt \langle v_x(0)v_x(t) \rangle. \quad (2.5.23)$$

Eq. (2.5.23) relating a transport coefficient to an integral of a time correlation function, is an example of a so-called **Green-Kubo** relation [62]. In Chapter 5 we shall see that the mobility m is related to the self-diffusion coefficient D through the Einstein relation $m = \beta D$. In Section F.1 of the Appendix, we discuss how,

from the knowledge of certain time-correlation functions, we can compute the rate of dissipation due to an applied periodic perturbation. Such information is very useful when modelling, for instance, the absorption of radiation, be it that the expressions derived in the Appendix only apply to the classical case where $\hbar\omega \ll k_B T$.

Power spectra

Time-correlation functions are often computed (and also measured in spectroscopic experiments), by Fourier transforming from the frequency domain using the **Wiener-Khinchin (WK)** theorem (see e.g., [53]).

To derive the **WK** theorem, we first define the Fourier transform of the observable of interest over time interval T

$$\hat{a}(\omega) \equiv \int_0^T dt A(t) e^{i\omega t}. \quad (2.5.24)$$

Note that we define the Fourier transform over a finite time interval $\{0 - T\}$, because the simulation has a finite length. We now define $G_A(\omega)$, the *power spectrum* of A , as

$$\begin{aligned} G_A(\omega) &\equiv \lim_{T \rightarrow \infty} \frac{1}{2\pi T} \left\langle |\hat{a}(\omega)|^2 \right\rangle \\ &= \lim_{T \rightarrow \infty} \frac{1}{2\pi T} \int_0^T dt \int_0^T dt' \langle A(t) A(t') \rangle e^{i\omega t} e^{-i\omega t'} \\ &= \lim_{T \rightarrow \infty} \frac{1}{2\pi T} \int_0^T dt' \int_{-t'}^{T-t'} dt - t' \langle A(0) A(t-t') \rangle e^{i\omega(t-t')}, \end{aligned} \quad (2.5.25)$$

where we have used the fact that an equilibrium time correlation function only depends on the time difference $t - t'$. When T is much longer than the time it takes the correlation function to decay, we can now write (in the limit that $T \rightarrow \infty$):

$$G_A(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} d\tau \langle A(0) A(\tau) \rangle e^{i\omega\tau}, \quad (2.5.26)$$

where we have defined $\tau \equiv t - t'$. Equation (2.5.26) shows that $G_A(\omega)$ is the Fourier transform of $\langle A(0) A(\tau) \rangle$, and conversely that

$$\langle A(0) A(\tau) \rangle = \int_{-\infty}^{\infty} d\omega G_A(\omega) e^{-i\omega\tau}. \quad (2.5.27)$$

Equation (2.5.27) is often used to obtain correlation functions from spectroscopic data (see e.g., section F.1). For the calculation of correlation functions, it is important to note that correlating n points, requires $n(n - 1)/2$ multiplications, whereas a (fast) Fourier transform [38] only requires $n \ln n$ operations.

Power spectra can often be determined by measuring the dissipation in a system subject to a periodic perturbation (see Appendix F.1).

2.6 Questions and exercises

Question 1 (Number of configurations).

1. Consider a system A consisting of subsystems A_1 and A_2 , for which $\Omega_1 = 10^{20}$ and $\Omega_2 = 10^{22}$. What is the number of configurations available to the combined system? Also, compute the entropies S , S_1 , and S_2 .
2. By what factor does the number of available configurations increase when 10 m^3 of air at 1.0 atm and 300 K is allowed to expand by 0.001% at constant temperature? Here and in what follows, we assume that air behaves like an ideal gas.
3. By what factor does the number of available configurations increase when 150 kJ is added to a system containing 2.0 mol of particles at constant volume and $T = 300\text{ K}$?
4. A sample consisting of five molecules has a total energy 5ϵ . Each molecule is able to occupy states of energy ϵ_j , with $j = 0, 1, 2, \dots, \infty$. Draw up a table with columns by the energy of the states and write beneath them all configurations that are consistent with the total energy. Identify the type of configuration that is most probable.

Question 2 (Thermodynamic variables in the canonical ensemble). Starting with an expression for the Helmholtz free energy (F) as a function of N , V , T

$$F = \frac{-\ln [Q(N, V, T)]}{\beta},$$

one can derive all thermodynamic properties. Show this by deriving equations for U , p , and S .

Question 3 (Ideal gas (Part 1)). The canonical partition function of an ideal gas consisting of monoatomic particles is equal to

$$Q(N, V, T) = \frac{1}{h^{3N} N!} \int d\Gamma \exp[-\beta H] = \frac{V^N}{\lambda^{3N} N!},$$

in which $\lambda = h/\sqrt{2\pi m/\beta}$ and $d\Gamma = dq_1 \cdots dq_N dp_1 \cdots dp_N$.

Derive expressions for the following thermodynamic properties:

- $F(N, V, T)$ (hint: $\ln(N!) \approx N \ln(N) - N$)
- $p(N, V, T)$ (which leads to the ideal gas law)
- $\mu(N, V, T)$ (which leads to $\mu = \mu_0 + RT \ln \rho$)
- $U(N, V, T)$ and $S(N, V, T)$
- C_v (heat capacity at constant volume)
- C_p (heat capacity at constant pressure)

Question 4 (Van der Waals equation of state). *The van der Waals equation of state describes the behavior of non-ideal gasses.*

$$P = \frac{RT}{v - b} - \frac{a}{v^2},$$

where R is the gas constant. Show that the constants a and b can be related to the critical point:

$$a = \frac{27}{64} \frac{R^2 T_c^2}{P_c}$$

$$b = \frac{1}{8} \frac{RT_c}{P_c}.$$

In this book, we will use the Lennard-Jones fluid very often. The critical point of the Lennard Jones fluid is $T_c = 1.32$, $\rho_c = 0.32$ (molecules per unit volume), and $P_c = 0.131$ [63]. These constants are expressed in reduced units (see section 3.3.2.5).

Plot the equation of state (pressure as a function of the molar volume for $T = 2.00$ and $T = 1.00$.

Question 5 (Fugacity). *In most Chemical Engineering textbooks, the chemical potential is replaced by the fugacity, f .*

$$\beta\mu - \beta\mu^0 = \ln \frac{f}{f^0},$$

where f^0 is the fugacity of the reference state.

- Compute the fugacity of an ideal gas
- How would you choose the reference state f^0 for a real gas

The fugacity coefficient ϕ is introduced to quantify deviations from the ideal gas behavior:

$$\phi = \frac{f}{P},$$

where P is the pressure.¹⁰

Compute ϕ as a function of the molar volume for $T = 2.0$ and $T = 1.0$ for the van der Waals equation of state (see Question 4).

Question 6 (Reversible work). *The First Law of thermodynamics expresses the conservation of the energy E of a system*

$$dE = q + w,$$

¹⁰ Note that for an ideal gas, $P = \rho k_B T$. We can therefore also write $\Phi = f/(\rho_{\text{id}} k_B T)$. In the rest of the book, we use this relation to define the fugacity through $\Phi = f'/\rho_{\text{id}}$. Note that $f' = f/k_B T$. In what follows, we drop the prime.

where q and w denote the (infinitesimal) energy changes of the system due to heat flow and work, respectively. In the case of reversible changes, we can use the First and Second Laws together, to express how the different state functions (E, S, F, G, \dots) change in a reversible transformation. The best-known expression is

$$dE = TdS - PdV + \mu dN ,$$

or, in terms of the Helmholtz free energy F :

$$dF = -SdT - pdV + \mu dN .$$

This expression for dF applies to the case where the reversible work is due to a change of the volume against an external pressure P . However, we can also consider other forms of work, for instance, due to changing the polarization of a system at a constant applied electric field, or to changing the surface area A at constant surface tension γ . Here, we consider the latter case:

$$dF = -SdT + \gamma dA - PdV + \mu dN .$$

1. We will assume that both V and A are linear in N . Making use of the fact that the free energy of a system is extensive, show that

$$F = \mu N - PV + \gamma A$$

We can interpret this expression as a definition of the surface free energy $F_s \equiv \gamma A$.

2. Derive an expression for $(\partial F_s / \partial A)_{N,V,T}$. This expression may look strange, because we change A at constant N .
3. Under what conditions is

$$(\partial F_s / \partial A)_{N,V,T} = \gamma ?$$

Would this condition be satisfied for:

- a liquid-vapor interface?
- a solid-liquid interface?

Question 7 (Ising model). Consider a system of N spins arranged on a simple lattice (1d: a linear chain, 2d: a square lattice, 3d: a simple cubic lattice, etc.). In the presence of a magnetic field, H , the energy of the system is

$$U = - \sum_{i=1}^N H s_i - J \sum_{i>j} s_i s_j ,$$

where J is called the coupling constant ($J > 0$) and $s_i = \pm 1$. The second summation is a summation over all pairs ($d \times N$ for a periodic system, d is the dimensionality of the system). This system is called the Ising model.

1. Show that for positive J , and $H = 0$, the lowest energy of the Ising model is equal to

$$U_0 = -dN J .$$

2. Show that the free energy per spin of a 1d Ising model with zero field is equal to

$$\frac{F(\beta, N)}{N} = -\frac{\ln(2 \cosh(\beta J))}{\beta}$$

when $N \rightarrow \infty$. The function $\cosh(x)$ is defined as

$$\cosh(x) = \frac{\exp[-x] + \exp[x]}{2}. \quad (2.6.1)$$

3. Derive equations for the energy and heat capacity of this system.

Question 8 (The photon gas). An electromagnetic field in thermal equilibrium can be described as a photon gas. From the quantum theory of the electromagnetic field, it is found that the total energy of the system (U) can be written as the sum of photon energies:

$$U = \sum_{j=1}^N n_j \omega_j \hbar = \sum_{j=1}^N n_j \epsilon_j$$

in which ϵ_j is the characteristic energy of a photon with frequency ω_j , $n_j = 0, 1, 2, \dots, \infty$ is the so-called occupancy number of mode j , and N is the number of field modes (here we take N to be finite).

1. Show that the canonical partition function of the system can be written as

$$Q = \prod_{j=1}^N \frac{1}{1 - \exp[-\beta \epsilon_j]}. \quad (2.6.2)$$

Hint: you will have to use the following identity for $|x| < 1$:

$$\sum_{i=0}^{i=\infty} x^i = \frac{1}{1-x}. \quad (2.6.3)$$

For the product of partition functions of two independent systems A and B , we can write

$$Q_A \times Q_B = Q_{AB} \quad (2.6.4)$$

when $A \cap B = \emptyset$ and $A \cup B = AB$.

2. Show that the average occupancy number of state j , $\langle n_j \rangle$, is equal to

$$\langle n_j \rangle = \frac{\partial \ln Q}{\partial (-\beta \epsilon_j)} = \frac{1}{\exp[\beta \epsilon_j] - 1}. \quad (2.6.5)$$

3. Describe the behavior of $\langle n_j \rangle$ when $T \rightarrow \infty$ and when $T \rightarrow 0$.

Question 9 (Ideal gas (Part 2)). An ideal gas is placed in a constant gravitational field. The potential energy of N gas molecules at height z is Mgz , where $M = mN$ is the total mass of N molecules, and g the gravitational acceleration. The temperature in the system is uniform, and the system is infinitely large. We assume that the system is locally in equilibrium, so we are allowed to use a local partition function.

1. Show that the grand-canonical partition function of a system in volume V at height z is equal to

$$Q(\mu, V, T, z) = \sum_{N=0}^{\infty} \frac{\exp[\beta\mu N]}{h^{3N} N!} \int d\Gamma \exp[-\beta(H_0 + Mgz)] \quad (2.6.6)$$

in which H_0 is the Hamiltonian of the system at $z = 0$.

2. Explain that a change in z is equivalent to a change in chemical potential, μ . Use this to show that the pressure of the gas at height z is equal to

$$p(z) = p(z=0) \times \exp[-\beta mgz]. \quad (2.6.7)$$

(Hint: you will need the formula for the chemical potential of an ideal gas.)

Exercise 1 (Distribution of particles). Consider an ideal gas of N particles in a volume V at constant energy E . Let us divide the volume into p identical compartments. Every compartment contains n_i molecules such that

$$N = \sum_{i=1}^{i=p} n_i. \quad (2.6.8)$$

An interesting quantity is the distribution of molecules over the p compartments. Because the energy is constant, every possible eigenstate of the system will be equally likely. This means that in principle it is possible that one of the compartments is empty.

1. On the book's website you can find a program that calculates the distribution of molecules among the p compartments. Run the program for different numbers of compartments (p) and total number of gas molecules (N). The output of the program is the probability of finding x particles in a particular compartment as a function of x .
2. What is the probability that one of the compartments is empty?
3. Consider the case $p = 2$ and N even. The probability of finding $N/2 + n_1$ molecules in compartment 1 and $N/2 - n_1$ molecules in compartment 2 is given by

$$P(n_1) = \frac{N!}{(N/2 - n_1)!(N/2 + n_1)!2^N}. \quad (2.6.9)$$

Compare your numerical results with the analytical expression for different values of N . Show that this distribution is a Gaussian for small n_1/N . Hint: For

$x > 10$, it might be useful to use Stirling's approximation:

$$x! \approx (2\pi)^{\frac{1}{2}} x^{x+\frac{1}{2}} \exp[-x]. \quad (2.6.10)$$

Exercise 2 (Boltzmann distribution). Consider a system of N energy levels with energies $0, \epsilon, 2\epsilon, \dots, (N-1)\epsilon$ and $\epsilon > 0$.

1. Calculate, using the given program, the occupancy of each level for different values of the temperature. What happens at high temperatures?
2. Change the program in such a way that the degeneracy of energy level i equals $i+1$. What do you see?
3. Modify the program in such a way that the occupation of the energy levels, as well as the partition function (q), is calculated for a heteronuclear linear rotor with moment of inertia I . Compare your result with the approximate result

$$q = \frac{2I}{\beta \hbar^2} \quad (2.6.11)$$

for different temperatures. Note that the energy levels of a linear rotor are

$$U = J(J+1) \frac{\hbar^2}{2I} \quad (2.6.12)$$

with $J = 0, 1, 2, \dots, \infty$. The degeneracy of level J equals $2J+1$.

Exercise 3 (Coupled harmonic oscillators). Consider a system of N harmonic oscillators with a total energy U . A single harmonic oscillator has energy levels $0, \epsilon, 2\epsilon, \dots, \infty$ ($\epsilon > 0$). All harmonic oscillators in the system can exchange energy.

1. Invent a computational scheme to update the system at constant total energy (U). Compare your scheme with the scheme that is incorporated in the computer code that you can find on the book's website.
2. Make a plot of the energy distribution of the first oscillator as a function of the number of oscillators for a constant value of U/N . Which distribution is recovered when N becomes large? What is the function of the other $N-1$ harmonic oscillators? Explain.
3. Compare this distribution with the canonical distribution of a single oscillator at the same average energy (use the option NVT).
4. How does this exercise relate to the derivation of the Boltzmann distribution for a system at temperature T ?

Exercise 4 (Random walk on a 1d lattice). Consider the random walk of a single particle on a line. The particle performs jumps of fixed length. Assuming that the probability for forward or backward jumps is equal, the mean-squared displacement of a particle after N jumps is equal to N . The probability that, after N jumps, the net distance covered by the particle equals n is given by

$$\ln[P(n, N)] \approx \frac{1}{2} \ln\left(\frac{2}{\pi N}\right) - \frac{n^2}{2N}.$$

1. Derive this equation using Stirling's approximation for $\ln x!$.

2. Compare your numerical result for the root mean-squared displacement with the theoretical prediction (the computed function $P(n, N)$). What is the diffusivity of this system?
3. Modify the program in such a way that the probability to jump in the forward direction equals 0.8. What happens?

Exercise 5 (Random walk on a 2d lattice). Consider the random walk of N particles on a $M \times M$ lattice. Two particles cannot occupy the same lattice site. On this lattice, periodic boundaries are used. This means that when a particle leaves the lattices it returns on the opposite side of the lattice; i.e., the coordinates are given modulo M .

1. What is the fraction of occupied sites (θ) of the lattice as a function of M and N ?
2. Make a plot of the diffusivity D as a function of θ for $M = 32$. For low values of θ , the diffusivity can be approximated by

$$D \approx D_0(1 - \theta).$$

Why is this equation reasonable at low densities? Why does it break down at higher densities?

3. Modify the program in such a way that the probability of jumping in one direction is larger than the probability of jumping in the other direction. Explain the results.
4. Modify the program in such a way that periodic boundary conditions are used in one direction and reflecting boundary conditions in the other. What happens?

Chapter 3

Monte Carlo simulations

3.1 Preamble: molecular simulations

In the previous chapter, we introduced the basics of (classical) statistical mechanics. We found that many observables may either be expressed as time averages or as ensemble averages. In the remainder of this book, we discuss how these observables may be computed. We stress that once we start using numerical techniques, we give up on exact results. This is so because, first of all, simulations are subject to statistical noise. This problem can be alleviated, but not eliminated, by sampling longer. Secondly, simulations contain systematic errors related to the fact that the systems we simulate are far from macroscopic: finite-size effects may be substantial. Again, this problem may be alleviated by performing a systematic finite-size scaling analysis, but the problem should be recognized. Thirdly, we discretize Newton's equations of motion. The resulting trajectories are not the true trajectories of the system. This problem becomes less severe for shorter time steps, but it cannot be eliminated completely. Finally, if we aim to model real materials, we have to use an approximate description of the potential energy function $U(\mathbf{r}^N)$. The approximation may be good, —but it is never perfect. Hence, in simulations, we always predict the properties of a *model* system. When, in the remainder of this book, we mention simulations of real materials, the reader should always bear these approximations in mind, not to mention the fact that we will be using a classical rather than a quantum description. Quantum simulation techniques are beyond the scope of this book. But the general comments about the trade-off between cost and rigor still apply, and even more so than for classical simulations.

There are two distinct numerical approaches to compute the observable properties of classical many-body systems: one method is called **Molecular Dynamics (MD)**. As the name suggests, a Molecular-Dynamics simulation generates an approximate solution of Newton's equations of motion, yielding a trajectory $\mathbf{r}^N(t)$, $\mathbf{p}^N(t)$ of the system in phase space.¹ Using the MD approach, we can compute time averages of observable properties, both static (e.g., thermodynamic observables) and dynamic (e.g., transport coefficients). Whilst the practical implementation of MD is not always simple, the idea behind it is blindingly obvious.

¹ We denote the x , y , z coordinates of the N particles in our system by \mathbf{r}^N , and \mathbf{p}^N denotes the corresponding momenta.

In contrast, the idea behind the [Monte Carlo \(MC\)](#) method (or, more precisely, the [Markov-Chain Monte Carlo \(MCMC\)](#)) Method is highly nontrivial. In fact, the development of the MCMC method is arguably one of the greatest discoveries in computational science of the 20th century.² The present chapter deals with the basis of the MCMC method, as applied to classical many-body systems.

The [MCMC](#) method is not limited to computing the properties of many-body systems: whenever we have a system that can be found in a (very) large number of states, and we know how to compute the relative probability with which these states are occupied (e.g., the Boltzmann weight), the MCMC method can be used. Unlike MD, MCMC provides no information about the time evolution of a many-body system. Hence, MCMC cannot be used to compute transport properties.

This might suggest that, all things being equal, MD is always to be preferred over MC. However, all things are not always equal. In particular, as we shall see in subsequent chapters, the very strength of MD simulation is its weakness: it must follow the natural dynamics of the system. If the natural dynamics is slow, this limits the ability of an MD simulation to explore all accessible states of a system. MC suffers much less from this drawback: as we shall see, ingenious Monte Carlo schemes exist that can sample efficiently even where the natural dynamics is slow. Moreover, there are many interesting model systems, e.g., lattice models that have no natural dynamics. Such models can only be studied using Monte Carlo simulations.

3.2 The Monte Carlo method

Below, we describe the basic principles of the [Markov-Chain Monte Carlo](#) method, which from now on we will often refer to as *the Monte Carlo (MC) method*, unless explicitly stated otherwise. Initially, we focus on the basic MC method, which can be used to simulate systems with a fixed number of particles (N) in a given volume (V) at an imposed temperature (T). To keep the notation simple, we focus initially on particles with no internal degrees of freedom. To introduce the Monte Carlo method, we start from the classical expression for the partition function \mathcal{Q} , Eq. (2.2.19):

$$\mathcal{Q} = c \int d\mathbf{p}^N d\mathbf{r}^N \exp[-\mathcal{H}(\mathbf{r}^N, \mathbf{p}^N)/k_B T], \quad (3.2.1)$$

where the Hamiltonian, $\mathcal{H} \equiv \mathcal{H}(\mathbf{r}^N, \mathbf{p}^N)$ is defined as in Eq. (2.3.1). The Hamiltonian of a system expresses the total energy as a function of the coordinates and momenta of the constituent particles: $\mathcal{H} = \mathcal{K} + \mathcal{U}$, where \mathcal{K} is the kinetic energy and \mathcal{U} is the potential energy. The equilibrium average of an observable,

² Interesting accounts of the early history of the Metropolis method may be found in refs. [1,2,64, 65].

$A \equiv A(\mathbf{p}^N, \mathbf{r}^N)$ of a classical system at constant NVT is given by Eq. (2.2.20):

$$\langle A \rangle = \frac{\int d\mathbf{p}^N d\mathbf{r}^N A(\mathbf{p}^N, \mathbf{r}^N) \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}{\int d\mathbf{p}^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}, \quad (3.2.2)$$

\mathcal{K} is a quadratic function of the momenta, and, as we argued below Eq. (2.3.3), the integration over momenta can be carried out analytically, provided that A is a simple function of the momenta for which we can carry out the Gaussian integration analytically. Hence, averages of functions that depend on momenta only are usually easy to evaluate.³ The difficult part of the problem is the computation of averages of functions $A(\mathbf{r}^N)$. Only in a few exceptional cases can the multidimensional integral over particle coordinates be computed analytically; in all other cases, numerical techniques must be used.

In other words, we need a numerical technique that allows us to compute averages of functions that depend on a large number of coordinates (typically $\mathcal{O}(10^3\text{--}10^6)$). It might appear that the most straightforward approach would be to evaluate $\langle A \rangle$ in Eq. (3.2.2) by numerical quadrature, for instance, by using a high-dimensional version of Simpson's rule. However, it is easy to see that such an approach is completely useless even if the number of independent coordinates dN (d is the dimensionality of the system) is still quite small. As an example, consider the computational effort required to compute an integral of the type Eq. (3.2.2) by quadrature for a system of $N = 100$ particles. Such a calculation would involve carrying out the quadrature by evaluating the integrand on a mesh of points in the dN -dimensional configuration space. Let us assume that we take m equidistant points along each coordinate axis. The total number of points at which the integrand must be evaluated is then equal to m^{dN} . For all but the smallest systems, this number becomes astronomically large, even for small values of m . For instance, if we take 100 particles in three dimensions, and $m = 5$, then we would have to evaluate the integrand at 10^{210} points! Computations of such magnitude cannot be performed during the lifetime of the known universe. And this is fortunate because the answer that would be obtained would be subject to a large statistical error. After all, numerical quadratures work best on functions that are smooth over distances corresponding to the mesh size. But for most intermolecular potentials, the Boltzmann factor in Eq. (3.2.2) is a rapidly varying function of the particle coordinates. Hence an accurate quadrature would require a small mesh spacing (i.e., a large value of m). Moreover, when evaluating the integrand for a dense liquid (say), we would find that for the overwhelming majority of points, this Boltzmann factor is vanishingly small. For instance, for a fluid of 100 hard spheres at the freezing point, the Boltzmann factor would be non-zero for 1 out of every 10^{260} configurations! This problem is not related to the fact that the quadrature points are chosen on a mesh: estimating the integral as an average over 10^{260} randomly chosen configurations would be just ineffective.

³ However, special care is needed in systems subject to constraints; see section 10.2.1.

Clearly, other numerical techniques are needed to compute thermal averages. One such technique is the Monte Carlo method or, more precisely, the Monte Carlo importance-sampling algorithm introduced in 1953 by Metropolis et al. [6]. The application of this method to the numerical simulation of atomic and molecular systems is the subject of the present chapter.

3.2.1 Metropolis method

As argued above, it is, in general, not feasible to evaluate an integral, such as $\int d\mathbf{r}^N \exp[-\beta\mathcal{U}(\mathbf{r}^N)]$, by quadrature. However, in many cases, we are not interested in the configurational part of the partition function itself, but in averages of the type (see Eq. (2.3.9)):

$$\langle A \rangle = \frac{\int d\mathbf{r}^N \exp[-\beta\mathcal{U}(\mathbf{r}^N)] A(\mathbf{r}^N)}{\int d\mathbf{r}^N \exp[-\beta\mathcal{U}(\mathbf{r}^N)]}. \quad (3.2.3)$$

Hence, we wish to know the *ratio* of two integrals. What Metropolis et al. [6] showed is that it is possible to devise an efficient Monte Carlo scheme to sample such a ratio.

Note that the ratio $\exp(-\beta\mathcal{U})/Z$ in Eq. (3.2.3) is the probability density (\mathcal{N}) of finding the system in a configuration around \mathbf{r}^N (see Eqs. (2.3.10) and (2.3.7)):

$$\mathcal{N}(\mathbf{r}^N) \equiv \frac{\exp[-\beta\mathcal{U}(\mathbf{r}^N)]}{Z}.$$

For what follows, it is important that $\mathcal{N}(\mathbf{r}^N)$ is non-negative.

Suppose that we would somehow be able to generate L points in configuration space that are distributed according to the probability distribution $\mathcal{N}(\mathbf{r}^N)$. On average, the number of points n_j generated in a small configuration-space (hyper)volume⁴ Δ_{dN} around a point \mathbf{r}^N would then be equal to $L\mathcal{N}(\mathbf{r}^N)\Delta_{dN}$. We assume that there are M sub-volumes Δ_{dN} , which completely tile the configuration space V^N . Then

$$M = \frac{V^N}{\Delta_{dN}}.$$

In the limit of large L and large M (small Δ_{dN})

$$\langle A \rangle \approx \frac{1}{L} \sum_{j=1}^M n_j A(\mathbf{r}_i^N). \quad (3.2.4)$$

⁴ We use the notation Δ_{dN} to designate a small hyper-volume in the dN dimensional configuration space of the system.

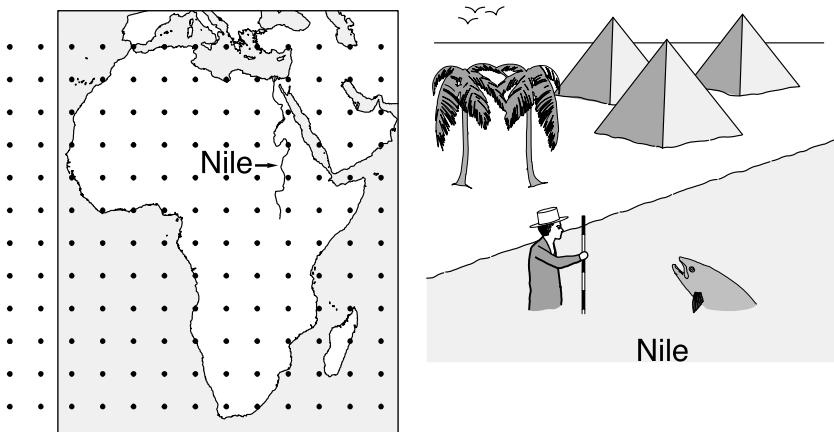


FIGURE 3.1 Measuring the depth of the Nile: a comparison of conventional quadrature (left), with the Metropolis scheme (right).

Rather than writing this as a sum over M cells, we can write this as a sum over the L points that have been sampled:

$$\langle A \rangle \approx \frac{1}{L} \sum_{i=1}^L A(\mathbf{r}_j^N). \quad (3.2.5)$$

The crucial point to note is that, in order to evaluate Eq. (3.2.5), we only need to know that n_j is *proportional* to the Boltzmann factor $\exp[-\beta U(\mathbf{r}^N)]$: the unknown normalizing factor (Z) is not needed, because n_j is normalized by $L = \sum_j n_j$. Hence, if we have a scheme that allows us to generate points with a frequency proportional to their Boltzmann weight, then we can estimate averages of the type given in Eq. (3.2.3). Now, we have to find out how to generate points with a frequency proportional to their Boltzmann weight. Such a procedure is called *importance sampling*, and the Metropolis Monte Carlo method [6] gives a simple recipe to realize such importance sampling.

The above introduction to the concept of importance sampling may sound rather abstract: let us, therefore, try to clarify the difference between quadrature and importance sampling with the help of a simple example (see Fig. 3.1). In this figure, we compare two ways of measuring the depth of the river Nile, by conventional quadrature (left) and by Metropolis sampling; that is, the construction of an importance-weighted random walk (right). In the conventional quadrature scheme, the value of the integrand is measured at a predetermined set of points. As the choice of these points does not depend on the value of the integrand, many points may be located in regions where the integrand vanishes (in the case of the Nile: one cannot perform a measurement of the depth of the Nile outside the Nile —yet most of our quadrature points are far away from the river). In contrast, in the Metropolis scheme, a random walk is constructed

through that region of space where the integrand is non-negligible (i.e., through the Nile itself). In this random walk, a trial move is rejected if it takes you out of the water, and it is accepted otherwise. After *every* trial move (accepted or not), the depth of the water is measured. The (unweighted) average of all these measurements yields an estimate of the average depth of the Nile. This, then, is the essence of the Metropolis method. In principle, the conventional quadrature scheme would yield the same answer, but it would require a very fine grid, and most sampled points would be irrelevant. The advantage of quadrature, if it were feasible, would be that it also allows us to compute the *total* area of the Nile. In the importance sampling scheme, information on the total area cannot be obtained directly, since this quantity is similar to the configurational integral Z . In later chapters, we discuss numerical schemes to compute Z .

Thus far, we have explained what is needed to compute averages by importance sampling: we must somehow be able to generate points in configuration space with a frequency proportional to their Boltzmann weight. The question is: *how?*

Ideally, we would use a scheme that generates a large number of independent points in configuration space with the desired Boltzmann weights. Such an approach is called *static* Monte Carlo sampling. Static MC has the advantage that every configuration is statistically independent of the previous ones. Traditionally, such an approach has been used in very simple cases, e.g., the generation of points that are normally distributed, in which case we can analytically map random numbers uniformly distributed between zero and one onto a Gaussian distribution [66]. However, for any nontrivial many-body system, such a rigorous analytical approach is not feasible.

This is why the overwhelming majority of MC simulations are *dynamic*: the simulation generates a (Markov) chain of configurations, where a new configuration is generated with an easily computable probability from the previous one. The Metropolis Monte Carlo method is the prime example of such a dynamic scheme. Using dynamic MC, it is possible to generate configurations with the correct Boltzmann weight. However, the disadvantage of the scheme is that successive configurations tend to be correlated. Hence, the number of statistically independent configurations is less than the total number of configurations. Before continuing with a description of (dynamic) MC methods, we note that static (or hybrid static/dynamic) MC methods are undergoing a revival due to the application of transformations that are obtained by Machine Learning (see section 13.4.1). However, in the present chapter, and most of those that follow, we focus on dynamic Monte Carlo schemes.

To rationalize the Metropolis algorithm, let us first consider the consequence of the fact that the desired algorithm must sample points with a frequency proportional to their Boltzmann weight. This implies that if we have L points that are already Boltzmann distributed, then applying one step of our algorithm on every single point should not destroy this Boltzmann distribution. Note that this is an “ensemble” argument: we consider a very large number (L) of copies of

the same model system. It is useful to consider the limit where $L \gg M$, where M denotes the number of points in configuration space. Of course, in reality, the number of points in configuration space is not countable—but in a numerical representation, it is, as floating point numbers have a limited number of digits. We now assume that the number of copies of the system in a given state (say i) is proportional to the Boltzmann factor of that state. Next, we carry out one step of our MC algorithm on all L systems. As a consequence of this step, individual systems may end up in another state, but, on average, the number of systems in every state should not change, as these numbers are given by the Boltzmann distribution.

Let us consider one such state, which we denote by o (for “old”). State o has an unnormalized Boltzmann weight, $\exp[-\beta\mathcal{U}(o)]$. We can compute this Boltzmann weight because we can compute the potential energy. A Monte Carlo algorithm is not deterministic: a single step can have different outcomes with different probabilities (hence the name “Monte Carlo”). Let us consider the probability of moving during one MC step from state o to a new state n with Boltzmann weight $\exp[-\beta\mathcal{U}(n)]$. We denote this transition probability by $\pi(o \rightarrow n)$. Let us denote the number of copies of our system that are originally in state o , by $m(o)$. As we start with a Boltzmann distribution, $m(o)$ is proportional to $\exp[-\beta\mathcal{U}(o)]$. To maintain the Boltzmann distribution, $m(n)$, the number of copies in state n should be proportional to $\exp[-\beta\mathcal{U}(n)]$. The transition probability $\pi(o \rightarrow n)$ should be constructed such that it does not destroy the Boltzmann distribution. This means that, in equilibrium, the average number of accepted trial moves that result in the system leaving state o must be exactly balanced by the number of accepted trial moves from all other states n to state o . We note that $\langle m(o) \rangle$, the average number of systems in state o , is proportional to $\mathcal{N}(o)$ (and similarly for n). The transition probability $\pi(o \rightarrow n)$ should then satisfy the following balance equation:

$$\mathcal{N}(o) \sum_n \pi(o \rightarrow n) = \sum_n \mathcal{N}(n) \pi(n \rightarrow o). \quad (3.2.6)$$

Eq. (3.2.6) expresses the balance between the flow into and the flow out of state o . Note that $\pi(o \rightarrow n)$ can be interpreted as a matrix (the “transition matrix”) that maps the original states of the system onto the new states. In this matrix interpretation, we require that the Boltzmann distribution is an eigenvector of the transition matrix, with eigenvalue 1. We also note that the probability of making a transition from a given state o to a new state n depends only on the current state of the system. Hence, the process described by the matrix $\pi(o \rightarrow n)$ is a Markov process.

The original Metropolis algorithm was based on a stronger condition than Eq. (3.2.6), namely that in equilibrium, the average number of accepted moves from o to any given state n is *exactly* balanced by the number of reverse moves.

This condition is known under the name detailed balance⁵:

$$\mathcal{N}(o)\pi(o \rightarrow n) = \mathcal{N}(n)\pi(n \rightarrow o). \quad (3.2.7)$$

Many possible forms of the transition matrix $\pi(o \rightarrow n)$ satisfy Eq. (3.2.7). To progress towards the Metropolis algorithm, we make use of the fact that we can decompose a Monte Carlo move into two stages. First, we carry out a trial move from state o to state n . We denote the transition matrix that determines the probability of performing a trial move from o to n by $\alpha(o \rightarrow n)$, where α is usually referred to as the underlying matrix of the Markov chain [67].

The next stage involves the decision to either accept or reject this trial move. Let us denote the probability of accepting a trial move from o to n by $\text{acc}(o \rightarrow n)$. Clearly,

$$\pi(o \rightarrow n) = \alpha(o \rightarrow n) \times \text{acc}(o \rightarrow n). \quad (3.2.8)$$

In the original Metropolis scheme, α was chosen to be a symmetric matrix ($\alpha(o \rightarrow n) = \alpha(n \rightarrow o)$). However, in later sections, we shall see several examples where α is chosen *not* to be symmetric. If α is symmetric, we can rewrite Eq. (3.2.6) in terms of the $\text{acc}(o \rightarrow n)$:

$$\mathcal{N}(o) \times \text{acc}(o \rightarrow n) = \mathcal{N}(n) \times \text{acc}(n \rightarrow o). \quad (3.2.9)$$

From Eq. (3.2.9) it follows that⁶

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}. \quad (3.2.10)$$

Again, many choices for $\text{acc}(o \rightarrow n)$ satisfy this condition, subject, of course, to the constraint that the probability $\text{acc}(o \rightarrow n)$ cannot exceed one. The choice of Metropolis et al. was

$$\begin{aligned} \text{acc}(o \rightarrow n) &= \mathcal{N}(n)/\mathcal{N}(o) && \text{if } \mathcal{N}(n) < \mathcal{N}(o) \\ &= 1 && \text{if } \mathcal{N}(n) \geq \mathcal{N}(o). \end{aligned} \quad (3.2.11)$$

Other choices for $\text{acc}(o \rightarrow n)$ are possible (for a discussion, see for instance [21]), but under most conditions, the original choice of Metropolis et al. appears to result in a more efficient sampling of configuration space than the other strategies that have been proposed.

⁵ In Chapter 13 we discuss powerful MC algorithms that do not satisfy detailed balance. They do, however, satisfy *balance*, and therefore maintain the Boltzmann distribution.

⁶ Clearly, Eq. (3.2.10) only makes sense if \mathcal{N} is non-negative.

In summary: in the Metropolis scheme, the transition probability for going from state o to state n is given by

$$\begin{aligned}\pi(o \rightarrow n) &= \alpha(o \rightarrow n) & \mathcal{N}(n) &\geq \mathcal{N}(o) \\ &= \alpha(o \rightarrow n)[\mathcal{N}(n)/\mathcal{N}(o)] & \mathcal{N}(n) &< \mathcal{N}(o) \\ \pi(o \rightarrow o) &= 1 - \sum_{n \neq o} \pi(o \rightarrow n).\end{aligned}\quad (3.2.12)$$

Note that we have not yet specified the matrix α , except for the fact that it must be symmetric. Indeed, we have considerable freedom in choosing the symmetric matrix α . This freedom will be exploited in subsequent sections.

One thing that we have not yet explained is how to decide whether a trial move is to be accepted or rejected. The usual procedure is as follows. Suppose that we have generated a trial move from state o to state n , with $\mathcal{U}(n) > \mathcal{U}(o)$. According to Eq. (3.2.10) this trial move should be accepted with a probability

$$\text{acc}(o \rightarrow n) = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} < 1.$$

In order to decide whether to accept or reject the trial move, we generate a (quasi) random number, here denoted by \mathcal{R} from a uniform distribution in the interval $[0, 1]$. Clearly, the probability that \mathcal{R} is less than $\text{acc}(o \rightarrow n)$ is equal to $\text{acc}(o \rightarrow n)$. We now accept the trial move if $\mathcal{R} < \text{acc}(o \rightarrow n)$ and reject it otherwise. This rule guarantees that the probability of accepting a trial move from o to n is indeed equal to $\text{acc}(o \rightarrow n)$. Obviously, it is important that our quasi-random-number generator does generate numbers uniformly in the interval $[0, 1]$. Otherwise, the Monte Carlo sampling will be biased. The quality of random-number generators should never be taken for granted. A good discussion of random-number generators can be found in *Numerical Recipes* [38] and in *Monte Carlo Methods* by Kalos and Whitlock [37].

Thus far, we have not mentioned another condition that the matrix $\pi(o \rightarrow n)$ should satisfy, namely that it must be *irreducible* (i.e., every accessible point in configuration space can be reached in a finite number of Monte Carlo steps from any other point). Irreducibility plays the same role in MC as ergodicity in MD, and to facilitate the comparison, we will use the terms interchangeably. Although some simple MC schemes are guaranteed to be irreducible, these are often not the most efficient schemes. Conversely, many efficient Monte Carlo schemes have either not been proven to be irreducible or, worse, been proven to violate irreducibility. The solution is usually to mix the efficient, non-ergodic scheme with an occasional trial move of the less-efficient but ergodic scheme. The method as a whole will then be ergodic (at least, in principle).

The one criterion that MC moves do *not* have to satisfy is generating physically plausible trajectories. In this respect, MC differs fundamentally from MD, and what sounds like a disadvantage is often a strength. As we shall see later, “unphysical” moves MC can greatly speed up the exploration of configuration space and ensure irreducibility. However, in numerous applications, MC algorithms are designed to generate physically plausible trajectories, in particular

when simulating particles that undergo diffusive Brownian motion. We will mention a few examples later (see section 13.3.2). At this stage, we only note that all these schemes must also be valid MC schemes. Hence the present discussion applies.

3.2.2 Parsimonious Metropolis algorithm

In the standard Metropolis MC method, we first compute the energy change due to a trial move, and then draw a random number to decide on its acceptance. In the so-called **Parsimonious Metropolis Algorithm** (**PMA**) [68], the first step is drawing a random number \mathcal{R} uniformly distributed between 0 and 1. Obviously, the Metropolis rule implies that trial moves with $\beta\Delta E < -\ln \mathcal{R}$ will be accepted, and so it would seem that nothing is gained by computing \mathcal{R} first. However, the **PMA** exploits the fact that it is often possible to place bounds ΔE_{\max} and ΔE_{\min} on ΔE . Clearly, trial moves with $\beta\Delta E_{\max} < -\ln \mathcal{R}$ will be accepted, and those with $\beta\Delta E_{\min} > -\ln \mathcal{R}$ will be rejected. The idea behind the **PMA** is to use bounds that can be (pre)computed cheaply. How this is achieved depends on the specific implementation: for more details, see ref. [68]. We will see related examples where \mathcal{R} is computed first in section 13.4.4 (Eq. (13.4.30)) and, in a slightly different form, in the kinetic Monte Carlo algorithm described in Section 13.3.2 (see Eq. (13.3.5)).

3.3 A basic Monte Carlo algorithm

It is not very helpful to discuss Monte Carlo or Molecular Dynamics algorithms in abstract terms. The best way to explain how they work is to write them down. This will be done in the present section.

The core of most Monte Carlo or Molecular Dynamics programs is simple: typically only a few hundred lines long. These MC or MD algorithms may be embedded in standard program packages or programs that have been tailor-made for specific applications. All these programs and packages tend to be different, and there is, therefore, no such thing as a standard Monte Carlo or **Molecular Dynamics** program. However, the cores of most MD/MC programs are, if not identical, at least very similar—if you understand one, you understand them all.

Below, we shall construct the “core” of an MC program. It will be rudimentary, and efficiency has been traded for clarity. But it should demonstrate how the Monte Carlo method works. In the next chapter, we will do the same for a Molecular Dynamics program.

3.3.1 The algorithm

As explained in the previous section, the Metropolis (Markov-Chain) Monte Carlo method is constructed such that the probability of visiting a particular point \mathbf{r}^N in configuration space is proportional to the Boltzmann factor

Algorithm 1 (Core of Metropolis MC code)

program MC	basic Metropolis algorithm
for $1 \leq i_{\text{trial}} \leq n_{\text{trial}}$ do	perform n_{trial} MC trial moves
mcmove	trial move procedure
if ($i_{\text{trial}} \% n_{\text{sample}}$) == 0 then	% denotes the Modulo operation
sample	sample procedure
endif	
enddo	
end program	

Specific Comments (for general comments, see p. 7)

1. Function **mcmove** attempts to displace a randomly selected particle (see Algorithm 2).
2. Function **sample** samples observables every n_{sample} -th trial move.

$\exp[-\beta\mathcal{U}(\mathbf{r}^N)]$. In the standard implementation of the approach introduced by Metropolis et al. [6], we use the following scheme:

1. Select a particle at random,⁷ and calculate its contribution $\mathcal{U}(\mathbf{r}^N)$ to the energy of the system.⁸
2. Give the particle a random displacement, $\mathbf{r}' = \mathbf{r} + \Delta$, and calculate its new potential energy $\mathcal{U}(\mathbf{r}'^N)$.
3. Accept the move from \mathbf{r}^N to \mathbf{r}'^N with probability

$$\text{acc}(o \rightarrow n) = \min\left(1, \exp\{-\beta[\mathcal{U}(\mathbf{r}'^N) - \mathcal{U}(\mathbf{r}^N)]\}\right). \quad (3.3.1)$$

An implementation of this basic Metropolis scheme is shown in Algorithms 1 and 2.

3.3.2 Technical details

Computing averages in Monte Carlo simulations can be accelerated using a number of tricks. Ideally, schemes to save computer time should not affect the results of a simulation in a systematic way, but that is often not completely true.

⁷ Selecting particles at random is standard practice in MC because other choices yield algorithms that do not satisfy detailed balance. Interestingly, the original Metropolis algorithm [6] moved particles sequentially, and hence violated detailed balance. It did, however, satisfy the balance condition. In section 13.4.4, we will discuss how breaking detailed balance may speed up the convergence of MC simulations. An explicit description of trial moves involving a particle selected at random can be found in the 1957 paper by Wood and Parker [69].

⁸ This is straightforward for pairwise additive interactions. However, for systems with many-body interactions, it would be that part of the potential energy of the system that depends on the coordinate of the selected particle. In the worst case, this may simply be the total potential energy of the system.

Algorithm 2 (Monte Carlo trial displacement)

<pre>function mcmove i=int($\mathcal{R} \cdot npart$)+1 eno = ener(x(i),i) xn=x(i)+($\mathcal{R} - 0.5$)$\cdot delx$ enn = ener(xn,i) if $\mathcal{R} < \exp[-\beta * (enn - eno)]$ then x(i)=xn endif end function</pre>	Metropolis MC trial move select random particle with $1 \leq i \leq npart$ eno: energy particle i at “old” position $x(i)$ trial position xn for particle i enn: energy of i at xn Metropolis criterion Eq. (3.3.1) if accepted, $x(i)$ becomes xn
---	--

Specific Comments (*for general comments, see p. 7*)

1. *npart: number of particles. $x(npart)$ position array. $T = 1/\beta$, maximum steps size = $0.5 \cdot delx$*
2. *The function $\text{ener}(x)$: computes the interaction energy of a particle at position x , using the approach shown in Algorithm 5.*
3. *\mathcal{R} generates a random number uniformly between 0 and 1*
4. *$\text{int}(z)$ returns the integer part of z*
5. *Note that, if a configuration is rejected, the old configuration is retained.*

The main reason is that simulations of macroscopic systems are not feasible: we, therefore, carry out simulations on microscopic (hundreds to millions of particles) systems. The properties of such finite systems are often slightly, but sometimes very different from those of macroscopic systems. Similarly, even for moderately-sized systems, it is often advantageous to avoid explicit calculation of very weak intermolecular interactions between particles that are far apart. In both cases, the effect of these approximations can be mitigated, as we discuss below. But, whatever the situation, the key point is that we should be aware of possible systematic errors introduced by time-saving tricks and correct them wherever possible.

Many of the time-saving devices that we discuss below are similar for MC and MD simulations. Rather than repeat the present section in the chapter on MD, we shall refer in our discussion below to both types of simulations whenever this is relevant. However, we will not yet assume that the reader is familiar with the technical details of MD simulations. The only feature of MD that, at this stage, is relevant for our discussion is that in MD simulations, we must compute the forces acting on all particles.

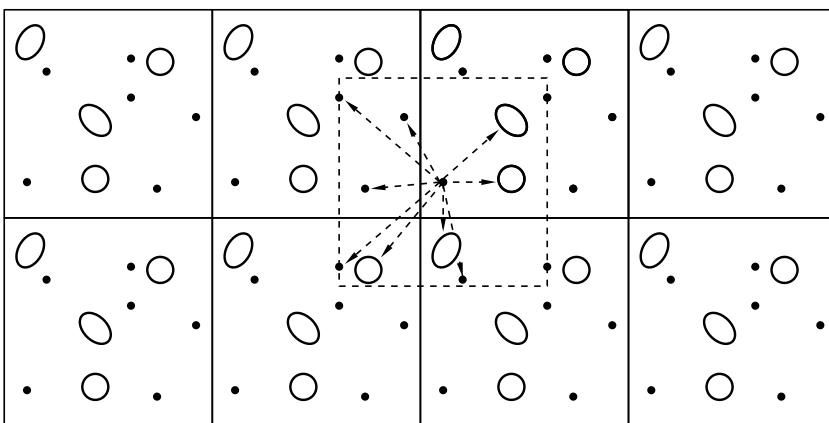


FIGURE 3.2 Schematic representation of periodic boundary conditions.

3.3.2.1 Boundary conditions

Monte Carlo and Molecular Dynamics simulations of atomic or molecular systems aim to provide information about the properties of a macroscopic sample. Yet, the number of degrees of freedom that are usually studied in molecular simulations ranges from hundreds to millions.⁹ Clearly, such particle numbers are still far removed from the thermodynamic limit. In particular, for small systems it cannot be safely assumed that the choice of the boundary conditions (e.g., free or hard or periodic) has a negligible effect on the properties of the system. In fact, in a three-dimensional N -particle system with free boundaries, the fraction of all molecules that is at the surface is proportional to $N^{-1/3}$. For instance, in a simple cubic crystal of 1000 atoms, some 49% of all atoms are at the surface, and for 10^6 atoms, this fraction has only decreased to 6%. As a consequence, we should expect that the properties of such small systems will be strongly influenced by surface effects.

To simulate bulk phases, it is essential to choose boundary conditions that mimic the presence of an infinite bulk surrounding our N -particle model system. This is usually achieved by employing periodic boundary conditions. The volume containing the N particles is treated as the primitive cell of an infinite periodic lattice of identical cells (see Fig. 3.2). A given particle (i , say) now interacts with all other particles in this infinite periodic system, that is, all other particles in the same periodic cell and all particles (including its own periodic image) in all other cells. For instance, if we assume that all intermolecular interactions are pairwise additive, then the total potential energy of the N particles

⁹ Larger systems containing billions of particles can be simulated, but in view of the high computational cost, such simulations will only be performed if the essential physics cannot be captured by smaller systems.

in any one periodic box is

$$\mathcal{U}_{\text{tot}} = \frac{1}{2} \sum'_{i,j,\mathbf{n}} u(|\mathbf{r}_{ij} + \mathbf{n}L|),$$

where L is the diameter of the periodic box (assumed cubic, for convenience) and \mathbf{n} is an arbitrary vector of three integer numbers, while the prime over the sum indicates that the term with $i = j$ is to be excluded when $\mathbf{n} = \mathbf{0}$. In this general form, periodic boundary conditions are not particularly useful, because to simulate bulk behavior, we had to rewrite the potential energy as an infinite sum rather than a finite one.¹⁰ As discussed in the next section, we are in practice often dealing with short-range interactions. In that case, it is usually permissible to truncate all intermolecular interactions beyond a certain cutoff distance r_c , and account approximately for all longer-ranged interactions.

Although the use of periodic boundary conditions is a surprisingly effective method for emulating homogeneous bulk systems, one should always be aware of the fact that such boundary conditions still cause spurious correlations that are not present in a macroscopic bulk system. In particular, one consequence of the periodicity of the model system is that only fluctuations with a wavelength compatible with the periodicity of the boundary conditions are allowed: the longest wavelength that still fits in the periodic box, is the one for which $\lambda = L$. In cases where long-wavelength fluctuations are expected to be important (as, for instance, near a critical point), one should expect problems with the use of periodic boundary conditions. Periodic boundary conditions also break the rotational symmetry of the system. One consequence is that the average density profile around a particle in an isotropic fluid is not spherically symmetric, but reflects the cubic (or lower) symmetry of the periodic boundary conditions [70]. Such effects become less important with increasing system size.

Periodic Boundaries: a misnomer?

The term *Periodic Boundary Conditions* sometimes creates confusion, due to the use of the word *boundary*. The origin of the periodic lattice of primitive cells may be chosen anywhere in the model system under study, and this choice will not affect any observable property. Hence, the “boundary” has no physical significance (see, however, section 11.2.2). In contrast, the shape of the periodic cell and its orientation may not always be chosen at will. In particular, for crystals or other systems with positional ordering, the shape and orientation of the simulation box must be compatible with the intrinsic periodicity of the physical system.

¹⁰ In fact, in the first MC simulation of three-dimensional Lennard-Jones particles, Wood and Parker [69] discuss the use of such infinite sums in relation to the now conventional approach discussed here.

3.3.2.2 Truncation of interactions

Let us consider the specific case of a simulation of a system with *short-ranged*, pairwise-additive interactions. In this context, short-ranged means that the total potential energy of a given particle i is dominated by interactions with neighboring particles that are closer than some cutoff distance r_c . The error that results from ignoring interactions with particles at larger distances can be made arbitrarily small by choosing r_c sufficiently large. When using periodic boundary conditions, the case where r_c is less than $L/2$ (half the diameter of the periodic box) is of special interest because, in that case, we need to consider the interaction of a given particle i only with the nearest periodic image (sometimes called the *Minimum Image*) of any other particle j (see the dashed box in Fig. 3.2). Strictly speaking, one could also choose to limit the computation of the pair interaction to the nearest image, without imposing a spherical cut-off. However, in that case, the pair interaction at distances larger than $L/2$ would depend on the direction of the line joining the centers of the particles. Obviously, such a spurious angle-dependence could cause artifacts and is usually to be avoided (but not always: see Chapter 11).

If the intermolecular potential is not rigorously zero for $r \geq r_c$, truncation of the intermolecular interactions at r_c will result in a systematic error in \mathcal{U}^{tot} . As we discuss below, it is often possible to correct approximately the truncation of the pair potential.

Ideally, the computed values of observable properties should be insensitive to the precise choice of the cut-off distance of the pair potential, provided we correct for the effect of the cut-off procedure. However, for the most common choices of the cut-off distance, different cut-off procedures can yield different answers, and the consequences have been serious. For instance, the literature is full of studies of models with Lennard-Jones-like pair potentials. However, different authors often use different cut-off procedures and, as a result, it is difficult to compare different simulations that claim to study the same model.

Several factors make truncation of the potential a tricky business. First of all, it should be realized that although the absolute value of the potential energy function decreases with inter-particle separation r , for sufficiently large r , the number of neighbors is a rapidly increasing function of r : the number of particles at a distance r from a given atom increases asymptotically as r^{d-1} , where d is the dimensionality of the system. As an example, let us compute the effect of truncating the pair potential for a simple model system. In a homogeneous phase, the average potential energy (in three dimensions) of a given atom i is given by

$$u_i = (1/2) \int_0^\infty dr 4\pi r^2 \rho(r) u(r),$$

where $\rho(r)$ denotes the average number density at a distance r from atom i . The factor $(1/2)$ has been included to correct for double counting of intermolecular interactions. If we truncate the potential at a distance r_c , we ignore the tail

contribution u^{tail} :

$$u^{\text{tail}} \equiv (1/2) \int_{r_c}^{\infty} dr 4\pi r^2 \rho(r) u(r), \quad (3.3.2)$$

where we have dropped the subscript i because in a bulk fluid, all atoms in the system experience an equivalent environment. To simplify the calculation of u^{tail} , it is commonly assumed that for $r \geq r_c$, the density $\rho(r)$ is equal to the average number density ρ in the bulk of the fluid.

If the intermolecular interactions decay sufficiently rapidly, one may correct for the systematic error that results from truncating the potential at r_c by adding a tail contribution to \mathcal{U}^{tot} :

$$\mathcal{U}^{\text{tot}} = \sum_{i < j} u^{\text{trunc}}(r_{ij}) + \frac{N\rho}{2} \int_{r_c}^{\infty} dr u(r) 4\pi r^2, \quad (3.3.3)$$

for a three-dimensional atomic system. The extension of Eq. (3.3.3) to other dimensions is straightforward. In Eq. (3.3.3), $u^{\text{trunc}}(r)$ stands for the pair potential energy function that has been truncated for distances r larger than the cut-off distances r_c , and ρ is the average number density in the bulk of the fluid.

The approximation that results from the assumption that the density $\rho(r)$ is equal to the bulk density ρ for $r > r_c$ becomes worse for smaller choices for r_c . From Eq. (3.3.3), it can be seen that the tail correction to the potential energy diverges unless the potential energy function $u(r)$ decays more rapidly than r^{-3} (in three dimensions). This condition is satisfied if the long-range interaction between molecules is dominated by dispersion forces. However, for the important case of Coulomb or dipolar interactions, the tail correction diverges, and hence the nearest-image convention cannot be used for such systems. In such cases, the interactions with all periodic images should be taken into account explicitly. Chapter 11 discusses how this problem can be addressed.

To give explicit examples of truncation procedures that have been used in the literature, let us consider the Lennard-Jones (LJ) [71,72] potential that is very popular in molecular simulations¹¹ Lennard-Jones 12 – 6 potential is of the following form:

$$u_{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right]. \quad (3.3.4)$$

In what follows, we shall refer to this potential as the Lennard-Jones potential. For a cut-off distance r_c , the tail correction u^{tail} for the LJ pair potential in three

¹¹ Lennard-Jones-type potentials of the form $r^{-n}r^{-m}$ were first introduced by J.E. Lennard-Jones when he was still called J.E. Jones [71], but the choice for the 12 – 6 form was only made a few years later [72]. The popularity of the LJ 12–6 potential (as it is commonly called) in simulations was initially due to the fact that it was cheap to evaluate and that it worked well (be it for the wrong reasons [73]) for liquid argon. By now, its popularity is mainly due to the fact that ... it is so widely used.

dimensions becomes

$$\begin{aligned} u^{\text{tail}} &= \frac{1}{2} 4\pi\rho \int_{r_c}^{\infty} dr r^2 u(r) \\ &= \frac{1}{2} 16\pi\rho\epsilon \int_{r_c}^{\infty} dr r^2 \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] \\ &= \frac{8}{3}\pi\rho\epsilon\sigma^3 \left[\frac{1}{3} \left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3 \right]. \end{aligned} \quad (3.3.5)$$

For a cutoff distance $r_c = 2.5\sigma$ the potential has decayed to a value that is about 1/60th of the well depth. This seems to be a small value, but in fact, the tail correction is usually non-negligible. For instance, at a typical liquid density, $\rho\sigma^3 = 1$, we find $u^{\text{tail}} = -0.535\epsilon$. This number is certainly not negligible compared to the total potential energy per atom (almost 10% at a typical liquid density); hence although we can truncate the potential at 2.5σ , we cannot ignore the effect of this truncation. Note that the tail correction described above assumes that we consider a homogeneous system. Near an interface, the simple expressions for the tail correction lose their validity, and, in fact, computed surface tensions are depressingly sensitive to the cut-off procedure used [74]. For adsorption in solids, Jablonka et al. [75] showed that adding tail corrections helps to get more consistent results.

There are several ways to truncate interactions in a simulation —some truncate the potential, some the force, and some make the force vanish continuously at r_c . Although all these methods are designed to yield similar results, it should be realized that they yield results that may differ significantly, in particular in the vicinity of critical points [76–78] (see Fig. 3.3).

Frequently used methods to truncate the potential are

1. Simple truncation
2. Truncation and shift
3. Truncation and force-shift (i.e., both $u(r)$ and the force are made to vanish at r_c).

Simple truncation

The simplest method to truncate potentials is to ignore all interaction beyond r_c . In that case, we would be simulating a model with a potential of the form:

$$u^{\text{trunc}}(r) = \begin{cases} u_{\text{LJ}}(r) & r \leq r_c \\ 0 & r > r_c \end{cases}. \quad (3.3.6)$$

As mentioned, such a truncation may result in an appreciable error in our estimate of the potential energy of the true Lennard-Jones model (3.3.4). Moreover, as the truncated pair potential changes discontinuously at r_c , such a potential is not suited for standard Molecular Dynamics simulations.

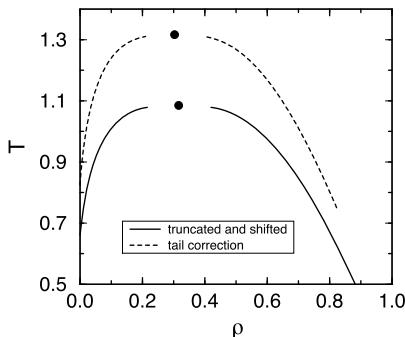


FIGURE 3.3 Vapor-liquid coexistence curves for three-dimensional Lennard-Jones models with different truncations. The figure shows the effect of the truncation of the potential on the estimated location of the critical point (large black dots). The upper curve gives the phase envelope for the Lennard-Jones potential truncated at $r_c = 2.5\sigma$ and with a tail correction added to account for interactions at distances larger than r_c . The lower curve shows the estimated vapor-liquid coexistence curve for a potential that is used in Molecular Dynamics simulations, namely the truncated and shifted potential (also with $r_c = 2.5\sigma$) [78].

Truncated potentials could be used in Monte Carlo simulations, and sometimes they are (e.g., in simulations of simple model systems with square-well or square-shoulder potentials). However, in such cases, one should be aware of the fact that there is an “impulsive” contribution to the pressure due to the discontinuous change of the potential at r_c . For otherwise continuous potentials, it is not recommended to use truncation pure and simple (although, if necessary, the impulsive contribution to the pressure could be computed).

The usual method to compensate (approximately) for the truncation of the pair potential is to add tail corrections to the structural properties that we compute. The tail correction for the potential energy was discussed above, Eq. (3.3.5). We will discuss the computation of the pressure in Chapter 5, Eq. (5.1.21). For the sake of completeness, we here give the expression for the tail correction to the pressure in three dimensions:

$$\begin{aligned}\Delta P^{\text{tail}} &= \frac{4\pi\rho^2}{6} \int_{r_c}^{\infty} dr r^2 \mathbf{r} \cdot \mathbf{f}(r) \\ &= \frac{16}{3}\pi\rho^2\epsilon\sigma^3 \left[\frac{2}{3} \left(\frac{\sigma}{r_c} \right)^9 - \left(\frac{\sigma}{r_c} \right)^3 \right],\end{aligned}\quad (3.3.7)$$

where, in the second line, we consider the specific example of a Lennard-Jones potential. Typically, the tail correction to the pressure is even larger than to the potential energy. For the LJ model at $\rho = 1$ with $r_c = 2.5$, $\Delta P^{\text{tail}} \approx -1.03$, which is clearly very substantial.

Truncated and shifted

In Molecular Dynamics simulations, it is important that forces are not singular and it is desirable to make even higher derivatives of the pair potential continuous.

The simplest procedure just shifts the pair potential so that it vanishes continuously at r_c . Then the force is not singular, but its first derivative is. The truncated-and-shifted potential is

$$u^{\text{tr-sh}}(r) = \begin{cases} u(r) - u(r_c) & r \leq r_c \\ 0 & r > r_c. \end{cases} \quad (3.3.8)$$

Of course, the potential energy and pressure of a system with a truncated and shifted potential differ from the values obtained with untruncated pair potentials. As before, we can approximately correct for the effect of the modification of the intermolecular potential on both the potential energy and the pressure. For the pressure, the tail correction is the same as in Eq. (3.3.7). For the potential energy, we must add, in addition to the long-range correction (3.3.5), a contribution equal to the average number of particles that are within a distance r_c from a given particle, multiplied by half the value of the (untruncated) pair potential at r_c . The factor of one-half is included to correct for overcounting of the intermolecular interactions.

One should be careful when applying truncated and shifted potentials in models with anisotropic interactions. In that case, the truncation should not be carried out at a fixed value of the distance between the molecular centers of mass but at a point where the pair potential has a fixed value, because otherwise the potential cannot be shifted to 0 for all points on the cut-off surface. For Monte Carlo simulations, this creates no serious problems, but for Molecular Dynamics simulations, this would be quite disastrous, as the system would no longer conserve energy, unless the impulsive forces due to the truncating and shifting are taken into account explicitly.

Truncated and force-shifted

To make also the forces vanish continuously at r_c , one can use a truncated, shifted and force-shifted potential. There are many choices for potentials that have this property. The simplest is:

$$u^{\text{tr-sh}}(r) = \begin{cases} u(r) - u(r_c) + f(r_c)(r - r_c) & r \leq r_c \\ 0 & r > r_c. \end{cases} \quad (3.3.9)$$

Again, the corresponding tail corrections can be worked out. An example of such a truncated and force-shifted Lennard-Jones potential is the one proposed by Kincaid and Hafskjold [79,80].

3.3.2.3 Whenever possible, use potentials that need no truncation

As should be clear from the above discussion, the use of different cut-off procedures for the pair potential creates many problems. Of course, if we need to use models where the use of truncation and tail correction is inevitable, we have no choice.

The Lennard-Jones potential is very often simply used as a “typical” short-ranged inter-atomic potential, where the precise behavior at larger distances is irrelevant. Under those conditions, one would be much better off using a cheap potential that vanishes quadratically at r_c . One such potential is [81]:

$$u(r)_{WF} \equiv \epsilon \begin{cases} \left[\left(\frac{\sigma}{r} \right)^2 - 1 \right] \left[\left(\frac{r_c}{r} \right)^2 - 1 \right]^2 & \text{for } r \leq r_c \\ = 0 & \text{for } r > r_c. \end{cases} \quad (3.3.10)$$

The Wang-Ramirez-Dobnikar-Frenkel (WF) potential is Lennard-Jones-like for $r_c = 2$, whereas, for $r_c = 1.2$, it can describe the typical short-ranged interaction between colloids (see [81]). In this book, we use this potential in many exercises, because it eliminates the need for tail corrections.

As shown in the preceding paragraphs, the cost of computing pairwise-additive interactions can be reduced by only considering explicitly the interactions between particles that are less than a cut-off distance r_c apart. However, even if we use a cut-off procedure, we still have to decide whether a given pair is within a distance r_c . As there are $N(N - 1)/2$ (nearest image) distances in an N -particle system, it would seem that we would have to compute $\mathcal{O}(N^2)$ distances, before we can compute the relevant interactions. In the most naive simulations (which are, in fact, quite adequate for systems that are less than some 3-4 r_c in diameter), this is exactly what we would do. However, for larger systems (say 10^6 particles), $N(N - 1)/2$ corresponds to some 10^{12} distances, and that makes the simulation unnecessarily expensive. Fortunately, there exist more advanced schemes that allow us to pre-sort the particles in such a way that evaluating all truncated interactions only requires $\mathcal{O}(N)$ operations. Some of these tricks are discussed in Appendix I. However, in order to *understand* a basic MC or MD algorithm, these time-saving devices are not essential. Hence, in the examples that we discuss in section 4.2.2, we will describe a Molecular Dynamics program that uses a simple $\mathcal{O}(N^2)$ algorithm to compute all interactions (see Algorithm 5).

3.3.2.4 Initialization

To start a simulation, we should assign initial positions to all particles in the system. As the equilibrium properties of the system do not (or, at least, should not) depend on the choice of initial conditions, all reasonable initial conditions are in principle acceptable. If we wish to simulate the solid state of a particular model

system, it is logical to prepare the system in the crystal structure of interest. In contrast, if we are interested in the fluid phase, there are several options: one that is increasingly popular for systems with continuous interactions, is to prepare the system in a random configuration (which will typically have very high energy) and then relax the system (by energy minimization) to a less pathological configuration. Such an approach is less suited for dense systems with hard-core interactions, as minimization does not work for such systems. For such systems, but in fact also for all others, one could simply prepare the system in any convenient crystal structure at the desired target density and temperature. If we are lucky, this crystal will melt spontaneously during the early stages of a simulation (i.e., during the *equilibration run*) because at the temperature and the density of a typical liquid-state point, the solid-state may not be mechanically stable. However, one should be careful here, because the crystal structure may be metastable, even if it is not thermodynamically stable. It is, therefore, unwise to use a crystal structure as the starting configuration for a simulation of a liquid close to the freezing curve. In such cases, it is better to start the simulation from the final (liquid) configuration of a system at a higher temperature or a lower density, where the solid is not even metastable. In any event, to speed up equilibration, it is often advisable to use the final (well-equilibrated) configuration of an earlier simulation at a nearby state point (if such a configuration is available) as the starting configuration for a new run and adjust the temperature and density to the desired values.

The equilibrium properties of a system should not depend on the initial conditions. If such a dependence nevertheless is observed in a simulation, there are two possibilities. The first is that our results reflect the fact that the system that we simulate really behaves non-ergodically. This is the case, for instance, in glassy materials or low-temperature, substitutionally disordered alloys. The second, more common, explanation is that the system that we simulate is ergodic but our sampling of configuration space is inadequate; in other words, we have not yet reached equilibrium.

3.3.2.5 Reduced units

One way to save time in simulations is avoiding duplication, i.e., avoiding doing the same simulation twice for the same state point. At first sight, one might think that such duplications could only happen by mistake, but that is not quite true. Suppose that one group has studied the thermodynamic properties of the Lennard-Jones model for argon at 60 K and a density of 840 kg/m³. Another group wishes to study the properties of the Lennard-Jones model for xenon at 112 K and a density of 1617 kg/m³. One might expect that these simulations would yield quite different results. However, if, instead of using SI units, we use as our units the natural units of the model (i.e., the LJ diameter σ , the LJ well-depth ϵ , and the molecular mass m), then in these “reduced” units the two state-points are identical. The Ar results can simply be rescaled to obtain the Xe results.

For the above example of Ar and Xe, both simulations would be characterized by a reduced density $(N\sigma^3/V) \equiv \rho^* = 0.5$ and a reduced temperature $k_B T/\epsilon \equiv T^* = 0.5$. The use of reduced units allow us to characterize the state of the system by dimensionless numbers is very common in science and engineering. For instance, dimensionless groups, such as the Reynolds number (vL/η) or the Peclet number (vL/D) are widely used in fluid dynamics to compare the properties of systems that, at first sight, might seem quite different: think of a model airplane being tested in a wind tunnel to predict the behavior of a real airplane under the same (or at least similar) conditions.

Reduced units are popular with simulators, but not so popular with experimentalists, who like to see things expressed in SI units (although even that is not quite true, as is clear from the popularity of non-SI units, such as *kcal*'s and Ångstroms). To construct reduced units in simulations, we need a natural choice for the unit of length in the model, and for the units of energy and mass. These choices are not unique, but for pair potentials of the form $u(r) = \epsilon f(r/\sigma)$ (see Eq. (3.3.4)), it is logical to choose σ as the unit of length and ϵ as the unit of energy.

The well-known Lennard-Jones potential is of the form $u(r) = \epsilon f(r/\sigma)$ (see Eq. (3.3.4)) and hence a natural (though not unique) choice for our basic units is the following:

- Unit of length, σ
- Unit of energy, ϵ
- Unit of mass, m (the mass of the atoms in the system)

and from these basic units, all other units follow. For instance, our unit of time is

$$\sigma \sqrt{m/\epsilon}$$

and the unit of temperature is

$$\epsilon/k_B.$$

In terms of these reduced units, denoted with superscript *, the reduced pair potential $u^* \equiv u/\epsilon$ is a dimensionless function of the reduced distance $r^* \equiv r/\sigma$. For instance, the reduced form for the Lennard-Jones potential is

$$u_{\text{LJ}}^*(r^*) = 4 \left[\left(\frac{1}{r^*} \right)^{12} - \left(\frac{1}{r^*} \right)^6 \right]. \quad (3.3.11)$$

With these conventions, we can define the following reduced units: the potential energy $U^* = U\epsilon^{-1}$, the pressure $P^* = P\sigma^3\epsilon^{-1}$, the density $\rho^* = \rho\sigma^3$, and the temperature $T^* = k_B T \epsilon^{-1}$. For the simple potential given in Eq. (3.3.10) [81], the reduced form would be

$$u_{\text{WF}}^*(r^*) = \left[\left(\frac{1}{r^*} \right)^2 - 1 \right] \left[\left(\frac{2}{r^*} \right)^2 - 1 \right]^2.$$

TABLE 3.1 Translation of reduced units to real units for Lennard-Jones argon ($\epsilon/k_B = 119.8$ K, $\sigma = 3.405 \times 10^{-10}$ m, $M = 0.03994$ kg/mol).

Quantity	Reduced units		Real units
temperature	$T^* = 1$	\leftrightarrow	$T = 119.8$ K
density	$\rho^* = 1.0$	\leftrightarrow	$\rho = 1680$ kg/m ³
time	$\Delta t^* = 0.005$	\leftrightarrow	$\Delta t = 1.09 \times 10^{-14}$ s
pressure	$P^* = 1$	\leftrightarrow	$P = 41.9$ MPa

Note that the choice of units is not unique: for the LJ potential, we could equally well have chosen as our unit of length the position of the minimum $r_m = \sigma \times 2^{1/6}$. In that case, the reduced LJ potential would be of the form:

$$u_{\text{LJ}}^{**}(r^{**}) = \left[\left(\frac{1}{r^{**}} \right)^{12} - 2 \left(\frac{1}{r^{**}} \right)^6 \right],$$

where we have used the subscript ** to indicate that this scaled potential was obtained by scaling to r_m , rather than to σ . Of course, for the results of the simulations, the choice of the length scale is unimportant.

Another practical reason for using reduced units is the following: when we work with real (SI) units, we find that the absolute numerical values of the quantities that we are computing (e.g., the average energy of a particle or its acceleration) are either much less or much larger than one. If we multiply several such quantities using standard floating-point multiplication, we face the risk that, at some stage, we will obtain a result that creates an overflow or underflow. Conversely, in reduced units, almost all quantities of interest are of order 1 (say, between 10^{-3} and 10^3). Hence, if we suddenly find a very large (or very small) number in our simulations (say, 10^{42}), then there is a good chance that we have either found something interesting, or we have made a mistake somewhere (usually the second). In other words, reduced units make it easier to spot errors.

Simulation results that are obtained in reduced units can always be translated back into real units. For instance, if we wish to compare the results of a simulation on a Lennard-Jones model at $T^* = 1$ and $P^* = 1$ with experimental data for argon ($\epsilon/k_B = 119.8$ K, $\sigma = 3.405 \times 10^{-10}$ m, $M = 0.03994$ kg/mol), then we can use the translation given in Table 3.1 to convert our simulation parameters into SI units.¹²

Example 1 (Equation of state of the Lennard-Jones fluid-I). One important application of molecular simulation is computing the phase diagram

¹² In what follows we will use reduced units unless explicitly indicated otherwise. For this reason, we will henceforth omit the superscript * to denote reduced units.

of a given model system. In several chapters, we discuss some of the techniques that have been developed to study phase transitions, including direct-coexistence simulations. However, direct-coexistence simulations may suffer from hysteresis, which may make them less suited for locating phase transitions. In the present Example, we illustrate some of the problems that occur when we use standard Monte Carlo simulation to determine a phase diagram. As an example, we focus on the vapor-liquid curve of the Lennard-Jones fluid. Of course, as was already mentioned in section 3.3.2.2, the phase behavior is quite sensitive to the detailed form of the intermolecular potential that is used. In this Example, we approximate the full Lennard-Jones potential as follows:

$$u(r) = \begin{cases} u_{\text{LJ}}(r) & r \leq r_c \\ 0 & r > r_c, \end{cases}$$

where the cutoff radius r_c is set to half the box length. The contribution of the particles beyond this cutoff is estimated with the usual tail corrections; that is, for the energy

$$u^{\text{tail}} = \frac{8}{3}\pi\rho \left[\frac{1}{3} \left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right]$$

and for the pressure

$$P^{\text{tail}} = \frac{16}{3}\pi\rho^2 \left[\frac{2}{3} \left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right].$$

The equation of state of the Lennard-Jones fluid has been investigated by many groups using Molecular Dynamics or Monte Carlo simulations, starting with the work of Wood and Parker [69]. A first systematic study of the equation of state of the Lennard-Jones fluid was reported by Verlet [14]. Subsequently, many more studies have been published. In 1979, the data available at that time were compiled by Nicolas et al. [82] into an accurate equation of state. This equation has been refitted by Johnson et al. [83] using the best data then available. Since the work of Johnson et al., many additional studies on the equation of state of the Lennard-Jones fluid have appeared. This resulted in further improvements in the equation of state. There are now so many equations of state for the Lennard-Jones fluid that we have to refer to the review of Stephan et al. [84] for a discussion which of these equations to choose for which property. In the present study, we compare our numerical results with the equation of state by Johnson et al., which is sufficiently accurate for this purpose.

We performed several simulations using Algorithms 1 and 2. During the simulations, we determined the energy per particle and the pressure. The pressure was calculated using the virial, \mathcal{W}

$$P = \frac{\rho}{\beta} + \frac{W}{V}, \quad (3.3.12)$$

where the virial is defined by

$$W = \frac{1}{3} \sum_i \sum_{j>i} \mathbf{f}(\mathbf{r}_{ij}) \cdot \mathbf{r}_{ij}, \quad (3.3.13)$$

where $\mathbf{f}(\mathbf{r}_{ij})$ is the intermolecular force. Fig. 3.4 (left) compares the pressure as obtained from a simulation above the critical temperature with the equation of state of Johnson et al. [83]. The agreement is excellent (as is to be expected).

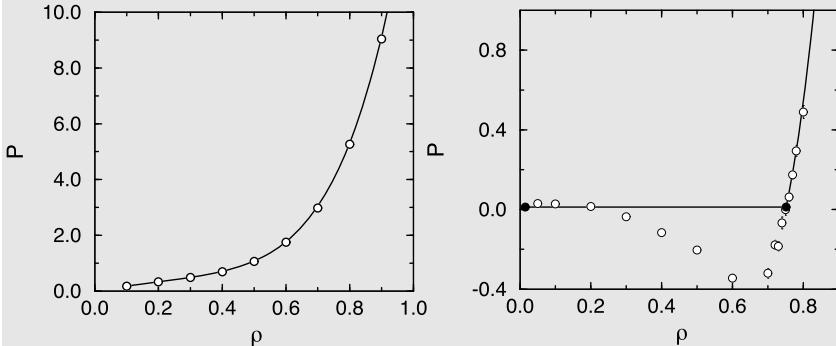


FIGURE 3.4 Equation of state of the Lennard-Jones fluid. (Left) Isotherm at $T = 2.0$. (Right) Isotherm below the critical temperature ($T = 0.9$); the horizontal line is the saturated vapor pressure, and the filled circles indicate the densities of the coexisting vapor and liquid phases. The solid curve represents the equation of state of Johnson et al. [83], and the circles are the results of the simulations ($N = 500$). The horizontal line in the right-hand figure corresponds to the coexistence pressure obtained by applying the Maxwell construction to the Johnson equation of state. The Maxwell construction exploits the fact that the equality of the chemical potentials of the coexisting phases implies that $P_{\text{coex}} = \int_{V_{\text{liq}}}^{V_{\text{vap}}} dV P(V)/(V_{\text{vap}} - V_{\text{liq}})$. The statistical errors in the numerical data are smaller than the symbol sizes.

Fig. 3.4 (right) shows a typical isotherm below the critical temperature. If we cool the system below the critical temperature, we should expect to observe vapor-liquid coexistence. However, standard Monte Carlo or Molecular Dynamics simulations of small model systems are not suited to study the coexistence between two phases. Using the Johnson equation of state, we predict how the pressure of a macroscopic Lennard-Jones system would behave in the two-phase region (see Fig. 3.4). For densities inside the coexistence region, the pressure is expected to be constant and equal to the saturated vapor pressure. If we now perform a Monte Carlo simulation of a finite system (500 LJ particles), we find that the computed pressure is not at all constant in the coexistence region (see Fig. 3.4). In fact we observe that, over a wide density range, the simulated system is metastable and may even have a negative pressure. The reason is that, in a finite system, a relatively important free-energy cost is associated with the creation of a liquid-vapor interface. So much so that, for sufficiently small systems, it is favorable for the system not to phase separate at all [85]. Clearly, these problems will be most severe for

small systems and in cases where the interfacial free energy is large. For this reason, standard NVT -simulations are not recommended to determine the vapor-liquid coexistence curve or, for that matter, any strong first-order phase transition in small systems.

To determine the liquid-vapor coexistence curve, we should determine the equation of state for a large number of state points outside the coexistence region. These data can then be fitted to an analytical equation of state. With this equation of state, we can determine the vapor-liquid curve (this is exactly the procedure used by Nicolas et al. [82] and Johnson et al. [83]).

Of course, if we simulate a system consisting of a sufficiently large number of particles, it is possible to simulate a liquid phase in coexistence with its vapor. However, such simulations are time-consuming because it takes a long time to equilibrate a two-phase system.

The equation of state in Fig. 3.4 is for the full Lennard-Jones potential (as approximated with tail corrections). However, differences in truncating the potential result in different equations of state for the Lennard-Jones fluid. For example, Thon et al. have developed an equation of state for the truncated and shifted Lennard-Jones fluid [86], which is typically used in a molecular dynamics simulation.

For more details, see SI (Case Study 1).

3.3.3 Detailed balance versus balance

There is no obvious bound to the number of valid **Markov-Chain Monte Carlo** algorithms, and in this book, we shall encounter a tiny (but hopefully important) fraction of all the algorithms that have been proposed. Clearly, it is important to be able to show that a proposed algorithm will lead to a correct sampling of the equilibrium properties of a system. There are two main factors that may contribute to incorrect sampling: the first is that the algorithm is not ergodic, i.e., that does not connect all equilibrium configurations that have non-vanishing Boltzmann weight.¹³ Non-ergodicity is arguably a larger problem for MD than for MC. It is certainly much harder to prove ergodic-like properties for an MD algorithm than for a properly designed MC algorithm. The second factor that leads to incorrect sampling arises when the states that are visited are not sampled with the correct weight. This is not a problem for MD algorithms that properly emulate Newton's equations of motion (we will state this more clearly in Chapter 4). With Monte Carlo algorithms, it is crucial to demonstrate that they generate the correct Boltzmann distribution of the states sampled.

The most common way to ensure that MC algorithms achieve Boltzmann sampling is to impose the condition of detailed balance (see Eq. (3.2.7)). However, as mentioned the detailed-balance condition is sufficient, but not necessary.

¹³ Of course, MC algorithms are not limited to applications in Molecular Simulation. Hence, the term “Boltzmann weight” stands for any non-negative weight function that determines the distribution over states.

Manousiouthakis and Deem [87] have shown that the weaker “balance condition” is a necessary and sufficient requirement to ensure Boltzmann sampling of the accessible states. Dropping the detailed-balance constraint gives us the freedom to use a much wider range of Monte Carlo strategies. But this freedom comes at a cost: proving the validity of non-detailed-balance MC algorithms is not always easy. This is why non-detailed-balance MC algorithms were often avoided in the past. However, in recent years Krauth and collaborators have developed a class of powerful non-detailed-balance MC algorithms [88,89]. We will discuss some of these algorithms in Chapter 13.

Here, we just give a simple example of an MC algorithm that is valid yet does not satisfy detailed balance: In the simple Monte Carlo scheme shown in Algorithm 2 we selected a particle at random and gave it a random trial displacement. The *a priori* probability to select the *same* particle in the next trial move and attempt to move it back to its original position is the same as the probability of carrying out the forward move. Combining this procedure to generate trial moves with the Metropolis acceptance rule results in an algorithm that satisfies detailed balance.

An alternative scheme would be to attempt moving all particles sequentially, i.e., an attempt to move particle 1 is followed by an attempt to move particle 2, etc. In this sequential scheme, the probability that a single-particle move is followed by its reverse is zero. Hence, this scheme clearly violates detailed balance. However, Manousiouthakis and Deem showed (under rather weak conditions —see Ref. [87]) that such a sequential updating scheme does obey balance and does therefore result in correct MC sampling.

There is every reason to distrust detailed-balance-violating algorithms if it has not been demonstrated that they satisfy balance. This is true in particular for “composite” algorithms that combine different trial moves.

The detailed-balance condition is, therefore, an important guiding principle in developing novel Monte Carlo schemes: when writing a new code, it is advisable to start with a version that imposes detailed balance and use that to obtain some reference data. If, subsequently, it turns out that the performance of a working program can be improved considerably by using a “balance-only” algorithm, then it is worth implementing it, —but it is then crucial to prove that balance is satisfied.

Example 2 (Importance of detailed balance). Monte Carlo simulations aim to sample points in configuration space according to their Boltzmann weights. If the system is ergodic, imposing detailed-balance is sufficient, but not necessary to ensure Boltzmann sampling. In other words: some sampling schemes that violate detailed balance may nevertheless lead to correct Boltzmann sampling (see Section 13.4.4). However, one should avoid non-detailed-balance schemes, unless one can prove that they satisfy the more general *balance* condition. In the present Example, we show that systematic errors result when using a scheme that does not satisfy the *balance* condition.

Consider an ordinary N, V, T move; a new trial position is generated by giving a randomly selected particle, say i , a random displacement:

$$x_n(i) = x_o(i) + \Delta_x(\mathcal{R} - 0.5),$$

where Δ_x is twice the maximum displacement. We now make a seemingly small change in the algorithm and generate a new position using

$$x_n(i) = x_o(i) + \Delta_x(\mathcal{R} - 0.0) \quad \text{wrong!}$$

i.e., we give the particles only a *positive* displacement. With such a move detailed balance is violated, since the reverse move —putting the particle back at x_o — is not possible.^a

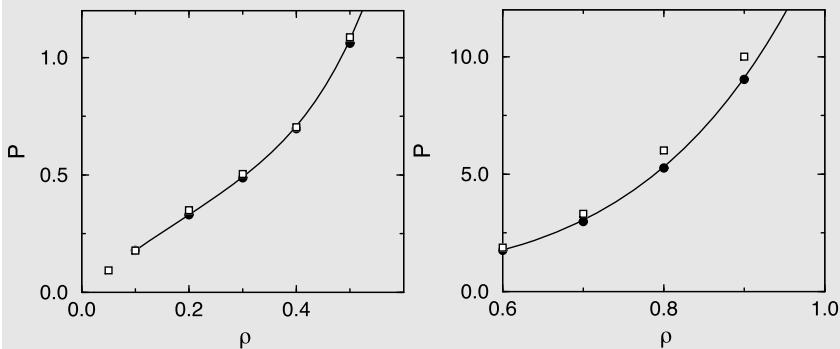


FIGURE 3.5 Equation of state of the Lennard-Jones fluid ($T = 2.0$); comparison of a displacement scheme that obeys detailed balance (circles) and one that does not (squares). Both simulations have been performed with 500 particles. The solid curve is the equation of state of Johnson et al. [83]. The figure at the left corresponds to the low-pressure regime. The high-pressure regime is shown in the right-hand figure.

For the Lennard-Jones fluid, we can use the program of Case Study 1 to compare the two sampling schemes. The results of these simulations are shown in Fig. 3.5. At first sight, the results of the incorrect scheme look reasonable; in fact, at low densities, the results of the two schemes do not show significant differences. But at high densities, the incorrect scheme overestimates the pressure. This overestimate of the pressure is a systematic error: it does not disappear when we perform longer simulations.

The above example illustrates something important: one cannot decide if an algorithm is correct on the basis of the fact that the results *look* reasonable. Simulation results must always be tested on cases where we *know* the correct answer: either a known (and trusted) numerical result, but preferably an exact result that may be known in some limiting cases (dilute vapor, dense solid, etc.).

Usually, one does not know *a priori* the size of the optimal maximum displacement in a Monte Carlo simulation. By doing short test runs, we can explore what maximum displacement achieves the best sampling for a given

amount of simulation time. However, the maximum step-size should not be changed during a production run, because then one would violate detailed balance [90]: if from one move to the next, the maximum displacement is decreased, then the *a priori* probability for a particle to return to its previous position could be zero, which violates microscopic reversibility.

For more details, see SI (Case Study 2).

^a In Event-Chain MC (Section 13.4.4) we will see examples of forward-only algorithms that have been designed to satisfy balance.

3.4 Trial moves

Having discussed specified the general structure of the Metropolis MC algorithm, we now consider its implementation. To carry out a simulation, we need a model for the system of interest and an algorithm to sample its properties. At this stage, we assume that we have a model (i.e., a prescription to compute all intermolecular interactions, given the particle coordinates) and that we have prepared our model system in a suitable (not too pathological) starting configuration. We must now we must decide how we are going to generate trial moves. That is: we must specify the matrix α introduced in Eq. (3.2.8). We must distinguish between trial moves that involve only the molecular centers of mass and those that change the orientation or even the conformation of a molecule. Later, we shall see that a great advantage of the Monte Carlo method is that we can carry out “unphysical” trial moves that have no counterpart in the natural dynamics of a system. Often such unphysical trial moves can enhance the sampling efficiency of an MC algorithm dramatically. In the present chapter, we only briefly refer to such unphysical moves (see Example 4). Still, in later chapters, we shall encounter numerous illustrations of the power of unphysical MC moves.

3.4.1 Translational moves

We first consider trial moves of the molecular centers of mass. A perfectly acceptable method for creating a trial displacement is to add random displacements between $-\Delta/2$ and $+\Delta/2$ to the x , y , and z coordinates of the molecular center of mass:

$$\begin{aligned}x'_i &\rightarrow x_i + \Delta (\mathcal{R} - 0.5) \\y'_i &\rightarrow y_i + \Delta (\mathcal{R} - 0.5) \\z'_i &\rightarrow z_i + \Delta (\mathcal{R} - 0.5),\end{aligned}\tag{3.4.1}$$

where $\Delta/2$ is the maximum size of the displacement and, as before, \mathcal{R} denotes a random number uniformly distributed between 0 and 1. Clearly, the reverse trial

move is equally probable (hence, α is symmetric).¹⁴ We are now faced with two questions: how large should we choose Δ , and should we attempt to move all particles simultaneously or one at a time? In the latter case, we should pick the molecule that is to be moved at random to ensure that the underlying Markov chain remains symmetric. All other things being equal, we should choose the most efficient sampling procedure. But, to this end, we must first define what we mean by *efficient sampling*. In rather vague terms, sampling is efficient if it gives you good value for money. Good value in a simulation corresponds to high statistical accuracy, and “money” is simply *money*: the money that buys your computer time and even your own time. For the sake of argument, we assume that the average scientific programmer is poorly paid. In that case, we only have to worry about your computer budget and the carbon footprint of the simulation.¹⁵ Then we could use the following definition of an optimal sampling scheme: a Monte Carlo sampling scheme can be considered optimal if it yields the lowest statistical error in the quantity to be computed for a given expenditure of computing budget. Usually, computing budget is translated into CPU time.

From this definition, it is clear that, in principle, a sampling scheme may be optimal for one quantity but not for another. Actually, the preceding definition is all but useless in practice (as are most definitions). For instance, it is just not worth the effort to measure the error estimate in the pressure for a number of different Monte Carlo sampling schemes in a series of runs of fixed length. However, it is reasonable to assume that the mean-square error in the observables is inversely proportional to the number of uncorrelated configurations visited in a given amount of CPU time. And the number of independent configurations visited is a measure of the distance covered in configuration space. This suggests a more manageable, albeit rather *ad hoc* criterion for estimating the efficiency of a Monte Carlo sampling scheme: the sum of the squares of all accepted trial displacements divided by computing time. This quantity should be distinguished from the mean-squared displacement per unit of computing time, because the latter quantity goes to 0 in the absence of diffusion (e.g., in a solid or glass), whereas the former does not.

Using this approximate measure of efficiency, we can offer a tentative explanation why, in simulations of condensed phases, it is usually better to perform random displacements of one particle at a time (as we shall see later, the situation is different for correlated displacements). To see why random single-particle moves are preferred, consider a system of N spherical particles,

¹⁴ Although almost all published MC simulations on atomic and molecular systems generate trial displacements in a cube centered around the original center of mass position, this is by no means the only possibility. Sometimes, it is more convenient to generate trial moves in a spherical volume, and it is not even necessary that the distribution of trial moves in such a volume be uniform, as long as it has inversion symmetry. For an example of a case where another sampling scheme is preferable, see ref. [91].

¹⁵ Still, we should stress that it is not worthwhile to spend a lot of time developing a fancy computational scheme that will be only marginally better than existing, simpler schemes, unless your program will run very often and a few percent gains in speed is important.

interacting through a potential energy function $\mathcal{U}(\mathbf{r}^N)$. Typically, we expect that a trial move will be rejected if the potential energy of the system changes by much more than $k_B T$. Yet we try to make the Monte Carlo trial steps as large as possible without having a very low acceptance. A displacement that would, on average, give rise to an increase of the potential energy by $k_B T$ would still have a reasonable acceptance. In the case of a single-particle trial move, we then have

$$\begin{aligned}\langle \Delta\mathcal{U} \rangle &= \left\langle \frac{\partial\mathcal{U}}{\partial r_i^\alpha} \right\rangle \overline{\Delta r_i^\alpha} + \frac{1}{2} \left\langle \frac{\partial^2\mathcal{U}}{\partial r_i^\alpha \partial r_i^\beta} \right\rangle \overline{\Delta r_i^\alpha \Delta r_i^\beta} + \dots \\ &= 0 + f(\mathcal{U}) \overline{\Delta r_i^2} + \mathcal{O}(\Delta^4),\end{aligned}\quad (3.4.2)$$

where the angle brackets denote averaging over the ensemble and the horizontal bar denotes averaging over random trial moves. The second derivative of \mathcal{U} has been absorbed into the function $f(\mathcal{U})$, the precise form of which does not concern us here. If we now equate $\langle \Delta\mathcal{U} \rangle$ on the left-hand side of Eq. (3.4.2) to $k_B T$, we find the following expression for $\overline{\Delta r_i^2}$:

$$\overline{\Delta r_i^2} \approx k_B T / f(\mathcal{U}). \quad (3.4.3)$$

If we attempt to move N particles, one at a time, most of the computation involved is spent on the evaluation of the change in potential energy. Assuming that we use a neighbor list or a similar time-saving device (see Appendix I), the total time spent on evaluating the potential energy change is proportional to nN , where n is the average number of interaction partners per molecule. The sum of the mean-squared displacements will be proportional to $N \overline{\Delta r^2} \sim N k_B T / f(\mathcal{U})$. Hence, the mean-squared displacement per unit of CPU time will be proportional to $k_B T / (n f(\mathcal{U}))$. Now suppose that we try to move all particles at once. The cost in CPU time will still be proportional to nN . But, using the same reasoning as in Eqs. (3.4.2) and (3.4.3), we estimate that the sum of the mean-squared displacements is smaller by a factor $1/N$. Hence the total efficiency will be down by this same factor. This simple argument explains why most simulators use single-particle, rather than collective trial moves. It is important to note that we have assumed that a collective MC trial move consists of N independent trial displacements of the particles. As will be discussed in section 13.3.1, efficient collective MC moves *can* be constructed if the trial displacements of the individual particles are not chosen independently.

Optimal acceptance of trial moves

Next, consider the choice of the parameter Δ which determines the size of the trial move. How large should Δ be? If it is very large, it is likely that the resulting configuration will have a high energy and the trial move will probably be rejected. If it is very small, the change in potential energy is probably small

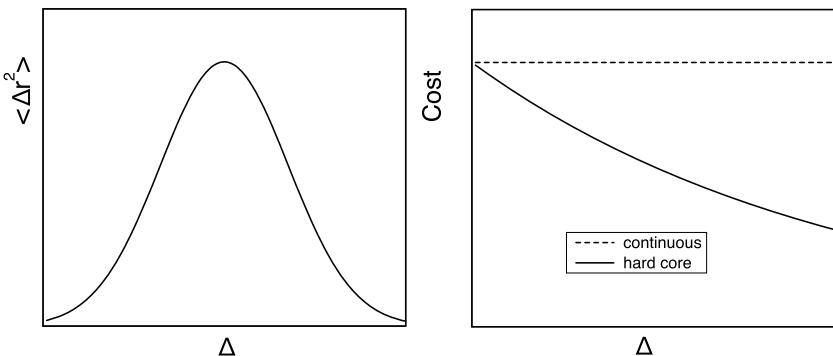


FIGURE 3.6 (Left) Typical dependence of the mean-squared displacement of a particle on the average size Δ of the trial move. (Right) Typical dependence of the computational cost of a trial move on the step-size Δ . For continuous potentials, the cost is constant, while for hard-core potentials it decreases rapidly with the size of the trial move.

and most moves will be accepted. In the literature, one often finds the mysterious statement that an acceptance of approximately 50% should be optimal. This statement has no justification. The optimum acceptance ratio is the one that leads to the most efficient sampling of configuration space. In fact, Roberts et al. [92,93] performed an analysis of Metropolis sampling in high-dimensional spaces for a special form of the n -dimensional distribution (not the Boltzmann form) and find that sampling is most efficient when the acceptance is 23.4%. In ref. [93] it is argued that this finding will probably generalize to more complex “non-pathological” distributions. Roberts et al., therefore, formulated the following rule of thumb: *Tune the proposal variance so that the average acceptance rate is roughly 1/4*. In our language, the “proposal variance” is determined by the size of the trial steps. So, all else being equal, it is probably preferable to aim for an acceptance of 25%, rather than 50%. But in certain MC simulations of many-body systems, the optimal acceptance may even be lower. This happens when the computational cost of a rejected move is less than that of an accepted move. To see why this might be so, let us consider the “diffusion coefficient” (i.e., the mean-squared displacement per unit CPU time) of the MC random walk in the (high-dimensional) configuration space. We will use this diffusion coefficient to compare the sampling efficiencies of different MC algorithms. It is then easy to see that different Monte Carlo codes will have different optimal acceptance ratios. The reason is that it makes a crucial difference whether the amount of computing required to test whether a trial move is accepted depends on the magnitude of the move (see Fig. 3.6). In the conventional Metropolis scheme, all continuous interactions have to be computed before a move can be accepted or rejected. Hence, for continuous potentials, the amount of computation does not depend on the size of a trial move. In contrast, for simulations of particles with hard repulsive cores, a move can be rejected as soon as an overlap with any neighbor is detected. In that case, a rejected move is cheaper than an ac-

cepted one, and hence the average computing time per trial move goes down as the step size is increased. As a result, the optimal acceptance ratio for hard-core systems is appreciably lower than for systems with continuous interactions. Exactly how much, depends on the nature of the program, on how the information about neighbor lists is stored, and even on the computational “cost” of random numbers and exponentiation. The consensus seems to be that for hard-core systems the optimum acceptance ratio is around, or even below, 20%. However, this is just another rule of thumb that should be checked.¹⁶

A distinct disadvantage of the efficiency criterion discussed previously is that it does not allow us to detect whether the sampling of configuration space is ergodic. To take a specific example, suppose that our system consists of a number of particles that are trapped in different potential energy minima. Clearly, we can sample the vicinity of these minima quite well and still have totally inadequate sampling of the whole of the configuration space. A criterion that would detect such non-ergodicity has been proposed by Mountain and Thirumalai [94]. These authors consider the difference between the variance of the time average of the single-particle (potential) energy. Let us denote the time average of the energy of particle j in time interval t by $\bar{e}_j(t)$:

$$\bar{e}_j(t) = \frac{1}{t} \int_0^t dt' e_j(t').$$

The average of all single-particle energies for this time interval is

$$\bar{e}(t) \equiv \frac{1}{N} \sum_{j=1}^N \bar{e}_j(t).$$

The variance of interest is

$$\sigma_E^2(t) \equiv \frac{1}{N} \sum_{j=1}^N [\bar{e}_j(t) - \bar{e}(t)]^2.$$

If all particles sample the whole configuration space, $\sigma_E^2(t)$ will approach zero as $t \rightarrow \infty$:

$$\sigma_E^2(t)/\sigma_E^2(0) \rightarrow \tau_E/t,$$

where τ_E is a measure for the characteristic time to obtain uncorrelated samples. However, if the system is non-ergodic, as in a (spin) glass, σ_E will not decay to 0. The work of Mountain and Thirumalai suggests that a good method for optimizing the efficiency of a Monte Carlo scheme is to minimize the product of τ_E

¹⁶ In section 13.4.3, Eq. (13.4.21), we show how, even in the case of continuous potentials, it is possible to reject trial moves before all interactions have been evaluated. With such a sampling scheme, the distinction between the sampling of hard-core and continuous potentials all but disappears. In recent years, this idea has gained more traction: see section 13.4.3.

and the computer time per trial move. Using this scheme, Mountain and Thirumalai already concluded that, even for the Lennard-Jones system, a trial move acceptance of 50% is far from optimal. They found that an acceptance probability of 20% was twice as efficient, which is in agreement with the observations made earlier in this section.

Of course, a scheme based on the energy fluctuations of a particle is not useful to monitor the rate of convergence of simulations of hard-core systems, or for that matter, of any system for which we cannot define the potential energy of a given particle (e.g., a system that cannot be described by a pair potential). But the essence of the method is not that one must necessarily probe single-particle energies: any quantity that is sensitive to the local environment of a particle should do. For instance, a robust criterion would look at the convergence of the time-averaged Voronoi signature of a particle. Different environments yield different signatures. Only if every particle samples all environments, will the average Voronoi signatures of all particles decay to the mean.

Of course, in some situations, an efficiency criterion based on ergodicity is not useful. By construction, it cannot be used to optimize simulations of glasses. But also when studying interfaces (e.g., solid-liquid or liquid-vapor), the ergodicity criterion would suggest that every particle should have ample time to explore both coexisting phases. This is clearly unnecessary: ice can be in equilibrium with water, even though the time of equilibration is far too short to allow the complete exchange of the molecules in the two phases.

Example 3 (Why count the old configuration again?). A somewhat counter-intuitive feature of the Metropolis sampling scheme is that, if a trial move is rejected, we should once again count the contributions of the old configuration to the average that we are computing (see acceptance rule (3.2.12)). The aim of this Example is to show that this recounting is really essential. In the Metropolis scheme, the acceptance rule for a move from o to n is

$$\begin{aligned} \text{acc}(o \rightarrow n) &= \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} & \mathcal{U}(n) \geq \mathcal{U}(o) \\ &= 1 & \mathcal{U}(n) < \mathcal{U}(o). \end{aligned}$$

These acceptance rules lead to a transition probability

$$\begin{aligned} \pi(o \rightarrow n) &= \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} & \mathcal{U}(n) \geq \mathcal{U}(o) \\ &= 1 & \mathcal{U}(n) < \mathcal{U}(o). \end{aligned}$$

Note that this transition probability must be normalized:

$$\sum_n \pi(o \rightarrow n) = 1.$$

From this normalization, it follows that the probability that we accept the old configuration again is by definition

$$\pi(o \rightarrow o) = 1 - \sum_{n \neq o} \pi(o \rightarrow n).$$

This last equation implies that we should count the contribution of the old configuration again.

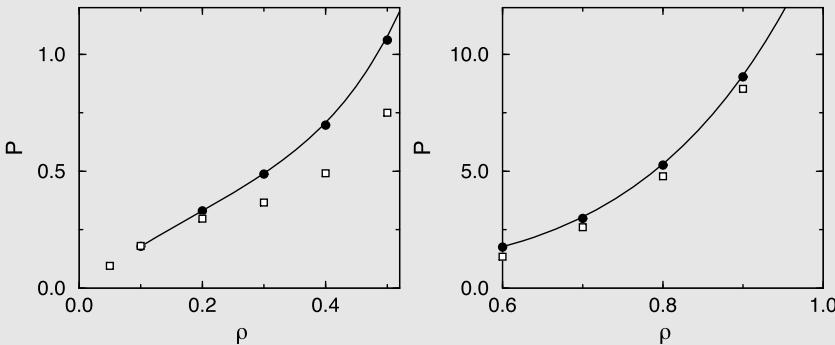


FIGURE 3.7 Equation of state of the Lennard-Jones fluid ($T = 2.0$); comparison of a scheme in which particles are displaced until a move is accepted (squares) with the conventional scheme (circles). Both simulations have been performed with 108 particles. The solid curve is the equation of state of Johnson et al. [83]. The left figure is at low pressure and the right one at high pressure.

It is instructive to use the Lennard-Jones program from Case Study 1 to investigate numerically the error that is made when we only include accepted configurations in our averaging. In essence, this means that in Algorithm 2 we continue attempting to displace the selected particle until a trial move has been accepted.^a In Fig. 3.7, we compare the results of the correct scheme with those obtained by the scheme in which we continue to displace a particle until a move is accepted. Again the results look reasonable, but the figure shows that large, systematic errors are being made.

For more details, see SI (Case Study 3).

^a It is easy to see that this approach leads to the wrong answer if we try to compute the average energy of a two-level system with energy levels E_0 and E_1 . If we include only accepted trial moves in our averaging, we would find that $\langle E \rangle = (E_0 + E_1)/2$, independent of temperature.

3.4.2 Orientational moves

When simulating molecules rather than atoms,¹⁷ we must use a repertoire of trial to change both the position and the orientation of the molecules. Rotational

¹⁷ Here, and in what follows, we use the term “molecule” to denote any type of particle, large or small, whose orientation can affect the potential energy of the system. Similarly, much of our discussion of “atomic” particles applies to any particle that is spherically symmetric: in this respect, a spherical colloid can be considered to be an “atom”.

moves are more subtle than translational moves: it almost requires an effort to generate translational trial moves with a distribution that does not satisfy the symmetry requirement of the underlying Markov chain. For rotational moves, the situation is different: it is only too easy to introduce a systematic bias in the orientational distribution function of the molecules by using a non-symmetrical orientational sampling scheme. Several different strategies to generate rotational displacements are discussed in [21]. Here we only mention one approach, for the sake of illustration.

3.4.2.1 Rigid, linear molecules

Consider a system consisting of N linear molecules. We specify the orientation of the i -th molecule by a unit vector $\hat{\mathbf{u}}_i$. One possible procedure to change $\hat{\mathbf{u}}_i$ by a small, random amount is the following. First, we generate a unit vector $\hat{\mathbf{v}}$ with a random orientation. This is quite easy to achieve (see Algorithm 38). Next, we multiply this random unit vector $\hat{\mathbf{v}}$ by a scale factor γ . The magnitude of γ determines the magnitude of the trial rotation. We now add $\gamma\hat{\mathbf{v}}$ to $\hat{\mathbf{u}}_i$. Let us denote the resulting sum vector by \mathbf{t} : $\mathbf{t} = \gamma\hat{\mathbf{v}} + \hat{\mathbf{u}}_i$. Note that \mathbf{t} is not a unit vector. Finally, we normalize \mathbf{t} , and the result is our trial orientation vector $\hat{\mathbf{u}}'_i$. We still have to fix γ , which determines the acceptance probability for the orientational trial move. The optimum value of γ is determined by essentially the same criteria as for translational moves. We have not yet indicated whether the translational and orientational trial moves should be performed simultaneously. Both procedures are acceptable. However, if rotation and translation correspond to separate moves, then the selection of the type of move should be probabilistic rather than deterministic if we wish to satisfy detailed balance.

3.4.2.2 Rigid, nonlinear molecules

The case of nonlinear, rigid molecules is slightly more complex than that of linear molecules. In molecular physics, it is conventional to describe the orientation of nonlinear molecules in terms of the Eulerian angles (ϕ, θ, ψ) . However, for most simulations, the use of these angles is less convenient because all rotation operations should then be expressed in terms of trigonometric functions, and these are computationally expensive. For Molecular Dynamics simulations, the situation is even worse: the equations of motion in terms of Euler angles have a singularity at $\theta = 0$. It is usually better to express the orientation of rigid, nonlinear molecules in terms of quaternion parameters (for a discussion of quaternions in the context of computer simulation, see [21]). The rotation of a rigid body can be specified by a quaternion of unit norm \mathbf{Q} . Such a quaternion may be thought of as a unit vector in four-dimensional space:

$$\mathbf{Q} \equiv (q_0, q_1, q_2, q_3) \quad \text{with } q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1. \quad (3.4.4)$$

There is a one-to-one correspondence between the quaternion components q_α and the Eulerian angles,

$$\begin{aligned} q_0 &= \cos \frac{\theta}{2} \cos \left(\frac{\phi + \psi}{2} \right) \\ q_1 &= \sin \frac{\theta}{2} \cos \left(\frac{\phi - \psi}{2} \right) \\ q_2 &= \sin \frac{\theta}{2} \sin \left(\frac{\phi - \psi}{2} \right) \\ q_3 &= \cos \frac{\theta}{2} \sin \left(\frac{\phi + \psi}{2} \right), \end{aligned} \quad (3.4.5)$$

and the rotation matrix R , which describes the rotation of the molecule-fixed vector in the laboratory frame is given by (see, e.g., [95])

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (3.4.6)$$

To generate trial rotations of nonlinear, rigid bodies, we must rotate the vector (q_0, q_1, q_2, q_3) on the four-dimensional ($4d$) unit sphere. The procedure just described for the rotation of a $3d$ unit vector is easily generalized to $4d$. An efficient method for generating random vectors uniformly on the $4d$ unit sphere has been suggested by Vesely [95].

3.4.2.3 Non-rigid molecules

If the molecules under consideration are not rigid, then we must also consider Monte Carlo trial moves that change the internal degrees of freedom of a molecule. In the present section, we discuss molecules that are not completely rigid but that have a single dominant conformation (or a small number of dominant conformations). Methods to perform Monte Carlo simulations on molecules with large numbers of distinct conformations are discussed in Chapter 12.

If a molecule is not completely rigid, the situation is relatively simple: we can carry out normal trial moves on the Cartesian coordinates of the individual atoms in the molecule (in addition to center-of-mass moves). If some of the atoms are strongly bound, it is advisable to carry out small trial moves on those particles (no rule forbids the use of trial moves of different sizes for different atoms, as long as the moves for one particular atom type are always sampled from the same distribution). But such “atomic” trial moves should be combined with trial moves that attempt to translate or rotate the molecule as a whole.

At first sight, then, sampling intra-molecular degrees of freedom would seem straightforward, in particular, if we choose suitable coordinates (e.g., bond

lengths or normal modes) to describe the structure of the molecules. However, if we use generalized coordinates (e.g., bond angles), we should bear in mind that the transformation from cartesian coordinates to generalized coordinates results in the appearance of Jacobians or, more precisely, the absolute value of the determinant of the Jacobian matrix, in the partition function (a simple case is when we transform from $\{x, y, z\}$ to $\{r, \theta, \phi\}$: the volume element $dx dy dz$ transforms into $r^2 \sin \theta dr d\theta d\phi$. The Jacobian $r^2 \sin \theta$ reflects the fact that uniform sampling in x, y, z does not correspond to uniform sampling in r, θ, ϕ). As we explain below, the appearance of Jacobians in the Monte Carlo weights may be a complication, but not an insurmountable one. The problem gets more tricky when we consider situations where certain coordinates are fixed (e.g. a bond length). It turns out that, in this case, it makes a difference if the constraint on the coordinates is imposed by considering the limit of a very stiff spring constant, or by eliminating the coordinate and its conjugate velocity altogether, by imposing the appropriate constraints on the Lagrangian (see Appendix A). The method of Lagrangian constraints is extensively used in MD simulations ((see Chapter 14) and ref. [96]), but the results of such simulations can only be compared with MC simulations if the constraints have been imposed in exactly the same way (see [97]). Different ways of imposing the same constraints can lead to observable differences. A well-known example is a freely jointed trimer with fixed bond lengths. As shown by Van Kampen, the limit of constraining the bond lengths with stiff springs yields a different distribution of the internal angle than when the same constraint would have been imposed at the level of the Lagrangian [98] (see section 14.1.1).

To understand why Monte Carlo simulations of flexible molecules with a number of stiff (or even rigid) bonds (or bond angles) can become complicated, let us return to (2.2.20) for a thermal average of a function $A(\mathbf{r}^N)$:

$$\langle A \rangle = \frac{\int d\mathbf{p}^N d\mathbf{r}^N A(\mathbf{r}^N) \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}{\int d\mathbf{p}^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}.$$

We will first consider the case that the constraints are just stiff. After that, we consider what happens if certain coordinates are rigorously constrained. To perform Monte Carlo sampling on the generalized coordinates \mathbf{q}^N , we must express the Hamiltonian in Eq. (2.2.20) in terms of these generalized coordinates and their conjugate momenta. This is done most conveniently by first considering the Lagrangian, $\mathcal{L} \equiv \mathcal{K} - \mathcal{U}$, where \mathcal{K} is the kinetic energy of the system ($\mathcal{K} = \sum_{i=1}^N (1/2)m_i \dot{r}_i^2$) and \mathcal{U} the potential energy. When we transform from Cartesian coordinates \mathbf{r} to generalized coordinates \mathbf{q} , \mathcal{L} changes to

$$\begin{aligned} \mathcal{L} &= \sum_{i=1}^N \frac{1}{2} m_i \frac{\partial \mathbf{r}_i}{\partial q_\alpha} \frac{\partial \mathbf{r}_i}{\partial q_\beta} \dot{q}_\alpha \dot{q}_\beta - \mathcal{U}(\mathbf{q}^N) \\ &\equiv \frac{1}{2} \dot{\mathbf{q}} \cdot \mathbf{G} \cdot \dot{\mathbf{q}} - \mathcal{U}(\mathbf{q}^N). \end{aligned} \quad (3.4.7)$$

In the second line of Eq. (3.4.7), we have defined the matrix G . The momenta conjugate to \mathbf{q}^N are easily derived using

$$p^\alpha \equiv \frac{\partial \mathcal{L}}{\partial \dot{q}_\alpha}.$$

This yields $p^\alpha = G_{\alpha\beta}\dot{q}_\beta$. We can now write down the Hamiltonian \mathcal{H} in terms of the generalized coordinates and conjugate momenta:

$$\mathcal{H}(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\mathbf{p} \cdot G^{-1} \cdot \mathbf{p} + \mathcal{U}(\mathbf{q}^N). \quad (3.4.8)$$

If we now insert this form of the Hamiltonian into Eq. (3.2.2), and carry out the (Gaussian) integration over the momenta, we find that

$$\begin{aligned} \langle A \rangle &= \frac{\int d\mathbf{q}^N \exp[-\beta\mathcal{U}(\mathbf{q}^N)] A(\mathbf{q}^N) \int d\mathbf{p}^N \exp(-\beta\mathbf{p} \cdot G^{-1} \cdot \mathbf{p}/2)}{\int d\mathbf{q}^N d\mathbf{p}^N \exp(-\beta\mathcal{H})} \\ &= \frac{\int d\mathbf{q}^N \exp[-\beta\mathcal{U}(\mathbf{q}^N)] A(\mathbf{q}^N) |G|^{\frac{1}{2}}}{\int d\mathbf{q}^N d\mathbf{p}^N \exp(-\beta\mathcal{H})}. \end{aligned} \quad (3.4.9)$$

The problem with Eq. (3.4.9) is the term $|G|^{\frac{1}{2}}$. Although the determinant $|G|$ can be computed fairly easily for small flexible molecules, its evaluation can become an unpleasant task in the case of larger molecules. Note that the factor $|G|^{\frac{1}{2}}$ could be written as $\exp(\ln|G|^{\frac{1}{2}})$: it, therefore, appears as a correction to the Boltzmann factor.

Next, let us consider the case that a subset $\{\sigma\}$ of the generalized coordinates are constrained to have a fixed value, which necessarily also implies that $\dot{\sigma} = 0$. These hard constraints must be imposed at the level of the Lagrangian and lead to a reduction in the number of degrees of freedom of the system. They also lead to a different form for the Hamiltonian in Eq. (3.4.8) and to another determinant in Eq. (3.4.9). Again, all this can be taken into account in the Monte Carlo sampling (see [97]). An example of such a Monte Carlo scheme is the concerted rotation algorithm that has been developed by Theodorou and co-workers [99] to simulate polymer melts and glasses (see SI L.8.2). The idea of this algorithm is to select a set of adjacent skeletal bonds in a chain (up to seven bonds). These bonds are given a collective rotation while the rest of the chain is unaffected. By comparison, Molecular Dynamics simulations of flexible molecules with hard constraints have the advantage that these constraints enter directly into the equations of motion (see [96]). In Chapter 12, we shall discuss other Monte Carlo sampling schemes that are particularly suited for flexible molecules. Still, these schemes do not eliminate the problem associated with the introduction of hard constraints.

Example 4 (Mixture of hard disks). One of the important disadvantages of the Monte Carlo scheme is that it does not reproduce the natural dynamics of the particles in the system. However, sometimes this limitation of the method can be made to work to our advantage. Below we show how the equilibration of a Monte Carlo simulation can be sped up by many orders of magnitude through the use of unphysical trial moves.

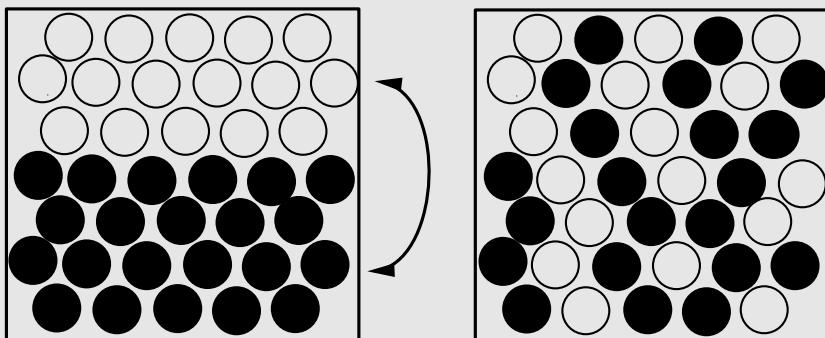


FIGURE 3.8 A mixture of hard disks, where the identities of two particles are swapped.

In a Molecular Dynamics simulation of, for instance, a binary ($A - B$) mixture of hard disks (see Fig. 3.8), the efficiency with which configuration-space is sampled is greatly reduced by the fact that concentration fluctuations decay very slowly: typically the relaxation time $\tau \sim D_{AB}/\lambda^2$, where D_{AB} is the mutual diffusion coefficient, and λ is the wavelength of the concentration fluctuation. As a consequence, very long runs would be needed to ensure equilibration of the local composition of the mixture. In solids, equilibration may not take place at all on simulation, or even experimental time scales. In contrast, in a Monte Carlo simulation, it is permissible to carry out trial moves that swap the identities of two particles of species A and B . Such moves, even if they have only a moderate rate of acceptance (a few percent will do), greatly speed up the sampling of concentration fluctuations in crystalline solids [100,101] and polydisperse glasses [102].

3.5 Questions and exercises

Question 10 (Reduced units). Typical parameters for the Lennard-Jones potentials of argon and krypton, truncated and shifted at $r_c = 2.5\sigma$, are $\sigma_{Ar} = 3.405 \text{ \AA}$, $\epsilon_{Ar}/k_B = 119.8 \text{ K}$ and $\sigma_{Kr} = 3.63 \text{ \AA}$, $\epsilon_{Kr}/k_B = 163.1 \text{ K}$ [103].

1. At the reduced temperature $T^* = 2.0$, what is the temperature in kelvin of argon and krypton?
2. A typical time step for MD is $\Delta t^* = 0.001$. What is this in SI units for argon and krypton?

- If we simulate argon at $T = 278\text{ K}$ and density $\rho = 2000\text{ kg m}^{-3}$ with a Lennard-Jones potential, for which conditions of krypton can we use the same data? If we assume ideal gas behavior, compute the pressure in reduced and normal units.
 - List the main reasons to use reduced units.

- 4.** List the main reasons to use reduced units.

The WF potential [8J] is a simple, Lennard-Jones-like potential that goes quadratically to zero at the cut-off radius r_c . Its functional form is particularly simple for a cut-off distance $r_c = 2\sigma$ (see Eq. (3.3.10)):

$$\phi(r) \equiv \epsilon \left[\left(\frac{\sigma}{r} \right)^2 - 1 \right] \left[\left(\frac{r_c}{r} \right)^2 - 1 \right]^2 \quad \text{for } r \leq r_c \\ = 0 \quad \quad \quad \text{for } r > r_c.$$

In reduced units, the critical temperature $k_B T_c/\epsilon \approx 1.04$ and the reduced critical density $\rho_c \sigma^3 \approx 0.32$.

- Given that the critical temperature of Argon is $T_c = 83.1\text{ K}$, give $\epsilon^{(WF)}$ in SI units.
 - Given that the critical density of Argon is 536 kg m^{-3} , estimate $\sigma^{(WF)}$ for Argon (in SI units).

Question 11 (An alien potential). On the planet Krypton, the pair potential between two Gaia atoms is given by the Lennard-Jones 10-5 potential

$$U(r) = 5\epsilon \left[\left(\frac{\sigma}{r}\right)^{10} - \left(\frac{\sigma}{r}\right)^5 \right].$$

Kryptonians are notoriously lazy and it is therefore up to you to derive the tail corrections for the energy, pressure, and chemical potential. If we use this potential in an MD simulation in the truncated and shifted form we still have a discontinuity in the force. Why? If you compare this potential with the Lennard-Jones potential, will there be any difference in efficiency of the simulation? (Hint: there are two effects!)

Exercise 6 (Calculation of π). Consider a circle of diameter d surrounded by a square of length l ($l \geq d$). Random coordinates are generated uniformly within the square. The value of π can be calculated from the fraction of points that fall within the circle.

1. How can π be calculated from the fraction of points that fall in the circle?
 2. Complete the small Monte Carlo program to calculate π using this method. Compare your result with the “exact” value of π : in some languages, π is pre-computed, in others it can easily be computed e.g., by using $\pi = 4.0 \times \arctan(1.0)$.
 3. How does the accuracy of the result depend on the ratio l/d and the number of generated coordinates? Derive a formula to calculate the relative standard deviation of the estimate of π .

4. Why is this not an efficient method for computing π accurately?

Exercise 7 (The photon gas). The average occupancy number of state j of the photon gas, $\langle n_j \rangle$, can be calculated analytically; see Eq. (2.6.5). It is possible to estimate this quantity using a Monte Carlo scheme. In this exercise, we will use the following procedure to calculate $\langle n_j \rangle$:

- (i) Start with an arbitrary n_j .
- (ii) Decide at random to perform a trial move to increase or decrease n_j by 1.
- (iii) Accept the trial move with probability

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp[-\beta(U(\text{n}) - U(\text{o}))]).$$

Of course, n_j cannot become negative!

1. Does this scheme obey detailed balance when $n_j = 0$?
2. Is the algorithm still correct when trial moves are performed that change n_j with a random integer from the interval $[-5, 5]$? What happens when only trial moves are performed that change n_j with either -3 or $+3$?
3. Assume that $N = 1$ and $\epsilon_j = \epsilon$. Write a small Monte Carlo program to calculate $\langle n_j \rangle$ as a function of $\beta\epsilon$. Compare your result with the analytical solution.
4. Modify the program in such a way that the averages are not updated when a trial move is rejected. Why does this lead to erroneous results? At which values of β does this error become more pronounced?
5. Modify the program in such a way that the distribution of n_j is calculated as well. Compare this distribution with the analytical expression.

Exercise 8 (Monte Carlo simulation of a Lennard-Jones system). In this exercise, we study a 3d Lennard-Jones system (see also online Case Study 1). Chapter 5 provides the details on how the different observables are computed in a simulation.

1. In the code that you can find on the book's website, the pressure of the system is not calculated. Modify the code in such a way that the average pressure can be calculated. You will only have to make some changes in the function that computes the energy.
2. Perform a simulation at $T = 2.0$ and at various densities. Up to what density is the ideal gas law

$$\beta p = \rho \tag{3.5.1}$$

a good approximation?

3. The program produces a sequence of snapshots of the state of the system. Try to visualize these snapshots using, for example, the program [Visual Molecular Dynamics \(VMD\)](#) [104].
4. For the heat capacity at constant volume, one can derive

$$C_v = \frac{\langle U^2 \rangle - \langle U \rangle^2}{k_B T^2}$$

in which U is the total energy of the system. Derive a formula for the dimensionless heat capacity. Modify the program that can be found on the book's website in such a way that C_v is calculated.

5. Instead of performing trial moves in which one particle at a time is displaced, one can make trial moves in which all particles are displaced. Compare the maximum displacements of these moves when 50% of all displacements are accepted.
6. Instead of using a uniformly distributed displacement, one can also use a Gaussian displacement. Does this increase the efficiency of the simulation?

Exercise 9 (Scaling as a Monte Carlo move). Consider a system in which the energy is a function of one variable (x) only,

$$\exp[-\beta U(x)] = \theta(x) \theta(1-x),$$

in which $\theta(x)$ is the Heaviside step function: $\theta(x < 0) = 0$ and $\theta(x > 0) = 1$. We wish to calculate the distribution of x in the canonical ensemble. We will consider two possible algorithms (we will use $\delta > 0$):

- (i) Generate a random change in x between $[-\delta, \delta]$. Accept or reject the new x according to its energy.
 - (ii) Generate a random number ϕ between $[1, 1 + \delta]$. With a probability of 0.5, invert the value ϕ thus obtained. The new value of x is obtained by multiplying x with ϕ .
1. Derive the correct acceptance/rejection rules for both schemes.
 2. Complete the computer code to calculate the probability density of x .
 3. What happens when the acceptance rule of method (i) is used in the algorithm of method (ii)?

Chapter 4

Molecular Dynamics simulations

Molecular Dynamics (MD) simulation is a technique for computing the equilibrium and transport properties of a classical many-body system. In this context, the word *classical* means that the nuclear motion of the constituent particles obeys the laws of classical mechanics, which is an excellent approximation for the translational and rotational motion of a wide range of molecules. However, quantum effects cannot be ignored when considering the translational or rotational motion of light atoms or molecules (He , H_2 , D_2) or vibrational motions with a frequency ν such that $h\nu > k_B T$.

Molecular Dynamics simulations were pioneered by Alder and Wainwright in the mid-1950s [18], and have since become a standard tool in many areas of the physical, biological, and engineering sciences. Literally, dozens of books have been written about the method. In addition, many books describe applications of MD simulations to specific subject areas, or the use of MD simulations in the context of specific program packages.

In what follows, we do not aim to cover all aspects of MD simulations. Rather, as in the rest of this book, we focus on the basics of the method: what it is, what it can do, and, importantly, what it cannot do —and, at the simplest level, to give the reader an idea of how to do it. We urge the readers who are interested in other aspects of MD to consult some of the many excellent books on the topic [21–24,26,105]. In addition to books, there are many conference proceedings that provide snapshots of the development of the field [44–46,49].

4.1 Molecular Dynamics: the idea

Molecular Dynamics simulations are in many respects similar to real experiments. When performing a real experiment, we proceed as follows: We prepare a sample of the material that we wish to study. We connect this sample to a measuring instrument (e.g., a thermometer, pressure gauge, or viscometer), and we record the property of interest over a certain time interval. If our measurements are subject to statistical noise (as most measurements are), then the longer we average, the more accurate our measurement becomes. In a Molecular Dynamics simulation, we follow the same approach. First, we prepare a sample: we select a model system consisting of N particles and we solve Newton's equations of motion for this system until the properties of the system no longer change with time (i.e., we equilibrate the system). After equilibration, we perform the actual measurement. In fact, some of the most common mistakes that can be made

when performing a computer experiment are similar to the ones that can be made in real experiments: e.g., the sample has not been prepared correctly, the duration of the measurement has been too short, the system undergoes an irreversible change during the simulation, or we are not measuring what we think we are.

To measure an observable quantity in a Molecular Dynamics simulation, we must first of all be able to express this observable as a function of the positions and momenta of the particles in the system. For instance, a convenient definition of the temperature in a (classical) many-body system makes use of the equipartition of energy over all degrees of freedom that enter quadratically in the Hamiltonian of the system. In the thermodynamic limit, we have

$$\left\langle \frac{1}{2}mv_\alpha^2 \right\rangle = \frac{1}{2}k_B T. \quad (4.1.1)$$

However, this relation is not quite correct for a finite system. In particular, for a hypothetical atomic system with fixed total kinetic energy, we can define the kinetic temperature using the microcanonical relation

$$\frac{1}{k_B T} = \frac{\partial \ln \Omega(E_{\text{kin}})}{\partial E_{\text{kin}}}. \quad (4.1.2)$$

We then find that, for a d -dimensional system of N atoms with fixed total momentum, the instantaneous temperature $k_B T$ is equal to $2E_{\text{kin}}/(d(N - 1) - 1)$. Hence the number of degrees of freedom that enter into the equipartition relation is $N_f = dN - (d + 1)$. In an interacting system, the total kinetic energy fluctuates, and so does the instantaneous temperature:

$$T(t) = \sum_{i=1}^N \frac{m_i v_i^2(t)}{k_B N_f}. \quad (4.1.3)$$

As the variance in the total kinetic energy of an interacting system scales as $N \approx N_f$ [106] (see section 5.1.8), the relative fluctuations in the temperature will be of order $1/\sqrt{N_f}$. As N_f for many simulations is in the range between $10^3 - 10^6$, the statistical fluctuations in the temperature are typically of the order of $\mathcal{O}(1\%) - \mathcal{O}(0.1\%)$. To get more accurate estimates of the temperature, one should average over many fluctuations. Note that for systems that contain very few particles, the correct counting of the number of degrees of freedom becomes important [107].

4.2 Molecular Dynamics: a program

The best introduction to Molecular Dynamics simulations is to consider a simple program. The program that we consider is kept as simple as possible to illustrate a number of important features of Molecular Dynamics simulations.

The program is constructed as follows:

1. We specify the parameters that set the conditions of the run (e.g., initial temperature, number of particles, density, time step, etc.).
2. We initialize the system (i.e., we select initial positions and velocities).
3. We compute the forces on all particles.
4. We integrate Newton's equations of motion. This step and the previous one make up the core of an MD simulation. These two steps are repeated until we have computed the time evolution of the system for the desired length of time.
5. After completion of the central loop, we compute and store the averages of measured quantities, and stop.

Algorithm 3 is a pseudo-algorithm for a Molecular Dynamics simulation of an atomic system. Below, we discuss the different steps in the program in more detail.

4.2.1 Initialization

To start the simulation, we should assign initial positions and velocities to all particles in the system. In particular for simulations of crystalline systems, the particle positions should be chosen compatible with the crystal structure that we are aiming to simulate. This requirement puts constraints on the number of particles in the periodic box: e.g., for a cubic box, the number of particles in an FCC crystal should be $N = 4n^3$, where n is an integer, and for BCC, $N = 2n^3$. In any event, the particles should not be positioned at positions that result in an appreciable overlap of the atomic or molecular cores. In Algorithm 4 we do not specify the initial crystal structure explicitly. For the sake of the argument, let us assume that it is a simple cubic lattice, which is mechanically unstable. To simulate a liquid, we typically choose the density and the initial temperature such that the simple cubic lattice melts rapidly. First, we put each particle on its lattice site and, in Algorithm 4, we attribute to each velocity component $\alpha = \{x, y, z\}$ of every particle a value that is drawn from a Gaussian distribution, using the Box-Muller method [66]: $v_{i,\alpha} \sim \sqrt{-\ln(R_1)} \cos(2\pi R_2)$, where R_1, R_2 are random numbers uniformly distributed between 0 and 1. Subsequently, we shift all velocities, such that the total momentum is zero and we scale the resulting velocities to adjust the mean kinetic energy to the desired value. We know that, in thermal equilibrium, the following relation should hold for $N \gg 1$:

$$\langle v_\alpha^2 \rangle = k_B T / m, \quad (4.2.1)$$

where v_α is the α component of the velocity of a given particle. We can use this relation to define an instantaneous temperature at time t , $T(t)$:

$$k_B T(t) \equiv \sum_{i=1}^N \frac{mv_{\alpha,i}^2(t)}{N_f}. \quad (4.2.2)$$

Algorithm 3 (Core of Molecular Dynamics program)

program MD	basic MD code
[...]	
setlat	function to initialize positions x
initv (temp)	function to initialize velocities v_x
t=0	
while (t < tmax) do	main MD loop
FandE	function to compute forces and total energy
Integrate-V	function to integrate equations of motion
t=t+delt	update time
sample	function to sample averages
enddo	
end program	

Specific Comments (for general comments, see p. 7)

1. The [...] at the beginning refers to the initialization of variables and parameters used in the program. We assume that the maximum run time `tmax` and the time step `delt` are global variables. The initial temperature `temp` is explicitly needed in the function `initv`, and is therefore shown as an argument.
2. The function `setlat` creates a crystal lattice of `npart` particles in a given volume (see Algorithm 20). The number of particles `npart` is usually chosen compatible with the number of unit cells (`nx, ny, nz`), and with `nc`, the number of particles per unit cell: `npart=nx*ny*nz*nc`.
3. To simulate disordered systems, many simulation packages generate an initial structure that is already disordered. Such a function may speed up equilibration, but requires additional steps.
4. The function `Initv` (see Algorithm 4) initializes the velocities `vx` such that the initial temperature is `temp`. From these velocities, the positions one time-step earlier `xm`.
5. The main loop consists of three steps
 - a. `FandE` (see Algorithm 5) computes the current energy and forces
 - b. `Sample` samples the desired observables at time t - not necessarily every time step. See Algorithms 8 and 9 for some examples of sampling algorithms.
 - c. Integrate Newton's equation of motion, using the Verlet algorithm (`Integrate-V`—see Algorithm 6) and update the time t .

Clearly, we can adjust the instantaneous temperature $T(t)$ to match the desired temperature T by scaling all velocities with a factor $(T/T(t))^{1/2}$. This initial setting of the temperature is not particularly critical, as the temperature will change anyway during equilibration.

Algorithm 4 (Initialization of a Molecular Dynamics program)

<pre> function initv(<i>temp</i>) sumv=0 sumv2=0 for 1 ≤ <i>i</i> ≤ npart do <i>x</i>(<i>i</i>) = lattice_pos(<i>i</i>) <i>vx</i>(<i>i</i>) = $\sqrt{-\ln(\mathcal{R})} \cos(2\pi \mathcal{R})$ sumv=sumv+<i>v</i>(<i>i</i>) enddo sumv=sumv/npart for 1 ≤ <i>i</i> ≤ npart do <i>vx</i>(<i>i</i>) = <i>vx</i>(<i>i</i>) - sumv sumv2=sumv2+<i>vx</i>(<i>i</i>)**2 enddo <i>fs</i>=$\sqrt{\text{temp}/(\text{sumv2}/\text{nf})}$ for 1 ≤ <i>i</i> ≤ npart do <i>vx</i>(<i>i</i>)=<i>vx</i>(<i>i</i>)*<i>fs</i> <i>xm</i>(<i>i</i>)=<i>x</i>(<i>i</i>)-<i>vx</i>(<i>i</i>)*dt enddo end function</pre>	initializes velocities for MD program Place the particle on a lattice Generate 1D normal distribution center of mass momentum ($m = 1$) center of mass velocity set desired kinetic energy and set Center of Mass velocity to zero kinetic energy <i>temp</i> = desired initial temperature set initial kinetic temperature position previous time step
--	---

Specific Comments (for general comments, see p. 7)

- Every call of the random-number routine yields a different random number \mathcal{R} , uniformly distributed between 0 and 1.*
- Strictly speaking, we need not generate a Maxwell-Boltzmann distribution for the initial velocities: upon equilibration, the distribution will become a Maxwellian.*
- nf is the number of degrees of freedom. In d dimensions, energy and momentum conservation imply that $\text{nf}=d*(\text{npart}-1)-1$.*
- During equilibration, the kinetic energy will change. Therefore, the temperature of the equilibrated system will differ from *temp*.*

As will appear later, we do not really use the velocities themselves in our algorithm to solve Newton's equations of motion. Rather, we use the positions of all particles at the present (x) and previous (x_m) time steps, combined with our knowledge of the force (f) acting on the particles, to predict the positions at the next time step. When we start the simulation, we must bootstrap this procedure by generating approximate previous positions. Without much consideration for any law of mechanics other than the conservation of linear momentum, we approximate x for a particle in a direction by $x_m(i) = x(i) - v(i)*dt$. Of course, we could make a better estimate of the true previous position of each particle. But as we are only bootstrapping the simulation, we do not worry about such subtleties.

4.2.2 The force calculation

What comes next is the most time-consuming part of most Molecular Dynamics simulations: the calculation of the forces acting on the particles. If we consider a model-system with pairwise-additive interactions, as we do in the present case, we have to consider the contribution to the force on particle i due to all particles within its range of interaction.¹ If we only consider the interaction between a particle and the nearest image of another particle, this implies that, for a system of N particles, we must evaluate $N \times (N - 1)/2$ pair distances. If we would use no time-saving devices, the time needed for the evaluation of the forces would scale as N^2 . There exist efficient techniques to speed up the evaluation of both short-range and long-range forces in such a way that the computing time scales as N (for long-ranged forces $N \ln N$), rather than N^2 . In Appendix I, we describe some of the more common techniques to speed up the calculation of forces between particles. Although the examples in that Appendix apply to Monte Carlo simulations, similar techniques can be used in a Molecular Dynamics simulation. However, in the present, simple example (see Algorithm 5) we will not attempt to make the program particularly efficient and we shall, in fact, consider all possible pairs of particles explicitly.

We first compute the current (vectorial) distances in the x , y , and z directions between each pair of particles i and j . These distances are denoted by xr . As in the Monte Carlo case, we use periodic boundary conditions (see section 3.3.2.1). In the present example, we use a cutoff at a distance r_c in the explicit calculation of intermolecular interactions, where r_c should be chosen to be less than half the diameter of the periodic box to ensure that a given particle i interacts only with the nearest periodic image of any other particle j .

In the present case, the diameter of the periodic box is denoted by `box`. If we use simple cubic periodic boundary conditions, the distance in any direction between i and the nearest image of j should always be less (in absolute value) than $\text{box}/2$. A compact way to compute the distance between i and the nearest periodic image of j uses the nearest integer function. We now first evaluate xr , the difference between the current x -coordinates of i and any j . Note that these coordinates need not be inside the same periodic box. To obtain the x -distance between the nearest images of i and j , we transform xr to $\text{xr} = \text{xr} - \text{box} * \text{nint}(\text{xr}/\text{box})$. Once we have computed all Cartesian components of \mathbf{r}_{ij} , the vector distance between i and the nearest image of j , we compute r_{ij}^2 (denoted by `r2` in the program). Next we test if r_{ij}^2 is less than r_c^2 , the square of the cutoff radius. If $r_{ij}^2 > r_c^2$, j does not interact with i and we skip to the next value of j . Note that we do not compute $|r_{ij}|$ itself, because that would involve the evaluation of a square root, which (at least at this stage) would be unnecessary.

¹ Increasingly, MD simulations use non-pairwise additive interactions. Even if the functional form of the potential energy is known in principle, computing the forces may become painful. In such cases, Automatic Differentiation (see e.g., [108]) may be an attractive option.

Algorithm 5 (Calculation of pair forces and energy forces)

```

function FandE
    rc2=rc**2
    en=0
    for 1 ≤ i ≤ npart do
        fx(i)=0
    enddo
    for 1 ≤ i ≤ npart-1 do
        for i+1 ≤ j ≤ npart do
            xr=x(i)-x(j)
            xr=xr-box*round(xr/box)
            r2=xr*xr
            if r2 <rc2 then
                r2i=1/r2
                r2im1=r2i-1.0
                rc2r2im1=rc2*r2i-1.0
                en=en+r2im1*rc2r2im1**2
                ff=6.0*r2i**2*rc2r2im1
                *(rc2r2im1-2)
                fx(i)=fx(i)+ff*xr
                fx(j)=fx(j)-ff*xr
            endif
        enddo
    enddo
end function

```

Specific Comments (for general comments, see p. 7)

1. Although **FandE** shares arrays and parameters with the rest of the program, it does not require specific input parameters.
2. The diameter of the simulation box is denoted by **box**.
3. The function **round**, rounds a floating point number to the nearest integer.
4. In this algorithm, we use the **WF** potential [81], which is a simple LJ-like potential. In reduced units, the pair energy:

$$en = \left[(1/r)^2 - 1 \right] \left[(r_c/r)^2 - 1 \right]^2, \text{ for } r < r_c.$$

5. The pair force **ff** of the **WF** potential vanishes smoothly at **rc**. Hence, this potential does not require truncating and shifting at **rc** (unlike **LJ**).
6. For typical atomic systems, the default value for **rc** is 2.0 in units of σ , the smallest value of r where the potential crosses zero. For (nano)colloids, we can use a smaller value of **rc**, but with a different pre-factor (see [81]).

If a given pair of particles is close enough to interact, we must compute the force between these particles. We also compute the contribution of pair ij to the total potential energy, because it is useful to monitor possible energy drift. However, we do not need to know the potential energy of interaction to integrate Newton's equation of motion.

To give an example: the x -component of the force is given by

$$\begin{aligned} f_x(r) &= -\frac{\partial u(r)}{\partial x} \\ &= -\left(\frac{x}{r}\right)\left(\frac{\partial u(r)}{\partial r}\right), \end{aligned}$$

which, for a Lennard-Jones system equals (in reduced units):

$$f_x(r) = \frac{48x}{r^2} \left(\frac{1}{r^{12}} - 0.5 \frac{1}{r^6} \right).$$

4.2.3 Integrating the equations of motion

Now that we have computed all forces between the particles, we can integrate Newton's equations of motion. Several algorithms have been designed to do this. There are good reasons to prefer certain MD algorithms over others. To explain the relevant considerations, we will discuss some MD algorithms in more detail. In the simple example shown in Algorithm 6, we have used the so-called Verlet algorithm. As we will discuss later, the Verlet algorithm is not only one of the simplest, but it is also usually the best.

To derive the Verlet algorithm, we start with a Taylor expansion of the coordinate of a particle, around time t ,

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\Delta t^3}{3!} \ddot{r} + \mathcal{O}(\Delta t^4),$$

similarly,

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\Delta t^3}{3!} \ddot{r} + \mathcal{O}(\Delta t^4).$$

Summing these two equations, we obtain

$$r(t + \Delta t) + r(t - \Delta t) = 2r(t) + \frac{f(t)}{m}\Delta t^2 + \mathcal{O}(\Delta t^4)$$

or

$$r(t + \Delta t) \approx 2r(t) - r(t - \Delta t) + \frac{f(t)}{m}\Delta t^2. \quad (4.2.3)$$

Algorithm 6 (Integrating the equations of motion)

<pre> function Integrate-V sumv=0 sumv2=0 for 1 ≤ i ≤ npart do xx=2*x(i)-xm(i)+delt**2*fx(i) vi=(xx-xm(i))/(2*delt) sumv=sumv+vi sumv2=sumv2+vi**2 xm(i)=x(i) x(i)=xx enddo temp=sumv2/(nf) etot=(en+0.5*sumv2)/npart end function </pre>	integrate equations of motion MD loop Verlet algorithm (4.2.3) velocity (4.2.4) velocity center of mass total kinetic energy update “old” positions update “current” positions current temperature and total energy per particle may be used elsewhere
--	--

Specific Comments (for general comments, see p. 7)

1. In the absence of external forces, Newton’s equations of motion conserve momentum. This also holds for the discretized form of the equations of motion. Hence, during a single timestep, `sumv/npart`, the velocity of the center of mass of the system, should remain conserved (usually zero) to within machine precision, unless there are obstacles (e.g., walls) in the system that can act as momentum sinks.
2. Newton’s equations of motion also conserve the total energy of an isolated system. However, in the discretized version, the conservation of `etot` is only approximate. Yet, good algorithms will typically suppress the long-time drift in the energy. It is therefore important to monitor `etot`, as a drift in this quantity may signal programming errors.
3. In this function, we use the Verlet algorithm (4.2.3) to integrate the equations of motion. The velocities are calculated using Eq. (4.2.4).
4. To compute the temperature, the `sumv2` is divided by `nf`, the number of degrees of freedom: see Algorithm 4.

The estimate of the new position contains an error that is of order Δt^4 , where Δt is the time step in our Molecular Dynamics scheme. Note that the Verlet algorithm does not use the velocity to compute the new position. One can, however, derive the velocity from knowledge of the trajectory, using

$$r(t + \Delta t) - r(t - \Delta t) = 2v(t)\Delta t + \mathcal{O}(\Delta t^3)$$

or

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2). \quad (4.2.4)$$

This expression for the velocity is only accurate to order Δt^2 . As shown below (Eq. (4.3.7)), it is possible to obtain more accurate estimates of the velocity (and thereby of the kinetic energy) using a Verlet-like algorithm (i.e., an algorithm that yields trajectories identical to that given by Eq. (4.2.3)). In our program, we use the velocities only to compute the kinetic energy and, thereby, the instantaneous temperature.

Now that we have computed the new positions, we may discard the positions at time $t - \Delta t$. The current positions become the old positions and the new positions become the current positions.

After each time step, we compute the current temperature (`temp`), the current potential energy (`en`) calculated in the force loop, and the total energy (`etot`). Note that the total energy should be approximately conserved.

The above example completes our introduction to the Molecular Dynamics method. We kept it as brief as possible to illustrate the fact that the core of an MD program is very simple. The reader should now be able to write a basic Molecular Dynamics program for liquids or solids consisting of spherical particles.

In what follows, we discuss in more detail, the methods available to integrate the equations of motion. In Chapter 5 we discuss measurements in Molecular Dynamics simulations. Important extensions of the Molecular Dynamics method are discussed in Chapter 7.

4.3 Equations of motion

The core of a good Molecular Dynamics program is a good algorithm to integrate Newton's equations of motion. The choice of the algorithm is therefore crucial. However, although it is easy to recognize a *bad* algorithm, it is not immediately obvious what criteria a *good* algorithm should satisfy. Below we list a number of possible criteria and show that the widely-used Verlet algorithm performs well where it matters most.

We stress that our description below lacks any semblance of mathematical rigor. We refer readers who wish to read an in-depth discussion about the good and bad properties of MD algorithms to the book by Leimkuhler and Matthews [23].

4.3.1 Accuracy of trajectories and the Lyapunov instability

One might expect that a good MD algorithm would accurately predict the trajectories of all particles for both short and long times. Sadly, no such algorithm exists. For essentially all systems that we study by MD simulations, we are in the regime where the trajectory of the system through phase space depends sensitively on the initial conditions. This means that two trajectories that are initially very close will diverge exponentially as time progresses. Dynamical systems that exhibit such extreme sensitivity on initial conditions are called *chaotic*. Not

all many-body systems show this extreme sensitivity to initial conditions, but most many-body systems at not-too-low temperatures do.

Small differences in our description of the initial conditions can appear due to many causes: the limited number of digits of floating-point numbers, or the fact that we use an approximate algorithm to integrate the equations of motion: in that case, chaotic behavior implies that trajectories obtained with slightly different time steps will still inevitably diverge exponentially from each other.

Small errors that grow exponentially lead to a complete decorrelation between the “true” trajectory of a system and the corresponding trajectory generated in a simulation.

To illustrate the extreme sensitivity of the trajectories to small differences in initial conditions, let us consider a liquid of N atoms. Given the initial atomic positions and momenta $(\mathbf{r}^N(0), \mathbf{p}^N(0))$, we could use the Verlet MD code introduced in the previous section to predict the positions and momenta of these atoms at a later time t .

$$\mathbf{r}(t) = f \left[\mathbf{r}^N(0), \mathbf{p}^N(0); t \right].$$

Let us now consider how the positions of the particles at time t would change if we perturbed the initial conditions (say, some of the momenta) by a small amount ϵ . Due to this change in the initial conditions, we would obtain a different value for r at time t :

$$\mathbf{r}'(t) = f \left[\mathbf{r}^N(0), \mathbf{p}^N(0) + \epsilon; t \right].$$

We denote the difference between $\mathbf{r}(t)$ and $\mathbf{r}'(t)$ by $\Delta\mathbf{r}(t)$. For sufficiently short times and small ϵ , $\Delta\mathbf{r}(t)$ is linear in ϵ . However, the coefficient of the linear dependence diverges exponentially; that is,

$$|\Delta\mathbf{r}(t)| \sim \epsilon \exp(\lambda t). \quad (4.3.1)$$

This so-called *Lyapunov* instability of the trajectories is responsible for our inability to accurately predict a trajectory for all but the shortest simulations. It should be stressed that the existence of a Lyapunov instability is a feature of the *system*, not of the algorithm. Hence, it cannot be fixed by choosing another algorithm to integrate Newton’s equations of motion.

Although the Lyapunov instability may seem to be bad news, it is actually good news because systems that are not chaotic (e.g., an ideal gas in a cubic container with smooth hard walls) will not be sampling all of the available phase space, not even in an infinitely long run: they are *non-ergodic* (see section 2.4).

The exponent λ in Eq. (4.3.1) is called the Lyapunov exponent, or more precisely, the largest Lyapunov exponent. There are $2dN$ Lyapunov exponents, but the largest one dominates the long-time exponential divergence of initially close trajectories. To illustrate that it is effectively impossible to beat the Lyapunov instability, let us assume that we wish to maintain a certain bound Δ_{\max} on

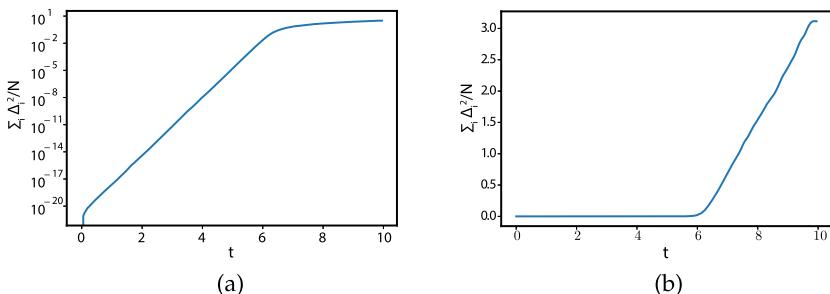


FIGURE 4.1 Illustration of the Lyapunov instability in a simulation of a WF system of 500 atoms at a density $\rho = 0.8$ and a temperature $T = 1.0$. The figure shows the time dependence of the squared distance per particle (in units σ^2) between two trajectories that were initially very close (see text). The total length of the run in reduced units was 10. We chose a rather small time-step $\Delta t = 0.0005$ for which the energy conservation is better than 1 in 10^6 . (a) shows that initially, the original and perturbed trajectories diverge exponentially. However, after $t \approx 6$, the trajectories diverge diffusively, meaning that their mean-square separation increases linearly with t (see panel b).

$|\Delta\mathbf{r}(t)|$, in the interval $0 < t < t_{\max}$; how large an initial error (ϵ) can we afford? From Eq. (4.3.1), we deduce

$$\epsilon \sim \Delta_{\max} \exp(-\lambda t_{\max}).$$

Hence, the acceptable error in our initial conditions decreases exponentially with t_{\max} , the length of the run. To illustrate that this effect is real, we show the result of two almost identical simulations: the second differs from the first in that the velocity of one particle was increased by 10^{-8} and another decreased by the same amount. We monitor the sum of the squares of the differences of the positions of all particles:

$$\sum_{i=1}^N |\mathbf{r}_i(t) - \mathbf{r}'_i(t)|^2.$$

As can be seen in Fig. 4.1a, this measure of the distance does indeed grow exponentially with time for short times. However, as in the early universe, the exponential growth cannot continue, and Fig. 4.1b shows a transition to a diffusive divergence, which is to be expected if the trajectories are fully uncorrelated.

After 10 time units, the two systems that were initially very close have become very nearly uncorrelated. It should be stressed that this run was performed using perfectly normal parameters (density, temperature, time step). The only unrealistic thing about this simulation is that it was very short. Most Molecular Dynamics simulations explore the evolution of a system over a time interval that is many orders of magnitude longer.

It is instructive to consider the difference between modeling the properties of an N -body system and predicting the trajectory of, say, a satellite: usually,

the trajectories of satellites are not very chaotic. But we do need to predict their orbit as accurately as possible. This requires other algorithms than MD: not better, just more focused on short to medium-term accuracy, often at the expense of long-term energy drift. In contrast, most systems studied by MD simulations are in the regime where the system's trajectory through phase space depends sensitively on the initial conditions.²

We should expect that, upon integrating the equations of motion, any initial error, no matter how small, will always cause the simulated trajectory to diverge exponentially from the true trajectory that starts from the same initial conditions. The Lyapunov instability might seem to deal a devastating blow to the very idea of Molecular Dynamics simulations. However, we have good reasons to assume that the situation is, as the saying goes *desperate but not serious*.

We need to explain why the Lyapunov instability does not make all MD simulations meaningless. First of all, we note that the aim of an MD simulation is *hardly ever* to predict in detail how the positions and momenta of a system will evolve in time given the initial conditions: rather, we are typically interested in statistical predictions. We wish to predict the *average properties* of a system that has been prepared in an initial state characterized by some macroscopic descriptors (e.g., its total energy or the phase that it is in): we need not specify all coordinates or momenta.

In this respect, the philosophy behind MD simulations differs fundamentally from that underlying numerical schemes for predicting the trajectory of a satellite through space: in the latter case, we really wish to predict the true trajectory. We cannot afford to launch an ensemble of satellites and make statistical predictions about their destination. Yet, in MD simulations, statistical predictions are good enough. Still, this would not justify the use of inaccurate trajectories unless the trajectories obtained numerically are, in some sense, representative of true trajectories.

An example from another chaotic system may help illustrate this point: the time evolution of the weather is chaotic. That is why it is difficult to get accurate weather predictions for periods longer than a few days. However, by considering a large number of such weather simulations, we can still make reliable statements about the “average” weather, i.e., the climate.

As MD simulations aim to measure the time-averaged properties of a system sufficiently close to a real trajectory, we wish our MD simulation to provide a good approximation to *a* (any) real trajectory, but not necessarily the one with the same initial conditions. A good MD simulation is, therefore, one that generates a numerical path in phase space that remains close to a real trajectory (the so-called *shadow orbit*), even though we do not know what that trajectory is.

Surprisingly (and fortunately), it appears that shadow orbits are better behaved (i.e., track the numerical trajectories better) for systems in which small

² Most systems, but not all: systems that are almost harmonic. (e.g., low-temperature solids), are typically not, or barely, chaotic.

differences in the initial conditions lead to an exponential divergence of trajectories than for the seemingly simpler systems that show no such divergence [109]. Despite this reassuring evidence (see also section 4.3.5 and ref. [110]), it should be emphasized that it is just evidence and not proof. In fact, the existence of shadow orbits has not been proven for any class of systems that is of interest for MD simulations, although there is supporting, circumstantial numerical evidence (see e.g., [109]). In short: our trust in Molecular Dynamics simulation as a tool to generate good approximations to real trajectories of a many-body system is based largely on belief or, to phrase it more kindly: it is a working hypothesis. To conclude this discussion, let us say that there is clearly a skeleton in the closet. We believe this skeleton will not haunt us, and we quickly close the closet. For more details, the reader is referred to [31,110–112].

In the following section, we argue that the Verlet algorithm is a *good* algorithm.

4.3.2 Other desirable features of an algorithm

Speed

Although, at first sight, the speed of an algorithm seems important, it is usually not very relevant because the fraction of time spent on integrating the equations of motion (as opposed to computing the interactions) is small, at least for atomic and simple molecular systems.

Maximum permissible time step

Stability when using large time steps might seem more important, because the longer the time step that we can use, the fewer evaluations of the forces are needed per unit of simulation time. This observation would suggest that it is advantageous to use sophisticated algorithms that allow us to use long time steps. In practice, this problem is usually addressed by still using a simple algorithm, but using multiple time-steps (see section 14.3 in Chapter 14).

Energy conservation

Energy conservation is an important criterion, but actually, we should distinguish two kinds of energy conservation, namely: short-time and long-time energy conservation. The more sophisticated higher-order algorithms tend to have very good energy conservation for short times (i.e., during a small number of time steps). However, they often have the undesirable feature that the overall energy drifts for long times. In contrast, Verlet-style algorithms tend to have only moderately good short-term energy conservation but little long-term drift. Below, we shall rationalize these observations.

Time-reversibility

As Newton's equations of motion are time-reversible, our numerical trajectory should obey that same property: i.e., if we run our MD algorithm in reverse,

we should get back to our initial condition (but for possible rounding errors at the level of the machine precision). It is simple to demonstrate that the Verlet algorithm obeys time reversibility. Denoting the coordinates generated by the Verlet algorithm by r_V , we note that the Verlet equations of motion that relate $r_V(t)$ to $r_V(t + \Delta t)$ and $r_V(t - \Delta t)$ is invariant under a change of sign of Δt :

$$r_V(t + \Delta t) + r_V(t - \Delta t) = 2r_V(t) + \frac{f(t)}{m} \Delta t^2.$$

In fact, many algorithms, for instance, the predictor-corrector schemes, see SI L.1, and some of the schemes used to deal with constraints, are *not* time reversible. That is: future and past phase space coordinates do not play a symmetric role in such algorithms. As a consequence, if one were to reverse the momenta of all particles at a given instant, the system would not trace back its trajectory in phase space, even if the simulation would be carried out with infinite machine accuracy.

In addition to the lack of time-reversibility, many seemingly reasonable algorithms differ in another crucial respect from Hamilton's equation of motion: true Hamiltonian dynamics leaves the magnitude of any volume element in phase space unchanged.³ Non-conservation of volume in phase space may sound like an esoteric objection to an algorithm, but it is not. Without attempting a rigorous formulation of the problem, let us simply note that all trajectories that correspond to a particular energy E are contained in a (hyper) surface Ω in phase space.⁴ If we let Hamilton's equation of motion act on all points in this volume (i.e., we let the volume evolve in time), then we end up with exactly the same volume. However, a non-area-preserving algorithm will map the volume $\Omega(E)$ onto another (usually larger) volume Ω' . After sufficiently long times, we expect that the non-area-preserving algorithm will have greatly expanded the volume of our system in phase space. Expanding the volume in phase space is not compatible with energy conservation. Hence, it is plausible that non-reversible algorithms will have serious long-term energy drift problems. Reversible, area-preserving algorithms will not change the magnitude of the volume in phase space. This property is not sufficient to guarantee the absence of long-term energy drift, but it is at least compatible with it. It is possible to check whether an algorithm is area-preserving by computing the Jacobian associated with the transformation of old to new phase space coordinates.

³ Let us denote the velocity of a system through phase space by $\dot{\mathbf{r}}$. Now consider a small volume element in phase space, bounded by a (hyper)surface S , and consider the rate at which its volume ΔV changes: $\Delta \dot{V} = \int_{\text{surface}} \dot{\mathbf{r}} \cdot dS$. Using Gauss's theorem, we can write this as $\int_{\Delta V} \nabla \cdot \dot{\mathbf{r}} dV$. Using Hamilton's equations of motion $\partial H / \partial q_\alpha = -\dot{p}_\alpha$ and $\partial H / \partial p_\alpha = \dot{q}_\alpha$, it follows that $\nabla \cdot \dot{\mathbf{r}} = 0$, and hence the volume ΔV is conserved.

⁴ As $\Omega(E)$ is a hyper-surface, people usually refer to the volume-preserving property of the Hamilton equations of motion as *area-preserving*. Hamilton's equations of motion conserve volume in phase space, even when the Hamiltonian contains terms that depend explicitly on time such that the energy is not constant.

Finally, it should be noted that even when we integrate a time-reversible algorithm, we shall find that the numerical implementation is hardly ever truly time reversible. This is so because we work on a computer with finite machine precision using floating-point arithmetic that results in rounding errors (on the order of the machine precision).

In the remainder of this section, we shall discuss some of these points in more detail. Before doing so, let us first consider how the Verlet algorithm scores on the points mentioned above. First of all, the Verlet algorithm is fast. But we had argued that this is relatively unimportant. Second, the error in the positions/velocities scales as $\Delta t^4/\Delta t^2$. Hence it is not particularly accurate for longer time steps and we have to compute the forces on all particles rather frequently. Third, it requires about as little memory as is at all possible. This is less relevant now than it was in Verlet's time, when it was a crucial advantage, but it is still useful when we simulate very large systems. Fourth, its short-term energy conservation is only fair (slightly better for the versions that use a more accurate expression for the velocities: see Eq. (4.3.7)). What is much more important is that it exhibits little long-term energy drift. This is related to the fact that the Verlet algorithm is time reversible and area-preserving. In fact, although the Verlet algorithm does not conserve the total energy of the system exactly, it does conserve a pseudo-Hamiltonian approaching the true Hamiltonian in the limit of infinitely short time steps (see section 4.3.4).⁵ The accuracy of the trajectories generated with the Verlet algorithm is not impressive. But then, it would hardly help to use a better algorithm, as such an algorithm would only postpone the unavoidable exponential growth of the error in the trajectory by a few hundred time-steps (see section 4.3.1), but no algorithm is good enough to keep the trajectories close to the true trajectories for a time comparable to the duration of a typical Molecular Dynamics run.

4.3.3 Other versions of the Verlet algorithm

Let us now briefly look at some alternatives to the Verlet algorithm, but before we do, we first consider a naive algorithm that is based simply on a truncated Taylor expansion of the particle coordinates:

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \dots$$

If we truncate this expansion beyond the term in Δt^2 , we obtain the so-called forward-Euler algorithm. Although this algorithm looks similar to the Verlet algorithm, it is much worse on virtually all counts. In particular, it is not reversible nor area-preserving and suffers from a (catastrophic) energy drift. To say that the Euler algorithm is not recommended is, therefore, an understatement.

⁵ The rigorous conservation of the pseudo-Hamiltonian only holds if the potential energy function is infinitely differentiable: introducing a cut-off, even a smoothed cut-off, will undermine this argument.

Several algorithms are rigorously equivalent to the Verlet scheme. The simplest among these is the so-called Leap Frog algorithm [28]. This algorithm evaluates the velocities at half-integer time steps and uses these velocities to compute the new positions. To derive the Leap-Frog algorithm from the Verlet scheme, we start by defining the velocities at half-integer time steps as follows:

$$v(t - \Delta t/2) \equiv \frac{r(t) - r(t - \Delta t)}{\Delta t}$$

and

$$v(t + \Delta t/2) \equiv \frac{r(t + \Delta t) - r(t)}{\Delta t}.$$

From the latter equation, we trivially obtain an expression for the new positions based on the old positions and velocities:

$$r(t + \Delta t) = r(t) + \Delta t v(t + \Delta t/2). \quad (4.3.2)$$

From the Verlet algorithm, we get the following expression for the update of the velocities:

$$v(t + \Delta t/2) = v(t - \Delta t/2) + \Delta t \frac{f(t)}{m}. \quad (4.3.3)$$

As the Leap Frog algorithm can be derived from the Verlet algorithm, it gives rise to identical trajectories. Note, however, that the velocities are not defined at the same times as the positions. As a consequence, kinetic and potential energy are also not defined at simultaneously, and hence we cannot directly compute the total energy in the Leap Frog scheme.

It is, however, possible to cast the Verlet algorithm in a form that uses positions and velocities computed at equal times. This so-called velocity Verlet algorithm [113] looks like a Taylor expansion for the coordinates:

$$r(t + \Delta t) = r(t) + v(t) \Delta t + \frac{f(t)}{2m} \Delta t^2. \quad (4.3.4)$$

However, the update of the velocities is different from the Euler scheme:

$$v(t + \Delta t) = v(t) + \frac{f(t + \Delta t) + f(t)}{2m} \Delta t. \quad (4.3.5)$$

Note that, in this algorithm, we can compute the new velocities only after we have computed the new positions and, from these, the new forces. It is not immediately obvious that this scheme is equivalent to the original Verlet algorithm. To show this, we note that

$$r(t + 2\Delta t) = r(t + \Delta t) + v(t + \Delta t) \Delta t + \frac{f(t + \Delta t)}{2m} \Delta t^2,$$

and Eq. (4.3.4) can be written as

$$r(t) = r(t + \Delta t) - v(t)\Delta t - \frac{f(t)}{2m}\Delta t^2.$$

By addition, we get

$$r(t + 2\Delta t) + r(t) = 2r(t + \Delta t) + [v(t + \Delta t) - v(t)]\Delta t + \frac{f(t + \Delta t) - f(t)}{2m}\Delta t^2.$$

Substitution of Eq. (4.3.5) yields

$$r(t + 2\Delta t) + r(t) = 2r(t + \Delta t) + \frac{f(t + \Delta t)}{m}\Delta t^2,$$

which, indeed, is the coordinate version of the Verlet algorithm. During one time step, the Verlet algorithm introduces an error of $\mathcal{O}(\Delta t^4)$ in the change of the particle positions. However, the error in the velocity update is of order Δt^2 . It is possible to obtain a more accurate velocity estimate using the so-called velocity-corrected Verlet algorithm, for which the error in both the positions and the velocities is of order $\mathcal{O}(\Delta t^4)$.

The velocity-corrected Verlet algorithm is derived as follows. First write down a Taylor expansion for $r(t + 2\Delta t)$, $r(t + \Delta t)$, $r(t - \Delta t)$ and $r(t - 2\Delta t)$:

$$r(t + 2\Delta t) = r(t) + 2v(t)\Delta t + \dot{v}(t)(2\Delta t)^2/2! + \ddot{v}(2\Delta t)^3/3! + \dots$$

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \dot{v}(t)\Delta t^2/2! + \ddot{v}\Delta t^3/3! + \dots$$

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \dot{v}(t)\Delta t^2/2! - \ddot{v}\Delta t^3/3! + \dots$$

$$r(t - 2\Delta t) = r(t) - 2v(t)\Delta t + \dot{v}(t)(2\Delta t)^2/2! - \ddot{v}(2\Delta t)^3/3! + \dots.$$

It then follows that

$$8[r(t + \Delta t) - r(t - \Delta t)] - [r(t + 2\Delta t) - r(t - 2\Delta t)] + \mathcal{O}(\Delta t^5) = 12v(t)\Delta t \quad (4.3.6)$$

or, equivalently,

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{1.5\Delta t} - \frac{r(t + 2\Delta t) - r(t - 2\Delta t)}{12\Delta t} + \mathcal{O}(\Delta t^4). \quad (4.3.7)$$

Note that this estimate of the velocity, which is correct up to $\mathcal{O}(\Delta t^4)$, can only be evaluated after the positions at time $t \pm \Delta t$ and $t \pm 2\Delta t$ have been generated using the Verlet algorithm (Eq. (4.2.3)).

Velocity-Verlet: a misnomer?

We should point out that the current discussion of the velocity-Verlet algorithm is an oversimplification, as the “velocities” in the velocity-Verlet scheme are not the position time-derivatives $\dot{x}(t_i)$, but only quantities formally defined as

Algorithm 7 (Velocity-Verlet time step)

function vel_verlet	
K=0	assume current x, v and f known from previous time step
for $1 \leq i \leq npart$ do	kinetic energy accumulator
$v(i)=v(i)+f(i)*delt2/m$	half-update of v_i (Eq. (4.3.5))
$x(i)=x(i)+v(i)*delt$	update x_i using Eq. (4.3.4)
enddo	
$f = \text{force}$	update all forces
for $1 \leq i \leq npart$ do	2nd half-update of v_i (Eq. (4.3.5))
$v(i)=v(i)+f(i)*delt2/m$	update kinetic energy.
$K=K+m*v(i)*v(i)/2$	
enddo	
end function	

Specific Comments (for general comments, see p. 7)

1. In this function $delt = \Delta t$ and $delt2 = \Delta t/2$. We show the mass m explicitly, although in other examples we assume that m is the unit of mass (i.e., $m=1$).
2. The function **force** calculates the forces on all particles.

$v_i \equiv p_i/m_i$. For $\Delta t \rightarrow 0$, $p_i/m \rightarrow \dot{x}(t_i)$. However, as was pointed out by Gans and Shalloway [114], for any finite Δt , v_i differs from the instantaneous velocity that is obtained by differentiating a smooth interpolation of the positions. In ref. [114], it is argued that the “velocity-Verlet” algorithm should actually be called the “momentum-Verlet” algorithm since momenta, but not velocities, are correctly computed. The error that results from interpreting the Verlet “velocity” as the velocity in Hamilton’s equation of motion can have noticeable effects on many observables, most directly on the estimate of the temperature of a system (see Chapter 7). At this stage, we ignore this subtle but important point, mainly to keep the discussion simple. For more details, we refer the reader to refs. [114,115]. However, we urge all readers to be aware that interpreting the Verlet “velocity” as a true (consistent) velocity may lead to (un)expected problems (see, for example, Illustration 1).

Illustration 1 (Role of momentum in the velocity-Verlet algorithm). One of the characteristics of the original (position) Verlet algorithm is that we do not need to know the velocity to propagate the positions of the particles. However, to compute the temperature of a system, we need to know the average kinetic energy of the particles, which is usually expressed in terms of the particle velocities. In the position-Verlet algorithm, one can estimate the velocities

from the displacements of the particles during a time step, but this is a rather *ad hoc* procedure. In contrast, the velocities of the particles appear explicitly in the otherwise completely equivalent velocity-Verlet algorithm. However, the velocity Verlet algorithm only yields the actual velocity in the limit that the timestep approaches zero. For a non-zero time step, the algorithm updates the momenta, which are not equal to mv .

In Hamiltonian dynamics (see Appendix A), the velocity of a particle i with generalized coordinate q_i and conjugate momentum p_i is given by:

$$v_i = \dot{q}_i = \left(\frac{\partial \mathcal{H}}{\partial p_i} \right),$$

where \mathcal{H} is the Hamiltonian of the total system.

In a typical molecular dynamics simulation, the dynamics follow from the Hamiltonian given by:

$$\mathcal{H}_0(\mathbf{p}, \mathbf{q}) = \frac{\mathbf{p}^2}{2m} + \mathcal{U}(\mathbf{q}).$$

And if we would solve these equations of motions exactly, we have for the velocities:

$$v = \frac{p}{m}.$$

In a simulation, we solve these equations of motion only approximately by discretizing time. The positions at the discrete times lie on a “shadow” trajectory, along which the “shadow Hamiltonian” is conserved (see section 4.3.1). One consequence is that the velocity at time t , defined as the derivative of the position along this continuous shadow path, is *not* given by the quantity that is called the velocity in the velocity-Verlet algorithm. To bring out the role of momentum in the “velocity”-Verlet algorithm, it is better to use the notation of Eastwood et al. [115] and write:

$$\begin{aligned} \mathbf{q}(t + \delta t) &= \mathbf{q}(t) + \delta t \frac{\mathbf{p}(t)}{m} + \frac{1}{2} (\delta t)^2 \frac{\mathbf{F}(t)}{m} \\ \mathbf{p}(t + \delta t) &= \mathbf{p}(t) + \frac{1}{2} \delta t [\mathbf{F}(t) + \mathbf{F}(t + \delta t)]. \end{aligned}$$

Gans and Shalloway [114] showed that, for a harmonic oscillator, there can be a significant difference between $p(t)/m$ and \dot{q} , as obtained by interpolation of the discretized trajectory. The consequence of this nontrivial difference is that we would obtain a different temperature depending on whether we would compute the temperature using momenta or velocities. The difference between \dot{q} and p/m implies that we should be careful when computing the temperature. As explained in section 5.1.1, the following relation follows directly from the properties of the canonical ensemble:

$$\langle p_i \dot{q}_i \rangle = k_B T .$$

As is clear from the above discussion, we should expect the following three ways of estimating the temperature to be different for a finite time step (δt):

$$k_B T_{pv} = \langle p_i v_i \rangle_{\delta t}, \quad (4.3.8)$$

$$k_B T_{p^2} = \langle p_i p_i / m_i \rangle_{\delta t}, \quad (4.3.9)$$

$$k_B T_{v^2} = \langle m_i v_i v_i \rangle_{\delta t}. \quad (4.3.10)$$

Eastwood et al. [115] computed the different temperature estimates as a function of δt for a fluid of diatomic molecules and compared their results with those for a simple 1d harmonic oscillator, for which the expressions of Eq. (4.3.8) can be computed analytically. In Fig. 4.2a, we show the analytical results as a function of δt . Only in the limit $\delta t \rightarrow 0$ do the three temperatures converge to the same value. For typical time steps in a simulation (2 fs), we do see a non-negligible difference between the three temperatures. Importantly, Eq. (4.3.8) yields the correct temperature independent of the time step. Eastwood et al. [115] verified that their numerical results for a diatomic fluid agreed well with the analytical results for the 1d harmonic oscillator (see Fig. 4.2a).

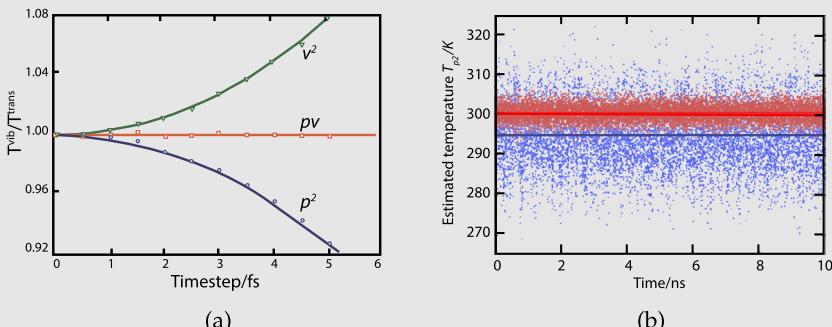


FIGURE 4.2 (a) Equipartition for the diatomic fluid over a range of integration time steps for the different ways of computing the temperatures. The lines are analytical results, and the symbols are the results of simulations. (b) Temperatures of the protein ubiquitin (blue (dark gray in print version) points) and water solvent (red (mid gray in print version) points) are shown as a function of simulation time. Figures based on data from Eastwood et al. [115].

However, a 1d harmonic oscillator is hardly a representative example of the systems studied in an MD simulation. Therefore, Eastwood et al. [115] also simulated a protein in solution and showed that using the conventional expression for the kinetic energy (4.3.10) yields a marked apparent difference between the temperature of the solvent and the protein (see Fig. 4.2b). This difference disappears if Eq. (4.3.8) is used instead of Eq. (4.3.10) or Eq. (4.3.9). Note that the apparent temperature difference is not small. Using the latter equations one might conclude incorrectly that the system is not yet equilibrated.

Higher-order schemes

For most Molecular Dynamics applications, Verlet-like algorithms are perfectly adequate. However, historically, higher-order algorithms, which require information about higher-order derivatives of the particle coordinates, were often used (mainly because some applied mathematicians criticized the Verlet algorithm as too crude). At present, most higher-order algorithms have disappeared from MD codes (see SI L.1). We refer the reader who wishes to know more about the relative merits of other algorithms for Molecular Dynamics simulations to the excellent early review by Berendsen and van Gunsteren [116].

4.3.4 Liouville formulation of time-reversible algorithms

Thus far, we have considered algorithms for integrating Newton's equations of motion by starting from a Taylor expansion of the position in powers of the time step. However, a more physics-based perspective on Verlet-style algorithms was introduced by Tuckerman et al. [117], who showed how to derive time-reversible, area-preserving MD algorithms systematically from the Liouville formulation of classical mechanics. The same approach has been developed independently by Sexton and Weingarten [118] in the context of hybrid Monte Carlo simulations (see section 13.3.1). As the Liouville formulation provides considerable insight into what makes an algorithm a good algorithm, we briefly review the Liouville approach.

Let us consider an arbitrary function f that depends on all the coordinates and momenta of the N particles in a classical many-body system. We assume that $f(\mathbf{p}^N(t), \mathbf{r}^N(t))$ depends on the time t implicitly, that is, only through the dependence of $(\mathbf{p}^N, \mathbf{r}^N)$ on t . The time derivative of f is \dot{f} :

$$\begin{aligned}\dot{f} &= \dot{\mathbf{r}} \frac{\partial f}{\partial \mathbf{r}} + \dot{\mathbf{p}} \frac{\partial f}{\partial \mathbf{p}} \\ &\equiv iL f,\end{aligned}\tag{4.3.11}$$

where we have used the shorthand notation \mathbf{r} for \mathbf{r}^N and \mathbf{p} for \mathbf{p}^N . The last line of Eq. (4.3.11) defines the *Liouville operator*, iL

$$iL = \dot{\mathbf{r}} \frac{\partial}{\partial \mathbf{r}} + \dot{\mathbf{p}} \frac{\partial}{\partial \mathbf{p}}.\tag{4.3.12}$$

If iL is not explicitly time dependent, we can integrate Eq. (4.3.11) to obtain

$$f \left[\mathbf{p}^N(t), \mathbf{r}^N(t) \right] = \exp(iLt) f \left[\mathbf{p}^N(0), \mathbf{r}^N(0) \right].\tag{4.3.13}$$

In all cases of practical interest, we cannot do much with this formal solution because evaluating the right-hand side is still equivalent to the exact integration of the classical equations of motion. However, in a few simple cases, the formal solution is known explicitly. In particular, suppose that our Liouville operator

contained only the first term on the right-hand side of Eq. (4.3.12). We denote this part of iL by iL_r :

$$iL_r \equiv \dot{\mathbf{r}}(0) \frac{\partial}{\partial \mathbf{r}}, \quad (4.3.14)$$

where $\dot{\mathbf{r}}(0)$ is the value of $\dot{\mathbf{r}}$ at time $t = 0$. If we insert iL_r in Eq. (4.3.13) and use a Taylor expansion of the exponential on the right-hand side, we get

$$\begin{aligned} f(t) &= f(0) + iL_r t f(0) + \frac{(iL_r t)^2}{2!} f(0) + \dots \\ &= \exp\left(\dot{\mathbf{r}}(0)t \frac{\partial}{\partial \mathbf{r}}\right) f(0) \\ &= \sum_{n=0}^{\infty} \frac{(\dot{\mathbf{r}}(0)t)^n}{n!} \frac{\partial^n}{\partial \mathbf{r}^n} f(0) \\ &= f\left[\mathbf{p}^N(0), (\mathbf{r} + \dot{\mathbf{r}}(0)t)^N\right]. \end{aligned} \quad (4.3.15)$$

Hence, the effect of $\exp(iL_r t)$ is a simple shift of coordinates. Similarly, the effect of $\exp(iL_p t)$, with iL_p defined as

$$iL_p \equiv \dot{\mathbf{p}}(0) \frac{\partial}{\partial \mathbf{p}}, \quad (4.3.16)$$

is a simple shift of momenta. The total Liouville operator, iL , is equal to $iL_r + iL_p$. Unfortunately, we cannot replace $\exp(iLt)$ by $\exp(iL_r t) \times \exp(iL_p t)$, because iL_r and iL_p are non-commuting operators. For non-commuting operators A and B , we have

$$\exp(A + B) \neq \exp(A) \exp(B). \quad (4.3.17)$$

However, we can make use of the *Trotter identity*:

$$e^{(A+B)} = \lim_{P \rightarrow \infty} \left(e^{A/2P} e^{B/P} e^{A/2P} \right)^P. \quad (4.3.18)$$

In the limit $P \rightarrow \infty$, this relation is formally correct, but of limited practical value. However, for large but finite P , we have

$$e^{(A+B)} = \left(e^{A/2P} e^{B/P} e^{A/2P} \right)^P e^{\mathcal{O}(1/P^2)}. \quad (4.3.19)$$

Now let us apply this expression to the formal solution of the Liouville equation. To this end, we make the identification

$$\frac{A}{P} \equiv \frac{iL_p t}{P} \equiv \Delta t \dot{\mathbf{p}}(0) \frac{\partial}{\partial \mathbf{p}}$$

and

$$\frac{B}{P} \equiv \frac{iL_r t}{P} \equiv \Delta t \dot{\mathbf{r}}(0) \frac{\partial}{\partial \mathbf{r}},$$

where $\Delta t = t/P$. The idea is now to replace the formal solution of the Liouville equation by the discretized version, Eq. (4.3.19). In this scheme, one time-step corresponds to once applying the operator

$$e^{iL_p \Delta t / 2} e^{iL_r \Delta t} e^{iL_p \Delta t / 2}.$$

Let us see what the effect is of this operator on the coordinates and momenta of the particles. First, we apply $\exp(iL_p \Delta t / 2)$ to f and obtain

$$e^{iL_p \Delta t / 2} f \left[\mathbf{p}^N(0), \mathbf{r}^N(0) \right] = f \left\{ \left[\mathbf{p}(0) + \frac{\Delta t}{2} \dot{\mathbf{p}}(0) \right]^N, \mathbf{r}^N(0) \right\}.$$

Next, we apply $\exp(iL_r \Delta t)$ to the result of the previous step

$$\begin{aligned} & e^{iL_r \Delta t} f \left\{ \left[\mathbf{p}(0) + \frac{\Delta t}{2} \dot{\mathbf{p}}(0) \right]^N, \mathbf{r}^N(0) \right\} \\ &= f \left\{ \left[\mathbf{p}(0) + \frac{\Delta t}{2} \dot{\mathbf{p}}(0) \right]^N, [\mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(\Delta t / 2)]^N \right\}, \end{aligned}$$

and finally, we apply $\exp(iL_p \Delta t / 2)$ once more, to obtain⁶

$$f \left\{ \left[\mathbf{p}(0) + \frac{\Delta t}{2} \dot{\mathbf{p}}(0) + \frac{\Delta t}{2} \dot{\mathbf{p}}(\Delta t) \right]^N, [\mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(\Delta t / 2)]^N \right\}.$$

Note that every step in the preceding sequence corresponds to a simple shift operation in either \mathbf{r}^N or \mathbf{p}^N . It is of particular importance to note that the shift in \mathbf{r} is a function of \mathbf{p} only (because $\dot{\mathbf{r}} = \mathbf{p}/m$), while the shift in \mathbf{p} is a function of \mathbf{r} only (because $\dot{\mathbf{p}} = \mathbf{F}(\mathbf{r}^N)$). The Jacobian of the transformation from $\{\mathbf{p}^N(0), \mathbf{r}^N(0)\}$ to $\{\mathbf{p}^N(\Delta t), \mathbf{r}^N(\Delta t)\}$ is simply the product of the Jacobians of the three elementary transformations. But, as each of these Jacobians is equal to 1, the overall Jacobian is also equal to 1.

If we now consider the overall effect of this sequence of operations on the positions and momenta, we find the following:

$$\mathbf{p}(0) \rightarrow \mathbf{p}(0) + \frac{\Delta t}{2} (\mathbf{F}(0) + \mathbf{F}(\Delta t)) \quad (4.3.20)$$

⁶ Actually, we are cheating a bit: as the operators iL_p and iL_r act on the phase-space coordinates at time $t = 0$, we should carry out a similarity transform $e^{iL_p \Delta t / 2} e^{iL_r \Delta t} e^{-iL_p \Delta t / 2}$ on $e^{iL_p \Delta t / 2} f(0)$ to ensure that iL_r acts on the phase space coordinates at $t = 0$, and then yet another similarity transform to account for the effect of the final factor $e^{iL_p \Delta t / 2}$. However, as the net effect is still the same, we hide this subtlety in a footnote.

$$\begin{aligned}\mathbf{r}(0) &\rightarrow \mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(\Delta t/2) \\ &= \mathbf{r}(0) + \Delta t \dot{\mathbf{r}}(0) + \frac{\Delta^2 t}{2m} \mathbf{F}(0).\end{aligned}\quad (4.3.21)$$

But these are precisely the equations of the Verlet algorithm (in the velocity form). As we have shown that the Jacobian associated with one time-step is equal to one, we have demonstrated that the Verlet algorithm conserves volume in phase space. In fact, it does more, it conserves a number of other quantities that have a less intuitive appeal than volume in phase space. The more general conservation properties of Hamiltonian dynamics, and of the Verlet algorithm, are that they are *Symplectic* (see Appendix A.4.2).

Finally, we mention another property of the Verlet algorithm that makes it less susceptible to long-term energy drift. When we use the Verlet algorithm, we replace the true Liouville operator $\exp(iLt)$ by $\exp(iL_r \Delta t/2) \exp(iL_P \Delta t) \exp(iL_r \Delta t/2)$. In doing so, we make an error. If all (n th-order) commutators of L_p and L_r exist (i.e., if the Hamiltonian is an infinitely differentiable function of \mathbf{p}^N and \mathbf{r}^N), then, at least in principle, we can evaluate the error that is involved in this replacement:

$$\exp(iL_r \Delta t/2) \exp(iL_P \Delta t) \exp(iL_r \Delta t/2) = \exp(iL \Delta t + \epsilon), \quad (4.3.22)$$

where ϵ is an operator that can be expressed in terms of the commutators of L_p and L_r :

$$\epsilon = \sum_{n=1}^{\infty} (\Delta t)^{2n+1} c_{2n+1}, \quad (4.3.23)$$

where c_m denotes a combination of m th-order commutators. For instance, the leading term is

$$-(\Delta t)^3 \left(\frac{1}{24} [\mathbf{i}L_r, [\mathbf{i}L_r, \mathbf{i}L_p]] + \frac{1}{12} [\mathbf{i}L_p, [\mathbf{i}L_r, \mathbf{i}L_p]] \right).$$

Now the interesting thing to note is that if the expansion in Eq. (4.3.23) converges, then we can define a pseudo-Liouville operator

$$\mathbf{i}L_{\text{pseudo}} \equiv \mathbf{i}L + \epsilon/\Delta t.$$

This pseudo-Liouville operator corresponds to a pseudo-Hamiltonian, and the remarkable consequence is that this pseudo-Hamiltonian (H_{pseudo}) is rigorously conserved by Verlet style (or generalized multi-time-step) algorithms [119–122]. The difference between the conserved pseudo-Hamiltonian and the true Hamiltonian of the system is of order $(\Delta t)^{2n}$ (where n depends on the order of the algorithm). Clearly, by choosing Δt small (and, if necessary, n large), we can make the difference between the true and the pseudo-Hamiltonian as

small as we like. As the true Hamiltonian is forced to remain close to a conserved quantity, we can now understand why there is no long-term drift in the energy with Verlet-style algorithms. In some cases, we can explicitly compute the commutators (for instance, for a harmonic system) and can verify that the pseudo-Hamiltonian is indeed conserved [111]. And, even if we cannot compute the complete series of commutators, the leading term will give us an improved estimate of the pseudo-Hamiltonian. Toxvaerd [111] has verified that even for a realistic many-body system, such an approximate pseudo-Hamiltonian is very nearly a constant of motion.

The Liouville formalism allows us to derive the Verlet algorithm as a special case of the Trotter expansion of the time-evolution operator. It should be realized that the decomposition of iL as a sum of iL_r and iL_p is arbitrary. Other decompositions are possible and lead to algorithms that are sometimes more convenient.

4.3.5 One more way to look at the Verlet algorithm...

In Molecular Dynamics simulations, the Newtonian equations of motion are integrated approximately. An alternative route would be to *first* write down a time-discretized version of the action (see Appendix A), and *then* find the set of coordinates (i.e., the discretized trajectory) that corresponds to the extremum of this action. Below, we sketch this approach, which is discussed in more detail in a paper by Gillilan and Wilson [110]. We start with the continuous-time version of the action, S

$$S = \int_{t_b}^{t_e} dt \left[\frac{1}{2}m \left(\frac{dx(t)}{dt} \right)^2 - U(x) \right]$$

and discretize it as follows:

$$S_{\text{discr}} = \sum_{i=i_b}^{i_e-1} \Delta t \left[\frac{1}{2}m \left(\frac{x_{i+1} - x_i}{\Delta t} \right)^2 - U(x_i) \right],$$

where $t_b = i_b \Delta t$ and $t_e = i_e \Delta t$. As in the continuous case, we can determine the set of values of the coordinates x_i for which S_{discr} is stationary. At stationarity, the derivative of S_{discr} with respect to all x_i vanishes. It is easy to verify that this implies that

$$m \left(\frac{2x_i - x_{i+1} - x_{i-1}}{\Delta t} \right) - \Delta t \frac{\partial U(x_i)}{\partial x_i} = 0$$

or

$$x_{i+1} = 2x_i - x_{i-1} - \frac{\Delta t^2}{m} \left(\frac{\partial U(x_i)}{\partial x_i} \right),$$

which is, of course, the Verlet algorithm. This illustrates that the trajectories generated by the Verlet algorithms have an interesting “shadow” property (see ref. [110] and section 4.3): a “Verlet trajectory” that connects point x_{i_b} and x_{i_e} in a time interval $t_e - t_b$ will tend to lie close to the true trajectory that connects these two points. However, this true trajectory is not identical to the one that has the same initial velocity as the Verlet trajectory. That is,

$$\left(\frac{dx(t_b)}{dt} \right)_{\text{true}} \neq \left(\frac{x_{i_b+1} - x_{i_b-1}}{2\Delta t} \right)_{\text{Verlet}}.$$

Nevertheless, as discussed in section 4.3, the Verlet algorithm is a good algorithm in the sense that it follows from an extremum principle that forces it to approximate a true dynamical trajectory of the system under consideration in the limit that the time-step approach zero. The nice thing about the Lagrangian formulation is that we can give a simple physical interpretation of this shadow trajectory.

This attractive feature of algorithms that can be derived from a discretized action has inspired Elber and co-workers to construct a novel class of MD algorithms that are designed to yield reasonable long-time dynamics with very large time steps [123,124]. In fact, Elber and co-workers do not base their approach on the discretization of the classical action but on the so-called Onsager-Machlup action [125]. The reason for selecting this more general action is that the Onsager-Machlup action is a *minimum* for the true trajectory, while the Lagrangian action is only an extremum. It would carry too far to discuss the practical implementation of the algorithm based on the Onsager-Machlup action. For details, we refer the reader to refs. [123,124].

4.4 Questions and exercises

Question 12 (Integrating the equations of motion).

1. Which of the following quantities are conserved in the MD simulation of Case Study 4: a) potential energy, b) total momentum, c) position of the center of mass of the system, d) total angular momentum?
2. Derive that the Verlet and velocity-Verlet algorithms lead to identical trajectories.
3. Derive the Leap-Frog Algorithm by using Taylor expansions for $v(t + \frac{\Delta t}{2})$, $v(t - \frac{\Delta t}{2})$, $x(t + \Delta t)$, and $x(t)$.

Question 13 (Liouville operator). In the text we have derived the Verlet equation of motion using the Liouville operator. In this derivation we applied the operator

$$e^{iL_p \Delta t / 2} e^{iL_r \Delta t} e^{iL_p \Delta t / 2}.$$

However, in the Trotter expansion L_r and L_p can be exchanged. This exchange leads to a different set of equations of motion. Derive these equations of motion.

Compare the accuracy of the two schemes for the prediction of the momenta and the position.

Question 14 (Lyapunov exponents). In the text, we mentioned that a slight perturbation in the initial conditions makes two numerical trajectories diverge exponentially. The rate of divergence is determined by the Lyapunov exponent or, more precisely, the largest Lyapunov exponent. During a single time step, an MD algorithm transforms the positions/momenta $\{\mathbf{r}^N(t), \mathbf{p}^N(t)\}$ at time t to $\{\mathbf{r}^N(t + \Delta t), \mathbf{p}^N(t + \Delta t)\}$. This transformation is linear, which means that we can write it as a matrix equation $\Gamma(t + \Delta t) = \mathbf{M}\Gamma(t)$. The Lyapunov exponents are then the eigenvalues of the matrix \mathbf{M} .

For very short times, we can relate $\Gamma(t + \Delta t)$ and $\Gamma(t)$ use a Taylor expansion to linear order in Δt .

1. Use your knowledge of Hamiltonian dynamics to show that for an isolated system the sum of all Lyapunov exponents vanishes.
2. Show, using Eq. (4.3.20), that the sum of the Lyapunov exponents also vanishes for the Verlet algorithm with arbitrary Δt .
3. Give a geometric interpretation of this property of the Verlet algorithm.

Chapter 5

Computer experiments

In the preceding chapters, we described the basics of [Monte Carlo](#) and [Molecular Dynamics](#) simulations. With these techniques, we can sample the equilibrium configurations of a classical many-body system, and in the case of MD simulations, we can follow its time evolution. But this is only the first step. The most common aims of [Molecular Dynamics](#) are to predict observable properties or to test theoretical predictions. In other words, we use our simulations to perform *measurements*. Measurements in a computer simulation are in many ways similar to experiments in physical systems: we have to prepare a sample, we have to decide on the best measuring technique, we have to accumulate enough data, and we should analyze the effect of possible systematic and statistical errors in our results. For these reasons, we use the word *measurement* to refer to the calculation of observable properties, mainly for lack of a better word. Whenever confusion might arise, we will refer to measurements on real physical systems as *physical measurements*, as opposed to *numerical measurements*.

There is an important distinction between physical and numerical measurements. In physical measurements, we register how a physical probe responds when brought in contact with a many-body system. Examples of such probes are pressure-gauges, thermometers, or beams of photons or neutrons. In contrast, in simulations, we have to deduce the value of observable properties from our knowledge of the coordinates and momenta of all the particles in the system. Sometimes it is obvious how the value of the relevant macroscopic observable follows from our knowledge at the microscopic level. For instance, the average flow velocity in a one-component system is simply the average velocity of all particles. However, in most cases, the relation between the macroscopic observable and the raw simulation data is more subtle. To give an example, in experiments, the pressure is often determined by measuring the displacement of a pressure gauge under the influence of the force exerted on it by the molecules in the sample. But in simulations, we usually wish to determine the pressure from the properties of a bulk system subject to periodic boundary conditions. Introducing a pressure gauge in a simulation would be a bad idea because systems with interfaces exhibit large finite-size effects. Fortunately, Statistical Mechanics allows us to relate the observable properties of a macroscopic system to the microscopic information about coordinates and momenta, which we obtain during a simulation. Below, we discuss some of the more important of these relations.

In what follows, we distinguish between *static* and *dynamical* properties. Static properties can be computed by sampling the equilibrium configurations on a many-body system, which can be done by both MC and MD simulations. Typical examples of static properties that can be sampled in a simulation are the quantities that appear in thermodynamic relations, such as internal energy, temperature, pressure, and heat capacity, but not quantities such as entropy or free energy that cannot be expressed as an ensemble average. We will discuss free-energy-related properties separately in Chapter 8.

Dynamical properties describe how a system evolves in time in response to an external perturbation. Examples are shear flow, heat flow, or diffusion. Of course, when a system has been subjected to an external perturbation, it is brought out of equilibrium. However, as discussed in section 2.5.2, we can compute the dynamical response of a system to a weak external perturbation by studying the decay of fluctuations in equilibrium. This linear-response theory discussed allows us to derive expressions for transport coefficients in terms of time-correlation functions of microscopic fluxes that can be expressed explicitly in terms of the coordinates and momenta of the particles in the system—and that is precisely what we need in simulations.

In what follows, we first discuss the statistical mechanical basis of techniques for computing static properties of a system by MC or MD. After that, we discuss the numerical measurement of transport properties.

Planck's constant in classical simulations

The simulation techniques described in this book are all based on classical statistical mechanics. As a consequence, none of the computed measurable quantities can depend on the value of Planck's constant \hbar . Some quantities (for instance, the chemical potential, discussed in section 8.5) seem to depend on \hbar through the thermal De Broglie wavelength, Δ . However, in that case, Planck's constant has been inserted by hand to achieve consistency with known quantum results. Changing the value of \hbar would result in an overall shift of the chemical potential, which does not affect a single measurable property of a purely classical system. In reality, the separation between the classical and the quantum world is not clean. Many molecules have internal vibrations (or even rotations), with level spacings that are not small compared to the thermal energy. In that case, a choice has to be made: if the excitation energies are much larger than $k_B T$, these modes will be mostly in their ground state, and it is reasonable to ignore them. However, in many cases, there is some degree of coupling between the quantum degrees of freedom and the classical bath. One consequence of the quantum nature of the internal degrees of freedom of a molecule is that the thermal contribution to its average internal energy is lower than the corresponding classical value. Many molecular properties are barely affected by the quantized internal degrees of freedom, but quantum effects can have a large effect on quantities such as heat capacity and thermal conductivity. This problem cannot be resolved within the context of purely classical simulations.

5.1 Static properties

The first step in a simulation study is usually to characterize the thermodynamic state of the model system. That is: we wish to determine the relation between “control parameters”, such as the temperature, pressure, or applied electrical/magnetic field (i.e., all *intensive* quantities), and the resulting state functions, such as energy, volume or polarization (which are all extensive quantities). Depending on the nature of the simulation, some of these quantities may have been imposed, e.g., the temperature in constant-*NVT* MC simulations, or the energy in constant-*NVE* simulations. In Chapters 6 and 7, we will discuss techniques to carry out simulations in other ensembles than *NVT* for MC and *NVE* for MD. For instance, we can carry out MD simulations at constant *NPT*. In that case, the primary thermodynamic quantities to measure in a simulation would be the internal energy E and the average volume V . However, even when we impose P and T , it is still useful to measure the temperature and the pressure to verify that these quantities are indeed equal to the imposed values. Such measurements provide a powerful diagnostic tool.

5.1.1 Temperature

As was briefly discussed in Chapter 4, the temperature is often measured by computing the average kinetic energy per degree of freedom. For a system with f degrees of freedom, the temperature T is given by

$$k_B T = \frac{\langle 2\mathcal{K} \rangle}{f}. \quad (5.1.1)$$

The number of degrees of freedom of an N -particle system is equal to $Nd - N_c$, where d is the dimensionality, and N_c is the number of constraints (e.g., bond lengths) or conserved quantities (e.g., momentum or energy). In a system with periodic boundary conditions, angular momentum is not a conserved quantity, but in the absence of external forces, the total momentum is conserved. It is important to distinguish N from the number of molecules in a system: if a system consists of M molecules, each comprising m atoms (or groups described as atoms), then $N = mM$. For a system without hard constraints, f is equal to $Nd - (d + 1)$, which is usually close to Nd , but not equal.

Discretization errors

The above description is an oversimplification for MD simulations, where we necessarily use discrete time steps. The main reason is that, as pointed out in refs. [114,115], the relation between the velocity, as generated by the velocity-Verlet algorithm, and the momentum, is not simply $v_i = p_i/m_i$. Rather, the “true” velocity is defined through Hamilton’s equations of motion $\dot{r}_i^{\text{true}} \equiv \partial\mathcal{H}/\partial p_i$. For simulations with a finite time step, the Hamiltonian is not con-

served, but rather the “shadow”-Hamiltonian \mathcal{H}_S (see Eq. (4.3.22)).¹ It is this Hamiltonian that determines the density in phase space, and that should enter in the Boltzmann factor. We can then use

$$\left\langle p_i \left(\frac{\partial \mathcal{H}_S}{\partial p_i} \right) \right\rangle_{NVT} = \langle p_i \dot{r}_i^{\text{true}} \rangle_{NVT} k_B T. \quad (5.1.2)$$

But, for the shadow Hamiltonian

$$\dot{r}_i^{\text{true}} \neq (p_i/m_i). \quad (5.1.3)$$

Rather, to compute the correct temperature, we have to estimate the true velocities \dot{r}_i^{true} by interpolation from several consecutive positions [115]. Then $\langle p_i \dot{r}_i^{\text{true}} \rangle = k_B T$.

In most simple MD codes, this procedure is not followed. However, the effect of using the wrong velocities can lead to a serious discrepancy between the translational and vibrational temperature, when estimating the temperature of a system of molecules that have high-frequency internal motions, e.g., proteins in solution [115] (see Illustration 1).

5.1.2 Internal energy

Measuring the internal energy E of a system is usually straightforward, as it follows from our knowledge of the Hamiltonian of the system. However, there may be exceptions, in particular, if the interaction between particles is described by an effective, temperature-dependent potential, which is not really a potential but a free-energy (an example is the depletion interaction [128]). In that case, the energy follows from $E = (\partial\beta F / \partial\beta)$, which vanishes for depletion forces.

5.1.3 Partial molar quantities

Many applications of **Molecular Dynamics** focus on the study of mixtures. In those situations, we often need to know how the extensive properties of a system (e.g., internal energy, enthalpy, volume) change with the composition of the mixture. Let us take the enthalpy H , of an m -component mixture as an example. The partial molar enthalpy h of species α is defined as

$$h_\alpha \equiv \left(\frac{\partial H}{\partial N_\alpha} \right)_{P,T,\{N_\beta \neq \alpha\}}.$$

The total enthalpy of a mixture can be written as

$$H = E + PV = \sum_{\alpha=1}^m h_\alpha N_\alpha.$$

¹ Strictly speaking, this conservation law only holds for potentials that are infinitely differentiable. But the present discussion appears to apply, even when this is not strictly the case.

Naively, one might think that we can compute the partial molar enthalpy of a species by computing the average energy and volume per particle of species α . This, however, is not correct (or, more precisely, it is usually ill-defined). Hence, somewhat surprisingly, special techniques are required to compute partial molar quantities, even when computing the total quantity (such as H) is straightforward. The simplest (but not the cheapest) approach is to compute H for two systems that differ only in N_α . That method works —even for solid mixtures, where other techniques often fail. However, for liquids and dense gases, more efficient techniques are available [129].

5.1.4 Heat capacity

It is a bit misleading to suggest that the internal energy of a system can be measured experimentally. Thermodynamic experiments only determine the internal energy up to an additive constant. However, what experiments *can* measure is the variation of the internal energy with temperature or pressure.

As an example, we focus on the C_V , the heat capacity of a system at constant N and V :

$$C_V \equiv \left(\frac{\partial E}{\partial T} \right)_{N,V} . \quad (5.1.4)$$

Clearly, if we measure E in a simulation (up to a constant), then we can obtain C_V by carrying out simulations over a range of temperatures and estimating C_V by numerical differentiation. However, we can also determine C_V at a fixed temperature, by studying the spontaneous fluctuations in the internal energy. We start with the expression

$$C_V = \left(\frac{\partial E}{\partial T} \right)_{NV} = \left(\frac{\partial E}{\partial \beta} \right)_{NV} \left(\frac{d\beta}{dT} \right) = \frac{-1}{k_B T^2} \left(\frac{\partial E}{\partial \beta} \right)_{NV} . \quad (5.1.5)$$

Next, using Eqs. (2.2.13) and (2.2.14), we can write

$$\left(\frac{\partial E}{\partial \beta} \right)_{NV} = - \left(\frac{\partial^2 \ln Q(N, V, T)}{\partial \beta^2} \right)_{NV} = - \left[\langle E^2 \rangle - \langle E \rangle^2 \right] , \quad (5.1.6)$$

and hence

$$C_V = \frac{1}{k_B T^2} \left[\langle E^2 \rangle - \langle E \rangle^2 \right] . \quad (5.1.7)$$

We note that Eq. (5.1.7) is an example of a static fluctuation expression of the type discussed in section 2.5.2.

As C_V is related to the fluctuations in the energy at a constant temperature, it would seem that we cannot use a fluctuation expression to determine C_V from an MD simulation at constant N , V , and E . However, Lebowitz et al. [106] showed that we can determine C_V at constant N , V , E by measuring the fluctuations in

the *kinetic* energy, \mathcal{K} :

$$\langle \mathcal{K}^2 \rangle_{NVE} - \langle \mathcal{K} \rangle_{NVE}^2 = \frac{3Nk_B^2 T^2}{2} \left(1 - \frac{3Nk_B}{2C_V} \right). \quad (5.1.8)$$

For a more detailed discussion, see ref. [21].

Eq. (5.1.7) suggests how similar expressions can be derived for the heat capacity at constant pressure, C_p :

$$C_P = \left(\frac{\partial H}{\partial T} \right)_{NP}, \quad (5.1.9)$$

the isothermal compressibility and a number of other “susceptibilities”, i.e., quantities that describe that variation of an extensive thermodynamic quantity with an intensive variable. In Appendix F.4 discusses how, for instance, the elastic constants of a solid can be obtained from a fluctuation expression.

5.1.5 Pressure

One of the most important thermodynamic observables of a many-body system is its pressure. In experiments, the operational definition of the pressure of a system is the average force exerted by the system on a unit area of the wall of the confining container. The above definition is less attractive in simulations because the introduction of a physical wall in a system would result in large finite-size effects in most observables. For this reason, expressions for the pressure, P , in a periodic system are often derived starting from the thermodynamic relation (Eq. (2.1.35) in Chapter 2):

$$P = - \left(\frac{\partial F}{\partial V} \right)_{N,T}. \quad (5.1.10)$$

From Eq. (2.3.6), we know that

$$F = -k_B T \ln Q(N, V, T) = c(N, T) - k_B T \ln \int_V \cdots \int_V d\mathbf{r}^N \exp \left[-\beta U(\mathbf{r}^N) \right], \quad (5.1.11)$$

in which $c(N, T)$ will not contribute to the pressure as it is independent of V .

Differentiating with respect to volume is a bit tricky, as the limits of the configurational integral depend on the volume of the system. This complication can be addressed by defining *scaled* coordinates \mathbf{s} through

$$\mathbf{s}_i \equiv \frac{\mathbf{r}_i}{L} \quad \text{for } i = 1, 2, \dots, N, \quad (5.1.12)$$

where L is the diameter of the box. To simplify the notation, we have assumed that the periodically repeated box is cubical in shape, such that $V = L^3$.

The integral over \mathbf{s} ranges from 0 to 1, independent of V . Then

$$\int_V \cdots \int_V d\mathbf{r}^N \exp[-\beta U(\mathbf{r}^N)] = V^N \int_0^1 \cdots \int_0^1 ds^N \exp[-\beta U(s^N; L)]. \quad (5.1.13)$$

$U(s^N; L)$ depends on L , because if we change L while keeping all \mathbf{s}_i constant, all real distances are changed. For the sake of compactness, we will replace $\int \cdots \int$ with a single integral sign. We can then write

$$P = k_B T \left(\frac{\partial \left\{ \ln V^N + \ln \int_0^1 ds^N \exp[-\beta U(s^N; L)] \right\}}{\partial V} \right)_{N,T}, \quad (5.1.14)$$

where we have used the fact that $c(N, T)$ is independent of V . The first term on the right of Eq. (5.1.14) yields the ideal-gas pressure $Nk_B T/V$. The second term describes the excess pressure due to intermolecular interactions:

$$\begin{aligned} P_{\text{exc}} &= k_B T \left(\frac{\partial \left\{ \ln \int_0^1 ds^N \exp[-\beta U(s^N; L)] \right\}}{\partial V} \right)_{N,T} \\ &= - \left\langle \left(\frac{\partial U(s^N; L)}{\partial V} \right) \right\rangle_{N,T}. \end{aligned} \quad (5.1.15)$$

To proceed, we note that the potential energy U depends on the volume of the system because the real positions ($\mathbf{r}_i = \mathbf{s}_i L$) scale with L . But in some cases, the potential energy may also have a contribution that depends on V , but not on the particle coordinates [130]. This happens, for instance, when the potential energy contains a density-dependent term that does not depend on the coordinates of the particles within a periodic box, but on the distance between the centers of different periodic boxes. Using the chain rule, we can then write

$$\begin{aligned} P_{\text{exc}} &= - \left\langle \sum_{i=1}^N \left(\frac{\partial U}{\partial \mathbf{r}_i} \right) \frac{\partial \mathbf{r}_i}{\partial V} + \left(\frac{\partial U}{\partial V} \right)_{\mathbf{r}_i} \right\rangle_{N,T} \\ &= \frac{1}{dV} \left\langle \sum_{i=1}^N \mathbf{F}_i \cdot \mathbf{r}_i \right\rangle_{N,T} - \left\langle \left(\frac{\partial U}{\partial V} \right)_{\mathbf{r}_i} \right\rangle_{N,T}, \end{aligned} \quad (5.1.16)$$

where d is the dimensionality, and we have used the fact that

$$\left(\frac{\partial \mathbf{r}_i}{\partial V} \right)_{\mathbf{s}_i} = \frac{\mathbf{r}_i}{dV}. \quad (5.1.17)$$

Eq. (5.1.16) is often written as

$$P_{\text{exc}} = \frac{1}{dV} \langle W \rangle, \quad (5.1.18)$$

which defines the *virial* W :

$$W \equiv \left\langle \sum_{i=1}^N \mathbf{F}_i \cdot \mathbf{r}_i \right\rangle_{N,T} - dV \left\langle \left(\frac{\partial U}{\partial V} \right)_{\mathbf{s}_i} \right\rangle_{N,T}. \quad (5.1.19)$$

At first sight, Eq. (5.1.16) looks less suited for systems with periodic boundary conditions, as it seems to depend on where (in which replica box) we choose \mathbf{r}_i . However, this is not much of a problem, as the total force on the particles in the periodically repeated box is zero. Thompson and Plimpton [131] have generalized Eq. (5.1.16) to the case where the potential energy can be expressed as a sum of group contributions (not necessarily pairwise additive), and where the sums of the intra-group forces vanish, as in the case of pair potentials where $f_{ij} + f_{ji} = 0$. The more general decomposition can be achieved if the potential can be written as a sum of n -body terms —although it excludes the case of forces derived *on the fly* from quantum calculations. We refer the reader to [131] for details about the group method. In Chapter 8, we discuss how, using a free-energy perturbation expression (8.6.11), we can compute the pressure of a system with many-body interactions that cannot be decomposed.

In simulations, we often use potential-energy functions that can be written as a sum of pairwise contributions:

$$U(\mathbf{r}^N) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N u(r_{ij}).$$

In that case, we can write:

$$\begin{aligned} P_{\text{exc}} &= -\frac{1}{dV} \left\langle \sum_{i=1}^N \sum_{j \neq i} \left(\frac{\partial u(r_{ij})}{\partial \mathbf{r}_i} \right) \cdot \mathbf{r}_i \right\rangle_{N,T} \\ &= \frac{1}{dV} \left\langle \sum_{i=1}^N \sum_{j \neq i} \mathbf{f}(r_{ij}) \cdot \mathbf{r}_i \right\rangle_{N,T}. \end{aligned} \quad (5.1.20)$$

We now use the fact that for pair potentials, $\mathbf{f}(r_{ij}) = -\mathbf{f}(r_{ji})$, and that i and j are dummy indices:

$$\begin{aligned} P &= \rho k_B T + \frac{1}{2dV} \left\langle \sum_{i=1}^N \sum_{j \neq i} \mathbf{f}(r_{ij}) \cdot \mathbf{r}_i + \sum_{j=1}^N \sum_{i \neq j} \mathbf{f}(r_{ji}) \cdot \mathbf{r}_j \right\rangle_{N,T} \\ &= \rho k_B T + \frac{1}{2dV} \left\langle \sum_{i=1}^N \sum_{j \neq i} \mathbf{f}(r_{ij}) \cdot \mathbf{r}_{ij} \right\rangle_{N,T} \end{aligned}$$

$$= \rho k_B T + \frac{\rho^2}{2d} \int d\mathbf{r} g(r) r f(r) . \quad (5.1.21)$$

The decomposition of the potential energy into group terms may affect what part of the pressure in Eq. (5.1.16) is considered to be explicitly volume dependent. To illustrate this point, consider a perfect atomic crystal at $T = 0$, and at a finite pressure P . At $T = 0$, the system is at a minimum of the potential energy, and hence the net forces \mathbf{F}_i on all particles i vanish. According to Eq. (5.1.16), the pressure is then exclusively due to the explicit volume dependence of U . However, if, in the spirit of ref. [131], we decompose the net forces into pair forces, then we find that $\sum_{i,j>i} \mathbf{f}_{ij} \cdot \mathbf{r}_{ij}$ does not vanish and Eq. (5.1.21) gives a correct description of the pressure.

For molecular systems, we have different choices to compute the virial: one is based on the forces between atoms (or, more precisely, force centers), and the other is based on forces between the molecular centers of mass. For the average value of the virial, the choice makes no difference. However, it does make a difference for the statistical error, in particular, if we describe intra-molecular forces by stiff spring constants. The reason is that the mean-squared fluctuations in such forces may be very large, even if the average vanishes.²

5.1.5.1 Pressure by thermodynamic integration

There are cases where Eq. (5.1.14) cannot be used, for instance, for lattice models, where the volume is not a continuous variable. In such cases, we can compute the pressure of a fluid using the thermodynamic relation

$$d(PV)_{VT} = N d\mu . \quad (5.1.22)$$

Methods to perform simulations under conditions where μ , V , and T are the control variables, are discussed in section 6.5.

5.1.5.2 Local pressure and method of planes

Eq. (5.1.14) yields a *global* expression for the pressure. Even though Eq. (5.1.21) suggests that, for pairwise additive interactions, the pressure could be decomposed into contributions due to individual particles, it would be wrong to interpret these contributions as *local* pressures. The mechanical definition of the pressure does have a local meaning as the force acting per unit area on a plane (say, at position x) in the system. We can have different choices for x , and hence these could yield different pressures. However, for a system in mechanical equilibrium, the average pressure should not depend on x , otherwise, there would be a net force acting on a volume element bounded by planes at $x + \Delta x$ and at x .

² However, the strong fluctuations of the intra-molecular forces have little effect on the accuracy of Green-Kubo integrals (see sections 2.5.2 and 5.3.2).

If we would take the local virial pressure, for instance, near a hard wall at $x = 0$, we would find that this measure of the pressure is not constant: hence its gradient is not related to a mechanical force.

But we can compute the mechanical pressure directly. Let us consider a fictitious plane at x . We can then compute the force on that plane as the average momentum transfer through that plane due to all particles on the left (say) of that plane. This force has two contributions: 1) momentum transfer due to particles that carry their own momentum, exerting a net force $\rho(x)k_B T$ and 2) the force due to the fact that the particles on the left of the dividing plane interact with particles on the right-hand side (note: the choice of “left” or “right” is immaterial). We can compute this force for any plane (and for any potential, even a many-body potential). However, for pairwise additive potentials, the expression simplifies because we can write the force acting through a plane as the sum of all pair forces $f_x(r_{ij})$ for which $x_i < x$ and $x_j > x$. This method of computing the pressure is commonly referred to as the “method of planes” [132]. By construction, the mechanical force thus obtained is independent of x for a system in mechanical equilibrium.

5.1.5.3 Virtual volume changes

For non-pairwise-additive interactions we cannot use the standard virial route to compute the pressure. For such systems—but also for systems of non-spherical hard-core particles, for which the virial approach becomes rather cumbersome—it may be attractive to compute the pressure by using a finite-difference version of Eq. (2.1.35):

$$P \approx -(\Delta F / \Delta V)_{NT}. \quad (5.1.23)$$

To this end, we must compute the free-energy difference between a system contained in volume V and the same system contained in a volume $V' = V + \Delta V$, where ΔV must be chosen sufficiently small that ΔF is linear in ΔV . As ΔV is small, we can use a perturbation expression for the free energy (see Eq. (8.6.10)) to compute ΔF :

$$\begin{aligned} -\frac{\Delta F}{\Delta V} &= \frac{kT}{\Delta V} \ln \frac{Q(N, V', T)}{Q(N, V, T)} \\ &= \frac{kT}{\Delta V} \ln \left(\frac{V'^N \int d\mathbf{s}^N \exp(-\beta U(\mathbf{s}^N; V'))}{V^N \int d\mathbf{s}^N \exp(-\beta U(\mathbf{s}^N; V))} \right), \end{aligned} \quad (5.1.24)$$

or:

$$P = P_{id} - \lim_{\Delta V \rightarrow 0} \frac{kT}{\Delta V} \ln \left\langle \exp \left(-\beta \Delta U(\mathbf{s}^N) \right) \right\rangle, \quad (5.1.25)$$

where $\Delta U \equiv U(\mathbf{s}^N; V) - U(\mathbf{s}^N; V')$ and P_{id} is the ideal gas pressure. For systems with continuous potential-energy functions, ΔV can be chosen both

positive and negative. For hard-core systems, the situation may be a bit more tricky because (for spherical particles) ΔU is always zero upon expansion. In such cases, one should use $\Delta V < 0$. However, for sufficiently non-spherical particles even volume expansion may occasionally lead to overlaps. In that case, the results for simulations with positive and negative ΔV should be combined as will be explained in section 8.6.3.

In practice, the virtual-volume-move approach can be made much more efficient by decomposing the free-energy changes in contributions due to individual particles. Such an approach becomes rigorous in the limit $\Delta V \rightarrow 0$ see [133].

5.1.5.4 Compressibility

Once we have computed the pressure of a system as a function of density, we can obtain the isothermal compressibility β_T from

$$\beta_T \equiv -\frac{1}{V} \left(\frac{\partial V}{\partial P} \right)_{N,T} = \frac{1}{\rho} \left(\frac{\partial \rho}{\partial P} \right)_{N,T}. \quad (5.1.26)$$

However, as in the case of the heat capacity, we can use a fluctuation expression to estimate the compressibility from a simulation at constant pressure at a single state point. We use the fact that

$$\langle V \rangle_{N,P,T} = -k_B T \left(\frac{\partial \ln Q(N, P, T)}{\partial P} \right). \quad (5.1.27)$$

It then follows that

$$\beta_T = \frac{-1}{V} \left(\frac{\partial V}{\partial P} \right)_{N,T} = \frac{\langle V^2 \rangle - \langle V \rangle^2}{\langle V \rangle k_B T}. \quad (5.1.28)$$

Similar expressions exist for the elastic constants of solids (see section F.4).

5.1.6 Surface tension

Up to this point, we have been discussing how simulations can be used to estimate the bulk properties of materials, where it is useful to employ periodic boundary conditions as these minimize the finite-size effects associated with the presence of surfaces. However, the properties of surfaces are interesting in their own right. Here we discuss the calculation of one key surface property, namely the surface tension γ , which measures the free-energy cost associated with a change, at constant N , V , and T , of the area of a flat, unstructured surface or interface. We focus initially on unstructured interfaces because, as we shall see later, computing the free energy of structured interfaces (e.g., a crystal-liquid interface) requires different methods.

We start with the expression for the variation of the Helmholtz free energy of a one-component system with N , V , T , and with the surface area A :

$$dF = -SdT - PdV + \mu dN + \gamma dA, \quad (5.1.29)$$

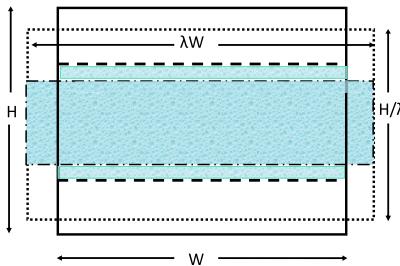


FIGURE 5.1 The figure with the solid frame shows a (periodically repeated) box with height H and width W , containing two phases separated by flat interfaces (dashed lines). The original box is then deformed such that the volume remains constant. This is achieved by scaling W by a factor λ , and H by a factor λ^{-1} . The boundary of the deformed box is indicated by dotted lines. Due to this transformation, the surface area separating the two phases changes from $2S$ to $2\lambda S$ (there are two surfaces).

and hence,

$$\gamma \equiv \left(\frac{\partial F}{\partial A} \right)_{N,V,T}. \quad (5.1.30)$$

We consider a periodically repeated system containing two phases in parallel slabs (see Fig. 5.1). We assume that the surface is perpendicular to the z -direction, and consider the effect of stretching the surface in the x -direction by a factor λ : the new surface area A' is then related to the original surface area by $A' = \lambda A$. Note, however, that the system contains two interfaces, hence the total surface area $A = 2S$, where S is the area per interface. The height of the box in the z -direction is scaled by a factor $1/\lambda$, such that the volume of the box remains unchanged. Due to this transformation, all x -coordinates in the system are scaled by a factor λ , and all z -coordinates are scaled by a factor λ^{-1} . We can then use the statistical mechanical expression for the Helmholtz free energy to arrive at an expression for the surface tension. In analogy with Eq. (5.1.15) we write³

$$\begin{aligned} \gamma &= \left(\frac{\partial F}{\partial A} \right)_{N,V,T} \\ &= -k_B T \left(\frac{\partial \left\{ \ln \int_0^1 ds^N \exp[-\beta U(s^N; \lambda W, H/\lambda)] \right\}}{\partial A} \right)_{N,V,T} \\ &= \frac{1}{2S} \left(\left(\frac{\partial U(s^N; \lambda W, H/\lambda)}{\partial \lambda} \right) \right)_{N,V,T}. \end{aligned} \quad (5.1.31)$$

³ The coordinate transformation due to scaling will also change the momenta (see Appendix F.4), however for a volume conserving transformation, rescaling of the momenta does not change Eq. (5.1.31).

We now focus on continuous pairwise-additive potentials.⁴ For continuous potentials, we can write

$$\begin{aligned} \left(\frac{\partial U(\mathbf{s}^N; \lambda W, H/\lambda)}{\partial \lambda} \right)_{\lambda=1} &= \sum_{i=1}^N \left[\left(\frac{\partial U(\mathbf{r}^N)}{\partial x_i} \right) x_i - \left(\frac{\partial U(\mathbf{r}^N)}{\partial z_i} \right) z_i \right] \\ &= - \sum_{i=1}^N [f_{i;x} x_i - f_{i;z} z_i], \end{aligned} \quad (5.1.32)$$

where $f_{i;\alpha}$ denotes the force on particle i in the α direction. For pairwise additive potentials, we can write $f_{i;\alpha} = \sum_{j \neq i} f_{ij;\alpha}$, where $f_{ij;\alpha}$ is the pair-force between particles i and j in the α -direction. As in Eq. (5.1.21), we now use the fact that i and j are dummy indices that can be permuted.

$$\left(\frac{\partial U(\mathbf{s}^N; \lambda W, H/\lambda)}{\partial \lambda} \right)_{\lambda=1} = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i} [f_{ij;z} z_{ij} - f_{ij;x} x_{ij}], \quad (5.1.33)$$

and hence

$$\gamma = \frac{1}{4S} \left\langle \sum_{i=1}^N \sum_{j \neq i} [f_{ij;z} z_{ij} - f_{ij;x} x_{ij}] \right\rangle. \quad (5.1.34)$$

It would seem that there is something wrong with Eq. (5.1.34) because the number of particles in the numerator scales with the volume V , whereas the denominator scales with the surface area S . In fact, there is no problem, because the environment of a particle (say i) far away from the surface is isotropic, then

$$\left\langle \sum_{j \neq i} f_{ij;z} z_{ij} \right\rangle = \left\langle \sum_{j \neq i} f_{ij;x} x_{ij} \right\rangle. \quad (5.1.35)$$

The net result is that pairs ij that are in the bulk of the liquid do not contribute to the surface tension. In a simulation, it is advisable not to include such pairs in the sum in Eq. (5.1.34), because they would contribute to the statistical noise, but not to the average. The derivation above is just one route to computing the surface tension. Other approaches are described in refs. [134,135] —see also SI L.2. However, as was shown by Schofield and Henderson [136], the most commonly used expressions are equivalent.

Surface tension from virtual moves

In complete analogy with the direct approach to measure the pressure by performing virtual volume moves (Section 5.1.5.3), we can also compute the surface tension by considering (say) a vertical slab of liquid in the system sketched

⁴ The expression for particles with discontinuous interactions can be recovered from the continuous case by replacing the force acting on a particle with the average momentum transfer per unit of time.

in Fig. 5.1. Just as in Eq. (5.1.25), we can compute the free-energy change due to a change in the surface area at constant total volume. The finite-difference form of Eq. (5.1.31), is usually referred to as the “test-area method”. This method remains valid when estimating the surface tension of a system with arbitrary non-pairwise-additive interactions [134,137]. For flat fluid-fluid interfaces, the test-area method remains correct for finite virtual-area changes, because the surface tension is independent of the area. In practice, large test area changes are not advisable if the energy changes in forward and backward test-area moves do not overlap (see section 8.6.1). An illustration of the problems caused by non-overlapping distributions in the test-area method can be found in ref. [138]).

Surface free-energy density and surface stress

In the previous section, we considered the surface tension of a flat liquid interface, or for that matter, the surface tension of a liquid at a perfectly flat solid wall. The expression for γ derived above makes use of the fact that we can change the surface area of a liquid by an infinitesimal amount, without changing the bulk properties. Such an approach will not work if any of the two phases is a solid, because as we stretch the surface of a solid, we change its interfacial free energy.

For solids, we can still write the contribution of the surface to the free energy as $F_s = \gamma A$, where γ is now called the surface free-energy density. But now we cannot use Eq. (5.1.31) to compute γ , because

$$\left(\frac{\partial F_s}{\partial A} \right) = \gamma + A \left(\frac{\partial \gamma}{\partial A} \right) \equiv t_s , \quad (5.1.36)$$

where we have introduced the surface stress t_s .⁵ For liquids, γ does not depend on A , and hence $\gamma = t_s$, but this equality does not hold for solids⁶: special free-energy techniques (as discussed in section 8.4.2) are needed to compute γ for solid interfaces [141], however to compute the free-energy *change* upon bringing a solid and a liquid into contact, one can use the relatively straightforward thermodynamic-integration technique proposed by Leroy and Müller-Plathe [142].

Free energy of curved surfaces

In general, the surface tension of an interface will depend on its curvature. Curvature effects become important when at least one of the radii of curvature of the surface is not much larger than a typical molecular diameter.

In contrast to the case of a flat interface, the value of the surface tension of a curved surface depends on our choice of the location of the surface. These and

⁵ Of course, deforming the surface of a solid may also change its bulk elastic free energy, but that effect can be computed separately.

⁶ The distinction has observable consequences. For instance, the Laplace pressure inside a small crystal is not determined by γ but by t_s , which can be negative [139,140].

other features of curved surfaces imply that computing the free energy of curved surfaces is subtle and full of pitfalls. We will not discuss this topic, but refer the reader to ref. [143] for further background information.

5.1.7 Structural properties

Thus far we discussed the measurement of thermodynamic observables. However, many experiments provide information about the microscopic structure of a system. Although some experiments, such as confocal microscopy, can provide an instantaneous snapshot of the configuration of a system, most experiments yield information about some averaged descriptors of the local structure in a system. Scattering experiments (X-ray, neutron) yield information about the mean-squared value of the Fourier transform of the scattering density, whereas real-space experiments such as confocal microscopy can be used to obtain information about the averaged local density profile around a selected particle. As we discuss below, the two quantities are related.

5.1.7.1 Structure factor

Static scattering experiments usually probe the angle-dependence of the intensity of radiation scattered by the sample. The scattered intensity is proportional to the mean-squared value of the scattering amplitude $A(\mathbf{q})$, where \mathbf{q} denotes the scattering wave-vector; for instance, for monochromatic X-rays with wavelength λ_0 : $\mathbf{q} = (4\pi/\lambda_0) \sin(\theta/2)$. The instantaneous scattering amplitude depends on the configuration of the system and is typically of the form

$$A(\mathbf{q}) \sim \sum_{i=1}^N b_i(q) e^{i\mathbf{q}\cdot\mathbf{r}_i}, \quad (5.1.37)$$

where $b_i(q)$ is the scattering amplitude of particle i . The $b_i(q)$ depends on the internal structure of the particles. We note that if $b(q)$ is a known function of q , simulations can be used to predict the scattering intensity. Often the data of scattering experiments are analyzed to yield information about the so-called structure factor $S(q)$, which is equal to $1/N$ times mean-squared fluctuation in the amplitude of $\rho(\mathbf{q})$, the Fourier transform of the single-particle density. $\rho(\mathbf{q})$ is equal to

$$\rho(\mathbf{q}) = \sum_{i=1}^N e^{i\mathbf{q}\cdot\mathbf{r}_i} = \int_V d\mathbf{r} \rho(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}}, \quad (5.1.38)$$

where the real-space single-particle density $\rho(\mathbf{r})$ is defined by

$$\rho(\mathbf{r}) \equiv \sum_{i=1}^N \delta(\mathbf{r} - \mathbf{r}_i). \quad (5.1.39)$$

With this definition, we can write

$$\begin{aligned} S(\mathbf{q}) &= \frac{1}{N} \left[\langle |\rho(\mathbf{q})|^2 \rangle - |\langle \rho(\mathbf{q}) \rangle|^2 \right] \\ &= \frac{1}{N} \int_V \int_V d\mathbf{r} d\mathbf{r}' \left[\langle \rho(\mathbf{r}) \rho(\mathbf{r}') \rangle - \langle \rho \rangle^2 \right] e^{i\mathbf{q} \cdot (\mathbf{r} - \mathbf{r}')} . \end{aligned} \quad (5.1.40)$$

The quantity $\langle \rho(\mathbf{r}) \rho(\mathbf{r}') \rangle$ is the density correlation function. It is often written as

$$\langle \rho(\mathbf{r}) \rho(\mathbf{r}') \rangle = \langle \rho(\mathbf{r}) \rangle \langle \rho(\mathbf{r}') \rangle g(\mathbf{r}, \mathbf{r}') . \quad (5.1.41)$$

Eq. (5.1.41) defines the pair distribution function $g(\mathbf{r}, \mathbf{r}')$. In an isotropic, homogeneous liquid $\langle \rho(\mathbf{r}) \rangle$ is constant and equal to the average density ρ , and $g(\mathbf{r}, \mathbf{r}')$ only depends on the scalar distance $r \equiv |\mathbf{r} - \mathbf{r}'|$. $g(r)$ is called the radial distribution function: it probes how the local density around a particle in a classical fluid is decreased/enhanced due to the intermolecular interactions. $g(r)$ plays a key role in liquid-state theory. In the next section, we discuss how $g(r)$ can be measured in a simulation.

As $S(q)$ is related to $g(r)$, $g(r)$ can be obtained from $S(q)$ by inverse Fourier transform. This might seem to be a needlessly complicated route for obtaining $g(r)$. However, a naive calculation of $g(r)$ requires $\mathcal{O}(N^2)$ operations, whereas the computational effort required to compute $S(q)$ by Fast-Fourier transform, scales as $N \ln N$.

It might seem straightforward to obtain $S(q)$ of a liquid from $g(r)$ using

$$S(q) = \rho \int_V d\mathbf{r} [g(r) - 1] e^{i\mathbf{q} \cdot \mathbf{r}} . \quad (5.1.42)$$

However, in simulations, this procedure is tricky. The reason is that $g(r)$ is usually computed up to a spherical cut-off distance $r_{\max} = L/2$, where L is the diameter of the simulation box. But often $r^2(g(r) - 1)$ has not yet decayed to zero at r_{\max} . In that case, spherical truncation of the integral can give rise to unphysical behavior of the apparent $S(q)$ —for instance, it may show oscillations and even negative values at small q values. For this reason, it is safer to compute $S(q)$ using Eq. (5.1.40). Computationally, this is not a big problem because Fast Fourier Transforms are ... fast [38].

5.1.7.2 Radial distribution function

Computing the radial distribution function is probably one of the very first measurements that a novice in simulations will perform because it is such a simple calculation. For a given instantaneous configuration, we can easily compute all $N(N - 1)/2$ pair distances between the particles in the system. We can then make a histogram of the number of pairs with a distance between r and $r + \Delta$. Choosing the bin-width Δ is a compromise between resolution (favoring a small value of Δ) and statistical accuracy (the relative error in $g(r)$)

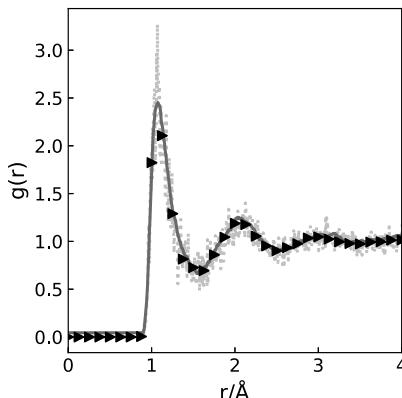


FIGURE 5.2 The figure shows three different calculations of the radial distribution function of an 864-particle Lennard-Jones fluid [144]. The noisy curve (dots) was obtained using the conventional histogram method for a *single* liquid configuration (histogram bin-width equal to 0.005σ). The other two, almost indistinguishable curves, are the results using the histogram method for a run of 10,000 simulation steps (triangles) and the result obtained for a *single* configuration, using the method of [144] (gray curve). Figure: courtesy of Samuel Coles.

scales as $1/\sqrt{\Delta}$). Suppose that the number of pairs in the interval $\{r, r + \Delta r\}$ is $N_p(r)$, then we divide this number by the average number of pairs that would be found in the same range in an ideal (non-interacting) system. That number is $N_p^{\text{id}}(r) = \frac{1}{2}N\rho(4\pi/3)[(r + \Delta r)^3 - r^3]$ (in three dimensions). The factor (1/2) is due to the fact that we count every pair only once. Then our estimate for $g(r)$ is

$$g(r) = \frac{\langle N_p(r) \rangle}{N_p^{\text{id}}(r)}. \quad (5.1.43)$$

This calculation is so simple that it seems hard to imagine that one can do better, and indeed during the first six decades of molecular simulation, the above approach was overwhelmingly used to compute $g(r)$. However, in 2013 Borgis et al. [144,145] (see also [146]) proposed an alternative method to compute $g(r)$ that has two advantages: 1) it yields a smaller statistical error and 2) it does not require binning. In deriving the result of Ref. [144] we follow a slightly different approach than that paper.

The value of the radial distribution function at a distance r from a reference particle is equal to the angular average of $\rho(\mathbf{r})/\rho$:

$$g(r) = \frac{1}{\rho} \int d\hat{\mathbf{r}} \langle \rho(\mathbf{r}) \rangle_{N-1} = \frac{1}{\rho} \int d\hat{\mathbf{r}} \left\langle \sum_{j \neq i} \delta(\mathbf{r} - \mathbf{r}_j) \right\rangle_{N-1}, \quad (5.1.44)$$

where N is the total number of particles in the system, ρ denotes the average number density ($\rho \equiv (N/V)$) and \mathbf{r}_j is the distance of particle j from the origin, where particle i is located. $\hat{\mathbf{r}}$ is the unit vector in the direction of \mathbf{r} . For simplicity,

we have written down the expression for $g(r)$ for a given particle i , and hence the sum of $j \neq i$ is keeping i fixed, but in practice the expression is averaged over all equivalent particles i . The angular brackets denote the thermal average

$$\langle \dots \rangle_{N-1} \equiv \frac{\int d\mathbf{r}^{N-1} e^{-\beta U(\mathbf{r}^N)} (\dots)}{\int d\mathbf{r}^{N-1} e^{-\beta U(\mathbf{r}^N)}}, \quad (5.1.45)$$

where we integrate over $N - 1$ coordinates, because particle i is held fixed.

We can now write

$$\left(\frac{\partial g(r)}{\partial r} \right) = \frac{1}{\rho} \frac{\partial}{\partial r} \int d\hat{\mathbf{r}} \left\langle \sum_{j \neq i} \delta(\mathbf{r} - \mathbf{r}_j) \right\rangle. \quad (5.1.46)$$

The only term that depends on r (the length of \mathbf{r}) is the δ -function. We can therefore write

$$\left(\frac{\partial g(r)}{\partial r} \right) = \frac{1}{\rho} \int d\hat{\mathbf{r}} \left\langle \sum_{j \neq i} \hat{\mathbf{r}} \cdot \nabla_{\mathbf{r}} \delta(\mathbf{r} - \mathbf{r}_j) \right\rangle. \quad (5.1.47)$$

As the arguments of the δ -function is $\mathbf{r} - \mathbf{r}_j$, we can replace $\hat{\mathbf{r}} \cdot \nabla_{\mathbf{r}}$ by $-\hat{\mathbf{r}}_j \cdot \nabla_{\mathbf{r}_j}$ and perform a partial integration:

$$\begin{aligned} \left(\frac{\partial g(r)}{\partial r} \right) &= \frac{-1}{\rho} \frac{\int d\hat{\mathbf{r}} \int d\mathbf{r}^{N-1} e^{-\beta U(\mathbf{r}^N)} \sum_{j \neq i} \hat{\mathbf{r}} \cdot \nabla_{\mathbf{r}} \delta(\mathbf{r} - \mathbf{r}_j)}{\int d\mathbf{r}^{N-1} e^{-\beta U(\mathbf{r}^N)}} \\ &= \frac{-\beta}{\rho} \frac{\int d\hat{\mathbf{r}} \int d\mathbf{r}^{N-1} e^{-\beta U(\mathbf{r}^N)} \sum_{j \neq i} \delta(\mathbf{r} - \mathbf{r}_j) \hat{\mathbf{r}}_j \cdot \nabla_{\mathbf{r}_j} U(\mathbf{r}^N)}{\int d\mathbf{r}^{N-1} e^{-\beta U(\mathbf{r}^N)}} \\ &= \frac{\beta}{\rho} \int d\hat{\mathbf{r}} \left\langle \sum_{j \neq i} \delta(\mathbf{r} - \mathbf{r}_j) \hat{\mathbf{r}}_j \cdot \mathbf{F}_j(\mathbf{r}^N) \right\rangle_{N-1}, \end{aligned} \quad (5.1.48)$$

where $\hat{\mathbf{r}} \cdot \mathbf{F}_j \equiv F_j^{(r)}$ denotes the force on particle j in the radial direction. We can now integrate with respect to r

$$\begin{aligned} g(r) &= g(r=0) + \frac{\beta}{\rho} \int_0^r dr' \int d\hat{\mathbf{r}}' \left\langle \sum_{j \neq i} \delta(\mathbf{r}' - \mathbf{r}_j) F_j^{(r)}(\mathbf{r}^N) \right\rangle_{N-1} \\ &= g(r=0) + \frac{\beta}{\rho} \int_{r' < r} dr' \left\langle \frac{\sum_{j \neq i} \delta(\mathbf{r}' - \mathbf{r}_j) F_j^{(r)}(\mathbf{r}^N)}{4\pi r'^2} \right\rangle_{N-1} \\ &= g(r=0) + \frac{\beta}{\rho} \sum_j \left\langle \theta(r - r_j) \frac{F_j^{(r)}(\mathbf{r}^N)}{4\pi r_j^2} \right\rangle_{N-1}, \end{aligned} \quad (5.1.49)$$

where θ denotes the Heaviside step function. To make a connection to the results of ref. [144], we note that in a homogeneous system, all particles i of the same species are equivalent. We can therefore write

$$g(r) = g(r=0) + \frac{\beta}{N\rho} \sum_{i=1}^N \sum_{j \neq i} \left\langle \theta(r - r_{ij}) \frac{F_j^{(r)}(\mathbf{r}^N)}{4\pi r_{ij}^2} \right\rangle_{N-1}.$$

But i and j are just dummy indices. So we obtain the same expression for $g(r)$ by permuting i and j , except that if $\hat{\mathbf{r}} = \hat{\mathbf{r}}_{ij}$, then $\hat{\mathbf{r}} = -\hat{\mathbf{r}}_{ji}$. Adding the two equivalent expressions for $g(r)$ and dividing by two, we get

$$g(r) = g(r=0) + \frac{\beta}{2N\rho} \sum_{i=1}^N \sum_{j \neq i} \left\langle \theta(r - r_{ij}) \frac{F_j^{(r)}(\mathbf{r}^N) - F_i^{(r)}(\mathbf{r}^N)}{4\pi r_{ij}^2} \right\rangle_{N-1}. \quad (5.1.50)$$

Eq. (5.1.50) is equivalent to the result of Ref. [144].

The remarkable feature of Eq. (5.1.50) is that $g(r)$ depends not just on the number of pairs at distance r , but on *all* pair distances less than r . We stress that we have not assumed that the interactions in the system are pairwise additive: $F_i - F_j$ is *not* a pair force.

Note that the choice of r in Eqs. (5.1.50) and (5.1.52) is arbitrary, hence no binning is required and therefore the statistical accuracy of Eq. (5.1.46) does not depend on the choice of a bin width. In the examples shown in Fig. 5.2, the prediction based on Eq. (5.1.52) appears to be more accurate than that obtained in a direct calculation of $g(r)$ with a reasonable choice for the bin width. As explained in Illustration 2, the statistical error can be further reduced by combining two independent estimates of $g(r)$.

Illustration 2 (Force-method to estimate the $g(r)$). At large r , $g(r)$ approaches one. It then follows from Eq. (5.1.50) that, provided that $g(r=0)=0$,

$$1 = g(r) = \frac{\beta}{2N\rho} \sum_{i=1}^N \sum_{j \neq i} \left\langle \frac{F_j^{(r)}(\mathbf{r}^N) - F_i^{(r)}(\mathbf{r}^N)}{4\pi r_{ij}^2} \right\rangle_{N-1} = g(r) - h(r), \quad (5.1.51)$$

where $h(r) \equiv g(r) - 1$. It then follows that

$$h(r) = -\frac{\beta}{2N\rho} \sum_{i=1}^N \sum_{j \neq i} \left\langle \theta(r_{ij} - r) \frac{F_j^{(r)}(\mathbf{r}^N) - F_i^{(r)}(\mathbf{r}^N)}{4\pi r_{ij}^2} \right\rangle_{N-1}. \quad (5.1.52)$$

In the above equation, $h(r)$ depends only on pairs with a distance *larger than* r . Interestingly, this expression in Eq. (5.1.52) for $g(r) - 1$ is numerically not identical to Eq. (5.1.49): one expression is more accurate for small r , the other for large r ; the variance in the estimate of $g(r)$ can be reduced by combining the two results [147]: see Fig. 5.3.

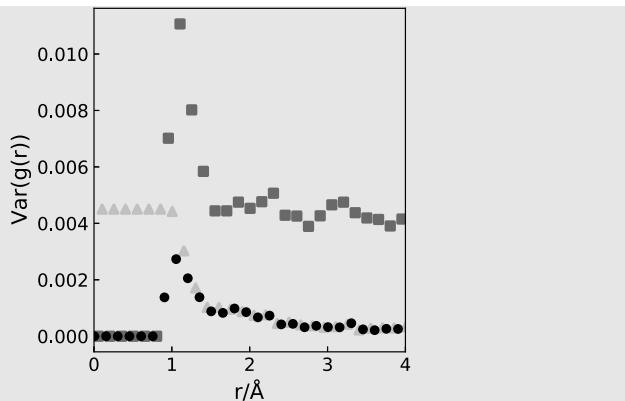


FIGURE 5.3 This figure shows the variance in the estimate of $g(r)$ using, Eq. (5.1.50) (triangles), Eq. (5.1.52) (squares) and the optimal, weighted combination of the two [147] (circles). The figure shows that the overall variance can be reduced by combining the two independent estimates of $g(r)$. The system studied was a Lennard-Jones fluid ($N=864$, $T = 1.35$ and $\rho = 0.8$). Figure: courtesy of Samuel Coles.

The limitation of the method of Ref. [144] is that it only works for systems in thermal equilibrium. But that limitation can also be an advantage: if Eq. (5.1.49) disagrees with the standard expression for $g(r)$, then that is an indication that the system is not in thermal equilibrium (the converse is not true).

Example 5 (Static properties of the Lennard-Jones fluid). Let us illustrate the results of the previous sections with an example. Like in the section on Monte Carlo simulations, we choose the Lennard-Jones fluid as our model system. We use a truncated and shifted potential (see also section 3.3.2.2):

$$u^{\text{tr-sh}}(r) = \begin{cases} u^{\text{lj}}(r) - u^{\text{lj}}(r_c) & r \leq r_c \\ 0 & r > r_c \end{cases}, \quad (5.1.53)$$

where $u^{\text{lj}}(r)$ is the Lennard-Jones potential and for these simulations $r_c = 2.5\sigma$ is used.

During a simulation, we must check if the system has equilibrated, or at least has reached a state that is stable on the timescale of the simulations. We then collect data about observable quantities and, at the end of the simulation, we compute averages and estimate the statistical errors. This Example illustrated how such a simulation might proceed.

At the beginning of the simulation, we prepare the system in state that, of course, is not yet equilibrated. Here, we will assume that the particles were originally positioned on the lattice sites of a face-centered cubic crystal.

We initialize the velocities of the particles such that the initial kinetic energy corresponds to a temperature $T = 0.728$. The density is fixed at $\rho =$

0.8442, which is a typical liquid density, close to the triple (gas-liquid-solid) point of the Lennard-Jones fluid. When we start the MD simulation from this initial configuration, the potential energy will decrease and, because energy is conserved, the kinetic energy will go up —Fig. 5.4.

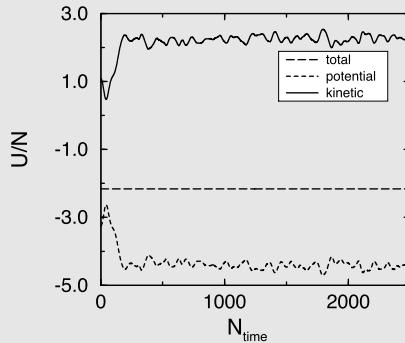


FIGURE 5.4 Total, potential, and kinetic energy per particle U/N as a function of the number of time steps N_{time} .

In Fig. 5.4 shows the time evolution from the start of the simulation of the total energy, kinetic energy, and potential energy. Note that the total energy, whilst fluctuating slightly, does not drift. The kinetic and potential energies change strongly during the equilibration period but after that they oscillate around their equilibrium value. The figure suggests that for this (very small) system, equilibration is complete in less than 1000 time steps. However, longer equilibrations will be needed for larger systems, and for glassy systems, MD may not achieve equilibration at all.

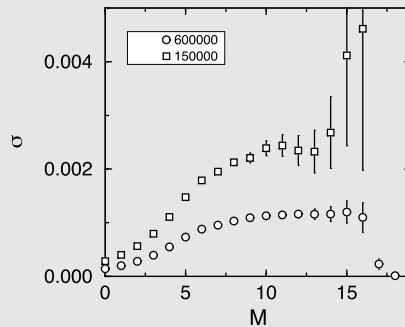


FIGURE 5.5 The standard deviation σ in the potential energy as a function of the number of block operations M for a simulation of 150,000 and 600,000 time steps. This variance is calculated using Eq. (5.3.28).

Next, we consider error estimates. We used the method of Flyvbjerg and Petersen [148] to estimate the statistical error in the potential energy (see Fig. 5.5. In this figure, the number of blocking operations by M , and from the plateau, we obtain an estimate of the standard deviation in our results. The

figure also shows the effect of increasing the total length of the simulation by a factor of 4; as expected, the statistical error in the potential energy decreases by a factor of 2.

We obtain the following results: for the potential energy $U = -4.4190 \pm 0.0012$ and for the kinetic energy $K = 2.2564 \pm 0.0012$, the latter corresponding to an average temperature of $T = 1.5043 \pm 0.0008$. For the pressure, we find 5.16 ± 0.02 .

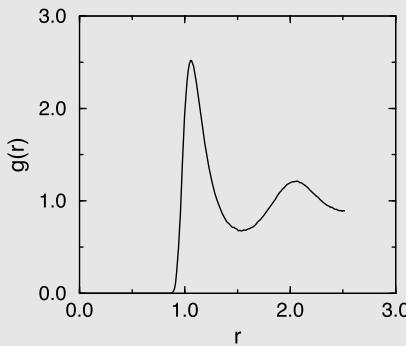


FIGURE 5.6 Radial distribution function of a Lennard-Jones fluid close to the triple point: $T = 1.5043 \pm 0.0008$ and $\rho = 0.8442$.

In Fig. 5.6, the radial distribution function is shown. To determine $g(r)$, we used Algorithm 8. This $g(r)$ shows the characteristics of a dense liquid. We can use the radial distribution function to calculate the energy and pressure. The potential energy per particle can be calculated from

$$\begin{aligned} U/N &= \frac{1}{2}\rho \int_0^\infty dr u(r)g(r) \\ &= 2\pi\rho \int_0^\infty dr r^2 u(r)g(r) \end{aligned} \quad (5.1.54)$$

and for the pressure from

$$\begin{aligned} P &= \rho k_B T - \frac{1}{3} \frac{1}{2} \rho^2 \int_0^\infty dr \frac{du(r)}{dr} r g(r) \\ &= \rho k_B T - \frac{2}{3} \pi \rho^2 \int_0^\infty dr \frac{du(r)}{dr} r^3 g(r), \end{aligned} \quad (5.1.55)$$

where $u(r)$ is the pair potential.

Eqs. (5.1.54) and (5.1.55) can be used to check the consistency of the energy and pressure calculations and the determination of the radial distribution function. In our example, we obtained from the radial distribution function for the potential energy $U/N = -4.419$ and for the pressure $P = 5.181$, which is in good agreement with the direct calculation.

For more details, see SI (Case Study 4).

Algorithm 8 (The radial distribution function)

function grsample	
delg=box/(2*nhis)	delg is bin size
ngr=ngr+1	calling <i>grsample</i> increments <i>ngr</i>
for 1 ≤ i ≤ npart-1 do	loop over all pairs
for i+1 ≤ j ≤ npart do	nearest image only
xr=x(i)-x(j)	In 3d: same for y and z
xr=xr-box*round(xr/box)	only consider distances < <i>box</i> /2
r=sqrt(xr*xr+yr*yr+yz*yz)	
if r < box/2 then	histogram incremented for pair <i>ij</i>
ig=int(r/delg)	
g(ig)=g(ig)+2	
endif	
enddo	
enddo	
end function	
function grnormalize	
gfac = (4/3)*pi*delg**3	<i>gfac</i> convert bins to 3d shells
for 1 ≤ i ≤ nhis do	
vb=gfac*((i+1)**3-i**3)	3d volume in <i>i</i> -th bin
nid=vb*p	number of ideal gas particles in <i>vb</i>
g(i)=g(i)/(ngr*npart*nid)	normalize <i>g(r)</i>
enddo	
end function	

Specific Comments (for general comments, see p. 7)

1. Function **grsample** accumulates a histogram of pair distances
2. Function **grnormalize** normalizes the radial distribution function at the end of the simulation
3. The array *g* contains *nhis* bins. It accumulates a histogram of pair distances
4. One bin in *g* corresponds to a radial shell with thickness *delg*
5. Before the first call of **grsample**, we zero the array *g(nhis)* and *ngr*, which counts the number of calls of the function **grsample**.
6. The number-density of the system is denoted by *p*
7. For reasons of computational efficiency, the sampling *g(r)* is usually combined with the force calculation (see Algorithm 5).

5.2 Dynamical properties

Both the thermodynamic properties and the structural properties mentioned previously do not depend on the time evolution of the system: they are static equilibrium averages. Such averages can be obtained equally well by **Molec-**

ular Dynamics and Monte Carlo simulations. However, in addition to the static equilibrium properties, we can also measure dynamic equilibrium properties in a Molecular Dynamics simulation. At first sight, a dynamic equilibrium property appears to be a contradiction: in equilibrium all properties are independent of time, hence any time dependence in the macroscopic properties of a system would seem to be related to non-equilibrium behavior. However, as explained in section 2.5.2 on Linear Response Theory, the time-dependent behavior of a system that is only weakly perturbed is completely described by the time-correlation function of fluctuations in the system at equilibrium.

Before discussing the relation between time-correlation functions and transport coefficients, we first describe another widely used method to study transport properties using equilibrium simulations, using the coefficient of self-diffusion as an example.

5.2.1 Diffusion

Diffusion is the process whereby an initially nonuniform concentration profile (e.g., an ink drop in water) is smoothed in the absence of flow (no stirring). Diffusion is caused by the molecular motion of the particles in the fluid. The macroscopic law that describes diffusion is known as Fick's law, which states that the flux \mathbf{j} of the diffusing species is proportional to the negative gradient in the concentration of that species:

$$\mathbf{j} = -D \nabla c, \quad (5.2.1)$$

where D , the constant of proportionality, is referred to as the *diffusion coefficient*.⁷ In what follows, we shall be discussing a particularly simple form of diffusion, namely, the case where the molecules of the diffusing species are identical to the other molecules, but for a label that does not affect the interaction of the labeled molecules with the others. For instance, this label could be a particular polarization of the nuclear spin of the diffusing species (see e.g., [149]) or a modified isotopic composition. Diffusion of a labeled molecule among other identical molecules is called *self-diffusion*.⁸

We can use Fick's law to compute the time-dependence of the concentration profile $c(\mathbf{r}, t)$ of the tagged species, under the assumption that, at time $t = 0$, the tagged species was concentrated at the origin of our coordinate frame. To compute the time evolution of the concentration profile, we combine Fick's law with

⁷ In Appendix D we discuss diffusion in the context of non-equilibrium thermodynamics, where the primary driving force is the gradient of the chemical potential, rather than the concentration gradient.

⁸ Eq. (5.2.1) is a simplification of the basic diffusion equation relating the particle flux to the gradient of the chemical potential. The full expression can be found in ref. [57]. However, for self-diffusion, Eq. (5.2.1) is not an approximation.

an equation that expresses conservation of the total amount of labeled material:

$$\frac{\partial c(\mathbf{r}, t)}{\partial t} + \nabla \cdot \mathbf{j}(\mathbf{r}, t) = 0. \quad (5.2.2)$$

Combining Eq. (5.2.2) with Eq. (5.2.1), we obtain

$$\frac{\partial c(\mathbf{r}, t)}{\partial t} - D \nabla^2 c(\mathbf{r}, t) = 0. \quad (5.2.3)$$

We can solve Eq. (5.2.3) with the boundary condition

$$c(\mathbf{r}, 0) = \delta(\mathbf{r}) \quad (5.2.4)$$

($\delta(\mathbf{r})$ is the d -dimensional Dirac delta function) to yield

$$c(r, t) = \frac{1}{(4\pi Dt)^{d/2}} \exp\left(-\frac{r^2}{4Dt}\right), \quad (5.2.5)$$

where r is the scalar distance from the origin. As before, d denotes the dimensionality of the system. For what follows we do not need $c(r, t)$ itself, but only the time dependence of its second moment:

$$\langle r^2(t) \rangle \equiv \int d\mathbf{r} c(r, t) r^2, \quad (5.2.6)$$

where we have used the fact that we have imposed

$$\int d\mathbf{r} c(r, t) = 1. \quad (5.2.7)$$

We can obtain an expression for the time evolution of $\langle r^2(t) \rangle$ by multiplying Eq. (5.2.3) by r^2 and integrating over all space. We then obtain:

$$\frac{\partial}{\partial t} \int d\mathbf{r} r^2 c(r, t) = D \int d\mathbf{r} r^2 \nabla^2 c(r, t). \quad (5.2.8)$$

The left-hand side of this equation is simply equal to

$$\frac{\partial \langle r^2(t) \rangle}{\partial t}. \quad (5.2.9)$$

Applying partial integration to the right-hand side, we obtain

$$\begin{aligned} \frac{\partial \langle r^2(t) \rangle}{\partial t} &= D \int d\mathbf{r} r^2 \nabla^2 c(r, t) \\ &= D \int d\mathbf{r} \nabla \cdot (r^2 \nabla c(r, t)) - D \int d\mathbf{r} \nabla r^2 \cdot \nabla c(r, t) \end{aligned}$$

$$\begin{aligned}
&= D \int d\mathbf{S} (r^2 \nabla c(r, t)) - 2D \int d\mathbf{r} \mathbf{r} \cdot \nabla c(r, t) \\
&= 0 - 2D \int d\mathbf{r} (\nabla \cdot \mathbf{r} c(r, t)) + 2D \int d\mathbf{r} (\nabla \cdot \mathbf{r}) c(r, t) \\
&= 0 + 2dD \int d\mathbf{r} c(r, t) \\
&= 2dD.
\end{aligned} \tag{5.2.10}$$

Eq. (5.2.10) relates the (self)diffusion coefficient D to the width of the concentration profile. Eq. (5.2.10) was derived by Einstein and is therefore called an *Einstein relation*. The important feature of Eq. (5.2.10) is that it relates a macroscopic transport coefficient (D), to a microscopic observable ($\langle r^2(t) \rangle$), which is the mean-squared distance over which the labeled molecules have moved in a time interval t . Eq. (5.2.10) suggests how to measure D in a computer simulation. For every particle i , we measure the distance traveled in time t , $\Delta\mathbf{r}_i(t)$, and we plot the mean square of these distances as a function of the time t :

$$\langle \Delta r(t)^2 \rangle = \frac{1}{N} \sum_{i=1}^N \Delta\mathbf{r}_i(t)^2. \tag{5.2.11}$$

An example of such a plot is shown in Fig. 5.9. We should be specific about what we mean with the displacement of a particle in a system with periodic boundary conditions. The displacement that we are interested in, is the time integral of the velocity of the tagged particle:

$$\Delta\mathbf{r}(t) = \int_0^t dt' \mathbf{v}(t'). \tag{5.2.12}$$

Eq. (5.2.12) allows us to express the diffusion coefficient in terms of the particle velocities. We start with the relation

$$2D = \lim_{t \rightarrow \infty} \frac{\partial \langle x^2(t) \rangle}{\partial t}, \tag{5.2.13}$$

where, for convenience, we consider only one Cartesian component of the mean-squared displacement. Expressing $x(t)$ as the time integral of the x -component of the tagged-particle velocity, we obtain

$$\begin{aligned}
\langle x^2(t) \rangle &= \left\langle \left(\int_0^t dt' v_x(t') \right)^2 \right\rangle \\
&= \int_0^t \int_0^t dt' dt'' \langle v_x(t') v_x(t'') \rangle \\
&= 2 \int_0^t \int_0^{t'} dt' dt'' \langle v_x(t') v_x(t'') \rangle.
\end{aligned} \tag{5.2.14}$$

The quantity $\langle v_x(t')v_x(t'') \rangle$ is the velocity autocorrelation function of the tagged particle (see section 2.5.2, Eq. (2.5.9)). It measures the correlation between the velocity of a particle at times t' and t'' . As time correlation functions in equilibrium only depend on the difference of t' and t'' , we can write

$$\langle v_x(t')v_x(t'') \rangle = \langle v_x(t - t'')v_x(0) \rangle. \quad (5.2.15)$$

Inserting Eq. (5.2.14) in Eq. (5.2.13), we obtain

$$\begin{aligned} 2D &= \lim_{t \rightarrow \infty} 2 \int_0^t dt'' \langle v_x(t - t'')v_x(0) \rangle \\ D &= \int_0^\infty d\tau \langle v_x(\tau)v_x(0) \rangle. \end{aligned} \quad (5.2.16)$$

In the last line of Eq. (5.2.16) we introduced the coordinate $\tau \equiv t - t''$. Eq. (5.2.16) shows that we can relate the diffusion coefficient D to the integral of the velocity auto-correlation function. Eq. (5.2.16) is an example of a *Green-Kubo relation* (see section 2.5.2).

One note of caution: the calculation of the diffusion coefficient is subject to large and slowly decaying finite-size effects. Due to the hydrodynamic interactions between a particle and its periodic images, the diffusion coefficient approaches the infinite-system limit as $1/N^{1/3}$ [150–152] in 3d. In 2d, the diffusion coefficient diverges.⁹

Illustration 3 (Diffusion coefficients). There are different ways to measure diffusion coefficients experimentally. An interesting example is the diffusion coefficient of gases adsorbed in a porous material (e.g., a zeolite or metal-organic framework). These materials are used to separate gases or as membranes. For these applications, it is important to have a molecular understanding of the diffusion of these gases in the pores.

With pulse-field-gradient NMR, one can measure the self-diffusion coefficient, D^s [153]. However, in practical applications of these materials, we are more interested in the transport or Fick diffusion coefficient D^t . This diffusion coefficient is often estimated from the rate at which the weight of material increases upon adsorption. These two diffusion coefficients are not the same, so in comparing molecular simulation results with experimental data, it is important to be aware of these differences.

The transport or Fick diffusion coefficient D^t , is obtained by measuring the flux, J caused by a gradient in the concentration:

$$J(c) \equiv -D^t(c)\nabla c. \quad (5.2.17)$$

⁹ The fact that the integral in Eq. (5.2.16) may diverge does not mean that particles diffuse infinitely fast, just that $\langle x^2(t) \rangle$ increases faster than linear with t .

However, as discussed in Appendix D, Eq. (D.1.5), the driving force for diffusion is a gradient in the chemical potential, μ . With this definition, we obtain the third definition of a diffusion coefficient, which we refer to as the collective diffusion coefficient (or corrected, or Maxwell-Stefan diffusion coefficient), D^c , which for a system at constant temperature reads:

$$J(c) \equiv -\frac{L(c)}{k_B T} \nabla \mu = -\frac{D^c(c)}{k_B T c} \nabla \mu, \quad (5.2.18)$$

where $L(c)$ is the Onsager transport coefficient. As the flux we measure is independent of how we define the diffusion coefficient, the relation between these diffusion coefficients is:

$$D^t = D^c \frac{1}{k_B T c} \frac{\nabla \mu}{\nabla \ln c} = \Gamma D^c,$$

where Γ is the thermodynamic coefficient:

$$\Gamma = \frac{1}{k_B T} \frac{\partial \mu}{\partial \ln c} = \frac{\partial \ln f}{\partial \ln c},$$

where we have replaced the chemical potential by the fugacity, f . For adsorption in a porous material, the relation between the concentration in the pores and the pressure is given by the adsorption isotherm. Fig. 5.7a gives a typical isotherm. For low pressures, we can assume ideal gas behavior, $f = P$. Hence, the adsorption is given by the Henry coefficient:

$$c = H P = H f$$

and

$$\Gamma_{P \rightarrow 0} = \frac{k_B T}{c} \frac{\partial \ln f}{\partial \ln c} = 1.$$

At saturation, one needs an infinite increase of the chemical potential to increase the concentration inside the pores by 1 more molecule. Hence:

$$\Gamma_{P \rightarrow \infty} = \infty.$$

Hence, close to saturation $D^t \rightarrow \infty$. This result may appear counter-intuitive as one would expect that, at saturation, the molecules are packed tightly and unable to move much. However, here the difference between the self-diffusion coefficient and the transport-diffusion coefficient becomes important. The self-diffusion coefficient is a measure of the mobility of a single, labeled molecule in the fluid. In contrast, the transport diffusion coefficient measures the flux of molecules; hence, if a pore is saturated and we add one molecule on one end of the crystal, instantaneously, on the other end of the crystal, one molecule drops out. Hence, the flux, the number of molecules

that pass a unit surface per time unit, is infinite. While at the same time, our labeled molecule has hardly moved.

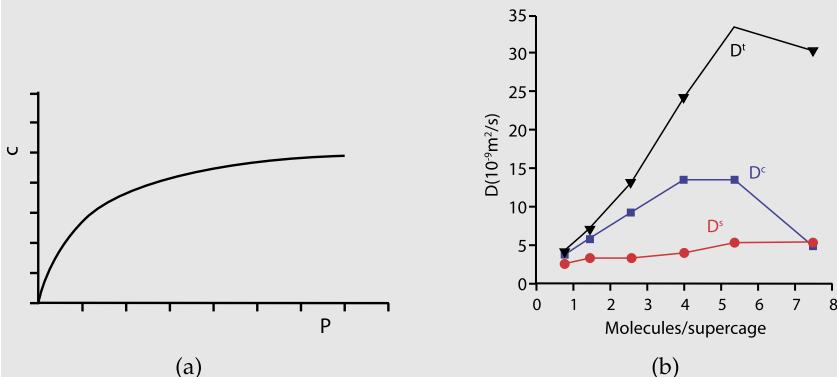


FIGURE 5.7 (a) a typical example of an adsorption isotherm of a gas adsorbed in a porous media c is the loading (molecules per unit volume) as a function of the pressure. (b) the three different diffusion coefficients as a function of loading c ; transport diffusion coefficient D^t , collective diffusion coefficient D^c , and self diffusion coefficient D^s , figure based on data from [154].

The collective diffusion coefficient is computed from the velocity correlation function:

$$D^c = \frac{1}{3} \int_0^\infty d\tau \sum_{i,j} \langle \mathbf{v}_i(o) \mathbf{v}_j(\tau) \rangle. \quad (5.2.19)$$

In the limit of $P \rightarrow 0$, one does not expect any correlations between the velocities of particles i and j . Hence for $i \neq j$, we have $\langle \mathbf{v}_i(o) \mathbf{v}_j(\tau) \rangle = 0$. Hence in this limit, the collective diffusion coefficient is equal to the self-diffusion coefficient.

$$D^c = \frac{1}{3N} \int_0^\infty d\tau \sum_{i,j} \langle \mathbf{v}_i(o) \mathbf{v}_j(\tau) \rangle \approx \frac{1}{3N} \int_0^\infty d\tau \sum_{i,j} \langle \mathbf{v}_i(o) \mathbf{v}_i(\tau) \rangle \delta_{ij} = D^s. \quad (5.2.20)$$

We now have seen three diffusion coefficients that each have a different dependence on the loading in pores. In Fig. 5.7b, these diffusion coefficients are reported for molecules in a pore. Interestingly, in the limit of zero loading, these three diffusion coefficients take the same value.

Green-Kubo relations have been derived for many other transport coefficients, such as the shear viscosity η ,

$$\eta = \frac{1}{V k_B T} \int_0^\infty dt \langle \sigma^{xy}(0) \sigma^{xy}(t) \rangle \quad (5.2.21)$$

where, for pairwise-additive potentials,

$$\sigma^{xy} = \sum_{i=1}^N \left(m_i v_i^x v_i^y + \frac{1}{2} \sum_{j \neq i} x_{ij} f_y(r_{ij}) \right); \quad (5.2.22)$$

the thermal conductivity λ_T ,

$$\lambda_T = \frac{1}{V k_B T^2} \int_0^\infty dt \langle j_z^e(0) j_z^e(t) \rangle \quad (5.2.23)$$

with, in the pairwise-additive case,¹⁰

$$j_z^e = \frac{d}{dt} \sum_{i=1}^N z_i \frac{1}{2} \left(m_i v_i^2 + \sum_{j \neq i} u(r_{ij}) \right); \quad (5.2.24)$$

and electrical conductivity σ_e

$$\sigma_e = \frac{1}{V k_B T} \int_0^\infty dt \langle j_x^{\text{el}}(0) j_x^{\text{el}}(t) \rangle \quad (5.2.25)$$

with

$$j_x^{\text{el}} = \sum_{i=1}^N q_i v_i^x. \quad (5.2.26)$$

A simple derivation of the **Green-Kubo (GK)** relations for electrical conductivity and viscosity is given in Appendix F.2 and F.3, respectively. For a more rigorous derivation (including the expression for the thermal conductivity), see ref. [59].

When using the **GK** expression to compute the viscosity of a molecule with internal vibrations, we can either compute the stress tensor using the forces acting on individual atoms, or we can consider only the forces acting between molecular centers of mass. The integral of the stress correlation is the same in both cases. However, the atomic description results in high-frequency oscillations in the stress auto-correlation function, which tends to obscure the overall shape of the stress **Auto-Correlation Function (ACF)** [159]. However, these high-frequency components do not affect the **GK** integral, which corresponds to the zero-frequency component of the stress **ACF**.

In some cases, e.g., polymers, expressing the stress in terms of center-of-mass forces is not useful. In that case, we can perform coarse-graining (or, more generally, filtering) of the stress **ACF** [160] following an approach similar to the

¹⁰ In the general case of non-pairwise additive interactions, the expressions for the heat flux are no longer simple (see e.g., [155]). Under those conditions, it may be attractive to impose a (small) explicit energy flux [156–158] and measure the resulting temperature gradient. The same argument applies to measuring the viscosity in non-pairwise additive systems.

one described in section 5.2.2 below. Similar arguments apply to the calculation of the thermal conductivity.

Time-correlation functions can be measured directly in a **Molecular Dynamics** simulation. For classical systems, the **GK** relation for D and the Einstein relation are strictly equivalent. There may be practical reasons for preferring one approach over the other, but in classical mechanics, the distinction is never fundamental. Algorithm 9 provides a simple illustration of the calculation of the mean-squared displacement and velocity auto-correlation function.

5.2.2 Order-n algorithm to measure correlations

The calculation of transport coefficients from the integral of a time-correlation function, or from a (generalized) Einstein relation, is not efficient in memory use (if we need to correlate for long times) nor in memory use (in the naive version sketched in Algorithm 9, the computing time scales quadratically with the length of the maximum time separation over which we compute correlations).

To illustrate this problem, consider the calculation of the velocity autocorrelation function and the diffusion coefficient of a molecular liquid. In a dense medium, the velocity autocorrelation function changes rapidly on microscopic time scales ($\mathcal{O}(10^{-13})$ s). We must therefore sample the velocity at even shorter time intervals. Yet, when probing the long-time decay of the velocity auto-correlation function, it is not necessary to sample with the same frequency. The example shown in Algorithm 9 does not allow for adjusting the sampling frequency.

Below, we describe an algorithm that allows us to measure fast and slow decay simultaneously at minimal computational cost. This scheme can be used to measure the correlation function itself, but in the example that we discuss, we show how it can be used to compute the associated transport coefficient. Let us denote by Δt the time interval between successive measurements of the velocity of the particles in the system. We can define block sums of the velocity of a given particle as follows:

$$\mathbf{v}^{(i)}(j) \equiv \sum_{l=(j-1)n+1}^{jn} \mathbf{v}^{(i-1)}(l) \quad (5.2.27)$$

with

$$\mathbf{v}^{(0)}(l) \equiv \mathbf{v}(l), \quad (5.2.28)$$

where $\mathbf{v}(l)$ is the velocity of a particle at time l . Eq. (5.2.27) is a recursive relation between block sums of level i and $i - 1$. The variable n determines the number of terms in the summation. For example, $\mathbf{v}^{(3)}(j)$ can be written as

$$\mathbf{v}^{(3)}(j) = \sum_{l_1=(j-1)n+1}^{jn} \mathbf{v}^{(2)}(l_1)$$

Algorithm 9 (Diffusion)

function dif	
if switch == 0 then	diffusion; switch = 0 init.
ntel=0	= 1 sample, and = 2 results
dtime=d़t*nsamp	Initialization
for 1 ≤ i ≤ tmax do	time counter
ntime(i)=0	time between two samples
vacf(i)=0	tmax total number of time step
r2t(i)=0	number of samples for time i
enddo	
else if switch == 1 then	sample
ntel=ntel+1	decide to take a new $t = 0$
if mod(ntel,it0) == 0 then	update number of $t = 0$
t0 = t0 + 1	see note 1
tt0=mod(t0-1,t0max)+1	store the time of $t = 0$
time0(tt0)=ntel	
for 1 ≤ i ≤ npart do	store position for given $t = 0$
x0(i,tt0)=x(i)	store velocity for given $t = 0$
vx0(i,tt0)=vx(i)	
enddo	
endif	
for 1 ≤ t ≤ min(t0,t0max) do	update vacf and r2, for $t = 0$
delt=ntel-time0(t)+1	actual time minus $t = 0$
if delt < tmax then	
vacf(delt)=vacf(delt)+	update $v_x(0)v_x(t)$
+ vx(i)*vx0(i,t)	
r2t(delt)=r2t(delt)+	update $(\Delta x(t))^2$
+ (x(i)-x0(i,t))**2	
enddo	
endif	
enddo	determine results
else if switch == 2 then	
for 1 ≤ i ≤ tmax do	time
time=d़t*(i+0.5)	$\langle v_x(0)v_x(t) \rangle$
vacf(i)=vacf(i)	
+ /(npart*ntime(i))	$\langle (\Delta x(t))^2 \rangle$
r2t(i)=r2t(i)/(npart*ntime(i))	
enddo	
endif	
end function	

Specific Comments (for general comments, see p. 7)

1. We define a new time origin every $it0$ -th function call. For each time origin, we store the current positions and velocities. The term $t0max$ is the maximum number of time origins that we can store. If we sample more time origins, the first one will be overwritten: this overwriting of old data limits the maximum time over which we can correlate to $t0max * it0$, which should not be smaller than $tmax$, the total number of time steps we sample. In section 5.2.2 and Algorithm 10, we show how to correlate for arbitrarily long times.
2. Because $nsamp$ determines the frequency at which the function **dif** is called, the time between two calls is $nsamp * delt$, where $delt$ is the time step.

$$\begin{aligned}
 &= \sum_{[l_1=(j-1)n+1]}^{jn} \sum_{[l_2=(l_1-1)n+1]}^{l_1n} \sum_{[l_3=(l_2-1)n+1]}^{l_2n} \mathbf{v}(l_3) \\
 &= \sum_{l=(j-1)n^3+1}^{n^3 j} \mathbf{v}(l) \\
 &\approx \frac{1}{\Delta t} \int_{l=(j-1)n^3+1}^{n^3 j} dt \mathbf{v}(t) = \frac{\mathbf{r}(n^3 j) - \mathbf{r}(n^3(j-1) + 1)}{\Delta t}.
 \end{aligned}$$

Clearly, the block sum of the velocity is related to the displacement of the particle in a time interval $n^i \Delta t$. In Fig. 5.8 the blocking operation is illustrated. From the block sums defined above, we can compute the **Velocity autocorrelation function (VACF)** with a resolution that decreases with increasing time. At each level of blocking, we need to store $n \times N$ block sums, where N is the number of particles (in practice, it will be more convenient to store the *block-averaged* velocities).

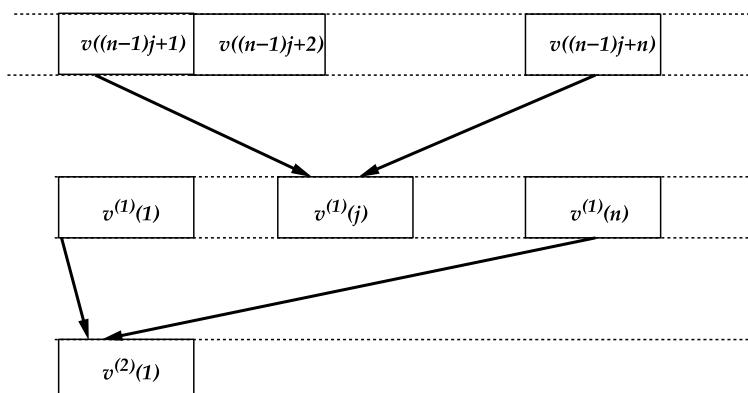


FIGURE 5.8 Coarse graining the velocities.

The total storage per particle for a simulation of length $t = n^i \Delta t$ is $i \times n$. This should be compared to the conventional approach where, to study correlations over the same time interval, the storage per particle would be n^i . In the conventional calculation of correlation functions, the number of floating-point operations scales at t^2 (or $t \ln t$, if the Fast Fourier technique is used). In contrast, in the present scheme, the number of operations scales as t . At every time step, we have to update $\mathbf{v}^{(0)}(t)$ and correlate it with all n entries in the $\mathbf{v}^{(0)}$ -array. The next block sum has to be updated and correlated once every n time steps, the third every n^2 steps, etc. This yields, for the total number of operations,

$$\frac{t}{\Delta t} \times n \left(1 + \frac{1}{n} + \frac{1}{n^2} + \cdots + \frac{1}{n^i} \right) < \frac{t}{\Delta t} n \frac{n}{n-1}. \quad (5.2.29)$$

Using this approach, we can quickly and efficiently compute a wide variety of correlation functions, both temporal and spatial. However, it should be stressed that each blocking operation leads to more coarse-graining. Hence, any high-frequency modulation of the long-time behavior of such correlation functions (e.g., a long-lived oscillation) will be washed out.

Interestingly, even though the VACF itself is approximate at long times, we can still compute the integral of the VACF (i.e., the diffusion coefficient), with no loss in numerical accuracy. Below, we discuss this coarse-grained technique for computing the diffusion coefficient.

Let us define

$$\Delta \bar{\mathbf{r}}^{(i)}(j) \equiv \sum_{l=0}^j \mathbf{v}^{(i)}(l) \Delta t = \mathbf{r}(n^i) - \mathbf{r}(0). \quad (5.2.30)$$

The square of the displacement of the particle in a time interval $n^i \Delta t$ can be written as

$$(\Delta \bar{\mathbf{r}}^2)^{(i)}(j) = [\mathbf{r}(n^i) - \mathbf{r}(0)]^2 = \Delta \bar{\mathbf{r}}^{(i)}(j) \cdot \Delta \bar{\mathbf{r}}^{(i)}(j). \quad (5.2.31)$$

To compute the diffusion coefficient, we should follow the time dependence of the mean-squared displacement. As a first step, we must determine $\Delta \bar{\mathbf{r}}^{(i)}(j)$ for all i and all j . Moreover, to improve the statistics, we wish to use every sample point as a new time origin. To achieve this, we again create arrays of length n . However, these arrays do not contain the same block sums as before, but partial block sums (see Algorithm 10). In particular:

1. At every time interval Δt , the lowest-order blocking operation is performed through the following steps:
 - a. We first consider the situation that all lowest-order accumulators have already been filled at least once (this is true if $t > n \Delta t$). The value of the current velocity $v(t)$ is added to

$$\mathbf{v}_{\text{sum}}(1, j) = \mathbf{v}_{\text{sum}}(1, j+1) + \mathbf{v}(t) \quad (5.2.32)$$

for $j = 1, n-1$, and

$$\mathbf{v}_{\text{sum}}(1, j) = \mathbf{v}(t) \quad (5.2.33)$$

for $j = n$.

- b.** These operations yield

$$\mathbf{v}_{\text{sum}}(1, l) = \sum_{j=l-n+l}^{j=t} \mathbf{v}(j). \quad (5.2.34)$$

The equation allows us to update the accumulators for the mean-squared displacement (5.2.31) for $l = 1, 2, \dots, n$:

$$(\Delta \bar{\mathbf{r}}^2)^{(0)}(l) = (\Delta \bar{\mathbf{r}}^2)^{(0)}(l) + \mathbf{v}_{\text{sum}}^2(1, l) \Delta t^2. \quad (5.2.35)$$

- 2.** If the current time step is a multiple of n , we perform the first blocking operation. If it is a multiple of n^2 we perform the second blocking, etc. Performing blocking operation i involves the following steps:

- a.** As before, we first consider the situation that all i th-order accumulators have already been filled at least once (i.e., $t > n^i \Delta t$). Using the $i - 1$ th block sum ($\mathbf{v}_{\text{sum}}(i - 1, 1)$), we update

$$\mathbf{v}_{\text{sum}}(i, j) = \mathbf{v}_{\text{sum}}(i, j + 1) + \mathbf{v}_{\text{sum}}(i - 1, 1) \quad (5.2.36)$$

for $j = 1, n-1$, and

$$\mathbf{v}_{\text{sum}}(i, j) = \mathbf{v}_{\text{sum}}(i - 1, 1) \quad (5.2.37)$$

for $j = n$.

- b.** These operations yield

$$\mathbf{v}_{\text{sum}}(i, l) = \sum_{j=n-l+1}^{j=n} \mathbf{v}_{\text{sum}}(i - 1, j). \quad (5.2.38)$$

The equations allow us to update the accumulators for the mean-squared displacement, Eq. (5.2.31), for $l = 1, 2, \dots, n$:

$$(\Delta \mathbf{r}^2)^{(i)}(l) = (\Delta \mathbf{r}^2)^{(i)}(l) + \mathbf{v}_{\text{sum}}^2(i, l) \Delta t^2. \quad (5.2.39)$$

- 3.** Finally, we must consider how to handle arrays that have not yet been completely filled. Consider the situation that only n_{max} of the n locations of the array that contains the i th-level sums have been initialized. In that case, we should proceed as follows:

- a.** Update the current block length: $n_{\text{max}} = n_{\text{max}} + 1$ ($n_{\text{max}} < n$).
b. For $j = 1, n_{\text{max}} - 1$

$$\mathbf{v}_{\text{sum}}(i, j) = \mathbf{v}_{\text{sum}}(i, j) + \mathbf{v}_{\text{sum}}(i - 1, 1). \quad (5.2.40)$$

c. For $j = n_{\max}$

$$\mathbf{v}_{\text{sum}}(i, j) = \mathbf{v}_{\text{sum}}(i - 1, 1). \quad (5.2.41)$$

The update of Eq. (5.2.31) remains the same.

In Case Study 6, we compare the present algorithm and the conventional algorithm for computing the diffusion coefficient of the Lennard-Jones fluid.

5.2.3 Comments on the Green-Kubo relations

As was discussed in section 2.5.2, the GK relations follow naturally from the statistical-mechanical description of the response of a many-body system to a weak, Hamiltonian perturbation. In some cases (e.g., heat conduction), the

Algorithm 10 (Diffusion: order- n algorithm)

function dif	Sample every n_{sample} MD steps
if switch == 0 then	switch = 0: initialization,
ntel=0	= 1: sample, = 2: results
dtime=dtime+nsamp	0: initialization
for 1 ≤ ib ≤ ibmax do	time counter for this function
ibl(ib)=0	time between two samples
for 1 ≤ j ≤ n do	ibmax max. number of blocks
tel(ib,j)=0	length of current block
delr2(ib,j)=0	n number of steps in a block
for 1 ≤ i ≤ npart do	counter number of averages
vxsum(ib,j,i)=0	running average mean-sq. displ.
enddo	
enddo	blocked velocity particle i
enddo	
else if switch == 2 then	2: print results
ibm = max(ibmax,ibl)	
for 1 ≤ ib ≤ ibm do	
jbm = min(ibl(ib),n)	
for 2 ≤ j ≤ jbm do	
time=dtime*j*n*(ib-1)	time
r2=delr2(ib,j)*dtime**2	mean-squared displacement
/tel(ib,j)	
enddo	
enddo	
... (continue)....	

```

...(continue)....
else if switch == 1 then
    ntel=ntel+1
    iblm=MaxBlock(ntel,n)

for 1 ≤ ib ≤ iblm do
    if ntel % n*(ib-1)==0 then
        ibl(ib)=ibl(ib)+1
        inm=max(ibl(ib),n)
    for 1 ≤ i ≤ npart do
        if ib == 1 then
            delx=vx(i)
        else
            delx=vxsum(ib-1,1,i)
        endif
    for 1 ≤ in ≤ inm do
        if inm ≠ n then
            inp=in
        else
            inp=in+1
        endif
        if in < inm then
            vxsum(ib,in,i)=
                vxsum(ib,inp,i)+delx
        else
            vxsum(ib,in,i)=delx
        endif
    enddo
    for 1 ≤ in ≤ inm do
        tel(ib,in)=tel(ib,in)+1
        delr2(ib,in)=delr2(ib,in)
        +vxsum(ib,inm-in+1,i)**2
    enddo
    endif
enddo
endif
end function

```

sample
maximum number of possible blocking operations
is $ntel$ a multiple of n^{ib} ?
increase current block length
set maximum block length to n
0th block: ordinary velocity
previous block velocity
test block length equal to n
Eqs. (5.2.36) or (5.2.40)
Eqs. (5.2.37) or (5.2.41)
counter number of updates
update Eq. (5.2.39)

Specific Comments (for general comments, see p. 7)

1. `MaxBlock(ntel,n)` gives the maximum number of blocking operations that can be performed on the current time step `ntel`.

Hamiltonian route is problematic. In that case, it is still possible to derive GK relations by considering a perturbation of the Lagrangian equations of motion [33].

Note, however, that the heat flux, or for that matter, the stress tensor, is not uniquely defined. However, as was pointed out by Baroni and collaborators [155,161], the GK integrals (i.e., the observable transport properties) are not changed by the addition of the flux associated with the time evolution of a *bounded*, conserved function of coordinates and momenta. Adding such a flux would change the time-dependence of the GK integrand and would therefore affect the interpretation of the finite-frequency response obtained by Fourier-transforming the flux auto-correlation function. But the integral (i.e., the zero-frequency response) would not be affected.

The robustness of the zero-frequency response is to be expected in view of Onsager's earlier expression for transport coefficients in terms of auto-correlation functions of fluxes [55,56], which was based on irreversible thermodynamics rather than on statistical mechanics. Onsager's expressions for transport coefficients in terms of integrals of flux-correlation functions required little more than the assumption that small fluctuations of extensive quantities in many-body systems are Gaussian and that, to the lowest order, the fluxes are linear in the thermodynamic restoring forces. The ability to transform the value of the GK integrands without changing the value of the integral, can be used to improve the statistical accuracy of the integrals [155,161].

Improving the accuracy of GK integration is important because correlation functions obtained in simulations are intrinsically noisy. Hence, the signal-to-noise ratio goes to zero as we increase the upper limit of the integral. Once the value of the integrand is not significantly different from zero, further integration only makes the results less accurate. By using the trick proposed in refs. [155,161], it is possible to mitigate (but not eliminate) this inevitable noise problem. Let us consider a simple example to clarify the effect of statistical noise in transport coefficients due to noise in the relevant correlation function. Suppose that we wish to measure a transport coefficient \mathcal{L}_{AA} given by the Green-Kubo relation

$$\mathcal{L}_{AA} = \lim_{t \rightarrow \infty} \int_0^t dt' \langle A(0)A(t') \rangle.$$

It is convenient to use the discretized form

$$\mathcal{L}_{AA} = \langle A^2(0) \rangle \Delta t / 2 + \lim_{n_{\max} \rightarrow \infty} \sum_{n=1}^{n_{\max}} \Delta t \langle A(0)A(n\Delta t) \rangle.$$

Let us consider the extreme case that $\langle A(0)A(n\Delta t) \rangle = 0$ for all $n \geq 1$. In that case, $\mathcal{L}_{AA} = \langle A(0)^2 \rangle \Delta t / 2$, independent of n_{\max} . However, the variance in \mathcal{L}_{AA} does depend on n_{\max} . In what follows, we assume that the fluctuations in A follow a 1d normal distribution:

$$\sigma_L^2 = \sum_{n=1}^{n_{\max}} \left(\langle \mathcal{L}_{AA}^2(n_{\max}) \rangle - \langle \mathcal{L}_{AA} \rangle^2 \right)$$

$$= \text{constant} + 2(\Delta t)^2 \left(A(0)^2 \right)^2 (n_{\max} - 1). \quad (5.2.42)$$

Eq. (5.2.42) shows that the numerical estimate of σ_L^2 grows linearly with n_{\max} , which corresponds to a linear variation with t_{\max} in the continuous case. Of course, the real σ_L^2 will be lower because it will be divided by the number of independent starting points. A reasonable way to truncate the GK integration is therefore to plot σ_L vs $\sqrt{t_{\max}}$ and determine the point where the linear part of σ_L becomes non-negligible compared with $\langle \mathcal{L}_{AA}(t_{\max}) \rangle$. See also Exercise 11.

A second comment on GK integrals is also in place: a more direct route to compute the linear response of a system to an external perturbation was proposed in the 1970's by Ciccotti et al. [162]. In this approach, the effect of a (very) weak perturbation on the time evolution of the system (i.e., on its trajectory through phase space) is computed directly. The response of the system is obtained by computing the time evolution of the observable of interest along the perturbed and unperturbed trajectories. Subtracting the unperturbed from the perturbed value of the observable yields (upon averaging) the time-dependent linear response. The approach of ref. [162] is more general than the GK route, but it also runs into a noise problem at longer times.

Example 6 (Dynamic properties of the Lennard-Jones fluid). As an example of the numerical determination of a dynamical property we consider the calculation of the self-diffusion coefficient. As shown in the previous section, the diffusion coefficient can be determined either from the mean-squared displacement, or from the velocity auto-correlation function (VACF). Here, we follow the approach of Algorithm 9 to compute these observables.

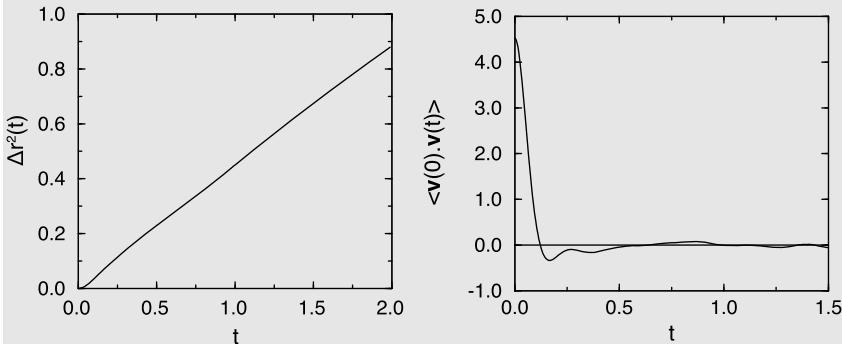


FIGURE 5.9 (Left) Mean-squared displacement $\Delta r(t)^2$ as a function of the simulation time t . Note that for long times, $\Delta r(t)^2$ varies linearly with t . The slope is then given by $2dD$, where d is the dimensionality of the system and D the self-diffusion coefficient. (Right) Velocity auto-correlation function $\langle \mathbf{v}(0) \cdot \mathbf{v}(t) \rangle$ as a function of the simulation time t .

In Fig. 5.9 the mean-squared displacement is shown as a function of the simulation time. From the mean-squared displacement we can determine the

diffusion using Eq. (5.2.13). This equation is however only valid in the limit $t \rightarrow \infty$. In practice this means that we have to verify that we compute the mean-squared displacement in the long-time regime where all transient effects have died down, and only the linear dependence on t remains.

The Green-Kubo integral of the VACF provides an alternative route to calculate the diffusion coefficient (Eq. (5.2.16)). Strictly speaking, the upper limit of the GK integral corresponds to $t \rightarrow \infty$. In practice, we must stop at an earlier time t_c , because once the VACF has decayed into the noise, further integration increases the noise but not the signal. A rough estimate of t_c can be obtained by determining the time where the integral seems to have reached a plateau value. Hopefully, all transients in the mean-squared displacement to have died down beyond t_c . However, if an ACF has a long-time tail that has a small amplitude, but a large integral, then truncation at t_c would introduce a large systematic error.

For more details, see SI (Case Study 5).

Example 7 (Algorithms to calculate the mean-squared displacement). In this Example, we compare the conventional (Algorithm 9), and the order- n methods (Algorithm 10) to determine the time-dependence of mean-squared displacement. The example considers a Lennard-Jones fluid.

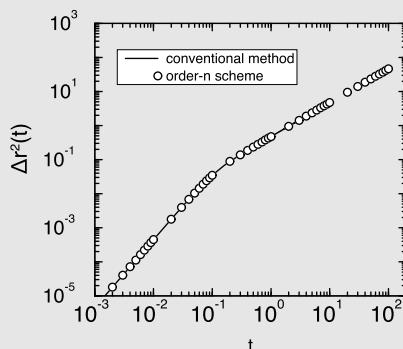


FIGURE 5.10 Mean-squared displacement as a function of time for the Lennard-Jones fluid ($\rho = 0.844$, $N = 108$, and $T = 1.50$); comparison of the conventional method with the order- n scheme.

In Fig. 5.10 we compare the time-dependence of the mean-squared displacement computed with the conventional method and the one obtained with the order- n scheme. For the order- n scheme, the memory requirement grows logarithmically with time, but for the conventional scheme it grows linearly. For the present simulation settings, the calculation using the conventional scheme could not be extended beyond $t > 10$. For the same memory allocation the order- n scheme the calculation could be extended to much longer times with no difficulty: it only depends on the simulation time. In the

present example, we stopped at $t = 200$. It is interesting to compare the accuracy of the two schemes. In the conventional scheme, the velocities of the particles at the current time step are used to update the mean-squared displacement of all time intervals. In the order- n scheme, the current time step is only used to update the lowest-order array of \mathbf{v}_{sum} (see Algorithm 10). The block sums of level i are updated only once every n^i time step. Therefore, for a total simulation of M time steps, the number of samples is much less for the order- n scheme; for the conventional scheme, we have M samples for all time steps, whereas the order- n scheme has M/n^i samples for the i th block velocity. Naively, one would think that the conventional scheme therefore is more accurate. In the conventional scheme, however, the successive samples will have much more correlation and therefore are not independent. To investigate the effect of these correlations on the accuracy of the results, we used the method of Flyvbjerg and Petersen [148] (see Appendix 5.3.3 and Case Study 4). In this method, the standard deviation is calculated as a function of the number of data blocks. If the data are correlated, the standard deviation will increase as a function of the number of blocks until the number of blocks is sufficient that the data in a data block are uncorrelated. If the data are uncorrelated, the standard deviation will be independent of the number of blocks. This limiting value is the standard deviation of interest.

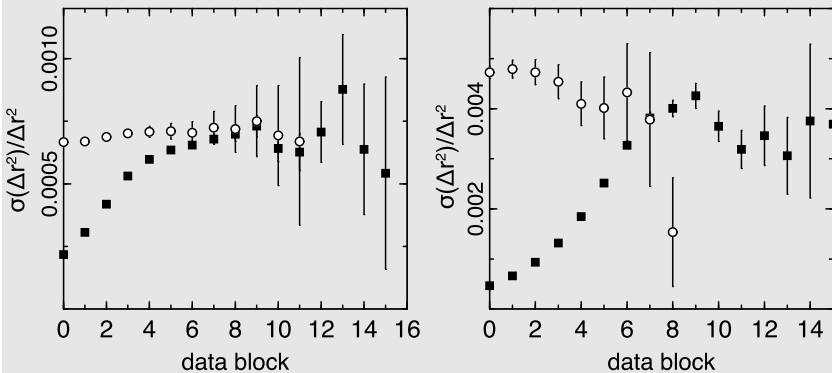


FIGURE 5.11 Relative error in the mean-squared displacement as a function of the number of data blocks as defined by Flyvbjerg and Petersen. The figures compare the conventional scheme (solid squares) with the order- n method (open circles) to determine the mean-squared displacement. The right figure is for $t = 0.1$ and the left figure for $t = 1.0$.

In these simulations the time step was $\Delta t = 0.001$ and the block length was set to $n = 10$. For both methods, the total number of time steps was equal. To calculate the mean-squared displacement, we have used 100,000 samples for all times in the conventional scheme. For the order- n scheme, we have used 100,000 samples for $t \in [0, 0.01]$, 10,000 for $t \in [0.01, 0.1]$, 1,000 for $t \in [0.1, 1]$, etc. This illustrates that the number of samples in the order- n scheme is considerably less than in the conventional scheme. The accuracy of the results, however, turned out to be the same. This is shown in Fig. 5.11

for $t = 0.1$ and $t = 1.0$. Since the total number of data blocking operations that can be performed on the data depends on the total number of samples, the number of blocking operations is less for the order- n method. Fig. 5.11 shows that for $t = 0.1$, the order- n scheme yields a standard deviation that is effectively constant after three data blocking operations, indicating the samples are independent, whereas the standard deviation using the conventional method shows an increase for the first six to eight data blocking operations. For $t = 1.0$, the order- n method is independent of the number of data blocks, the conventional method only after 10 data blocks. This implies that one has to average over $2^{10} \approx 1000$ successive samples to have two independent data points. In addition, the figure shows that the plateau value of the standard deviation is essentially the same for the two methods, which implies that for this case, the two methods are equally accurate, but, as shown below, the order- n method is computationally cheaper.

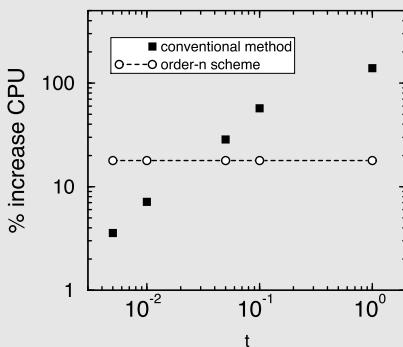


FIGURE 5.12 Percentage increase of the total CPU time as a function of the total time for which we determine the mean-squared displacement; comparison of the conventional scheme with the order- n scheme for the same system as is considered in Fig. 5.10.

In Fig. 5.12 we compare the CPU requirements of the two algorithms for simulations with a fixed total number of time steps. This figure shows the increase of the total CPU time of the simulation as a function of the total time for which the mean-squared displacement has been calculated. With the order- n scheme, the CPU time should be (nearly) independent of the total time for which we determine the mean-squared displacement, which is indeed what we observe. For the conventional scheme, however, the required CPU time increases significantly for longer times. At $t = 1.0$ the order- n scheme gives an increase of the total CPU time of 17%, whereas the conventional scheme shows an increase of 130%.

This example illustrates that the saving in memory as well as in CPU time of the order- n scheme can be significant, especially if we are interested in the mean-squared displacement at long times.

For more details, see SI (Case Study 6).

5.3 Statistical errors

Measurements in computer “experiments” are subject to the same sources of error as real experiments —and a few more.

We will not discuss systematic errors due to mistakes in the calculations, as the number of possible errors is unbounded. Some systematic errors have to do with the fact that we consider a finite system, and the properties of a finite system (even with periodic boundary conditions) are not identical to those of a macroscopic system. Such errors disappear as N , the number of particles in the system, tends to ∞ . Of course, we cannot achieve this limit, but often the finite-size effects decay as $1/N$, which means that for systems of a few thousand particles, the effects are typically small.

However, as discussed in section 5.2.1, below Eq. (5.2.16), some finite-size effects decay slower than $1/N$. The most striking examples of irreducible finite-size effects occur in the vicinity of a critical point. Binder [42,163] has developed the so-called finite-size scaling method to extract the correct limiting properties of systems at or near the critical point from simulations on finite-size systems. For more details on this approach, we refer the reader to the extensive literature on this topic, starting with the works of Binder himself.

The remainder of this section discusses the estimate of statistical errors in simulation results.

5.3.1 Static properties: system size

Let us consider the statistical accuracy of the measurement of a dynamical quantity A in a **Molecular Dynamics** simulation (the present discussion applies, with minor modifications, to **Monte Carlo** simulations). During a simulation of total length τ , we obtain the following estimate for the equilibrium-average of A :

$$A_\tau = \frac{1}{\tau} \int_0^\tau dt A(t), \quad (5.3.1)$$

where the subscript on A_τ refers to averaging over a time τ . If the ergodic hypothesis is justified then $A_\tau \rightarrow \langle A \rangle$, as $\tau \rightarrow \infty$, where $\langle A \rangle$ denotes the ensemble average of A . We can estimate $\sigma^2(A)$, the variance in A_τ :

$$\begin{aligned} \sigma^2(A) &= \left\langle A_\tau^2 \right\rangle - \langle A_\tau \rangle^2 \\ &= \frac{1}{\tau^2} \int_0^\tau \int_0^\tau dt dt' \langle [A(t) - \langle A \rangle][A(t') - \langle A \rangle] \rangle. \end{aligned} \quad (5.3.2)$$

Note that $\langle [A(t) - \langle A \rangle][A(t') - \langle A \rangle] \rangle$ in Eq. (5.3.2) is simply the time correlation function of fluctuations in the variable A . Let us denote this correlation function by $C_A(t - t')$. If the duration of the sampling τ is much larger than the

characteristic decay time t_A^c of C_A , then we may rewrite Eq. (5.3.2) as

$$\begin{aligned}\sigma^2(A) &\approx \frac{1}{\tau} \int_{-\infty}^{\infty} dt C_A(t) \\ &\approx \frac{2t_A^c}{\tau} C_A(0).\end{aligned}\quad (5.3.3)$$

In the last equation, we have used the definition of t_A^c as the integral from 0 to ∞ of the normalized correlation function $C_A(t)/C_A(0)$. The relative variance in A_τ , therefore, is given by¹¹

$$\frac{\sigma^2(A)}{\langle A \rangle^2} \approx (2t_A^c/\tau) \frac{\langle A^2 \rangle - \langle A \rangle^2}{\langle A \rangle^2}. \quad (5.3.4)$$

Eq. (5.3.4) shows that the root-mean-square error in A_τ is proportional to $\sqrt{t_A^c/\tau}$. This result is hardly surprising. It simply states the well-known fact the variance in a measured quantity is inversely proportional to the number of uncorrelated measurements. In the present case, this number is proportional to τ/t_A^c . This almost trivial result is important because it shows how the lifetime and amplitude of fluctuations in an observable A affect the statistical accuracy. This is of particular importance in the study of fluctuations associated with hydrodynamical modes or pretransitional fluctuations near a symmetry-breaking phase transition. Such modes usually have a characteristic lifetime proportional to the square of their wavelengths. To minimize the effects of the finite system size on such phase transitions, it is preferable to study systems with a box size L large compared with all relevant correlation lengths in the system. However, due to the slow decay of long-wavelength fluctuations, the number of time steps needed to keep the relative error fixed should scale as L^2 . As the CPU time for a run of fixed length is proportional to the number of particles (at best), the CPU time needed to maintain constant accuracy increases quite rapidly with the linear dimensions of the system (e.g., as L^5 in three dimensions).

The example above focused on the case of a quantity $A(t)$ computed in a [Molecular Dynamics](#) simulation, but the example carries over to [Monte Carlo](#) simulations, if we replace time with the number of [Monte Carlo](#) cycles. The above statement does not imply that cycles in MC simulations are equivalent to time in MD simulations; just that it may take a number of MC cycles before a fluctuation in quantity A has decayed.

One aspect of Eq. (5.3.4) is not immediately obvious; namely that it makes a difference whether the observable A can be written as a sum of weakly correlated single-particle properties. If this is the case, then the ratio

¹¹ Some quantities have correlation functions that integrate rigorously to zero (for instance, the auto-correlation function of the force on a particle). Typically, such quantities are time derivatives of other (bounded) observables ($\dot{B} = A$). In that case it follows from Eq. (5.3.1) that $\sigma^2(A)$ scales as $\sigma^2(B)/\tau^2$.

$(\langle A^2 \rangle - \langle A \rangle^2) / \langle A \rangle^2$ is inversely proportional to the number of particles, N . To see this, consider the expressions for $\langle A \rangle$ and $\langle A^2 \rangle - \langle A \rangle^2$ in this case:

$$\langle A \rangle = \sum_{i=1}^N \langle a_i \rangle = N \langle a \rangle \quad (5.3.5)$$

and

$$\langle A^2 \rangle - \langle A \rangle^2 = \sum_{i=1}^N \sum_{j=1}^N \langle [a_i - \langle a \rangle] [a_j - \langle a \rangle] \rangle. \quad (5.3.6)$$

If the correlations in the fluctuations in a_i and a_j can be ignored, then we find that

$$\frac{\langle A^2 \rangle - \langle A \rangle^2}{\langle A \rangle^2} = \frac{1}{N} \frac{\langle a^2 \rangle - \langle a \rangle^2}{\langle a \rangle^2}. \quad (5.3.7)$$

From Eq. (5.3.7) it is clear that the statistical error in a single-particle property is inversely proportional to \sqrt{N} . Hence, for single-particle properties, the accuracy improves as we go to larger systems (at fixed length of the simulation). In contrast, no such advantage is to be gained when computing truly collective properties.

5.3.2 Correlation functions

We can apply essentially the same arguments to estimate the statistical errors in time correlation functions. Suppose that we perform a simulation to measure the (auto)correlation function¹² of the dynamical quantity A . To obtain an estimate of $C_A(\tau) \equiv \langle A(0)A(\tau) \rangle$, we average the product $A(t)A(t + \tau)$ over the initial time t . Suppose that the length of our run is τ_0 , then our numerical estimate of $C_A(\tau)$ is

$$\bar{C}_A(\tau) = 1/\tau_0 \int_0^{\tau_0} dt A(t)A(t + \tau), \quad (5.3.8)$$

where the bar over C_A denotes the average over a finite time τ_0 . Next, we consider the variance in $\bar{C}_A(\tau)$ [164]:

$$\begin{aligned} & \langle \bar{C}_A(\tau)^2 \rangle - \langle \bar{C}_A(\tau) \rangle^2 \\ &= (1/\tau_0^2) \int_0^{\tau_0} \int_0^{\tau_0} dt' dt'' \langle A(t')A(t' + \tau)A(t'')A(t'' + \tau) \rangle \\ & \quad - (1/\tau_0^2) \int_0^{\tau_0} \int_0^{\tau_0} dt' dt'' \langle A(t')A(t' + \tau) \rangle \langle A(t'')A(t'' + \tau) \rangle. \end{aligned} \quad (5.3.9)$$

¹² The extension to cross-correlation functions of the type $\langle A(t)B(0) \rangle$ is straightforward and is left as an exercise to the reader.

The first term on the right-hand side of Eq. (5.3.9) contains a fourth-order correlation function. To simplify matters, we shall assume that the fluctuations in A follow a Gaussian distribution. This is not the simple Gaussian distribution that describes, for instance, the Maxwell distribution of particle velocities in equilibrium, but a multidimensional distribution that describes all correlations between fluctuations of A at different times. Here we only consider real fluctuations at discrete times. Then the generalized Gaussian distribution would be of the following form:

$$P(A(t_1), A(t_2), \dots, A(\tau_{0n})) = \text{const.} \times \exp \left[-\frac{1}{2} \sum_{i,j} A(t_i) \alpha(t_i - t_j) A(t_j) \right], \quad (5.3.10)$$

where the matrix $\alpha(t_i - t_j)$ is the “matrix” inverse of the (discrete) time correlation function $C_A(t_i - t_j)$, meaning that $\int d\tau C_A(t - \tau) \alpha(\tau - t') = \delta(t - t')$. For Gaussian variables, we can factorize all higher-order correlation functions. In particular,

$$\begin{aligned} & \langle A(t') A(t' + \tau) A(t'') A(t'' + \tau) \rangle \\ &= \langle A(t') A(t' + \tau) \rangle \langle A(t'') A(t'' + \tau) \rangle \\ &+ \langle A(t') A(t'') \rangle \langle A(t' + \tau) A(t'' + \tau) \rangle \\ &+ \langle A(t') A(t'' + \tau) \rangle \langle A(t' + \tau) A(t') \rangle. \end{aligned} \quad (5.3.11)$$

Inserting Eq. (5.3.11) in Eq. (5.3.9), we get

$$\begin{aligned} & \langle \bar{C}_A(\tau)^2 \rangle - \langle \bar{C}_A(\tau) \rangle^2 \\ &= (1/\tau_0^2) \int_0^{\tau_0} \int_0^{\tau_0} dt' dt'' \langle A(t') A(t'') \rangle \langle A(t' + \tau) A(t'' + \tau) \rangle \\ &+ (1/\tau_0^2) \int_0^{\tau_0} \int_0^{\tau_0} dt' dt'' \langle A(t') A(t'' + \tau) \rangle \langle A(t' + \tau) A(t'') \rangle \\ &= (1/\tau_0^2) \int_0^{\tau_0} \int_0^{\tau_0} dt' dt'' \langle A(t' - t'') A(0) \rangle^2 \\ &+ (1/\tau_0^2) \int_0^{\tau_0} \int_0^{\tau_0} dt' dt'' \langle A(t' - t'' - \tau) A(0) \rangle \langle A(t' - t'' + \tau) A(0) \rangle. \end{aligned} \quad (5.3.12)$$

As we consider the case where the length of the simulation, τ_0 , is much longer than the characteristic decay time of the fluctuations of A , we can write

$$\begin{aligned} & \langle \bar{C}_A(\tau)^2 \rangle - \langle \bar{C}_A(\tau) \rangle^2 \\ &= (1/\tau_0) \int_{-\infty}^{\infty} dx \left(\langle A(x) A(0) \rangle^2 + \langle A(x - \tau) A(0) \rangle \langle A(x + \tau) A(0) \rangle \right), \end{aligned} \quad (5.3.13)$$

where we have defined the variable x as $t' - t''$. It is instructive to consider two limiting cases, $\tau = 0$ and $\tau \rightarrow \infty$. For $\tau = 0$, we can write

$$\begin{aligned} \langle \bar{C}_A(\tau)^2 \rangle - \langle \bar{C}_A(\tau) \rangle^2 &= (2/\tau_0) \int_{-\infty}^{\infty} dx \langle A(x)A(0) \rangle^2 \\ &= 4 \langle A^2(0) \rangle^2 \frac{\tau^c}{\tau_0}. \end{aligned} \quad (5.3.14)$$

The last line of this equation defines the correlation time τ^c :

$$\tau^c \equiv \frac{\int_0^{\infty} dx \langle A(x)A(0) \rangle^2}{\langle A^2(0) \rangle^2}. \quad (5.3.15)$$

For $\tau \rightarrow \infty$, the product

$$\langle A(x - \tau)A(0) \rangle \langle A(x + \tau)A(0) \rangle \quad (5.3.16)$$

vanishes, and we have

$$\langle \bar{C}_A(\tau)^2 \rangle - \langle \bar{C}_A(\tau) \rangle^2 = 2 \langle A^2(0) \rangle^2 \frac{\tau^c}{\tau_0}. \quad (5.3.17)$$

Comparison of Eq. (5.3.14) with Eq. (5.3.17) shows that the *absolute* error in $C_A(\tau)$ changes only little with τ . As a consequence, the *relative* error in time correlation functions increases rapidly as $C_A(\tau)$ decays to 0. In this derivation, we have assumed that the total number of samples for each τ is equal; however, we typically have fewer samples for large τ , and hence for long times, the estimate of the error based on Eq. (5.3.17) is overly optimistic.

It should be stressed that the preceding error estimate is approximate, because it relies on the validity of the Gaussian approximation. Of course, deviations from the Gaussian approximation may occur. However, Jones and Mandadapu have presented evidence showing that, in a number of cases of practical interest, the Gaussian approximation is actually quite good [165].

5.3.3 Block averages

It is useful to know that the predicted variance of an observable quantity A is related to the integral of its auto-correlation function (Eq. (5.3.3)). But we do not really want to compute the auto-correlation function of every quantity that we measure, because that would require more effort than computing the average itself.

The standard approaches for estimating statistical errors in static quantities make use of some form of block-averaging. A block average is simply a time average over a finite time interval t_B (or, in the MC case, an average over n_B

MC cycles):

$$\bar{A}_B \equiv \frac{1}{t_B} \int_0^{t_B} dt A(t). \quad (5.3.18)$$

A simulation yields a sequence of values of A equally space in time. With these data, we can easily accumulate block averages for a given block length t_B . Our choice of t_B is arbitrary but typically such that the total simulation time τ_0 must contain a fair number of blocks (less than 10 becomes risky —see Fig. 5.5). Having accumulated these block averages, we can then average adjacent block averages to obtain averages for blocks of length $\tau_B \equiv (2, 3, \dots, n) \times t_B$. The number of such blocks is $n_B = \tau_0 / \tau_B$.

Now consider the variance in the block averages for a given value of τ_B :

$$\sigma^2(\bar{A}_B) = \frac{1}{n_B} \sum_{b=1}^{n_B} (\bar{A}_B - \langle A \rangle)^2. \quad (5.3.19)$$

If τ_B is much larger than the correlation time t_A^c , we know from Eq. (5.3.4) that

$$\sigma^2(\bar{A}_B) \approx \left(\langle A^2 \rangle - \langle A \rangle^2 \right) \frac{t_A^c}{\tau_B}. \quad (5.3.20)$$

But, as yet, we do not know t_A^c . We, therefore, compute the product

$$P(\tau_B) \equiv \tau_B \times \frac{\sigma^2(\bar{A}_B)}{\langle A^2 \rangle - \langle A \rangle^2}. \quad (5.3.21)$$

In the limit $\tau_B \gg t_A^c$, we know that $P(\tau_B)$ must approach t_A^c . We can now plot $P(\tau_B)$ versus τ_B (or, more conveniently, $1/P(\tau_B)$ versus $1/\tau_B$) and estimate the limit of $P(\tau_B)$ for $\tau_B \rightarrow \infty$. This yields our estimate of t_A^c and thereby our error estimate for \bar{A} . The above analysis of block averages provides a powerful tool to determine whether a simulation is long enough to yield a sufficiently accurate estimate of a particular quantity: if we find that $P(\tau_B)$ is still strongly dependent on τ_B in the limit $\tau_B = \tau$, then we know that τ_0 , the total length of the simulation, is too short.

A widely used implementation of the block-averaging approach for estimating the statistical error in averages obtained in a simulation was proposed by Flyvbjerg and Petersen [148]. Let A_1, A_2, \dots, A_L be L consecutive samples of some fluctuating quantity A . We want to estimate $\langle A \rangle$, the equilibrium average of A , and estimate its statistical error. The equilibrium average of A is estimated as the average over all L sampled data points

$$\langle A \rangle \approx \bar{A} \equiv \frac{1}{L} \sum_{i=1}^L A_i. \quad (5.3.22)$$

We also need an estimator of the variance of $\sigma^2(\bar{A})$. From standard statistical analysis (see e.g., [38]), we know that if we have L *uncorrelated* data points, the estimator of the true variance in the estimate of the average is

$$\sigma^2(\bar{A}) \approx \frac{1}{L(L-1)} \sum_{i=1}^L [A_i - \bar{A}]^2. \quad (5.3.23)$$

However, Eq. (5.3.23) would only be valid if all distinct samples were uncorrelated (of course, a sample is always correlated with itself). For the raw data, successive points are usually strongly correlated, which is why we cannot use Eq. (5.3.23) directly on the sequence of measure A -values.

To address this problem, ref. [148] applied repeated blocking of the data until successive blocks are effectively uncorrelated.

As every step in the blocking procedure is the same (averaging to adjacent blocks, thereby halving the number of blocks), it is best written as a recursive procedure: We start with the data set A_1, A_2, \dots, A_L , and apply one blocking step to generate a data set $A'_1, A'_2, \dots, A'_{L'}$, which has half the size of the original set:

$$A'_i = 0.5(A_{2i-1} + A_{2i}) \quad (5.3.24)$$

with

$$L' = 0.5L. \quad (5.3.25)$$

Note that the average of the new set \bar{A}' is the same as for the original one. However, the mean-squared error in A' is given by¹³

$$\sigma^2(A') = \langle A'^2 \rangle - \langle A' \rangle^2 = \frac{1}{L'} \sum_{i=1}^{L'} A'^2_i - \bar{A}'^2. \quad (5.3.26)$$

As long as successive blocks are significantly correlated, $\sigma^2(A')/(L' - 1)$ changes (increases) with every blocking step. But when the blocks are long enough to be effectively uncorrelated, we have

$$\frac{\sigma^2(A')}{L' - 1} \approx \text{Constant}. \quad (5.3.27)$$

This limiting value is used as our estimator of the variance in the measured average. Flyvbjerg and Petersen [148] also give an estimate of the statistical error in the error:

$$\sigma^2(\bar{A}) \approx \frac{\sigma^2(A')}{L' - 1} \pm \sqrt{\frac{2\sigma^4(A')}{(L' - 1)^3}}. \quad (5.3.28)$$

¹³ It is important to distinguish between the mean-squared error in the *data* (A') and in the variance in their average (\bar{A}').

It is important to have such an estimate, because if we perform too many blocking operations, the error in the error may increase again (see Fig. 5.5).

In Example 5, the method of ref. [148] method is used to compute the standard deviation of the energy in a **Molecular Dynamics** simulation. In Fig. 5.5 a typical plot of this estimate of the variance as a function of block size is shown. For small values of the number of blocking operations, M , the data are correlated, and as a consequence, the variance will increase if we perform the blocking operation. However, for very high values of M , we have only a few samples, and as a result, the statistical error in the error estimate will be large. The plateau in between gives us the value of $\sigma^2(A)$ we are interested in.

5.4 Questions and exercises

Question 15 (Heat capacity). *The heat capacity at constant volume can be calculated from fluctuations in the total energy in the canonical ensemble.*

$$C_v = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}.$$

1. Derive this equation.
2. In a constant- NVT MC simulation, one does not calculate fluctuations in the total energy E but in the potential energy U . Is it then still possible to calculate the heat capacity? Explain.
3. The heat capacity can be also calculated by differentiating the total energy of a system with respect to temperature. Discuss the advantages or disadvantages of this approach.
4. Eq. (5.1.8) relates the variance in the kinetic energy of a system at constant NVE to the heat capacity at constant volume (C_V). What constraint does this expression place on the value of C_V ? Explain this constraint in words.

Question 16 (Virial pressure of molecules with high frequency modes). *When computing the pressure of a molecular liquid, one can either consider the virial due only to forces between the molecular centers of mass, or due to all forces: inter-molecular and intra-molecular. The average value of the virial computed by both methods should be the same, but the statistical noise is different. To see this, consider an ideal gas of dimers consisting of two harmonically bound atoms at an average separation a . For convenience, we consider a 1d system. The dimers have an internal spring constant k_s .*

1. What is the average value of the force between the two atoms in a dimer.
2. What is the mean-squared value of the force at a temperature T .
3. What happens to the statistical noise in the virial as $k_s \rightarrow \infty$?

Question 17 (Correlation functions).

1. The value of the velocity autocorrelation function (VACF) at $t = 0$ is related to an observable quantity. Which one?

2. What is the limiting value of the VACF for $t \rightarrow \infty$? Motivate your answer.
3. What is the physical meaning of $\text{VACF} < 0$?
4. In MD simulations, we can compute the self-diffusion coefficient and the viscosity. However, for the same simulation time, the relative errors in the two quantities will be different. Explain.
5. Rather than measuring correlation functions, we can also measure power spectra. Suppose that we study a fluctuating observable $A(t)$. The numerically determined power spectrum of A is then defined as

$$G_A(\omega) = \lim_{M \rightarrow \infty} \frac{1}{M} |A_M(\omega)|^2$$

where

$$A_M((m-1)\Delta\omega) \equiv \sum_{n=1}^M A(n\Delta t) e^{i(m-1)\Delta\omega(n-1)\Delta t},$$

where Δt is the resolution in time, and $\Delta\omega = 2\pi/T$, where T is the simulation time ($T = M\Delta t$). If A is a Gaussian variable, then so is $A(\omega)$. But as a system in equilibrium is invariant under translation of the time origin, different Fourier components of A are uncorrelated. Use the arguments that were used to derive Eq. (5.3.14) to estimate the dependence of the relative error in the power spectrum on the length of the simulation.

- Explain your observation.
- Now consider what happens if you increase the $\Delta\omega$ of a longer simulation (say T_2) to match that of a shorter simulation (say T_1).

Exercise 10 (Molecular dynamics of a Lennard-Jones system). On the book's website, you can find a Molecular Dynamics (MD) program for a Lennard-Jones fluid in the NVE ensemble.

We have planted three errors in the code and, as a consequence, the total energy will not be conserved during a run.

1. Find the three errors in the code. Hint: there are two errors in the integration of the equations of motion, and one in the force calculation. See the file `system.inc` for documentation about some of the variables used in this code.
2. How is the temperature controlled in this program? What does it mean to "control the temperature": after all, in a simulation at constant NVE it is the total energy of the system that should be constant, whilst the temperature fluctuates.
3. To test the energy drift ΔE of the numerical integration algorithm for a given time step Δt after N integration steps, one usually computes [126]

$$\Delta E(\Delta t) = \frac{1}{N} \sum_{i=1}^{i=N} \left| \frac{E(0) - E(i\Delta t)}{E(0)} \right|.$$

In this equation, $E(x)$ is the total energy (kinetic + potential) of the system at time x . Change the program (only the central loop) in such a way that ΔE is

computed and make a plot of ΔE as a function of the time step. How does the time step for a given energy drift change with the temperature and density?

4. An important quantity of a liquid or gas is the so-called self-diffusion coefficient D . Here we the calculation of D in a three-dimensional system. There are two methods to calculate D :

- a. by integrating the velocity autocorrelation function (VACF):

$$\begin{aligned} D &= \frac{1}{3} \int_0^\infty \langle \mathbf{v}(t) \cdot \mathbf{v}(t+t') \rangle dt' \\ &= \frac{\int_0^\infty \sum_{i=1}^{i=N} \langle \mathbf{v}(i,t) \cdot \mathbf{v}(i,t+t') \rangle dt'}{3N} \end{aligned} \quad (5.4.1)$$

in which N is the number of particles and $\mathbf{v}(i,t)$ is the velocity of particle i at time t . To obtain independent samples of the VACF, one should choose t in such a way that independent time origins are taken, i.e., $t = ia\Delta t$, $i = 1, 2, \dots, \infty$ and $\langle \mathbf{v}(t) \cdot \mathbf{v}(t+a\Delta t) \rangle \approx \mathbf{0}$.

- b. by calculating the mean-squared displacement:

$$D = \lim_{t' \rightarrow \infty} \frac{\langle |\mathbf{x}(t+t') - \mathbf{x}(t)|^2 \rangle}{6t'} \quad (5.4.2)$$

One should be careful with the calculation of the mean-squared displacement when periodic boundary conditions are used. Why?

Modify the program in such a way that the self-diffusion coefficient can be calculated using both methods. Some modifications in codes provided in the SI are needed. It is not a good idea (although not wrong) to take every time step as a new time origin for the calculation of the mean-squared displacement and of the VACF. Explain. What is the unit of D in SI units? How can one transform D into dimensionless units?

5. For Lennard-Jones liquids, Naghizadeh and Rice [127] report the following equation for self-diffusion coefficient (dimensionless units, $T^* < 1.0$ and $p^* < 3.0$):

$${}^{10}\log(D^*) = 0.05 + 0.07p^* - \frac{1.04 + 0.1p^*}{T^*}. \quad (5.4.3)$$

Compare this phenomenological equation with your own simulations. How can one translate D^* to a diffusion coefficient in SI units?

6. For a system of particles interacting through a pair potential $u(r)$, the average potential energy U can be computed, if we know the radial distribution function $g(r)$. Derive an expression for $\langle U \rangle$ in terms of $u(r)$ and $g(r)$. Compare this calculation with a direct calculation of the average energy. A similar method can be used to compute the average pressure.
7. In the current version of the code, the equations of motion are integrated by the Verlet algorithm. Make a plot of the energy drift ΔE for the following integration algorithms:
 - Verlet
 - Velocity Verlet
 - Euler (**never use this algorithm in real simulations**).

Exercise 11 (Optimal truncation of Green-Kubo integrals). As explained in the text, Green-Kubo integrals should be truncated once the statistical noise is dominated by random noise.

1. Derive Eq. (5.2.42)
2. Compute the diffusion coefficient D of a [WF](#) system, using the [GK](#) relation between D and the integral of the [VACF](#).
3. Compute the variance in the computed D as a function of the truncation time and estimate where the integration should be truncated.

Chapter 6

Monte Carlo simulations in various ensembles

In its original form Molecular Dynamics simulation solves a discretized version of Newton's equations of motion, for which the total energy E and the total linear momentum \mathbf{P} are constants of motion. As a consequence, Molecular Dynamics simulations measure time averages in the constant- $NVE\text{-}\mathbf{P}$ ensemble, which is very similar to the more familiar micro-canonical constant- NVE ensemble (see [166]). In contrast, Metropolis-style Monte Carlo simulations probe the properties of a system at constant- NVT , (i.e., the canonical ensemble). The fact that these ensembles are different leads to observable differences in the statistical averages computed in Molecular Dynamics and Monte Carlo simulations. Most of these differences disappear in the thermodynamic limit and are already relatively small for systems of a few hundred particles. However, the choice of ensemble does make a difference when computing the amplitude of fluctuations in thermodynamic quantities. For instance: in the limit of vanishing time steps, where the numerical noise due to the discretization of the equations of motion disappears, the total energy in an MD simulation does not fluctuate. However, for systems at constant NVT , the energy *must* fluctuate, and is in fact related to the heat capacity of the system (see Eq. (5.1.5)). In Section 5.1.4 we discussed techniques to relate fluctuations in different ensembles [106].

In the years following the introduction of the MC and MD methods, methods were developed to perform such simulations in ensembles other than the original ones. In most cases, these techniques were first developed for Monte Carlo simulations. MD in other ensembles than constant- $NVE\text{-}\mathbf{P}$, came later: we will discuss them in Chapter 7. As MC simulations in ensembles other than NVT are more “natural”, we discuss them first. The flexibility of the MC method is such that it is usually straightforward to create an MC algorithm that will keep any desired set of intensive variables (P, T, μ_i, \dots) constant, as long as we fix at least one extensive variable.

Over the years, MC algorithms for a wide range of ensembles have been proposed: isobaric-isothermal, constant-stress-isothermal, grand canonical (i.e., constant μVT), semigrand (i.e., constant $\mu_A - \mu_B, N_A + N_B, V, T$), and even microcanonical [167–172]. We also discuss the so-called “Gibbs-ensemble” method, even though, strictly speaking, this method does not refer to a different ensemble. The microcanonical MC method [172] is briefly discussed in SI L.3.

As explained in Section 3.2, Markov-Chain MC is designed to visit different parts of configuration space with a frequency proportional to their Boltzmann weight. As a consequence, an unweighted average over all visited configurations converges to a Boltzmann-weighted average over all configuration space. In Section 3.2 we made use of the principle of *detailed balance* to demonstrate that the Metropolis algorithm visits point in configuration space according to their Boltzmann weights.

The condition of detailed balance is, in fact, too strong a condition to guarantee Boltzmann sampling and, as we will see in Chapter 13, powerful MC algorithms exist that satisfy *balance* but not *detailed balance*. However, if detailed balance is obeyed, we are *guaranteed* to have a correct sampling scheme. More importantly: it is usually relatively easy to construct algorithms that satisfy detailed balance, whereas proving balance is often more subtle.

For this reason, we shall discuss Monte Carlo simulations in other ensembles than constant NVT by imposing detailed balance. Valid non- NVT algorithms that only satisfy balance can be constructed, but considering them would complicate the discussion without any obvious benefit.

6.1 General approach

We will consider MC simulations for a number of different ensembles using the same approach over and over again. Doing so may seem repetitive, and indeed it is, but hopefully, it gets the point across that constructing a valid MC algorithm for a “new” ensemble is safe, as long as you simply follow the recipe.

In the following sections, we, therefore, use the following procedure to demonstrate the validity of our Monte Carlo algorithms:

1. Decide which distribution we want to sample. This distribution, denoted by \mathcal{N} , will depend on the details of the ensemble.
2. Impose the condition of detailed balance,

$$K(o \rightarrow n) = K(n \rightarrow o), \quad (6.1.1)$$

where $K(o \rightarrow n)$ is the flow of configuration o to n . This flow is given by the product of the probability of being in configuration o , the probability of generating configuration n , and the probability of accepting this move,

$$K(o \rightarrow n) = \mathcal{N}(o) \times \alpha(o \rightarrow n) \times \text{acc}(o \rightarrow n). \quad (6.1.2)$$

3. Determine the probabilities of generating a particular configuration.
4. Derive the condition that needs to be fulfilled by the acceptance rules.

6.2 Canonical ensemble

To start, let us apply the above recipe to the standard Metropolis scheme. In the canonical ensemble, the number of particles, temperature, and volume are

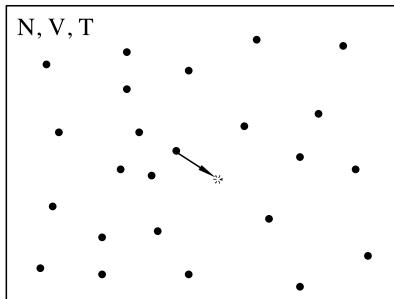


FIGURE 6.1 Canonical ensemble. The number of particles, volume, and temperature are constant. Shown is a Monte Carlo move in which a particle is displaced.

constant (see Fig. 6.1). For a system at constant NVT , the probability of finding configuration \mathbf{r}^N is proportional to the Boltzmann weight:

$$\mathcal{N}(\mathbf{r}^N) \propto \exp[-\beta \mathcal{U}(\mathbf{r}^N)]. \quad (6.2.1)$$

6.2.1 Monte Carlo simulations

Simulations in the canonical ensemble should sample the distribution given by Eq. (6.2.1). This can be done as follows scheme:

1. Select a particle at random and calculate the energy of configuration $\mathcal{U}(o)$.
2. Give this particle a random displacement (see Fig. 6.1) e.g.,

$$\mathbf{r}(o) \rightarrow \mathbf{r}(o) + \Delta(\mathcal{R} - 0.5), \quad (6.2.2)$$

where $\Delta/2$ is the maximum displacement. The value of Δ should be chosen such that the sampling scheme is optimal (see Section 3.4). The *trial* configuration is denoted n and its energy $\mathcal{U}(n)$.

3. The move is accepted with a probability (see Eq. (3.2.11))

$$\text{acc}(o \rightarrow n) = \min(1, \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}). \quad (6.2.3)$$

If rejected, the old configuration is kept.

An implementation of this basic Metropolis scheme is shown in Section 3.3 (Algorithms 1 and 2).

6.2.2 Justification of the algorithm

The generation of trial configurations according to Eq. (6.2.2) satisfies microscopic reversibility

$$\alpha(o \rightarrow n) = \alpha(n \rightarrow o) = \alpha. \quad (6.2.4)$$

Substitution of this equation in the condition of detailed balance (6.1.1), with Eq. (6.1.2) and substitution of the desired distribution (6.2.1) gives as condition for the acceptance rules

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}. \quad (6.2.5)$$

It is easy to verify that the acceptance rule (6.2.3) obeys this condition.

6.3 Isobaric-isothermal ensemble

The isobaric-isothermal (constant-*NPT*) ensemble is widely used in Monte Carlo simulations. This is not surprising because most real experiments are carried at constant pressure and temperature. An advantage of constant-*NPT* simulations is that they can be used to measure the equation of state of a model system for which the evaluation of the virial expression for the pressure is cumbersome. This is the case, for instance, for systems with non-pairwise additive interactions, but also for certain models of non-spherical hard-core molecules.

Finally, it is often convenient to use constant-*NPT* Monte Carlo to simulate systems in the vicinity of a first-order phase transition, because, given enough time, a system at constant pressure is free to transform completely into the state of lowest (Gibbs) free energy, whereas in a constant-*NVT* simulation the system may be kept at a density where, in a macroscopic system, it would separate into two bulk phases of different density, but is prevented from doing so due to finite-size effects.

Monte Carlo simulations at constant pressure were first described by Wood [167] in the context of a simulation study of two-dimensional hard disks. Although the method introduced by Wood is elegant, it is not readily applicable to systems with arbitrary continuous potentials. McDonald [168] was the first to apply constant-*NPT* simulations to a system with continuous intermolecular forces (a Lennard-Jones mixture), and the constant-pressure method of McDonald is now widely used. It is McDonald's method that we discuss next.

6.3.1 Statistical mechanical basis

We will derive the basic equations of constant-pressure Monte Carlo in a way that may appear unnecessarily complicated. However, this derivation has the advantage that the same framework can be used to introduce some of the other non-*NVT* Monte Carlo methods to be discussed later. For the sake of convenience we shall initially assume that we are dealing with a system of N identical atoms. The partition function for this system is given by

$$Q(N, V, T) = \frac{1}{\Lambda^{3N} N!} \int_0^L \cdots \int_0^L d\mathbf{r}^N \exp[-\beta \mathcal{U}(\mathbf{r}^N)]. \quad (6.3.1)$$

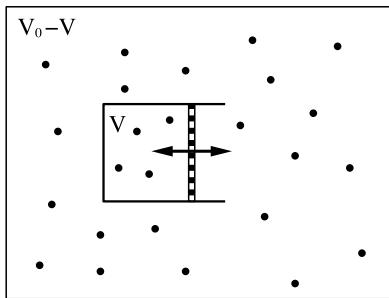


FIGURE 6.2 Ideal gas (m particles, volume $V_0 - V$) can exchange volume with an N -particle system (volume V).

It is convenient to rewrite Eq. (6.3.1) in a slightly different way. For convenience, we assume that the system is contained in a cubic box with diameter $L = V^{1/3}$. We now define scaled coordinates \mathbf{s}^N by

$$\mathbf{s}_i \equiv \frac{\mathbf{r}_i}{L} \quad \text{for } i = 1, 2, \dots, N. \quad (6.3.2)$$

If we now insert these scaled coordinates in Eq. (6.3.1), we obtain

$$Q(N, V, T) = \frac{V^N}{\Lambda^{3N} N!} \int_0^1 \cdots \int_0^1 d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.3.3)$$

In Eq. (6.3.3), we have written $\mathcal{U}(\mathbf{s}^N; L)$ to indicate that \mathcal{U} depends on the real rather than the scaled distances between the particles. The expression for the Helmholtz free energy of the system is

$$\begin{aligned} F(N, V, T) &= -k_B T \ln Q \\ &= -k_B T \ln \left(\frac{V^N}{\Lambda^{3N} N!} \right) - k_B T \ln \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)] \\ &= F^{\text{id}}(N, V, T) + F^{\text{ex}}(N, V, T). \end{aligned} \quad (6.3.4)$$

In the last line of this equation, we have identified the two contributions to the Helmholtz free energy as the ideal gas expression plus an excess part. We now consider the case that the system consists of two non-interacting subsystems with volume V and $V_0 - V$, with $V_0 \gg V$, and V_0 fixed. For the sake of visualization, we show the two systems in Fig. 6.2 as two bounded systems separated by a piston, although in reality the subsystems should be considered as completely separate and subject to periodic boundary conditions. We will refer to the system in volume $V_0 - V$ as the *reservoir*. We denote the total number of particles in the combined system by M . Of these, $M - N$ are in volume $V_0 - V$, and N are in volume V . The partition function of the combined system is simply

the product of the partition functions of the two (non-interacting) subsystems:

$$\begin{aligned} Q(N, M, V, V_0, T) &= Q(M, V_0 - V, T) \frac{V^N}{\Lambda^{3M} N!} \int d\mathbf{s}^N e^{-\beta \mathcal{U}(\mathbf{s}^N; L)} \quad (6.3.5) \\ &= e^{-\beta F_R(M, V_0 - V, T)} \frac{V^N}{\Lambda^{3M} N!} \int d\mathbf{s}^N e^{-\beta \mathcal{U}(\mathbf{s}^N; L)}, \end{aligned}$$

where F_R denotes the Helmholtz free energy of the reservoir. The total free energy of this combined system is $F^{\text{tot}} = -k_B T \ln Q(N, M, V, V_0, T)$. Now let us assume that the two subsystems can exchange volume. In that case, the volume V of the N -particle subsystem can fluctuate. The most probable value of V will be the one that minimizes the free energy of the combined system. The probability density $\mathcal{N}(V)$ that the N -particle subsystem has a volume V is given by¹

$$\mathcal{N}(V) = \frac{\exp[-\beta F_R(M, V_0 - V, T)] V^N \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]}{\int_0^{V_0} dV' \exp[-\beta F_R(M, V_0 - V', T)] V'^N \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L')]} \quad (6.3.6)$$

We now consider the limit that the size of the reservoir tends to infinity ($V_0 \rightarrow \infty, M \rightarrow \infty, (M - N)/V_0 \rightarrow \rho$). In that limit, a volume change of the small system does not change the pressure P_R of the reservoir. In other words, the large system works as a barostat for the small system. In that case, we can simplify Eqs. (6.3.5) and (6.3.6). Note that in the limit $V/V_0 \rightarrow 0$, we can write

$$\begin{aligned} F_R(M, V_0 - V, T) &= F_R(M, V_0, T) + V \left(\frac{\partial F_R(M, V_0 - V, T)}{\partial V} \right)_{V=0} \\ &= F_R(M, V_0, T) + P_R V. \end{aligned} \quad (6.3.7)$$

The combined partition function (6.3.5) can then be written as

$$Q(N, P, T) \equiv \frac{\beta P}{\Lambda^{3N} N!} \int dV V^N \exp(-\beta PV) \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)], \quad (6.3.8)$$

where we have included a factor βP to make $Q(N, P, T)$ dimensionless (this choice is not obvious —see footnote 1). This gives, for Eq. (6.3.6),

$$\mathcal{N}_{N, P, T}(V) = \frac{V^N \exp(-\beta PV) \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]}{\int_0^{V_0} dV' V'^N \exp(-\beta PV') \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L')]} \quad (6.3.9)$$

¹ Actually, this step is hard to justify. The reason is that there is no natural “metric” for the volume integration. Unlike the degeneracy of energy levels or the number of particles in a system, we cannot *count* volume. This problem has been addressed by several authors[173,174]. Attard [173] approaches the problem from an information-theory point of view and concludes that the integration variable should be $\ln V$, rather than V . In contrast, Koper and Reiss [174] aim to reduce the problem to one of counting the number of quantum states compatible with a given volume. They end up with an expression that is almost identical to the one discussed here.

In the same limit, the difference in free energy of the combined system and that of the reservoir in the absence of the N -particle subsystem is the well-known Gibbs free energy G :

$$G(N, P, T) = -k_B T \ln Q(N, P, T). \quad (6.3.10)$$

Eq. (6.3.9) is the starting point for constant- NPT Monte Carlo simulations. The idea is that the probability density to find the small system with volume V in a particular configuration of the N atoms, as specified by \mathbf{s}^N , is given by

$$\begin{aligned} \mathcal{N}(V; \mathbf{s}^N) &\propto V^N \exp(-\beta PV) \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)] \\ &= \exp\{-\beta[\mathcal{U}(\mathbf{s}^N, V) + PV - N\beta^{-1} \ln V]\}. \end{aligned} \quad (6.3.11)$$

We can now carry out Metropolis sampling on the reduced coordinates \mathbf{s}^N and the volume V .

In the constant- NPT Monte Carlo method, V is simply treated as an additional coordinate, and trial moves in V must satisfy the same rules as trial moves in \mathbf{s} ; in particular, we should maintain the microscopic reversibility of the underlying Markov chain. Let us assume that our trial moves consist of an attempted change of the volume from V to $V' = V + \Delta V$, where ΔV is a random number uniformly distributed over the interval $[-\Delta V_{\max}, +\Delta V_{\max}]$. In the Metropolis scheme such a random, volume-changing move will be accepted with the probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \exp\{-\beta[\mathcal{U}(\mathbf{s}^N, V') - \mathcal{U}(\mathbf{s}^N, V) + P(V' - V) - N\beta^{-1} \ln(V'/V)]\} \right). \quad (6.3.12)$$

Instead of attempting random changes in the volume itself, one might construct trial moves in the box length L [168] or in the logarithm of the volume [133]. Such trial moves are equally legitimate, as long as the microscopic reversibility of the underlying Markov chain is maintained. However, such alternative schemes result in a slightly different form for Eq. (6.3.12). The partition function (6.3.8) can be rewritten as

$$Q(N, P, T) = \frac{\beta P}{\Lambda^{3N} N!} \int d(\ln V) V^{N+1} \exp(-\beta PV) \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.3.13)$$

This equation shows that, if we perform a random walk in $\ln V$, the probability of finding volume V is given by

$$\mathcal{N}(V; \mathbf{s}^N) \propto V^{N+1} \exp(-\beta PV) \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.3.14)$$

This distribution can be sampled with the following acceptance rule:

$$\text{acc}(o \rightarrow n) = \min \left(1, \exp \left\{ -\beta [\mathcal{U}(\mathbf{s}^N, V') - \mathcal{U}(\mathbf{s}^N, V) + P(V' - V) - (N+1)\beta^{-1} \ln(V'/V)] \right\} \right). \quad (6.3.15)$$

6.3.2 Monte Carlo simulations

The frequency with which trial moves in the volume should be attempted depends on the efficiency with which volume space is sampled. If, as before, we use as our criterion of efficiency

$$\frac{\text{sum of squares of accepted volume changes}}{t_{\text{CPU}}}, \quad (6.3.16)$$

then it is obvious that the frequency with which we attempt moves depends on their cost. In general, a volume trial move will require that we recompute all intermolecular interactions. It, therefore, is comparable in cost to carrying out N trial moves on the molecular positions. In such cases, it is common practice to perform one volume trial move for every cycle of positional trial moves. Note that to guarantee detailed balance, rather than just balance, volume moves should be attempted with a probability $1/N$. However, attempting volume moves every N steps should satisfy balance, and that should also be OK.

The criteria determining the optimal acceptance of volume moves are no different from those for particle moves.

For one class of potential energy functions, volume trial moves are very cheap, namely those for which the total interaction energy can be written as a sum of powers of the interatomic distances,

$$\begin{aligned} U_n &= \sum_{i < j} \epsilon (\sigma / r_{ij})^n \\ &= \sum_{i < j} \epsilon [\sigma / (L s_{ij})]^n, \end{aligned} \quad (6.3.17)$$

or, possibly, a linear combination of such sums (the well-known Lennard-Jones potential belongs to the latter category). Note that U_n in Eq. (6.3.17) changes in a trivial way if the volume is modified such that the linear dimensions of the system change for L to L' :

$$U_n(L') = \left(\frac{L}{L'} \right)^n U_n(L). \quad (6.3.18)$$

Clearly, in this case, computing the probability of acceptance of a volume-changing trial move is extremely cheap. Hence such trial moves may be attempted with high frequency, for example, as frequently as particle moves. One should be careful when using the scaling property (6.3.18) if, at the same time,

Algorithm 11 (Basic NPT-ensemble simulation)

program mc_npt	Constant-NPT MC simulation
for $1 \leq \text{icycl} \leq \text{ncycl}$ do	perform ncycl MC cycles
$\text{ran} = \mathcal{R} \times (\text{npart} + 1) + 1$	\mathcal{R} is uniform random: $0 \leq \mathcal{R} < 1$
if $\text{ran} \leq \text{npart}$ then	
mcmove	perform particle displacement
else	
mcvol	perform volume change
endif	
if $\text{icycl} \% \text{nsamp} == 0$ then	
sample	sample observables
endif	
enddo	
[...]	Compute averages of observables
end program	

Specific Comments (for general comments, see p. 7)

1. This algorithm ensures that, during each MC step, detailed balance is obeyed and that, per cycle, we carry out (on average) npart attempts to move particles, and one attempt to change the system volume.
2. Function **mcmove** attempts to displace a randomly selected particle (Algorithm 2), and function **mcvol** attempts to change the volume (Algorithm 12), and function **sample** samples observables every nsampth cycle.

one uses a cutoff (say r_c) on the range of the potential. Use of Eq. (6.3.18) implicitly assumes that the cutoff radius r_c scales with L , such that $r'_c = r_c(L'/L)$. The corresponding tail correction to the potential (and the virial) should also be recomputed to take into account both the different cutoff radii and the different densities of the system. Algorithms 2, 11, and 12 show the basic structure of a simulation in the *NPT* ensemble.

It is useful to compute the virial pressure as a diagnostic tool, during a constant pressure simulation. On average, the virial pressure should be equal to the applied pressure, as can be demonstrated as follows: We note that the virial pressure $P_v(V)$ of an N -particle system at volume V is equal to

$$P_v(V) = - \left(\frac{\partial F}{\partial V} \right)_{NT} . \quad (6.3.19)$$

In an isothermal-isobaric ensemble, the probability density \mathcal{N} of finding the system with volume V is equal to

Algorithm 12 (Trial move in $\ln V$)

function mcvol	
vo=box**3	attempts volume change
eno=toterg(vo)	vo is current volume
lnvn=log(vo)+(\mathcal{R} -0.5)*d1nv	total energy old conf.
vn=exp(lnvn)	attempt random step in $\ln V$
boxn=vn***(1/3)	vn is trial volume
for 1 ≤ i ≤ npart do	new box length
x(i)=x(i)*boxn/box	rescale center of mass
enddo	
enn=toterg(vn)	total energy trial conf.
arg=-beta*((enn-eno)+p*(vn-vo)	
+ -(npart+1)*log(vn/v0)/beta)	appropriate weight function!
if $\mathcal{R} \geq \exp(\text{arg})$ then	acceptance rule (6.2.3)
for 1 ≤ i ≤ npart do	REJECTED
x(i)=x(i)*box/boxn	restore the old positions
enddo	
endif	
end function	

Specific Comments (for general comments, see p. 7)

1. A random walk in $\ln V$ is performed using acceptance rule (6.3.15).
2. The function **toterg** calculates the total energy, first for volume **vo**, later for volume **vn**. This function is not shown explicitly: it is similar to Algorithm 5. Usually the energy of the old configuration is known; therefore this function is called only once.
3. For spherical particles interacting through (sums of) power-law potentials (see Eq. (6.3.18)), the old and new energies are related by simple scaling factors and volume-changing trial moves become very cheap.

$$\mathcal{N}(V) = \frac{\exp[-\beta(F(V) + PV)]}{Q(NPT)}, \quad (6.3.20)$$

where

$$Q(NPT) \equiv \beta P \int dV \exp[-\beta(F(V) + PV)]. \quad (6.3.21)$$

The average value of the virial pressure is given by

$$\begin{aligned} \langle P_v \rangle &= -\frac{\beta P}{Q(NPT)} \int dV (\partial F(V)/\partial V) \exp[-\beta(F(V) + PV)] \\ &= \frac{\beta P}{Q(NPT)} \int dV \beta^{-1} (\partial \exp[-\beta F(V)]/\partial V) \exp(-\beta PV) \end{aligned}$$

$$\begin{aligned}
 &= \frac{P}{Q(NPT)} \int dV P \exp[-\beta(F(V) + PV)] \\
 &= P,
 \end{aligned} \tag{6.3.22}$$

which is the applied pressure. The third line in this equation follows from partial integration.

Thus far, we have limited our discussion of Monte Carlo at constant pressure to one-component, atomic systems. Extension of the technique to molecular systems and to mixtures is straightforward. However, in the case of molecular systems, it is important to note that only the coordinates of the centers-of-mass of the molecules should be scaled in a volume move, never the relative positions of the constituent atoms in the molecule. This has one practical consequence, namely, that the simple scaling relation (6.3.18) can not be used in molecular systems with site-site interactions. The reason is that, even if the center-of-mass distances between molecules scale linearly with the system size, the site-site separations do not.

6.3.3 Applications

Example 8 (Equation of state of the Lennard-Jones fluid). Simulations at constant pressure can also be used to determine the equation of state of a pure component. In such a simulation, the density is determined as a function of the applied pressure and temperature. Fig. 6.3 shows that, for the Lennard-Jones fluid, the results of an *NPT* simulation agree with those obtained in Case Study 1.

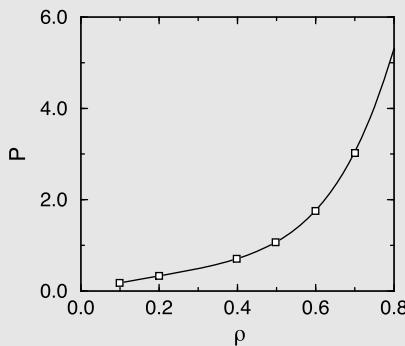


FIGURE 6.3 Equation of state of the Lennard-Jones fluid as obtained from *NPT* simulations; isotherms at $T = 2.0$. The solid line is the equation of state of Johnson et al. [83] and the squares are the results from the simulations ($N = 108$).

In simulations of realistic models of molecular liquids, it is important to verify that the model fluid has the same density as the real fluid under atmospheric pressure. Using *NVT* simulations, one would require several

simulations to determine the density for which the pressure is approximately 1 atm. In an *NPT* simulation, one would obtain this result in a single simulation. In reduced units, atmospheric pressure is typically much less than one. Hence, a good first estimate of the liquid density at atmospheric pressure follows from an *NPT* simulation at $P = 0$.^a

The Fortran code to generate this Example can be found in the online-SI, Case Study 7.

^a Strictly speaking, the liquid is only metastable at $P = 0$. However, as the nucleation barrier for bubble formation is typically much larger than $k_B T$, the lifetime of this metastable state is usually much longer than the simulation time.

Example 9 (Phase equilibria from constant-pressure simulations). In Examples 1 and 7 *NVT* or *NPT* simulations are used to determine the equation of state of a pure substance. In principle, we can construct the liquid-vapor coexistence curve by first fitting the simulation data to an analytical expression, and then determining the liquid and vapor densities for which P , T , and μ are the same. Although this route to locate a coexistence curve is quite general, it requires many simulations. More efficient ways to determine the vapor-liquid coexistence curve are discussed in section 6.6. However, in cases where the vapor pressure of the liquid is very small ($\ll 1$ in reduced units), we can determine the density of the coexisting liquid by performing simulations at zero pressure.

To carry out an *NPT*-simulation at zero pressure it is best to start with a liquid density above the estimated coexistence density. During the simulation, the system will quickly reach the (metastable) density for which $P = 0$. However, starting from a lower density is not recommended as the system might then expand without bound.

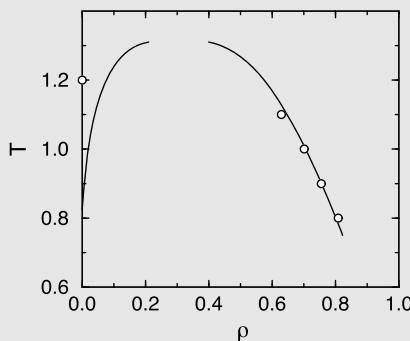


FIGURE 6.4 Vapor-liquid coexistence curve of the Lennard-Jones fluid; the solid curve was computed using the Johnson equation of state [83]. The circles indicate the average liquid densities obtained from *NPT* simulations at $P = 0$. As can be seen from the figure, the zero-pressure method becomes less reliable for $T > 1.2$.

Fig. 6.4 shows that the densities obtained in a zero-pressure simulation agree well with the true equation of state of the Lennard-Jones fluid, even up to a reduced temperature of 1, but not as the equilibrium density of the vapor becomes appreciable: for $T > 1$, the zero-pressure simulations predict liquid densities that are too low. Moreover, as the critical temperature T_c is approached, the surface tension tends to zero and, as a consequence, bubble nucleation becomes more likely. Under those conditions, the metastable liquid at $P = 0$ is increasingly likely to evaporate during a simulation. In short: do not use $P = 0$ simulations close to T_c . On a more positive note: not too close to the critical temperature, a reasonable estimate of the liquid density can be obtained by carrying out *NPT* simulations at $P = 0$.

The Fortran code to generate this Example can be found in the online-SI, Case Study 8.

As explained below Eq. (6.3.19), $\mathcal{N}(V)$, the probability density of finding a system with volume V , is proportional to $\exp[-\beta(F(V) + PV)]$. For a given temperature, this probability density can, in principle, be obtained from a single constant-pressure simulation by constructing a histogram of the number of times a certain volume V is observed during the simulation.

Once we have obtained $\mathcal{N}(V)$, we can obtain the coexistence volumes (and hence densities) from a common-tangent construction on:

$$\ln \mathcal{N}(V) = \beta[F(V) + PV].$$

The idea behind this widely used construction is that if two points on the curve $\ln \mathcal{N}(V) \text{ vs } V$ have the same slope, then they have the same pressures. And if they have the same intercept, then they have the same chemical potentials. Hence, if two points have a common tangent, then these points have the same values for μ , P , and T , and hence they are in equilibrium. In practice the histogram method for deriving $F(V)$ from $\mathcal{N}(V)$ only works near the critical point [163,175–177] unless special sampling techniques are used (see Section 8.6.6). However, very close to the critical point, the common-tangent approach also fails, due to irreducible fluctuation effects.

6.4 Isotension-isothermal ensemble

The *NPT*-MC method is robust for homogeneous fluids. However, for solids and for inhomogeneous systems, isotropic volume changes in the simulation box may not be sufficient to ensure equilibration. For instance, for non-cubic crystals, the equilibrium shape of the unit cell may change with temperature. If the shape of the simulation box is fixed, stresses will develop in the solid as the temperature is changed. Fixing the box shape is even more of a problem in the case of phase transformations from one crystalline phase to another. It should be noted that, in general, solid-solid phase transitions may involve a change in the

number of particles in the unit cell, or other drastic changes, and cannot be studied in a simulation with the same number or arrangement of unit cells. However, certain solid-solid transitions, called displacive phase transformations, involve a change in the shape of the crystal unit cell, with only minor displacements of the particles in the unit cell. Even so, most displacive phase transitions cannot be studied in a simulation if the shape of the unit cell is fixed.

To study “displacive” changes in the shape of the crystal unit cell, the shape of the simulation box should be free to change in such a way that the solid can remain stress-free without creating defects.

In the context of MD simulations, this problem was first tackled by Parrinello and Rahman [178,179], who developed an extension of the constant-pressure Molecular Dynamics technique introduced by Andersen [180] to simulate solids under constant stress. In fluids, the stress σ is just minus the hydrostatic pressure, but in solids σ can have six independent components: three compressive/tensile stresses and three shear stresses.

The extension of the Parrinello-Rahman method to Monte Carlo simulations by Najafabadi and Yip [169] is straightforward. In fact, the approach of ref. [169] is simpler than the original MD approach.

To explain the constant stress method, it is useful to generalize the coordinate scaling of the conventional constant pressure *MC* to the case for non-cubic (parallelepiped) boxes. In that case, the transformation between \mathbf{s} and \mathbf{r} is given by a matrix \mathbf{h} :

$$r_\alpha = h_{\alpha\beta} s_\beta. \quad (6.4.1)$$

The volume of the simulation box V is equal to $|\det \mathbf{h}|$. If the simulation box is a cube, the transformation matrix \mathbf{h} is a diagonal matrix, with all diagonal elements equal to L and Eq. (6.4.1) becomes equivalent to Eq. (6.3.2).

Without loss of generality, we can choose \mathbf{h} to be a symmetric matrix with six independent components.² Changing the independent elements of the \mathbf{h} -matrix causes this parallelepiped to change its size and/or shape.³

The difference between a simulation at constant stress, and one at constant pressure is that the term PV in the Boltzmann factor is replaced by $V_0 \text{Tr } \epsilon : \sigma$, where $V_0 = |\text{Det } \mathbf{h}_0|$ is the volume of the undeformed box [169]. The correct description for the case of larger strains, where nonlinear effects may be important, is given in ref. [181,182].

In a simulation at constant stress, we sample the independent elements of the \mathbf{h} -matrix, in addition to the scaled particle coordinates. As there is no natural metric for deformations, it would be equally realistic, but not completely

² \mathbf{h} must be symmetric because any anti-symmetric component would correspond to a rotation that would not change the physical state of the system.

³ To make contact with the notation used in the theory of elasticity, we can write \mathbf{h} as $\mathbf{h} \equiv \mathbf{h}_0 \cdot \mathbf{h}_0^{-1} \cdot \mathbf{h}$ and $\mathbf{h}_0^{-1} \cdot \mathbf{h} \equiv [\mathbf{I} + \boldsymbol{\epsilon}]$, where \mathbf{h}_0 describes the shape of the undeformed solid, and $\boldsymbol{\epsilon}$ is the so-called strain tensor.

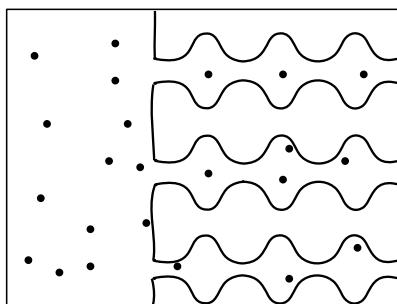


FIGURE 6.5 Adsorbent (for example, a zeolite) in direct contact with a gas.

equivalent, to sample the elements of the metric tensor $G = h^T h$, where h^T is the transpose of h .

In the case of an isotropic (hydrostatic) applied pressure, the constant-stress Monte Carlo method is almost equivalent to constant-pressure Monte Carlo.⁴

Elastic constants

One obvious application of the constant stress method is the measurement of elastic constants $C_{\alpha\beta\gamma\delta}$ of a solid, using $\sigma_{\alpha\beta} = C_{\alpha\beta\gamma\delta} \epsilon_{\gamma\delta}$. In practice, constant-stress simulations measure the elastic compliance, which is the inverse of the tensor $C_{\alpha\beta\gamma\delta}$. For more details on elastic constants, see Appendix F.4.

6.5 Grand-canonical ensemble⁵

The intensive thermodynamic variables P , T and the μ_i of all components, are linearly dependent (Eq. (2.1.17)). In particular, at constant temperature, we have

$$Nd\mu = VdP. \quad (6.5.1)$$

This means that changing P at constant temperature, will change the density of the system, and thereby μ , or conversely, changing μ changes P . As a consequence, we can use either P or μ as an intensive control variable. Above, we discussed simulation techniques where P is the control variable, in which case the volume can vary and the number of particles is fixed. We could also use μ as the control variable, in which case N can vary and V is fixed. But, the fact that we *can* do simulations at constant μ does not imply that we should. It turns out, however, that there are many situations where we cannot use the constant NPT method, but we *can* carry out simulations at constant μVT . For instance,

⁴ However, one should never use the constant-stress method for uniform fluids, because the latter offer no resistance to the volume-conserving deformations of the unit box and very strange (flat, elongated, etc.) box shapes may result. Such strongly deformed simulation boxes tend to exhibit appreciable finite-size effects.

⁵ Familiarity with the material in Chapter 8 can be useful when reading this section.

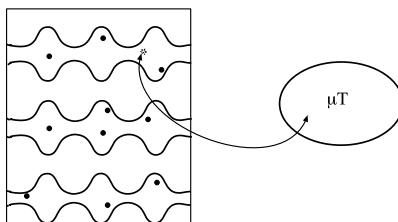


FIGURE 6.6 Adsorbent in contact with a reservoir that imposes constant chemical potential and temperature by exchanging particles and energy.

we cannot use the standard NPT method for molecules inside a tube, slit, or porous matrix; in a porous material, volume-changing moves would attempt to change the volume of the matrix, which typically is barely compressible: as the matrix would carry the stress, the molecules inside the pores would not “feel” the imposed pressure. Of course, we could bring the porous matrix in contact with a fluid reservoir where we *can* impose the pressure (see Fig. 6.5). However, in that case, we would be introducing interfaces in the system, which typically cause serious finite-size effects. Later we will encounter examples where we are specifically interested in the properties of such interfaces.

Note that, in general, the pressure of a fluid inside a porous matrix such as a zeolite, is thermodynamically *not* well-defined. In contrast, the chemical potential remains well-defined, and we can relate this chemical potential to the pressure of the fluid outside the porous medium. In the case where we simulate a porous medium in contact with a reservoir, we should expect the equilibration to be slow. In fact, due to slow diffusion in porous media, equilibration in real adsorption experiments may take minutes, hours, or longer, depending on the type of gas molecules. And the same slow diffusion will also slow down simulations of adsorption.

By carrying out simulations at constant μ , V , and T , we can avoid most of the problems mentioned above as particles can be added/removed anywhere inside the porous medium, be it that the constant μVT technique becomes less competitive at high (liquid) densities. In short: for studying adsorption phenomena, the constant μVT (“grand-canonical”) ensemble is to be preferred over other ensembles.

For convenience, we will consider the constant μVT ensemble of a periodically repeated system (see Fig. 6.6). The number of adsorbent particles in the system can be changed by addition and deletion. The volume of the system is kept fixed and the temperature and chemical potential are imposed. Importantly, the number of particles is allowed to fluctuate during a simulation.

6.5.1 Statistical mechanical basis

The grand-canonical Monte Carlo method was pioneered by Norman and Filinov [170] for the case of classical fluids, and extended by a number of other

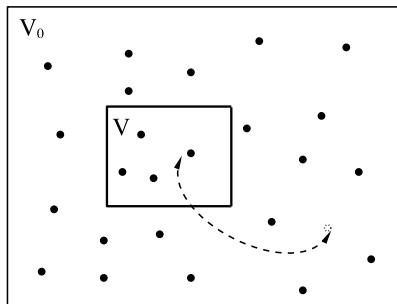


FIGURE 6.7 A reservoir with volume V_0 can exchange particles with a system with volume $V \ll V_0$). We denote the (fluctuating) number of particles in the system by N . For that value of N , the reservoir contains $M - N$ particles.

groups [171,183–190]. To explain the statistical-mechanical basis of the grand-canonical Monte Carlo technique, we return to Eq. (6.3.5) of Section 6.3. Eq. (6.3.5) expresses the partition function of a combined system of N interacting particles in volume V and $M - N$ molecules in a *reservoir* volume V_0 :

$$Q(N, M, V, V_0, T) = e^{-\beta F_R(M - N, V_0, T)} \frac{V^N e^{\beta \mu N}}{\Lambda^{3N} N!} \int_V ds^N e^{-\beta \mathcal{U}(s^N; L)}. \quad (6.5.2)$$

We will allow the system and the reservoir to exchange particles (see Fig. 6.7). In the limit $V_0 \rightarrow \infty$, $M \rightarrow \infty$, $(M - N)/V_0 \rightarrow \rho$, we can write

$$\begin{aligned} F_R(M - N, V_0, T) &= F_R(M, V_0, T) + N \left(\frac{\partial F_R(M - N, V_0, T)}{\partial N} \right)_{N=0} \\ &= F_R(M, V_0, T) - \mu N. \end{aligned} \quad (6.5.3)$$

The combined partition function, usually denoted by the symbol Ξ , can then be written as

$$\Xi(\mu, V, T) \equiv \sum_{N=0}^{\infty} \frac{V^N \exp(\beta \mu N)}{\Lambda^{3N} N!} \int ds^N \exp \left[-\beta \mathcal{U}(s^N; L) \right], \quad (6.5.4)$$

where we have omitted a constant factor $\exp[-\beta F_R(M, V_0, T)]$.

Eq. (6.5.4) can be used as the starting point for grand-canonical simulations. However, unlike the pressure or temperature imposed by a reservoir, the chemical potential is only known up to an arbitrary constant. It is therefore better to express the chemical potential in terms of an observable equilibrium property of the reservoir. As the precise nature of the reservoir is unimportant, we consider a reservoir that contains an ideal gas of the same molecules as the system in volume V . We can then define the fugacity f of a molecular system as the number density in the reservoir, where different molecules do not interact, but where all

intra-molecular interactions are unchanged.⁶ In books on statistical thermodynamics, the fugacity is usually viewed as a convenient, but purely theoretical concept. However, in simulations, we could really switch off intermolecular interactions in a reservoir, if we would wish to do so.

Let us now relate μ to f . The full expression for the chemical potential of a molecular gas at density ρ is:

$$\mu^g = k_B T \ln \left(\frac{\Lambda^3 \rho}{q_{\text{int}}(T)} \right), \quad (6.5.5)$$

where $q_{\text{int}}(T)$ is the intra-molecular part of the molecular partition functions due to rotations, vibrations, etc.

Below, we first consider the situation where the inter-molecular interactions do not depend on the internal degrees of freedom of the molecules. After that, we consider the case where the intermolecular interactions depend on some of the internal degrees of freedom.

Uncoupled internal degrees of freedom

If the internal degrees of freedom of a molecule do not affect its intermolecular interactions, we can use the fact that we can write $\mu^{\text{id gas}}$ as

$$\mu^{\text{id gas}} = k_B T \ln \left(\frac{\Lambda^3}{q_{\text{int}}(T)} \right) + k_B T \ln \rho^{\text{id gas}} \equiv \mu^\phi + k_B T \ln \rho^{\text{id gas}}. \quad (6.5.6)$$

Similarly, the chemical potential of the interacting system is

$$\mu^{\text{sys}} = \mu^\phi + k_B T \ln \rho^{\text{sys}} + \mu^{\text{ex}}, \quad (6.5.7)$$

and hence, for an ideal gas with density (= fugacity) f in equilibrium with an interacting system, we have:

$$k_B T \ln \rho^{\text{sys}} + \mu^{\text{ex}} = k_B T \ln f. \quad (6.5.8)$$

The advantage of the above expression is that the irrelevant terms in μ^ϕ have dropped out.⁷

We can now re-write the grand-canonical partition function as

$$\Xi(f, V, T) \equiv \sum_{N=0}^{\infty} \frac{(fV)^N}{N!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N); L], \quad (6.5.9)$$

⁶ Normally, the fugacity is defined in terms of the *pressure* of a hypothetical ideal gas of the same particles that would be in equilibrium with the system. However, as the ideal-gas pressure obeys $P_{\text{id}} = \rho k_B T$, we may just as well use the density.

⁷ However, the term μ^ϕ becomes important when we consider chemical equilibria.

and the corresponding probability density for a particular N -particle configuration is

$$\mathcal{N}_{f,V,T}(\mathbf{s}^N; L) \propto \frac{(fV)^N}{N!} \exp[-\beta \mathcal{U}(\mathbf{s}^N); L]. \quad (6.5.10)$$

Now consider a trial move in which we attempt to move a particle from the reservoir to an arbitrary position in volume V . We should make sure that we construct an underlying Markov chain satisfying microscopic reversibility. Moreover, we choose to make trial moves to add and remove a particle equally likely. The probability of acceptance of a trial move in which we move a particle to or from volume V must be chosen such that states with $N+1$ and N particles are visited with the relative probabilities given by the ratio of the corresponding probability densities in Eq. (6.5.10):

$$\frac{\mathcal{N}_{f,V,T}(\mathbf{s}^{N+1}; L)}{\mathcal{N}_{f,V,T}(\mathbf{s}^N; L)} = \frac{fV}{(N+1)} \exp \left\{ -\beta [\mathcal{U}(\mathbf{s}^{N+1}; L) - \mathcal{U}(\mathbf{s}^N; L)] \right\}. \quad (6.5.11)$$

Note that in this probability ratio all explicit reference to the reservoir has disappeared. Equally important is the fact that Planck's constant, which was hidden inside the thermal De Broglie wavelength Δ , has also disappeared, as it should for a classical simulation.

Coupled internal degrees of freedom

For many molecular systems, the inter-molecular interactions depend on the internal degrees of freedom of a molecule. For example, the interaction between molecules will typically depend on their orientation, or on the molecular conformation, e.g., *cis* or *trans*. In that case, we can still use the Grand-canonical scheme described above. However, we then should sample molecules to be inserted from the reservoir according to the Boltzmann weight associated with their internal energy. We discuss this case in Section 6.5.3.

6.5.2 Monte Carlo simulations

In a grand-canonical simulation, we have to sample the distribution (6.5.10). Acceptable trial moves are

1. *Displacement of particles.* A particle is selected at random and given a new conformation: for example, in the case of atoms, a random displacement. This move is accepted with a probability

$$\text{acc}(s \rightarrow s') = \min \left(1, \exp \{-\beta [\mathcal{U}(s') - \mathcal{U}(s)]\} \right). \quad (6.5.12)$$

2. *Insertion and removal of particles.* A particle is inserted at a random position or a randomly selected particle is removed. From Eq. (6.5.11) it follows that

Algorithm 13 (Basic grand-canonical ensemble simulation)

program mc_gc	
for 1 ≤ icycl ≤ ncycl do	constant- <i>f</i> VT MC-code perform ncycl MC cycles
ran=int($\mathcal{R} \times (\text{npav} + \text{nexc}) + 1$)	
if ran ≤ npav then	
mcmove	attempt to move a particle
else	
mcexc	attempt particle exchange with the reservoir
endif	
if icycl % nsamp == 0 then	
sample	sample observables
endif	
enddo	
[...]	Compute averages of observables
end program	

Specific Comments (for general comments, see p. 7)

1. By selecting particles at random, the algorithm satisfies microscopic reversibility, as forward and backward trial moves are equally likely. The algorithm as a whole satisfies detailed balance. Per cycle, we perform on average npav attempts to move a particle, and nexc attempts to exchange particles with the reservoir.
2. Function **mcmove** performs a trial displacement (Algorithm 2), function **mcexc** attempts exchanging a particle with a reservoir (Algorithm 14), and function **sample** samples observables of interest every nsamp cycles.
3. In the Grand-Canonical ensemble, the state of a system is commonly characterized by μ , V , T . However, for reasons explained in Section 6.5.1 (Eq. (6.5.8)), we use the fugacity f rather than the chemical potential μ as a control variable in **mcexc**.

a valid (Metropolis-style) acceptance rule for particle insertion would be

$$\text{acc}(N \rightarrow N+1) = \min \left[1, \frac{fV}{(N+1)} \exp\{-\beta[\mathcal{U}(N+1) - \mathcal{U}(N)]\} \right] \quad (6.5.13)$$

and the removal of a particle is accepted with a probability

$$\text{acc}(N \rightarrow N-1) = \min \left[1, \frac{N}{fV} \exp\{-\beta[\mathcal{U}(N-1) - \mathcal{U}(N)]\} \right]. \quad (6.5.14)$$

Algorithm 13 shows the basic structure of a simulation in the grand-canonical ensemble.

Algorithm 14 (Attempt to exchange a particle with a reservoir)

<pre> function mcexc if $\mathcal{R} < 0.5$ then if npart == 0 return o=int(npart*\mathcal{R})+1 eno = ener(x(o),o) arg=npart*exp(beta*eno) + /(f*vol) if $\mathcal{R} < \text{arg}$ then x(o)=x(npart) npart=npart-1 endif else xn=\mathcal{R}*box enn = ener(xn, npart+1) arg=f*vol*exp(-beta*enn) + /(npart+1) if $\mathcal{R} < \text{arg}$ then x(npart+1)=xn npart=npart+1 endif endif end function</pre>	attempt to exchange a particle with a reservoir decide to remove or add a particle can only remove particles if npart>0 select a particle to be removed energy particle o acceptance rule (6.5.14) accepted: remove particle o new particle at a random position energy of particle inserted at xn acceptance rule (6.5.13) accepted: add new particle
---	---

Specific Comments (for general comments, see p. 7)

- f denotes the fugacity of the molecules, which can be interpreted as the density of a hypothetical ideal gas of the same molecules, which acts as the reservoir.*
- The function **ener** calculates the energy of a particle at a given position. For addition, we give the particle the label npart+1, which is the label it will keep if the move is accepted.*

Justification of the algorithm

It is instructive to demonstrate that the acceptance rules (6.5.12)–(6.5.14) indeed lead to a sampling of distribution given by Eq. (6.5.10). Consider a move in which we start with a configuration with N particles and move to a configuration with $N + 1$ particles by inserting a particle in the system. Recall that we have to demonstrate that detailed balance is obeyed:

$$K(N \rightarrow N + 1) = K(N + 1 \rightarrow N), \quad (6.5.15)$$

with

$$K(N \rightarrow N + 1) = \mathcal{N}(N) \times \alpha(N \rightarrow N + 1) \times \text{acc}(N \rightarrow N + 1). \quad (6.5.16)$$

In Algorithm 13, at each Monte Carlo step the probability that an attempt is made to remove a particle is equal to the probability of attempting to add one:

$$\alpha_{\text{gen}}(N \rightarrow N + 1) = \alpha_{\text{gen}}(N + 1 \rightarrow N), \quad (6.5.17)$$

where the subscript “gen” refers to the fact that α measures the probability to generate this trial move. Substitution of this equation together with Eq. (6.5.10) into the condition of detailed balance gives

$$\begin{aligned} \frac{\text{acc}(N \rightarrow N + 1)}{\text{acc}(N + 1 \rightarrow N)} &= \frac{(fV)^{N+1} \exp[-\beta\mathcal{U}(\mathbf{s}^{N+1}); L]}{(N+1)!} \\ &\quad \times \frac{N! \exp[\beta\mathcal{U}(\mathbf{s}^N)]}{(fV)^N} \\ &= \frac{fV}{N+1} \exp\{-\beta[\mathcal{U}(\mathbf{s}^{N+1}; L) - \mathcal{U}(\mathbf{s}^N); L]\}. \end{aligned}$$

It is straightforward to show that the acceptance rules (6.5.13) and (6.5.14) obey this condition.

Illustration 4 (Adsorption isotherms of zeolites). Zeolites are crystalline inorganic polymers that form a three-dimensional network of micropores (see Fig. 6.8). These pores are accessible to various guest molecules. The large internal surface, the thermal stability, and the presence of thousands of acid sites make zeolites an important class of catalytic materials for petrochemical applications. For a rational use of zeolites, it is essential to have a detailed knowledge of the behavior of the adsorbed molecules inside the pores of the zeolites. Since this type of information is very difficult to obtain experimentally, simulations are an attractive alternative. One of the first attempts to study the thermodynamic properties of a molecule adsorbed in a zeolite was made by Stroud et al. [191]. Reviews of the various applications of computer simulations of zeolites can be found in [192, 193].

Besides zeolites, many other porous materials exist with many interesting properties. In ref. [194] a review is given of phase separations in these materials.

For small absorbents such as methane or noble gases, grand-canonical Monte Carlo simulations can be applied to calculate the adsorption isotherms in the various zeolites [195–201]. An example of an adsorption isotherm of methane in the zeolite silicalite is shown in Fig. 6.9. These calculations are based on the model of Goodbody et al. [197]. The agreement with the experimental data is very good, which shows that for these well-characterized systems simulations can give data that are comparable with experiments.

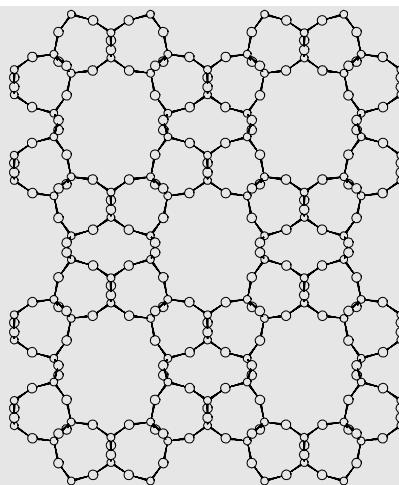


FIGURE 6.8 Example of a zeolite structure (Theta-1), the pore size is approx. $4.4 \times 5.5 \text{ \AA}^2$. The Si atoms have four bonds and the O atoms two.

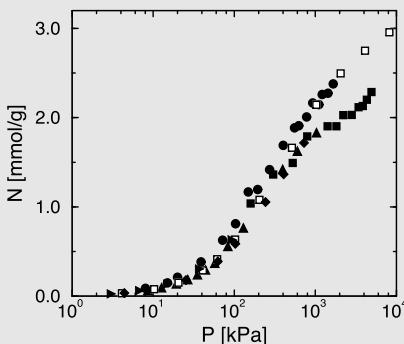


FIGURE 6.9 Adsorption isotherms of methane in silicalite, showing the amount of methane adsorbed as a function of the external pressure. The black symbols are experimental data (see [202] for details). The open squares are the results of grand-canonical simulations using the model of [197].

For long-chain alkanes (butane and longer) it is very difficult to perform a successful insertion; in almost all attempts one of the atoms of the molecule will overlap with one of the atoms of the zeolite. As a consequence, the number of attempts has to be astronomically large to have a reasonable number of successful exchanges with the reservoir. In Chapter 12 we show how this problem can be solved.

6.5.3 Molecular case

To discuss grand-canonical MC for molecular systems with internal degrees of freedom that affect the intermolecular interactions, it is useful to treat all

molecules in the same internal state i of the molecule as if they were a separate species with intra-molecular energy ϵ_i . As the ideal gas reservoir with a total of M molecules is assumed to be in thermal equilibrium, the number of copies of state i is given by $N_i^0 = M e^{-\beta \epsilon_i} / q(T)$, where $q(T) \equiv \sum_i e^{-\beta \epsilon_i}$ denotes the intra-molecular partition function. We can then write the partition function of the combined system as

$$Q(M, V_0, V, T) = \frac{(V_0 q(T))^M}{\Lambda^{3M} \prod_i N_i^0!} \times \sum_{\{N_i\}=0}^{\infty} \prod_i \frac{V^{N_i} N_i^0!}{V_0^{N_i} (N_i^0 - N_i)! N_i!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.5.18)$$

In Eq. (6.5.18), we have left out a factor $\exp(-\beta \sum_i N_i^0 \epsilon_i)$, as it does not depend on whether the molecules are in the reservoir or the system. We now make use of the fact that, as $M \rightarrow \infty$, $N_i^0 \gg N_i$ and hence

$$Q(M, V_0, V, T) = \frac{(V_0 q(T))^M}{\Lambda^{3M} \prod_i N_i^0!} \times \sum_{\{N_i\}=0}^{\infty} \prod_i \frac{V^{N_i} (N_i^0 / V_0)^{N_i}}{N_i!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.5.19)$$

Moreover, $N_i^0 = M e^{-\beta \epsilon_i} / q(T)$ and hence, using $M/V_0 \equiv f$, and leaving out the constant pre-factor $(V_0 q(T) / \Lambda^3)^M / \prod_i (N_i^0)!$, we get for the Grand-Canonical partition function, Ξ

$$\Xi(f, V, T) = \sum_{\{N_i\}=0}^{\infty} \prod_i \frac{[f V e^{-\beta \epsilon_i} / q(T)]^{N_i}}{N_i!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)], \quad (6.5.20)$$

and hence the probability distribution $\mathcal{N}_{f, V, T; \{N_i\}}(\mathbf{s}^N; L)$ becomes

$$\mathcal{N}_{f, V, T; \{N_i\}}(\mathbf{s}^N; L) \propto \prod_i \frac{[f V e^{-\beta \epsilon_i} / q(T)]^{N_i}}{N_i!} \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.5.21)$$

The detailed-balance condition $K(N \rightarrow N+1) = K(N+1 \rightarrow N)$ now implies:

$$\begin{aligned} \frac{\text{acc}(N_i \rightarrow N_i + 1)}{\text{acc}(N_i + 1 \rightarrow N_i)} &= \frac{f V e^{-\beta \epsilon_i}}{q(T)(N_i + 1)} \times \\ &\exp[-\beta \Delta U(N_i \rightarrow N_i + 1)] \frac{\alpha(N_i \rightarrow N_i + 1)}{\alpha(N_i + 1 \rightarrow N_i)}. \end{aligned} \quad (6.5.22)$$

As we sample the internal states of the molecules in the ideal gas reservoir according to their Boltzmann weight, the probability of attempting a trial move

to increase N_i to $N_i + 1$ is $\alpha(N_i \rightarrow N_i + 1) = \exp(-\beta\epsilon_i)/q(T)$. Conversely, the probability to remove a molecule of type i , randomly selected from the $N + 1$ molecules in the system, is $\alpha(N_i + 1 \rightarrow N_i) = (N_i + 1/N + 1)$.

Inserting these expressions for the attempt probability, we get a very simple expression

$$\frac{\text{acc}(N_i \rightarrow N_i + 1)}{\text{acc}(N_i + 1 \rightarrow N_i)} = \frac{fV}{(N + 1)} \exp[-\beta\Delta U(N_{i \rightarrow N_i+1})]. \quad (6.5.23)$$

Note that the internal energy of the molecule to be inserted/removed does not enter into this expression.

However, if we perform a trial move at constant N , where we attempt to change the internal energy of a randomly selected molecule for state i to state j , then we get:

$$\frac{\text{acc}(i \rightarrow j)}{\text{acc}(j \rightarrow i)} = e^{-\beta(\epsilon_j - \epsilon_i)} \exp[-\beta\Delta U(N_{i \rightarrow j})] \quad (6.5.24)$$

Hence, in that case, the internal energy of the molecule must be included in the acceptance rule.

Comments

In grand-canonical Monte Carlo simulations, the fugacity f or, equivalently, the chemical potential μ , of the molecules is imposed, while the number of particles N is free to fluctuate. During the simulation we may measure other thermodynamic quantities, such as the pressure P , the average density $\langle\rho\rangle$, or the internal energy $\langle\mathcal{U}\rangle$. As we impose the chemical potential, we can derive all other *thermal* properties, such as the Helmholtz free energy or the entropy. This may seem surprising, as we argued in Section 3.2 that Metropolis sampling cannot be used to sample absolute free energies and related quantities. Yet, with grand-canonical Monte Carlo, we seem to be doing precisely that. The answer is that, in fact, we do not. What we measure is not an absolute but a relative free energy. In grand-canonical Monte Carlo, we are equating the chemical potential of the molecules in an interacting fluid to that of the same molecules in a hypothetical ideal gas phase at density $\rho = f$.⁸

The grand-canonical Monte Carlo method fails when the acceptance probability of trial moves for particle addition or removal becomes very small. For atomic fluids, this condition effectively limits the maximum density at which the method can be used to about twice the critical density. Special tricks are needed to extend the **GCMC**-method to somewhat higher densities [188]. Unlike constant- NPT simulations, GCMC can be used for inhomogeneous systems, e.g., systems containing interfaces.

⁸ As we mentioned before, we define f as the *density* of the hypothetical ideal gas.

Example 10 (Equation of state of the Lennard-Jones fluid - II). In Case Studies 1 and 7, we computed the equation of state of a Lennard-Jones fluid, using respectively *NVT* and *NPT* simulations. A third way to determine the equation of state is to perform a *Grand-canonical* simulation, imposing the temperature T and the chemical potential μ at constant V , and sample the resulting density and pressure. An example of such a calculation is shown in Fig. 6.10.

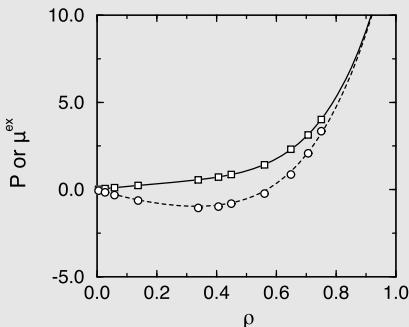


FIGURE 6.10 Equation of state of the Lennard-Jones fluid; isotherm at $T = 2.0$. The solid curve represents the equation of state of Johnson et al. [83]; the squares are the results from grand-canonical simulations (with volume $V = 250.047$). The dotted curve is the excess chemical potential as calculated from the equation of state of ref. [83], and the circles are the simulation results. Note that the excess chemical potential is related to the fugacity f through $\beta\mu^{\text{ex}} = \ln(f/\rho)$.

Grand-canonical simulations are not particularly useful for computing the equation-of-state of a homogeneous fluid, because there will be statistical errors in both the pressure and the density. However, for systems where the pressure itself is ill-defined (e.g., for nano-porous materials), grand-canonical simulations are the method of choice.

The Fortran code to generate this Example can be found in the online-SI, Case Study 9.

Although the grand-canonical Monte Carlo technique can be applied to simple models of non-spherical molecules, the method becomes inefficient for moderately dense fluids of polyatomic molecules, because the acceptance-probability of trial insertions becomes very low. In Section 12.6.1 we discuss techniques that have been designed to deal with this problem.

6.5.4 Semigrand ensemble

The discussion in Section 6.5.3 indicates that the Grand-Canonical MC method can be readily extended to molecular mixtures. If we have a mixture of species $\alpha = 1, 2, \dots$, in contact with a reservoir that fixes the fugacities of the various species at values f_α , then we can write the Grand-Canonical partition function, Ξ , as

$$\Xi(\{f_\alpha\}, V, T) = \sum_{\{N_\alpha\}=0}^{\infty} \prod_\alpha \frac{[f_\alpha V]^{N_\alpha}}{N_\alpha!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (6.5.25)$$

However, for mixtures, it is often advantageous to consider another ensemble, the so-called *Semigrand Ensemble*, where we keep the total number of particles $N = \sum_\alpha N_\alpha$ fixed, but allow the composition of the mixture to fluctuate, i.e., we allow trial moves of the type $\alpha \leftrightarrow \beta$.⁹

The semigrand ensemble finds applications in the study of (fluid-fluid) phase equilibria of mixture, and in the simulation of mixtures of species that interconvert chemically (see Example 11). **Semigrand Monte Carlo (SGMC)** simulations [203] are also useful to simulate polydisperse systems, which occur frequently in soft matter science (see Example 12). As SGMC simulations involve particle swaps, rather than insertions or deletions, they can often be performed at densities where GCMC simulations would fail due to the low acceptance of trial moves. Moreover, SGMC simulations can be carried out at constant pressure, rather than constant volume, which is advantageous when studying phase coexistence.

As noted in Section 8.5.3, the Boltzmann factor associated with the change of identity of a particle in a mixture is related to the difference in excess chemical potential of the two species that take part in the interchange. We can get good statistics on $\Delta\mu_{\text{ex}}$ even when the particle insertion method for measuring the excess chemical potential of the individual species would fail, as is for instance the case in a substitutionally disordered crystalline solid [204]. The normal grand-canonical Monte Carlo (GCMC) method has about the same range of applicability as the particle insertion method. It is logical therefore to infer that it should be possible to construct a simulation scheme based on particle *interchanges* that should work at densities where the GCMC scheme is likely to fail (see Fig. 8.3).

To introduce the SGMC method, we start from Eq. (6.5.25), the expression for the grand-canonical partition function Ξ of an n -component mixture. Note that in Eq. (6.5.25) there is no constraint on the total number of particles $N = \sum_{\alpha=1}^n N_\alpha$.

Next, we consider one term in Eq. (6.5.25) with a fixed value of N . Once we fix N , the N_α are linearly dependent because their sum is fixed. Kofke and Glandt [203] used this dependence to remove one N_α , say N_1 , from the sum in Eq. (6.5.25) [203]. However, as there often is no obvious “preferred” species, we proceed by noting that

$$\sum'_{N_1, \dots, N_n} \prod_{\alpha=1}^n \frac{f_\alpha^{N_\alpha}}{N_\alpha!} = \frac{\left(\sum_{\alpha=1}^n f_\alpha\right)^N}{N!} \equiv \frac{f_{\text{tot}}^N}{N!}, \quad (6.5.26)$$

⁹ In what follows, we deviate from the original derivation of the semigrand canonical ensemble given by Kofke and Glandt [203], to emphasize the close relation between the semigrand ensemble and the Grand Canonical ensemble for molecular systems.

where Σ' denotes a sum over all N_α such that $\sum_{\alpha=1}^n N_\alpha = N$ and f_{tot} is the sum of the fugacities of all species. We can then express Ξ in Eq. (6.5.25) as

$$\Xi = \sum_{N=0}^{\infty} f_{\text{tot}}^N \sum'_{N_1, \dots, N_n} \prod_{\alpha=1}^n \frac{(f_\alpha V/f_{\text{tot}})^{N_\alpha}}{N_\alpha!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N)]. \quad (6.5.27)$$

For what follows we define the *fugacity fraction*, ξ_α , as the ratio $\xi_\alpha = f_\alpha / f_{\text{tot}}$. ξ_α can be viewed as the mole fraction of species α in the ideal gas reservoir with total fugacity f_{tot} .

In the case of **GCMC** of molecular systems, it was convenient to view the system as a mixture of different species, where every internal state i corresponded to a different species. Now, we do exactly the opposite: we will consider all n different species as if they were different manifestations of the same particle, that is: a particle at position \mathbf{r}_i can take on n different identities. We can then change the sum \sum' over particle numbers, by a sum over the n identities that every particle can take on.

This sounds strange, so we shall use an analogy to explain what we mean. Assume that we have a group of 100 people, made up of three groups: eaters, drinkers, and sleepers. In fact, we want to consider all possible combinations of these groups, with the constraint that the total number is fixed. One such combination would be 30 eaters, 30 drinkers, and 40 sleepers. Then, we make a discovery: the same person can be an eater, a drinker, or a sleeper but not simultaneously. Now our sum over all combinations becomes different: we have 100 “persons” who can all take on anyone out of the three possible identities. In that case, we have many more ways in which we can make a group of 30 eaters, 30 drinkers, and 40 sleepers, namely $100!/(30!30!40!)$. If we wish to have the same total number of terms in our sum as before, we have to divide by this factor.

Let us now translate this example back to the sum over particles in Eq. (6.5.27). We replace the sum over numbers of particles of species α with a sum over all possible identities of all particles. But then we must correct for over-counting by dividing by $N!/\prod N_\alpha!$. Then, Eq. (6.5.27) becomes

$$\begin{aligned} \Xi(f_{\text{tot}}, \{\xi_\alpha\}, V, T) &= \sum_{N=0}^{\infty} f_{\text{tot}}^N \sum_{\text{identities}} \frac{\prod_{\alpha=1}^n (V\xi_\alpha)^{N_\alpha}}{N!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N)] \\ &\equiv \sum_{N=0}^{\infty} f_{\text{tot}}^N \mathcal{Y}(N, \{\xi_\alpha\}, V, T), \end{aligned} \quad (6.5.28)$$

where the last equality defines a new partition function \mathcal{Y} that is a function of N , $\{\xi_\alpha\}$, V and T . In Eq. (6.5.28) the “sum over identities” implies a sum where all N particles can take on all n possible identities.

Note that $\mathcal{Y}(N, \{\xi_\alpha\}, V, T)$ is the partition function of a system with constant N , V , T and $\{\xi_\alpha\}$. Using the fact that $k_B T \ln \Xi = PV$ (Eq. (2.3.21)), and using the maximum-term method mentioned in Section 2.3.3, it then follows that

$k_B T \ln \mathcal{Y} = PV - Nk_B T \ln f_{\text{tot}}$. $\mathcal{Y}(N, \{\xi_\alpha\}, V, T)$ can be viewed as the semi-grand canonical partition function at constant N , $\{\xi_\alpha\}$, V and T . It is often more convenient, i.e., more relevant for the comparison with experiment, to consider the semigrand partition function at constant pressure:

$$\begin{aligned}\mathcal{Y}'(N, \{\xi_\alpha\}, P, T) &\equiv \beta P \int_0^\infty dV \exp(-\beta PV) \mathcal{Y}(N, \{\xi_\alpha\}, V, T) \\ &= \beta P \int_0^\infty dV \exp(-\beta PV) \\ &\quad \times \sum_{\text{identities } \alpha=1}^n \prod_{\alpha=1}^n (V \xi_\alpha)^{N_\alpha} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N)].\end{aligned}\quad (6.5.29)$$

The relation with thermodynamics is given by

$$-k_B T \ln \mathcal{Y}'(N, \{\xi_\alpha\}, P, T) = Nk_B T \ln f_{\text{tot}}. \quad (6.5.30)$$

We realize that the above sequence of seemingly random hops from one ensemble to the next may have confused the reader. Let us therefore briefly summarize what we did: we started with a grand-canonical ensemble for a multi-component mixture with independent variables ($f_{\text{tot}}, \{\xi_\alpha\}, V, T$). We then transformed to a semigrand ensemble where we fixed N at the expense of f_{tot} . Finally, we transformed from an ensemble at constant $N, \{\xi_\alpha\}, V, T$ to a ensemble with fixed $N, \{\xi_\alpha\}, P, T$. Monte Carlo simulations in this ensemble allow us to study the properties of a mixture where the relative fugacities of the components have been fixed, but where the total fugacity (f_{tot}) a variable that depends on P , N , and T . We stress that we cannot determine f_{tot} directly from an SGMC simulation: it plays a role similar to the free energy in the canonical ensemble, and must be computed separately.

6.5.4.1 Phase coexistence in the semigrand ensemble

If two phases coexist, they must be at the same temperature, pressure, and fugacities. In an **SGMC** simulation, we *impose* the control parameters P, T and $n - 1$ independent $\{\xi_\alpha\}$. To ensure that two phases (*I* and *II*) with the same values of P, T and $\{\xi_\alpha\}$ are in equilibrium, we must find the set of control parameters for which $f_{\text{tot}}^I = f_{\text{tot}}^{II}$. Typically, we use thermodynamic integration to find this point. Studying the variation of f_{tot} with P , at constant $\{\xi_\alpha\}$ is simplest, but it will work for at most one phase —and possibly not even one. The new thermodynamic integration requires a path in ξ -space from the mixture under study to a pure compound, whilst avoiding phase transitions. We assume that, using the techniques discussed in Chapter 8, we can compute the Gibbs free energy and hence the fugacity of the pure phase of one of the components in the mixture (say 1). In the pure phase, $f_{\text{tot}} = f_1(P, T)$. We should now study the variation of f_{tot} as we change the fugacity ratios from $\xi_1 = 1, \xi_{\alpha \neq 1} = 0$ to the desired $\{\xi_\alpha\}$. To this end, we define a parametrized path in ξ -space where every $\xi_\alpha(\lambda)$ is

a function of the parameter λ , such that $\xi_1 = 1$ for $\lambda = 0$ and the target $\{\xi_\alpha\}$ corresponds to $\lambda = 1$. The path is chosen such that for all values of λ , $\sum \xi_\alpha(\lambda) = 1$. We can then write:

$$\begin{aligned} \frac{d \ln f_{\text{tot}}(\lambda)}{d \lambda} &= \sum_{\alpha=1}^n \left(\frac{\partial \ln f_{\text{tot}}(\lambda)}{\partial \xi_\alpha} \right)_{P,T,\{\xi_{\beta \neq \alpha}\}} \frac{d \xi_\alpha}{d \lambda} \\ &= - \sum_{\alpha=1}^n \left(\frac{\langle N_\alpha \rangle / N}{\xi_\alpha} \right) \frac{d \xi_\alpha}{d \lambda} \\ &= - \sum_{\alpha=1}^n \frac{\langle x_\alpha \rangle}{\xi_\alpha} \frac{d \xi_\alpha}{d \lambda}, \end{aligned} \quad (6.5.31)$$

where $x_\alpha \equiv \langle N_\alpha \rangle / N$ denotes the measured mole fraction of component α at λ, P, T . We then obtain

$$\ln f_{\text{tot}}(\lambda = 1) = \ln f_1(\lambda = 0) + \int_0^1 d\lambda \frac{d \ln f_{\text{tot}}(\lambda)}{d \lambda}. \quad (6.5.32)$$

We should then compute the value of $f_{\text{tot}}(\lambda = 1)$ for both phases in the mixture. In general, for a given set $\{\xi_\alpha\}, P, T$, we will find that $f_{\text{tot}}^I \neq f_{\text{tot}}^{II}$. The final step is then to use

$$\left(\frac{\partial k_B T \ln f_{\text{tot}}}{\partial P} \right)_{T,\{\xi_\alpha\}} = \frac{\langle V \rangle}{N} \quad (6.5.33)$$

to find the pressure where the two phases coexist. Note that we have not specified the nature of the phases I and II. They could be fluid, solid, or liquid crystalline.

Our derivation of the semigrand ensemble is different from the one used in ref. [203], but the results are, of course, the same. In particular, we can easily establish the link with the usual thermodynamic description of mixtures by noting that the simulations directly yield $\langle x_\alpha \rangle$ as a function of ξ_α . Once we have computed $\ln f_{\text{tot}}$, e.g., by thermodynamic integration, we can compute the molar Gibbs free energy of the mixture

$$\frac{G(N, P, T, \{\xi_\alpha\})}{N} = k_B T \sum_{\alpha=1}^n \langle x_\alpha \rangle \ln[f_{\text{tot}} \xi_\alpha], \quad (6.5.34)$$

and from there we can derive, for instance, the phase diagram.

We have not yet specified what trial moves we carry out to change the identity of a particle. There are many possible choices. One of the simplest is to choose α' with a probability $\xi_{\alpha'}$, which is similar to the way we treated intramolecular states in the Grand Canonical ensemble. In that case,

$$\text{acc}(\xi_\alpha \rightarrow \xi_{\alpha'}) = \min \left\{ 1, \exp \left[-\beta \Delta U(s^N) \right] \right\}. \quad (6.5.35)$$

6.5.4.2 Chemical equilibria

Up to this point, we have assumed that we can *impose* the fugacity fractions ξ_α . However, if we have a mixture of compounds that are in chemical equilibrium, then there are relations between their fugacities. *This is where, finally, quantum mechanics becomes important* because the chemical potential of species α in the ideal gas phase is equal to

$$\mu_\alpha^{\text{id gas}} = k_B T \ln \left(\frac{\Lambda_\alpha^3}{q_{\text{int } \alpha}(T)} \right) + k_B T \ln f_\alpha = \mu^\circ + k_B T \ln f_\alpha , \quad (6.5.36)$$

and both Λ and q_{int} depend on Plank's constant. If there is a chemical reaction where v_α molecules of type α , v_β molecules of type β , etc. —can convert into $v_{\alpha'}$ molecules of type α' etc., then equilibrium implies that

$$\sum_\alpha v_\alpha \mu_\alpha = \sum_{\alpha'} v_{\alpha'} \mu_{\alpha'} \quad (6.5.37)$$

and hence

$$K^\circ \equiv e^{-\beta \left[\sum_{\alpha'} v_{\alpha'} \mu_{\alpha'}^\circ - \sum_\alpha v_\alpha \mu_\alpha^\circ \right]} = \frac{\prod_{\alpha'} f_{\alpha'}^{v_{\alpha'}}}{\prod_\alpha f_\alpha^{v_\alpha}} \quad (6.5.38)$$

Eq. (6.5.38) implies that every chemical reaction imposes a relation between the fugacities. When we carry out an SGMC simulation of a mixture of compounds that are in chemical equilibrium, we have fewer independent ξ_α s. For a given value of f_{tot} , we should then first express the dependent fugacities in terms of the independent fugacities: which fugacities are deemed to be independent is usually a matter of practical convenience. Note that in a reaction where the total number of molecules changes, the ξ_α will depend on f_{tot} . In addition, the equilibrium shifts with temperature. For an illustration of the SGMC method applied to a mixture of compounds in chemical equilibrium, see Example 11.

Comment

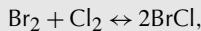
In its simplest form, the semigrand ensemble method can be used only to study chemical equilibria that involve reactions in which the total number of molecules is conserved. For reactions in which the total number of molecules is not conserved, it becomes necessary to include particle insertions/deletions, in addition to particle identity changes. As with GCMC, particle insertion moves become less efficient at high densities (see, however, refs. [205,206]).

Even more ensembles

In between the grand and the semi-grand ensemble, there exist mixed forms. The most common among these is the osmotic ensemble [207], where the particle-number of some species (solute) is kept fixed, while other molecules (solvent) can exchange with a reservoir, much as what happens in experiments where

some species can cross a semi-permeable membrane, and others cannot. In soft matter, the osmotic ensemble is often used implicitly in the study of solutions of fixed numbers of mesoscopic particles (e.g., colloids) under the influence of depletants with a fixed chemical potential [208]. With the method described in the present chapters the Monte Carlo sampling of these other, these hybrid ensembles should hold no surprises for the reader.

Example 11 (Vapor-liquid equilibria of Br₂-Cl₂-BrCl). The vapor-liquid coexistence curve of the ternary system Br₂-Cl₂-BrCl is an example of a phase equilibrium problem in which the components are also in chemical equilibrium. The chemical reaction of interest is



with equilibrium constant

$$K^\circ(T) = \frac{f_{\text{BrCl}}^2}{f_{\text{Br}_2} f_{\text{Cl}_2}}. \quad (6.5.39)$$

This equilibrium constant is approximately 10 (at $T = 273$ K). Since in this chemical reaction the total number of molecules is conserved, we can use the standard semigrand ensemble technique to locate the liquid-vapor coexistence curve.

Let us first consider what the approach would be if we were to perform ordinary N , P , T simulations to determine the vapor-liquid coexistence curve. In that case, we would determine the fugacities of the three components in both phases and then find the points for which the fugacity of each is the same in both phases, subject to the constraint imposed by Eq. (6.5.39). Kofke and Glandt [203] have shown that the use of the semigrand ensemble can simplify this procedure significantly.

In the constant-pressure version of the semigrand ensemble, we have as independent variables: the pressure, the temperature, the total number of particles, and the fugacity ratios of the three components. However, these ratios must add to one:

$$\xi_{\text{BrCl}} + \xi_{\text{Br}_2} + \xi_{\text{Cl}_2} \equiv 1. \quad (6.5.40)$$

Combining Eq. (6.5.39) with Eq. (6.5.40), yields a quadratic equation that allows us to express both ξ_{BrCl} and ξ_{Cl_2} as a function of ξ_{Br_2} . The next step is the calculation of the fugacity of Br₂ along the path defined by Eqs. (6.5.39) and (6.5.40) for both the liquid and the vapor phases.^a The change in the fugacity of Br₂ (henceforth referred to as species 1) along a path in ξ -space described by the functions $\xi_\alpha = \xi_\alpha(\lambda)$, is given by

$$\ln f_1^{(b)} - \ln f_1^{(a)} = \int_{\lambda^{(a)}}^{\lambda^{(b)}} d\lambda \sum_{\alpha=1}^n \left(\frac{\partial \ln f_1}{\partial \xi_\alpha} \right)_{N, P, T, \{\xi_{\alpha'} | \alpha' \neq \alpha\}} \frac{d\xi_\alpha}{d\lambda}. \quad (6.5.41)$$

For our system the integration variable λ is $\lambda = \xi_{Br_2}$. Equation (d) can be used to determine the change in f_{tot} , and from that the fugacity of all compounds, along the path defined by Eqs. (6.5.39) and (6.5.40). In practice, the simulation proceeds as follows. For the liquid phase the following steps are performed:

1. We start the integration of Eq. (6.5.41) from a state point where the chemical potential of the reference compound can be computed relatively easily. The most natural starting point would be to determine the fugacity of pure liquid Br_2 using one of the methods described in Chapter 8.
2. We then integrate Eq. (6.5.41) from $\lambda^{(a)} = \xi_{Br_2} = 1$ to $\lambda^{(b)} = \xi_{Br_2} = 0$. The integrand in Eq. (6.5.41) is an ensemble average that is conveniently measured in a semigrand ensemble simulation. Once ξ_{Br_2} is specified, ξ_{Cl_2} and ξ_{BrCl} follow. During the simulation, a trial move may involve either the attempted displacement of a particle or an attempt to change its chemical identity. Attempted identity changes are accepted with a probability given by Eq. (6.5.35).

In principle, the same scheme could be used to compute the chemical potential of Br_2 in the vapor phase. However, if the vapor phase is dilute, it is often more convenient to compute the lowest few virial coefficients of the mixture. The chemical potential of Br_2 can then be computed analytically from knowledge of these virial coefficients.

Once the dependence of the total fugacity f_{tot} on the fugacity fraction ξ_{Br_2} is known for both phases, we can determine the point where f_{tot} and hence all fugacities are equal in the vapor and the liquid.

When compared to the Gibbs ensemble technique (section 6.6), a drawback of the semigrand ensemble method is that it is necessary to compute the free energy of the system in the pure phases. However, once this information is known, the semigrand scheme can, unlike the Gibbs-ensemble method, be applied to dense phases, such as solids.

^a Here the approach of ref. [203] is slightly different from the one that we described in section 6.5.4.1.

The SGMC scheme can also be used to simulate phase equilibria in continuously-polydisperse systems, including polydisperse solids. We describe an application of the semigrand ensemble for polydisperse systems in Example 12.

Example 12 (Freezing of polydisperse hard spheres). One of the early successes of molecular simulation was the discovery that a fluid of hard-spheres can freeze in the absence of any attraction [18,19]. Certain colloidal solutions are excellent experimental realizations of hard-sphere fluids. However, real colloidal solutions are never perfectly monodisperse. The polydispersity of colloidal suspensions can have a strong effect on the location of the freezing transition [209]. These experiments stimulated interest in a systematic study of the effect of polydispersity on the location of the freezing transition.

At first sight, one might think that it would be natural to use the grand-canonical ensemble to study polydisperse systems. In this ensemble, we can impose the chemical potential distribution that generates a continuous size distribution. However, for the numerical study of freezing, the GCMC approach is less suited, as the probability of a successful insertion/deletion of a particle in the solid, or for that matter in the dense liquid, is very low. To avoid this problem, Bolhuis and Kofke [210] used the semigrand ensemble to study the dependence of the freezing curve of polydisperse hard spheres on polydispersity. To trace out the solid-fluid coexistence curve, they combined the semigrand ensemble (see section 6.5.4) with the Gibbs-Duhem integration technique (see section 8.3.1.3).

In experiments, the polydispersity of a suspension is characterized by the probability density $p(\sigma)$, where σ is the diameter of a hard sphere. In contrast, in a grand-canonical simulation, one would impose $f(\sigma)$, i.e., the fugacity of colloids of diameter σ , as a function of σ .

The actual size distribution $p(\sigma)$ is then measured in the simulation. In a semigrand-ensemble simulation we fix the fugacity fractions $\xi(\sigma)$. At low densities, the measured $p(\sigma)$ is the same as $\xi(\sigma)$, the distribution of fugacity fractions. However, at high densities $p(\sigma)$ will differ from $\xi(\sigma)$. As described above, keep the total number of particles (N) fixed. By varying the pressure P , we can then vary f_{tot} , although we still need a separate calculation to determine its absolute value. In hard-sphere systems, the pressure is proportional to the temperature, hence if we vary P , we can keep T constant, or conversely.

In ref. [210], it is assumed that the fugacity-fractions are normally distributed:

$$\xi(\sigma) = \frac{e^{-(\sigma-\sigma_0)^2/2\nu}}{\sqrt{2\pi\nu}},$$

where σ_0 sets the length-scale in the simulations.

In order to locate coexistence, we again should find points where f_{tot} of the two phases are the same. However, now we have an infinite number of components. Eq. (6.5.31) then becomes

$$\frac{d \ln f_{\text{tot}}(\lambda)}{d\lambda} = - \int d\sigma \langle p(\sigma) \rangle \frac{d \ln \xi(\sigma)}{d\lambda}, \quad (6.5.42)$$

where λ is a parameter, yet to be chosen, that we vary to perform a thermodynamic integration to a phase with known properties. In the case of hard spheres, the properties of the mono-disperse phase are well known from simulation. We therefore choose $\lambda = \nu$, the variance of the fugacity-fraction distribution. Then

$$\frac{d \ln f_{\text{tot}}(\nu)}{d\nu} = - \frac{\langle (\sigma - \sigma_0)^2 \rangle}{2\nu^2} + \frac{1}{2\sqrt{2\pi\nu^3}}. \quad (6.5.43)$$

To trace the coexistence curve, starting from the monodisperse hard sphere limit, we do not need to know f_{tot} itself. We know that in the monodisperse

limit, at coexistence, $f_{\text{tot}}^{\text{solid}} = f_{\text{tot}}^{\text{liquid}}$. Hence, to trace the coexistence curve, we should vary ν , and the pressure P , such that the f_{tot} s for the solid and the liquid remain the same. Combining Eqs. (6.5.33) and (6.5.43), we then get:

$$\frac{dP}{d\nu} = \frac{\Delta \langle (\sigma - \sigma_0)^2 \rangle}{2\beta\nu^2(\Delta V/N)},$$

where $\Delta V/N$ is the difference in molar volume of the solid and the liquid at coexistence, and $\Delta \langle (\sigma - \sigma_0)^2 \rangle$ denotes the difference in the value of $\langle (\sigma - \sigma_0)^2 \rangle$ between the solid and liquid phases.

Note that, whereas ν is the same for the coexisting phases, as they must have the same distribution of fugacity fractions, the actual width of the size distribution is not: typically, the size distribution in the solid is narrower than that in the coexisting liquid. Note also that the average of σ depends on ν and is, in general, different from σ_0 , and it is not the same in the two coexisting phases.

The above description of the method of ref. [210] is over-simplified, and we refer the reader to the original paper for more details.

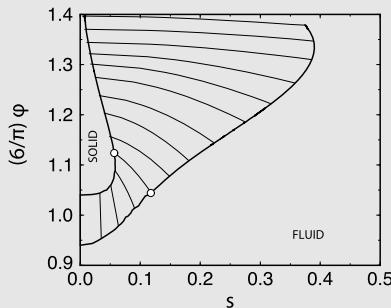


FIGURE 6.11 Coexisting fluid and solid phases in the volume-fraction – polydispersity plane.

In Fig. 6.11 we show the solid-fluid coexistence curve of poly-disperse hard spheres as a function of the *measured* polydispersity s , which is defined through

$$s^2 \equiv \frac{\langle \sigma^2 \rangle}{\langle \sigma \rangle^2} - 1,$$

and is obtained from the simulations.^a

As can be seen from the figure, the fcc phase, which is the stable crystal phase for monodisperse hard spheres, becomes unstable for larger polydispersities: the hard-sphere fcc crystal structure cannot support a polydispersity larger than 5.7% of the average sphere diameter.

^a There is a typo in the definition of s in ref. [210].

TABLE 6.1 Percentage of particles (P_{int}) in the interface of a cubic domain containing N particles. Only the outermost particles are assumed to belong to the interface.

N	125	1,000	64,000	1,000,000
P_{int}	78%	49%	14%	6%

6.6 Phase coexistence without boundaries

In many respects, computer simulations resemble experiments. Yet, in the study of first-order phase transitions, there seems to be a difference. In experiments, a first-order phase transition is easy to locate: at the right density and temperature, we will observe that an initially homogeneous system will separate into two distinct phases, divided by an interface. Measurement of the properties of the coexisting phases is then quite straightforward. In contrast, in a simulation we often locate a first-order phase transition by computing the thermodynamic properties of the individual phases, and then finding the point where the temperature, pressure, and chemical potential(s) of the two bulk phases are equal.

The reason why we are often forced to follow this more indirect route in a simulation is related to finite-size effects. If two phases coexist in small, or even moderately large systems, a relatively large fraction of all particles resides in or near the interface dividing the phases. To estimate this effect, consider the idealized case that we have a cubic domain of one phase, surrounded by the other. We assume that the outermost particles in the cube belong to the interface and that the rest is bulk-like. The fraction of particles in the interface depends on the system size. As can be seen in Table 6.1, systems with fewer than 1000 particles are interface dominated. And, even for quite large systems, the fraction of particles in the interface is non-negligible. As explained in Section 8.3.1.1, direct coexistence simulations, therefore, require rather large systems and long runs.

In the mid-1980s, Panagiotopoulos [211] devised a computational scheme for studying (fluid-fluid) first-order phase transitions that had many of the advantages of a direct simulation of coexistence yet few of its disadvantages. Where applicable, this scheme, usually referred to as the **Gibbs Ensemble** (GE), results in a significant reduction of the computer time required for phase equilibrium calculations, in particular, because a single simulation is enough to locate a point on the phase coexistence curve.

However, with the increase in computing power, and the development of expanded-ensemble simulation methods (Section 13.1.2), there are now powerful alternatives to the Gibbs-ensemble method. Moreover, the Grand-Canonical finite-size scaling approach of refs. [176,212] provides a more reliable estimate of the critical point than the Gibbs-ensemble method.

For this reason, the Gibbs-ensemble method, although still widely used, is now just one technique among many. Having said that, the Gibbs-ensemble

method is easy to use, and provides insight into the factors that play a role in fluid-fluid phase equilibria.

6.6.1 The Gibbs-ensemble technique

The condition for coexistence of two or more phases I, II, \dots is that the pressure of all coexisting phases must be equal ($P_I = P_{II} = \dots = P$), as must be the temperature ($T_I = T_{II} = \dots = T$) and the chemical potentials of all species ($\mu_I^\alpha = \mu_{II}^\alpha = \dots = \mu^\alpha$). Hence, one might be inclined to think that the best ensemble for studying would be the “constant- μPT ensemble”. The quotation marks around the name of this “ensemble” are intentional because, strictly speaking, no such ensemble exists. The reason is simple: if we specify only intensive parameters, such as P , T , and μ , the extensive variables (such as V) are unbounded. Another way to say the same thing is that the set P , T , μ is linearly dependent. To get a decent ensemble, we must fix at least one extensive variable. In the case of constant-pressure Monte Carlo simulations, this variable is the number of particles N , while in grand-canonical Monte Carlo the volume V of the system is fixed.

After this introduction, it may come as a surprise that the Gibbs ensemble method of Panagiotopoulos [211,213] comes very close to achieving the impossible: simulating phase equilibria under conditions where the pressure, temperature, and chemical potential(s) of the coexisting phases are equal. The reason why this method can work is that the method considers a composite system: in equilibrium, the μ , P , and T of the sub-systems are equal, but the *total* system is a perfectly well-behaved constant N , V , T system. Below, we derive the Gibbs-ensemble method, borrowing from the approaches developed in the context of constant- N , P , T and constant- μ , V , T simulations.

We focus on the version of the Gibbs ensemble where the total number of particles and the total volume of the two boxes remain constant; that is, the total system is at N , V , T conditions.

In Appendix L.4 we demonstrate that, in the thermodynamic limit, the (constant- V) Gibbs ensemble is rigorously equivalent to the canonical ensemble.

The description of the N , P , T version can be found in [213]. This constant- P method can only be applied to systems containing two or more components because in a one-component system, the two-phase region is a *line* in the P - T plane. Hence, in a one-component system, the probability that any specific choice of P and T will actually be *at* the phase transition is vanishingly small. In contrast, for two-component systems, the two-phase region corresponds to a finite *area* in the P - T plane.

Note that in either formulation of the Gibbs method, the total number of particles is fixed. The method can be extended to study inhomogeneous systems [214] and is well suited for studying phase equilibria in multicomponent mixtures [213]. A review of applications of the Gibbs-ensemble technique is

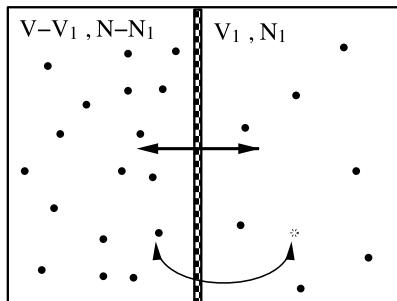


FIGURE 6.12 Schematic sketch of the “Gibbs ensemble” in which two systems can exchange both volume and particles in such a way that total volume V and the total number of particles N are fixed.

given in [215]. The advantage of the Gibbs method over earlier techniques for studying phase coexistence is that, in the Gibbs-ensemble method, the system spontaneously “finds” the densities and compositions of the coexisting phases. Hence, there is no need to compute the relevant chemical potentials as a function of pressure at a number of different compositions and then *construct* the coexistence line.

As in the case of grand-canonical MC simulations, the **Gibbs Ensemble (GE)** method relies on a reasonable number of successful particle insertions to achieve equality of the chemical potential in the coexisting phases. At high densities, where particle insertions have a low acceptance rate, the GE method is not efficient.

6.6.2 The partition function

We start our discussion with the expression for the partition function for a system of N particles distributed over two volumes V_1 and $V_2 = V - V_1$, subject to the constraint that $V = V_1 + V_2$ remains constant (see Fig. 6.12). As V_1 is variable, the total partition function is obtained by integrating not just over the particle coordinates, but also over V_1 [215–217],

$$\begin{aligned} Q_G(N, V, T) \equiv & \sum_{n_1=0}^N \frac{1}{V \Lambda^{3N} n_1! (N - n_1)!} \int_0^V dV_1 V_1^{n_1} (V - V_1)^{N - n_1} \\ & \times \int d\mathbf{s}_1^{n_1} \exp[-\beta \mathcal{U}(\mathbf{s}_1^{n_1})] \int d\mathbf{s}_2^{N-n_1} \exp[-\beta \mathcal{U}(\mathbf{s}_2^{N-n_1})]. \end{aligned} \quad (6.6.1)$$

It then follows that the probability of finding a configuration with n_1 particles in box 1 with a volume V_1 and positions $\mathbf{s}_1^{n_1}$ and $\mathbf{s}_2^{N-n_1}$ is given by

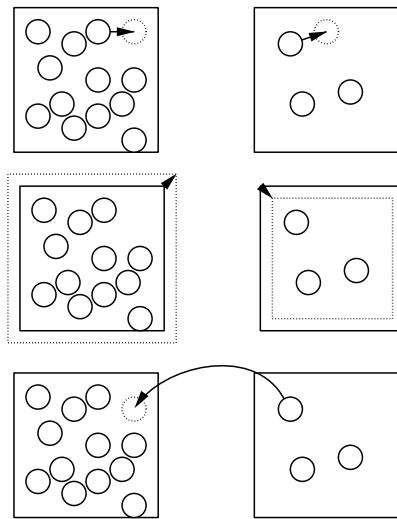


FIGURE 6.13 Monte Carlo steps in the Gibbs-ensemble method: particle displacement, volume change, and exchange of particles.

$$\mathcal{N}(n_1, V_1, \mathbf{s}_1^{n_1}, \mathbf{s}_2^{N-n_1}) \propto \frac{V_1^{n_1} (V - V_1)^{N-n_1}}{n_1! (N - n_1)!} \exp \left\{ -\beta [\mathcal{U}(\mathbf{s}_1^{n_1}) + \mathcal{U}(\mathbf{s}_2^{N-n_1})] \right\}. \quad (6.6.2)$$

We shall use Eq. (6.6.2) to derive the acceptance rules for trial moves in Gibbs-ensemble simulations.

6.6.3 Monte Carlo simulations

Eq. (6.6.2) suggests the following Monte Carlo scheme for sampling all possible configurations of two systems that can exchange particles and volume. In this scheme, we consider the following trial moves (see Fig. 6.13):

1. Displacement of a randomly selected particle.
2. Change of the volume in such a way that the total volume remains constant.
3. Transfer of a randomly selected particle from one box to the other.

The acceptance rules for these steps in the Gibbs ensemble can be derived from the condition of detailed balance

$$K(o \rightarrow n) = K(n \rightarrow o), \quad (6.6.3)$$

where $K(o \rightarrow n)$ is the flow of configuration o to n , which is equal to the product of the probability of being in configuration o , the probability of generating configuration n , and the probability of accepting this move:

$$K(o \rightarrow n) = \mathcal{N}(o) \times \alpha(o \rightarrow n) \times \text{acc}(o \rightarrow n). \quad (6.6.4)$$

Particle displacement

We assume that state n is obtained from state o via the displacement of a randomly selected particle in box 1. The ratio of the statistical weights of these two configurations is given by

$$\frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \frac{\exp[-\beta\mathcal{U}(\mathbf{s}_n^{n_1})]}{\exp[-\beta\mathcal{U}(\mathbf{s}_o^{n_1})]}. \quad (6.6.5)$$

Substitution of this ratio into the condition of detailed balance (6.1.1) gives, as an acceptance rule,

$$\text{acc}(o \rightarrow n) = \min \left(1, \exp\{-\beta[\mathcal{U}(\mathbf{s}_n^{n_1}) - \mathcal{U}(\mathbf{s}_o^{n_1})]\} \right). \quad (6.6.6)$$

This acceptance rule is identical to that used in a conventional N, V, T ensemble simulation (see Section 6.2).

Volume change

For a change of the volume of box 1 by an amount ΔV , $V_1^n = V_1^o + \Delta V$, the ratio of the statistical weights of the configurations after and before the move is given by

$$\frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \frac{(V_1^n)^{n_1} (V - V_1^n)^{N-n_1}}{(V_1^o)^{n_1} (V - V_1^o)^{N-n_1}} \frac{\exp[-\beta\mathcal{U}(\mathbf{s}_n^N)]}{\exp[-\beta\mathcal{U}(\mathbf{s}_o^N)]}. \quad (6.6.7)$$

Imposing the condition of detailed balance gives, as an acceptance rule for this volume change,

$$\text{acc}(o \rightarrow n) = \min \left\{ 1, \frac{(V_1^n)^{n_1} (V - V_1^n)^{N-n_1}}{(V_1^o)^{n_1} (V - V_1^o)^{N-n_1}} \exp\{-\beta[\mathcal{U}(\mathbf{s}_n^N) - \mathcal{U}(\mathbf{s}_o^N)]\} \right\}. \quad (6.6.8)$$

This way of changing the volume was originally proposed by Panagiotopoulos et al. [211,213]. A more natural choice for generating a new configuration in the volume change step is to make a random walk in $\ln[V_1/(V - V_1)]$ instead of in V_1 (see also Chapter 6 for the N, P, T ensemble). This has the advantage that the domain of this random walk coincides with all possible values of V_1 . Furthermore, the average step size turns out to be less sensitive to the density. To adopt this method to the Gibbs ensemble, the acceptance rule for the volume has to be modified.

If we perform a random walk in $\ln[V_1/(V - V_1)]$, it is natural to rewrite Eq. (6.6.1) as

$$\begin{aligned} Q_{N,V,T} &= \frac{1}{\Lambda^{3N} N!} \sum_{n_1=0}^N \binom{N}{n_1} \\ &\times \int_{-\infty}^{\infty} d \ln \left(\frac{V_1}{V - V_1} \right) \frac{V_1(V - V_1)}{V} V_1^{n_1} (V - V_1)^{N-n_1} \end{aligned}$$

$$\times \int d\mathbf{s}_1^{n_1} \exp[-\beta \mathcal{U}(\mathbf{s}_1^{n_1})] \int d\mathbf{s}_2^{N-n_1} \exp[-\beta \mathcal{U}(\mathbf{s}_2^{N-n_1})].$$

The statistical weight of a configuration n with volume V_1 is proportional to

$$\mathcal{N}(n) \propto \frac{(V_1^n)^{n_1+1} (V - V_1^n)^{N-n_1+1}}{V n_1! (N - n_1)!} \exp[-\beta \mathcal{U}(\mathbf{s}_n^N)]. \quad (6.6.9)$$

Imposing detailed balance for this move leads to the acceptance rule

$$\text{acc}(o \rightarrow n) = \min \left\{ 1, \left(\frac{V_1^n}{V_1^o} \right)^{n_1+1} \left(\frac{V - V_1^n}{V - V_1^o} \right)^{N-n_1+1} \times \exp \left\{ -\beta [\mathcal{U}(\mathbf{s}_n^N) - \mathcal{U}(\mathbf{s}_o^N)] \right\} \right\}. \quad (6.6.10)$$

Note that this modification does not affect the acceptance rules for the particle displacement or particle exchange.

Particle exchange

Let us assume that we generate configuration n from configuration o (n_1 particles in box 1) by removing a particle from box 1 and inserting this particle in box 2. The ratio of statistical weights of the configurations is given by

$$\begin{aligned} & \frac{\mathcal{N}(n)}{\mathcal{N}(o)} \\ &= \frac{n_1!(N - n_1)! V_1^{n_1-1} (V - V_1)^{N-(n_1-1)}}{(n_1 - 1)!(N - (n_1 - 1))! V_1^{n_1} (V - V_1)^{N-n_1}} \exp\{-\beta [\mathcal{U}(\mathbf{s}_n^N) - \mathcal{U}(\mathbf{s}_o^N)]\}. \end{aligned}$$

Imposing detailed balance for this move leads to the following acceptance rule:

$$\text{acc}(o \rightarrow n) = \min \left\{ 1, \frac{n_1(V - V_1)}{(N - n_1 + 1)V_1} \exp\{-\beta [\mathcal{U}(\mathbf{s}_n^N) - \mathcal{U}(\mathbf{s}_o^N)]\} \right\}. \quad (6.6.11)$$

Implementation

A convenient method for generating trial configurations is to perform a simulation in cycles. One cycle consists of (on average) N_{part} attempts to displace a (randomly selected) particle in one of the (randomly chosen) boxes, N_{vol} attempts to change the volume of the subsystems, and N_{swap} attempts to exchange particles between the boxes. It is important to ensure that at each step of the simulation the condition of microscopic reversibility is fulfilled. Possible Gibbs-ensemble algorithms are shown in Algorithm 15, together with the ones in the SI (Algorithms 42 and 43).

The implementation of trial moves for particle displacement and volume change in Gibbs-ensemble simulations is very similar to that of the corresponding trial moves in a normal N, V, T or N, P, T simulation. However, the

Algorithm 15 (Basic Gibbs-ensemble simulation)

program mc_Gibbs	Gibbs-ensemble simulation
for $1 \leq \text{icycl} \leq \text{ncycle}$ do	perform ncycl MC cycles
$\text{ran} = \mathcal{R}^*(\text{npart} + \text{nvol} + \text{nswap})$	
if $\text{ran} \leq \text{npart}$ then	
mcmove	attempt to displace a particle
else if $\text{ran} \leq (\text{npart} + \text{nvol})$ then	
mcvol	attempt to change the volume
else	
mcswap	attempt to swap a particle
endif	
sample	sample observables
enddo	
$[\dots]$	Compute averages of observables
end program	

Specific Comments (for general comments, see p. 7)

1. This algorithm ensures that, in each Monte Carlo step, detailed balance is obeyed. On average, we perform per cycle npart attempts to displace particles, nvol attempts to change the volume, and nswap attempts to swap particles between the two boxes.
2. Function **mcmove** attempts to displace a randomly selected particle; this algorithm is very similar to Algorithm 2 (but remember that particles are in two different boxes). Function **mcvol** attempts to change the volume of the two boxes (see Algorithm 42 in the SI), function **mcswap** attempts to swap a particle between the two boxes (see Algorithm 43 in the SI), and function **sample** samples observables.

attempts to exchange particles require some care. To ensure that detailed balance is obeyed, it is important to first select at random from which box a particle will be removed and subsequently select a particle at random in this box. An alternative would be to first select one particle at random (from all N particles) and then try to move this particle to the other simulation box. However, in that case, acceptance rule (6.6.11) has to be replaced by a slightly different one [218].

The number of attempts to exchange particles will depend on the conditions of the system. For example, it can be expected that close to the critical temperature, the percentage of accepted exchanges will be higher than close to the triple point. As a possible check whether the number of attempts is sufficient, calculate the chemical potential. Since the calculated energy of a particle that is to be inserted corresponds to just the test particle energy, the chemical potential can be calculated without additional costs.

Inspection of the partition function (6.6.1) shows that one must allow for $n_1 = 0$ (box 1 empty) and $n_1 = N$ (box 2 empty) to calculate ensemble averages correctly. It is important therefore to ensure that the program can handle the case that one of the boxes is empty. As is clear from Eq. (6.6.11), the acceptance rule is constructed such that it indeed rejects trial moves that would attempt to remove particles from a box already empty.

One can also calculate the chemical potential during the exchange step. However, to calculate the chemical potential correctly (see Appendix L.4) one should continue to add *test particles* when one of the boxes is full.

Example 13 (Phase equilibria of the Lennard-Jones fluid). One of the early examples of the Gibbs ensemble is the determination of the vapor-liquid co-existence curve of the Lennard-Jones fluid [78,211].

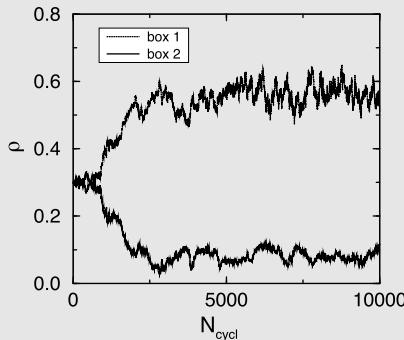


FIGURE 6.14 Particle number density in the two boxes of the Gibbs ensemble as a function of the number of Monte Carlo cycles for a system of Lennard-Jones particles; the number of particles was $N = 256$ and temperature $T = 1.2$.

In Fig. 6.14, the density of the fluid in the two boxes is plotted as a function of the number of Monte Carlo cycles (as defined in Algorithm 15). The simulation was started with equal density in both boxes. During the first 1000 Monte Carlo cycles, the system has not yet “decided” which box would evolve to a liquid density and which to the vapor. After 5000 Monte Carlo cycles, the system seems to have reached equilibrium, and the coexistence properties can be determined.

In Fig. 6.15, the phase diagram of the Lennard-Jones as obtained from Gibbs-ensemble simulations is compared with the phase diagram obtained from the equation of state of Johnson et al. [83]. We point out that comparison of the GE phase diagram with the literature data on the critical point (e.g., [212]) is complicated by the fact that different authors use different truncations of the Lennard-Jones potential (see e.g., [80,219]). Such truncations have a large effect on the location of the critical temperature.

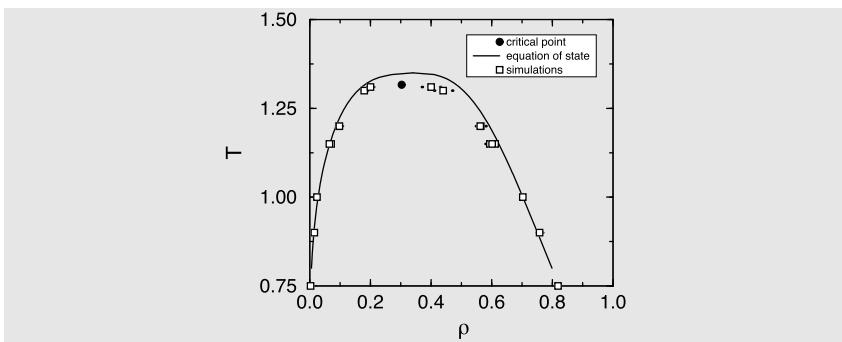


FIGURE 6.15 Phase diagram of the Lennard-Jones fluid, using tail correction beyond the cut-off of 2.5σ to mimic the full Lennard-Jones potential, as calculated with the Gibbs-ensemble technique (squares) and equation of state of Johnson et al. (solid curves). The solid circle indicates the estimated critical point.

The Fortran code to generate this Example can be found in the online-SI, Case Study 10.

Analyzing the results

Assuming that we have a working algorithm to perform a simulation in the Gibbs ensemble, we must now address the question whether the numbers generated in a simulation are reliable. First of all, the equilibrium conditions should be fulfilled:

- The pressure in both subsystems must be equal.
- The chemical potential must be equal in both phases.

There are interesting, additional methods to analyze the data and judge whether a simulation has been successful.

Estimating the critical point

Close to the critical point, the free energy associated with the formation of the liquid-vapor interface becomes very small. As a consequence, the penalty on the creation of an interface in either box becomes small, while the formation of such interfaces is entropically favorable. For this reason, just below the critical point, vapor-liquid coexistence can no longer be observed in a Gibbs-ensemble simulation [220]. Therefore, the highest temperature at which the coexistence can be observed is not a proper estimate of the critical temperature of the system. To estimate the critical temperature, the results can be fitted to the law of rectilinear diameter [221]:

$$\frac{\rho_l + \rho_g}{2} = \rho_c + A(T - T_c), \quad (6.6.12)$$

where $\rho_l(\rho_g)$ is the density of the liquid (gas) phase, ρ_c the critical density, and T_c the critical temperature. Furthermore, the temperature dependence of the density difference of the coexisting phases is fitted to a scaling law [143]

$$\rho_l - \rho_g = B(T - T_c)^\beta, \quad (6.6.13)$$

where β is the critical exponent¹⁰ (for three-dimensional systems $\beta \approx 0.32$ and for two-dimensional systems $\beta = 0.125$ [143]). A and B depend on the system and are obtained from the fit.

These equations should be used with care. Strictly speaking, they cannot be used for a simulation of a finite system. The reason is that, at the critical point, the correlation length that measures the spatial extent of spontaneous density fluctuations diverges. In a finite system, however, these fluctuations are constrained by the size of the simulation box. If we suppress long-range fluctuations, we in fact are modeling a classical system, which has mean-field critical exponents. We therefore can expect to observe a crossover temperature; below this temperature, we sample all relevant fluctuations and we expect to observe nonclassical behavior. Above this temperature, we expect classical behavior. The crossover temperature will depend on the kind of ensemble used in the simulation.

For the Lennard-Jones fluid in three and two dimensions the finite-size effects for the Gibbs ensemble have been analyzed by Panagiotopoulos [222] (see Example 30). The results of this study indicate that for off-lattice systems this crossover temperature is very close to the critical temperature. This suggests that, for applications in which we want to obtain an estimate of the critical temperature, it is safe to use Eqs. (6.6.12) and (6.6.13). In cases where finite-size effects are nevertheless thought to be significant, it is always possible to perform some simulations using different system sizes; although it seems natural to perform such additional simulations on larger systems, an estimate of the importance of finite-size effects can usually be obtained with much less effort from simulations on smaller systems. Of course, if one is interested specifically in finite-size effects or in the accurate determination of critical exponents then one has to be more careful and a proper finite-size scaling analysis should be performed (see, for example, the work of Rovere et al. [175,177,223] and Wilding and Bruce [176]). For such calculations, the Gibbs-ensemble technique is not particularly suited.

Why the Gibbs ensemble fails for solids

The acceptance rate of particle-exchange moves will get very low if one of the coexisting phases is a crystalline solid: in such cases, successful insertion relies on the possibility of finding a vacancy in the recipient crystal. As the equilibrium

¹⁰ Strictly speaking, the use of a scaling law with nonclassical critical exponents is not consistent with the use of the law of rectilinear diameters. However, within the accuracy of the simulations, deviations from the law of rectilinear diameters will be difficult to observe.

concentration of such defects is usually very small, the Gibbs-ensemble method does not work for solids.

However, there is a deeper reason why the GE approach fails for crystalline solids: once we accept that the number of particles in a crystal is not fixed by the number of lattice sites, we should allow for the fact that, even at constant volume and constant number of particles, the number of lattice sites (N_c) may vary. In the case of a one-component system, the expression for a variation in the Helmholtz free energy is then [224]

$$dF = -SdT - VdP + \mu dN + \mu_c dN_c , \quad (6.6.14)$$

where μ_c is the “chemical potential” associated with the number of lattice sites. In equilibrium, μ_c must be zero, because the equilibrium number of lattice sites minimizes the free energy. Therefore, Eq. (6.6.14) is not used to describe the thermodynamics of macroscopic crystalline materials.

However, in a simulation, changing the number of lattice sites independently of the number of particles, is not easy, although it has been observed [225]. Standard Gibbs-ensemble simulations can change the volume of a system and the number of particles, but there is no trial move to change the number of lattice sites. As a consequence, if the GE method were applied to a soft solid with many vacancies (or interstitials), the system would reach a state where the pressure and the chemical potential of the two phases would be equal, but μ_c would be non-zero and different. Hence, such simulations would not sample two systems in equilibrium. Similar problems could occur in the simulation of some liquid-crystalline phases, or phases characterized by periodic defect structures.

6.6.4 Applications

The Gibbs-ensemble technique has been used to study the phase behavior of a variety of systems. The results of these simulations are reviewed in [215,226, 227]. Here we discuss a few applications of the Gibbs ensemble for which the algorithm differs significantly from the one described in Section 6.6.3.

Illustration 5 (Polar and ionic fluids). Because of the long range of the dipolar and Coulombic interactions, the dipolar and Coulombic potential cannot simply be truncated. Special techniques, such as the Ewald summation or reaction field (see Chapter 11), have been developed to take into account the long-range nature of the potential in a simulation. A simple truncation of long-range intermolecular interactions at half the box diameter can lead to an incorrect estimate of the phase coexistence curve. In addition to the truncation of dipolar or Coulomb interactions never being admissible, there is a problem if the potential is truncated at half the diameter of the periodic box. As the size of the simulation box fluctuates during the simulation, the effective potential is also changing. As a consequence, a particle in a large

simulation box feels a very different interaction potential than a particle in a small box. The result is that a Gibbs-ensemble simulation with simulation boxes of different sizes may find apparent phase coexistence between two systems that are described by *different* potentials. In fact, this problem is not limited to Coulomb or dipolar interactions. Even for the relatively short-range Lennard-Jones potential, the phase diagram is very sensitive to the details of the truncation of the potential (see section 3.3.2.2). An example of an application of the Gibbs-ensemble technique to a fluid with Coulomb interactions is the study by Panagiotopoulos [228] of a simple model for an ionic solution, namely, the restricted primitive model.^a The estimate of the location of the critical point that follows from this simulation differed appreciably from an earlier estimate [229,230] obtained by truncating the Coulomb potential at half the box diameter.

When the Ewald summation method is used to account for Coulomb or dipolar interactions, then the system size dependence of the results of a Gibbs-ensemble simulation is usually quite small. Such weak system size dependence, for instance, was found in a Gibbs-ensemble simulation of the liquid-vapor transition in a Stockmayer fluid^b [231,232].

In the case of a closely related system, the dipolar hard-sphere fluid, the Gibbs-ensemble simulations have even resulted in new insights into an old problem, namely, the location of the liquid-vapor critical point. At first sight, nothing seems to be special about the liquid-vapor transition in the dipolar hard-sphere fluid. As the orientationally averaged interaction between two dipoles results in a van der Waals-like $1/r^6$ attraction, de Gennes and Pincus conjectured that the vapor-liquid coexistence would be similar to that of a conventional van der Waals fluid [233]. Kalikmanov [234] used this conjecture to estimate the critical point. More sophisticated liquid-state theories [235] gave qualitatively similar (though quantitatively different) results. And, indeed, early constant N , V , T Monte Carlo simulations of Ng et al. [236] found evidence for such liquid-vapor coexistence, supporting the theoretical predictions of the existence of liquid-vapor coexistence in a dipolar hard-sphere fluid.

However, simulations found no evidence for a liquid-vapor transition [237,238]. To be more precise, these simulations found no evidence for liquid-vapor coexistence in the temperature range predicted by the different theories. Even at the lowest temperatures that could be studied, liquid-vapor coexistence was not observed. Rather, it was found that at low temperatures the dipoles align nose to tail and form chains [238,239] or even closed rings [240]. These rings preempt a fluid-fluid phase transition.

^a The restricted primitive model is a hard-core potential with a point charge.

^b The Stockmayer potential is a Lennard-Jones potential plus a point dipole.

Illustration 6 (Mixtures). An important application of the Gibbs-ensemble technique is the simulation of the phase behavior of mixtures (see, e.g., [213, 241,242]). One of the main problems in studying liquid-liquid phase coex-

istence is that both phases are usually quite dense. It is difficult therefore to exchange particles between the two phases. This problem is more serious for the larger of the two species. Fortunately, it is not necessary to carry out such exchanges for all species, to impose equality of the chemical potentials in the coexisting phases. It is sufficient that the chemical potential of only one of the components, label i , is equal in both phases. For the other components, j , we impose that $\mu_j - \mu_i$ should be equal in the two phases. Of course, this implies that, when μ_i is the same in both phases, then so are all μ_j . However, the condition that $\mu_j - \mu_i$ is fixed is much easier to impose in a simulation. In practice, this is achieved by performing Monte Carlo trial moves in which change is attempted on the identity of a particle (e.g., from i to j). The imposed chemical potential difference determines the acceptance probability of such trial moves. This approach was first applied to Gibbs-ensemble simulation of mixtures by Panagiotopoulos [243]. In these simulations, only the smaller particles are swapped between the two simulation boxes, while for the larger particles only identity change moves are attempted.

The situation becomes even simpler when studying symmetric mixtures. In such systems, the densities of the coexisting phases are equal, while the molar compositions in boxes I and II are symmetry related ($x_I = 1 - x_{II}$). As a result, in Gibbs-ensemble simulations of such symmetric systems, it is not necessary to perform volume changes [244,245] or particle exchanges between the boxes [246].

It is often advantageous to combine the Gibbs-ensemble method for mixtures with the semigrand-ensemble approach discussed in Chapter 6. Using semigrand canonical sampling, we can impose the same fugacity ratios in the two simulation boxes.

In other words, we allow particles in either box to change identity while remaining in the same box. But in addition we allow trial moves where we attempt to move a particle of one (or possibly more) reference species from one box to the other. Below, we assume for simplicity, that we only attempt to move particles of one species (say species 1) from one box to the other. The selection of the particle to be swapped goes as follows: First select box I or box II with equal probability. Next, select any molecule of type 1 in the selected box and try to insert it in the other box. The acceptance probability of such a move is given by Eq. (6.6.8).^a The natural choice for the reference species 1 is clearly the one that can be swapped most efficiently, that is, the smallest molecule in the system.

^a We point out that the implementation that we suggest here is slightly different from the one advocated in [203] and closer to the approach of Stapleton et al. [247].

6.7 Questions and exercises

Question 18 (Trial moves).

1. In grand-canonical MC simulations, one could increase the fraction of trial moves that attempt to insert/remove a particle.

If one uses a very large fraction of particle-swap trial moves in the grand-canonical ensemble, one can increase the number of accepted insertions and deletions. However, is this an efficient approach? (Hint: assume that during these swap moves one has at least a successful deletion of a particle from the system, before the positions of the other particles are changed.)

2. Which trial move will be computationally most expensive: particle displacement, change of the volume, particle insertion, or deletion? Motivate your answer.
3. In a simulation of a molecule that consists of more than one interaction site, a trial move that rotates the molecule around its center of mass is usually included. Why? What is the acceptance/rejection rule for this trial move?
4. When a particle is added in the grand-canonical ensemble, the tail corrections to the potential energy may result in an energy change. Derive an expression for this energy change when a Lennard-Jones potential with cut-off radius r_c is used.

Question 19 (Multicomponent simulation). Consider a grand-canonical Monte Carlo simulation of a two-component mixture at temperature T . The fugacities of the two components are given by f_1 and f_2 respectively (see Eq. (6.5.8)).

1. To add or remove particles the following scheme is used:
 - Select with equal probability to attempt the insertion or removal of a particle.
 - Select with equal probability either component.
 - Attempt to add or remove a particle of this component.

Derive the acceptance rules for these trial moves.

2. An alternative scheme would be:
 - Select at random to add or remove a particle.
 - Select at random a particle, independent of its identity.

Is this scheme obeying detailed balance if the previous acceptance rules are used? If not can this be corrected? Hint: you might want to consult ref. [218].

Question 20 (Gibbs ensemble). When one of the boxes in the Gibbs ensemble is infinitely large and the molecules in this box do not have intermolecular interactions, the acceptance/rejection rule for particle swap becomes identical to the acceptance/rejection rule for particle swap in the grand-canonical ensemble. Derive this result.

Question 21 (Scaling of the potential). When an attempt is made to change the volume of a system in an NPT simulation, the energy of the new configuration can be calculated efficiently if we can exploit scaling properties of the potential. Consider a system of Lennard-Jones particles. The total potential energy U of

the system is equal to

$$U = \sum_{i < j} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right]. \quad (6.7.1)$$

Suppose that the box size of the system is changed from L to L' and $S \equiv L'/L$.

1. Derive expressions for the new potential energy U' and the new virial V' as a function of S , the old potential energy (U) and the old virial (V).
2. Why are volume-changing trial moves cheap in this case.
3. In practice, we do not compute LJ interactions explicitly beyond a cut-off radius r_c . Explain why the cutoff-radius must be scaled when using the scaling method.
4. How do the tail corrections scale with S ?

Exercise 12 (Monte Carlo in the NPT ensemble). On the book's website, you can find a program to simulate hard spheres (diameter 1) in the NPT ensemble using the Monte Carlo technique.

1. What problems would arise if you tried to calculate the virial for this system directly?
2. In the current code, a random walk is performed in $\ln(V)$ instead of V . Change the code in such a way that a random walk in V is performed. Check that the average densities calculated by both algorithms are equal.
3. Make a plot of the acceptance probability for volume displacements as a function of the maximum volume displacement for both algorithms.

Exercise 13 (Ising model). In this exercise, we consider a 2d Ising model. In this model, N spins s (± 1) are arranged on a square lattice. Every spin (i) has 4 nearest neighbors ($j = 1, 2, 3, 4$). The total energy of the system equals

$$U = -\frac{\epsilon}{2} \sum_{i=1}^{i=N} \sum_{j \in \text{nn}_i} s_i s_j, \quad (6.7.2)$$

in which $s_i = \pm 1$ and $\epsilon > 0$. The second summation is a summation over all nearest neighbors of spin i (nn_i). The total magnetization M equals the sum over all spins:

$$M = \sum_{i=1}^{i=N} s_i. \quad (6.7.3)$$

The 2d Ising model has a critical point close to $\beta_c \approx 0.44$.

1. Complete the given simulation code for this system that is provided on the book's website.
2. Calculate the distribution of M for $N = 32 \times 32$ and $\beta = 0.5$ in the canonical ensemble. This distribution should be symmetrical around $M = 0$:

$$p(M) = p(-M). \quad (6.7.4)$$

The simulation does not appear to reproduce such a symmetric distribution. What is the problem?

3. Instead of a simulation in the canonical ensemble, one can perform a biased simulation using a distribution function

$$\pi \propto \exp[-\beta U + W(M)]. \quad (6.7.5)$$

The average value of an observable O in the canonical ensemble is related to the “ π -average”, through

$$\langle O \rangle = \frac{\langle O \exp[-W(M)] \rangle_\pi}{\langle \exp[-W(M)] \rangle_\pi}, \quad (6.7.6)$$

in which $\langle \dots \rangle_\pi$ denotes an ensemble average in the biased system. Derive this relation.

4. Perform simulations with some given distributions $W(M)$ (see the appropriate files on the book’s website). Explain your results. How should one choose the function $W(M)$ to obtain the optimal efficiency?
5. What happens when $W(M)$ is a Gaussian,

$$W(M) = A \exp\left[-\frac{M^2}{2\sigma^2}\right] \quad (6.7.7)$$

with $A > 0$?

6. What biased distribution is generated if $W(M) = W(U) = \beta U$?

Exercise 14 (Vapor-liquid equilibrium). In this exercise, we use Widom’s test particle method (see section 8.5.1) to locate a vapor-liquid equilibrium. The results are compared with a Gibbs ensemble simulation.

1. Modify the *NVT* Monte Carlo program on the book’s website in such a way that the chemical potential can be calculated using Widom’s test particle method:

$$\mu = \mu_0 - \frac{\ln(\rho^{-1} \langle \exp[-\beta \Delta U^+] \rangle)}{\beta}, \quad (6.7.8)$$

where ρ denotes the number density of particles, U^+ is the change in potential energy due to the insertion of a test particle, and

$$\mu_0 = \frac{-\ln(\Lambda^3)}{\beta}. \quad (6.7.9)$$

In this exercise, we use instead the Widom expression for the fugacity f , Eq. (8.5.6):

$$f = \frac{\rho}{\langle \exp(-\beta \Delta U^+) \rangle}$$

- Make a plot of the fugacity and the pressure as a function of the density for $T = 0.8$.

- Why is it more difficult to calculate the fugacity at high densities than at low densities?
 - How can you use the information about f and P to locate the vapor-liquid coexistence densities?
2. Perform a Gibbs-ensemble simulation of the system at $T = 0.8$. In the Gibbs ensemble, the fugacity of the particles in box i is equal to [217]

$$f_i = \left\langle \frac{V_i}{n_i + 1} \exp[-\beta \Delta U_i^+] \right\rangle^{-1}, \quad (6.7.10)$$

in which n_i is the number of particles in box i and V_i is the volume of box i . Do the vapor-liquid coexistence densities and the fugacities agree with your previous results?

Chapter 7

Molecular Dynamics in various ensembles

The Molecular Dynamics technique discussed in Chapter 4 is an approximate scheme for solving Newton's equations of motion of a system of N particles in volume V . In the absence of external perturbations, the total energy E of a system evolving according to Newton's laws is a constant of motion. The MD simulations of Chapter 4 compute the time-averaged properties of a system at constant N , V , and E . If the system that we simulate is ergodic, then the time averages obtained in a standard MD simulation are equivalent to ensemble averages in the microcanonical (constant- NVE) ensemble. However, as was discussed in Chapter 6, it is often more convenient to perform simulations where we keep other thermodynamic parameters fixed, e.g., N , V and T or N , P and T . The question then arises if we can perform MD simulations in ensembles other than the microcanonical, and still obtain meaningful dynamical information. The answer to this question is: yes, but only up to a point. MD simulations have been extended to other ensembles, and these simulations will yield the correct static averages. However, care should be taken when extracting dynamical information from MD simulations that solve Newton's equations of motion. Below, we discuss some of the more commonly used MD methods that are used to perform MD in ensembles other than constant N , V , E . As will become clear, some of these methods become increasingly Newtonian as the system size is increased, whereas others give up on Newton altogether.¹

Two rather different solutions to this problem have been proposed. The simplest is based on the idea that dynamical simulation of other ensembles is possible by mixing Newtonian MD with occasional Monte-Carlo-like moves that ensure that the temperature (or pressure) is fixed. As a Monte Carlo move involves an instantaneous change in the system, it is clear that such an approach disrupts the continuous time evolution of the system. The second approach aims to fix the temperature of the system by ensuring that the total kinetic energy of the system satisfies the correct canonical distribution. Such a tuning of the

¹ Simulations under conditions other than constant NVE necessarily require an unphysical extension of Newton's equations of motion. Many different choices have been proposed: because these algorithms cannot be derived from the basic laws of mechanics, their choice is only limited by the creativity of their authors. Of the very many algorithms that have been proposed, some are popular, some less so—some yield the correct ensemble, others less so. We will not even attempt to give a comprehensive review, but just describe examples of the most important categories.

distribution of the kinetic energy can either be achieved by a reformulation of the Lagrangian equations of motion of the system, or by a stochastic rescaling of the global kinetic energy. Both methods yield a continuous time evolution that, although not really Newtonian, approaches Newtonian dynamics in certain well-defined limits.

More than just a barostat

We will pay special attention to the extended-Lagrangian method, first introduced by Andersen in the context of constant-pressure MD simulations [180], because the method of extended Lagrangians has become one of the most important tools for including additional dynamical variables in MD simulations.

Early examples of extended ensembles are the Parrinello-Rahman scheme to simulate crystalline solids under conditions of constant stress [178,179] and Nosé's version of isothermal MD [248]. But since then, the extended-Lagrangian method has been applied to a wide variety of classical (and quantum) simulations. Hence, when we discuss Andersen's approach to constant-pressure simulations, the reader should bear in mind that we are discussing more than *just* a numerical barostat. However, in our discussion of non-*NVE* MD simulations, we chose not to follow the historical order in which the different techniques were developed, but use a grouping by topic.

We start our discussion with thermostats.

7.1 Molecular Dynamics at constant temperature

When choosing a thermostat for an MD simulation, it is important to consider which physical laws are broken by the thermostat because, sadly, all thermostats break at least one physical law, and many break two or more (see e.g., [249]). Three “thermostat-crimes” stand out:

- 1. No momentum conservation** Most (but not all) “stochastic” thermostats do not conserve momentum. As a consequence, they cannot reproduce hydrodynamic flow (because hydrodynamics equals momentum conservation plus mass conservation). As a consequence, all transport properties that are sensitive to hydrodynamic interactions will not be reproduced correctly.
- 2. No Galilean invariance** Some algorithms rescale the total kinetic energy of a system. However, clearly, the kinetic energy associated with a uniform velocity of the system as a whole, is irrelevant for the temperature. This problem is usually be fixed by considering the “peculiar” velocity of particles, i.e., the velocity with respect to that of the center of mass of the system. But for out-of-equilibrium systems with a non-uniform driven flow, that correction can be carried out in several ways, depending on how one defines the relevant local reference velocity. Defining this reference velocity on too small a scale creates problems, as becomes clear by considering the limit where every particle defines its own reference velocity. Conversely, defining

the reference velocity on a scale larger than that of the flow inhomogeneities result in a situation where velocity rescaling changes the flow profile itself. Thermostats that act on the relative velocity of neighboring particles do not suffer from these problems related to a lack of Galilean invariance [249], but may suffer from others [250].

3. **Disruption of transport properties** Some thermostats act locally and drastically change the dynamics of all particles on a local scale. For such algorithms, one may still compute some transport properties, and these (e.g., the diffusion coefficient) may still be fairly realistic. However, others loose all meaning. For instance, it is meaningless to talk about the thermal conductivity of a system with a local thermostat: as energy is not conserved, heat transport does not satisfy Fourier's Law. In addition, systems that do not satisfy momentum conservation cannot obey the (Navier-)Stokes equation, and hence the viscosity coefficient has no meaning.

We mention these problems up front, because they are not always clearly advertised.

The dangers of fixing the kinetic energy

Before considering different schemes to perform Molecular Dynamics simulations at constant temperature, we should first specify what we mean by constant temperature. From a statistical mechanical point of view, there is no ambiguity: we can impose the temperature of a system by allowing it to exchange energy with a large heat bath (see section 2.2). Under those conditions, the probability of finding the system in a given energy state is given by the Boltzmann distribution and, for a classical system, the Maxwell-Boltzmann velocity distribution follows:

$$\mathcal{P}(p) = \left(\frac{\beta}{2\pi m} \right)^{3/2} \exp \left[-\beta p^2 / (2m) \right]. \quad (7.1.1)$$

The Maxwell-Boltzmann distribution establishes a simple relation between the imposed temperature T and the average (translational) kinetic energy per particle:

$$k_B T = m \langle v_\alpha^2 \rangle,$$

where m is the mass of the particle and v_α is the α -th component of its velocity. As discussed in Chapter 4, this relation is often used to measure the temperature in a (microcanonical) MD simulation. However, the condition of constant temperature is not equivalent to the condition that the kinetic energy per particle is constant. To see this, consider the relative variance of the kinetic energy per particle in a canonical ensemble. If we constrain the kinetic energy to be always equal to its average, then the variance vanishes by construction. Now consider a system that is in thermal equilibrium with a bath. The relative variance in the kinetic energy of any given particle is simply related to the second

and fourth moments of the Maxwell-Boltzmann distribution. For the second moment, $p^2 = \sum_{\alpha} p_{\alpha}^2$, we have

$$\langle p^2 \rangle = \int d\mathbf{p} p^2 \mathcal{P}(p) = \frac{3m}{\beta}$$

and for the fourth moment, $p^4 = (\sum_{\alpha} p_{\alpha}^2)^2$, we can write

$$\langle p^4 \rangle = \int d\mathbf{p} p^4 \mathcal{P}(p) = 15 \left(\frac{m}{\beta} \right)^2.$$

The relative variance of the kinetic energy of that particle is

$$\frac{\sigma_{p^2}^2}{\langle p^2 \rangle^2} \equiv \frac{\langle p^4 \rangle - \langle p^2 \rangle^2}{\langle p^2 \rangle^2} = \frac{15(m/\beta)^2 - (3m/\beta)^2}{(3m/\beta)^2} = \frac{2}{3}.$$

If we would use the kinetic energy per particle as a measure of the instantaneous temperature, then we would find that, in a canonical ensemble, this temperature (denoted by T_k) fluctuates. Its relative variance is

$$\begin{aligned} \frac{\sigma_{T_k}^2}{\langle T_k \rangle_{NVT}^2} &\equiv \frac{\langle T_k^2 \rangle_{NVT} - \langle T_k \rangle_{NVT}^2}{\langle T_k \rangle_{NVT}^2} \\ &= \frac{N \langle p^4 \rangle + N(N-1) \langle p^2 \rangle \langle p^2 \rangle - N^2 \langle p^2 \rangle^2}{N^2 \langle p^2 \rangle^2} \\ &= \frac{1}{N} \frac{\langle p^4 \rangle - \langle p^2 \rangle^2}{\langle p^2 \rangle^2} = \frac{2}{dN}. \end{aligned}$$

So indeed, in a canonical ensemble of a finite system, the instantaneous kinetic temperature T_k fluctuates. In fact, if we were to keep the average kinetic energy per particle rigorously constant, as is done in the so-called isokinetic MD scheme [33] or the more naive velocity-scaling schemes, then we would not simulate a true constant-temperature ensemble. Obviously, problems can be expected if isokinetic simulations are used to measure equilibrium averages that are sensitive to fluctuations, for instance, when computing the kinetic contribution to the heat capacity (see Eq. (5.1.7)).

But there is a less obvious, yet serious problem, with some of the most widely used algorithms that keep the kinetic energy constant by instantaneous or gradual rescaling of the velocities [251]. It has been shown by Harvey et al. [252] (see also [253]) that naive velocity rescaling can lead to the so-called “flying ice-cube” effect, whereby kinetic energy from vibrational, intra-molecular degrees of freedom is transferred to translational kinetic energy, hence the name flying ice-cube: the internal degrees of freedom of a molecule

become cold, but the translational motion becomes hot, thereby violating detailed balance. Due to the flying ice-cube effect, even the static properties of such a system would then be sampled incorrectly (see [252]). The flying ice-cube effect is usually small, and is observed for some velocity-rescaling algorithms, but as explained in ref. [253], there are a number of velocity-rescaling algorithms that do not suffer from this unphysical effect. In practice, a much more prevalent flying ice-cube effect is due to the inappropriate use of the “velocities” in the velocity-Verlet algorithm to compute the temperature (see Eq. (5.1.2) and ref. [115]).

Unphysical temperature regulation is often used to prepare the system at roughly the desired temperature (i.e., during equilibration), after which a more reliable thermostat may be used, or the system is run at constant NVE . However, it is probably unwise to “equilibrate” a system using an algorithm that creates a seemingly constant temperature state that is, in fact, very far from thermal equilibrium. As efficient MD schemes exist to generate a true canonical distribution, there is little need to use more suspect techniques to fix the temperature during equilibration. Below, we limit our discussion to stochastic thermostats, and to kinetic energy rescaling schemes that do not suffer from this drawback.

7.1.1 Stochastic thermostats

7.1.1.1 Andersen thermostat

In the constant-temperature method proposed by Andersen [180] the system is coupled to a heat bath that imposes the desired temperature. The coupling to a heat bath is represented by stochastic impulsive forces that occasionally act on randomly selected particles. These stochastic “collisions” with the heat bath can be considered as Monte Carlo moves that transport the system from one constant-energy shell to another. Between stochastic collisions, the system evolves at constant energy according to the normal Newtonian laws of motion. The stochastic collisions ensure that all accessible constant-energy shells are visited according to their Boltzmann weight. Importantly, the Andersen thermostat does not conserve momentum. Hence it should not be used when hydrodynamic effects are expected to be important, as is the case when there are driven flows in the system, but also for hydrodynamic mode-coupling effects at equilibrium.

The first step in preparing a constant-temperature simulation with the Andersen’s thermostat is to select the strength of the coupling to the heat bath. This coupling strength is determined by the frequency of the momentum-changing stochastic collisions. We denote this frequency by ν . In the Andersen scheme, the probability per unit time to generate a collision is independent of the timing of previous collisions: in other words, collisions are assumed to be Poisson distributed. For a Poisson process, the distribution of time intervals between two

successive collisions, $P(t; v)$, is of the form [254,255]

$$P(t; v) = v \exp[-vt], \quad (7.1.2)$$

where $P(t; v)dt$ is the probability that the next collision will take place in the interval $[t, t + dt]$, given that there was a collision at $t = 0$.

A constant-temperature simulation now consists of the following steps:

1. Start with an initial set of positions and momenta $\{\mathbf{r}^N(0), \mathbf{p}^N(0)\}$ and integrate the equations of motion for a time Δt .
2. m particles are selected at random to undergo a collision with the heat bath, where m is given by the Poisson distribution $P(m) = \bar{m}^m e^{-\bar{m}}$, with $\bar{m} = v\Delta t$.
3. If particle i has been selected to undergo a collision, its new velocity will be drawn from a Maxwell-Boltzmann distribution corresponding to the desired temperature T . All other particles are unaffected by this collision.

The mixing of Newtonian dynamics with stochastic collisions turns the Molecular Dynamics simulation into a stochastic Markov process [67]. As shown in [180], a canonical distribution in phase space is invariant under repeated application of the stochastic collisions, which, combined with the fact that the Markov chain is also irreducible and aperiodic [180,254,255], implies that the Andersen algorithm does generate a canonical distribution. In Algorithms 16 and 17, we show how the Andersen method can be implemented in a Molecular Dynamics simulation.

Example 14 (Use of Andersen thermostat). The Andersen thermostat [180] is arguably the simplest MD thermostat that has been proven to yield a canonical distribution. This implies that both the kinetic energy and the potential energy of the system under consideration are Boltzmann-distributed.

However, the Andersen algorithm conserves neither momentum nor energy, as it attributes new velocities drawn from a Maxwell-Boltzmann distribution to randomly selected particles. We refer to these updates as stochastic “collisions”. The effect of the stochastic collisions is that the dynamical properties of a system with an Andersen thermostat may differ substantially, or even catastrophically, from those of a system without a thermostat, or with a “global” thermostat [256,257].

In the case of diffusion, it is easy to understand intuitively that the Andersen thermostat will decrease the self-diffusion coefficient D : a Green-Kubo relation relates D to the integral of the velocity-auto-correlation function (VACF). The longer velocities persist, the larger D . Conversely, any effect that destroys the persistence in the velocity will decrease D . And destroying the persistence in \mathbf{v} is precisely what the Andersen thermostat does: the higher v , the frequency of stochastic collisions, the lower D . This effect is illustrated in Fig. 7.1.

In practical cases, ν is usually chosen such that the timescale τ_E for the decay of energy fluctuations in a simulation of a system with linear dimension L is comparable to that of thermal fluctuations with a wavelength equal to L , in an unbounded medium: $\tau_E \approx [L^2(N/V)C_P]/\lambda$, where C_P is the heat capacity per molecule (at constant pressure) and λ is the thermal conductivity.

For reasonably large systems such values of τ_E can only be achieved with rather low collision rates per particle, in which case the effect of collisions on the dynamics may be small [180], but at the same time, the thermostat becomes rather inefficient.

One should not use the Andersen thermostat when computing quantities such as the thermal conductivity or the viscosity.

The reason is that these quantities provide a measure for the rate at which energy or momentum diffuse. But such a description only makes sense if energy and momentum are conserved. With an Andersen thermostat, a local perturbation in the energy or momentum does not diffuse away, rather it is screened exponentially. This effect is not described by the (normal) diffusion equation.

Summarizing: do not use the Andersen method when computing transport properties. For static properties, it is fine - and very easy to implement.

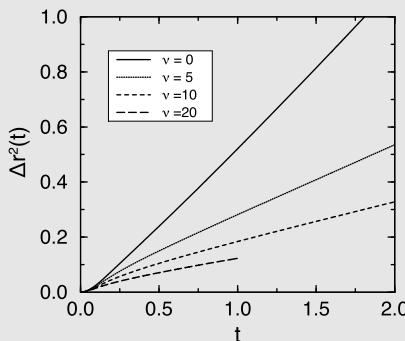


FIGURE 7.1 Mean-squared displacement as a function of the time of a Lennard-Jones fluid ($T = 2.0$ and $N = 108$) for various values of the collision frequency ν of the Andersen thermostat.

Fig. 7.1 shows that the frequency of stochastic collisions has a strong effect on the time dependence of the mean-squared displacement. The mean-squared displacement becomes only independent of ν in the limit of very low stochastic collision rates. Yet, all static properties such as the pressure or potential energy are rigorously independent of the stochastic collision frequency.

For more details, see SI (Case Study 11).

7.1.1.2 Local, momentum-conserving stochastic thermostat

By construction, the Andersen thermostat conserves neither energy nor momentum. The lack of energy conservation is unavoidable. However, as Lowe

Algorithm 16 (Molecular Dynamics: Andersen thermostat)

program md_Andersen	Constant-temperature MD
[Initialize system]	as in Algorithm 3
[Compute forces and energy]	as in Algorithm 5
t=0	
while t < tmax do	MD loop
switch = 1	
integrate-A (switch,temp)	Propagates half time step (Algorithm 17)
FandE	Algorithm 5
switch = 2	
integrate-A (switch,temp)	Propagates second half time step
t=t+dt	
sample	sample observables
end while	
end program	

Specific Comment (for general comments, see p. 7)

1. The desired temperature temp is set during initialization.
2. Because the Andersen thermostat acts on the velocities, it uses the velocity-Verlet algorithm (see section 4.3). See Eqs. (4.3.4) and (4.3.5). The present algorithm proceeds in two steps, in step 1 (**integrate-A**(1,temp)), we know the forces and velocities at time t, and we update $x(t)$ and determine

$$v' = v(t) + \frac{f(t)}{2m} dt.$$

Then, in **FandE**, we determine the forces at $t=t+dt$; and finally we determine in step 2, **integrate-A**(2,temp), the velocities at time $t=t+dt$,

$$v(t+dt) = v' + \frac{f(t+dt)}{2m} dt.$$

In the Andersen algorithm we add an extra step: at the end of the velocity-Verlet update of positions and velocities, we attribute new velocities, drawn from a Maxwell-Boltzmann distribution at temperature temp, to a randomly chosen subset of the particles. As a consequence, the velocities are updated at integer numbers of time steps. The function **integrate-A** is described in Algorithm 17.

showed [258], it is possible to follow Andersen's argument to construct a momentum-conserving stochastic algorithm that keeps the system at a fixed temperature. The key difference with the Andersen thermostat is that, rather than giving one particle a new velocity, drawn from a Maxwell-Boltzmann dis-

Algorithm 17 (Equations of motion: Andersen thermostat)

function integrate-A(switch,temp)	integrate equations of motion with Andersen thermostat
if switch == 1 then	first step velocity Verlet
for 1 ≤ i ≤ npart do	
x(i)=x(i)+dt*v(i)+	update positions current time
+ dt*dt*f(i)/2	
v(i)=v(i)+dt*f(i)/2	first update velocity
enddo	
else if switch == 2 then	second step velocity Verlet
for 1 ≤ i ≤ npart do	
v(i)=v(i)+dt*f(i)/2	second update velocity
enddo	
for 1 ≤ i ≤ npart do	
if $\mathcal{R} < \text{nu}*\text{dt}$ then	test for collision with bath
v(i)=gauss(0., $\sqrt{\text{temp}}$)	give particle new velocity drawn from Gaussian distribution
endif	
enddo	
endif	
end function	

Specific Comments (for general comments, see p. 7)

1. For reasons explained below Algorithm 16, **integrate-A** uses the velocity-Verlet algorithm [113].
2. The function **gauss(0., sigma)** returns a value taken from a Gaussian distribution with zero mean and standard deviation **sigma** (see e.g., Algorithm 36).
3. The collisions with the heat bath are assumed to be Poisson distributed (7.1.2). The collision frequency **nu** is set at the beginning of the simulation.
4. In this algorithm, neither the total energy nor the total momentum is conserved.
5. Because the Andersen thermostat does not conserve momentum it cannot be used to describe hydrodynamic flow. The Lowe-Andersen thermostat (Section 7.1.1.2) does not suffer from this drawback.

tribution, the Lowe-Andersen thermostat considers the radial relative velocity of a pair of particles (i.e., the component of the relative velocity along the line joining the centers of mass of the two particles), and replaces it with a new relative velocity, drawn from a Maxwell-Boltzmann distribution of relative velocities. The choice of the pair of particles for which to reset the relative velocity is to some extent arbitrary, provided that it is not biased. Lowe's choice was to select

two random particles within a distance r_c , where r_c can be chosen freely. By making r_c comparable to the average nearest neighbor distance between particles, the effect of the thermostat on the viscosity of the system - which is now well-defined—can be minimized (but not eliminated). On the whole though, the transport properties of a system with a Lowe-Andersen thermostat are closer to the unperturbed transport properties than is the case for the original Andersen thermostat [259].

One slightly subtle point about both the Lowe and the Andersen thermostats is that, as they are local thermostats, they act strongest where there are most particles. In the Andersen case, the rate of thermalization should scale with the local density, whereas for the Low-Andersen thermostat, this rate scales with the square of the local density (unless r_c is comparable to the system size).

7.1.1.3 Langevin dynamics

The idea behind Langevin dynamics pre-dates the introduction of the Molecular Dynamics method by almost half a century. In 1908, Paul Langevin published an article [260] proposing a simple equation describing the Brownian motion of a spherical colloidal particle in a viscous fluid. The original Langevin equation reads (considering only one component of the velocity):

$$m\dot{v}_x(t) = -\gamma v_x(t) + \mathcal{R}_x(t), \quad (7.1.3)$$

where v_x is the x -component of the velocity of the particle, m its mass, γ the friction coefficient, which accounts for viscous drag (for a particle with non-slip hydrodynamic boundary conditions, this drag coefficient is given by the Stokes expression $\gamma = 6\pi\eta a$, where η is the viscosity of the fluid and a the radius of the colloidal particle).

In simulations, the Langevin equation includes the effect of the conservative forces between the solute particles:

$$m\dot{v}_x(t) = -\gamma v_x(t) - \left(\frac{\partial \mathcal{U}(\mathbf{r}^N)}{\partial x} \right) + \mathcal{R}_x(t), \quad (7.1.4)$$

where \mathbf{r}^N denotes the coordinates of the N solute particles. The great innovation of the [Langevin Equation \(LE\)](#) was the introduction of the so-called *random force* \mathcal{R} , which represents the rapidly varying forces exerted on the colloid by the molecules of the solvent. \mathcal{R} has zero mean; it is called the *random force* to distinguish it from the hydrodynamic drag force on the colloid. The drag force is also due to the solvent but, unlike the random force, it is (linearly) correlated with the velocity of the colloid. In the absence of the random force, the velocity of the colloid would quickly decay to zero. However, the random force ensures that the colloid keeps moving, even though it will lose the memory of its original velocity.

Eq. (7.1.3) describes the stochastic time-evolution of the velocity of an isolated particle if a fluid in the absence of forces other than the random force \mathcal{R} and the friction force $-\gamma v$. Importantly, γ and \mathcal{R} are related:

$$\langle \mathcal{R}_x(0)\mathcal{R}_x(t) \rangle = 2\gamma k_B T \delta(t). \quad (7.1.5)$$

In simulations, we will use a discretized form of the Langevin equation, and we use that form to explain Eq. (7.1.5). The random force \mathcal{R}_x can act on a particle at the end of every time-step, resulting in a random momentum transfer $\mathcal{P}_x = \mathcal{R}_x \Delta t$. If a system is in equilibrium at a temperature T , the random momentum transfer should, on average, compensate for the kinetic energy loss due to friction, or

$$(m/2) \left\langle p_x^2 \right\rangle e^{-2(\gamma/m)\Delta t} - 1 \approx - \left\langle p_x^2 \right\rangle \gamma \Delta t,$$

where (for convenience) we have assumed that $(\gamma/m)\Delta t \ll 1$. The energy gain due to the random momentum transfer is

$$(1/2m) \left\langle (p_x + \mathcal{P}_x)^2 - p_x^2 \right\rangle = (1/2m) \left\langle \mathcal{P}_x^2 \right\rangle. \quad (7.1.6)$$

The two should balance at temperature T :

$$(1/2m) \left\langle \mathcal{P}_x^2 \right\rangle = m k_B T \gamma \Delta t, \quad (7.1.7)$$

$$\left\langle \mathcal{P}_x^2 \right\rangle = 2k_B T \gamma \Delta t, \quad (7.1.8)$$

which is the discretized form of Eq. (7.1.5).

In a Langevin simulation, we have to update the positions and momenta in the usual way, but in addition, we must include a step where we update the particle velocities due to the frictional and random forces (in the discrete implementation, it is better to think in terms of random momentum transfers). There are several ways to decompose the discrete-time propagation of the Langevin equation into these elementary sub-steps, and some decompositions are much better behaved than others.

In practice, we must decide the order in which we carry out position updates (R), velocity updates due to the conservative forces (V), and velocity updates due to the friction and random forces (O) (we use the O, V, R-notation of ref. [261]). Which Langevin algorithm is preferred depends (not surprisingly) on which features of the algorithm are considered to be most important [23,261]. In the literature, consensus has emerged that the approach by Leimkuhler and Matthews [23] (VRORV) is to be preferred [262]. There are many other popular Langevin integration schemes (see e.g., [261]) that can be viewed as limiting cases or approximate versions of the OVRVO-scheme of ref. [261].

One crucial drawback of the Langevin equation is that it does not conserve momentum. Another (closely related, but not identical) drawback is that it does

not account for the hydrodynamic interaction between different solute particles. Hydrodynamic interactions between solute particles are due to the fact that if one particle moves, it creates a flow field that exerts a drag force on other solute particles. If we account for these (extremely long-ranged) velocity-dependent drag forces, we should also account for the fact that the random forces on different particles are now correlated. Although methods have been developed to deal with hydrodynamic interactions in the context of the Langevin equation we will not discuss them because, typically, they become very cumbersome in complex, confined geometries, where they would be most important.² Rather, in Chapter 16 we will discuss simple, highly coarse-grained methods (Dissipative Particle Dynamics, Stochastic Rotation Dynamics), which account for the particulate nature of the solvent in the cheapest possible way. The method described in Chapter 16 do conserve momentum and reproduce hydrodynamic interactions, even in complex geometries.

Brownian dynamics

One important limit of Langevin dynamics is the case where the translational motion is strongly overdamped, i.e., when the inertial term in the Langevin equation can be ignored. In that case, the Langevin equation reduces to

$$0 = -\gamma \dot{x}(t) - \left(\frac{\partial U(\mathbf{r}^N)}{\partial x} \right) + \mathcal{R}_x(t), \quad (7.1.9)$$

which is the basic equation for **Brownian Dynamics (BD)** simulations. The relation between the random forces and the friction coefficient in Brownian Dynamics is the same as in the Langevin case. Note that Eq. (7.1.9) describes the time evolution of positions only. It is widely used to simulate the diffusive behavior of systems of colloids, polymers, or proteins. Like Langevin simulations, BD simulations become more painful when hydrodynamic interaction must be included.

7.1.2 Global kinetic-energy rescaling

The earliest thermostats in MD simulations, used an instantaneous or gradual global rescaling of all particle velocities to keep the kinetic energy of the system constants [251,264]. However, these early thermostats do not correspond to a well-defined ensemble and, more seriously, suffer from serious artifacts [252, 253].

Nevertheless, it is very attractive to have MD algorithms that sample the canonical ensemble by some global kinetic energy rescaling, rather than the local schemes implemented in the Andersen or Langevin thermostats. The main

² Moreover, the use of stochastic differential equations becomes problematic under conditions where the friction-coefficient depends on the particle coordinates, because of the Itô-Stratonovich indeterminacy [263].

reason why global kinetic energy control is attractive is that a) the effect of such thermostats is, on the whole, very mild as the velocities of individual particles are only changed by a small amount, and b) such thermostats can be designed to conserve momentum.

The earliest well-behaved global thermostat was the one due to Nosé [248, 265], which we will discuss briefly because it provides a nice illustration of the use of an extended Lagrangian to design new forms of molecular dynamics. Nosé's approach was inspired by the earlier use by Andersen of extended Lagrangians to simulate systems at constant pressure (see section 7.2). However, the constant-temperature method developed by Nosé is very different and deserves to be discussed separately, the more so as other extended Lagrangian methods are widely used in molecular simulations. Yet, even though the Nosé thermostat, in the form proposed by Hoover [257], is still widely used, we will not discuss it in great detail because it is less robust (and less simple) than the stochastic kinetic energy-rescaling thermostat of Bussi et al. [256].

7.1.2.1 Extended Lagrangian approach

Newton's equations of motion conserve energy. Hence, to perform MD at constant temperature, we must modify the equations of motion. The most robust way to construct new equations of motion is to start from the Lagrangian formulation of mechanics. The advantage of following that route is that, once the Lagrangian has been modified, the modified Hamiltonian follows and, crucially, is a constant of motion. This observation already indicates that for NVT-MD, the value of Hamiltonian cannot equal the energy E of the N -body system. The trick is then to find a suitable form of the Lagrangian that generates a dynamics for which the time averages are equal to those of an N -body system at constant NVT . The extended Lagrangian approach was pioneered by Andersen [180] to construct an algorithm for constant-pressure dynamics. Andersen also used a thermostat, but that was stochastic, as we have seen in section 7.1.1.1. We briefly discuss Andersen's constant-pressure approach in section 7.2. However, we start with Nosé's Lagrangian approach for constant-temperature MD.³

To construct isothermal Molecular Dynamics, Nosé proposed introducing an additional coordinate s in the Lagrangian of a classical N -body system:

$$\mathcal{L}_{\text{Nose}} = \sum_{i=1}^N \frac{m_i}{2} s^2 \dot{\mathbf{r}}_i^2 - \mathcal{U}(\mathbf{r}^N) + \frac{Q}{2} \dot{s}^2 - \frac{L}{\beta} \ln s, \quad (7.1.10)$$

where Q is an effective “mass” quantifying the inertia associated with the motion of s . At this stage, L is a parameter, which will be fixed later.

³ To understand Nosé's approach, some familiarity with the Lagrangian and Hamiltonian formulation of classical mechanics is required. Appendix A gives a very basic review of this formulation of classical mechanics. However, for details, we refer the reader to one of the many excellent textbooks, e.g., [54].

The momenta conjugate to \mathbf{r}_i and s follow directly from Eq. (7.1.10):

$$\mathbf{p}_i \equiv \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}_i} = m_i s^2 \dot{\mathbf{r}}_i \quad (7.1.11)$$

$$p_s \equiv \frac{\partial \mathcal{L}}{\partial \dot{s}} = Q \dot{s}. \quad (7.1.12)$$

Once we have the momenta, we can write down the Hamiltonian of the extended system of the N particles plus additional coordinate s :

$$\mathcal{H}_{\text{Nose}} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i s^2} + \mathcal{U}(\mathbf{r}^N) + \frac{p_s^2}{2Q} + L \frac{\ln s}{\beta}. \quad (7.1.13)$$

We consider a system containing N atoms. As $\mathcal{H}_{\text{Nose}}$ is conserved, the dynamics generated by this Hamiltonian samples a *microcanonical* ensemble, but in the *extended* system of $2dN + 2$ coordinates and momenta. The partition function of this ensemble⁴ is:

$$\begin{aligned} Q_{\text{Nose}} &= \frac{1}{N!} \int dp_s ds d\mathbf{p}^N d\mathbf{r}^N \delta(E - \mathcal{H}_{\text{Nose}}) \\ &= \frac{1}{N!} \int dp_s ds d\mathbf{p}'^N d\mathbf{r}^N s^{dN} \\ &\quad \times \delta \left[\sum_{i=1}^N \frac{\mathbf{p}'_i^2}{2m_i} + \mathcal{U}(\mathbf{r}^N) + \frac{p_s^2}{2Q} + \frac{L}{\beta} \ln s - E \right], \end{aligned} \quad (7.1.14)$$

where, in the second line, we have defined

$$\mathbf{p}' \equiv \mathbf{p}/s.$$

The rationale for the introduction of \mathbf{p}' will become clear later. We can then write the part of the Hamiltonian that depends on \mathbf{p}' and \mathbf{r} as

$$\mathcal{H}(\mathbf{p}', \mathbf{r}) \equiv \sum_{i=1}^N \frac{\mathbf{p}'_i^2}{2m_i} + \mathcal{U}(\mathbf{r}^N). \quad (7.1.15)$$

When a δ function has as argument a function $h(s)$, which has a single root at s_0 , we can write

$$\delta[h(s)] = \delta(s - s_0)/|h'(s_0)|,$$

⁴ Here, we assume for simplicity that $\mathcal{H}_{\text{Nose}}$ is the only conserved dynamical variable. However, typically, the total momentum of the system is also conserved. In the SI L.6.1, we consider the more general case where there are more conserved quantities.

where $h'(s)$ denotes the derivative of h with respect to s . If we substitute this expression into Eq. (7.1.14) and use Eq. (7.1.15), we find, for the partition function,

$$\begin{aligned} Q_{\text{Nose}} &= \frac{1}{N!} \int dp_s d\mathbf{p}'^N d\mathbf{r}^N ds \frac{\beta s^{dN+1}}{L} \\ &\quad \times \delta \left\{ s - \exp \left[-\beta \frac{\mathcal{H}(\mathbf{p}', \mathbf{r}) + p_s^2/(2Q) - E}{L} \right] \right\} \\ &= \frac{1}{N!} \frac{\beta \exp[\beta E(dN+1)/L]}{L} \int dp_s \exp \left[-\beta \frac{dN+1}{L} p_s^2/(2Q) \right] \\ &\quad \times \int d\mathbf{p}'^N d\mathbf{r}^N \exp \left[-\beta \frac{dN+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r}) \right] \\ &= C \frac{1}{N!} \int d\mathbf{p}'^N d\mathbf{r}^N \exp \left[-\beta \frac{dN+1}{L} \mathcal{H}(\mathbf{p}', \mathbf{r}) \right]. \end{aligned} \quad (7.1.16)$$

Hence, we see that Nosé's ingenious choice of additional variables ensures that the dynamics generates a probability density in $\{\mathbf{p}', \mathbf{r}\}$ space, which is proportional to $\exp(-\beta[(dN+1)/L]\mathcal{H}(\mathbf{p}', \mathbf{r}))$. Hence, with the choice $L = dN+1$, the probability-density in $\{\mathbf{p}', \mathbf{r}\}$ -space equals $\exp(-\beta\mathcal{H}(\mathbf{p}', \mathbf{r}))$! The ensemble average of a dynamical variable A can then be written as:

$$\begin{aligned} \langle A(\mathbf{p}/s, \mathbf{r}) \rangle_{\text{Nose}} &= \frac{(1/N!) \int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}', \mathbf{r}) \exp[-\beta\mathcal{H}(\mathbf{p}', \mathbf{r})]}{Q(NVT)} \\ &= \langle A(\mathbf{p}', \mathbf{r}) \rangle_{NVT}. \end{aligned} \quad (7.1.17)$$

However, there is a snag: the phase-space coordinates \mathbf{p}' , \mathbf{r} , which make the Hamiltonian look familiar, do not satisfy Hamilton's equation of motion, only the original \mathbf{p} and \mathbf{r} do.

The correct time average that corresponds to the canonical ensemble average is:

$$\bar{A} = \lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt A(\mathbf{p}(t)/s(t), \mathbf{r}(t)) \equiv \langle A(\mathbf{p}/s, \mathbf{r}) \rangle_{\text{Nose}}. \quad (7.1.18)$$

This is somewhat unpleasant, because the ratio between \mathbf{p} and \mathbf{p}' equals s , and s depends on time.

It is instructive to consider the role of the variable s . In the ensemble average in Eq. (7.1.17), the phase space is spanned by the coordinates \mathbf{r} and the scaled momenta \mathbf{p}' . As the scaled momentum is related most directly to observable properties (in particular, the kinetic energy equals $\mathbf{p}'^2/(2m)$, we refer to \mathbf{p}' as the real momentum, while \mathbf{p} is interpreted as a virtual momentum. We make a similar distinction between real and virtual for the other variables. Real variables are indicated by a prime, to distinguish them from their unprimed, virtual

counterparts. The real and virtual variables are related as follows:

$$r' = r \quad (7.1.19)$$

$$p' = p/s \quad (7.1.20)$$

$$s' = s \quad (7.1.21)$$

$$\Delta t' = \Delta t/s. \quad (7.1.22)$$

From Eq. (7.1.22) it follows that s can be interpreted as a scaling factor of the time step. Eq. (7.1.18) then suggests that time-averages should be obtained by sampling the observables at integer multiples of the (virtual) time step Δt , which corresponds to real-time steps that are not constant. However, it is also possible to sample at equal intervals in real-time. In that case, we measure a slightly different average. Instead of Eq. (7.1.18) we define

$$\lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' A [\mathbf{p}(t')/s(t'), \mathbf{r}(t')]. \quad (7.1.23)$$

Eq. (7.1.22) shows that the real and virtual measuring times τ' and τ , respectively, are related through

$$\tau' = \int_0^\tau dt 1/s(t).$$

This gives, for Eq. (7.1.23),

$$\begin{aligned} & \lim_{\tau' \rightarrow \infty} \frac{1}{\tau'} \int_0^{\tau'} dt' A [\mathbf{p}(t')/s(t'), \mathbf{r}(t')] \\ &= \lim_{\tau' \rightarrow \infty} \frac{\tau}{\tau'} \frac{1}{\tau} \int_0^\tau dt A [\mathbf{p}(t)/s(t), \mathbf{r}(t)]/s(t) \\ &= \frac{\lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt A [\mathbf{p}(t)/s(t), \mathbf{r}(t)]/s(t)}{\lim_{\tau \rightarrow \infty} \frac{1}{\tau} \int_0^\tau dt 1/s(t)} \\ &= \langle A(\mathbf{p}/s, \mathbf{r})/s \rangle / \langle 1/s \rangle. \end{aligned} \quad (7.1.24)$$

If we consider again the partition function (7.1.16), we can write for the ensemble average,

$$\begin{aligned} \frac{\langle A(\mathbf{p}/s, \mathbf{r})/s \rangle}{\langle 1/s \rangle} &\equiv \frac{\left\{ \frac{\int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}', \mathbf{r}) \exp[-\beta \mathcal{H}(\mathbf{p}', \mathbf{r})] dN/L}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}', \mathbf{r})] (dN+1)/L} \right\}}{\left\{ \frac{\int d\mathbf{p}'^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}', \mathbf{r})] dN/L}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}', \mathbf{r})] (dN+1)/L} \right\}} \\ &= \frac{\int d\mathbf{p}'^N d\mathbf{r}^N A(\mathbf{p}/s, \mathbf{r}) \exp[-\beta \mathcal{H}(\mathbf{p}', \mathbf{r})] dN/L}{\int d\mathbf{p}'^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}', \mathbf{r})] dN/L} \\ &= \langle A(\mathbf{p}/s, \mathbf{r}) \rangle_{NVT}. \end{aligned} \quad (7.1.25)$$

Note that in this case the canonical average is recovered if we choose $L = dN$. Therefore, if we use a sampling scheme based on equal time steps in real time, we have to use a different value for L . In Illustration 7, we discuss a system in which the correct choice of the number of degrees of freedom is essential.

Illustration 7 (Diffusion coefficients of confined gases). The total translational kinetic energy of a classical N -body system is equal to $\sum_{i=1}^N \sum_{\alpha=1}^d p_{i,\alpha}^2 / (2m_i)$. The equipartition law of classical statistical mechanics states that the average thermal energy associated with every term in this sum is equal to $(1/2)k_B T$. This expression would be correct if every momentum component were an independent degree of freedom.

For macroscopic systems, the naive equipartition law is reasonable as the number of degrees of freedom in the kinetic energy is approximately equal to dN . However, if we carry out a molecular dynamics simulation on a small system, we must be more careful. In simulations of bulk systems, we often use periodic boundary conditions and we fix the total momentum of the system. This constraint on the total momentum fixes L degrees of freedom. In three dimensions, the actual number of degrees of freedom in the kinetic energy is then $3N - 3$.

However, the number of degrees of freedom changes if we simulate a confined system, for example, a gas adsorbed in a pore. Most pores are modeled as rigid systems; hence, for the gas molecules inside the pore, the interactions with the atoms of the pore represent an external field. Because of this external field, the momentum of the center of mass of the N atoms is not fixed, and the degrees of freedom of these gas molecules are $3N$. We need the correct number of degrees of freedom in a simulation to compute the temperature and, in some algorithms, as input in a constant temperature simulation (e.g., Nosé-Hoover). If we simulate systems larger than 1,000 particles, the difference between N and $N - 3$ will typically be too small to have any visible impact (although is not an excuse to carry out a simulation incorrectly).

There are some systems for which one would like to simulate a much smaller number of particles. An example is the calculation of the diffusion coefficient in a porous material. Suppose you are interested in comparing with experimental diffusion coefficients of molecules adsorbed in such a porous medium. In that case, this comparison becomes easier if we extrapolate the computed diffusion coefficient to the limit of zero loading [193]. Hence, simulating a small number of gas molecules is of direct relevance. And for such a small number, the difference between $3N$ and $3N - 3$ can be significant. In Fig. 7.2 Xu et al. [107] compared the diffusion coefficient computed with a Nosé-Hoover simulation using the correct and incorrect number of degrees of freedom. In the Nosé-Hoover Hamiltonian, the partition function is given by (see Eq. (7.1.16))

$$Q_{\text{Nose}} = C \frac{1}{N!} \int d\mathbf{p}'^N dr^N \exp \left[-\frac{dN + 1}{Lk_B T} \mathcal{H}(\mathbf{p}', \mathbf{r}) \right].$$

Where L is the number of degrees of freedom, if we use $L = 3N - 3$, while the actual number is $3N$, we are effectively simulating at a lower temperature T' than we think we are.^a

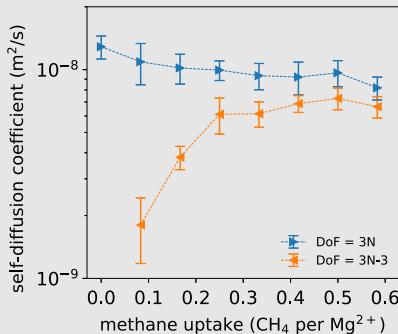


FIGURE 7.2 Self-diffusion coefficients of CH₄ in the metal-organic M2(dobdc) (M: Mg, Ni, and Zn) as a function of the loading. We compare the results obtained with the correct and incorrect degrees of freedom. The figure is based on the data of refs. [107,266].

The results in Fig. 7.2 show that the diffusion coefficient seems to decrease if we decrease the number of particles. This would be a surprising observation, as one would expect the diffusion coefficient to increase. However, with the incorrect number of degrees of freedom, the effective temperature decreases with decreasing number of particles—the lower the temperature, the lower the diffusion coefficient. The results with the correct degrees of freedom do show the expected behavior.

^a Spotting the problem with the temperature is difficult if the same incorrect expression is used to both set and report the temperature. The simulations were done at the wrong temperature, but this incorrect temperature was also miscalculated, and these two effects cancel.

From the Hamiltonian (7.1.13), we can derive the equations of motion for the virtual variables \mathbf{p} , \mathbf{r} , and t :

$$\begin{aligned} \frac{d\mathbf{r}_i}{dt} &= \frac{\partial \mathcal{H}_{\text{Nose}}}{\partial \mathbf{p}_i} = \mathbf{p}_i / (m_i s^2) \\ \frac{d\mathbf{p}_i}{dt} &= -\frac{\partial \mathcal{H}_{\text{Nose}}}{\partial \mathbf{r}_i} = -\frac{\partial \mathcal{U}(\mathbf{r}^N)}{\partial \mathbf{r}_i} \\ \frac{ds}{dt} &= \frac{\partial \mathcal{H}_{\text{Nose}}}{\partial p_s} = p_s / Q \\ \frac{dp_s}{dt} &= -\frac{\partial \mathcal{H}_{\text{Nose}}}{\partial s} = \left(\sum_i \mathbf{p}_i^2 / (m_i s^2) - \frac{L}{\beta} \right) / s. \end{aligned}$$

In terms of the real variables, these equations of motion can be written as

$$\frac{d\mathbf{r}'_i}{dt'} = s \frac{d\mathbf{r}_i}{dt} = \mathbf{p}_i / (m_i s) = \mathbf{p}'_i / m_i \quad (7.1.26)$$

$$\begin{aligned} \frac{d\mathbf{p}'_i}{dt'} &= s \frac{d\mathbf{p}_i / s}{dt} = \frac{d\mathbf{p}_i}{dt} - \frac{1}{s} \mathbf{p}_i \frac{ds}{dt} \\ &= -\frac{\partial \mathcal{U}(\mathbf{r}'^N)}{\partial \mathbf{r}'_i} - (s' p'_s / Q) \mathbf{p}'_i \end{aligned} \quad (7.1.27)$$

$$\frac{1}{s} \frac{ds'}{dt'} = \frac{s}{s} \frac{ds}{dt} = s' p'_s / Q \quad (7.1.28)$$

$$\begin{aligned} \frac{d(s' p'_s / Q)}{dt'} &= \frac{s}{Q} \frac{dp_s}{dt} \\ &= \left(\sum_i \mathbf{p}'_i^2 / m_i - \frac{L}{\beta} \right) / Q. \end{aligned} \quad (7.1.29)$$

For these equations of motion, the following quantity is conserved:

$$H'_{\text{Nose}} = \sum_{i=1}^N \frac{\mathbf{p}'_i^2}{2m_i} + \mathcal{U}(\mathbf{r}'^N) + \frac{s'^2 p'_s^2}{2Q} + L \frac{\ln s'}{\beta}. \quad (7.1.30)$$

We stress again that H'_{Nose} expressed as a function of \mathbf{p}' and \mathbf{r}' cannot be viewed as a Hamiltonian, in the sense that it the time evolution of \mathbf{p}' and \mathbf{r}' does not satisfy Hamilton's equation of motion.

One consequence is that the Nosé-Hoover (NH) algorithm is not area-preserving, in terms of the real variables. This failure to conserve phase-space volume (locally) is a property that the NH algorithm shares with other non-Hamiltonian algorithms. The problem has been analyzed in detail by Tuckerman et al. [267], who describe a general procedure to obtain correct ensemble averages for non-Hamiltonian systems. The approach of ref. [267] is briefly described in Appendix B. In the SI (section L.6.1.1), we also describe the practical implementation of the Nosé-Hoover algorithm.

Advantages and drawbacks of the Nosé thermostat

The method developed by Nosé is ingenious, but it is not intuitively obvious. As a consequence, it took a while for some of the problems with the method to be recognized. All known problems were fixed, but only at the cost of making the algorithm even less intuitively obvious.

Nevertheless, the NH algorithm is still widely used because it has a number of positive features:

1. The NH algorithm is a *mild* thermostat, meaning that it affects all particle motions a bit, but none of them strongly. Hence the dynamics generated by NH approaches Newtonian dynamics as the system-size grows.

2. The NH algorithm conserves *virtual* momentum and it conserves a Hamiltonian-like quantity. That is not quite the same as conserving real energy and momentum, but a) it allows us to test whether there is an error in our code and b) as $p' = p/s$, we obviously have conservation of both the total real and virtual momentum in the case where $\sum p' = 0$.

It is these attractive features that account for the popularity of the NH thermostat. However, the NH thermostat also has serious drawbacks: one was mentioned already, it is not area-preserving the phase space of the real coordinates and momenta. The other problem is that the NH-thermostat is not always ergodic. This problem is particularly severe when the thermostat is coupled to (almost) harmonic degrees of freedom, but it is true in general for a system with more than one conserved quantity. The problem is illustrated for an extreme case in Fig. 7.6.

Such non-ergodic behavior of the Nosé-Hoover algorithm was first observed by Hoover [257], but similar effects were also observed in the simulation of a realistic molecular model [268]. Tuckerman et al. [267] argued that the reason why the NH algorithm may fail to recover the canonical distribution is that the Nosé derivation assumed only conservation of energy, whereas, in fact, other quantities may be conserved. An obvious candidate is the conservation of momentum. But if the total momentum of the system is zero, conservation of momentum is not an issue, provided that we choose $L = d(N - 1)$. But that still does not solve the problem: Tuckerman et al. [267] suggest that in some cases (e.g., the one-dimensional harmonic oscillator), other, less obvious additional conservation laws may exist. SI (section L.6.1.1) illustrates that in the presence of such additional conservation laws, the algorithm does not generate the desired distribution.

To alleviate the non-ergodicity problems of the Nosé-Hoover thermostat, Martyna et al. [269] proposed a scheme in which the Nosé-Hoover thermostat is coupled to another thermostat or, if necessary, to a whole chain of thermostats. As shown in Appendix L.6.1.2 these chains take into account additional conservation laws. Martyna et al. [269] showed that this generalization of the original Nosé-Hoover method still generates a canonical distribution (provided that it is indeed ergodic): see SI (section L.6.1.2).

But all this wizardry with the NH thermostat and the subsequent development of Nosé-Hoover chains, whilst a bonanza for theoreticians, comes at the expense of complicating an approach that, whilst clever, is already non-intuitive. That does not bode well for the survival of the NH algorithm because, in particular for more complex systems, some skill is required in choosing the best way to achieve ergodic thermostating with a collection of Nosé-Hoover thermostats.

Moreover, there are now robust and simple global velocity-rescaling thermostats (see section 7.1.3), which do not suffer from the problems of the Nosé-Hoover algorithm, nor from those of the more *ad hoc* velocity-rescaling schemes [251,264]. Having said that: the NH (or NH-chain) algorithm, if used correctly, is powerful and is implemented in many program packages.

Multiple NH thermostats are often used in applications in cases where different degrees of freedom have to be maintained at different temperatures (for instance to make sure that the “cold” degrees of freedom adiabatically follow the hotter degrees of freedom).

7.1.2.2 Application

We illustrate some of the points discussed above in a Nosé-Hoover simulation of the Lennard-Jones fluid.

Example 15 (Use of Nosé-Hoover thermostat). As in Example 14, we start by showing that the Nosé-Hoover method reproduces the behavior of a system at constant NVT . In Fig. 7.3 we compare the velocity distribution generated by the Nosé-Hoover thermostat with the correct Maxwell-Boltzmann distribution for the same temperature (7.1.1). The figure illustrates that the velocity distribution indeed is independent of the value chosen for the coupling constant Q .

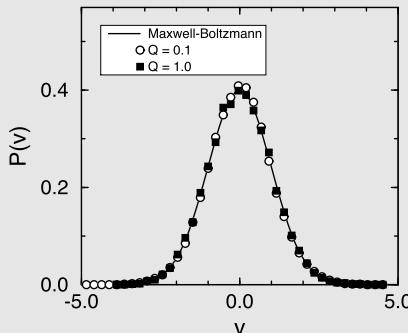


FIGURE 7.3 Velocity distribution in a Lennard-Jones fluid ($T = 1.0$, $\rho = 0.75$, and $N = 256$). The solid line is the Maxwell-Boltzmann distribution (7.1.1). The symbols were obtained in a simulation using the Nosé-Hoover thermostat [256].

It is instructive to see how the system reacts to a sudden increase in the imposed temperature. Fig. 7.4 shows the evolution of the kinetic temperature of the system. After 12,000 time steps the imposed temperature is suddenly increased from $T = 1$ to $T = 1.5$. The figure illustrates the role of the coupling constant Q . A small value of Q corresponds to a low inertia of the heat bath and leads to rapid temperature fluctuations. A large value of Q leads to a slow, ringing response to the temperature jump.

Next, we consider the effect of the Nosé-Hoover coupling constant Q on the diffusion coefficient. As can be seen in Fig. 7.5, the effect is much smaller than in Andersen’s method. However, it would be wrong to conclude that the diffusion coefficient is independent of Q . The Nosé-Hoover method simply provides a way to keep the temperature constant more gentle than Andersen’s method where particles suddenly get new, random velocities. For the calculations of transport properties, we prefer simple N, V, E simulations.

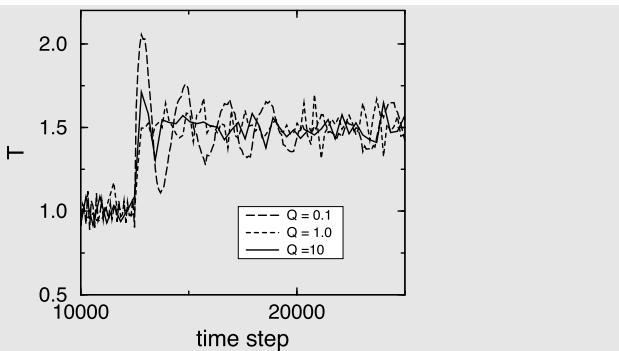


FIGURE 7.4 Response of the system to a sudden increase of the imposed temperature. The various lines show the actual temperature of the system (a Lennard-Jones fluid $\rho = 0.75$, and $N = 256$) as a function of the number of time steps for various values of the Nosé-Hoover coupling constant Q .

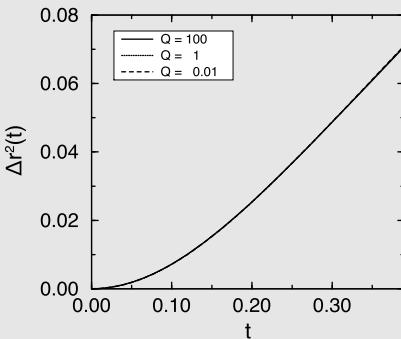


FIGURE 7.5 Effect of the coupling constant Q on the mean-squared displacement for a Lennard-Jones fluid ($T = 1.0$, $\rho = 0.75$, and $N = 256$).

For more details, see SI (Case Study 12).

In the preceding examples, we applied the Andersen and Nosé-Hoover thermostats to the Lennard-Jones fluid. We provided evidence suggesting that, for a system with total momentum equal to zero, the Nosé-Hoover thermostat generates a canonical distribution.

In the next example, we consider a particularly pathological case, namely, the one-dimensional harmonic oscillator, where the NH thermostat fails spectacularly.

Example 16 (Harmonic oscillator (I)). As the equations of motion of the harmonic oscillator can be solved analytically, this model system is often used to test algorithms. However, the harmonic oscillator is also a rather atypical

dynamical system, as will become clear when we apply the NH algorithm to this simple model system.

The potential energy function of the harmonic oscillator is

$$u(r) = \frac{1}{2}r^2.$$

The Newtonian equations of motion are

$$\begin{aligned}\dot{r} &= v \\ \dot{v} &= -r.\end{aligned}$$

If we solve the equations of motion of the harmonic oscillator for a given set of initial conditions, we can trace the trajectory of the system in phase space. Fig. 7.6 shows a closed loop, which a typical phase space trajectory of the periodic motion of the harmonic oscillator. It is straightforward to simulate a harmonic oscillator at constant temperature using the Andersen thermostat (see section 7.1.1.1). A trajectory is shown in Fig. 7.6. In this case the trajectories are points that are not connected by lines. This is due to the stochastic collisions with the bath. In this example, we allowed the oscillator to interact with the heat bath at each time step. As a result, the phase space density is a collection of discrete points. The resulting velocity distribution is Gaussian by construction; also for the positions, we find a Gaussian distribution.

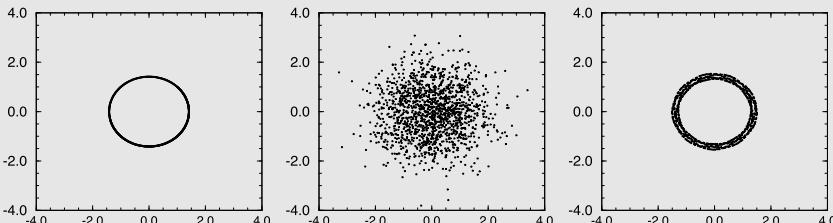


FIGURE 7.6 Trajectories of the harmonic oscillator: (from left to right) in the microcanonical ensemble, using the Andersen method, and using the Nosé-Hoover method. The y axis is the velocity and the x axis is the position.

We can also perform a constant-temperature Nosé-Hoover simulation using the algorithm described in SI L.6.2. A typical trajectory of the harmonic oscillator generated with the Nosé-Hoover scheme is shown in Fig. 7.6. The most striking feature of Fig. 7.6 is that, unlike the Andersen scheme, the Nosé-Hoover method does not yield a canonical distribution in phase space. Even for very long simulations, the entire trajectory would lie in the same ribbon shown in Fig. 7.6. Moreover, this band of trajectories depends on the initial configuration.

For more details, see SI (Case Study 13).

7.1.3 Stochastic global energy rescaling

The earliest MD thermostat maintained constant kinetic energy by instantaneous rescaling all particle velocities [264]. A more gradual version of the global rescaling algorithm was introduced in 1984 by Berendsen et al. [251].

The advantage of global velocity-rescaling thermostats is that they barely perturb the short-time dynamics of the system. However, as mentioned above (see [253]) the thermostats of refs. [251,264] may not equilibrate properly at long times, as they are not generating the correct canonical distribution.

However, in 1983, i.e., even before the Berendsen (or Nosé) thermostats had been proposed, Heyes [270] had proposed a velocity-rescaling scheme that could, at least in principle, have generated the correct canonical distribution (by Monte Carlo sampling of the total kinetic energy). Although the basic idea of ref. [270] was correct, the paper included one obvious (and trivial) mistake; it violated detailed balance in the kinetic-energy sampling. This mistake was corrected in ref. [271].

In 2007, Bussi et al. [256] proposed a purely dynamical velocity-rescaling thermostat where a Langevin equation is used to model the stochastic time evolution of the total kinetic energy, which generates the correct (Boltzmann) distribution of the kinetic energy. At each time step, the particle velocities are rescaled to approach the fluctuating kinetic energy.

The method of Bussi et al. [256] was essentially equivalent to the corrected approach of Heyes.⁵ The main difference between the schemes of refs. [270, 271] and [256] is how they carry out the stochastic update of the total kinetic energy: one uses MC, the other Langevin dynamics. But as the kinetic energy rescaling is unphysical anyway, the choice of the method would seem to be a matter of taste.

As the (corrected) method of Heyes, unlike the approach of Bussi et al., can be explained in one line, we use the MC approach (slightly reformulated) to explain the global velocity-rescaling approach, using the MC method. Both methods are non-deterministic, as they have a mild stochastic step in the kinetic energy rescaling. For a description of the Bussi algorithm, we refer the reader to the original paper [256].

In a practical implementation, the global velocity algorithm can be based on the standard velocity-Verlet algorithm by inserting the velocity-rescaling move in a symmetric, and thereby time-reversible way between the normal velocity-Verlet sub-steps that update the positions velocities of all particles. The bit that is new, is the update of the global kinetic energy.

As the thermostats of refs. [256,270,271] have no Galilean invariance, we should consider a system with a fixed center of mass. In that case, the equilib-

⁵ In fact, ref. [256] mentions the work of Heyes, but then dismisses it largely on aesthetic grounds, because the Heyes thermostat has no conserved Hamiltonian-like quantity, although it clearly would be easy to include such a diagnostic tool in the Heyes thermostat.

rium distribution of the total kinetic energy E_k at inverse temperature β is:

$$P(E_k) = \mathcal{C} E_k^{d(N-1)} \exp(-\beta E_k),$$

where \mathcal{C} is an (unimportant) normalization constant.

During a velocity rescaling move, we now attempt to rescale all velocities by a factor z , with $-\ln z$ in an interval $\{-\Delta_R, +\Delta_R\}$ (guaranteeing that the attempt frequency of forward and backward moves is equal). Δ_R is chosen such that we get a reasonable percentage of accepted trial moves.

The new kinetic energy is then $E'_k = z^2 E_k$. We then accept this trial move with a probability:

$$\text{acc}(E_k \rightarrow E'_k) = \min\left(1, z^{2d(N-1)} e^{-\beta E_k(z^2-1)}\right).$$

That is all.

As mentioned before, the Bussi thermostat is now widely used, and the basically equivalent Heyes thermostat is largely forgotten. Like the Nosé-Hoover thermostat, multiple Bussi thermostats can be coupled to different degrees of freedom (possibly at different temperatures). However, the use of multiple rescaling thermostats requires care, because, whilst a global velocity rescaling will not interfere with constraints (such as bond lengths), one should avoid using different rescaling factors to velocities of coordinates that are linked by constraints. Ceriotti et al. [272] have developed a generalization of the Bussi method to deal with situations where different degrees of freedom must be maintained at different temperatures.

7.1.4 Choose your thermostat carefully

As discussed in the previous sections, different thermostats have different strengths and weaknesses. In this context, we mentioned that several thermostats do not conserve momentum, which makes them useless for modeling hydrodynamic flows. Moreover, some versions of the NH thermostat may suffer from a lack of ergodicity.

But there is one more thermostating problem that deserves attention: typically, the effect of thermostats is to ensure that the kinetic energies of the particles in a system are Boltzmann-distributed. But that is not always what we want: for instance if a fluid is flowing, the velocity associated with the flow is not related to the temperature. Rather, what we should keep constant is the velocity distribution of the particles in a co-moving frame (such velocities are called *peculiar velocities*). If a system is flowing with a uniform velocity, it is usually easy enough to change the thermostat such that it does not affect the flow velocity. However, the situation becomes tricky when the hydrodynamic flows in a system are not uniform. Naive application of a global thermostat would then affect the local flow velocities. This problem can be mitigated by defining the

peculiar velocities locally, but that too has its problems: if we were to compute the local flow velocities in the cells of a fine grid, then there clearly is a problem if a cell contains $\mathcal{O}(1)$ particles, because then it becomes difficult to disentangle, on the fly, the local flow velocity and peculiar velocity.

In the case of inhomogeneous hydrodynamic flow, it is best to use a local, momentum conserving thermostat that controls the kinetic energy distribution of the relative motion of neighboring particles. Thermostats that do this are, for instance, the DPD thermostat (see Chapter 16) and the Lowe-Andersen thermostat (section 7.1.1.2).

7.2 Molecular Dynamics at constant pressure

In this Chapter, we have not followed the historical order of algorithmic developments. It is therefore important to repeat that, although discussed last in this chapter, the 1980 paper by Andersen [180] was the first to use extended Lagrangians to carry out Molecular Dynamics in other ensembles than *NVE*.

To perform an MD simulation of a system at constant pressure, Andersen chose to treat the system's volume as an additional dynamical variable, with its associated momentum.

Andersen's Lagrangian (in 3d) had the following form⁶:

$$\mathcal{L} = \frac{m}{2} Q^{2/d} \sum_{i=1}^N \dot{s}_i \cdot \dot{s}_i - \mathcal{U}(s^N; Q) + \frac{M}{2} \dot{Q}^2 - \alpha Q, \quad (7.2.1)$$

where, as before, s^N denoted the scaled particle coordinates.⁷

The momentum conjugate to s_i is $\pi_i \equiv \partial \mathcal{L} / \partial \dot{s}_i$. Solving the resulting Hamiltonian equation of motion generates trajectories whose time averages correspond to those of an isobaric-isenthalpic (*NPH*) ensemble. Note that although Andersen used an extended Lagrangian, there is no time-rescaling problem and, importantly, the transformation from the scaled variables to the real ones, is canonical. Hence, the Andersen algorithm is area-preserving in both scaled and original phase-space coordinates.

In terms of the real coordinates, and the real volume of the system, Andersen's equation of motion follow from the Lagrangian (Eq. (7.2.1)) and the relations $\mathbf{r}_i = Q^{1/d} s_i$, $\mathbf{p}_i = Q^{-1/d} \boldsymbol{\pi}_i$, $V = Q$ and $P = \alpha$:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m} + (1/d) \mathbf{r}_i \frac{d \ln V}{dt} \quad (7.2.2)$$

⁶ We do not follow Andersen's notation.

⁷ In practice, Q has the same numerical value as the volume V . However, in the equations of motion, it does not play quite the same role: changing V at constant s^N would move all particles and therefore contribute to the kinetic energy. But Q is introduced in the Lagrangian in such a way that the real velocity of a particle is $Q^{1/d} \dot{s}_i$. Hence, the kinetic energy of the particles does not depend on \dot{Q} . To obtain the correct ensemble average for a system at constant N , P , H (the isobaric-isenthalpic ensemble), the variable α must be chosen equal to the imposed pressure P .

$$\dot{\mathbf{p}}_i = - \sum_{j \neq i}^N \hat{\mathbf{r}}_{ij} u'(r_{ij}) - (1/d) \mathbf{p}_i \frac{d \ln V}{dt}$$

$$M \frac{d^2 V}{dt^2} = -P + \left(\frac{2}{d} \sum_{i=1}^N \mathbf{p}_i \cdot \mathbf{p}_i - \frac{1}{d} \sum_{i < j}^N r_{ij} u'(r_{ij}) \right) / V,$$

where we have assumed that the intermolecular interactions are pair-wise additive.

As the constant NPH ensemble is a bit inconvenient, Andersen then introduced his stochastic velocity-changing thermostat (section 7.1.1.1) to generate an algorithm that generates the correct NPT ensemble averages.

Subsequently, the Andersen method was recast in a form using global-velocity rescaling thermostats, as these are less disruptive to the dynamics of the molecules: initially, these approaches were based on the Nosé-Hoover approach, but more modern versions use Bussi's stochastic velocity-rescaling approach [273]. For details, we refer the reader to Ref. [273].

Finally, we note that, whilst the Andersen approach assumed that the shape of the simulation box was fixed, Parrinello and Rahman [178,179] extended the Andersen method to iso-tension MD simulations, where both the size and the shape of the simulation box can change. The Parrinello-Rahman technique is widely used to study solid-solid phase transitions, as in these transitions, the shape of the crystal unit cell (and therefore of the simulation box) may change. We discuss the Monte Carlo equivalent of the Parrinello-Rahman method in Section 6.4.

7.3 Questions and exercises

Question 22 (Andersen thermostat).

1. Explain why the static properties calculated using Andersen's constant- NVT MD are independent of the rate v at which velocities are thermalized?
2. Why does the diffusivity decrease with increasing v ?

Exercise 15 (Barrier crossing (Part 1)). Consider the movement of a single particle moving on a 1D potential energy surface of the following functional form:

$$U(x) = \begin{cases} \epsilon B x^2 & x < 0 \\ \epsilon (1 - \cos(2\pi x)) & 0 \leq x \leq 1 \\ \epsilon B (x - 1)^2 & x > 1 \end{cases}$$

The energy, force, and the derivative of the force are continuous functions of the position x and $\epsilon > 0$.

1. Derive an expression for B . Make a sketch of the potential energy landscape.
2. You can find programs on the book's website that can be used to compute the trajectory of the particle starting at $x(t=0) = 0$ using several methods:

- a. No thermostat (*NVE* ensemble). What do you expect the phase space trajectories to look like?
- b. Andersen thermostat. In this method, the velocity of the particle is coupled to a stochastic heat bath, which leads to a canonical distribution.
- c. A simple Monte Carlo scheme.

The Andersen thermostat and the *NVE* integration algorithm are not implemented yet in the programs that can be found in the SI. Try using all methods for a low temperature, $T = 0.05$, for which the system behaves like a harmonic oscillator. Pay special attention to the following:

- a. Why does the distribution generated at low temperatures with the MC scheme look so different from the ones generated with some of the MD schemes?
- b. Why does the phase space distribution of the *NVE* scheme look like a circle (or ellipse)?
- 3. Determine the approximate temperature at which the probability that the particle crosses the energy barrier becomes non-negligible.
- 4. Modify the potential-energy function in the program to

$$U = \epsilon [1 - \cos(2\pi x)] . \quad (7.3.1)$$

Calculate the diffusion coefficient as a function of the temperature. In standard constant-*NVE* MD, system behaves non-diffusively. Discuss the time-dependence of the mean-squared displacement of the particle as a function of the energy E .

Chapter 8

Free-energy calculations

8.1 Introduction

In the preceding chapters, we described the basics of Monte Carlo and Molecular Dynamics simulations. There exist many advanced techniques to tailor MC or MD simulations to specific types of models or conditions, and we will discuss a number of these techniques in subsequent chapters. However, there is one class of simulations that has to be discussed separately, namely *Free-Energy calculations*. As we explain below, these calculations are fundamentally different in spirit from normal MC or MD sampling.

8.1.1 Importance sampling may miss important states

To see why this is so, let us consider a Monte Carlo simulation. As was explained in Chapter 3, Markov-Chain Monte Carlo simulations sample the states of a system with a frequency that is proportional to their Boltzmann weight. For the computation of most equilibrium properties, only states with non-negligible Boltzmann weights are important, and hence the sampling generated in a Markov Chain Monte Carlo simulation is often called “importance sampling”. However, sometimes the term “importance” is misleading, for instance, if states with a very low Boltzmann weight are crucial for the time evolution of the system, as is typically the case for states at the top of an activation barrier. Hence, “importance sampling” may not always sample states that are important for the behavior of the system. Another example where normal MC or MD sampling often fail is in the study of first-order phase transitions. If a model system is prepared in a metastable phase, it usually will not transform into a more stable phase on typical simulation timescales. As such, this is nothing special: even in experiments, hysteresis in phase transitions is common; examples are undercooling, superheating, and, in particular, (non-displacive) solid-solid transitions. In order to use simulations to predict the relative stability of phases, and determine the location of the coexistence curve between these phases, we must circumvent the problem posed by hysteresis.

Locating a phase transition or computing the probability of finding the system at the top of an activation barrier, both have one feature in common: the need to compute the relative probability of finding a system in two distinct parts in phase space under conditions where normal MC or MD simulation would only explore one of the two parts. This happens when the other region in phase

space is unlikely to be visited, or when the other region, although large (possibly even larger than the original part) is in practice unreachable within a reasonable simulation time. So in both cases, brute-force MC or MD sampling fails.

8.1.2 Why is free energy special?

To clarify the nature of the underlying problem, it is useful to refer back to Fig. 3.1 in Chapter 3 when we used the measurement of the average depth of the Nile by performing a random walk in that river, as a naive example of importance sampling. The key point to note is that such a random walk is not a good strategy to sample the surface area of the Nile, which is our equivalent of the volume in phase space. Nor is a random walk useful if we want to determine the ratio of the areas of two rivers by comparing the relative time we spend in each.

Visiting improbable states is even more challenging: if I would like to know the relative probability of finding a single water-soluble tracer molecule in Lake Victoria or in my kitchen sink, I would have to wait for a long time to acquire the necessary statistics —yet this probability ratio is still less extreme than the typical probability ratios that play a role in, say, ice nucleation.

Illustration 8 (Molecular Dynamics versus Monte Carlo). Fig. 8.1 shows a schematic drawing of nanoporous crystalline material consisting of cavities connected by narrow “windows”. In a normal MD/MC simulation, the windows are barely sampled. This sampling problem is illustrated in Fig. 8.1(b)

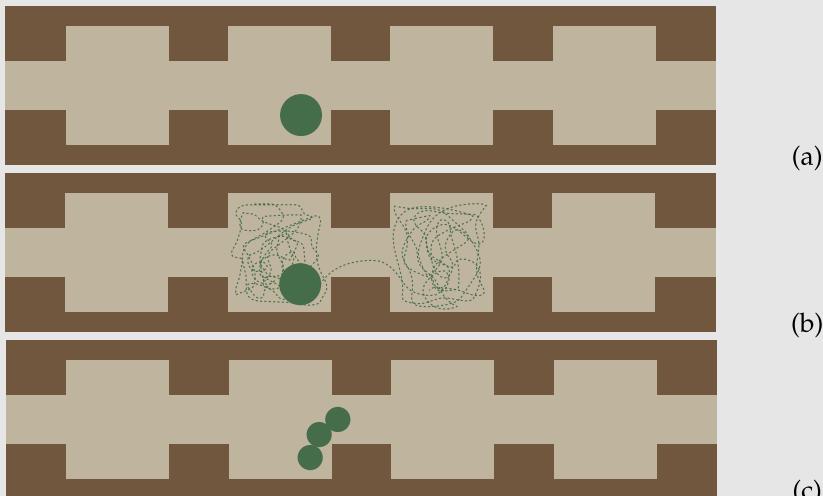


FIGURE 8.1 A model porous crystal consisting of a one-dimensional channel of cavities and windows. In (b) a typical molecular dynamics trajectory is shown. In (c) a three-atomic molecule in the pore. For this system, we would like to know the free energy of our molecule when it moves through the pore.

which shows a typical MD trajectory of a molecule in this pore: most of the time the molecule is trapped in one of the cavities and in the example shown in Fig. 8.1(b) it jumped only once to a neighboring cavity during the simulation. For larger molecules, the probability of finding a molecule in the window region can become too small to sample using brute-force simulation. Free-energy calculations offer the possibility to compute the probability to find the molecule in a window without having to wait for the event to happen spontaneously. The system shown in Fig. 8.1 is sufficiently simple that we can compute the free energy barrier for a single atom analytically. However, if there are more atoms, or more complex molecules in the pore, we need numerical free-energy calculations to compute the probability of finding a particle in a window.

Let us now consider free-energy calculations from a simulation point of view. As an example, consider a system at constant N , V , and T that can be found in two different domains in phase space: for instance, domain I corresponds to the crystalline state, and domain II to the liquid state. If the Boltzmann sampling of the states of the system is ergodic, then the ratios of the probabilities of finding the system in domains I and II would be:

$$\frac{P_I}{P_{II}} = \frac{Q_I(N, V, T)}{Q_{II}(N, V, T)} = e^{-\beta[F_I(N, V, T) - F_{II}(N, V, T)]}, \quad (8.1.1)$$

where Q_I denotes the partition function of all states in domain I , and similarly for II . As the probability ratio is equal to the ratio of the respective partition functions, this ratio can be expressed in terms of the Free-Energy difference $F_I - F_{II}$ between domains I and II . Eq. (8.1.1) shows that the relative probability of finding a system in two macro-states can be calculated if we can compute the free-energy difference between those two domains, even if the equilibrium probability to observe the system in one macro-state is much less for one system than for the other. Importantly, computing probability ratios by brute force becomes prohibitively expensive if the ratios are extreme, but computing free-energy differences does not suffer from this disadvantage. This, in a few words, is why free energy calculations are important and why they are usually preferred over brute-force Boltzmann sampling.

If we wish to compute the relative free energy of a larger number (say n) distinct domains, it would be inadvisable (though not wrong) to compute $n(n - 1)/2$ free energy differences. Rather, we compute the free-energy differences of all n macro-states with respect to a single reference state for which the free energy is known, either analytically or from earlier simulations. Such calculations are often called “absolute” free-energy calculations. However, they are nothing of the sort: a shift in all reference macro-states (e.g., by including the energy corresponding to the rest mass of the atoms in the free energy), will make no difference at all for any observable property. More importantly, the observable properties of classical many-body systems cannot depend on Planck’s

constant, even though this constant appears in the expression for the absolute free energy. Similarly, the phase behavior and static equilibrium properties of classical systems cannot depend on the masses m_i of the constituent particles. Any classical simulation result that appears to depend on h or in the case of Monte Carlo also on m_i is therefore wrong.¹

Order parameters and reaction coordinates

To carry out free-energy calculations, we need to be able to tell whether a system is in a given domain. In practice, that means that we must be able to compute a function of the particle coordinates, often referred to as an *order-parameter*. Different domains should then be distinguished by the value of this order parameter.

We may even have a continuous sequence of such order parameters (then often called *reaction coordinate*), for instance, if we consider a coordinate that measures the progress of the transformation from one domain (“reactants”) to another (“products”). If there are several such coordinates, we can even construct a higher-dimensional map of the free energy. However, and this is a point to note, free-energy landscapes can *only* be defined after such order parameters have been specified. There is no such thing as *the* free-energy landscape.

When do we need free-energy calculations?

Summarizing, there are two classes of problems for which it is advisable to use free-energy calculations. The first class involves computing the relative stability of two (or more) phases of a macroscopic system in cases where brute-force MC or MD sampling fails. Such failures typically happen when, as is the case for most first-order phase transitions, the rate of spontaneous transitions between the two phases is so low that we cannot accumulate sufficient statistics on the relative probabilities of finding the system in these phases.

We stress that, from a computational point of view, continuous phase transitions and, in particular, critical phenomena are no less challenging than first-order phase transitions. However, the technical problems are not free-energy related. Many excellent reviews of the numerical analysis of continuous phase transitions can be found in the literature (see e.g., [25,41–43,274]).

The second class of problems is different: it arises when we need to compute the free-energy barrier separating two (meta)stable basins of a system. Such calculations are important because knowledge of the height and shape of the free-energy barrier separating two basins allows us to estimate how frequently a system in (quasi)equilibrium would undergo transitions from one region to another, even if the transition rate is unobservably small. Tools for estimating

¹ In some program packages the user is protected against such mistakes as the code sets the thermal de Broglie wavelength Λ equal to one. Of course, it is not —after all, it is a length, and it is not the unit of length used in normal simulations. Λ is set to one for all species in a system, even if they have different masses, which is clearly impossible. Setting $\Lambda = 1$ protects us against mistakes, but may also make it harder to find mistakes.

the rates of such rare events will be discussed in Chapter 15; here, we just focus on the free-energy aspects.

8.2 General note on free energies

Non-standard simulation techniques are required to compute free-energy differences, and the development of such techniques started only a few years after the introduction of MC and MD simulations [275–277]. However, in recent years the calculation of free energies has become something of a cottage industry. By now, there are scores of distinct, but often not very different, techniques to compute free-energy differences, and attempting to give a summary of all these different schemes would be confusing —a detailed discussion would be exhausting. However, as this is a textbook and not a review, we have chosen to follow a different path: we will discuss *classes* of algorithms and explain them, using as our examples those approaches that we find most convenient for *didactical* purposes. These are not necessarily the best, nor the most popular,² and certainly not the fastest methods available. We will occasionally refer to more sophisticated approaches, but again our list is incomplete and will become more so as time progresses.

8.3 Free energies and first-order phase transitions

We now consider free-energy calculations in the context of first-order phase transitions, but as will be explained below, our discussion will be more general. Let us first consider what we mean when saying that at some temperature and pressure two phases (*I* and *II*) coexist. The answer to this question is that on one side of the coexistence curve, the probability of observing phase *I* approaches 100% in the thermodynamic limit, and on the other side, the probability of observing phase *II* approaches 100%. The reason for phrasing the question and answer like this is to stress that a 1-st order phase transition is the point where the *probability* of observing phase *I* in equilibrium changes from effectively 100% to 0%.³ Now let us look at the partition function. Let us assume we have an order parameter \mathcal{Q} that distinguishes phases *I* and *II*, such that $\mathcal{Q} < 0$ if the system is in *I*, and $\mathcal{Q} > 0$ otherwise. Then we can rewrite Eq. (8.1.1) as

$$\frac{p_I}{p_{II}} = \frac{\int d\mathcal{X} \exp[-\beta\mathcal{H}(\mathcal{X})]\theta(-\mathcal{Q})}{\int d\mathcal{X} \exp[-\beta\mathcal{H}(\mathcal{X})]\theta(+\mathcal{Q})} = \frac{Q_I(NVT)}{Q_{II}(NVT)}, \quad (8.3.1)$$

where we use \mathcal{X} as a shorthand for denoting the spatial coordinates \mathbf{r}^N of the system, and where we have omitted the identical factors $(\Lambda^{3N} N!)^{-1}$ in numerator and denominator. Note that Eq. (8.3.1) applies to a system at constant

² On the whole, MD implementations are much more popular than MC versions, simply because there are more widely available MD codes.

³ In small systems, the change in probability is more gradual.

volume, whereas coexistence curves delimit the transition between two phases at the same temperatures and pressures. However, if we define

$$Q_I(N, P, T) \equiv \frac{\beta P}{\Lambda^{3N} N!} \int dV \exp(-\beta PV) \int d\chi \exp[-\beta U(\chi)] \theta(-Q),$$

and similarly for $Q_{II}(N, P, T)$, then

$$\left(\frac{P_I}{P_{II}} \right)_{NPT} = \frac{Q_I(N, P, T)}{Q_{II}(N, P, T)} = e^{-\beta(G_I - G_{II})}. \quad (8.3.2)$$

Hence, the condition for phase coexistence of a one-component system at constant NPT is $G_I = G_{II}$, or equivalently $\mu_I = \mu_{II}$,⁴ which was of course well known, but now we have given a statistical mechanical expression for the Gibbs free energy $G_{I,(II)}(N, P, T)$ in terms of the order parameter that distinguishes the two phases.

Note that Eq. (8.3.2) is not limited to transitions between two phases: it allows us to describe the relative probability of any two domains in the phase space of a classical many-body system. We will return to this point in section 8.4.2.

In what follows, we briefly review some of the more commonly used techniques to compute free energies. We refer to several excellent reviews on this topic for more details [278–282].

8.3.1 Cases where free-energy calculations are not needed

8.3.1.1 Direct coexistence calculations

Before discussing free-energy calculations to locate first-order phase transitions, we should look at the obvious alternative: simply run a long simulation and try to identify the point where the two phases are in equilibrium.

Conceptually the simplest approach is to prepare the system in one phase and change the conditions (temperature, pressure) until it transforms spontaneously into a more stable phase. This method was explored in the early days of simulation [283–289]. However, in general, it suffers from a serious drawback: first-order phase transitions tend to exhibit appreciable hysteresis. Hence, the transformation to a new, more stable phase, if it happens at all on the time scale of a simulation, will typically only proceed well beyond the coexistence point, and then irreversibly. The reason why hysteresis is common in first-order phase transitions is that a large free energy barrier separates the two phases at, or near, coexistence. The height of this barrier is determined by the interfacial free energy of the interface separating the two coexisting phases. The larger the area of this interface, the higher the free energy barrier. In 1974, Streett et al. [290] proposed to estimate the true coexistence point by combining data

⁴ For multi-component systems, coexistence requires the equality of the chemical potentials $\mu_I^\alpha = \mu_{II}^\alpha$ for all species α .

from the metastable branches of the equation of state of both phases, and then using a Maxwell equal-area construction to locate the coexistence point. There is no *a priori* reason why such an approach should work, and indeed the method was shown to be not very accurate. A closely related was proposed in the early 2000s (a description can be found in [291]) and seems to be still popular, despite evidence presented in ref. [292] that the method does not compare well with free-energy calculations.

A better approach is to prepare a system containing two (periodically repeated) slabs of the target phases in the two phase-region, and then let the system reach equilibrium. For instance, one can prepare a system at constant N , V , and T , and at a density such that the two phases take up about the same volume. In general, this system will initially not be at the coexistence and, consequently, the interface will move until the pressures, and the chemical potentials of the two phases are the same. When this happens, the bulk densities of the two phases are the coexistence densities.

This sounds simple, and with modern computing power, it can be a viable option, in particular for fluid-fluid phase equilibria. An advantage is that a single simulation can yield the densities of the coexisting phases *and* the surface tension γ (see 5.1.6).

Nevertheless, there are a number of practical reasons why the method should be applied with care. First of all, the simulations require large-enough systems. The reason is that if the slabs of either phase are too thin, it may not be possible to identify a bulk region where the density can be measured. This problem gets more serious as we approach a possible critical point. Secondly, thin slabs may not be thermodynamically stable: to create the slabs with cross-section, S , we have to expend a surface free energy $2\gamma S$, where γ is the interfacial free-energy density —for liquids, this is the surface tension. For thin slabs, this free-energy cost may be larger than the free-energy cost that would result if the system were kept in a single, homogeneous phase, even though, in the bulk, that phase would not be thermodynamically stable. If at least one of the coexisting phases is a crystalline solid, we have the additional problem that the pressure in the bulk of the solid should be kept isotropic. This will require careful fine-tuning of the dimensions of the system in the plane of the slab. If both phases are crystals, it is usually impossible to accommodate them in the same simulation box because, typically, a box that accommodates one crystal form cannot accommodate the other. And even if this problem can be overcome, crystal interfaces tend to move slowly. Hence equilibration is a real issue. Still, for liquid-crystal transitions, the method can work: Hafskjold et al. [80] have shown that a solid-liquid coexistence curve can be traced out by considering a system in a temperature gradient.

In summary, the above observations imply that direct coexistence simulations tend to be relatively straightforward for fluid-fluid coexistence, harder for fluid-solid,⁵ and unattractive for first-order phase transitions between two solids.

⁵ If the solid-liquid interface is below the roughening transition, the addition/removal of a crystal plane involves a nucleation event, which may be rare. This problem can be partially circumvented

8.3.1.2 Coexistence without interfaces

In addition to the above methods, which simulate systems where two (or more) phases coexist in one simulation box, there exist several schemes to study phase coexistence without creating an interface. In fluids, the best-known method is the Gibbs-ensemble method of Panagiotopoulos [211,213,214,220], which is discussed in section 6.6. For what follows, it is important to point out the main limitation of the Gibbs-ensemble method: for equilibration, it relies on the exchange of volume and particles between two periodically repeated systems. However, if at least one of the two phases is dense (e.g., at a typical liquid density), the acceptance rate of the exchange of particles becomes negligible, and the Gibbs-ensemble method will fail. This problem is effectively insurmountable when one of the coexisting phases is a solid.⁶

For the case of a phase transition between two crystal phases, Parrinello and Rahman [178,179] designed a Molecular Dynamics scheme specifically for studying solid-solid transitions (see sections 6.4 and 7.2). This technique can be applied to *displacive* phase transitions that cause the crystal unit cell to deform without much rearrangement of the molecules within the unit cell. Even if these conditions are met, the Parrinello-Rahman method may suffer from some hysteresis. More importantly, when the two solids have very different unit cells then the Parrinello-Rahman method cannot be used.

8.3.1.3 Tracing coexistence curves

Even when we need to perform free-energy calculations to locate a first-order phase transition, we typically need only a few such calculations to trace a coexistence curve. The reason is that once we know the free energy of a given phase at one state point, we can use the thermodynamic relations between pressure, temperature, and chemical potential to compute the free energies at other state points that can be reached without crossing a phase transition.

A numerical technique for determining a complete coexistence curve, once a single point on the curve is known, was proposed by Kofke [293,294]. In its simplest form, Kofke's *Gibbs-Duhem integration* method is equivalent to the numerical integration of the Clausius-Clapeyron equation. Let us briefly recall the derivation of the Clausius-Clapeyron equation. When two phases α and β coexist at a given temperature T and pressure P , their chemical potentials must be equal. If we change both the pressure and the temperature by infinitesimal

by using helical boundary conditions —i.e., periodic boundary conditions where successive periodic boxes in (say) the x -direction are offset by lattice spacing in the direction perpendicular to the original crystal face (say z). In the y direction, an offset of one lattice spacing in the x direction should be imposed. Using such helical boundary conditions should result in the creation of step edges (x) and kinks y that cannot disappear as long as the crystal structure and alignment do not change. Kink sites are ideal for crystal growth/dissolution.

⁶ If a solid contains many vacancies, the Gibbs-ensemble method might still seem to work. However, in that case, another problem occurs: in a solid with many vacancies, the number of particles may differ substantially from the number of lattice sites. If this effect is ignored, the simulations will still give an answer, but it will be wrong (see. [224]).

amounts dP and dT , respectively, then the difference in chemical potential of the two phases becomes

$$d\mu_\alpha - d\mu_\beta = -(s_\alpha - s_\beta)dT + (v_\alpha - v_\beta)dP. \quad (8.3.3)$$

Along the coexistence curve $\mu_\alpha = \mu_\beta$, and hence

$$\frac{dP}{dT} = \frac{s_\alpha - s_\beta}{v_\alpha - v_\beta} = \frac{\Delta h}{T \Delta v}, \quad (8.3.4)$$

where we have used the fact that, at coexistence, $T \Delta s = \Delta h$, where h_α (h_β) denotes the molar enthalpy of phase α (β). As Δh , T , and Δv all can be computed directly in a simulation, dP/dT can be computed from Eq. (8.3.4). If one of the two coexisting phases is the (dilute) vapor phase, it is convenient to cast Eq. (8.3.4) in a slightly different form:

$$\frac{d \ln P}{d 1/T} = -\frac{\Delta h}{P \Delta v / T}. \quad (8.3.5)$$

Kofke and co-workers applied this method to locate the vapor-liquid [293,294] and solid-liquid coexistence curves of the Lennard-Jones fluid [295]. Other applications of the Kofke method can be found in [296–299]. It should be stressed that Gibbs-Duhem integration is not limited to the computation of coexistence curves in the P , T plane. A particularly important class of problems that can be treated in an analogous fashion is that where one studies the location of a phase transition as a function of a parameter in the intermolecular interaction potential. For instance, Agrawal and Kofke have investigated the effect of changing the steepness of the intermolecular potential in atomic systems on the melting point (see Illustration 9). Other examples of calculations that use generalizations of the Clausius-Clapeyron equation to study how a change in the Hamiltonian of a system affects the location of a first-order phase transition, can be found in refs. [300–303].

Although Gibbs-Duhem integration is potentially an efficient technique for tracing a coexistence curve, it is not necessarily robust, as it lacks built-in diagnostics. By this, we mean that the propagation of numerical errors in the integration of Eq. (8.3.4) may result in large deviations of the computed coexistence points from the true coexistence curve. Similarly, any error in the location of the initial coexistence points will lead to an incorrect estimate of the coexistence curve. For this reason, it is important to check the numerical stability of the scheme. This can be achieved by performing additional free energy calculations to fix two or more points where the two phases are in equilibrium (see e.g., [304]). The stability of the integration procedure can be checked by integrating backward and forward in the same interval. A discussion of various integration schemes to trace the coexistence curve can be found in ref. [299].

In some cases, for example for systems containing long-chain molecules, percolating systems, or lattice models it is difficult to perform changes in the

volume. Escobedo and de Pablo [305] have shown that under those conditions it may be preferable to combine Gibbs-Duhem integration with the grand-canonical (constant μ , V , T) ensemble. In this scheme, μ and T are the independent variables, rather than P and T . The variation in the pressure difference of phases α and β is given by

$$dP_\alpha - dP_\beta = (\rho_\alpha - \rho_\beta) d\mu + \left(\frac{s_\alpha}{v_\alpha} - \frac{s_\beta}{v_\beta} \right) dT. \quad (8.3.6)$$

Along coexistence, we have $P_\alpha = P_\beta$, which gives

$$\frac{d\beta\mu}{d\beta} = \frac{\rho_\alpha h_\alpha - \rho_\beta h_\beta}{\rho_\alpha - \rho_\beta} = \frac{\rho_\alpha u_\alpha - \rho_\beta u_\beta}{\rho_\alpha - \rho_\beta}. \quad (8.3.7)$$

Implementing this equation in a Gibbs-Duhem integration scheme implies that the volume changes of the constant pressure simulations are replaced by particle exchanges and removals. Escobedo [306] developed extensions of the Gibbs-Duhem integration technique for multicomponent fluid mixtures.

Illustration 9 (Freezing of soft spheres). The earliest simulations of freezing were performed by Alder and Wainwright [18] and Wood and Jacobson [19]. The exact location of this freezing transition was first determined by Hoover and Ree [307]. Subsequently, several authors studied the dependence of the freezing transition on the “softness” of the intermolecular potential. This is done most conveniently by considering a class of model systems of variable softness that contains the hard-sphere model as a limiting case. In this context, so-called soft-sphere models have been studied extensively. The soft-sphere model is characterized by a pair potential of the form

$$u(r) = \epsilon \left(\frac{\sigma}{r} \right)^n.$$

Limiting cases are the hard-sphere model ($n \rightarrow \infty$) and the one-component plasma ($n = 1$). Before the advent of the Gibbs-Duhem integration scheme, individual simulation studies had been performed to locate the freezing point for soft spheres with $n = 1$ [308], $n = 4, 6, 9$ [309,310], $n = 12$ [287,311–313], and $n = \infty$ [307,314]. Actually, the crystal structure at melting changes from **Face Centered Cubic** (FCC) (or possibly **Hexagonal Close Packed** (HCP)) for large n , to **Body Centered Cubic** (BCC) for small n . Hoover and Ree [307] have argued that the change from FCC to BCC takes place around $n = 6$. Agrawal and Kofke [295,315] showed that the Gibbs-Duhem integration technique can be used to locate the melting points of all soft-sphere models in one single simulation. The quantity that is changed in this Gibbs-Duhem integration is the softness parameter s , defined by $s \equiv 1/n$. We can interpret s as a thermodynamic variable, on the same footing as the pressure P and the temperature T . An infinitesimal variation in the thermodynamic variables T , P ,

and s results in a variation of the Gibbs free energy [G](#):

$$dG = -SdT + VdP + \frac{N\lambda}{\beta}ds,$$

where we have defined λ as the thermodynamic “force” conjugate to s (the factor N/β has been introduced to keep our notation consistent with that of [295,315]). We now consider phase coexistence at a constant temperature. If we vary both P and s , the difference in chemical potential of the two phases will change:

$$\beta\mu_\alpha - \beta\mu_\beta = \beta(v_\alpha - v_\beta)dP + (\lambda_\alpha - \lambda_\beta)ds,$$

where v_α (v_β) is the molar volume of phase α (β). Along the coexistence curve, $\mu_\alpha = \beta\mu_\beta$ and hence

$$\left(\frac{\partial \ln P}{\partial s}\right)_{\text{coex}} = -\frac{\Delta\lambda}{\beta P \Delta v}.$$

To use this equation in a simulation, we need the statistical mechanical expression for λ . The partition function of a system at constant pressure and temperature is given by Eq. (6.3.8)

$$\begin{aligned} Q(N, P, T) &= \frac{\beta P}{\Lambda^{3N} N!} \int dV \exp(-\beta PV) \int d\mathbf{r}^N \exp[-\beta \mathcal{U}(\mathbf{r}^N; s)] \\ &= \frac{\beta P}{\Lambda^{3N} N!} \int dV \exp(-\beta PV) \int d\mathbf{r}^N \prod_{i>j} \exp\left[-\beta\epsilon(\sigma/\mathbf{r}_{ij})^{1/s}\right]. \end{aligned}$$

The thermodynamic definition of λ is written

$$\lambda \equiv \left(\frac{\beta \partial G}{N \partial s}\right)_{T, P}.$$

Using $G = -k_B T \ln Q(N, P, T)$, we obtain

$$\begin{aligned} \lambda &= -\frac{\beta}{NQ(N, P, T)} \left[\frac{\partial Q(N, P, T)}{\partial s} \right]_{T, P} \\ &= -\frac{\beta\epsilon}{s^2} \left\langle \left(\frac{\sigma}{r}\right)^{1/s} \ln(\sigma/r) \right\rangle \\ &= -\frac{\beta\epsilon}{s^2} \langle u(r) \ln(\sigma/r) \rangle. \end{aligned}$$

The preceding expression is used to measure λ in the coexisting solid and liquid phases. This makes it possible to compute the melting curve in the (P, s) plane. Following this approach, Agrawal and Kofke were able to obtain the melting pressure of the soft-sphere model for all n between 1 and ∞ . They were also able to locate the fluid-FCC-BCC triple point at $s \approx 0.16$.

8.4 Methods to compute free energies

Let us now consider numerical methods to compute free energies. These methods can be used when direct-coexistence calculations are not possible. But even if there is a choice, free-energy calculations are often preferable because they tend to be more robust. Moreover, free energies can be computed with high precision, allowing us to locate transition points with higher accuracy than can be achieved with the same computational effort in typical direct-coexistence simulations.

There is, however, no denying that many users of molecular-simulation packages shy away from free-energy calculations because these are less intuitive than direct coexistence calculations. In the following sections, we aim to explain free-energy calculations and (hopefully) show that they greatly enhance the power of molecular simulations.

As was explained in section 8.2, we do not aim to provide an overview of all relevant free-energy-related algorithms. Rather, we will consider prototypical examples of methods that are being used and thereby make the broader literature accessible to the reader.

Below, we first focus on techniques for computing the free energy and chemical potential of individual macro-states, with a special emphasis on first-order phase transitions. After that, we discuss techniques to compute free-energy profiles or landscapes as a function of one or more order parameters.

8.4.1 Thermodynamic integration

As was explained in section 8.1.2, free-energy calculations are qualitatively different from simulations that compute Boltzmann-weighted averages of observables: Free energies are not averages over phase space, rather they measure the accessible volume in phase space itself. However —and this is a key point—the derivative of the free energy with respect to any of its control parameters *can be* expressed as a Boltzmann average. The problem of computing the free energy of a system can therefore be reduced to integrating its derivative with respect to any of its control parameters along a path between the state of interest and a state of known free energy. Typical control variables are the temperature or the volume of the system, but also any parameter that appears in the Hamiltonian, for instance, the strength or the range of the inter-particle interactions. Note that integrating a free-energy derivative along a path between two states can only yield a free energy *difference*. If we are interested in the absolute free energy of a system, we need to choose a reference for which we can determine the free energy analytically.

In what follows, we shall use the term “free energy” in a generic sense: we may refer to the Helmholtz free energy $F(N, V, T)$, the Gibbs free en-

ergy $G(N, P, T)$, or even the entropy $S(N, V, E)$.⁷ Below, we mainly focus on calculations of the Helmholtz free energy and comment on related free-energy calculations when necessary.

For a classical, atomic system the Helmholtz free energy F is related to the canonical partition function $Q(N, V, T)$ (Eq. (2.2.14)) by:

$$F = -k_B T \ln Q(N, V, T) \equiv -k_B T \ln \left(\frac{\int d\mathbf{p}^N d\mathbf{r}^N \exp[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N)]}{h^{dN} N!} \right),$$

where d is the dimensionality of the system. It is clear that $Q(N, V, T)$ is an integral over phase space, not a Boltzmann average. The same holds for the other free-energy-related quantities S and G . Such quantities cannot be sampled directly in either MD or MC simulations. We will use the adjective *thermal* to refer to quantities that depend directly on the available volume in phase space. In contrast, we use the adjective *mechanical* to refer to observables that can be expressed as a Boltzmann-weighted average over phase space of a function of $(\mathbf{p}^N, \mathbf{r}^N)$.

The fact that thermal quantities cannot be measured directly in a simulation is not a problem of simulations only: real experiments cannot measure thermal quantities either. When considering numerical schemes to determine the free energy it is therefore instructive to see how this problem is resolved in the real world. Experiments always probe a *derivative* of the free energy, such as the derivative with respect to volume V or temperature T :

$$\left(\frac{\partial F}{\partial V} \right)_{NT} = -P \quad (8.4.1)$$

and

$$\left(\frac{\partial F/T}{\partial 1/T} \right)_{VN} = E. \quad (8.4.2)$$

As the pressure P and the energy E are mechanical quantities, they can be measured in a simulation.⁸ To compute the free energy of a system at a given temperature and density, we should find a *reversible* path in the $V-T$ plane that links the state under consideration to a state of known free energy. The change in F along that path can then simply be evaluated by thermodynamic integration, that is, by integration of Eqs. (8.4.1) and (8.4.2). There are only a few thermodynamic states for which the free energy of a substance is known analytically. One such state is the ideal gas phase; the other is the low-temperature harmonic crystal.

⁷ The Grand Potential $-PV(\mu, V, T) = -k_B T \ln \Xi(\mu, V, T)$ is different: it can be computed directly, at least for homogeneous systems, but in that case, the problem is that in particular at high densities, simulations at constant μ , V and T converge slowly, if at all.

⁸ Be it that we have to specify our zero of energy. Again, the same holds for experiments.

In computer simulations, the situation is similar. To compute the free energy of a dense liquid, one may construct a reversible path to the very dilute gas phase. It is not necessary to go all the way to the ideal gas, but one should reach a state that is sufficiently dilute to ensure that the free energy can be computed accurately, either from a direct calculation of the chemical potential (see sections 8.5.1 and 8.5.2) or from knowledge of the first few terms in the virial expansion of the compressibility factor $PV/(Nk_B T)$, as explained below.

At low densities

$$\frac{P(\rho, T) - \rho k_B T}{\rho^2} \rightarrow B_2(T), \quad (8.4.3)$$

where B_2 denotes the second virial coefficient, which can be computed to any desired accuracy, using

$$B_2(T) = \frac{1}{2} \int d\mathbf{r} \left(1 - e^{-\beta u(r)} \right).$$

Once we know B_2 , we do not start from Eq. (8.4.1), but from an expression for the density-derivative of the excess free energy, $F_{\text{ex}}(\rho, T) \equiv F(\rho, T) - F_{\text{id}}(\rho, T)$:

$$\left(\frac{\partial F_{\text{ex}}(\rho, T)}{\partial \rho} \right)_{NT} = \frac{P(\rho, T) - \rho k_B T}{\rho^2}. \quad (8.4.4)$$

From Eq. (8.4.3), we know the limiting behavior of Eq. (8.4.4): it is $B_2 + \mathcal{O}(\rho)$, which is well behaved as $\rho \rightarrow 0$. The integral to compute $F_{\text{ex}}(\rho, T)$ at higher densities is therefore also well-behaved, provided that the integration path does not cross a first-order phase transition.

To compute the free energy of a crystalline solid, we do not use the ideal gas reference state, because the crystalline solid and the fluid phase are separated by a first-order phase transition, at least in three dimensions. Computing the free energy of solids is discussed in detail in Chapter 9.

We just note here that, once we know the free energy of the solid at low temperatures, where it behaves as a harmonic crystal, we can use Eq. (8.4.2) to evaluate the solid free energy at higher temperatures, using

$$\left(\frac{\partial (F - E_{\text{harmonic}})/T}{\partial T} \right)_{VN} = -(E - E_{\text{harmonic}})/T^2, \quad (8.4.5)$$

where E_{harmonic} is the energy of a harmonic solid at the same temperature as our model system. As $(E - E_{\text{harmonic}}) \sim T^2$ at low temperatures, Eq. (8.4.5) remains well behaved as $T \rightarrow 0$. Hence, computing the solid free energy, starting from a low-temperature harmonic crystal does not pose any special problems either, provided that the solid phase is mechanically stable down to $T = 0$.

When using thermodynamic integration to compute the free energy of a dense liquid, i.e., a fluid in the ρ, T -range between the critical point and the

triple point, it is best to integrate around the critical point. Such simulations start at low densities for $T > T_c$ and integrate Eq. (8.4.1) to the desired density, and subsequently integrate Eq. (8.4.2) to lower temperatures at constant ρ .

Note that the “natural” free energy techniques described above, will not work when it is not feasible to construct a natural reversible path to the state point of interest. In that case, other techniques are required, such as those described below and in Chapter 9.

8.4.2 Hamiltonian thermodynamic integration

In a simulation, we are not limited to using a physical thermodynamic integration path that can also be followed in experiments. Rather, as is clear from the discussion in the last paragraph of section 2.5.1, we can use all parameters in the potential energy function as if they were thermodynamic variables [316]. For example, if we know the free energy of the Lennard-Jones fluid, we can determine the free energy of the Stockmayer fluid⁹ by calculating the reversible work required to switch on the dipolar interactions in the Lennard-Jones fluid [317]. The formalism used to calculate this free energy difference is Kirkwood’s coupling-parameter method [60].

Let us consider the case where we wish to compute the free energy of an N -particle system with potential energy \mathcal{U}_{II} . Now assume that there is another potential energy function \mathcal{U}_{I} that is sufficiently simple that we can compute the free energy of the system analytically, e.g., an ideal gas, or a harmonic crystal. Alternatively, \mathcal{U}_{I} might be the potential energy function of a system for which the free energy is known accurately from earlier work, e.g., a fluid or solid of hard-spheres [318–322] or Lennard-Jones particles [73,83,323]. We now define a generalized potential energy function $\mathcal{U}(\lambda)$, such that $\mathcal{U}(\lambda = 1) = \mathcal{U}_{\text{II}}$ and $\mathcal{U}(\lambda = 0) = \mathcal{U}_{\text{I}}$. A simple choice for $\mathcal{U}(\lambda)$ is

$$\begin{aligned}\mathcal{U}(\lambda) &= (1 - \lambda)\mathcal{U}_{\text{I}} + \lambda\mathcal{U}_{\text{II}} \\ &= \mathcal{U}_{\text{I}} + \lambda(\mathcal{U}_{\text{II}} - \mathcal{U}_{\text{I}}).\end{aligned}\quad (8.4.6)$$

For example, system I might correspond to a Lennard-Jones fluid, while system II refers to the Stockmayer fluid.

The partition function for a system with a potential energy function $\mathcal{U}(\lambda)$ ($0 \leq \lambda \leq 1$) is

$$Q(N, V, T, \lambda) = \frac{1}{\Lambda^{3N} N!} \int d\mathbf{r}^N \exp[-\beta\mathcal{U}(\lambda)].$$

As explained above, $Q(N, V, T, \lambda)$ cannot be sampled in a simulation, but the derivative of the Helmholtz free energy $F(\lambda)$ with respect to λ can be written as

⁹ A Stockmayer fluid consists of Lennard-Jones particles with embedded point dipoles.

a Boltzmann-weighted average:

$$\begin{aligned}
 \left(\frac{\partial F(\lambda)}{\partial \lambda} \right)_{N,V,T} &= -\frac{1}{\beta} \frac{\partial}{\partial \lambda} \ln Q(N, V, T, \lambda) \\
 &= -\frac{1}{\beta Q(N, V, T, \lambda)} \frac{\partial Q(N, V, T, \lambda)}{\partial \lambda} \\
 &= \frac{\int d\mathbf{r}^N (\partial \mathcal{U}(\lambda) / \partial \lambda) \exp[-\beta \mathcal{U}(\lambda)]}{\int d\mathbf{r}^N \exp[-\beta \mathcal{U}(\lambda)]} \\
 &= \left\langle \frac{\partial \mathcal{U}(\lambda)}{\partial \lambda} \right\rangle_\lambda,
 \end{aligned} \tag{8.4.7}$$

where $\langle \dots \rangle_\lambda$ denotes a Boltzmann-weighted average for a system with a potential energy function $\mathcal{U}(\lambda)$ (8.4.6).

The free-energy difference between systems II and I can be obtained by integrating Eq. (8.4.7):

$$F(\lambda = 1) - F(\lambda = 0) = \int_{\lambda=0}^{\lambda=1} d\lambda \left\langle \frac{\partial \mathcal{U}(\lambda)}{\partial \lambda} \right\rangle_\lambda. \tag{8.4.8}$$

As in the case of natural Thermodynamic Integration (TI), there is one important constraint on the use of Eq. (8.4.8): along the path from $\lambda = 0$ to $\lambda = 1$, the system should not undergo irreversible changes, such as the nucleation of a new phase due to a first-order phases transition, because then the system would exhibit hysteresis, and the forward and backward integrals would yield different (but equally incorrect) answers.

In principle, we could perform the thermodynamic integration using any (in general, nonlinear) function $\mathcal{U}(\lambda)$, as long as this function is differentiable and satisfies the boundary condition: $\mathcal{U}(\lambda = 0) = \mathcal{U}_I$ and $\mathcal{U}(\lambda = 1) = \mathcal{U}_{II}$. However, the linear interpolation (8.4.6) is particularly convenient because in that case, we know the sign of $\partial^2 F / \partial \lambda^2$. Straightforward differentiation of Eq. (8.4.7) shows that

$$\left(\frac{\partial^2 F}{\partial \lambda^2} \right)_{N,V,T} = -\beta \left\{ \left\langle (\mathcal{U}_{II} - \mathcal{U}_I)^2 \right\rangle_\lambda - \langle \mathcal{U}_{II} - \mathcal{U}_I \rangle_\lambda^2 \right\} \leq 0.$$

In words, $(\partial F / \partial \lambda)$ can never increase with increasing λ . This *Gibbs-Bogoliubov* inequality can be used to test the validity or accuracy of the simulation results. In practice, the integration in Eq. (8.4.8) must be carried out numerically, for instance using Gaussian quadrature. Of course, such a numerical integration will work only if the integrand in Eq. (8.4.8) is a well-behaved function of λ . Occasionally, however, a linear parametrization of $\mathcal{U}(\lambda)$ may lead to a weak (and relatively harmless) singularity in Eq. (8.4.8) for $\lambda \rightarrow 0$. This point is discussed in more detail in section 9.2.2.

Hamiltonian thermodynamic integration is often used to compute the difference in excess free energy of similar but distinct molecules. Such calculations

are of particular importance in biomolecular modeling (see e.g., [324,325]). One can, for instance, compute the effect of a chemical substitution on the binding strength of a molecule to an enzyme. In such calculations, the thermodynamic integration involves a gradual replacement of part of the molecule by another building block; for instance, an H could be transformed into a CH₃ group.

It should be noted that the thermodynamic integration method based on Eq. (8.4.8) is intrinsically static; that is, the derivative of the free energy is obtained in a series of equilibrium MC or MD simulations. In section 8.7 we will discuss how free-energy differences can even be calculated if the Hamiltonian of the system is changed at a finite rate.

8.5 Chemical potentials

For systems at constant N , P , and T , equilibrium requires equality of the chemical potentials of all species in all phases. Moreover, in systems that can undergo chemical reactions, the condition for equilibrium requires that the (Gibbs) free energy does not change if an infinitesimal amount of reactants is converted into products. This condition imposes a linear relation between the chemical potentials of the reacting species in the system. Hence, to predict the conditions for equilibrium, we need to know the chemical potentials of reactants and products. For these reasons, an important aim of many free-energy calculations is to compute the chemical potential of the various molecules in the system. In the previous section, we discussed the calculation of the total free energy of a system. If the system contains only one type of molecule, computing the pressure and the free energy is enough to obtain the chemical potential, using $N\mu = F + PV$. However, for multi-component systems, the corresponding relation ($\sum_i N_i \mu_i = F + PV$) is not enough to determine the individual chemical potentials. In the present section, we discuss methods that have been developed specifically for computing chemical potentials, or chemical-potential differences. To keep the notation simple, we will start by considering one-component systems.

The chemical potential of a system can be written as (see section 2.1.3):

$$\mu = \left(\frac{\partial F}{\partial N} \right)_{V,T} = \left(\frac{\partial G}{\partial N} \right)_{P,T} = -T \left(\frac{\partial S}{\partial N} \right)_{V,E} .$$

Of course, the number of particles in a system is not a continuous variable, and hence, in simulations, we should define the chemical potential as the free-energy difference between systems containing $N + 1$ and N particles:

$$\mu \equiv \left(\frac{\partial F}{\partial N} \right)_{VT} \approx \frac{F(N+1) - F(N)}{N+1 - N} = F(N+1) - F(N) . \quad (8.5.1)$$

As the free energy difference in Eq. (8.5.1) is related to the logarithm of the ratio of two partition functions, μ can be expressed in terms of a quantity that can be

sampled. This may seem surprising, as we argued in the preceding section that “thermal” quantities, such as the Gibbs free energy, cannot be sampled directly. However, there is no contradiction: what we measure is not the absolute chemical potential but the excess chemical potential, that is, the *difference* between the chemical potential of a given species in a dense phase, and that of an ideal gas of the same species at the same density and temperature. This difference can be computed by Hamiltonian thermodynamic integration, and in fact, for dense systems, this is usually the only way to compute μ_{ex} . However, for less dense systems the thermodynamic integration can be carried out in a single step.

8.5.1 The particle insertion method

Eq. (8.5.1) is the starting point for deriving an expression for the chemical potential of a pure substance, but the expressions for a multi-component system are essentially the same. The expression that we derive below was presented by Widom in 1963 as a general theoretical tool in Statistical Mechanics.¹⁰ The application of Widom’s theoretical approach to simulations was explored by Romano and Singer [327]. The Widom expression can be derived in a few lines from the statistical-mechanical definition of the chemical potential μ_a of a species a . To keep the notation compact, we shall initially assume that we deal with a system of N identical atoms in a cubic volume V with diameter $L = V^{1/d}$, at constant temperature T .

The classical partition function of such a system is given by

$$Q(N, V, T) = \frac{V^N}{\Lambda^{dN} N!} \int_0^1 \cdots \int_0^1 d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)]. \quad (8.5.2)$$

The use of scaled coordinates $\mathbf{s}^N = \mathbf{r}^N / L$ is not essential but is convenient because it will make it easy to separate the ideal and excess parts of the chemical potential. In Eq. (8.5.2), we have written $\mathcal{U}(\mathbf{s}^N; L)$ to indicate that \mathcal{U} depends on the real rather than the scaled distances between the particles. The expression for the Helmholtz free energy of the system is

$$\begin{aligned} F(N, V, T) &= -k_B T \ln Q \\ &= -k_B T \ln \left(\frac{V^N}{\Lambda^{dN} N!} \right) - k_B T \ln \left\{ \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N; L)] \right\} \\ &= F_{\text{id}}(N, V, T) + F_{\text{ex}}(N, V, T). \end{aligned} \quad (8.5.3)$$

In the last line of this equation, we have identified the two contributions to the Helmholtz free energy on the previous line, as the ideal gas expression plus an excess part.

¹⁰ In 1962, Hoover and Poirier [326] derived a closely-related expression in the context of computing virial coefficients. However, they did not use as a “particle-insertion” method.

For sufficiently large N , the chemical potential is given by

$$\mu = -k_B T \ln(Q_{N+1}/Q_N).$$

If we use the explicit form, Eq. (8.5.3) for Q_N , we find that

$$\begin{aligned} \mu &= -k_B T \ln(Q_{N+1}/Q_N) \\ &= -k_B T \ln\left(\frac{V/\Lambda^d}{N+1}\right) - k_B T \ln\left\{\frac{\int d\mathbf{s}^{N+1} \exp[-\beta\mathcal{U}(\mathbf{s}^{N+1})]}{\int d\mathbf{s}^N \exp[-\beta\mathcal{U}(\mathbf{s}^N)]}\right\} \\ &\equiv \mu_{\text{id}}(\rho) + \mu_{\text{ex}}. \end{aligned} \quad (8.5.4)$$

In the last line of Eq. (8.5.4), we have separated the chemical potential in an ideal gas contribution μ_{id} and the excess part μ_{ex} . As $\mu_{\text{id}}(\rho)$ can be evaluated analytically, we focus on μ_{ex} . We now separate the potential energy of the $(N+1)$ -particle system into the potential energy function of the N -particle system, $\mathcal{U}(\mathbf{s}^N)$, and the interaction energy of the $(N+1)$ th particle with the rest: $\Delta\mathcal{U} \equiv \mathcal{U}(\mathbf{s}^{N+1}) - \mathcal{U}(\mathbf{s}^N)$. We stress that this separation does *not* assume pairwise additivity of the intermolecular interactions: it is just the *definition* of $\Delta\mathcal{U}$. Using this separation, we can write μ_{ex} as

$$\mu_{\text{ex}} = -k_B T \ln \int d\mathbf{s}_{N+1} \langle \exp(-\beta\Delta\mathcal{U}) \rangle_N = -k_B T \ln \langle \exp(-\beta\Delta\mathcal{U}) \rangle_N, \quad (8.5.5)$$

where $\langle \dots \rangle_N$ denotes Boltzmann averaging over the configuration space of the N -particle system at constant V and T , and \mathbf{s}_{N+1} is the scaled coordinate of particle number $N+1$.

It is important to note that the averaging in Eq. (8.5.5) cannot be computed using importance sampling of the coordinate \mathbf{s}_{N+1} . Rather, we compute the average of $\langle \exp(-\beta\Delta\mathcal{U}) \rangle_N$ by integrating *uniformly* over the scaled position of particle $N+1$. This integration can be performed by brute-force random sampling of the type that was pioneered by Metropolis and Ulam [328] before the Markov-Chain MC method had been developed. In practice the procedure is as follows: we carry out a conventional constant- NVT Monte Carlo simulation on the system of N particles. At frequent intervals during this simulation, we randomly generate a coordinate \mathbf{s}_{N+1} , uniformly over the unit cube. For this value of \mathbf{s}_{N+1} , we then compute $\exp(-\beta\Delta\mathcal{U})$. By averaging the latter quantity over all generated trial positions, we obtain the average that appears in Eq. (8.5.5). So, in effect, we are computing the average of the Boltzmann factor associated with the random insertion of an additional particle in an N -particle system, but we never accept any such trial insertions, because then we would no longer be sampling the average needed in Eq. (8.5.5). We note that for homogeneous systems, every trial-insertion point is equivalent. Hence, we could also carry out trial insertions at fixed positions in the simulation box (e.g., on a grid), as long as the insertion points are not correlated with the positions of the remaining

Algorithm 18 (Widom test particle insertion)

<pre>function Widom xtest=box*R entest = ener(xtest,npart+1) wtest=wtest + +exp(-beta*entest) end function</pre>	excess chemical potential through test-particle insertion generate random position determine energy update Boltzmann factor in (8.5.5)
--	--

Specific Comments (for general comments, see p. 7)

1. This algorithm shows the basic structure of the Widom test-particle method for systems at constant N , V and T . This function is usually called in the sampling step of a Monte Carlo simulation, for example, in function sample in Algorithm 1. Usually, many such test particle insertions are needed to obtain reliable statistics.
2. Because the test particle does not belong to the n_{part} particles in the system, we give it a (dummy) label $n_{\text{part}}+1$.
3. The excess chemical potential follows from $\beta\mu_{\text{ex}} = -\ln(w_{\text{test}}/M)$, where M is the total number of test particle insertions. The accuracy of $\beta\mu_{\text{ex}}$ can be estimated using $\sigma_{\beta\mu_{\text{ex}}} = \sigma_{w_{\text{test}}}/w_{\text{test}}$.
4. function ener calculates the energy of the test particle. Note that the test particle insertion is virtual and is never accepted.
5. For pairwise-additive interactions, we can approximately correct for the effect of the truncation of the intermolecular interactions on the value of the chemical potential by evaluating a tail correction. Because we do not have to correct for double counting, this correction is a factor of 2 larger than what is used for the potential energy per particle in a normal MC or MD simulations (see Case Study 15).

N particles. The Widom method provides us with a powerful scheme for computing the chemical potential of (not too dense) atomic and simple molecular liquids. In Algorithm 18, we show how the particle-insertion method can be implemented in a simulation. Not surprisingly, in view of the long-range of the Coulomb potential, special care is needed when inserting ions. One might think that charge neutrality would require the joint insertion of positive and negative ions that, together, are neutral. However, this is not necessary, provided that an ion is always inserted together with its uniform neutralizing background [329].

Relation to fugacity

The full expression for the chemical potential of a species depends on Planck's constant, through the thermal De Broglie wavelength and, possibly, the internal

partition function of a molecule. However, as we argued in section 8.1.2, the results of a classical simulation cannot depend on Planck's constant.

This statement also applies to the particle-insertion method. One can view this method as a technique to compute the fugacity f_α of a species α :

$$f_\alpha = \frac{\rho_\alpha}{\langle \exp(-\beta \Delta \mathcal{U}_\alpha) \rangle}, \quad (8.5.6)$$

where ρ_α denotes the number density of species α , and $\Delta \mathcal{U}_\alpha$ the change in the potential energy of the system upon random insertion of a test particle of type α . Clearly, f_α does not depend on Planck's constant.

Inhomogeneous systems

Eq. (8.5.5) was derived for a spatially homogeneous system. Widom [330] also considered the case of a spatially inhomogeneous system. In that case, μ_{ex} depends explicitly on the position \mathbf{r} . However, in equilibrium the chemical potential itself is constant throughout the system (see, however, Eq. 4 in ref. [139]). In other words

$$\mu = k_B T \ln \left(\frac{\rho(\mathbf{r})}{\langle \exp[-\beta \Delta \mathcal{U}(\mathbf{r})] \rangle_N} \right) \quad (8.5.7)$$

is constant.

Eq. (8.5.7) can be exploited to improve the statistical accuracy of the numerical estimate of μ . As an illustration, consider a simple example that we can treat analytically: infinitely thin needles of length L in an infinitely long slit with width $W \ll L$. The excess chemical potential of the needles is related to the probability $P_{\text{n.o.}}$ that a needle inserted in the slit with a random orientation, will not overlap with the walls: $\mu_{\text{ex}} = -k_B T \ln P_{\text{n.o.}}$. Clearly, if the orientation of the needle that is inserted is random, the probability that it will overlap with the walls will be close to 100%. However, if we only insert needles in the middle of the slit, that are aligned with the walls, then $P_{\text{n.o.}}^{\parallel} = 1$ and $\mu_{\text{ex}}^{\parallel} = 0$. In that case, $\mu = \mu^{\parallel} = k_B T \ln \rho^{\parallel}$. In other words, we do not have to carry out any particle insertions: we just determine the density ρ , and the orientational distribution of the needles in the middle of the slit. This information allows us to compute the probability density p^{\parallel} , and hence μ .

Example 17 (Chemical potential: particle-insertion method). In this Example, we use the Widom test-particle method to determine the excess chemical potential of a Lennard-Jones fluid. The algorithm that we use is a combination of the basic algorithm for performing Monte Carlo simulations at constant N , V , and T (Algorithms 1 and 2) and determining the excess chemical potential (Algorithm 18).

We stress that the tail correction for the chemical potential is similar, but not identical, to that for the potential energy. In the Widom test-particle

method, we determine the energy difference:

$$\Delta\mathcal{U} = \mathcal{U}(\mathbf{s}^{N+1}) - \mathcal{U}(\mathbf{s}^N).$$

The tail correction is

$$\begin{aligned}\beta\mu^{\text{tail}} &= \mathcal{U}(\mathbf{s}^{N+1})^{\text{tail}} - \mathcal{U}(\mathbf{s}^N)^{\text{tail}} \\ &= (N+1)u^{\text{tail}}((N+1)/V) - Nu^{\text{tail}}(N/V) \\ &= \left((N+1)\frac{N+1}{V} - N\frac{N}{V}\right) \frac{1}{2} 4\pi \int_{r_c}^{\infty} dr r^2 u(r) \\ &\approx \frac{2N}{V} \frac{1}{2} 4\pi \int_{r_c}^{\infty} dr r^2 u(r) \\ &= 2u^{\text{tail}}(\rho).\end{aligned}\quad (8.5.8)$$

In Case Study 9, we performed a grand-canonical Monte Carlo simulation to determine the equation of state of the Lennard-Jones fluid. In the grand-canonical ensemble the volume, chemical potential, and temperature are imposed; the density is determined during the simulation. Of course, we can also calculate the chemical potential during the simulation, using the Widom method. Fig. 8.2 shows a comparison of the imposed and measured chemical potentials.

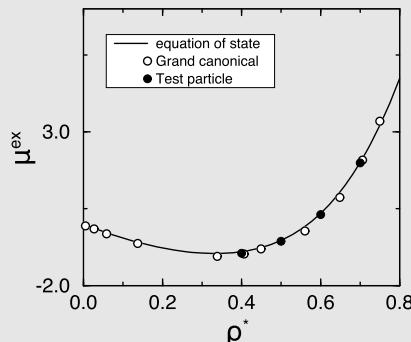


FIGURE 8.2 The excess chemical potential of the Lennard-Jones fluid ($T = 2.0$) as calculated from the equation of state, grand-canonical Monte Carlo, and the test particle insertion method.

The Fortran code to generate this Example can be found in the online-SI, Case Study 15.

8.5.2 Particle-insertion method: other ensembles

The extension of the Widom method to other ensembles, in particular to the NPT and the μ, V, T ensemble is relatively straightforward and is based on the same expression as Eq. (8.5.5), provided the averaging is done correctly.

The situation is a bit different for the NVE ensemble [331–333] and this case is discussed below.

For systems at constant μ , V , T , the chemical potential is imposed, and hence the Widom method can serve to check if the imposed chemical potential is equal to the one measured by particle insertion, as a validation, for instance, to check that this system is well equilibrated. Below, we show that the two should be rigorously the same.

In grand-canonical simulations, N is fluctuating, but we can compute $\exp(-\beta \Delta \mathcal{U})$ for particle trial insertions. From the previous section, we know that

$$\frac{V/\Lambda^d}{N+1} \langle \exp(-\beta \Delta \mathcal{U}) \rangle_N = \frac{Q(N+1, V, T)}{Q(N, V, T)} .$$

This result is for fixed N . The average of $V/(\Lambda^d(N+1)) \exp(-\beta \Delta \mathcal{U})$ over all realizations of the Grand-Canonical ensemble (i.e., with fluctuating N) is

$$\begin{aligned} \left\langle \frac{V/\Lambda^d}{N+1} \langle \exp(-\beta \Delta \mathcal{U}) \rangle_N \right\rangle_\mu &= \frac{\sum_N e^{\beta \mu N} Q(N, V, T) \frac{Q(N+1, V, T)}{Q(N, V, T)}}{\sum_N e^{\beta \mu N} Q(N, V, T)} \\ &= e^{-\beta \mu}, \end{aligned} \quad (8.5.9)$$

and hence,

$$-k_B T \ln \left\langle \frac{V/\Lambda^d}{N+1} \exp(-\beta \Delta \mathcal{U}) \right\rangle_\mu = \mu. \quad (8.5.10)$$

This result shows that if we perform a simulation at constant μ , V , T , the particle-insertion method should yield a value for the chemical potential equal to the one imposed. This result holds independent of system size and provides therefore an excellent diagnostic tool.

Next, we consider the particle-insertion method at constant N , P and T (see [334,335]). We start with the expression for the Gibbs free energy:

$$G(N, P, T) = -k_B T \ln \left\{ \int dV \frac{V^N \exp(-\beta PV)}{\Lambda^{dN} N!} \int ds^N \exp[-\beta \mathcal{U}(s^N; V)] \right\}.$$

Then $\mu = G(N+1, P, T) - G(N, P, T)$ is given by

$$\mu = -k_B T \ln \left\langle \frac{V/\Lambda^d}{(N+1)} \int ds_{N+1} \exp(-\beta \Delta \mathcal{U}) \right\rangle_P, \quad (8.5.11)$$

which is the same expression as for the NVT and $N\mu T$ cases. For simulations at a constant pressure, it is often convenient to express the ideal-gas contribution of the chemical potential in terms of the pressure, rather than the density, that is: we define the ideal gas reference state at the same pressure as the system under

study, rather than at the same average density.

$$\begin{aligned}\mu &= -k_B T \ln(k_B T / P \Lambda^d) - k_B T \ln \left\langle \frac{PV}{(N+1)k_B T} \int d\mathbf{s}_{N+1} \exp(-\beta \Delta \mathcal{U}) \right\rangle \\ &= \mu_{\text{id}}(P) + \mu_{\text{ex}}(P).\end{aligned}\quad (8.5.12)$$

In Eq. (8.5.12), the fluctuating quantity that we are averaging is no longer $\exp(-\beta \Delta \mathcal{U})$, but $V \exp(-\beta \Delta \mathcal{U})$. The fact that V in Eq. (8.5.11) is now fluctuating becomes important under conditions where large volume fluctuations are possible, for instance, in the vicinity of critical points.

To derive a Widom-style expression for the chemical potential in the constant- NVE ensemble, which is the one sampled by conventional Molecular Dynamics simulations, we start from the relation

$$\mu/T = -(\partial S/\partial N)_{VE}.$$

In the microcanonical ensemble, the entropy S is related to $\Omega(N, V, E)$, the total number of accessible states, by $S = k_B \ln \Omega(N, V, E)$. The classical expression for $\Omega(N, V, E)$ is

$$\Omega(N, V, E) = \frac{1}{h^{3N} N!} \int d\mathbf{p}^N d\mathbf{r}^N \delta(\mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) - E). \quad (8.5.13)$$

The derivation proceeds much as before, but we must now compute $\Omega(N+1, V, E)/\Omega(N, V, E)$. This is slightly more cumbersome (see [331–333]), and we quote only the final result:

$$\beta \mu_{\text{ex}} = -\ln \left\{ \langle T \rangle^{-3/2} \left\langle T^{3/2} \exp(-\Delta \mathcal{U}/k_B T) \right\rangle \right\}, \quad (8.5.14)$$

where T is the (fluctuating) temperature, as determined from the instantaneous kinetic energy of the particles. Such fluctuations tend to be large where the heat capacity of the system is large (see [106]).

The particle-insertion estimate for the chemical potential depends rather strongly on system size, at least in the case of a system at constant N, V, T or N, V, E [171,336]. The leading $\mathcal{O}(1/N)$ correction is known for the constant N, V, T case [337]:

$$\Delta \mu_{\text{ex}}(N) = \frac{1}{2N} \left(\frac{\partial P}{\partial \rho} \right) \left[1 - k_B T \left(\frac{\partial \rho}{\partial P} \right) - \rho k_B T \frac{(\partial^2 P / \partial \rho^2)}{(\partial P / \partial \rho)^2} \right]. \quad (8.5.15)$$

As shown in [337], the expression given in Eq. (8.5.15) agrees with the exact result for hard rods in one dimension and is in excellent agreement with numerical results for hard disks in two dimensions.

8.5.3 Chemical potential differences

The particle-insertion method can be modified to measure the *difference* in chemical potential between two species α and β in a mixture [335,338]. In this case, a trial move consists of an attempt to transform a particle of species α into species β , without ever accepting such trial moves.

The phase diagram of mixtures is determined by the composition-dependence of the molar Gibbs free energy

$$G(x_A) = x_A \mu_A + x_B \mu_B, \quad (8.5.16)$$

where $x_A (= 1 - x_B)$ denotes the mole fraction of species A and $\mu_A (\mu_B)$ denotes the chemical potentials of the component in the mixture.

Now suppose that we have computed the Gibbs free energy of one of the pure phases (for instance, by one of the thermodynamic integration methods described in section 8.4.1). At first sight, it might seem that, to compute G as a function of x_A , we would have to repeat such a thermodynamic integration for a large number of x_A values. Fortunately, this is usually not the case. Rather than recomputing $G(x)$ for a number of compositions, we can study the variation of $G(x)$ with x . To this end, we need to have a microscopic expression for

$$\begin{aligned} \left(\frac{\partial G(x)}{\partial x} \right)_{P,T,N} &= \mu_A - \mu_B \\ &= (\mu_A - \mu_B)_{\text{id}} + (\mu_A - \mu_B)_{\text{ex}}. \end{aligned} \quad (8.5.17)$$

In the first line of Eq. (8.5.17), we have used the Gibbs-Duhem relation. We assume that the ideal gas contributions to the chemical potential of both A and B are known. The quantity that we must compute is $\Delta\mu_{\text{ex}} \equiv (\mu_A - \mu_B)_{\text{ex}}$. Naively, we might try to measure this quantity by using the particle insertion method to obtain μ_{ex} of species A and B separately and then subtracting the result. Although such an approach would be correct in principle, it is time-consuming and not particularly accurate. Fortunately, $\Delta\mu_{\text{ex}}$ can be obtained more directly by measuring the Boltzmann factor associated with a virtual trial move, where a randomly selected particle of type B is transformed into a particle of type A [334,335,339] (see Fig. 8.3). We leave it as an exercise to the reader to derive that the resulting expression for $\Delta\mu_{\text{ex}}$ is

$$\Delta\mu_{\text{ex}} = -k_B T \ln \left(\frac{N_B}{N_A + 1} \exp(-\beta \Delta\mathcal{U}^{+-}) \right), \quad (8.5.18)$$

where $\Delta\mathcal{U}^{+-}$ denotes the change in potential energy of the system if one particle of type B is changed into type A ; $-k_B T \ln(N_B/[N_A + 1])$ is simply the ideal mixing contribution to the chemical potential. The point to note about Eq. (8.5.18) is that, for an ideal mixture (i.e., A and B have the same intermolecular interactions), $\ln(\exp(-\beta \Delta\mathcal{U}^{+-}))$ is identically equal to zero. In other

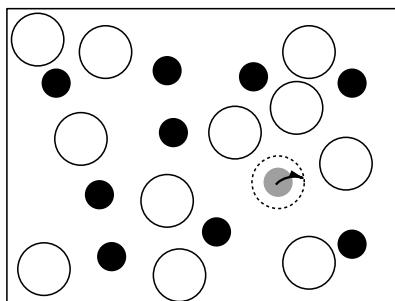


FIGURE 8.3 A schematic sketch of a particle-swap trial move to change the identity of the molecules in a mixture. The figure shows an attempt to transform a small molecule into a big one.

words, we may obtain good statistics on $\Delta\mu_{\text{ex}}$ even when the direct measurement of the excess chemical potential of the individual species would yield poor statistics.

We note that particle insertion and swapping techniques are not limited to the measurement of chemical potentials. In fact, a wide class of partial molar quantities (such as the partial molar enthalpy h_α or the partial molar volume v_α) can be measured in this way. For details, see [335,340].

8.6 Histogram methods

In section 8.5.1 we related the chemical potential of a species to the average Boltzmann factor associated with the attempted insertion of a particle into the system. We did not consider *particle removal* trial moves. Why not? After all, the chemical potential can also be written as

$$\begin{aligned}\mu &= +k_B T \ln(Q_N/Q_{N+1}) \\ &= \mu_{\text{id}} + k_B T \ln \langle \exp(+\beta \Delta U) \rangle_{N+1},\end{aligned}\quad (8.6.1)$$

where ΔU denotes the interaction energy of particle $N+1$ with the remaining N particles. Eq. (8.6.1) suggests that the chemical potential can also be estimated by sampling $\langle \exp(+\beta \Delta U) \rangle_{N+1}$ for random particle removals. Yet, as we explain below, such a procedure will not work in general. The reason is that the function $\exp(\beta \Delta U)$ need not be bounded. It can become arbitrarily large, if ΔU can diverge. The same is not true for $\exp(-\beta \Delta U)$, because one of the conditions that a system must satisfy to be describable by classical statistical mechanics is that its potential energy function must be bounded from below. The problem with Eq. (8.6.1) is that very large values of the integrand coincide with very small values $\mathcal{O}(\exp(-\beta \Delta U))$ of the Boltzmann factor, which determines how often a configuration is sampled during a Monte Carlo run. As a consequence, an appreciable contribution to the average in Eq. (8.6.1) comes from a part of configuration space that is hardly ever, or indeed never, sampled

during a run. Hard spheres offer a good illustration of the nature of the problem. As the potential energy function of non-overlapping hard spheres is always zero, a simple Monte Carlo sampling of Eq. (8.6.1) for a dense fluid of hard spheres would always yield the nonsensical estimate $\mu_{\text{ex}} = 0$, whereas, in fact, at freezing, $\mu_{\text{ex}}/k_B T \sim 15$. The correct way to obtain chemical potentials from simulations involving both particle insertions and particle removals has been indicated by Shing and Gubbins [341,342]. We find it convenient to discuss the Shing-Gubbins approach in the context of a more general technique for measuring free energy differences, introduced by Bennett [343], called the *overlapping distribution method*. This method provides an early illustration of a calculation of a free-energy landscape as a function of an order parameter.

8.6.1 Overlapping-distribution method

Consider two N -particle systems, labeled 0 and 1 with partition functions Q_0 and Q_1 . For convenience, we assume that both systems have the same volume V , but this is not essential. From Eq. (6.3.4) it follows that the free energy difference $\Delta F = F_1 - F_0$ can be written as

$$\begin{aligned}\Delta F &= -k_B T \ln(Q_1/Q_0) \\ &= -k_B T \ln\left(\frac{\int d\mathcal{X} \exp[-\beta U_1(\mathcal{X})]}{\int d\mathcal{X} \exp[-\beta U_0(\mathcal{X})]}\right).\end{aligned}\quad (8.6.2)$$

Suppose that we are carrying out a (Metropolis) sampling of the configuration space of system 1. For every configuration visited during this sampling of system 1 we can compute the potential energy of system 0 ($U_0(s^N)$) for the same configuration and, hence, the potential energy difference $\Delta U = U_1(s^N) - U_0(s^N)$. We use this information to construct a histogram that measures the probability density of the potential energy difference ΔU . Let us denote this probability density by $p_1(\Delta U)$. In the N, V, T -ensemble, $p_1(\Delta U)$ is given by

$$p_1(\Delta U) = \frac{\int d\mathcal{X} \exp(-\beta U_1) \delta(U_1 - U_0 - \Delta U)}{q_1}, \quad (8.6.3)$$

where we have denoted the scaled, configurational part of the partition function by a q (e.g., $q_1 = \int d\mathcal{X} \exp[-\beta U_1(\mathcal{X})]$). The δ -function in Eq. (8.6.3) allows us to substitute $U_0 + \Delta U$ for U_1 in the Boltzmann factor; hence,

$$\begin{aligned}p_1(\Delta U) &= \frac{\int d\mathcal{X} \exp[-\beta(U_0 + \Delta U)] \delta(U_1 - U_0 - \Delta U)}{q_1} \\ &= \frac{q_0}{q_1} \exp(-\beta \Delta U) \frac{\int ds^N \exp(-\beta U_0) \delta(U_1 - U_0 - \Delta U)}{q_0} \\ &= \frac{q_0}{q_1} \exp(-\beta \Delta U) p_0(\Delta U),\end{aligned}\quad (8.6.4)$$

where $p_0(\Delta\mathcal{U})$ is the probability density of finding a potential energy difference $\Delta\mathcal{U}$ between systems 1 and 0, while Boltzmann sampling the available configurations of system 0. As the free energy difference between systems 1 and 0 is simply $\Delta F = -k_B T \ln(q_1/q_0)$, we find from Eq. (8.6.4) that

$$\ln p_1(\Delta\mathcal{U}) = \beta(\Delta F - \Delta\mathcal{U}) + \ln p_0(\Delta\mathcal{U}). \quad (8.6.5)$$

To obtain ΔF from Eq. (8.6.5) in practical cases, it is convenient to define two functions f_0 and f_1 by

$$f_0(\Delta\mathcal{U}) = \ln p_0(\Delta\mathcal{U}) - \frac{\beta\Delta\mathcal{U}}{2} \quad (8.6.6)$$

and

$$f_1(\Delta\mathcal{U}) = \ln p_1(\Delta\mathcal{U}) + \frac{\beta\Delta\mathcal{U}}{2} \quad (8.6.7)$$

such that

$$f_1(\Delta\mathcal{U}) = f_0(\Delta\mathcal{U}) + \beta\Delta F.$$

Suppose that we have measured f_0 and f_1 in two separate simulations: one sampling system 0, the other system 1. We can then obtain ΔF by fitting the functions f_0 and f_1 to two polynomials in $\Delta\mathcal{U}$ that are identical but for the constant term. The constant offset between the two polynomials yields our estimate for ΔF . Note that, to perform such a fit, it is not really necessary that there exists a range of $\Delta\mathcal{U}$ where both f_0 and f_1 can be measured. However, in the absence of such a range of overlap, the statistical accuracy of the method is usually poor.

Now consider the particle insertion-removal problem. Let us assume that system 1 is a system with N interacting particles, while system 0 contains $N - 1$ interacting particles and 1 ideal gas particle. The difference in free energy between these two systems is obviously equal to μ_{ex} . Applying Eq. (8.6.5) to this particular case, we find that

$$\beta\mu_{\text{ex}} = f_1(\Delta\mathcal{U}) - f_0(\Delta\mathcal{U}). \quad (8.6.8)$$

Eq. (8.6.8) is equivalent to the result obtained by Shing and Gubbins. Using the overlapping distribution method it is possible to combine the results of simulations with trial insertions and trial removals to arrive at a more accurate estimate of the chemical potential. In section 8.6.10, we discuss the extension of the Bennett method to multiple histograms and indicate the relation with more recent developments in this field [42,344].

Example 18 (Chemical potential: overlapping distributions). In Case Study 15, we used the Widom test-particle method to determine the chemical potential of the Lennard-Jones fluid. This method fails at high densities, where it becomes extremely unlikely to insert a particle at a position where $\exp(-\beta\Delta\mathcal{U})$ in Eq. (8.5.5) is non-negligible. Yet it is those unlikely insertions that dom-

inate the average $\langle \exp(-\beta\mathcal{U}) \rangle$. Because favorable insertions are so rare, the number of such events is subject to relatively large statistical fluctuations, and hence our estimate for $\langle \exp(-\beta\mathcal{U}) \rangle$ is noisy. The *overlapping-distribution method* does not remove this problem, but it provides a good diagnostic tool for detecting such sampling problems.

To implement the overlapping-distribution method, we have to perform two simulations: one simulation using a system of $N + 1$ particles (system 1) and a second system with N particles and one ideal gas particle (system 0). For each of these systems, we determine the distribution of energy differences, Eqs. (8.6.3) and (8.6.4). For system 1, this energy difference $\Delta\mathcal{U}$ is defined as the change of the total energy of the system that would result if one particle, chosen at random, would be transformed into an ideal gas particle. We now make a histogram of the observed values of $\Delta\mathcal{U}$ in this system. This calculation can easily be appended to a standard MC move (Algorithm 2), because in a trial move, we randomly select a particle and compute its interaction energy before a trial move. But that interaction energy is precisely the $\Delta\mathcal{U}$ that we wish to compute. We thus obtain a probability distribution of $\Delta\mathcal{U}$: $p_1(\Delta\mathcal{U})$.

For system 0, we have to determine the energy difference $\Delta\mathcal{U}$, which is the difference in total energy when the ideal gas particle (which could be anywhere in the system) would be turned into an interacting particle. This energy difference equals the energy of a test particle in the Widom method (section 8.5.1). When we determine $p_0(\Delta\mathcal{U})$, at the same time, we can obtain an estimate of the excess chemical potential from the Widom particle insertion method. As explained in the text, it is convenient not to use $p_0(\Delta\mathcal{U})$ and $p_1(\Delta\mathcal{U})$, but the closely related functions $f_0(\Delta\mathcal{U})$ and $f_1(\Delta\mathcal{U})$, defined in Eqs. (8.6.6) and (8.6.7).

In Fig. 8.4 we show how $\mu_{\text{ex}}(\Delta\mathcal{U})$ can be obtained (by using Eq. (8.6.8)) from a plot of $f_0(\Delta\mathcal{U})$ and $f_1(\Delta\mathcal{U})$, as a function of $\Delta\mathcal{U}$. The results shown in the left part of Fig. 8.4 apply to a Lennard-Jones fluid at $\rho = 0.7$, the results on the right are for $\rho = 1.0$.

For the sake of comparison, we have also plotted the results obtained using the Widom particle insertion method. The figure shows that at $\rho = 0.7$ there is a sufficiently large range of energy differences for which the two functions overlap ($-10 < \Delta\mathcal{U} < -5$), in the sense that the noise in both functions is relatively small in this energy range. The result of the overlapping distribution function, therefore, is in good agreement with the results from the Widom method. However, at $\rho = 1.00$, the range of overlap is limited to the wings of the histograms p_0 and p_1 , where the statistical accuracy is poor. As a consequence, our estimate for $\mu_{\text{ex}}(\Delta\mathcal{U})$ is not constant (as it should) but appears to depend on $\Delta\mathcal{U}$. Moreover, the results from the overlapping distribution method are not consistent with the result of the Widom particle insertion method.

Note that two separate simulations are needed to determine the excess chemical potential from the overlapping distribution method. One might think that particle addition and particle removal histograms could be measured in a single simulation of an N -particle system. Such an approach would indeed be

correct if there were no difference between the histograms for particle removal from N and $N + 1$ particle systems, but for dense systems containing only a few hundred particles, the system-size dependence of μ_{ex} can be appreciable.

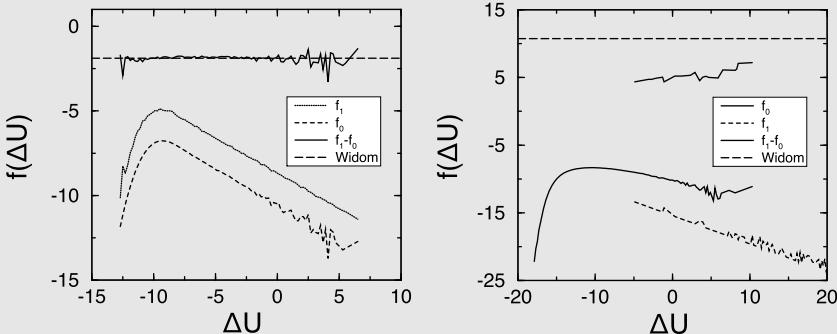


FIGURE 8.4 Comparison of the overlapping distribution function method and the Widom particle insertion scheme for measuring the chemical potential of the Lennard-Jones fluid at $T = 1.2$. The solid curve is the particle-insertion result, the dashed curve was obtained using the overlapping distribution method ($\beta\mu_{\text{ex}} = f_1 - f_0$). The units for $\beta\mu_{\text{ex}}$ are the same as for $f(\Delta U)$. The figure on the left corresponds to a moderately dense liquid ($\rho = 0.7$). In this case, the distributions overlap and the two methods yield identical results. The right-hand figure corresponds to a dense liquid ($\rho = 1.00$). In this case, the insertion probability is very low. The distributions f_0 and f_1 hardly overlap, and the two different estimates of $\beta\mu_{\text{ex}}$ do not coincide.

Of course, sometimes simulations of small systems are performed as a preliminary step to larger-scale simulations. Then it is advisable to compute both $p_0(\Delta U)$ and $p_1(\Delta U)$ for the small system in a single simulation, as this allows us to check whether the overlap between the two distributions is likely to be sufficient to justify a larger-scale simulation.

The Fortran code to generate this Example can be found in the online-SI, Case Study 16.

8.6.2 Perturbation expression

In section 8.6.1, we introduced the distribution functions $p_0(\Delta U)$ and $p_1(\Delta U)$, which measure the probability of finding system 0 (1) in an equilibrium configuration \mathcal{X} for which the potential energies of systems 1 and 0 differ by an amount ΔU . In theory, knowledge of either p_0 or p_1 should be sufficient for estimating the free energy difference between systems 0 and 1, because Eq. (8.6.5) states that

$$p_1(\Delta U) = p_0(\Delta U) \exp[\beta(\Delta F - \Delta U)].$$

If we integrate over ΔU on both sides of this equation, we obtain

$$\int_{-\infty}^{\infty} d\Delta U \, p_1(\Delta U) = \exp(\beta\Delta F) \int_{-\infty}^{\infty} d\Delta U \, p_0(\Delta U) \exp(-\beta\Delta U)$$

$$1 = \exp(\beta\Delta F) \langle \exp(-\beta\Delta\mathcal{U}) \rangle_0, \quad (8.6.9)$$

or

$$\exp(-\beta\Delta F) = \langle \exp(-\beta\Delta\mathcal{U}) \rangle_0. \quad (8.6.10)$$

Although Eq. (8.6.9) is useful to estimate free energy differences between two systems that are not too dissimilar, its applicability is limited. The problem is that, in many cases of practical interest, the largest contributions to the average $\langle \exp(-\beta\Delta\mathcal{U}) \rangle_0$ come from the region of configuration space, where $p_0(\Delta\mathcal{U})$ is very small while $\exp(-\beta\Delta\mathcal{U})$ is very large. As a result, the statistical error in ΔF can be large.

One important application of Eq. (8.6.10) is that it can be used to write down a simple expression for the pressure of a system with many-body interactions. To this end, we use the fact that¹¹

$$\begin{aligned} P &= - \left(\frac{\partial F}{\partial V} \right)_{N,T} = - \lim_{\Delta V \rightarrow 0} \frac{(F(V + \Delta V) - F(V))}{\Delta V} \\ &= \rho k_B T + k_B T \lim_{\Delta V \rightarrow 0} \frac{\ln \langle \exp(-\beta[\mathcal{U}(V + \Delta V) - \mathcal{U}(V)]) \rangle}{\Delta V}. \end{aligned} \quad (8.6.11)$$

8.6.3 Acceptance-ratio method

We consider one final method to compute the free energy difference between two states of a system: the so-called *acceptance ratio* method, introduced by Bennett in 1976 [343]. In fact, the method will later return in disguise when we discuss the so-called **Multistate Bennett Acceptance Ratio (MBAR)**. However, a separate discussion of the original Bennett method is justified, because it is arguably the earliest Monte Carlo algorithm that was explicitly optimized to minimize the statistical error and, on a more subjective level, because it is elegant.

The acceptance ratio method is designed to estimate the free energy difference between two systems (0 and 1) from two simulations: one of system 0 and one of system 1. In the context of the discussion of chemical potentials, system 0 might be a system with N particles, while system 1 contains $N + 1$ particles.

We wish to obtain an expression for the free-energy difference, by computing the ratio of the configurational parts of the partition function (Z_i) for the two systems, $i = 0$ and $i = 1$:

$$Z_i = \int d\mathcal{X} \exp[-\beta\mathcal{U}_i(\mathcal{X})].$$

¹¹ Care should be exercised when applying Eq. (8.6.11) to hard-core systems because the derivative of F with respect to ΔV is discontinuous at $\Delta V = 0$. For hard spheres, $\exp(-\beta[\mathcal{U}(V + \Delta V) - \mathcal{U}(V)]) = 1$ for all $\Delta V \geq 0$, but not for $\Delta V < 0$. However, for non-spherical hard particles, $\exp(-\beta[\mathcal{U}(V + \Delta V) - \mathcal{U}(V)])$ can be less than 1, for certain geometries, even when $\Delta V > 0$. If that is the case, we must combine the results for positive and negative ΔV , in the spirit of section 8.6.3.

We can write the following identity:

$$\begin{aligned} \frac{Z_0}{Z_1} &= \frac{Z_0 \int d\mathcal{X} w(\mathcal{X}) \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))]}{Z_1 \int d\mathcal{X} w(\mathcal{X}) \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))]} \\ &= \frac{\int d\mathcal{X} w(\mathcal{X}) \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))]}{Z_1} \\ &\quad \times \frac{Z_0}{\int d\mathcal{X} w(\mathcal{X}) \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))]} \\ &= \frac{\langle w \exp(-\beta \mathcal{U}_0) \rangle_1}{\langle w \exp(-\beta \mathcal{U}_1) \rangle_0}. \end{aligned} \quad (8.6.12)$$

Eq. (8.6.12) is valid for an arbitrary choice of w . The question is: what choice of w yields the highest statistical accuracy for the estimate of $\beta \Delta F = \ln(Z_0/Z_1)$? Let us first write ΔF in terms of w :

$$\beta \Delta F = \ln \langle w \exp(-\beta \mathcal{U}_0) \rangle_1 - \ln \langle w \exp(-\beta \mathcal{U}_1) \rangle_0. \quad (8.6.13)$$

The variance in $\beta \Delta F$ is due to the fact that we sample a finite number of points n_0 (or n_1). It is convenient to replace the integral over configuration space by a sum, which is a reasonable approximation because floating point numbers are, after all, discrete. Then

$$\langle w \exp(-\beta \mathcal{U}_0) \rangle_1 = \lim_{n_1 \rightarrow \infty} \frac{\sum_k w_k \exp(-\beta \mathcal{U}_0(k)) n_k^{(1)}}{\sum_k n_k^{(1)}},$$

where the total number of samples equals $n_1 \equiv \sum_k n_k^{(1)}$.

The variance in the finite- $n_1(n_0)$ estimate of the free energy difference is given by

$$\overline{(\beta \Delta F)^2} - \overline{(\beta \Delta F)}^2 \approx \sum_k \left(\frac{\partial \overline{\beta \Delta F}}{\partial n_k} \right)^2 \left[\langle n_k^2 \rangle - \langle n_k \rangle^2 \right].$$

As the variances of the two terms (one corresponding to system 1, the other to system 0) add up, we consider only one term and later add the corresponding second term. We first compute the derivative of F_1 with respect to n_k , the number of points sampled at state k

$$\left(\frac{\partial \ln \overline{w \exp(-\beta \mathcal{U}_0)}}{\partial n_k^{(1)}} \right) = \frac{w_k \exp(-\beta \mathcal{U}_0(k))}{\sum_{k'} w_{k'} \exp(-\beta \mathcal{U}_0(k')) n_{k'}^{(1)}} - \frac{1}{\sum_{k'} n_{k'}^{(1)}}$$

which, when replacing the sum over k' by its average, we can write as

$$\frac{1}{n_1} \left(\frac{w_k \exp(-\beta \mathcal{U}_0(k))}{\langle w \exp(-\beta \mathcal{U}_0) \rangle} - 1 \right) = \frac{1}{n_1} \frac{w_k \exp(-\beta \mathcal{U}_0(k)) - \langle w \exp(-\beta \mathcal{U}_0) \rangle}{\langle w \exp(-\beta \mathcal{U}_0) \rangle}.$$

We now square

$$\left(\frac{\partial \ln \overline{w \exp(-\beta \mathcal{U}_0)}}{\partial n_k^{(1)}} \right),$$

and multiply by the variance in $n_k^{(1)}$. If we assume that all samples are independent, then the $n_k^{(1)}$ are Poisson distributed, and hence

$$\langle (n_k^{(1)})^2 \rangle - \langle n_k^{(1)} \rangle^2 = \langle n_k^{(1)} \rangle.$$

Finally, we sum over all k , and do the same for the term related to system 0. Then we get:

$$\begin{aligned} \sigma_{\beta \Delta F}^2 &= \frac{\langle [w \exp(-\beta \mathcal{U}_1)]^2 \rangle_0 - \langle w \exp(-\beta \mathcal{U}_1) \rangle_0^2}{n_0 \langle w \exp(-\beta \mathcal{U}_1) \rangle_0^2} \\ &\quad + \frac{\langle [w \exp(-\beta \mathcal{U}_0)]^2 \rangle_1 - \langle w \exp(-\beta \mathcal{U}_0) \rangle_1^2}{n_1 \langle w \exp(-\beta \mathcal{U}_0) \rangle_1^2} \\ &= \int d\mathcal{X} \left[\left[(Z_0/n_0) \exp(-\beta \mathcal{U}_1(\mathcal{X})) + (Z_1/n_1) \exp(-\beta \mathcal{U}_0(\mathcal{X})) \right] \right. \\ &\quad \times w^2 \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))] \Big] \\ &\quad \times \frac{1}{\left\{ \int d\mathcal{X} w \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))] \right\}^2} - \frac{1}{n_0} - \frac{1}{n_1}. \end{aligned} \quad (8.6.14)$$

Note that the right-hand side of Eq. (8.6.14) does not change if we multiply w by a constant factor. Therefore, without loss of generality, we can choose the following normalization for w :

$$\int d\mathcal{X} w \exp[-\beta(\mathcal{U}_0(\mathcal{X}) + \mathcal{U}_1(\mathcal{X}))] = C. \quad (8.6.15)$$

Next, we minimize the statistical error in $\beta \Delta F$ with respect to w , with the constraint (8.6.15). This is done most conveniently using Lagrange multipliers:

$$\begin{aligned} 0 &= [(Z_0/n_0) \exp(-\beta \mathcal{U}_1) + (Z_1/n_1) \exp(-\beta \mathcal{U}_0)] \exp[-\beta(\mathcal{U}_0 + \mathcal{U}_1)] w \delta w \\ &\quad - \lambda \exp[-\beta(\mathcal{U}_0 + \mathcal{U}_1)] \delta w \end{aligned} \quad (8.6.16)$$

or

$$w = \frac{C}{(Z_0/n_0) \exp(-\beta \mathcal{U}_1) + (Z_1/n_1) \exp(-\beta \mathcal{U}_0)}. \quad (8.6.17)$$

If we now insert this expression for w in Eq. (8.6.12), we obtain

$$\frac{Z_0}{Z_1} = \frac{\langle \{1 + \exp[\beta(\mathcal{U}_0 - \mathcal{U}_1 + C)]\}^{-1} \rangle}{\langle \{1 + \exp[\beta(\mathcal{U}_1 - \mathcal{U}_0 - C)]\}^{-1} \rangle} \exp(\beta C), \quad (8.6.18)$$

where we have defined $\exp(\beta\mathcal{C}) \equiv (Z_0n_1)/(Z_1n_0)$. We can express Eq. (8.6.18) in terms of the Fermi-Dirac/logistic function $f(x) \equiv 1/[1 + \exp(\beta x)]$:

$$\frac{Z_0}{Z_1} = \frac{\langle f(\mathcal{U}_0 - \mathcal{U}_1 + C) \rangle_1}{\langle f(\mathcal{U}_1 - \mathcal{U}_0 - C) \rangle_0} \exp(\beta C). \quad (8.6.19)$$

Note that Eq. (8.6.19) is valid for any choice of C . However, the choice $C = \ln[(Z_0n_1)/(Z_1n_0)]$ is optimal.

At first sight, this choice of C seems problematic because it presupposes knowledge of the very quantity we wish to compute, (Z_0/Z_1) . In practice, C is determined by a self-consistency requirement, described next.

Suppose that we have obtained numerical estimates for $\langle f(\mathcal{U}_0 - \mathcal{U}_1 + C) \rangle_1$ and $\langle f(\mathcal{U}_1 - \mathcal{U}_0 - C) \rangle_0$ for a range of values of C :

$$\begin{aligned} \langle f(\mathcal{U}_0 - \mathcal{U}_1 + C) \rangle_1 &= \frac{1}{n_1} \sum_m f_m (\mathcal{U}_0 - \mathcal{U}_1 + C) \\ \langle f(\mathcal{U}_1 - \mathcal{U}_0 - C) \rangle_0 &= \frac{1}{n_0} \sum_{m'} f_{m'} (\mathcal{U}_1 - \mathcal{U}_0 - C), \end{aligned} \quad (8.6.20)$$

where \sum_m ($\sum_{m'}$) stands for the sum over all configurations sampled in a Monte Carlo simulation of system 1 (0). Inserting Eqs. (8.6.20) and (8.6.19) in Eq. (8.6.13), we obtain

$$\beta\Delta F = \ln \frac{\sum_1 f(\mathcal{U}_0 - \mathcal{U}_1 + C)}{\sum_0 f(\mathcal{U}_1 - \mathcal{U}_0 - C)} - \ln(n_1/n_0) + \beta C, \quad (8.6.21)$$

while the optimal choice for C can be rewritten as

$$\beta\Delta F = -\ln(n_1/n_0) + \beta C. \quad (8.6.22)$$

Clearly, Eqs. (8.6.21) and (8.6.22) are consistent only if

$$\sum_m f(\mathcal{U}_0 - \mathcal{U}_1 + C) = \sum_{m'} f(\mathcal{U}_1 - \mathcal{U}_0 - C). \quad (8.6.23)$$

In practical situations, C will be treated as an adjustable parameter that is varied until Eq. (8.6.23) is satisfied. For that value of C , $\beta\Delta F$ then follows immediately from Eq. (8.6.22).

8.6.4 Order parameters and Landau free energies

The overlapping-distribution method discussed in section 8.6.1 provides a first illustration of how histograms can be used in the context of free-energy calculations. We now consider the more general case where we wish to compute a free-energy profile, for instance in the context of estimating the barrier that a

system has crossed to go from an initial state (often referred to as the “reactant” state) to a final (“product”) state. The free-energy profile depends on a coordinate \mathcal{Q} , which measures the progress from the initial to the final state. \mathcal{Q} , often called the *reaction coordinate* or *order parameter* is a function of the microscopic (e.g., atomic) coordinates \mathcal{X} of the system:

$$\mathcal{Q} = \mathcal{Q}(\mathcal{X}).$$

Simple examples are cases where \mathcal{Q} is a bond length or an inter-atomic distance, but \mathcal{Q} can also depend on all coordinates, for instance, if it measures a particular Fourier component of the density. Above Eq. (8.3.1), we defined an order parameter to distinguish two distinct thermodynamic phases of a system, without specifying what such an order parameter might look like.

We stress (emphatically) that the value of the free-energy function $F(\mathcal{Q})$ depends on the choice of the functional form of \mathcal{Q} . To show this, we recall Eq. (8.1.1), which showed that the ratio of the probabilities to find a system in two macro-states is related to the free-energy difference between those states. We now generalize this relation to the case where a system can be found in different states characterized by the order parameter \mathcal{Q} . The probability density to find the system with a given value of \mathcal{Q} is:

$$\mathcal{P}(\mathcal{Q}) = \frac{\int d\mathcal{X} \delta(\mathcal{Q}(\mathcal{X}) - \mathcal{Q}) \exp[-\beta U(\mathcal{X})]}{\int d\mathcal{X} \exp[-\beta U(\mathcal{X})]}. \quad (8.6.24)$$

Our definition of $\mathcal{P}(\mathcal{Q})$ ensures that it is normalized. We can now define the free energy $F(\mathcal{Q})$ as a function of \mathcal{Q} :

$$\beta F(\mathcal{Q}) = \mathcal{C} - \ln[\mathcal{P}(\mathcal{Q})]. \quad (8.6.25)$$

A free-energy expressed as a function of one or more order parameters is often referred to as a *Landau free energy* [345]. The constant \mathcal{C} has been included because the choice of the zero of free energy is arbitrary: only free-energy differences matter. In what follows, we drop \mathcal{C} . The normalization of $\mathcal{P}(\mathcal{Q})$ then implies

$$\int d\mathcal{Q} e^{-\beta F(\mathcal{Q})} = \int d\mathcal{Q} \mathcal{P}(\mathcal{Q}) = 1. \quad (8.6.26)$$

Now let us choose a different order parameter $\mathcal{Q}' \equiv g(\mathcal{Q})$, where g is a monotonic function of \mathcal{Q} ; for instance: $\mathcal{Q}' = \mathcal{Q}^{17}$. We can now define a free-energy function $F'(\mathcal{Q}')$. As the probability remains unchanged. We must have

$$e^{-\beta F(\mathcal{Q})} d\mathcal{Q} = e^{-\beta F'(\mathcal{Q}')} d\mathcal{Q}' = e^{-\beta F'(\mathcal{Q}')} \left| \left(\frac{\partial \mathcal{Q}'}{\partial \mathcal{Q}} \right) \right| d\mathcal{Q},$$

and hence

$$F(\mathcal{Q}) = F'(\mathcal{Q}') - k_B T \ln \left| \left(\frac{\partial \mathcal{Q}'}{\partial \mathcal{Q}} \right) \right|.$$

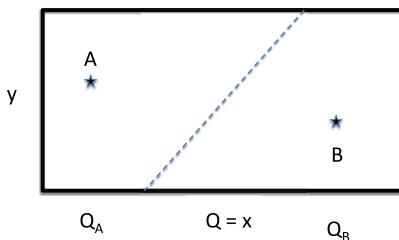


FIGURE 8.5 System with an infinitely thin, impenetrable barrier, oriented obliquely with respect to the ‘order parameter’ \mathcal{Q} . In this system, the free energy as a function of \mathcal{Q} is flat, yet there is no physical path from A to B .

The example above illustrates that we can only define free energies after having specified the order parameter \mathcal{Q}' . There is another problem with free energy profiles: the fact that the free-energy profile $F(\mathcal{Q})$ between two points A and B is relatively flat (i.e., that there is no free-energy barrier) does not necessarily imply that there is a physical path between A and B . To illustrate this, consider a trivial example: a two-dimensional system of a particle in a rectangular box. The box is cut obliquely by an infinitely thin, but infinitely high energy barrier (see Fig. 8.5). We now choose the x coordinate as our order parameter \mathcal{Q} . The free energy is computed by integrating the Boltzmann factor over y at constant \mathcal{Q} . The fact that the Boltzmann factor vanishes on one point along a line of constant \mathcal{Q} makes no difference for the value of the integral. Hence, $F(\mathcal{Q})$ is constant. The path between A (on the left) and B (on the right) appears barrier-free. Yet, clearly, there is no physical path from A to B . The above example, again, illustrates that it is meaningless to speak about *the* free-energy landscape. Identifying good order parameters is clearly important. We will come back to this issue in our discussion of rare events (Chapter 15).

As is clear from Eq. (8.6.25) we can predict the probability density that a system in equilibrium will have an order parameter \mathcal{Q} , if we know the free energy $F(\mathcal{Q})$. The relation between free energy and probability is, in fact, one of the main reasons for computing free-energy profiles (or landscapes). For instance, a system might have a free-energy minimum corresponding to a stable state α and another (local) minimum, corresponding to a metastable state β . We are, however, usually not interested in the probability of finding the system precisely at \mathcal{Q}_α or \mathcal{Q}_β , but rather at the relative probability of finding the system in the free-energy basins around \mathcal{Q}_α and \mathcal{Q}_β .

This relative probability is *not* given by $e^{-\beta[F(\mathcal{Q}_\alpha)-F(\mathcal{Q}_\beta)]}$, but by

$$\frac{P_\alpha}{P_\beta} = \frac{\int_\alpha d\mathcal{Q} e^{-\beta F(\mathcal{Q})}}{\int_\beta d\mathcal{Q} e^{-\beta F(\mathcal{Q})}}, \quad (8.6.27)$$

which is different from $e^{-\beta[F(\mathcal{Q}_\alpha)-F(\mathcal{Q}_\beta)]}$. To illustrate this point, let us assume that the free energy is a quadratic function near \mathcal{Q}_α and near \mathcal{Q}_β , but that the

curvatures of these parabolas are different. For instance, near \mathcal{Q}_α :

$$F(\mathcal{Q}) = F(\mathcal{Q}_\alpha) + \frac{1}{2}k_\alpha(\mathcal{Q} - \mathcal{Q}_\alpha)^2$$

and a similar expression near \mathcal{Q}_β , but with $k_\beta \neq k_\alpha$. Then

$$\frac{P_\alpha}{P_\beta} = e^{-\beta[F(\mathcal{Q}_\alpha) - F(\mathcal{Q}_\beta)]} \times \sqrt{k_\beta/k_\alpha}.$$

This result is hardly surprising: the wider the free-energy basin, the larger the corresponding probability.

8.6.5 Biased sampling of free-energy profiles

Up to this point, we have assumed that we can somehow obtain $P(\mathcal{Q})$ or, equivalently $F(\mathcal{Q})$ from a simulation that is sufficiently long to sample the \mathcal{Q} -range of interest. However, this is usually not the case. Free-energy calculations are often used to determine free-energy barriers that are typically much higher than the thermal energy $k_B T$. In such cases, the probability of sampling points in the barrier region during an equilibrium simulation is negligible.

Fortunately, we can still sample free-energy barriers in equilibrium simulations, provided that we *bias* the sampling towards the regions of interest. In practice this means that we perform a simulation of the same system, but now subject to an imposed potential $\mathcal{V}^{\text{bias}}(\mathcal{Q})$ that is a function of the order parameter(s) only, and that forces the system to sample the \mathcal{Q} -regions that would otherwise be barely visited. Of course, the probability distribution of \mathcal{Q} that is obtained in a biased simulation is not the original distribution $P_0(\mathcal{Q})$ (the subscript 0 denotes the unbiased probability distribution), but rather

$$P_B(\mathcal{Q}) \sim \exp[-\beta\mathcal{V}^{\text{bias}}(\mathcal{Q})]P_0(\mathcal{Q}), \quad (8.6.28)$$

where the subscript B of P_B refers to the biasing potential. However, we know $\mathcal{V}^{\text{bias}}(\mathcal{Q})$, and hence we can unbias $P_B(\mathcal{Q})$ to obtain $P_0(\mathcal{Q})$, up to a multiplicative constant. In other words (using Eq. (8.6.25)):

$$F_0(\mathcal{Q}) = F_B(\mathcal{Q}) - \mathcal{V}^{\text{bias}}(\mathcal{Q}) + \mathcal{C}. \quad (8.6.29)$$

Let us now return to the discussion of methods to sample free-energy profiles as a function of some order parameter \mathcal{Q} . Eq. (8.6.29) shows that it would be convenient to use a biasing potential $V(\mathcal{Q})$ that eliminates all barriers in the free energy landscape, such that $F_B(\mathcal{Q})$ is constant. In fact, such a bias potential exists: it is given by

$$\mathcal{V}^{\text{bias}}(\mathcal{Q}) = -F(\mathcal{Q}) \quad (8.6.30)$$

where, again, we have ignored an irrelevant additive constant. Of course, by writing Eq. (8.6.30) we have just shifted the problem from sampling the free-energy profile, to the problem of finding a biasing potential that gives us a flat free-energy profile.

Many different strategies have been developed to extract free-energy profiles from some form of biased simulation. We will discuss the ideas behind some of the more widely used approaches, but we will not attempt a comprehensive review. However, one comment applies to all calculations of free-energy profiles: as the aim of such calculations is to compute the relative probability of finding a system in different domains in the free-energy landscape, we are never interested in the absolute free energy, but only in free energy differences.

A particularly important class of biasing potentials are those that are simply proportional to the extensive thermodynamic variables that define the state of a system: E , V or $\{N_1, N_2, \dots, N_n\}$. For such thermodynamic biasing potentials, there is a simple physical interpretation of the biased distribution. Let us take energy biasing as an example. To be more precise, we will consider a biasing potential proportional to the potential energy $\mathcal{U}(\mathcal{X})$, where, as before, \mathcal{X} denotes the set of all particle coordinates. In that case

$$\mathcal{V}(\mathcal{Q}) = a \mathcal{U}(\mathcal{X}), \quad (8.6.31)$$

where the value of the factor a determines the strength of the bias. With this bias, the (unnormalized) biased probability \mathcal{P}_B of finding the system with a particular value U_0 of the potential energy is given by:

$$\begin{aligned} \mathcal{P}_B(U) &\sim \int d\mathcal{X} \delta(\mathcal{U}(\mathcal{X}) - U) \exp[-\beta a \mathcal{U}(\mathcal{X})] \exp[-\beta \mathcal{U}(\mathcal{X})] \\ &= \int d\mathcal{X} \delta(\mathcal{U}(\mathcal{X}) - U) \exp[-\beta(1+a)\mathcal{U}(\mathcal{X})], \end{aligned} \quad (8.6.32)$$

which is just the potential energy distribution of the same system, but at an inverse temperature $\beta' \equiv \beta(1+a)$. Hence, in this case, the effect of biasing is simply to sample the system at a different temperature.

Similarly, applying a bias proportional to the volume V in a constant pressure simulation results in sampling the system at a different pressure, and biasing N_i is equivalent to changing the chemical potential of species i in a grand-canonical simulation. We will return to these thermodynamic biasing methods in the context of extended-ensemble sampling schemes (see section 13.1.2).

If we consider the unbiased potential energy distribution $\mathcal{P}_0(U)$

$$\begin{aligned} \mathcal{P}_0(U) &\sim \int d\mathcal{X} \delta(\mathcal{U}(\mathcal{X}) - U) \exp[-\beta \mathcal{U}(\mathcal{X})] \\ &= \exp[-\beta U] \int d\mathcal{X} \delta(\mathcal{U}(\mathcal{X}) - U) \equiv \exp[-\beta U] \mathcal{N}(U), \end{aligned} \quad (8.6.33)$$

where $\mathcal{N}(U)$ is the (unnormalized) density of states of the system at potential energy U . Note that $\mathcal{N}(U)$ does not depend on the temperature. It is fixed once the Hamiltonian of the system has been specified. If we are able to compute the density of states of a system, then we can compute the (configurational part of the) free energy of this system, at any temperature, up to an (unimportant) additive constant:

$$F(N, V, T) = -k_B T \ln \left[\int dU \mathcal{N}(U) \exp(-\beta U) \right]. \quad (8.6.34)$$

Interestingly, Eq. (8.6.34) was already explored in the 1960s by McDonald and Singer [16,275,346].

A widely used Monte-Carlo scheme for obtaining the density of states was introduced by Wang and Landau (see section 8.6.8). Subsequently, we will discuss more recent techniques to reconstruct arbitrary free-energy landscapes. However, for didactical reasons, we first discuss the earliest systematic example of biased sampling, namely the so-called *umbrella sampling method* of Torrie and Valleau [347].

8.6.6 Umbrella sampling

The *umbrella* in the name of this technique is related to the fact that two systems (0 and 1) with different potential energy functions (\mathcal{U}_0 and \mathcal{U}_1) tend to occupy non-overlapping domains in configuration space. Hence, a numerical measurement of the free-energy difference between systems 0 and 1, using the perturbation expression given in Eq. (8.6.10) will fail, because the average of $\exp[-\beta(\mathcal{U}_1(\mathcal{X}) - \mathcal{U}_0(\mathcal{X}))] \equiv \exp(-\beta\Delta\mathcal{U})$ will be vanishingly small for all points sampled.

$$\langle \exp(-\beta\Delta\mathcal{U}) \rangle_0 = \frac{\int d\mathcal{X} \exp[-\beta(\mathcal{U}_1(\mathcal{X}) - \mathcal{U}_0(\mathcal{X}))] \exp[-\beta\mathcal{U}_0(\mathcal{X})]}{\int d\mathcal{X} \exp[-\beta\mathcal{U}_0(\mathcal{X})]} \approx 0.$$

The idea behind umbrella sampling is to construct an artificial (non-Boltzmann) weight function, denoted by $\pi(\mathcal{X})$, to be sampled by Monte Carlo simulation. $\pi(\mathcal{X})$ has to be chosen such that it allows us to sample a domain in configuration space that spans the domains populated by systems 0 and 1. In other words, this modified weight function acts as an “umbrella” that covers the states of both system 0 and system 1.

In the case of umbrella sampling, the expression for $\langle \exp(-\beta\Delta\mathcal{U}) \rangle_0$ becomes (see Eq. (8.6.2))

$$\langle \exp(-\beta\Delta\mathcal{U}) \rangle_0 = \frac{\int d\mathcal{X} \pi(\mathcal{X}) \exp[-\beta\mathcal{U}_1(\mathcal{X})]/\pi(\mathcal{X})}{\int d\mathcal{X} \pi(\mathcal{X}) \exp[-\beta\mathcal{U}_0(\mathcal{X})]/\pi(\mathcal{X})}, \quad (8.6.35)$$

or introducing the notation $\langle \cdots \rangle_\pi$ to denote an average over a probability distribution proportional to $\pi(\mathcal{X})$,

$$\langle \exp(-\beta \Delta \mathcal{U}) \rangle_0 = \frac{\langle \exp(-\beta \mathcal{U}_1)/\pi \rangle_\pi}{\langle \exp(-\beta \mathcal{U}_0)/\pi \rangle_\pi}. \quad (8.6.36)$$

For a suitable choice of the “umbrella” weight function $\pi(\mathcal{X})$, both the numerator and the denominator in Eq. (8.6.36) should be non-zero. Note that $\pi(\mathcal{X})$ need not be of the form $\exp(-\beta[\mathcal{U}_0(\mathcal{X}) + \mathcal{V}^{\text{bias}}(\mathcal{X})])$. In fact, $\pi(\mathcal{X})$ could be non-zero in parts of the configuration space where the Boltzmann weights of both systems 0 and 1 vanish: this would not be useful, but not disastrous. However, $\pi(\mathcal{X})$ should never vanish where either Boltzmann weight is non-zero.

Methods to construct $\pi(\mathcal{X})$ for calculations aimed at computing the properties of a system over a wide range of densities and temperatures are discussed in refs. [229,348,349]. A variant of the umbrella-sampling approach, using biasing in the energy, was introduced by Berg and Neuhaus [350] to obtain an accurate estimate of the density of states of a system. The method of ref. [350] is known as the “multi-canonical ensemble” approach (see SI section L.5).

Illustration 10 (Umbrella sampling of n-butane). The following example is meant to demonstrate the power of the umbrella sampling technique. Consider a model for n-butane, where all bond lengths and bond angles are fixed, except the torsional angle ϕ . Let us assume that we know the intramolecular energy function $\mathcal{U}_{\text{intra}}(\phi)$ associated with changes of the conformation of the molecule. In the dilute gas, the probability of finding a value of the torsion angle ϕ is proportional to $\exp[-\beta \mathcal{U}_{\text{intra}}(\phi)]$. For n-butane, this distribution has a maximum at $\phi = 0^\circ$ (the trans conformation) and two lower maxima at $\phi \approx 120^\circ$, corresponding to the gauche conformation. Let us suppose that we wish to know what happens to the probability of finding a molecule at the transition state between the two gauche conformations when the molecule is dissolved in an atomic liquid. The total potential energy function for the molecule plus solvent is

$$\mathcal{U}_{\text{tot}} = \mathcal{U}_{\text{inter}}(\mathbf{r}^N, \phi_1) + \mathcal{U}_{\text{intra}}(\phi_1).$$

The probability density $P(\phi)$ of finding a particular value of the angle ϕ is given by^a

$$P(\phi) = \frac{\int \exp(-\beta \mathcal{U}_{\text{tot}}) \delta(\phi - \phi_1) d\mathbf{r}^N d\phi_1}{\int \exp(-\beta \mathcal{U}_{\text{tot}}) d\mathbf{r}^N d\phi_1}.$$

Let us now choose the weighting function w equal to $\exp(+\beta \mathcal{U}_{\text{intra}})$. With this choice, we can rewrite $P(\phi)$ as

$$P(\phi) = \frac{\int \exp(-\beta \mathcal{U}_{\text{tot}}) w(\delta(\phi - \phi_1)/w) d\mathbf{r}^N d\phi_1}{\int \exp(-\beta \mathcal{U}_{\text{tot}}) w w^{-1} d\mathbf{r}^N d\phi_1}$$

$$\begin{aligned}
&= \frac{\int \exp(-\beta U_{\text{inter}})[\delta(\phi - \phi_1) \exp(-\beta U_{\text{intra}})] d\mathbf{r}^N d\phi_1}{\int \exp(-\beta U_{\text{inter}}) \exp(-\beta U_{\text{intra}}) d\mathbf{r}^N d\phi_1} \\
&= \frac{\langle \delta(\phi - \phi_1) \exp(-\beta U_{\text{intra}}) \rangle_{\text{inter}}}{\langle \exp(-\beta U_{\text{intra}}) \rangle_{\text{inter}}}.
\end{aligned} \tag{8.6.37}$$

But, as U_{intra} depends only on ϕ , we can rewrite Eq. (8.6.37) as

$$P(\phi) = \frac{\exp[-\beta U_{\text{intra}}(\phi)]}{\langle \exp[-\beta U_{\text{intra}}(\phi)] \rangle_{\text{inter}}} P_{\text{inter}}(\phi),$$

where $P_{\text{inter}}(\phi)$ is the probability of finding a conformation with internal angle ϕ in the absence of the intramolecular torsion barrier. $P_{\text{inter}}(\phi)$ can be computed accurately, even for values of ϕ that are very unlikely in the real system, due to the presence of the internal potential energy barrier U_{intra} .

^a For convenience we ignore that a Jacobian is associated with the transformation from Cartesian to generalized coordinates (see Chapter 14).

8.6.7 Density-of-states sampling

As is clear from Eq. (8.6.33), knowledge of the density of states $\mathcal{N}(U)$ of a system would be sufficient to compute the potential energy distribution at an arbitrary inverse temperature β :

$$\mathcal{P}(U; \beta) \sim \exp[-\beta U] \mathcal{N}(U).$$

However, computing $\mathcal{N}(U)$ is not trivial. For instance, we cannot compute the full $\mathcal{N}(U)$ from $\mathcal{P}(U; \beta) \exp[+\beta U]$, because normal MC or MD sampling explores only a narrow energy range ($\Delta U/U = \mathcal{O}(1/\sqrt{N})$), yet (depending on the system size) $\mathcal{N}(U)$ can easily change by thousands of orders of magnitude, as U is varied. For instance, for N three-dimensional harmonic oscillators $\mathcal{N}(U) \sim U^{(3/2)N}$). In other words: some biased sampling scheme is needed to compute $\mathcal{N}(U)$.

Visiting every state with equal probability, irrespective of its energy, would fail because such a scheme would be equivalent to Boltzmann sampling at infinite temperature, and the simulation would spend most of its time sampling extremely high potential energy states: often this is the unphysical regime where there are multiple particles overlap. We would barely sample the energy ranges of interest.

It would clearly be desirable to have an algorithm that samples all energy ranges rather than *states* with equal probability. This is what Wang-Landau-style algorithms [351,352] achieve by gradually disfavoring the sampling energies that correspond to a high density of states.

In what follows, we discuss the basics of density-of-states sampling. For more details, we refer to the many excellent reviews of this method (see e.g., [281,353]).

However, before proceeding, we recall that the (potential) energy U of a system is but one example of an order parameter. Hence, although the Wang-Landau approach was developed in the context of computing the density of states as a function of U , the idea behind it can be (and has been) modified and extended to sample the free-energy profiles associated with other order parameters \mathcal{Q} . It is in this context that we shall subsequently discuss the extension of the Wang-Landau method due to Calvo [354] and the Metadynamics approach of Laio and Parrinello [355] (see section 8.6.9).

8.6.8 Wang-Landau sampling

In the light of the preceding discussion, it comes as no surprise that the method proposed by Wang-Landau (WL) to sample the density of states $\mathcal{N}(U)$ is *not* based on Importance Sampling of Boltzmann weights. Rather, the algorithm uses a Monte Carlo procedure that, during the simulation, changes the weights of the states that it samples. As a consequence, the WL algorithm does not satisfy detailed balance. However, as the simulation converges, detailed balance is recovered [351,356,357].

As we do not know $\mathcal{N}(U)$ *a priori*, we start with a guess for instance $\mathcal{N}(U) = \text{constant}$. Of course, better initial guesses are possible if we have some simulation data. For instance, if we know $\langle U \rangle$ for one or more values of β , that information fixes the derivative of $\ln \mathcal{N}(U)$ at those values of $\langle U \rangle$ using

$$\left(\frac{\partial \ln \mathcal{N}(U)}{\partial U} \right) = \beta .$$

To distinguish the true density-of-states $\mathcal{N}(U)$ from our evolving estimate, we denote the estimate by $g(U)$: this is also the notation of many papers on the subject. In order to validate that our sampling scheme probes all energy ranges equally, we keep track of a histogram $H(U)$ that counts how often our sampling procedure visits the energy range between U and $U + \Delta U$. Note that ΔU defines a bin width. As is usual with histograms, it should be chosen such that $\mathcal{N}(U)$ does not vary much over the interval ΔU , yet the bin width should be large enough to ensure that, over the course of a simulation, all the bins in the histogram can fill up.

In what follows, we assume that initially, we have chosen $g(U)$ constant. During the simulation, $g(U)$ will vary by many orders of magnitude over the range of U . Hence, it is better to keep track of $\ln g(U)$, with the initial choice $\ln g_0(U) = 0$. The initial choice is unimportant, as long as $g(U)$ is non-zero over the entire range of interest of U (see, however, ref. [358]).

We now start the system at some physically meaningful initial configuration \mathcal{X}_0 , with initial energy $\mathcal{U}(\mathcal{X}_0)$.

From now on, we perform normal Monte Carlo trial moves that satisfy microscopic reversibility, i.e., attempting a move is as likely as attempting the

reverse move. We accept or reject such trial moves according to a Metropolis-like acceptance criterion. For instance, the probability to accept a trial move from configuration o to n is

$$\text{acc}(o \rightarrow n) = \min\left(1, \frac{g(U_o)}{g(U_n)}\right).$$

Note that this acceptance rule favors moves towards configurations with a lower value of $g(U)$. Clearly, as long as $g(U)$ is flat, all moves will be accepted, and if the move is random, we will most likely move to a configuration where $\mathcal{N}(U)$ is larger; after all, for a flat $g(U)$, WL sampling is simply Metropolis sampling at infinite temperature. To ensure that we also sample configurations where $\mathcal{N}(U)$ is small, we have to impose a penalty on moving towards (or staying at) a configuration (either o or n) where $\mathcal{N}(U)$ is largest.

To this end, after MC move number i has been accepted or rejected (or, more precisely, after a number of such moves —see below), we increase the value of $\ln g(U)$ for $U = U_{\text{now}}$, where $= U_{\text{now}}$ is the current value of the energy:

$$\ln g(U_{\text{now}}) \rightarrow \ln g(U_{\text{now}}) + \ln f , \quad (8.6.38)$$

where $\ln f$ is a tuning parameter with a value that is kept constant while we are filling the histogram $H(U)$. Moreover, after every move i , we add 1 to the histogram bin $H(U_i)$ for the current energy.

A typical initial value of f is e^1 . As we disfavor configurations in the region of U where the density of states is high, we will start visiting U values throughout the energy region of interest and $H(U)$ will become flatter, although still quite noisy. In fact, $H(U)$ is noisy for two reasons: one is simply that the (Poisson) fluctuations in the number of points in a bin of our histogram will be of order \sqrt{m} , where m is the expected number of points in that bin. This problem can be alleviated (but not eliminated) by increasing the number of samples used to create the histogram. In addition, $H(U)$ will be noisy, because we have been updating $\ln g(U)$ with steps of size 1: hence $g(U)$ can never be smooth [359]. This problem is addressed in the next step of the algorithm, where we reset H to zero for all U and decrease the value of $\ln f$, e.g.,

$$\ln f \rightarrow \frac{1}{2} \ln f .$$

Note that we do *not* reset $\ln g(U)$. Rather, in the next round of the algorithm, we keep refining $\ln g(U)$, whilst computing the corresponding $H(U)$, which should become increasingly flat. This procedure of decreasing $\ln f$ and resetting H is continued until H no longer becomes flatter. The reason why the procedure stops at some level of refinement is that $H(U)$ will always retain the Poisson noise, and that sets a limit on how far we can refine $g(U)$. Of course, we can decrease that noise by collecting more entries in the histogram.

A detailed discussion of the convergence properties of the WL algorithm can be found in refs. [359,360]. Zhou and Bhatt also argue that updating the density of states in every single Monte Carlo step leads to systematic errors due to correlations between successive updates. It is therefore important to ensure that the number of Monte Carlo steps between updates is sufficiently large [359].

Let us consider what happens at the end of such a simulation, when $g(U)$ has converged. We could then continue the Wang-Landau sampling, but now without updating $g(U)$. Then we recover detailed balance and we can write

$$\frac{\mathcal{N}(U_{(o)})}{\mathcal{N}(U_{(n)})} = \min\left(1, \frac{g(U_o)}{g(U_n)}\right)$$

which is clearly satisfied if

$$g(U) \sim \mathcal{N}(U).$$

In other words: $g(U)$ is our estimate for the density of states.

An obvious prerequisite of the Wang-Landau method is that we adequately sample all relevant configurations. The method will fail if the system is prevented from sampling certain energy values within the specified range. In such cases, special techniques are required to recover a flat histogram [358].

Another important caveat is that, at least for systems where the energy is unbounded from above (i.e., for normal, off-lattice systems —not for most spin systems), we need to impose a reasonable upper limit to the energy. If we do not impose this upper bound, all the simulation time would be spent sampling the density of states for unphysically high energies.

This last point becomes important when considering the extension of the Wang-Landau approach to order parameters other than the energy itself [354]. The reason is that, as we argued before, the Wang-Landau method can be viewed as an infinite-temperature method: the sampling of the density of states is not Boltzmann weighted. However, if we consider other order parameters than the energy, we must make sure that the system still only probes physically meaningful energies. For this reason, the extension of the WL Monte Carlo method to other order parameters [354] and also the meta-dynamics MD approach that we discuss below, are all based on biased Boltzmann sampling. The outcome of such a simulation is therefore not a density of states, but a free-energy profile.

Extension of Wang-Landau to other order parameters

Calvo [354] extended the Wang-Landau method to sample Boltzmann-weighted probability distribution for arbitrary order parameters $\mathcal{Q} = \mathcal{Q}(\mathcal{X})$. As before, the aim is to explore the probability distribution over the relevant range of the order parameter, not just the regions where $\mathcal{P}(\mathcal{Q})$ happens to be large. In the spirit of the Wang-Landau method, this is achieved by building up a bias such that the Monte Carlo sampling results in a flat distribution for the order parameter.

Superficially, the approach of Calvo is similar to WL, but there are subtle differences. The starting point is the order-parameter distribution itself:

$$\mathcal{P}(\mathcal{Q}) = \frac{\int d\mathcal{X} \delta(\mathcal{Q}(\mathcal{X}) - \mathcal{Q}) \exp[-\beta\mathcal{U}(\mathcal{X})]}{\int d\mathcal{X} \exp[-\beta\mathcal{U}(\mathcal{X})]}.$$

According to Eq. (8.6.25), $-k_B T \ln \mathcal{P}(\mathcal{Q})$ is equal to the Landau free energy $F(\mathcal{Q})$ (up to a constant). Therefore, $\mathcal{P}(\mathcal{Q})$ depends not just on \mathcal{Q} but also on the temperature.

Normal Metropolis sampling would only probe $\mathcal{P}(\mathcal{Q})$ near its maximum. To sample the distribution over the desired order-parameter range, we employ the Wang-Landau trick: we bias the acceptance of trial moves from state o to state n with the help of a function $g(\mathcal{Q})$ that we adjust dynamically:

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{\exp(-\beta\mathcal{U}_n) g(\mathcal{Q}_o)}{\exp(-\beta\mathcal{U}_o) g(\mathcal{Q}_n)} \right),$$

and as before, for the order parameter value that results after this move, we increase $\ln g$ by an amount $\ln f$ (same definition as before). In addition, we keep track of the frequency with which different values of \mathcal{Q} are visited by adding 1 to the relevant bin of a histogram $H(\mathcal{Q})$. The subsequent steps are also as for Wang-Landau: after having filled the histogram with a specified number of points, we reset all $H(\mathcal{Q})$ to zero and decrease the value of $\ln f$. This procedure is repeated until $H(\mathcal{Q})$ has reached the desired flatness.

Once $g(\mathcal{Q})$ has converged, we can again make use of the detailed balance condition to obtain the resulting biased probability distribution function $\mathcal{P}_g(\mathcal{X})$, where the subscript g denotes the biasing due to the function $g(U)$:

$$\frac{\mathcal{P}_g(\mathcal{X}_n)}{\mathcal{P}_g(\mathcal{X}_o)} = \frac{\exp(-\beta\mathcal{U}_n) g(\mathcal{Q}_o)}{\exp(-\beta\mathcal{U}_o) g(\mathcal{Q}_n)}. \quad (8.6.39)$$

The important point to note is that $\mathcal{P}_g(\mathcal{X})$ is a distribution over states in configuration space, not over order parameters. However, from Eq. (8.6.39) it follows that

$$\mathcal{P}_g(\mathcal{X}) \sim \frac{\exp(-\beta\mathcal{U}(\mathcal{X}))}{g(\mathcal{Q}(\mathcal{X}))}. \quad (8.6.40)$$

We can use Eq. (8.6.40) to obtain an expression for the biased distribution over order parameters $P_g(\mathcal{Q})$:

$$\begin{aligned} P_g(\mathcal{Q}) &\sim \int d\mathcal{X} \delta(\mathcal{Q}(\mathcal{X}) - \mathcal{Q}) \mathcal{P}_g(\mathcal{X}) \\ &= \frac{1}{g(\mathcal{Q})} \int d\mathcal{X} \delta(\mathcal{Q}(\mathcal{X}) - \mathcal{Q}) \exp[-\beta\mathcal{U}(\mathcal{X})] = \frac{\mathcal{P}(\mathcal{Q})}{g(\mathcal{Q})}. \end{aligned} \quad (8.6.41)$$

But we know that if our Wang-Landau sampling has converged, $P_g(\mathcal{Q})$ should be flat. Hence

$$g(\mathcal{Q}) \sim P(\mathcal{Q}) \quad (8.6.42)$$

or

$$\beta F(\mathcal{Q}) = (\text{constant}) - \ln g(\mathcal{Q}). \quad (8.6.43)$$

In other words, $-k_B T \ln g(\mathcal{Q})$ is equal to the Landau free energy $F(\mathcal{Q})$ (up to an irrelevant, additive constant). Incidentally, the approach by Calvo also provides a solution to the old problem of constructing a good biasing function in umbrella sampling.

The original [Wang-Landau \(WL\)](#) method focused on spin systems with discrete states. However, in discussing the approach of Calvo, we have implicitly assumed that the WL method can be extended to systems with continuous degrees of freedom. For instance, if the density of states is a continuous function of the energy, we can estimate this function by binning the energies [356,361]. A more accurate approach is to combine this binning with an interpolation scheme [356,362]. An alternative approach is to represent the density of state by a set of Gaussians that are updated during the simulation [363] (an approach which is similar to metadynamics, see section 8.6.9).

Thus far, we have discussed one-dimensional histograms, but the Wang-Landau method can be extended to multidimensional densities of states. For the simulations of liquids, this is particularly useful [356,364]. For example, we can consider a system at constant V and T , and compute the density of states $\mathcal{N}_{VT}(E, N)$, where N is the number of particles and E the (potential) energy. Once we know $\mathcal{N}_{VT}(E, N)$, we can compute the grand-canonical partition function $\Xi(\mu, V, T)$ (see Eq. (2.3.19))

$$\Xi(\mu, V, T) = \sum_N \int dE \mathcal{N}_{VT}(E, N) \exp(\beta N \mu - \beta E)$$

for any value of β and μ . Once we know $\mathcal{N}_{VT}(E, N)$, we can compute the usual thermodynamic properties. For example, we can get the pressure from:

$$P = \frac{k_B T}{V} \ln \Xi(\mu, V, T),$$

which is Eq. (2.3.21). Similarly, we can obtain the energy of an N -particle system at constant V and T from:

$$\langle E \rangle_{NVT} = \frac{\int dE \mathcal{N}_{VT}(E, N) E \exp(-\beta E)}{\int dE \mathcal{N}_{VT}(E, N) \exp(-\beta E)}.$$

This approach can be generalized to other ensembles. For instance, we can fix the pressure and temperature of the system and determine the density of states as

a function of V and E , or fix the chemical potential and temperature and obtain the density of states as a function of N and E [356,364].

Examples of applications include studies of the phase behavior of simple fluids, e.g., the determination of the vapor-liquid [356,364] and even the liquid-solid [365] coexistence curves of the Lennard-Jones system, but the approach has also been applied to more complex problems, such as protein folding [366].

8.6.9 Metadynamics

In the previous section, we discussed the Wang-Landau method and related Monte-Carlo schemes aimed at reconstructing the density of states of a model (as a function of U , V or N) and, more generally, methods to compute free-energy profiles or landscapes [354]. Not surprisingly, these methods have been extended to Molecular Dynamics simulations. However, the resulting “metadynamics” method of Laio and Parrinello [355] does more than just translate Wang-Landau to MD.

At its most basic level, metadynamics is similar to the approach of Calvo [354] (see section 8.6.8) in that it forces a many-body system to sample states of increasingly high free energy by penalizing lower free-energy states: a paper from 1994 by Huber et al. [367] provides an early illustration of such an approach in MD.

As the use of the discrete bins in order-parameter space, as used in WL sampling, would be problematic in MD, metadynamics penalizes revisiting the same range in order-parameter space by adding at fixed intervals in time $n\tau_G$, Gaussian repulsive potentials centered at the visited values Q_j of the order parameter Q :

$$V_b^{(n)}(Q) = \sum_{j=1}^{n=t/\tau_G} w_G \exp \left[-\frac{(Q - Q_j)^2}{2\sigma^2} \right], \quad (8.6.44)$$

where w_G is the weight of the deposited Gaussian, and σ is its width. In the simplest form of metadynamics, w_G and σ are kept fixed. Note that these Gaussian potentials play the same role as $\ln f$ in the WL procedure. If the time step τ_G is sufficiently small (meaning that the distance moved in order-parameter space is sufficiently small that the expected change in the free energy is less than the thermal energy), we can view the increase of the biasing potential as a continuous process, where the time is given by $t = n \times \tau_g$, or:

$$V_b(Q, t) = w_G \int_0^t dt' w_G \exp \left[-\frac{(Q - Q(t'))^2}{2\sigma^2} \right] \quad (8.6.45)$$

and

$$\frac{dV_b(Q, t)}{dt} = w_G \exp \left[-\frac{(Q - Q(t))^2}{2\sigma^2} \right]. \quad (8.6.46)$$

Before proceeding, we note that, after the original work of ref. [355], many improvements were made to metadynamics. So much so that, at present, metadynamics comes in many brands and flavors and is, in fact, rather different from the version of ref. [355], which was inspired by the work of Kevrekides and coworkers [368]. In particular, metadynamics started as a technique that used auxiliary variables and an extended Lagrangian, in the spirit of constant pressure and constant-stress MD methods of Andersen [180] and Parrinello and Rahman [178]. However, the currently most popular versions of metadynamics do not use the extended Lagrangian approach. An extended Lagrangian approach is however used in the Temperature-Accelerated Dynamics method of Maragliano and VandenEijnden [369].

In what follows, we will not attempt to review the many sophisticated techniques to make metadynamics better: rather, we focus on those aspects that make metadynamics different. For the more technical aspects of the implementation and for examples of applications, we refer the reader to some of the excellent reviews on this topic [282,370,371].

An important innovation in the development of metadynamics, was the introduction of **Well-Tempered Metadynamics (WTM)** [372], which improves the rate of convergence of the method and, importantly, has actually been proven to converge [373], something that had been generally assumed, but not demonstrated before the work of ref. [373].

WTM improves the convergence to a flat histogram by changing the amplitude of the Gaussians deposited in different parts of the landscape (slower deposition at values of q where $V_b(q; t)$ is already large). This is achieved [372] by changing the rate at which Gaussians are being deposited at point q at time t from w_G to $w_G \exp(-\alpha V_b(q; t))$, where α is a constant. This procedure does however affect the relation between the asymptotic flat histogram $P_g(q)$ (we use the same notation as in the previous section) and the underlying free-energy profile $F(q)$. Here, we use a more heuristic argument to arrive at this result than was used in ref. [372]. Note that the rate at which $V_b(q; t)$ grows is proportional to the probability to find the system at q , i.e., $P_g(q)$ multiplied by the deposition rate at q :

$$\dot{V}_b(q; t) \sim P_g(q) \exp(-\alpha V_b(q; t)).$$

Asymptotically, the shape of $P_g(q; t)$ should not change in time, and hence $\dot{V}_b(q; t)$ should be the same for all q : $V_b(q; t) = V_b(q) + c(t)$, where $c(t)$ is some function of t . Then

$$P_g(q) \exp(-\alpha V_b(q; t)) \sim e^{-\beta F(q)} e^{-\beta V_b(q; t)} e^{-\alpha V(q; t)} = (\text{independent of } q).$$

But this implies that (up to a function $c'(t)$ that does not depend on q).

$$\beta F(q) = -(\beta + \alpha)V_b(q; t) + c'(t).$$

In the notation of ref. [372] (where $\alpha \equiv 1/\Delta T$ and $\beta = 1/T$), and ignoring the q -independent offset $c'(t)$, we get $F(q) = [T/(T + \Delta T)]V_b(q; t)$. Hence, in the well-tempered case, $F(q) \neq -V_b(q)$.

One advantage of well-tempered metadynamics is that it progressively decreases the noise in the free-energy estimate. For these and other aspects of the metadynamics method, we refer the reader to the relevant literature [372–374]. For examples of the combination of metadynamics with the extended ensemble techniques such as parallel tempering, we refer the reader to refs. [375–377].

8.6.10 Piecing free-energy profiles together: general aspects

In the preceding sections, we considered various biasing techniques to achieve uniform sampling of free-energy landscapes. However, in simulations as in any other human activity, you should be careful what you wish for. As we argue below, wishing for (or, more precisely, constructing) a flat, biased probability distribution that can be sampled in a single simulation is not necessarily a good idea. Below, we discuss why it may be useful to break up the calculation of a free-energy profile into separate calculations for many (overlapping) windows.

Let us assume that we wish to compute the free-energy profile $F(Q)$ in the interval $Q_{\max} - Q_{\min} \equiv \Delta Q$. We will compare the estimated simulation time required for a single free-energy calculation over the entire ΔQ -range, with the time required for n such calculations spanning smaller windows $\Delta Q/n$. An example of such a multi-window umbrella sampling approach is the multi-canonical sampling scheme of Berg and Neuhaus [350].

The optimum choice of n would clearly be the one that samples the complete Q interval in the minimum computing time. To estimate this time, let us assume that the system performs a random walk in Q space within the window $\Delta Q/n$. The assumption of diffusion is reasonable because, once we have used WL or metadynamics to flatten the free-energy profile, the system simply diffuses in the interval $\Delta Q/n$. Associated with the random walk in Q space is a diffusion constant D_Q , which (for convenience) we assume to be constant; for the more general case of a varying diffusion coefficient, see [378].

The characteristic time needed to sample one interval $\Delta Q/n$ is then

$$\tau_n = \frac{(\Delta Q/n)^2}{D_Q}.$$

Clearly, the total time to sample all n windows is

$$\tau_{\text{tot}} = n\tau_n = \frac{(\Delta Q)^2}{nD_Q}.$$

Note that the estimated computing time *decreases* with increasing n . It would be incorrect, however, to assume that n should be chosen as large as possible. The actual equilibration time of a run in one of the Q windows also depends

on the rate at which all coordinates orthogonal to \mathcal{Q} are sampled. Let us denote this time by τ_{\perp} . Clearly, once τ_{\perp} becomes appreciably larger than τ_n , the total computation will scale as $n \times \tau_{\perp}$. This suggests that the optimum choice of n is the one for which $\tau_n \approx \tau_{\perp}$. For a more detailed discussion, see [58]. We will also come back to multi-window free-energy calculations in Chapter 13.

Splitting up free-energy profiles in multiple windows may speed up simulations (see e.g., [379–381]), but it comes at a cost: the free-energy profiles are related to the logarithm of the probability distribution, and these distributions are now normalized per window. The result is that the different parts of the free energy curve are shifted with respect to each other: to reconstruct the full curve, we must “stitch” the curves in the different windows together.

Many techniques have been proposed to combine partial free-energy profiles into a single curve, but at the time of writing this edition, one technique, namely **Multistate Bennett Acceptance Ratio (MBAR)** approach [382] has become dominant for several reasons. First of all, many earlier methods represented probability distributions as histograms. The problem with using histograms is that they necessitate choosing a bin width: choose it too wide and all detail is lost, choose it too narrow and many bins will be empty, which is a problem if we wish to use $\beta F(\mathcal{Q}) = -\ln P(\mathcal{Q})$. In contrast, MBAR is bin-less. The second advantage of MBAR is that it does not use interpolation, extrapolation or fitting: MBAR uses all the sampled data points, but only the data points. The final, and most important advantage of MBAR is that it is optimal in a well-defined sense. Below we describe the MBAR approach.

8.6.11 Piecing free-energy profiles together: MBAR

To explain how MBAR works, we start again from Eq. (8.6.24) for the order-parameter distribution that we wish to compute:

$$\mathcal{P}(\mathcal{Q}) = \frac{\int d\mathcal{X} \delta(\mathcal{Q}(\mathcal{X}) - \mathcal{Q}) \exp[-\beta \mathcal{U}(\mathcal{X})]}{\int d\mathcal{X} \exp[-\beta \mathcal{U}(\mathcal{X})]}.$$

In what follows, we will denote $\int d\mathcal{X} \exp[-\beta \mathcal{U}(\mathcal{X})]$ by Z , and $\beta F = -\ln Z$ where, for compactness, we have left out the contribution due to the integration over momenta.

Now suppose that we are not computing $\mathcal{P}(\mathcal{Q})$ in a single simulation, but rather carry out $k = 1 \cdots K$ simulations that have been biased with functions $V_k(\mathcal{Q})$. The expression for the k -th biased distribution is:

$$\mathcal{P}_k(\mathcal{Q}) = \frac{\int d\mathcal{X} \delta(\mathcal{Q}(\mathcal{X}) - \mathcal{Q}) \exp[-\beta (\mathcal{U}(\mathcal{X}) + V_k(\mathcal{Q}(\mathcal{X})))]}{\int d\mathcal{X} \exp[-\beta (\mathcal{U}(\mathcal{X}) + V_k(\mathcal{Q}(\mathcal{X})))]}.$$
 (8.6.47)

Again, we will use a shorthand notation: $Z_k \equiv \int d\mathcal{X} \exp[-\beta (\mathcal{U}(\mathcal{X}) + V_k(\mathcal{Q}(\mathcal{X})))]$, and $\beta F_k \equiv -\ln Z_k$. From the biased distribution $\mathcal{P}_k(\mathcal{Q})$ we can

reconstruct part of the full distribution by using

$$\mathcal{P}(\mathcal{Q}) = \exp[+\beta V_k(\mathcal{Q})] \mathcal{P}_k(\mathcal{Q}) \times \frac{Z_k}{Z}, \quad (8.6.48)$$

but clearly we can only reconstruct this part of $\mathcal{P}(\mathcal{Q})$ from $\mathcal{P}_k(\mathcal{Q})$ if we know Z_k/Z .

MBAR allows us to obtain estimates for the ratios Z_k/Z for the different biasing functions V_k . In fact, as the absolute free energy is never important, we only need to know $\beta \Delta F_k \equiv -\ln(Z_k/Z)$. We also note that Z_k depends on V_k , but at this stage, we need not specify our choice for the order parameter $\mathcal{Q}(\mathcal{X})$. Hence, in what follows, we will denote V_k as $V_k(\mathcal{X})$, without making the dependence on \mathcal{Q} explicit.

The key idea underlying MBAR is that we have information only for those points that have been sampled. Imagine that we had sampled only one point, at coordinate \mathcal{X}_i , then our best estimate for $P(\mathcal{X})$ would be $P(\mathcal{X}) = \delta(\mathcal{X} - \mathcal{X}_i)$. If, instead, we have performed K simulations at different bias potentials V_k ($k = \{1, 2, \dots, K\}$), with N_k points for bias V_k , then MBAR assumes that $P(\mathcal{X})$ must be of the form

$$P(\mathcal{X}) = Z^{-1} \sum_{k=1}^K \sum_{n=1}^{N_k} p_{k,n} \delta(\mathcal{X} - \mathcal{X}_{k,n}), \quad (8.6.49)$$

where

$$Z \equiv \sum_{k=1}^K \sum_{n=1}^{N_k} p_{k,n}. \quad (8.6.50)$$

At this stage, the weights $p_{k,n}$ are unknown. However, once we have determined the $p_{k,n}$, we can use Eq. (8.6.28) to write the biased distributions:

$$P_k(\mathcal{X}) = Z_k^{-1} \sum_{n=1}^{N_k} p_{k,n} \exp[-\beta V_k(\mathcal{X})] \delta(\mathcal{X} - \mathcal{X}_{k,n}), \quad (8.6.51)$$

with

$$Z_k \equiv \sum_{n=1}^{N_k} p_{k,n} \exp[-\beta V_k(\mathcal{X}_{k,n})]. \quad (8.6.52)$$

We determine $p_{k,n}$ by considering the likelihood of the observed sample. The likelihood L is the probability that the observed set of points would be generated for a given set $p_{k,n}$:

$$L = \prod_{k=1}^K \left[\prod_{n=1}^{N_k} P_k(\mathcal{X}_{k,n}) \right]. \quad (8.6.53)$$

Clearly, L depends on all $p_{k,n}$. We can make this explicit by inserting the expression for P_k in the above equation or, what is more convenient, in the expression for $\ln L$:

$$\ln L \equiv \sum_{k=1}^K \sum_{n=1}^{N_k} \ln \frac{p_{k,n}}{\mathcal{Z}_k} \exp[-\beta V_k(\mathcal{X}_{k,n})]. \quad (8.6.54)$$

To maximize the likelihood, we must differentiate with respect to all $p_{k,n}$ to find the condition for $\partial \ln L / \partial p_{k,n} = 0$: Note that only $\ln p_{k,n}$ and the \mathcal{Z}_j depend on $p_{k,n}$. In particular, $V_k(\mathcal{X}_{k,n})$ does not depend on $p_{k,n}$. Hence, we must differentiate:

$$\begin{aligned} \ln L &= \text{constant} + \sum_{k=1}^K \sum_{n=1}^{N_k} [\ln p_{k,n} - \ln \mathcal{Z}_k] \\ &= \text{constant} + \sum_{k=1}^K \sum_{n=1}^{N_k} \ln p_{k,n} - \sum_{k=1}^K N_k \ln \mathcal{Z}_k. \end{aligned} \quad (8.6.55)$$

In the second line of Eq. (8.6.55) we have used the fact that the second sum over n yields simply N_k times the same term. The condition for maximum likelihood is then:

$$0 = \frac{1}{p_{k,n}} - \sum_{k=1}^K N_k \frac{\exp[-\beta V_k(\mathcal{X}_{k,n})]}{\mathcal{Z}_k}. \quad (8.6.56)$$

We can now write an expression for $p_{k,n}/\mathcal{Z}$:

$$p_{k,n}/\mathcal{Z} = \frac{1}{\sum_{k=1}^K N_k \frac{\exp[-\beta V_k(\mathcal{X}_{k,n})]}{(\mathcal{Z}_k/\mathcal{Z})}}. \quad (8.6.57)$$

Having determined the weights $p_{k,n}$, the ratio $\mathcal{Z}/\mathcal{Z}_k$ is our best estimate for Z/Z_k , and hence $\mathcal{Z}_k/\mathcal{Z} = \exp(-\beta \Delta F_k)$. It then follows that

$$p_{k,n}/\mathcal{Z} = \frac{1}{\sum_{k=1}^K N_k \exp[-\beta(V_k(\mathcal{X}_{k,n}) - \Delta F_k)]}. \quad (8.6.58)$$

Finally, we can insert this expression for $p_{k,n}$ in Eq. (8.6.52) to obtain an expression for $\mathcal{Z}_i/\mathcal{Z}$ and thereby for ΔF_i :

$$\Delta F_i = -k_B T \ln \sum_{k=1}^K \sum_{n=1}^{N_k} \frac{\exp[-\beta V_i(\mathcal{X}_{k,n})]}{\sum_{k=1}^K N_k \exp[-\beta(V_k(\mathcal{X}_{k,n}) - \Delta F_k)]}. \quad (8.6.59)$$

Note that this is an implicit equation that has to be solved self-consistently for all ΔF_i .

Of course, a distribution such as Eq. (8.6.49), which is a sum of δ -functions, is not suited for graphical representation. In fact, it looks nothing like the histograms that we are used to. To see this, consider a distribution function that is generated in a single, unbiased simulation. Using Eq. (8.6.56) with $V_k = 0$, we see all $p_n = 1$, for all n . So, how can a distribution function that is a sum of δ functions with equal amplitude represent a Boltzmann distribution?

The answer is that in regions with a high Boltzmann weight, there are many more δ functions than in regions with a low weight. Hence, if we would construct a histogram by integrating the distribution function given by Eq. (8.6.49) over bins of width Δ , we would recover a histogram that does look like a Boltzmann distribution. The important thing is that with MBAR, we determine the optimal distribution once, and then we can do binning or curve fitting afterward.

The most common application of histogram-reconnection methods is computing the distribution of a common thermodynamic variable such as the number-density ρ , order of the energy density e , as a function of an intensive thermodynamic variable such as the inverse temperature β or the fugacity z . In those cases, we can reconstruct the desired histogram at other values of β or z than our reference states β_0 and z_0 by using Eqs. (8.6.49) and (8.6.51), where the bias potential V_k is now of the form $(\beta_k - \beta_0)\mathcal{U}(\mathcal{X})$ or $(\ln z - \ln z_0)N$. In Example 19 we will show an application of such a “histogram-reweighting” procedure.

Finally, where does the name **Multistate Bennett Acceptance Ratio (MBAR)** come from? The reason is simply that the method is a generalization of Bennett’s acceptance-ratio method to determine the free-energy offset between two distributions (see Eq. (8.6.21)).

Illustration 11 (Ideal gas molecule in an external field). To illustrate the self-consistent histogram method, let us consider a trivial example, namely, the simulation of an ideal gas molecule in an external field:

$$u(z) = \begin{cases} z & z > 0 \\ \infty & z \leq 0 \end{cases}.$$

For this system, the probability of finding an ideal gas molecule at a position z is given by the barometric distribution:

$$p_0(z) = C \exp[-\beta u(z)].$$

The Landau free energy as a function of the coordinate z is, in this case, simply equal to the potential energy:

$$F(z) = -k_B T \ln[p_0(z)] = u(z) = z,$$

where we have chosen our reference point at $z = 0$. A direct simulation of the barometric height distribution yields poor statistics if $\beta u(z) \gg 1$. This is

why we use the self-consistent histogram method. For window i we use the following window potential:

$$W_i(z) = \begin{cases} \infty & z < z_i^{\min} \\ 0 & z_i^{\min} < z < z_i^{\max} \\ \infty & z > z_i^{\max} \end{cases}.$$

We allow only neighboring windows to overlap:

$$\begin{aligned} z_{i-2}^{\max} < z_i^{\min} < z_{i-1}^{\max} \\ z_{i+2}^{\min} > z_i^{\max} > z_{i+1}^{\min}. \end{aligned}$$

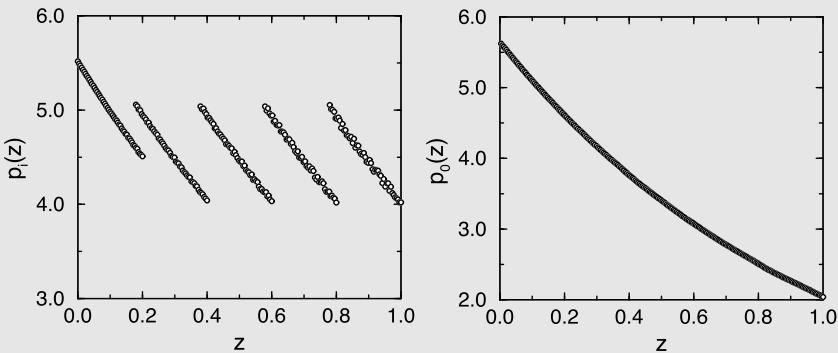


FIGURE 8.6 The probability of finding an ideal gas particle at position z . The figure on the left shows the results for the various windows, and the right figure shows the reconstructed distribution function as obtained from the self-consistent histogram method.

For each window, we perform M samples to estimate the probability $p_i(z)$ to find an ideal gas particle at a position z . The results of such a simulation are shown in Fig. 8.6(left). A self-consistent histogram method, such as **MBAR** (Eq. (8.6.59)), can be used to reconstruct the desired distribution $p_0(z)$. The result of this calculation is shown in Fig. 8.6(right).

A rather different method to reconstruct free energy surfaces is to make use of the fact that, in equilibrium, there is detailed balance between states characterized by different order parameters. If we carry out a simulation (MC or MD) and measure the forward and reverse rate of transitions between states with order parameter \mathcal{Q} and \mathcal{Q}' , denoted by $\mathcal{R}(\mathcal{Q} \rightarrow \mathcal{Q}')$ and $\mathcal{R}(\mathcal{Q}' \rightarrow \mathcal{Q})$, respectively, then we must have:

$$\mathcal{P}(\mathcal{Q})\mathcal{R}(\mathcal{Q} \rightarrow \mathcal{Q}') = \mathcal{P}(\mathcal{Q}')\mathcal{R}(\mathcal{Q}' \rightarrow \mathcal{Q})$$

or

$$\frac{\mathcal{P}(\mathcal{Q})}{\mathcal{P}(\mathcal{Q}')} = \frac{\mathcal{R}(\mathcal{Q}' \rightarrow \mathcal{Q})}{\mathcal{R}(\mathcal{Q} \rightarrow \mathcal{Q}')},$$

Hence, by measuring the transition rates between narrow windows in the order-parameter distribution, we can reconstruct the shape of $\mathcal{P}(Q)$ [383–386].

8.7 Non-equilibrium free energy methods

Above, we discussed a number of techniques for computing free-energy differences. All these techniques assume either that the system under study is in thermodynamic equilibrium or, as in metadynamics, that the system is changing slowly in time. This choice for systems in or close to equilibrium seems logical, as the free-energy difference between two states is equal to the reversible work needed to transform one state into the other. It is therefore surprising that the free energy difference between two systems can also be obtained by computing the *non-equilibrium* work that is needed to transform one system into the other. In fact, the relation that we discuss is even valid for arbitrarily short “switching” times t_s . In what follows, we briefly describe the non-equilibrium free-energy expression due to Jarzynski [387,388] and some of the generalizations proposed by Crooks [389–391]. As before, we consider two N -particle systems: one with a Hamiltonian $H_0(\Gamma)$ and the other with a Hamiltonian $H_1(\Gamma)$, where $\Gamma \equiv \{\mathbf{p}^N, \mathbf{r}^N\}$ represents the phase space coordinates of the system. We assume that we can switch the Hamiltonian of the N -particle system from H_0 to H_1 — that is, we introduce a Hamiltonian H_λ that is a function of a time-dependent switching parameter $\lambda(t)$, such that for $\lambda = 0$, $H_{\lambda=0} = H_0$, while for $\lambda = 1$, $H_{\lambda=1} = H_1$. Clearly, we can then write

$$H_1[\Gamma(t_s)] = H_0[\Gamma(0)] + \int_0^{t_s} dt \dot{\lambda} \frac{\partial H_\lambda[\Gamma(t)]}{\partial \lambda}. \quad (8.7.1)$$

Note that the work W performed on the system due to the switching of the Hamiltonian is equal to $\int_0^{t_s} dt \dot{\lambda} \frac{\partial H_\lambda[\Gamma(t)]}{\partial \lambda}$. If the switching takes place very slowly, the system remains in equilibrium during the transformation, and W reduces to the *reversible* work needed to transform system 0 into system 1. Under those conditions, $W(t_s \rightarrow \infty) = F_1 - F_0 \equiv \Delta F$. However, for a finite switching time, the average amount of work \overline{W} that must be expended to transform the system from state 0 to state 1 is *larger* than the free energy difference ΔF

$$\overline{W}(t_s) \geq \Delta F.$$

The work $W(t_s)$ depends on the path through phase space and, for a Hamiltonian system, this path itself depends on the initial phase space coordinate $\Gamma(0)$; later we shall consider more general situations where many paths connect $\Gamma(0)$ with $\Gamma(t_s)$. Let us next consider the average of $\exp[-\beta W(t_s)]$. The work $W(t_s)$ is a function of the initial phase space position $\Gamma(0)$. We assume that at time $t = 0$, the system is in thermal equilibrium, in which case the probability of finding the

system 0 with phase space position $\Gamma(0)$ is given by the canonical distribution

$$P_0[\Gamma(0)] = \frac{\exp\{-\beta H_0[\Gamma(0)]\}}{Q_0},$$

where Q_0 is the canonical partition function of the system 0. The average of $\exp[-\beta W(t_s)]$ is then given by

$$\begin{aligned} & \overline{\exp[-\beta W(t_s)]} \\ &= \int d\Gamma(0) P_0[\Gamma(0)] \exp\{-\beta W[t_s, \Gamma(0)]\} \\ &= \int d\Gamma(0) \frac{\exp\{-\beta H_0[\Gamma(0)]\}}{Q_0} \exp\{-\beta W[t_s, \Gamma(0)]\} \\ &= \int d\Gamma(0) \frac{\exp\{-\beta H_0[\Gamma(0)]\}}{Q_0} \exp\{-\beta [H_1(\Gamma(t_s)) - H_0(\Gamma(0))]\} \\ &= \int d\Gamma(0) \frac{\exp\{-\beta H_1[\Gamma(t_s)]\}}{Q_0}, \end{aligned} \quad (8.7.2)$$

where we have used the fact that $W(t_s) = H_1[\Gamma(t_s)] - H_0[\Gamma(0)]$. Finally, we use the fact that the Hamiltonian equations of motion are area-preserving. This implies that $d\Gamma(t_s) = d\Gamma(0)$. We then obtain Jarzynski's central result

$$\begin{aligned} \overline{\exp[-\beta W(t_s)]} &= \int d\Gamma(t_s) \frac{\exp\{-\beta H_1[\Gamma(t_s)]\}}{Q_0} \\ &= \frac{Q_1}{Q_0} = \exp(-\beta \Delta F). \end{aligned} \quad (8.7.3)$$

This is a surprising result because it tells us that we can obtain information about *equilibrium* free-energy differences from a *non-equilibrium* simulation. But actually, we already know two limiting cases of this result. First of all, in the limit of infinitely slow switching, we recover the relation between ΔF and the *reversible* work W_s , written in the form

$$\exp(-\beta \delta F) = \exp(-\beta W_s).$$

The other limit is instantaneous switching. In that case, W is simply equal to $H_1[\Gamma(0)] - H_0[\Gamma(0)]$ and we get

$$\exp(-\beta \Delta F) = \langle \exp(-\beta \Delta H) \rangle,$$

which is Eq. (8.6.10). Crooks [389] has given a more general derivation of Eq. (8.7.3) that is not limited to Hamiltonian systems. In particular, Crooks showed that Eq. (8.7.3) remains valid, provided that the dynamics of the system is Markovian and microscopically reversible. Crooks' result implies that Eq. (8.7.3) is also valid if the “time-evolution” of the system is determined by a Metropolis Monte Carlo scheme. For more details, see Appendix E.

Eq. (8.7.3) and Crooks' generalization, are both surprising and elegant and are of great conceptual importance. However, there is only little evidence that the non-equilibrium method to compute free-energy differences outperforms existing equilibrium methods [392].

The exception seems to be the calculation of free-energy differences in glassy systems [393] where thermodynamic integration and related methods simply fail. However, when other free-energy methods can also be used, they are at least as good as the Jarzynski method. The underlying reason limiting the practical use of the Jarzynski method is that for transformations far from equilibrium, the statistical accuracy of Eq. (8.7.3) may be quite poor. We already know this for the limit of instantaneous switching. Like in the particle-removal version of the Widom method (Eq. (8.6.1)), the dominant contribution to the average that we wish to compute comes from initial configurations that are rarely sampled. This was the reason why, for the measurement of chemical potentials, the “particle-removal” method was not a viable alternative to the “particle-insertion” scheme. To illustrate the problem in the context of non-equilibrium free-energy calculations, we consider a change in the Hamiltonian of the system that does not change the free energy of the system.

An example would be a Monte Carlo move that displaces one molecule in a liquid over a distance $+X$. If X is not small compared to the typical molecular dimensions, then the displacement of the particle will most likely require positive work to be performed. The same holds for the reverse situation where we move the particle back over a distance $-X$ from its new position to its starting point. However, the free energies of the initial and final situations are the same, and hence ΔF should be zero. This implies that the very small fraction of all configurations for which the work is *negative* makes an equal and opposite contribution to the average of $\exp(-\beta W)$. In fact, as in the particle-insertion/particle-removal case, the resolution of the problem lies in a combination of the forward and reverse schemes. We illustrate this by considering the Hamiltonian system. However, the result is general.

We now consider two non-equilibrium processes: one transforms the Hamiltonian from H_0 to H_1 in a time interval t_s , and the other process does the reverse. For both processes, we can make a histogram of the work that is expended during the transformation. For the forward process, we can write

$$p_0(W) = \int d\Gamma(0) \frac{\exp\{-\beta H_0[\Gamma(0)]\}}{Q_0} \delta[W - W(t_s)]. \quad (8.7.4)$$

If we multiply both sides of this equation by $\exp(-\beta W)$ and use the fact that $W(t_s) = H_1[\Gamma(t_s)] - H_0[\Gamma(0)]$, we get

$$\begin{aligned} \exp(-\beta W) p_0(W) &= \int d\Gamma(0) \frac{\exp\{-\beta H_0[\Gamma(0)]\}}{Q_0} \exp(-\beta W) \delta[W - W(t_s)] \\ &= \int d\Gamma(0) \frac{\exp\{-\beta H_0[\Gamma(0)]\}}{Q_0} \times \end{aligned}$$

$$\begin{aligned}
& \exp\{-\beta H_1[\Gamma(t_s)] - H_0[\Gamma(0)]\} \delta[W - W(t_s)] \\
&= \int d\Gamma(0) \frac{\exp\{-\beta H_1[\Gamma(t_s)]\}}{Q_0} \delta[W - W(t_s)] \\
&= \frac{Q_1}{Q_0} \int d\Gamma(t_s) \frac{\exp\{-\beta H_1[\Gamma(t_s)]\}}{Q_1} \delta[W - W(t_s)] \\
&= \exp(-\beta \Delta F) p_1(-W).
\end{aligned} \tag{8.7.5}$$

In the last line of Eq. (8.7.5), we have used the fact that the work that is performed on going from 1 to 0 is equal to $H_0[\Gamma(0)] - H_1[\Gamma(t_s)] = -W(t_s)$. Hence, just as in the overlapping distribution method (8.6.1), we can obtain ΔF reliably if the histograms of the forward and reverse work show some overlap. The above result provides a powerful diagnostic tool for testing when the approach of Jarzynski and Crooks can be used safely in numerical simulations.

The above result would seem to limit the applicability of the Jarzynski result to the situation where the distributions of forward and reverse work have a non-negligible overlap, which is typically the case close to equilibrium. However, as was shown by Hartmann [394], the method can also be made to work for extreme non-equilibrium situations, by using a biased path-sampling of the non-equilibrium trajectories. In addition, Nilmeier et al. [395,396] showed that the approach of [389] could be used to construct composite MC trial moves with a high acceptance under conditions where the acceptance of normal MC moves would be low.

8.8 Questions and exercises

Question 23 (Free energy).

1. Why does Eq. (8.6.1) fail for hard spheres?
2. Derive an expression for the error in the estimate of the chemical potential obtained by Widom's test particle method for a system of hard spheres. Assume that the probability of generating a trial position with at least one overlap is equal to p .
3. An alternative method for calculating the free-energy difference between state A and state B is to use an expression involving the difference between the two Hamiltonians:

$$F_A - F_B = \frac{-\ln [\langle \exp[-\beta(H_A - H_B)] \rangle_{N,V,T,B}]}{\beta}. \tag{8.8.1}$$

Derive this equation. What limits the practical applicability of the above expression? Show that Widom's test particle method is a special case of this equation.

Question 24 (Virtual volume change). As discussed in section 5.1.5.3, the virial equation is not particularly convenient for computing the pressure of a hard-sphere fluid.

- Why not?

It is more convenient to perform a constant-pressure simulation and compute the density. An alternative way to compute the pressure of a hard sphere fluid is to use a trial volume change. In this method, a virtual displacement of the volume is performed, and the probability that such a (virtual) move is accepted has to be computed. In general, one can consider both trial expansions and trial compressions.

- Discuss under what conditions the pressure can be calculated using only trial compressions. (Hint: consider the analogy of Widom's test particle method.)

Next, consider now a system of hard-core chain molecules.

- Explain how the acceptance ratio method can be used to compute the pressure. Hint: it is useful to start from Eq. (8.6.13), with $w = 1$.

Chapter 9

Free energies of solids

In three dimensions, the fluid and the crystalline states are separated by a first-order phase transition. Unlike the liquid-vapor transition, which ends in a critical point, there is no solid-fluid critical point [345,397,398].¹ As first-order phase transitions proceed irreversibly via a nucleation and growth process, there is no continuous, hysteresis-free path joining the bulk fluid and solid. The hysteresis problem is even worse for first-order solid-solid transitions. The most robust way to locate the coexistence curve involving one or more solid phases is, therefore, to compute the chemical potentials of all relevant phases as a function of temperature, pressure, and possible additional variables.²

In the absence of a natural reversible path between the crystal phase and the dilute gas, we cannot directly relate the chemical potential of a solid to that of a dilute-gas reference state. It is for this reason that special techniques are needed to compute the chemical potential of solids. In practice, such calculations usually involve computing the Helmholtz free energy F of the solid, and then using the relation $G = N\mu = F + PV$ to deduce the chemical potential of the solid. Note that the above relation applies to pure substances. In what follows, we will mainly discuss pure substances, but we will occasionally refer to the extra challenges posed by mixtures.

At the outset, we note that locating phase equilibria involving solids cannot be based on techniques such as the Gibbs-ensemble method of section 6.6, which rely on the possibility of exchanging particles between the coexisting phases. Such particle exchanges become extremely unlikely for a dense phase, in particular for solids: the successful trial insertion of a particle into the solid phase typically requires that there is a non-negligible concentration of vacancies in the solid. Such defects do occur in real solids, but their concentration is so low (for example, in the case of a hard-sphere crystal near melting, there is, on average, one vacancy per 8000 particles) that one would need a rather large crystal (or a biased simulation) to observe a reasonable number of holes in a simulation. Hence, the Gibbs ensemble technique, although still valid in principle,

¹ However, the transition between an isotropic fluid and a (liquid) crystal may end in a critical point in the presence of a strong enough symmetry-breaking field [399]. This approach has been used in simulations in ref. [303].

² In Chapter 8, we discussed direct-coexistence simulations, which are simple in principle but, for transitions involving solids, tricky in practice. Moreover, locating solid-solid coexistence by direct-coexistence simulations is not feasible.

would not be very practical for studying solid-liquid or solid-solid coexistence.³ In cases where the number of particles in a crystal can differ significantly from the number of lattice sites [224,225], particle insertions/removals may be easier, but applying the Gibbs-ensemble method is even more problematic. The reason is that to find the minimal free-energy state of such systems, the number of lattice sites should be free to vary independently of the volume and the number of particles [224]. The standard Gibbs ensemble method tends to constrain the number of lattice sites and will yield nonsensical answers.

Below, we briefly describe some of the salient features of free-energy calculations of crystalline solids. Our discussion is not meant to be comprehensive: for complementary information, we refer the reader to the reviews of Vega et al. [402] and of Monson and Kofke [403].

9.1 Thermodynamic integration

Thermodynamic integration is the method most commonly used in the study of the solid-liquid transition. For the liquid phase, this calculation is straightforward and was already discussed in section 8.4.1: the Helmholtz free energy F of the liquid is determined by integrating this equation of state, starting at low densities where the fluid behaves effectively as an ideal gas:

$$\frac{F(\rho)}{Nk_B T} = \frac{F^{\text{id}}(\rho)}{Nk_B T} + \frac{1}{k_B T} \int_0^\rho d\rho' \left(\frac{P(\rho') - \rho' k_B T}{\rho'^2} \right), \quad (9.1.1)$$

where the equation of state as a function of the density (ρ) is denoted by $P(\rho)$, and $F^{\text{id}}(\rho)$ is the free energy of an ideal gas at density ρ . An important condition is that the integration path in Eq. (9.1.1) should be reversible. If the integration path crosses a strong first-order phase transition, hysteresis may occur, and Eq. (9.1.1) can no longer be used. For a liquid phase, this problem can be avoided by performing the integration in two steps. Start the simulation at a temperature well above the critical temperature and determine the equation of state for compression along an isotherm to the desired density. In the second step, the system is cooled at constant density to the desired temperature. The free energy change in this step is given by

$$\frac{F(T = T_{\text{II}})}{k_B T_{\text{II}}} - \frac{F(T = T_{\text{I}})}{k_B T_{\text{I}}} = \int_{T_{\text{I}}}^{T_{\text{II}}} d(1/T) U(T, N, V). \quad (9.1.2)$$

The solid-liquid coexistence curve itself does not end at a critical point, and hence there exists no “natural” reversible path from the solid to the ideal gas that does not cross a first-order phase transition. It is usually possible, however, to construct a reversible path to other states of known free energy. The construction of such paths is the main topic of the present chapter.

³ In some special cases, the method of [400] (see section 6.6) may make a direct Gibbs ensemble simulation feasible, a related scheme has been proposed for solid mixtures [401].

Various routes arrive at a state of known free energy. In the mid-1960s, Hoover and Ree introduced the so-called *single-occupancy cell method* [276, 307]. In the single-occupancy cell method, the solid is modeled as a lattice gas; each particle is assigned to a single lattice point and is allowed to move only in its “cell” around this lattice point. The lattice sites coincide with the average positions of the atoms of the unconstrained solid. If the density is sufficiently high—such that the walls of the cells have a negligible influence on the properties of the system—the free energy of this lattice model is identical to that of the original solid. The single-occupancy cell model can be expanded uniformly without melting (or, more precisely, without losing its translational order). In this way, we obtain a (presumably reversible) integration path to a dilute lattice gas, the free energy of which can be calculated analytically. The earliest application of the single-occupancy cell method was the calculation by Hoover and Ree of the free energy of the hard-disk [276] and hard-sphere solid [307].

An alternative for the single-occupancy cell method was also developed by Hoover and co-workers [309,311]. In this approach, the solid is cooled to a temperature sufficiently low for it to behave as a harmonic crystal. The Helmholtz free energy of a harmonic crystal can be calculated analytically, using lattice dynamics. The free energy of the solid at higher temperatures then follows from integration of Eq. (9.1.2).⁴

In practice, both the single-occupancy cell method and the method using the harmonic solid have some limitations that make a more general scheme desirable. For example, there is some evidence that the isothermal expansion of the single-occupancy cell model may not be completely free of hysteresis [313]: at the density where the solid would become mechanically unstable in the absence of the artificial cell walls, the equation of state of the single-occupancy cell model appears to develop a cusp or possibly even a weak first-order phase transition. This makes the accurate numerical integration of Eq. (9.1.1) difficult.

The harmonic-solid method can work only if the solid phase under consideration can be cooled reversibly all the way down to the low temperatures where the solid becomes effectively harmonic. However, many molecular solids undergo one or more first-order phase transitions on cooling. Even more problematic are model systems for which the particles interact via a discontinuous (e.g., hard-core) potential. The crystalline phase of such model systems can never be made to behave like a harmonic solid. For complex molecular systems, the problem is of a different nature. Even if these materials can be cooled to become a harmonic crystal, computing the Helmholtz free energy in that limit may be nontrivial.

In the present chapter, we discuss methods that do not suffer from these limitations and can be applied to arbitrary solids [404,405]. Although the method is generally applicable, it is advantageous to make small modifications depending

⁴ If we use Eq. (9.1.2) directly, the integration will diverge in the limit $T \rightarrow 0$. This divergence can be avoided if we determine the difference in free energy of the solid of interest and the corresponding harmonic crystal.

on whether we study an atomic solid with a discontinuous potential [314], or with a continuous potential [406], or a molecular solid [405,407].

9.2 Computation of free energies of solids

The method discussed in this section is a Hamiltonian thermodynamic integration technique (see Section 2.5.1) for computing the Helmholtz free energy of an atomic solid. The basic idea is to transform the solid under consideration reversibly into an Einstein crystal. To this end, the atoms are coupled harmonically to their lattice sites. If the coupling is sufficiently strong, the solid behaves as an Einstein crystal, the free energy of which can be calculated exactly. The method was first used for continuous potentials by Broughton and Gilmer [408], while Frenkel and Ladd [314] used a slightly different approach to compute the free energy of the hard-sphere solid. Extensions to atomic and molecular substances can be found in [406,407].

9.2.1 Atomic solids with continuous potentials

Let us first consider a system that interacts with a continuous potential, $\mathcal{U}(\mathbf{r}^N)$. We shall use thermodynamic integration (Eq. (8.4.8)) to relate the free energy of this system to that of a solid of known free energy. For our reference solid, we choose an Einstein crystal, i.e., a solid of noninteracting particles that are all coupled to their respective lattice sites by harmonic springs. During the thermodynamic integration, we switch on these spring constants and switch off the intermolecular interactions. To this end, we consider a potential energy function

$$\mathcal{U}(\mathbf{r}^N; \lambda) = \mathcal{U}(\mathbf{r}_0^N) + (1 - \lambda) \left[\mathcal{U}(\mathbf{r}^N) - \mathcal{U}(\mathbf{r}_0^N) \right] + \lambda \sum_{i=1}^N \alpha_i (\mathbf{r}_i - \mathbf{r}_{0,i})^2, \quad (9.2.1)$$

where $\mathbf{r}_{0,i}$ is the lattice position of atom i and $\mathcal{U}(\mathbf{r}_0^N)$ is the static contribution to the potential energy (i.e., the potential energy of a crystal with all atoms at their lattice positions), λ is the switching parameter, and α_i is the Einstein-crystal spring constant coupling atom i to its lattice site. Note that for $\lambda = 0$ we recover the original interactions; for $\lambda = 1$, we have switched off the intramolecular interactions completely (except for the constant static term) and the system behaves like an ideal (noninteracting) Einstein crystal. The free energy difference is calculated using Eq. (8.4.8):

$$\begin{aligned} F &= F_{\text{Ein}} + \int_{\lambda=1}^{\lambda=0} d\lambda \left\langle \frac{\partial \mathcal{U}(\lambda)}{\partial \lambda} \right\rangle_\lambda \\ &= F_{\text{Ein}} + \int_{\lambda=1}^{\lambda=0} d\lambda \left\langle \sum_{i=1}^N \alpha_i (\mathbf{r}_i - \mathbf{r}_{0,i})^2 - [\mathcal{U}(\mathbf{r}^N) - \mathcal{U}(\mathbf{r}_0^N)] \right\rangle_\lambda. \end{aligned} \quad (9.2.2)$$

The configurational free energy of the noninteracting Einstein crystal is given by

$$F_{\text{Ein}} = \mathcal{U}(\mathbf{r}_0^N) - \frac{d}{2\beta} \sum_{i=1}^N \ln(\pi/\alpha_i \beta). \quad (9.2.3)$$

As we shall see later, it is computationally more convenient to consider a crystal with fixed center of mass. This will result in a slight modification of Eq. (9.2.3) (see section 9.2.5). The “spring constants” α_i can be adjusted to optimize the accuracy of the numerical integration of Eq. (9.2.2). It is reasonable to assume that the integration is optimal if the fluctuations of the quantity $\sum_{i=1}^N \alpha_i (\mathbf{r}_i - \mathbf{r}_{0,i})^2 - \mathcal{U}(\mathbf{r}^N)$ are minimal, which implies that the interactions in the pure Einstein crystal should differ as little as possible from those in the original system. This suggests that α_i should be chosen such that the mean-squared displacements for $\lambda = 1$ and $\lambda = 0$ are equal:

$$\left\langle \sum_{i=1}^N (\mathbf{r}_i - \mathbf{r}_{0,i})^2 \right\rangle_{\lambda=0} \approx \left\langle \sum_{i=1}^N (\mathbf{r}_i - \mathbf{r}_{0,i})^2 \right\rangle_{\lambda=1}.$$

Using the expression for the mean-squared displacement in an Einstein crystal (9.2.10) we find the following condition for α :

$$\frac{3}{2\beta\alpha_i} = \left\langle (\mathbf{r}_i - \mathbf{r}_{0,i})^2 \right\rangle_{\lambda=0}. \quad (9.2.4)$$

For systems with diverging short-range repulsive interactions, such as, for instance, the Lennard-Jones potential, the integrand in Eq. (9.2.2) will exhibit a weak, integrable divergence. This divergence is due to the fact that the potential energy function of the Einstein crystal does not completely exclude configurations where two particles have the same center-of-mass coordinates.

The amplitude of the diverging contribution can be strongly suppressed by increasing the value of the α 's. However, in order to increase the accuracy of the calculation, it is better to perform the thermodynamic integration for $\lambda = 0$ to $\lambda = 1 - \Delta\lambda$, and compute the free energy difference between the states at $\lambda = 1$ and at $\lambda = 1 - \Delta\lambda$ using the perturbation expression $\Delta F = -k_B T \ln(\exp(-\beta \Delta \mathcal{U}))$, where $\Delta \mathcal{U} \equiv \mathcal{U}(\mathbf{r}^N; \lambda = 1) - \mathcal{U}(\mathbf{r}^N; \lambda = \Delta\lambda)$. The precise value of $\Delta\lambda$ is unimportant, but if it is chosen too large, the perturbation expression becomes inaccurate, and if it is chosen too small, the numerical quadrature used in the Hamiltonian integration becomes less accurate. We get back to this point in section 9.2.2.

Other approaches

Of course the choice of the Einstein crystal as a reference state is a matter of choice. We can also use other reference states for which the free-energy is

known analytically. The most obvious choice is the (classical) harmonic crystal, i.e., a model crystal for which the potential energy is only expanded to quadratic order in the displacements of the particles with respect to their lattice position [335,409]. From our knowledge of the Hessian at the potential minimum of the crystal, we can obtain all non-zero eigenfrequencies ω_i ($i = 1, \dots, d(N - 1)$) of the harmonic phonon modes. The free energy $F_h(N, V, T)$ of the (fixed-center-of-mass) harmonic crystal is given by

$$\beta F_h(N, V, T) = \beta \mathcal{U}_0 + \sum_{i=1}^{d(N-1)} \ln(\beta \hbar \omega_i), \quad (9.2.5)$$

where \mathcal{U}_0 is the value of the potential energy at its minimum.⁵

The difference between the average excess potential energy of the interacting crystal equals $\mathcal{U}_{\text{exc}}(N, V, T) = \mathcal{U}(N, V, T) - \mathcal{U}(N, V, T = 0)$. For the harmonic crystal, $\mathcal{U}_h^{\text{exc}}(N, V, T) = d(N - 1)k_B T / 2$.

At a low enough temperature T_L , the crystal with the full intermolecular interactions will (usually) become increasingly harmonic, and hence very similar to the harmonic crystal. Cheng and Ceriotti [409] proposed that when $\Delta \mathcal{U} = \mathcal{U}^{\text{exc}}(N, V, T_L) - \mathcal{U}_h^{\text{exc}}(N, V, T_L) = \mathcal{O}(k_B T_L)$, we can use the thermodynamic perturbation expression Eqn. 8.6.10 to obtain the free energy of the interacting crystal:

$$\beta \Delta F = -\ln \langle \exp(-\beta \Delta \mathcal{U}) \rangle_h.$$

Note that, as before with the Einstein crystal, the use of a perturbation expression near the harmonic limit, eliminates the divergence that would have been present if Hamiltonian [Thermodynamic Integration \(TI\)](#) would have been used.

Note, however, that \mathcal{U} is an extensive quantity, hence the larger systems, the lower T_L will have to be chosen. For this reason, it may be necessary to perform Hamiltonian TI at a higher temperature —like in the Einstein-crystal limit.

We stress that using MD for sampling the potential energy of almost harmonic solids requires the use of thermostats (see Chapter 7) that are guaranteed to be ergodic. Hence, the simple Nosé-Hoover thermostat should definitely be avoided. Note, though, that it is never necessary to simulate the harmonic (or Einstein) limit: as we know the Gaussian distribution of the particle or phonon displacements, we can generate uncorrelated configurations in the harmonic/Einstein limit, using the Box-Muller method [66].

Not all solids structures are mechanically stable at very low temperatures, meaning that the Hessian corresponding to the high-temperature lattice structure need not be positive-definite. In that case, the Einstein-crystal method is more robust.

⁵ Expression (9.2.5) for the harmonic free energy includes the contribution due to the momenta and \hbar explicitly. However, \hbar is irrelevant: for all free-energy differences between different phases, all factors \hbar cancel. So, still: classical phase equilibria do *not* depend on \hbar .

Integrating around the critical point

Although the transition from solid to liquid always involves a phase transition that, in three dimensions, is always first-order, it is possible to avoid this phase transition altogether by applying an artificial field that stabilizes the structure of the crystal. When applied to the liquid phase, this field breaks the isotropic symmetry. Hence, there is no symmetry difference between the liquid and the crystal in the presence of this field. For weak fields, there is still a first-order phase transition between a solid-like and a liquid-like phase, but for strong enough fields the phase transition ends at a critical point, and we can go continuously from liquid to crystal, in the same way, that we can link liquid and vapor by a continuous path around the critical point [399]. The same approach can be used to construct a reversible path around a first-order phase transition involving liquid-crystalline phases, such as the isotropic-nematic transition and the nematic-smectic transition [303].

Alternatives reference states

There are many ways to prepare a reference solid with known free energy. The Einstein crystal or the harmonic solid are but two examples. Sometimes it is convenient to use a reference solid that does allow multiple particles to share the same lattice site: such behavior is relevant in the study of cluster solids [410]. An Einstein-crystal-like method has even been used to compute the free energy of disordered phases [411].

Free-energy differences between different solid phases

Often one is interested in knowing the relative stability of two crystalline phases. In that case, it suffices to compute the free energy *difference* between those phases. It is then not always necessary to compute the free energy of the individual phases: if it is possible to transform one solid reversibly into the other, we can perform a thermodynamic integration along that path.

A rather different method that can be used to measure free-energy differences between different solid phases is the *Lattice Switch* MC method of Bruce et al. [412]. This method is discussed in Example 12.

9.2.2 Atomic solids with discontinuous potentials

In the preceding examples, we have considered reference states where the intermolecular interactions were switched off gradually, or where a harmonic perturbation expression was used to compute the free energy difference between a harmonic crystal and a crystal with the full interaction potential.

However, such an approach is not possible if the model under study has hard-core interactions: to be more precise, an infinitely strong repulsion cannot be switched off using a linear coupling parameter. Moreover, for a hard-core crystal, the harmonic reference state does not exist.

One way to treat such systems with hard a core interaction \mathcal{U}_0 , is simply *not* to switch it off but to make it harmless. One possible solution would be to use a method where the Einstein lattice is expanded to very low densities, where the hard-core overlaps become extremely unlikely (the same method has been used for molecular crystals [404,407]).

An alternative is to consider a system where we can switch on the spring constants, while leaving the hard-core interactions between the particles unaffected:

$$\mathcal{U}(\lambda) = \mathcal{U}_0 + \lambda \mathcal{U} = \mathcal{U}_0 + \lambda \sum_{i=1}^N (\mathbf{r}_i - \mathbf{r}_{0,i})^2, \quad (9.2.6)$$

where N denotes the total number of particles and $\mathbf{r}_{0,i}$ the position of the lattice site to which particle i is assigned. The free energy difference between the system with coupling λ and the hard-sphere fluid is then

$$F_{\text{HS}} = F(\lambda_{\max}) - \int_0^{\lambda_{\max}} d\lambda \left\langle \mathcal{U}(\mathbf{r}^N, \lambda) \right\rangle_\lambda. \quad (9.2.7)$$

At sufficiently high values of λ_{\max} , the hard particles do not “feel” each other and the free energy reduces to that of a noninteracting Einstein crystal. Clearly, the value of the spring constant λ should be sufficiently large to ensure that the harmonically bound crystal is indeed behaving as an Einstein crystal. At the same time, λ should not be too large, because this would make the numerical integration of Eq. (9.2.7) less accurate. In general, the choice of the optimal value for λ depends on the details of the model. In Case Study 17, we show how to choose λ for a particular model system and we discuss other practical issues that are specific to hard-core interactions.

9.2.3 Molecular and multi-component crystals

Molecular crystals

In contrast to atoms, the interactions between molecules depend on their relative orientation. Molecules come in a wide range of shapes, and typically these shapes can be packed into a periodic structure in many different ways. But it is not just packing that matters: the energetic interactions of molecules depend on their local charge distributions and sometimes also on their ability to form hydrogen bonds. Which molecular-crystal *polymorphs* form upon solidification is therefore the result of an interplay between many different factors. Even a simple molecule such as nitrogen has at least seven different solid phases [413,414]. It is in this context that free energy calculations of molecular solids become important: it is clearly of interest to know which polymorph has the lowest Gibbs free energy, at a given temperature and pressure. In this section, we discuss how free-energy calculations of molecular crystals can differ from this of atomic crystals.

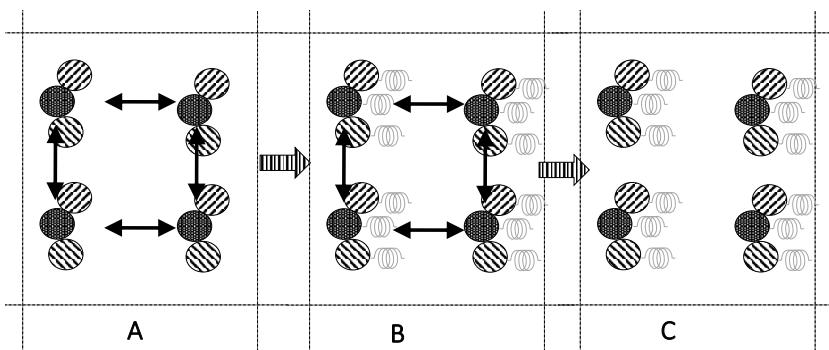


FIGURE 9.1 Sketch of the use of the Einstein Crystal method for molecular crystals. The simulation uses a sequence of Hamiltonian thermodynamic integrations to compute the free-energy difference between a molecular crystal (panel A) and an “Einstein” crystal where both the orientation of the molecules and their center of mass position are constrained by harmonic springs (panel C). Panel A shows the original crystal; some of the intermolecular interactions are shown as double arrows. In the transformation from Panel A to Panel B, the Einstein harmonic forces are switched on, while the intermolecular forces are still on. Finally, in the transformation from B to C, the intermolecular forces (but not the intramolecular forces) are switched off. Computing the free energy of the molecular Einstein crystal is a bit more complex than for an atomic Einstein crystal, as the normal-mode frequencies of the harmonically bound crystal must be determined.

However, a word of caution is in place: even if we know which polymorph is the most stable, there is no guarantee that this is the form that will form in an experiment. As was already noted by Ostwald in 1897 [415], the polymorph that crystallizes is usually not the one that is thermodynamically most stable.

Conceptually, the calculation of the free energy of a molecular crystal is no different from that of an atomic crystal: we need to select a suitable reference state, and we need to find a reversible Hamiltonian thermodynamic integration path. However, in practice, more steps are involved, depending on the nature of the reference state. In particular, we need to impose both the translational and (possibly) the orientational order of the liquid crystal.

One way to create an orientationally ordered reference state is to couple the molecular orientation to an external aligning field [416]. However, it is also possible to connect several atoms in the molecules with harmonic springs to their average lattice positions [405] (see Fig. 9.1). An example of an alternative approach to compute the free energy of a nitrogen crystal is given in Example 33.

Multi-component crystals

Free-energy calculations for ordered crystals that contain more than one species per unit cell are not fundamentally different from those for one-component crystals [417–419].

The situation is more subtle in the case of mixed crystals, such as alloys with substitutional disorder. In such cases, we need to sample over different re-

alizations of the substitutional disorder. If the sizes of the various components in the mixed solid are not too different, we can use Monte Carlo “swap moves” to sample over configurations [101]. Note that we cannot simply average over random permutations of the different components, because even in substitutionally disordered solids, there exist local correlations between the positions of the different components.

The easiest way to compute the free energy of a crystal with substitutional disorder is to compute first the free-energy change associated with the transformation of a mixed crystal to a pure crystal. There are several ways of doing this (see e.g. [101,210] and Example 12), involving the transformation of all particles to the same type, and computing the free-energy change in the process. Once we have a pure crystal, we can use the standard methods discussed above to compute the free energy.

However, it may be that no reversible path exists from the substitutional alloy to a one-component crystal. In that case, it is better to use a free energy calculation method that is compatible with swap moves. One can, for instance, in the spirit of ref. [410] switch on a periodic potential that is compatible with the lattice structure, such that, for a sufficiently strong field, all particles are forced to be close to a lattice position, be it that unlike ref. [410], double occupancy of lattice sites will typically be excluded. This field should be “color-blind”: hence it does not matter which species is at what lattice site. Once the field is strong enough, we can switch off the intermolecular forces and we have reduced the system to a state for which we can compute the free energy analytically (or almost analytically) - in this limit, the swap moves will sample a randomly substituted alloy.

9.2.4 Einstein-crystal implementation issues

If all particles are coupled to the Einstein lattice, the crystal as a whole does not move. However, in the limit $\lambda \rightarrow 0$, there is no penalty for moving the particles away from their “Einstein” lattice position. As a consequence, the crystal as a whole may start to drift and the mean-squared particle displacement $\langle r^2 \rangle$ becomes on the order L^2 . If this happens, the integrand in Eq. (9.2.7) becomes sharply peaked around $\lambda = 0$. This would seem to imply that the numerical integration of Eq. (9.2.7) requires many simulations for low values of λ . This problem can be avoided if we perform the simulation under the constraint that the center of mass of the solid remains fixed. In this case, $\langle r^2 \rangle$ tends to $\langle r^2 \rangle_0$, the mean-squared displacement of a particle from its lattice site in the normal (i.e., interacting) crystal.⁶

⁶ The divergence of the mean-square displacement at low values of λ can also be achieved by fixing the center of mass of one of the particles in the system. The net result of imposing such an “Einstein-Molecule” constraint [420] is similar, though not quite identical to fixing the center of mass of the system as a whole.

To perform a Monte Carlo simulation under the constraint of a fixed center of mass we have to ensure that, if a particle is given a random displacement, all particles are subsequently shifted in the opposite direction such that the center of mass remains fixed. In practice, it is not very convenient to keep the center of mass in place by moving all particles every time a single-particle trial move is carried out. Rather, we update the center-of-mass position every time a single-particle trial move is accepted. We need to correct for the shift of the center of mass only when computing the potential energy of the harmonic springs connecting the particles to their lattice sites. In contrast, the calculation of the intermolecular potential can be carried out without knowledge of the position of the center of mass, as a shift of the center of mass does not change the distance *between* particles.

It is convenient to distinguish between the “absolute” coordinates (\mathbf{r}) of a particle (i.e., those that have been corrected for center-of-mass motion) and the uncorrected coordinates ($\mathbf{r}^{(U)}$). When computing the potential energy of the harmonic springs, we need to know $\sum_{i=1}^N (\mathbf{r}_i - \mathbf{r}_{0,i})^2$. To compute the distance of a particle i to its lattice site, $\mathbf{r}_i - \mathbf{r}_{0,i}$, we must keep track of the shift of the center of mass:

$$\Delta\mathbf{r}_i \equiv \mathbf{r}_i - \mathbf{r}_{0,i} = \mathbf{r}_i^{(U)} - \mathbf{r}_{0,i}^{(U)} - \Delta\mathbf{R}_{CM},$$

where $\Delta\mathbf{R}_{CM}$ denotes the accumulated shift of the center of mass of the system. Every time a particle is moved from $\mathbf{r}^{(U)} \rightarrow \mathbf{r}^{(U)} + \Delta\mathbf{r}$, $\Delta\mathbf{R}_{CM}$ changes to $\Delta\mathbf{R}_{CM} + \Delta\mathbf{r}/N$.

The computation of the change in energy of the harmonic interaction between all particles and their lattice site is quite trivial. Suppose that we attempt to move particle i that is at a distance $\Delta\mathbf{r}_i$ from its lattice site $r_{0,i}$, by an amount Δ_i . This causes a shift Δ_i/N in the center of mass. The change in the harmonic potential energy is

$$\begin{aligned}\Delta\mathcal{U}_{Harm}(\lambda) &= \lambda \sum_{j \neq i}^N \left[(\Delta\mathbf{r}_j - \Delta_i/N)^2 - \Delta\mathbf{r}_j^2 \right] \\ &\quad + \lambda \left[(\Delta\mathbf{r}_i + (1 - 1/N)\Delta_i)^2 - \Delta\mathbf{r}_i^2 \right] \\ &= \lambda \left(2\Delta\mathbf{r}_i \cdot \Delta_i + \frac{N-1}{N} \Delta_i^2 \right),\end{aligned}\tag{9.2.8}$$

where, in the last line, we used the fact that $\sum_{i=1}^N \Delta\mathbf{r}_i = 0$.

One more caveat should be considered: it is common (though not advisable) practice to put a particle that moves out of the original simulation box, back in at the other side. However, when simulating a system with a fixed center of mass, moving a particle back into the original simulation box creates a discontinuous change in the position of the center of mass and hence a sudden change in the

Algorithm 19 (Fixed-CoM MC of crystal bound to reference lattice)

function mcmove	
setlat	attempts to move a particle keeping the center of mass fixed
o=int(\mathcal{R} *npart)+1	set up the reference lattice
dis=(\mathcal{R} -0.5)*delx	select particle at random
xn=x(o)+dis	give particle random displ.
dx=x(o)-x0(o)-dxcm	
del=lambda*(2*dx*dis+	
+ dis*dis*(npart-1)/npart)	calculate $\Delta\mathbf{r}_i$
arg1=-beta*del	energy difference with lattice
if $\mathcal{R} < \exp(arg1)$ then	Eq. (9.2.8)
eno = ener(x(o))	energy old configuration
enn = ener(xn)	energy new configuration
arg2=-beta*(enn-eno)	
if $\mathcal{R} < \exp(arg2)$ then	
dxcm=dxcm+(xn-x(o))/npart	new shift center of mass
x(o)=xn	accepted: replace $x(o)$ by xn
endif	
endif	
end function	

Specific Comments (for general comments, see p. 7)

1. The function `setlat` has been used to set up the fixed reference lattice $x0$ (Algorithm 20). At the beginning of the simulation $x=x0$. `ener` calculates the intermolecular interaction energy.
2. If a move is accepted, the shift of the center of mass (CoM) of the system ($dxcm$) is updated, and the same shift is applied to the reference lattice.
3. The term λ (`lambda`) is the coupling constant as defined in Eq. (9.2.6) and $dxcm = \Delta\mathbf{R}_{CM}$ is the accumulated shift of the center of mass.
4. For hard-core systems, it is important to compute first the Boltzmann factor associated with the potential-energy change of the harmonic springs and apply the Metropolis rule to see if the move should be rejected. Only if this test is passed should we attempt to perform the more expensive test for overlaps.

energy of the Einstein lattice. Therefore, in a simulation with a fixed center of mass, particles that move out of the original simulation box should definitely *not* be put back in. Algorithms 19 and 20 sketch how the Einstein-crystal method is implemented in a Monte Carlo simulation.

Algorithm 20 (Generate fcc crystal)

```

function setlat(nx,ny,nz)
    generates 3d fcc-crystal
    of nx*ny*nz unit cells
    each containing 4 particles
    a1: unit-cell diameter

    a1=(vol/(nx*ny*nz))**(1/3)
    i=0
    xcm0=0
    for 1 ≤ iz ≤ 2*nz do
        for 1 ≤ iy ≤ 2*ny do
            for 1 ≤ ix ≤ 2*nx do
                if (ix+iy+iz)%2 == 0 then
                    i=i+1
                    x0(i)=a0*ix+
                    + 0.5*a0*(iy+iz)%2
                    y0(i)=a0*iy+
                    + 0.5*a0*(ix+iz)%2
                    z0(i)=a0*iz+
                    + 0.5*a0*(ix+iy)%2
                    xcm0=xcm0+x0(i)
                    y and z similar
                endif
            enddo
        enddo
    enddo
    xcm0=xcm0/npart
    x center of mass; y and z similar
end function

```

Specific Comments (for general comments, see p. 7)

1. This algorithm generates face-centered cubic (fcc) lattice and calculates the position of its center of mass (here only the x-component is shown).
2. Note that, in a periodic system, the center of mass is ill-defined: here we take the center of mass of the particles in the original simulation box.
3. When following the displacement of the center of mass (see Algorithm 19), we compute the average (possibly mass-weighted) displacement of all particles: in that case; note that we should not force particles to be inside the original simulation box.

Example 19 (Solid-liquid equilibrium of hard spheres). In this Example, we locate the solid-liquid coexistence densities of the hard-sphere model. We determine these densities by equating the chemical potential and the pressure of the two phases.

For the liquid phase, we use the equation of state of Speedy [421], which is based on a Padé approximation to simulation data on both the equation of

state and the virial coefficients of hard spheres:

$$z_{\text{liquid}} = \frac{P\beta}{\rho} = 1 + \frac{x + 0.076014x^2 + 0.019480x^3}{1 - 0.548986x + 0.075647x^2}.$$

For the solid phase of the hard-sphere model, Speedy proposed the following equation of state [320]:

$$z_{\text{solid}} = \frac{3}{1 - \rho^*} - 0.5921 \frac{\rho^* - 0.7072}{\rho^* - 0.601}, \quad (9.2.9)$$

where $\rho^* = \sigma^3 \rho / \sqrt{2}$. In Fig. 9.2, we compare the predictions of this equation of state for the liquid and solid phases with the results from the computer simulations of Alder and Wainwright [422] and Adams [171]. As can be seen, the empirical equations of state reproduce the simulation data quite well. To calculate the chemical potential of the liquid phase, we integrate the equation of state (see (9.1.1)) starting from the dilute gas limit. This yields the Helmholtz free energy as a function of the density. The chemical potential then follows from

$$\beta\mu(\rho) = \frac{\beta G}{N} = \frac{\beta F}{N} + \frac{P}{\rho k_B T}.$$

The free energy per particle of the ideal gas is given by

$$\beta f^{\text{id}}(\rho) = \frac{F^{\text{id}}(\rho)}{Nk_B T} = \ln \rho \Delta^3 - 1,$$

where Δ is the de Broglie thermal wavelength. In what follows, we shall write

$$\beta f^{\text{id}}(\rho) = \ln \rho - 1.$$

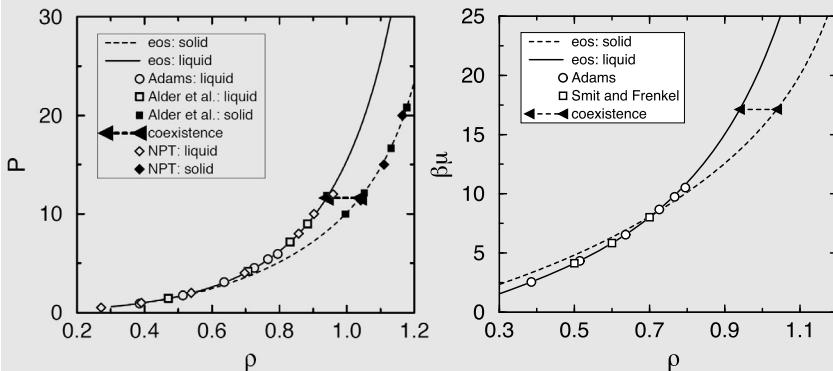


FIGURE 9.2 Pressure P (left) and chemical potential μ (right) as a function of the density ρ . The solid curves, showing the pressure and chemical potential of the liquid phase, are obtained from the equation of state of Speedy [421]. The dashed curve gives the pressure of the solid phase as calculated from the equation of state of ref. [320]. The open and filled symbols are the results of computer simulations for the liquid [171,422,423] and solid phases [422], respectively. The coexistence densities are indicated with horizontal lines.

That is, we shall work with the usual reduced densities and ignore the additive constant $3 \ln(\Lambda/\sigma)$, as it plays no role in the location of phase equilibria for classical systems.

Fig. 9.2 compares the chemical potential that follows from the Hall equation of state with some of the available simulation data (namely, grand-canonical ensemble simulations of [171] and direct calculations of the chemical potential using the Widom test-particle method [423] (see Chapter 8)).

These results show that we have an accurate equation of state for the liquid phase and the solid phase. Since we know the absolute free energy of the ideal gas phase, we can calculate the free energy and hence the chemical potential of the liquid phase. For the solid phase we can use the equation of state to calculate only free energy differences; to calculate the absolute free energy, we have to determine the free energy at a particular density. To perform this calculation, we use the lattice coupling method.

We must now select the upper limit of the coupling parameter λ (λ_{\max}) and the values of λ for which we perform the simulation. For sufficiently large values of λ we can calculate $\sum_{i=1}^N (\mathbf{r}_i - \mathbf{r}_{0,i})^2$ analytically, using

$$\langle r^2 \rangle_\lambda = \frac{1}{N} \frac{\partial F(\lambda)}{\partial \lambda}.$$

For the noninteracting Einstein crystal, the mean-squared displacement is given by

$$\langle r^2 \rangle_\lambda = \frac{3}{2\beta\lambda}. \quad (9.2.10)$$

For a noninteracting Einstein crystal with fixed center of mass, the free energy is given by Eq. (9.2.23), which gives

$$\langle r^2 \rangle_{\text{Ein},\lambda} = \frac{1}{\beta} \frac{3}{2} \frac{N-1}{N} \frac{1}{\lambda}. \quad (9.2.11)$$

In [314] an analytical expression is derived for the case of an interacting Einstein crystal, which reads

$$\begin{aligned} \langle r^2 \rangle_\lambda &= \langle r^2 \rangle_{\text{Ein},\lambda} - \frac{\beta n}{2} \frac{1}{2a(2\pi\beta\lambda)^{(1/2)} \left(1 - \langle P_{\text{overlap}}^{\text{nn}} \rangle_\lambda\right)} \\ &\times \left\{ [\sigma a - \sigma^2 - 1/(\beta\lambda)] \exp[-\beta\lambda(a-\sigma)^2/2] \right. \\ &\left. + [\sigma a + \sigma^2 - 1/(\beta\lambda)] \exp[-\beta\lambda(a+\sigma)^2/2] \right\}, \end{aligned} \quad (9.2.12)$$

where a is the separation of two nearest neighbors i and j , $a = \mathbf{r}_{0,i} - \mathbf{r}_{0,j}$, σ is the hard-core diameter, and n is the number of nearest neighbors (for example, $n = 12$ for fcc (face-centered cubic) and hcp (hexagonal close-packed) solids or $n = 8$ for bcc (body-centered cubic)); $\langle P_{\text{overlap}}^{\text{nn}} \rangle_\lambda$ is the probability that two nearest neighbors overlap. Such probability is given by

$$\begin{aligned} \langle P_{\text{overlap}}^{\text{nn}} \rangle_{\lambda} = & \frac{\operatorname{erf}\left[(\beta\lambda/2)^{1/2}(\sigma+a)\right] + \operatorname{erf}\left[(\beta\lambda/2)^{1/2}(\sigma-a)\right]}{2} \\ & - \frac{\exp[-\beta\lambda(\sigma-a)^2/2] - \exp[-\beta\lambda(\sigma+a)^2/2]}{(2\pi\beta\lambda)^{1/2}a}. \end{aligned} \quad (9.2.13)$$

This equation can also be used to correct the free energy of a noninteracting Einstein crystal (9.2.23):

$$\frac{\beta F_{\text{Ein}}(\lambda)}{N} = \frac{\beta F_{\text{Ein}}}{N} + \frac{n}{2} \ln \left(1 - \langle P_{\text{overlap}}^{\text{nn}} \rangle_{\lambda} \right). \quad (9.2.14)$$

We choose λ_{\max} such that, for values of λ larger than this maximum value, $\langle r^2 \rangle_{\lambda}$ obeys the analytical expression. Typically, this means that the probability of overlap of two harmonically bound particles should be considerably less than 1%. The results of these simulations are presented in Fig. 9.3. This figure shows that if we rely only on the analytical results of the noninteracting Einstein crystal, we have to take a value for $\lambda_{\max} \approx 1000$ –2000. If we use Eq. (9.2.12) for $\langle r^2 \rangle_{\lambda}$, $\lambda_{\max} = 500$ –1000 is sufficient.

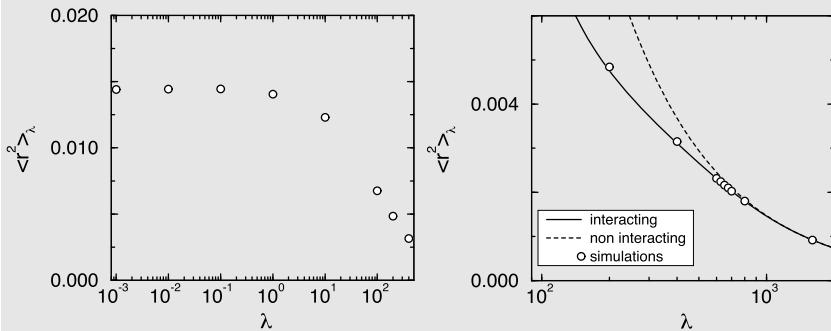


FIGURE 9.3 The mean-squared displacement $\langle r^2 \rangle_{\lambda}$ as a function of the coupling parameter λ for a hard-sphere (FCC) solid of 54 particles (6 layers of 3×3 close-packed atoms at a density $\rho = 1.04$). The figure on the left shows the simulation results for low values of λ , the figure on the right for high values. The solid line takes into account nearest-neighbor interactions (9.2.12); the dashed line assumes a noninteracting Einstein crystal (9.2.11). The open symbols are the simulation results.

We should now integrate

$$\frac{\Delta F}{N} = \int_0^{\lambda_{\max}} d\lambda \langle r^2 \rangle_{\lambda}.$$

In practice, this integration is carried out by numerical quadrature. We, therefore, must specify the values of λ for which we are going to compute $\langle r^2 \rangle_{\lambda}$. To improve the accuracy of the numerical quadrature, it is convenient to transform to another integration variable:

$$\frac{\Delta F}{N} = \int_0^{\lambda_{\max}} \frac{d\lambda}{g(\lambda)} g(\lambda) \langle r^2 \rangle_\lambda = \int_{G^{-1}(0)}^{G^{-1}(\lambda_{\max})} d[G^{-1}(\lambda)] g(\lambda) \langle r^2 \rangle_\lambda,$$

where $g(\lambda)$ is an as-yet arbitrary function of λ and $G^{-1}(\lambda)$ is the primitive of the function $1/g(\lambda)$. If we can find a function $g(\lambda)$ such that the integrand, $g(\lambda) \langle r^2 \rangle_\lambda$, is a slowly varying function, we need fewer function evaluations to arrive at an accurate estimate. To do this, we need to have an idea about the behavior of $\langle r^2 \rangle_\lambda$.

For $\lambda \rightarrow 0$, $\langle r^2 \rangle_\lambda \rightarrow \langle r^2 \rangle_0$, which is the mean-squared displacement of an atom around its lattice site in the normal hard-sphere crystal. At high values of λ , where the system behaves like an Einstein crystal, we have $\langle r^2 \rangle_\lambda \rightarrow 3k_B T/(2\lambda)$. This leads to the following guess for the functional form of $g(\lambda)$:

$$g(\lambda) \approx k_B T / \langle r^2 \rangle_\lambda \approx c + \lambda,$$

where $c = k_B T / \langle r^2 \rangle_0$. Here, $\langle r^2 \rangle_0$ can be estimated from Fig. 9.3. The value of c clearly depends on density (and temperature). For $\rho = 1.04$, extrapolation to $\lambda \rightarrow 0$ gives $\langle r^2 \rangle_0 \approx 0.014$, which gives $c = 70$. If we use this function $g(\lambda)$, the free energy difference is calculated from

$$\frac{\Delta F}{N} = \int_{\ln c}^{\ln(\lambda_{\max}+c)} d[\ln(\lambda+c)] (\lambda+c) \langle r^2 \rangle_\lambda.$$

For the numerical integration, we use a n -point Gauss-Legendre quadrature [424]. As the integrand is a smooth function, a 10-point quadrature is usually adequate. As discussed in section 9.2.5, the resulting free energy still depends (slightly) on the system size. An example of the system-size dependence of the excess free energy of a hard-sphere crystal is shown in Fig. 9.4 [425]. From this figure, we can estimate the excess free energy of the infinite system to be $\beta f^{\text{ex}} = 5.91889(4)$. This is in good agreement with the estimate of Frenkel and Ladd, $\beta f^{\text{ex}} = 5.9222$ [314].

Once we have one value of the absolute free energy of the solid phase at a given density, we can compute the chemical potential of the solid phase at any other density, using the equation of state of Speedy (see Fig. 9.2). The coexistence densities follow from the condition that the chemical potentials and pressures in the coexisting phases should be equal.

Using the value of 5.91889(4) from [425] for the solid at $\rho = 1.04086$, we arrive at a freezing density $\rho_l = 0.9391$ and a melting density $\rho_s = 1.0376$. At coexistence, the pressure is $P_{\text{coex}} = 11.567$ and the chemical potential is $\mu_{\text{coex}} = 17.071$. In fact, as we shall argue below, the presence of vacancies in the equilibrium crystal lowers the coexistence pressure slightly: $P_{\text{coex}} = 11.564$. These results are in surprisingly good agreement with the original data of Hoover and Ree [307], who obtained an estimate for the

solid-liquid coexistence densities $\rho_s = 1.041 \pm 0.004$ and $\rho_l = 0.943 \pm 0.004$ at a pressure 11.70 ± 0.18 .

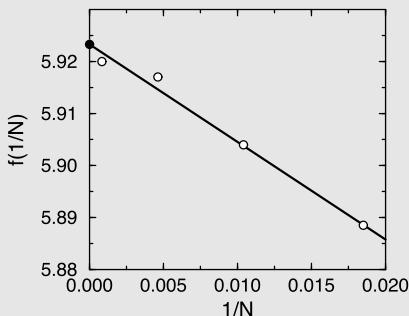


FIGURE 9.4 $\beta F^{\text{ex}}/N + \ln(N)/N$ versus $1/N$ for an fcc crystal of hard spheres at a density $\rho\sigma^3 = 1.0409$. The solid line is a linear fit to the data. The coefficient of the $1/N$ term is $-6.0(2)$, and the intercept (i.e., the infinite system limit of $\beta F^{\text{ex}}/N$) is equal to $5.91889(4)$.

The free energy difference between the FCC and HCP for large hard-sphere crystals at melting is very close to 0, but the FCC structure appears to be the more stable phase [303,412,426,427].

For more details, see SI (Case Study 17).

9.2.5 Constraints and finite-size effects

The constraint that the center of mass of the system is fixed eliminates a number of degrees of freedom from the system, and this has an effect on the free energy. Strictly speaking, the change in free energy due to any hard constraint is infinite. However, as we shall always consider *differences* in free energy, the infinities drop out. The remaining change in the free energy becomes negligible in the thermodynamic limit. However, as simulations are necessarily performed on finite systems, it is important to have an estimate of the magnitude of the finite size effects. Below, we describe in some detail how the free energy of an unconstrained crystal is computed using simulations of a system with a fixed center of mass. To keep the discussion general, we will consider a d -dimensional crystal system of N_{mol} molecules composed of a total of N atoms. The partition function for the unconstrained solid is given by

$$Q = c_N \int d\mathbf{r}^{dN} d\mathbf{p}^{dN} \exp[-\beta \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)], \quad (9.2.15)$$

where $c_N = (h^{dN_{\text{mol}}} N_1! N_2! \dots N_m!)^{-1}$, where N_1 denotes the number of indistinguishable particles of type 1, N_2 the number of particles of type 2, etc., and $N_1 + N_2 + \dots + N_m = N_{\text{mol}}$. In all calculations of phase equilibria between systems that obey classical statistical mechanics, Planck's constant h drops out

of the result. Hence, in what follows, we omit all factors h . As discussed in ref. [428], one can write the partition function Q^{con} of a constrained system as

$$Q^{\text{con}} = c_N \int d\mathbf{r}^{dN} d\mathbf{p}^{dN} \exp[-\beta \mathcal{H}(\mathbf{r}_i, \mathbf{p}_i)] \\ \times \delta[\boldsymbol{\sigma}(\mathbf{r})] \delta(\mathbf{G}^{-1} \cdot \dot{\boldsymbol{\sigma}}), \quad (9.2.16)$$

where $\boldsymbol{\sigma}(\mathbf{r})$ and $\dot{\boldsymbol{\sigma}}$ are the constraints and time derivatives of the constraints, respectively, and

$$G_{kl} = \sum_{i=1}^N \frac{1}{m_i} \nabla_{\mathbf{r}_i} \sigma_k \cdot \nabla_{\mathbf{r}_i} \sigma_l. \quad (9.2.17)$$

In order to constrain the center of mass (CM), we take $\boldsymbol{\sigma}(\mathbf{r}) = \sum_{i=1}^N \mu_i \mathbf{r}_i$, and, thus, $\dot{\boldsymbol{\sigma}} = \sum_{i=1}^N (\mu_i/m_i) \mathbf{p}_i$, where $\mu_i \equiv m_i / \sum_i m_i$. To simplify matters, we have assumed that there are no additional internal molecular constraints, such as fixed bond lengths or bond angles.

We first consider the case of an Einstein crystal, which has a potential energy function given by

$$U_{\text{Ein}} = \frac{1}{2} \sum_{i=1}^N \alpha_i (\mathbf{r}_i - \mathbf{r}_{0,i})^2,$$

where $\mathbf{r}_{0,i}$ are the equilibrium lattice positions. Note that the particles in a crystal are associated with specific lattice points and therefore behave as if they are distinguishable —thus, $c_N = 1$ (as we omit the factor $1/h^{d(N-1)}$). It is easy to show that

$$Q_{\text{Ein}}^{\text{CM}} = Z_{\text{Ein}}^{\text{CM}} P_{\text{Ein}}^{\text{CM}}, \quad (9.2.18)$$

with

$$Z_{\text{Ein}}^{\text{CM}} = \int d\mathbf{r}^{dN} \prod_{i=1}^N \exp\left[-(\beta \alpha_i/2) r_i^2\right] \delta\left(\sum_{i=1}^N \mu_i \mathbf{r}_i\right) \quad (9.2.19)$$

and

$$P_{\text{Ein}}^{\text{CM}} = \int d\mathbf{p}^{dN} \prod_{i=1}^N \exp\left[-(\beta/2m_i) p_i^2\right] \delta\left(\sum_{i=1}^N \mathbf{p}_i\right) \\ = \left(\frac{\beta}{2\pi M}\right)^{d/2} \prod_{i=1}^N \left(\frac{2\pi m_i}{\beta}\right)^{d/2} \\ = \left(\frac{\beta}{2\pi M}\right)^{d/2} P_{\text{Ein}}, \quad (9.2.20)$$

where $M = \sum_i m_i$ and Z_{Ein} and P_{Ein} are the configurational and kinetic contributions to Q_{Ein} , the partition function of the unconstrained Einstein crystal. It then follows that

$$Q_{\text{Ein}}^{\text{CM}} = \left(\frac{\sum_i m_i}{\sum_i m_i^2 / \alpha_i} \right)^{\frac{d}{2}} \left(\beta^2 / 4\pi^2 \right)^{\frac{d}{2}} Q_{\text{Ein}}. \quad (9.2.21)$$

In fact, this expression can be further simplified if we make the specific choice $\alpha_i = \alpha m_i$. In that case,

$$Q_{\text{Ein}}^{\text{CM}} = \left(\beta^2 \alpha / 4\pi^2 \right)^{d/2} Q_{\text{Ein}}. \quad (9.2.22)$$

There is a good reason for making this choice for α_i : in this case the net force on the center of mass of the crystal, due to the harmonic springs is always zero, provided that it is zero when all particles are on their lattice sites. This makes it easier to perform MD simulations on Einstein crystals with fixed center of mass. The free-energy difference between the constrained and the unconstrained Einstein crystals is then

$$F_{\text{Ein}}^{\text{CM}} = F_{\text{Ein}} - k_B T \ln \left(\frac{\beta^2 \alpha}{4\pi^2} \right)^{d/2}. \quad (9.2.23)$$

For an arbitrary crystalline system in the absence of external forces, the partition function subject to the CM constraint is given by

$$Q^{\text{CM}} = Z^{\text{CM}} (\beta / 2\pi M)^{d/2} \prod_{i=1}^N (2\pi m_i / \beta)^{d/2}, \quad (9.2.24)$$

with

$$Z^{\text{CM}} = \int d\mathbf{r}^{dN} \exp[-\beta U(\mathbf{r}_i)] \delta \left(\sum_{i=1}^N \mu_i \mathbf{r}_i \right), \quad (9.2.25)$$

while the partition function of the unconstrained crystal is given by

$$Q = Z \prod_{i=1}^N (2\pi m_i / \beta)^{d/2}, \quad (9.2.26)$$

with

$$Z = \int d\mathbf{r}^{dN} \exp[-\beta U(\mathbf{r}_i)]. \quad (9.2.27)$$

Note that, as far as the kinetic part of the partition function is concerned, the effect of the fixed center-of-mass constraint is the same for an Einstein crystal as for an arbitrary “realistic” crystal. Using Eqs. (9.2.24) and (9.2.26), the

Helmholtz free energy difference between the constrained and unconstrained crystal is given by

$$F^{\text{CM}} = F - k_B T \ln(Z^{\text{CM}}/Z) - k_B T \ln(\beta/2\pi M)^{d/2} \quad (9.2.28)$$

We note that

$$\begin{aligned} \frac{Z^{\text{CM}}}{Z} &= \frac{\int d\mathbf{r}^{dN} \exp[-\beta U(\mathbf{r}_i)] \delta\left(\sum_i \mu_i \mathbf{r}_i\right)}{\int d\mathbf{r}^{dN} \exp[-\beta U(\mathbf{r}_i)]} \\ &= \left\langle \delta\left(\sum_i \mu_i \mathbf{r}_i\right) \right\rangle \\ &= \mathcal{P}(\mathbf{r}_{\text{CM}} = \mathbf{0}), \end{aligned} \quad (9.2.29)$$

where $\mathbf{r}_{\text{CM}} \equiv \sum_i \mu_i \mathbf{r}_i$, and $\mathcal{P}(\mathbf{r}_{\text{CM}})$ is the probability distribution function of the center of mass, \mathbf{r}_{CM} . To calculate $\mathcal{P}(\mathbf{r}_{\text{CM}})$ we exploit the fact that the probability distribution of the center of mass of the lattice is evenly distributed over a volume equal to that of the Wigner-Seitz cell⁷ of the lattice. The reason the integration over the center-of-mass coordinates is limited to a single Wigner-Seitz cell is that if the center of mass were to another Wigner-Seitz cell, we would have created a copy of the crystal that simply corresponds to another permutation of the particles. Such configurations are not to be counted as independent. It then follows that $\mathcal{P}(\mathbf{r}_{\text{CM}}) = 1/V_{\text{ws}} = N_{\text{ws}}/V$, where V_{ws} is the volume of a Wigner-Seitz cell, and N_{ws} is the number of such cells in the system. Thus, $Z^{\text{CM}}/Z = \mathcal{P}(\mathbf{r}_{\text{CM}} = \mathbf{0}) = N_{\text{ws}}/V$. In the case of one molecule per cell, this implies $Z^{\text{CM}}/Z = N_{\text{mol}}/V$, where N_{mol} is the number of molecules in the system.

In numerical free energy calculations, the actual simulation involves computing the free energy difference between the Einstein crystal and the normal crystal, both with constrained centers of mass. We denote this free energy difference by

$$\Delta F^{\text{CM}} \equiv F^{\text{CM}} - F_{\text{Ein}}^{\text{CM}}.$$

The free energy per particle of the unconstrained crystal (in units of $k_B T$) is then

$$\frac{\beta F}{N} = \frac{\beta \Delta F^{\text{CM}}}{N} + \frac{\beta F_{\text{Ein}}}{N} + \frac{\ln(N_{\text{mol}}/V)}{N} - \frac{d}{2N} \ln(\beta \alpha M/2\pi). \quad (9.2.30)$$

If we consider the special case of a system of identical atomic particles ($m_i = m$ and $N = N_{\text{mol}}$), we obtain the following:

$$\frac{\beta F}{N} = \frac{\beta \Delta F^{\text{CM}}}{N} + \frac{\beta F_{\text{Ein}}}{N} + \frac{\ln \rho}{N} - \frac{d}{2N} \ln N - \frac{d}{2N} \ln\left(\frac{\beta \alpha m}{2\pi}\right). \quad (9.2.31)$$

⁷ A Wigner-Seitz cell is constructed by drawing lines to connect a given lattice point to all nearby lattice points. At the midpoints of these lines surfaces normal to these lines are constructed. The smallest enclosed volume defines the Wigner-Seitz cell.

In practice, we usually calculate the excess free energy, $F^{\text{ex}} \equiv F - F^{\text{id}}$, where F^{id} is the ideal gas free energy. Let us, therefore, compute the finite-size corrections to the latter quantity: Given that

$$\beta F^{\text{id}}/N = -\ln[V^N(2\pi m/\beta)^{dN/2}/N!]/N,$$

we find that

$$\begin{aligned} \frac{\beta F^{\text{ex}}}{N} &= \frac{\beta \Delta F^{\text{CM}}}{N} + \frac{\beta F_{\text{Ein}}}{N} + \frac{\ln \rho}{N} - \frac{d}{2N} \ln \left(\frac{\beta \alpha m}{2\pi} \right) \\ &\quad - \frac{d+1}{2} \frac{\ln N}{N} - \ln \rho + 1 - \frac{\ln 2\pi}{2N}, \end{aligned} \quad (9.2.32)$$

where we have used the Stirling approximation:

$$\ln N! \approx N \ln N - N + (\ln 2\pi N)/2.$$

Hoover has analyzed the system-size dependence of the entropy of a classical harmonic crystal with periodic boundaries [429]. In this study, it was established that the leading finite-size correction to the free energy per particle of a harmonic crystal is equal to $k_B T \ln N/N$. Assuming that this result can be generalized to arbitrary crystals, we should expect that $\beta F^{\text{ex}}/N + (d-1) \ln N/(2N)$ will scale as N^{-1} , plus correction terms of order $\mathcal{O}(1/N^2)$. Fig. 9.4 shows the N -dependence of $\beta F^{\text{ex}}/N + (d-1) \ln N/(2N)$ for three-dimensional hard spheres. The figure clearly suggests that the remaining system-size dependence scales as $1/N$. This is a useful result because it provides us with a procedure to extrapolate free energy calculations for a finite system to the limit $N \rightarrow \infty$. For more details, see ref. [425].

Illustration 12 (FCC or HCP?). Hard-sphere crystals can occur in different crystal phases. The best known among these are the [Face Centered Cubic \(FCC\)](#) and [Hexagonal Close Packed \(HCP\)](#) structures. It is not easy to determine which phase is thermodynamically most stable. The reason is that the free energy differences between the various structures are on the order of $10^{-3} k_B T$ per particle, or less. As a consequence, the earliest numerical studies aimed at computing this free energy difference [314] were not conclusive. Subsequent studies [303,412] showed conclusively that the fcc structure is the most stable. While one of the latter simulations used the Einstein-crystal method of ref. [314], the others were based on a different approach. Here, we briefly discuss the so-called *lattice-switch Monte Carlo* method of Bruce et al. [412].

A close-packed crystal consists of hexagonally close-packed two-dimensional planes that are stacked up in the vertical direction. Assume that we construct the crystal by stacking planes. For every new plane, there are two distinct possibilities of stacking it on the previous plane in such a way that

all the atoms fit in the triangular holes between the atoms of the previous plane. Let us denote these two positions of the new plane by B and C , and the position of the original plane by A . With this notation, the FCC stacking obeys the following sequence $\dots ABCABCABC\dots$, while the HCP structure is characterized by $\dots ABABABA\dots$. In addition, many hybrid close-packed structures are possible, as long as we never stack two identical planes on top of one another (i.e., $BAA B$ is forbidden).

At any given instant, the atoms in a layer are not exactly on a lattice point. We can therefore write

$$\mathbf{r}_i = \mathbf{R}_i(\alpha) + \mathbf{u}_i,$$

where $\mathbf{R}_i(\alpha)$ is the ideal reference lattice position of particle i in structure α , where α labels the crystal structure (e.g., FCC or HCP). We can now perform a Monte Carlo simulation where, in addition to the usual particle displacement moves, we also attempt moves that do not affect the displacement vectors, \mathbf{u}_i , but that switch the reference lattice, $\mathbf{R}_i(\alpha)$, from FCC to HCP. In principle, the free energy difference between these two structures would follow directly from the relative probabilities of finding the two structures in such a Monte Carlo simulation:

$$F_{\text{hcp}} - F_{\text{fcc}} = k_B T \ln \left(\frac{P(\text{fcc})}{P(\text{hcp})} \right).$$

However, in practice, such a lattice switch has a very low acceptance probability. The usual solution for such a problem is to decompose the large trial move into many small steps, each of which has a reasonable acceptance probability. The lattice-switch method of Bruce et al. employs the multi-canonical method of Berg and Neuhaus [350]. This method is a version of the umbrella-sampling scheme described in section 8.6.6. The first step in this procedure is to define a convenient “order parameter” that connects the two states. To this end, Bruce et al. defined an overlap order parameter \mathcal{M} :

$$\mathcal{M}(\mathbf{u}^N) = M(\mathbf{u}^N, \text{fcc}) - M(\mathbf{u}^N, \text{hcp}),$$

where $M(\mathbf{u}^N, \alpha)$ is the number of pairs of hard spheres that overlap for configuration \mathbf{u}^N if the α lattice is used as a reference. For example, $M(\mathbf{u}^N, \text{hcp})$ is zero for a set of displacement vectors, \mathbf{u}^N , that do not yield a single overlap if we choose an HCP reference lattice. Of particular interest are those configurations for which $\mathcal{M}(\mathbf{u}^N) = 0$, since for these configurations lattice switches are always accepted. Let us define the biased distribution

$$P(\mathbf{u}^N, \alpha | \{\eta\}) \propto P(\mathbf{u}^N, \alpha) \exp \left\{ \eta [\mathcal{M}(\mathbf{u}^N)] \right\},$$

where $P(\mathbf{u}^N, \alpha)$ is the unweighted distribution and $\eta[\mathcal{M}(\mathbf{u}^N)]$ are the weights that have to be set. These weights should be chosen such that all relevant values of \mathcal{M} are sampled. From a given simulation one can make an estimate

of these weights and these are then subsequently used and updated in the next (longer) simulation until the desired accuracy has been achieved.

Bruce et al. [412] used this method to compute the free energy difference between the HCP and the FCC structures with a statistical error of $10^{-5} k_B T$. These calculations of Bruce et al. gave further support for the observation that the FCC structure is more stable than the HCP structure. Mau and Huse [430] showed that all hybrids of FCC and HCP stacking have a free energy higher than that of a pure FCC structure.

9.3 Vacancies and interstitials

Thus far, we have described crystals as if they were free of imperfections. However, any real crystal will contain point defects, such as vacancies and interstitials. In addition, one may find extended defects such as dislocations and grain boundaries. In equilibrium, point defects are the most common. Clearly, to have a realistic description of a crystal, it is important to have an expression for the equilibrium concentration of vacancies and interstitials, and their contribution to the free energy. This is not completely trivial, as the concept of a point defect is inextricably linked to that of a lattice site. And lattice sites lose their meaning in a disordered state. So, we should first address the question: when is it permissible to count states with a different number of lattice sites as distinct? The answer is, of course, that this is only true if these different states can be assigned to distinct volumes in phase space. This is possible if we impose that every particle in a crystal is confined to its Wigner-Seitz cell. In three-dimensional crystals, this constraint on the positions of all particles has little effect on the free energy (in contrast, in a *liquid* it is not at all permissible). Below, we derive an expression for the vacancy concentration in a crystal, following the approach first given by Bennett and Alder [431].

9.3.1 Defect free energies

The equilibrium concentration of vacancies in a crystal is usually very low. We shall therefore make the approximation that vacancies do not interact. This assumption is not as reasonable as it seems, as the interaction of vacancies through their stress fields is quite long range. The assumption that vacancies are ideal implies that $F^{(n)}$, the Helmholtz free energy of a crystal with n vacancies *at specified positions*, can be written as

$$F^{(n)} = F^{(0)} - nf_1 = Mf_0 - nf_1, \quad (9.3.1)$$

where M is the number of lattice sites of the crystal, f_0 is the free energy per particle in the defect-free crystal, and $-f_1$ is the change in free energy of a crystal due to the creation of a single vacancy at a specific lattice point. Let us now consider the effect of vacancies on the Gibbs free energy of a system

of N particles at constant pressure and temperature. First, we define g^{vac} as the variation in the Gibbs free energy of a crystal of M particles due to the introduction of a single vacancy *at a specific lattice position*

$$\begin{aligned} g^{\text{vac}} &\equiv G_{M+1,1}(N, P, T) - G_{M,0}(N, P, T) \\ &= F_{M+1,1}(V_{M+1,1}) - F_{M,0}(V_{M,0}) + P(V_{M+1,1} - V_{M,0}). \end{aligned} \quad (9.3.2)$$

In the above equation, the first subscript refers to the number of lattice sites in the system, and the second subscript to the number of vacancies. Clearly, the number of particles N is equal to the difference between the first and second subscripts. The next step is to write

$$\begin{aligned} F_{M+1,1}(V_{M+1,1}) - F_{M,0}(V_{M,0}) &= F_{M+1,1}(V_{M+1,1}) - F_{M+1,1}(V_{M+1,0}) \\ &\quad + F_{M+1,1}(V_{M+1,0}) - F_{M+1,0}(V_{M+1,0}) \\ &\quad + F_{M+1,0}(V_{M+1,0}) - F_{M,0}(V_{M,0}). \end{aligned} \quad (9.3.3)$$

The first line on the right-hand side of this equation is equal to $-P\Delta v$, where $\Delta v \equiv v^{\text{vac}} - v^{\text{part}}$ is the difference in the volume of the crystal as one particle is replaced by a vacancy, at constant pressure and constant number of lattice sites. The second line on the right-hand side is simply equal to $-f_1$, defined in Eq. (9.3.1):

$$-f_1 \equiv F_{M+1,1}(V_{M+1,0}) - F_{M+1,0}(V_{M+1,0}).$$

To rewrite the third line on the right-hand side of Eq. (9.3.3), we note that the Helmholtz free energy is extensive. We express this by introducing f_0 , the Helmholtz free energy per particle of a defect-free crystal, and writing $F_{M,0}(V_{M,0}) = Mf_0$. Obviously, $F_{M+1,0}(V_{M+1,0}) - F_{M,0}(V_{M,0}) = f_0$. Combining these three terms, we find that

$$F_{M+1,1}(V_{M+1,1}) - F_{M,0}(V_{M,0}) = -P\Delta v - f_1 + f_0. \quad (9.3.4)$$

The volume is also an extensive quantity; hence

$$V_{M,0} = \frac{M}{M+1} V_{M+1,0}.$$

It then follows that

$$\begin{aligned} P(V_{M+1,1} - V_{M,0}) &= P(V_{M+1,1} - V_{M+1,0} + V_{M+1,0} - V_{M,0}) \\ &= P(\Delta v + V/N). \end{aligned}$$

Hence, the Gibbs free energy difference, associated with the formation of a vacancy at a specific lattice site, Eq. (9.3.2), is then

$$g^{\text{vac}} = P(V_{M+1,1} - V_{M,0}) - f_1 + (\Delta v + V/N)P + f_0$$

$$\begin{aligned}
&= P(V/N) - f_1 + f_0 \\
&= (P/\rho + f_0) - f_1 \\
&= \mu_0 - f_1,
\end{aligned}$$

where we have defined $\mu_0 \equiv (P/\rho + f_0)$. Now we have to include the entropic contribution due to the distribution of n vacancies over M lattice sites. The total Gibbs free energy then becomes

$$\begin{aligned}
G &= G_0(N) + ng^{\text{vac}} + Mk_B T \left[\frac{n}{M} \ln \left(\frac{n}{M} \right) + \left(1 - \frac{n}{M} \right) \ln \left(1 - \frac{n}{M} \right) \right] \\
&\approx G_0(N) + ng^{\text{vac}} + nk_B T \ln \frac{n}{M} - nk_B T.
\end{aligned}$$

If we minimize the Gibbs free energy with respect to n , we find that

$$\langle n \rangle \approx M \exp(-\beta g^{\text{vac}}),$$

where we have ignored a small correction due to the variation of $\ln M$ with n . If we insert this value in the expression for the total Gibbs free energy, we find that

$$G = G_0(N) + \langle n \rangle g^{\text{vac}} - \langle n \rangle g^{\text{vac}} - \langle n \rangle k_B T = G_0 - \langle n \rangle k_B T.$$

The total number of particles is $M - \langle n \rangle$. Hence the Gibbs free energy *per particle* is

$$\begin{aligned}
\mu &= \frac{G_0 - \langle n \rangle k_B T}{N} = \mu_0 - \frac{\langle n \rangle k_B T}{N} \\
&\approx \mu_0 - x_v k_B T,
\end{aligned} \tag{9.3.5}$$

where we have defined $x_v \equiv n/N$. Hence the change in the chemical potential of the solid due to the presence of vacancies is

$$\Delta\mu = -x_v k_B T \tag{9.3.6}$$

from which it follows that the change in *pressure* of the solid at fixed chemical potential is equal to

$$\Delta P = x_v \rho_s k_B T. \tag{9.3.7}$$

9.3.1.1 Vacancies

Numerically, it is straightforward to compute the equilibrium vacancy concentration. The central quantity that needs to be computed is $-f_1$, the change in free energy of a crystal due to the creation of a single vacancy at a specific lattice point. In fact, it is more convenient to consider $+f_1$, the change in free energy due to the removal of a vacancy at a specific lattice point. This quantity can be computed in several ways. For instance, we could use a particle-insertion

method. We start with a crystal containing one single vacancy and attempt a trial insertion in the Wigner-Seitz cell surrounding that vacancy. Then f_1 is given by

$$f_1 = -k_B T \ln \left(\frac{V_{\text{WS}} \langle \exp(-\beta \Delta U) \rangle}{\Lambda^d} \right), \quad (9.3.8)$$

where V_{WS} is the volume of the Wigner-Seitz cell, and ΔU is the change in potential energy associated with the insertion of a trial particle. For hard particles

$$f_1 = -k_B T \ln \left(\frac{V_{\text{WS}} P_{\text{acc}}(V_{\text{WS}})}{\Lambda^d} \right),$$

where $P_{\text{acc}}(V_{\text{WS}})$ is the probability that the trial insertion in the Wigner-Seitz cell will be accepted. As most of the Wigner-Seitz cell is not accessible, and it is more efficient to attempt insertion in a subvolume (typically on the order of the cell volume in a lattice-gas model of the solid). However, then we also should consider the reverse move—the removal of a particle from a subvolume v of the Wigner-Seitz cell, in a crystal without vacancies. The only thing we need to compute in this case is $P_{\text{rem}}(v)$, the probability that a particle happens to be inside this volume. The expression for f_1 is then

$$f_1 = -k_B T \ln \left(\frac{v P_{\text{acc}}(v)}{P_{\text{rem}}(v) \Lambda^d} \right).$$

Of course, in the final expression for the vacancy concentration, the factor Λ^d drops out (as it should), because it is cancelled by the same term in the ideal part of the chemical potential. A direct calculation of the vacancy concentration [431, 432] suggests that this concentration in a hard-sphere solid near coexistence is approximately 2.6×10^{-4} . Let us assume that the defect-free crystal is in equilibrium with the liquid at a pressure P and chemical potential μ . Then it is easy to verify that the shift in the coexistence pressure due to the presence of vacancies is

$$\delta P_{\text{coex}} = \frac{-x(0)k_B T}{v_l - v_s},$$

where v_l (v_s) is the molar volume of the liquid (solid). The corresponding shift in the chemical potential at coexistence is

$$\delta \mu_{\text{coex}} = \frac{\delta P_{\text{coex}}}{\rho_l}.$$

Inserting the numerical estimate $x(0) \approx 2.6 \times 10^{-4}$, the decrease in the coexistence pressure due to vacancies is $\delta P_{\text{coex}} \approx -2.57 \times 10^{-3}$. The corresponding shift in the chemical potential at coexistence is $\delta \mu_{\text{coex}} = -2.74 \times 10^{-3}$. Note that these shifts are noticeable when compared to the accuracy of absolute free-energy calculations of the crystalline solid.

9.3.1.2 *Interstitials*

Thus far, we have ignored interstitials. However, it is not *a priori* obvious that these can be ignored. The only new ingredient in the calculation of the interstitial concentration is the determination of f_I . This is best done by thermodynamic integration. To this end, we first simulate a crystal with one interstitial. We then determine the excursions of the interstitial from its average position. Next, we define a volume v_0 such that the interstitial is (with overwhelming probability) inside this volume. The probability that a point particle inserted at random in a Wigner-Seitz cell will be inside this volume is

$$P_{\text{acc}} = \frac{v_0}{V_{\text{WS}}}. \quad (9.3.9)$$

Next, we “grow” the particle to the size of the remaining spheres. This will require a reversible work w . The latter quantity can easily be calculated, because the simulation yields the pressure exerted on the surface of this sphere. The total free energy change associated with the addition of an interstitial in a given octahedral hole is then

$$f_I = -k_B T \ln \left(P_{\text{acc}} \frac{V_{\text{WS}}}{\Lambda^3} \right) + w \quad (9.3.10)$$

and

$$x_I = \exp \left\{ -\beta \left[w - k_B T \ln \left(P_{\text{acc}} \frac{V_{\text{WS}}}{\Lambda^3} \right) - \mu \right] \right\}. \quad (9.3.11)$$

As before, the Λ^3 term drops out of the final result (as it should). For more details, see refs. [433–437].

Chapter 10

Free energy of chain molecules

In Chapter 8, we introduced the test-particle insertion scheme as a powerful method for determining chemical potentials. However, this method fails when the Boltzmann factor associated with the trial insertion becomes very small. One consequence is that the simple particle insertion method is ill-suited for computing the chemical potential of larger molecules, except at very low densities. The reason is that, to a first approximation, the probability of successful insertion of a molecule that excludes a volume v_{excl} to the solvent decays exponentially with the solvent density ρ_S and the excluded volume: $\text{acc} \sim \exp(-v_{\text{excl}}\rho_S)$.

Fortunately, it is possible to overcome this problem, at least partially, by performing non-random sampling. Here, we discuss several of the techniques that have been proposed to compute the chemical potential of chain molecules, for which the problem of random insertion is particularly severe. The methods described can, however, be applied to any composite object that can be inserted in stages, provided that, at every stage, there is some choice about where to insert the next unit.

Many approaches have been proposed to improve the efficiency of the original Widom scheme, of which we describe three that are representative of a wider class of algorithms. The most straightforward of these techniques is thermodynamic integration scheme. Next, we discuss a method based on (generalizations of) the Rosenbluth algorithm for generating polymer conformations [438]. And, finally, we mention a recursive algorithm.

10.1 Chemical potential as reversible work

The excess chemical potential of a (chain) molecule is simply the reversible work needed to add such a molecule to a liquid in which N other (possibly identical) molecules are already present. If we choose to break up the insertion of the molecule into a number of steps, then clearly, the reversible work needed to insert the whole molecule is equal to the sum of the contributions of the substeps. At this stage, we are still free to choose the elementary steps, just as we are free to choose whatever reversible path we wish when performing thermodynamic integration. One obvious possibility is to start with an ideal (noninteracting) chain molecule and then slowly switch on the interaction of this molecule with the surrounding particles (and, if necessary, also the nonbonded intramolecular interactions). This could be done in the way described in section 8.4.1. In fact, this approach was followed by Müller and Paul [439], who performed a simu-

lation in which the polymer interaction is switched on gradually. Although this simulation could have been performed with straightforward thermodynamic integration, a multiple-histogram method (see section 8.6.10) was used instead, but this does not change the overall nature of the calculation. As stated before, the advantage of thermodynamic integration (and related techniques) is that it is robust. The disadvantage is that it is no longer possible to measure the excess chemical potential in a single simulation.

A closely related method for measuring the chemical potential of a chain molecule was proposed by Kumar et al. [440,441]. In this scheme, the chain molecule is built up monomer by monomer. The method of Kumar et al. resembles the gradual insertion scheme for measuring excess chemical potentials that had been proposed by Mon and Griffiths [442]. The reversible work involved in the intermediate steps is measured using the Widom method; that is, the difference in excess free energy of a chain of length ℓ and $\ell + 1$ is measured by computing $\Delta\mathcal{U}(\ell \rightarrow \ell + 1)$, the change in potential energy associated with the addition of the $(\ell + 1)$ th monomer. The change in free energy is then given by

$$\begin{aligned}\Delta F_{\text{ex}}(\ell \rightarrow \ell + 1) &\equiv \mu_{\text{ex}}^{\text{incr}}(\ell \rightarrow \ell + 1) \\ &= -k_B T \ln \langle \exp[-\beta \Delta\mathcal{U}(\ell \rightarrow \ell + 1)] \rangle.\end{aligned}\quad (10.1.1)$$

This equation defines the incremental excess chemical potential $\mu_{\text{ex}}^{\text{incr}}(\ell \rightarrow \ell + 1)$. The excess chemical potential of the complete chain molecule is simply the sum of the individual incremental excess chemical potentials. As the latter contributions are measured using the Widom method, the scheme of Kumar et al. is referred to as *the modified Widom method*. This method is subject to the same limitations as the original Widom method (i.e., the insertion probability of the individual monomers should be appreciable). In this respect, it is less general than thermodynamic integration. As with the multiple-histogram method used by Müller and Paul [439], the computation of the excess chemical potential may require many individual simulations [441,443].

10.2 Rosenbluth sampling

Several proposals for measuring the chemical potential of a chain molecule in a *single* simulation have been made. Harris and Rice [444] and Siepmann [445] showed how to compute the chemical potential of chain molecules with discrete conformations using an algorithm to generate polymer conformations due to Rosenbluth and Rosenbluth [438]. A generalization to continuously deformable molecules was proposed by Frenkel et al. [446,447] and by de Pablo et al. [448]. As the extension of the sampling scheme from molecules with discrete conformations to continuously deformable molecules is nontrivial, we shall discuss the two cases separately. The approach followed here is closely related to the configurational-bias Monte Carlo scheme described in section 12.2.1. However, we have attempted to make the presentation self-contained.

10.2.1 Macromolecules with discrete conformations

It is instructive to recall how we compute μ_{ex} of a chain molecule with the Widom technique. To this end, we introduce the following notation: the position of the first segment of the chain molecule is denoted by \mathbf{q} and the conformation of the molecule as a whole is described by Γ . The configurational part of the partition function of a system of chain molecules can be written as¹

$$Q_{\text{chain}}(N, V, T) = \frac{1}{N!} \int d\mathbf{q}^N \sum_{\Gamma_1, \dots, \Gamma_N} \exp[-\beta \mathcal{U}(\mathbf{q}^N, \Gamma^N)]. \quad (10.2.1)$$

The excess chemical potential of a chain molecule is obtained by considering the ratio

$$\frac{Q(N+1, V, T)}{Q(N, V, T) Q_{\text{non-interacting}}(1, V, T)},$$

where the numerator is the (configurational part of) the partition function of a system of $N+1$ interacting chain molecules while the denominator is the partition function for a system consisting of N interacting chains and one chain that does not interact with the others. The latter chain plays the role of the ideal gas molecule (see section 8.5.1). Note, however, that although this molecule does not interact with any of the other molecules, it *does* interact with itself through both bonded and nonbonded interactions.

As explained in section 6.5.1, the fact that we do not know the configurational part of the partition function of an isolated self-avoiding chain *a priori* is unimportant if we work in terms of the fugacity.

However, if for some reason we wish to determine the *absolute* chemical potential of the chain molecule, it is better to use another reference state, namely that of the isolated non-self-avoiding chain (i.e., a molecule in which all non-bonded interactions have been switched off), because for such molecules we can compute the intramolecular part of the partition function analytically. But we stress that, if used consistently, the choice of the reference system makes no difference in the computation of any observable property.

Here, we start with the description of the case of a non-self-avoiding reference state, simply because it is easier to explain. We subsequently consider the case of intramolecular interactions.

Let us consider a lattice polymer that consists of ℓ segments. Starting from segment 1, we can add segment 2 in k_2 equivalent directions, and so on. For instance, for a polymer on a simple cubic lattice, there are six possible directions of the first segment and, if we choose to exclude conformations where bonds can reverse, five for all subsequent ones. Clearly, the total number of non-self-avoiding conformations is $\Omega_{\text{id}} = \prod_{i=2}^{\ell} k_i$. For convenience, we have assumed that, for a given i , all k_i directions are equally likely (i.e., we ignore gauche-trans potential energy differences). Moreover, we assume for convenience that

¹ We assume that there are no hard constraints on the intramolecular degrees of freedom.

all k_i are the same, which means that we even allow the ideal chain to retrace its steps.

These limitations are not essential but they simplify the notation (though not the computational efficiency). Hence, for the simple model that we consider, $\Omega_{\text{id}} = k^{\ell-1}$. Using this ideal chain as our reference system, the expression for the excess chemical potential becomes

$$\begin{aligned}\beta\mu_{\text{ex}} &= -\ln \left(\frac{Q_{\text{chain}}(N+1, V, T)}{Q(N, V, T) Q_{\text{ideal}}(1, V, T)} \right) \\ &= -\ln \left\langle \exp[-\beta\Delta\mathcal{U}(\mathbf{q}^N, \boldsymbol{\Gamma}^N; \mathbf{q}_{N+1}, \boldsymbol{\Gamma}_{N+1})] \right\rangle, \end{aligned}\quad (10.2.2)$$

where $\Delta\mathcal{U}$ denotes the interaction of the test chain with the N chains already present in the system *and with itself*, while $\langle \dots \rangle$ indicates averaging over all starting positions and all ideal chain conformations of a randomly inserted chain.

The problem with the Widom approach to Eq. (10.2.2) is that almost all randomly inserted ideal chain conformations will overlap either with particles already present in the system or internally. The most important contributions to μ_{ex} will come from the extremely rare cases, where the trial chain happens to be in just the right conformation to fit into the available space in the fluid. Clearly, it would be desirable if we could restrict our sampling to those conformations that satisfy this condition. If we do that, we introduce a bias in our computation of the insertion probability and we must somehow correct for that bias.

The Rosenbluth approach used in [444,445] consists of two steps: in the first step a chain conformation is generated with a bias that ensures that “acceptable” conformations are created with a high probability. The next step corrects for this bias by multiplying it by a weight factor. A scheme that generates acceptable chain conformations with a high probability was developed by Rosenbluth and Rosenbluth in the early 1950s [438]. In the Rosenbluth scheme, a conformation of a chain molecule is constructed segment by segment. For every segment, we have a choice of k possible directions. In the Rosenbluth scheme, this choice is not random but favors the direction with the largest Boltzmann factor. To be more specific, the following scheme is used to generate a conformation of one polymer with ℓ monomers:

1. The first monomer is inserted at a random position and its energy is denoted by $u^{(1)}(n)$. We define the Rosenbluth weight of this monomer as $w_1 = k \exp[-\beta u^{(1)}(n)]$.²
2. For all subsequent segments $i = 2, 3, \dots, \ell$, we consider all k trial positions adjacent to segment $i-1$ (see Fig. 10.1). The energy of the j th trial position is denoted by $u^{(i)}(j)$. From the k possibilities, we select one, say, n , with a

² The factor k is included in the definition of w_1 only to keep the notation consistent with that of section 12.2.1.

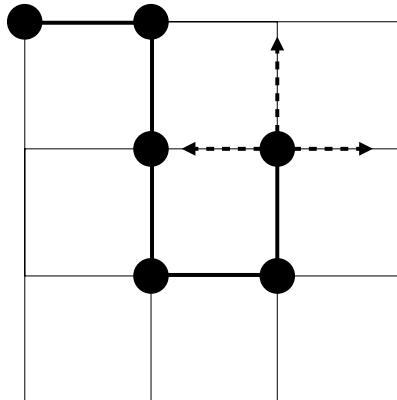


FIGURE 10.1 Rosenbluth scheme to insert a polymer segment by segment. The arrows indicate the trial positions for the next segment.

probability

$$p^{(i)}(n) = \frac{\exp[-\beta u^{(i)}(n)]}{w_i}, \quad (10.2.3)$$

where w_i is defined as

$$w_i = \sum_{j=1}^k \exp[-\beta u^{(i)}(j)]. \quad (10.2.4)$$

The energy $u^{(i)}(j)$ excludes the interactions with the subsequent segments $i+1$ to ℓ . Hence, the total energy of the chain is given by $\mathcal{U}(n) = \sum_{i=1}^{\ell} u^{(i)}(n)$.

3. Step 2 is repeated until the entire chain is grown, and we can compute the normalized Rosenbluth factor of conformation n :

$$\mathcal{W}(n) = \prod_{i=1}^{\ell} \frac{w_i}{k}. \quad (10.2.5)$$

We use this scheme to generate a large number of conformations, and ensemble-averaged properties of these chains are calculated as follows:

$$\langle A \rangle_{\mathcal{R}} = \frac{\sum_{n=1}^M \mathcal{W}(n) A(n)}{\sum_{n=1}^M \mathcal{W}(n)}, \quad (10.2.6)$$

where $\langle \dots \rangle_{\mathcal{R}}$ indicates that the conformations have been generated by the Rosenbluth scheme. This label is important, because the Rosenbluth algorithm does not generate chains with the correct Boltzmann weight. We refer to the

distribution generated with the Rosenbluth procedure as the Rosenbluth distribution. In the Rosenbluth distribution, the probability of generating a particular conformation n is given by

$$P(n) = \prod_{i=1}^{\ell} \frac{\exp[-\beta u^{(i)}(n)]}{w_i} = k^{\ell} \frac{\exp[-\beta \mathcal{U}(n)]}{\mathcal{W}(n)}. \quad (10.2.7)$$

An important property of this probability is that it is normalized; that is,

$$\sum_n P(n) = 1,$$

where the sum runs over all possible conformations of the polymer. We can recover canonical averages from the Rosenbluth distribution by attributing different weights to different chain conformations. And this is precisely what is done in Eq. (10.2.6):

$$\langle A \rangle_{\mathcal{R}} = \frac{\sum_n \mathcal{W}(n) A(n) P(n)}{\sum_n \mathcal{W}(n) P(n)}. \quad (10.2.8)$$

Substitution of Eqs. (10.2.5) and (10.2.7) gives

$$\begin{aligned} \langle A \rangle_{\mathcal{R}} &= \frac{\sum_n \mathcal{W}(n) k^{\ell} A(n) \exp[-\beta \mathcal{U}(n)] / \mathcal{W}(n)}{\sum_n \mathcal{W}(n) k^{\ell} \exp[-\beta \mathcal{U}(n)] / \mathcal{W}(n)} \\ &= \frac{\sum_n A(n) \exp[-\beta \mathcal{U}(n)]}{\sum_n \exp[-\beta \mathcal{U}(n)]} \\ &= \langle A \rangle, \end{aligned} \quad (10.2.9)$$

which shows that Eq. (10.2.6) indeed yields the correct ensemble average.

Here, we introduced the Rosenbluth factor as a correction for the bias in the sampling scheme. The Rosenbluth factor itself is also of interest, since it can be related to the excess chemical potential. To see this, let us assume that we use the Rosenbluth scheme to generate a large number of chain conformations while keeping the coordinates of all other particles in the system fixed. For this set of conformations, we compute the average of the Rosenbluth weight factor \mathcal{W} , $\overline{\mathcal{W}}$. Subsequently, we also perform an ensemble average over all coordinates and conformations of the N particles in the system, and we obtain

$$\langle \mathcal{W} \rangle = \left\langle \sum_{\Gamma} P_{\Gamma}(\mathbf{q}^N, \boldsymbol{\Gamma}^N) \mathcal{W}_{\Gamma}(\mathbf{q}^N, \boldsymbol{\Gamma}^N) \right\rangle, \quad (10.2.10)$$

where the angular brackets denote the ensemble average over all configurations of the system $\{\mathbf{q}^N, \boldsymbol{\Gamma}^N\}$ of the solvent. Note that the test polymer does not form part of the N -particle system. Therefore the probability of finding the remaining

particles in a configuration \mathbf{q}^N does not depend on the conformation Γ of the polymer.

To simplify the expression for the average in Eq. (10.2.10), we first consider the average of the Rosenbluth factor for a given configuration $\{\mathbf{q}^N, \Gamma^N\}$ of the solvent:

$$\bar{\mathcal{W}}(\{\mathbf{q}^N, \Gamma^N\}) = \sum_{\Gamma} P_{\Gamma}(\mathbf{q}^N, \Gamma^N) \mathcal{W}_{\Gamma}(\{\mathbf{q}^N, \Gamma^N\}). \quad (10.2.11)$$

Substitution of Eqs. (10.2.3) and (10.2.5) yields

$$\begin{aligned} \bar{\mathcal{W}} &= \sum_{\Gamma} \left[k \exp[-\beta u^{(1)}(\Gamma_1)] \prod_{i=2}^{\ell} \frac{\exp[-\beta u^{(i)}(\Gamma_i)]}{w_i} \right] \left[\prod_{i=1}^{\ell} \frac{w_i}{k} \right] \\ &= \sum_{\Gamma} k \exp[-\beta u^{(1)}(\Gamma_1)] \prod_{i=2}^{\ell} \frac{1}{k} \exp[-\beta u^{(i)}(\Gamma_i)] \\ &= \sum_{\Gamma} \frac{1}{k^{\ell-1}} \exp[-\beta \mathcal{U}_{\Gamma}], \end{aligned} \quad (10.2.12)$$

where we have dropped all explicit reference to the solvent coordinates $\{\mathbf{q}^N, \Gamma^N\}$. Note that Eq. (10.2.12) can be interpreted as an average over all *ideal* chain conformations of the Boltzmann factor $\exp[-\beta \mathcal{U}_{\Gamma}]$. If we now substitute Eq. (10.2.12) in Eq. (10.2.11), we obtain

$$\langle \mathcal{W} \rangle = \frac{\sum_{\Gamma} \langle \exp[-\beta \Delta \mathcal{U}(\mathbf{q}^N, \Gamma^N; \mathbf{q}_{N+1}, \Gamma_{N+1})] \rangle}{\sum_{\Gamma}}. \quad (10.2.13)$$

If we compare Eq. (10.2.13) with Eq. (10.2.2), we see that the ensemble average of the Rosenbluth factor is directly related to the excess chemical potential of the chain molecule:

$$\beta \mu_{\text{ex}} = -\ln \langle \mathcal{W} \rangle. \quad (10.2.14)$$

This completes our demonstration that a measurement of the average Rosenbluth factor of a trial chain can indeed be used to estimate the excess chemical potential of a polymer in a dense fluid. We should stress that the preceding method for measuring the chemical potential is in no way limited to chain molecules in a lattice. What *is* essential is that the number of possible directions for each segment (k) relative to the previous one is finite.

10.2.2 Extension to continuously deformable molecules

The numerical computation of the excess chemical potential of a flexible chain with or without terms in the intramolecular potential that depend on bending and

torsion angles, is rather different from the corresponding calculation for a chain molecule that has a large but fixed number of undeformable conformations.

Here, we consider the case of a flexible molecule *with* intramolecular potential energy. Fully flexible chains, of course, are included as a special case. Consider a semi-flexible chain of ℓ linear segments. The potential energy of the molecule is divided into two contributions: the “internal” potential energy $\mathcal{U}_{\text{bond}}$, which includes the bonded intramolecular interactions, and the “external” potential energy \mathcal{U}_{ext} , which accounts for the remainder of the interactions - including nonbonded intramolecular interactions. A chain in the absence of external interactions is defined as an *ideal* chain.

The conformational partition function of the ideal chain is equal to

$$Q_{\text{id}} = c \int \cdots \int d\Gamma_1 \cdots d\Gamma_\ell \prod_{i=1}^{\ell} \exp[-\beta u_{\text{bond}}(\theta_i)], \quad (10.2.15)$$

where c is a numerical constant. We assume that Q_{id} is known. Our aim is to compute the effect of the external interactions on the conformational partition function. Hence, we wish to evaluate Q/Q_{id} , where Q denotes the partition function of the interacting chain. The excess chemical potential of the interacting chain is given by

$$\mu_{\text{ex}} = -k_B T \ln(Q/Q_{\text{id}}).$$

Before considering the “smart” approach to computing μ_{ex} , let us briefly review two not-so-smart methods.

The most naive way to compute the excess chemical potential of the interacting chain is to generate a very large number of completely random conformations of the freely jointed chain. For every conformation we compute both $\exp(-\beta \mathcal{U}_{\text{bond}})$ and $\exp[-\beta (\mathcal{U}_{\text{bond}} + \mathcal{U}_{\text{ext}})]$. The average of the former quantity is proportional to Q_{id} , while the average of the latter Boltzmann factor is proportional to Q . The ratio of these two averages therefore should yield Q/Q_{id} .

The problem with this approach is that the overwhelming majority of randomly generated conformations correspond to semi-flexible chains with a very high internal energy (and therefore very small Boltzmann weights). Hence, the statistical accuracy of this sampling scheme will be very poor.

The second scheme is designed to alleviate this problem. Rather than generating conformations of a freely jointed chain, we now sample the internal angles in the chain in such a way that the probability of finding a given angle θ_i is given by the Boltzmann weight

$$P(\theta_i) = \frac{\exp[-\beta u(\theta_i)]}{\int d\Gamma_i \exp[-\beta u(\theta_i)]}.$$

Such sampling can be performed quite easily using a rejection method (see, e.g., [21]). In what follows, we use the symbol Γ_i to denote the unit vector that

specifies the orientation of the i th segment of the chain molecule. For every conformation thus generated, we compute the Boltzmann factor $\exp(-\beta\mathcal{U}_{\text{ext}})$. The average of this Boltzmann weight is then equal to

$$\langle \exp(-\beta\mathcal{U}_{\text{ext}}) \rangle = \frac{\int \prod d\Gamma \exp[-\beta(\mathcal{U}_{\text{bond}} + \mathcal{U}_{\text{ext}})]}{\int \prod d\Gamma \exp(-\beta\mathcal{U}_{\text{bond}})} = Q/Q_{\text{id}}. \quad (10.2.16)$$

This approach is obviously superior to the first scheme. However, in many practical situations it will still yield poor statistics, because most ideal chain conformations will not correspond to energetically favorable situations for the interacting chain. Hence the Boltzmann weights, again, will be small for most conformations, and the statistical accuracy will not be very good.

The problem with both these schemes is that neither allows us to focus on those conformations that should contribute most to Q , namely, those for which the sum of the internal and external potential energies is not much larger than a few $k_B T$ per degree of freedom. It would clearly be desirable to *bias* the sampling toward such favorable conformations.

It turns out that we can use a procedure similar to that used in section 10.2.1 to compute the excess chemical potential of a chain molecule with many fixed conformations. To compute μ_{ex} , we apply the following recipe for constructing a conformation of a chain of ℓ segments. The construction of chain conformations proceeds segment by segment. Let us consider the addition of one such segment. To be specific, let us assume that we have already grown i segments and we are trying to add segment $i+1$. This is done as follows:

1. Generate a fixed number of (say, k) trial segments with orientations distributed according to the Boltzmann weight associated with the internal potential energy $u(\theta)$. We denote the different trial segments by indices $1, 2, \dots, k$. Importantly, our final result for the excess chemical potential is valid for any choice of $k \geq 1$, but the accuracy of the result depends strongly on our choice for k .
2. For all k trial segments, we compute the external Boltzmann factor $\exp[-\beta u_{\text{ext}}^{(i)}(j)]$.
3. Select one of the trial segments, say, n , with a probability

$$p^{(i)}(n) = \frac{\exp[-\beta u_{\text{ext}}^{(i)}(n)]}{w_i^{\text{ext}}}, \quad (10.2.17)$$

where we have defined

$$w_i^{\text{ext}} \equiv \sum_{j=1}^k \exp[-\beta u_{\text{ext}}^{(i)}(j)].$$

4. Add this segment as segment $i + 1$ to the chain and repeat this procedure until the entire chain is completed. The normalized Rosenbluth factor \mathcal{W} of the entire chain is given by

$$\mathcal{W}^{\text{ext}}(n) = \prod_{i=1}^{\ell} \frac{w_i^{\text{ext}}}{k},$$

where, for the first segment, $w_1^{\text{ext}} = k \exp[-\beta u_{\text{ext}}^{(1)}(1)]$.

The desired ratio Q/Q_{id} is then equal to the average value (over many trial chains) of the product of the partial Rosenbluth weights:

$$Q/Q_{\text{id}} = \langle \mathcal{W}^{\text{ext}} \rangle. \quad (10.2.18)$$

To show that Eq. (10.2.18) is correct, let us consider the probability with which we generate a given chain conformation. This probability is the product of a number of factors. Let us first consider these factors for one segment and then later extend the result to the complete chain. The probability of generating a given set of k trial segments with orientations Γ_1 through Γ_k is

$$P_{\text{id}}(\Gamma_1) P_{\text{id}}(\Gamma_2) \cdots P_{\text{id}}(\Gamma_k) d\Gamma_1 \cdots d\Gamma_k. \quad (10.2.19)$$

The probability of selecting any one of these trial segments follows from Eq. (10.2.17):

$$p^{(i)}(j) = \frac{\exp[-\beta u_{\text{ext}}^{(i)}(\Gamma_j)]}{w_i^{\text{ext}}(\Gamma_1, \dots, \Gamma_k)}, \quad (10.2.20)$$

for $j = 2, 3, \dots, \ell$. We wish to compute the average of w_i^{ext} over all possible sets of trial segments and all possible choices of the segment. To this end, we must sum over all j and integrate over all orientations $\prod_{j=1}^k d\Gamma_j$ (i.e., we average over the normalized probability distribution for the orientation of segment $i + 1$):

$$\begin{aligned} \left\langle \frac{w_i^{\text{ext}}}{k} \right\rangle &= \int \prod_{j=1}^k d\Gamma_j P_{\text{id}}(\Gamma_j) \sum_{j'=1}^k \frac{\exp[-\beta u_{\text{ext}}(j')]}{w_i^{\text{ext}}(\Gamma_1, \dots, \Gamma_k)} \frac{w_i^{\text{ext}}(\Gamma_1, \dots, \Gamma_k)}{k} \\ &= \int \prod_{j=1}^k d\Gamma_j P_{\text{id}}(\Gamma_j) \sum_{j'=1}^k \frac{\exp[-\beta u_{\text{ext}}(j')]}{k}. \end{aligned} \quad (10.2.21)$$

But the labeling of the trial segments is arbitrary. Hence, all k terms in the sum in this equation yield the same contribution, and this equation simplifies to

$$\left\langle \frac{w_i^{\text{ext}}}{k} \right\rangle = \int d\Gamma P_{\text{id}}(\Gamma) \exp[-\beta u_{\text{ext}}(\Gamma)] \quad (10.2.22)$$

$$= \frac{\int d\Gamma \exp\{-\beta[u_{\text{bond}}(\Gamma) + u_{\text{ext}}(\Gamma)]\}}{\int d\Gamma \exp[-\beta u_{\text{bond}}(\Gamma)]} \quad (10.2.23)$$

$$= \frac{Q^{(i)}}{Q_{\text{id}}^{(i)}}, \quad (10.2.24)$$

which is indeed the desired result but for the fact that the expression in Eq. (10.2.24) refers to segment i (as indicated by the superscript in $Q^{(i)}$). The extension to a chain of ℓ segments is straightforward, although the intermediate expressions become a little unwieldy.

The final result is a relation between the normalized Rosenbluth factor and the excess chemical potential:

$$\beta\mu^{\text{ex}} = -\ln \frac{\langle \mathcal{W}_{\text{ID}}^{\text{ext}} \rangle}{\langle \mathcal{W}_{\text{ID}}^{\text{ext}} \rangle}, \quad (10.2.25)$$

where $\mathcal{W}_{\text{ID}}^{\text{ext}}$ is the normalized Rosenbluth factor of an isolated chain *with* non-bonded intramolecular interactions. This Rosenbluth factor has to be determined from a separate simulation using exactly the same approach: the only difference being that we now have to compute \mathcal{W} for an isolated molecule with nonbonded intra-molecular interactions.

In principle, the results of the Rosenbluth sampling scheme are exact in the sense that, in the limit of an infinitely long simulation, the results are identical to those of a Boltzmann sampling. In practice, however, there are important limitations. In contrast to the configurational-bias Monte Carlo scheme (see Chapter 12), the Rosenbluth scheme generates an unrepresentative sample of all polymer conformations as the probability of generating a given conformation is *not* proportional to its Boltzmann weight. Accurate values can be calculated only if these distributions have a sufficient overlap. If the overlap is small, then the tail of the Rosenbluth distribution makes the largest contribution to the ensemble average (10.2.6); conformations that have a very low probability of being generated in the Rosenbluth scheme may have Rosenbluth factors so large that they tend to dominate the ensemble average. Precisely because such conformations are generated very infrequently, the statistical accuracy may be poor. If the relevant conformations are never generated during a simulation, the results will even deviate systematically from the true ensemble average. This drawback of the Rosenbluth sampling scheme is well known, in fact (see, the article of Batoulis and Kremer [449,450] and Illustration 14).

Illustration 13 (Henry coefficients in porous media). For many practical applications of porous media, we need to know the “adsorption isotherm”, which describes the dependence of the number of adsorbed molecules of a given species at a given temperature on its external pressure or, more generally, on

its fugacity. Examples 4 and 18, show how a complete adsorption isotherm can be computed using Grand Canonical Monte Carlo simulations.

However, if the external pressure is sufficiently low, a good estimate of the adsorption isotherm can be obtained from the Henry coefficient K_H . Under these conditions, the number of adsorbed molecules per unit volume (ρ_a) is proportional to the Henry coefficient and external pressure P :

$$\rho_a = K_H P.$$

The Henry coefficient is directly related to the excess chemical potential of the adsorbed molecules. To see this, consider the ensemble average of the average density in a porous medium. In the grand-canonical ensemble, this ensemble average is given by (see section 6.5, Eq. (6.5.10))

$$\begin{aligned} \left\langle \frac{N}{V} \right\rangle &= \frac{1}{\Xi} \sum_{N=0}^{\infty} \frac{(fV)^N}{N!} \int d\mathbf{s}^N \exp[-\beta U(\mathbf{s}^N)] N/V \\ &= \frac{f}{\Xi} \sum_{N'=0}^{\infty} (fV)^{N'}/N'! \\ &\quad \times \int d\mathbf{s}^{N'} \exp[-\beta U(\mathbf{s}^{N'})] \int d\mathbf{s}_t \exp[-\beta U(\mathbf{s}_t)] \\ &= f \langle \exp(-\beta \Delta U^+) \rangle, \end{aligned}$$

where $N' = N - 1$, \mathbf{s}_t denotes the scaled position where we insert a test particle and ΔU^+ is defined as the change in the potential energy of the system due to the insertion of the test particle. In the limit $P \rightarrow 0$, the reservoir can be considered to be an ideal gas, in which case its fugacity becomes

$$f \rightarrow \beta P,$$

and hence

$$\left\langle \frac{N}{V} \right\rangle = \beta P \langle \exp(-\beta \Delta U^+) \rangle$$

This gives, for the Henry coefficient,

$$K_H = \beta \exp(-\beta \mu^{\text{ex}}).$$

Maginn et al. [451] and Smit and Siepmann [452,453] used the approach described in this section to compute the Henry coefficients of linear alkanes adsorbed in the zeolite silicalite. The potential describing the alkane interactions is divided into an external potential and an internal potential. The internal potential includes bond bending and torsion:

$$u^{\text{int}} = u_{\text{bend}} + u_{\text{tors}}.$$

The alkane model uses a fixed bond length. The external interactions include the remainder of the intramolecular interactions and the interactions with the zeolite:

$$u^{\text{ext}} = u^{\text{intra}} + u^{\text{zeo}}.$$

Since the Henry coefficient is calculated at infinite dilution, there is no need to consider the intermolecular alkane-alkane interactions. Smit and Siepmann used the internal interactions to generate the trial conformations (see section 12.3) and determine the normalized Rosenbluth factor using the external interactions only; this Rosenbluth factor is related to the excess chemical potential according to

$$\beta \mu^{\text{ex}} = -\ln \frac{\langle \mathcal{W}^{\text{ext}} \rangle}{\langle \mathcal{W}_{\text{IG}}^{\text{ext}} \rangle},$$

where $\langle \mathcal{W}_{\text{IG}}^{\text{ext}} \rangle$ is the Rosenbluth factor of a molecule in the ideal gas phase (no interactions with the zeolite) [454]. For an arbitrary alkane, the calculation of the Henry coefficient requires two simulations: one in the zeolite and one in the ideal gas phase. However, for butane and the shorter alkanes, all isolated (ideal gas) molecules are ideal chains, as there are no nonbonded interactions. For such chains, the Rosenbluth factors in the ideal gas phase are by definition equal to 1.

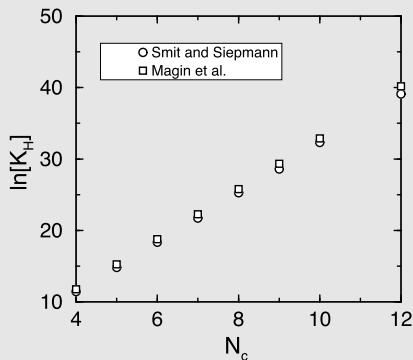


FIGURE 10.2 Henry coefficients K_H of n -alkanes in the zeolite silicalite as a function of the number of carbon atoms N_c as calculated by Maginn et al. [451] and Smit and Siepmann [453].

In Fig. 10.2 the Henry coefficients of the n -alkanes in silicalite as calculated by Smit and Siepmann are compared with those of Maginn et al. If we take into account that the models considered by Maginn et al. and Smit and Siepmann are slightly different, the results of these two independent studies are in good agreement.

10.2.3 Overlapping-distribution Rosenbluth method

Although the Rosenbluth particle-insertion scheme described in section 10.2 is correct in principle, it may run into practical problems when the excess chemical potential becomes large. Fortunately, it is possible to combine the Rosenbluth

scheme with the overlapping distribution method to obtain a technique with built-in diagnostics. This scheme is explained in the SI (section L.10). As with the original overlapping distribution method (see section 8.6.1), the scheme described in SI (section L.10) constructs two histograms, but now as a function of the logarithm of the Rosenbluth weight rather than the potential energy difference. If the sampled distributions do not overlap, then one should expect the estimate of the excess chemical potential of chain molecules to become unreliable and the Rosenbluth method should not be used. As shown in ref. [455], there is indeed a tendency for the two distributions to move apart when long chains are inserted into a moderately dense fluid. Yet, at least in the case studied in ref. [455], the statistical errors in μ_{ex} become important before the systematic errors due to inadequate sampling show up.

Illustration 14 (Rosenbluth sampling for polymers). Batoulis and Kremer [450] made a detailed analysis of the Rosenbluth algorithm for self-avoiding walks on a lattice. The Rosenbluth scheme was used to generate one walk on a lattice. Batoulis and Kremer found that, with a random insertion scheme, the probability of generating a walk of 100 steps without overlap is on the order of 0.022% (FCC-lattice). If, on the other hand, we use the Rosenbluth scheme, this probability becomes close to 100%. In Fig. 10.3, the distribution of the radius of gyration of the polymer as calculated with the corrected ensemble average (10.2.6) is compared with the uncorrected average (i.e., using the Rosenbluth scheme to generate the schemes and using $\langle A \rangle = (1/M) \sum_{n=1}^M A(n)$ instead of Eq. (10.2.6) to calculate the ensemble averages). The figure shows that the Rosenbluth scheme generates chains that are more compact. Batoulis and Kremer showed that, for longer chain lengths, this difference increases exponentially. One therefore should be careful when using such a non-Boltzmann sampling scheme.

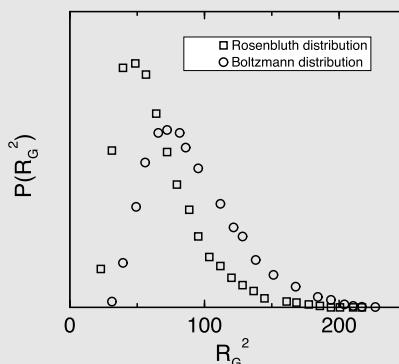


FIGURE 10.3 Probability distribution of the radius of gyration R_G . The circles show the Boltzmann distribution and the squares the Rosenbluth distribution. The results are of an FCC-lattice for a walk of 120 steps (data taken from ref. [450]).

10.2.4 Recursive sampling

In view of the preceding discussion, it would seem attractive to have unbiased sampling schemes to measure the chemical potential. Of course, thermodynamic integration methods are unbiased and the modified Widom scheme, although biased at the level of the insertion of a single monomer (like the original Widom scheme), is less biased than the Rosenbluth method. Yet, these methods cannot be used to measure μ_{ex} in a single simulation (see section 10.1).

It turns out that nevertheless, it is possible to perform unbiased sampling of μ_{ex} in a single simulation. Here, we briefly sketch the basic idea behind this method. In our description, we follow the approach proposed by Grassberger and Hegger [456,457]. Their technique is quite similar to a Monte Carlo scheme developed a few years earlier by Garel and Orland [458].

Like the Rosenbluth and modified Widom schemes, the recursive sampling approach is based on a segment-by-segment growth of the polymer. But that is about where the similarity ends. In recursive sampling, the aim is to generate a *population* of trial conformations. The excess chemical potential of a chain molecule is directly related to the average number of molecules that have survived the growth process.

The first step of the procedure is to attempt a trial insertion of a monomer in the system. Suppose that the Boltzmann factor associated with this trial insertion is $b_0 \equiv \exp[-\beta u_0(\mathbf{r}^N)]$. We now allow the monomer to make multiple copies of itself, such that the average number of copies, $\langle n_0 \rangle$, is equal to

$$\langle n_0 \rangle = \pi_0 b_0,$$

where π_0 is a constant multiplicative factor that remains to be specified. A convenient rule for determining how many copies should be made is the following. Denote the fractional part of $\pi_0 b_0$ by f_0 and the integer part by i_0 . Our rule is then to generate $i_0 (i_0 + 1)$ copies of the inserted particle with a probability $1 - f_0$ (f_0). Clearly if $i_0 = 0$, there is a probability $1 - f_0$ that the monomer will “die.” Assume that we have generated at least one copy of the monomer. Every copy from now on proceeds independently to generate offspring. For instance, to generate a dimer population, we add a segment to every surviving monomer. We denote the Boltzmann weight associated with these trial additions by $b_1(i)$, where the index i indicates that every surviving monomer will give rise to a different dimer. As before, we have to decide how many copies of the dimers should survive. This is done in exactly the same way as for the monomer; that is, the average number of dimers that descends from monomer i is given by

$$\langle n_1(i) \rangle = \pi_1 b_1(i),$$

where π_1 , just like π_0 before, is a constant to be specified later. The number of dimers generated may either be larger or smaller than the original number of monomers. We now proceed with the same recipe for the next generation

(trimers) and so on. In fact, as with the semi-flexible molecules discussed in section 10.2.1, it is convenient to include the intramolecular bond-bending, bond-stretching, and torsional energies in the probability distribution that determines with what orientation new segments should be added.

The average number of surviving molecules at the end of the ℓ th step is

$$\langle N_\ell \rangle = \left(\prod_{i=0}^{\ell} \pi_i \right) \left\langle \exp \left[-\beta \mathcal{U}_\ell(\mathbf{r}^N) \right] \right\rangle,$$

where $\mathcal{U}_\ell(\mathbf{r}^N)$ is the total interaction of the chain molecule with the N solvent molecules (and the nonbonded intramolecular interactions). The angular brackets denote a canonical average over the coordinates and over the intramolecular Boltzmann factors of the ideal (nonself-avoiding) chain. In other words,

$$\langle N_\ell \rangle = \left(\prod_{i=0}^{\ell} \pi_i \right) \exp[-\beta \mu_{\text{ex}}(\ell)].$$

Hence, the excess chemical potential is given by

$$\mu_{\text{ex}}(\ell) = -k_B T \ln \left(\frac{\langle N_\ell \rangle}{\prod_{i=0}^{\ell} \pi_i} \right). \quad (10.2.26)$$

The constants π_i should be chosen such that there is neither a population explosion nor mass extinction. If we have a good guess for $\mu_{\text{ex}}(\ell)$ then we can use this to estimate π_i . In general, however, π_i must be determined by trial and error.

This recursive algorithm has several nice features. First of all, it is computationally quite efficient (in some cases, more than an order of magnitude faster than the Rosenbluth scheme, for the same statistical accuracy). In fact, in actual calculations, the algorithm searches in *depth* first, rather than in *breadth*. That is to say, we try to grow a polymer until it has been completed (or has died). We then continue from the last branch of the tree from where we are allowed to grow another trial conformation. In this way, we work our way back to the root of the tree. The advantage of this scheme is that the memory requirements are minimal. Moreover, the structure of the program is very simple indeed if we make use of recursive function calls. Last but not least, the recursive scheme generates an *unbiased* (i.e., Boltzmann) population of chain conformations [459].

10.2.5 Pruned-enriched Rosenbluth method

An important extension of the Rosenbluth scheme has been proposed by Grassberger [460]. It is called the *pruned-enriched* Rosenbluth method (PERM). One of the reasons why the conventional Rosenbluth method fails for long chains or at high densities is that the distribution of Rosenbluth weights becomes very

broad. As a consequence, it can happen that a few conformations with a high Rosenbluth weight completely dominate the average. If this is the case, we should expect to see large statistical fluctuations in the average. It would, of course, be desirable to focus the simulations on those classes of conformations that contribute most to the average, and spend little time on conformations that have a very low Rosenbluth weight. The PERM algorithm is a generalization of the recursive-sampling scheme discussed above. It also generates a *population* of chains with different conformations. And it shares the advantage that, due to the recursive nature of the algorithm, we need not keep more than one conformation (plus a set of pointers) in memory. The “birth” and “death” rules of this algorithm are such that it generates many copies of conformations with a high Rosenbluth weight, while low-weight structures have a high probability of “dying.” The Rosenbluth weight of the remaining conformations is adjusted in such a way that our birth-death rules do not affect the desired average. As the PERM algorithm is recursive, it uses little memory. To summarize the algorithm in a few words: conformations with a high Rosenbluth weight are multiplied by a factor k and their weight is reduced by the same factor. Conformations with a low weight are “pruned”—half the low-weight conformations are discarded, while the weight of the remainder is doubled. Once all chains that have started from a common “ancestor” have been grown to completion (or have been discarded), we simply add the (rescaled) Rosenbluth weights of all surviving chains.

Below, we briefly sketch how the algorithm is implemented. Let us introduce an upper and a lower threshold of the Rosenbluth weight of a chain with length l , \mathcal{W}_i^{\max} and \mathcal{W}_i^{\min} , respectively. If the partial Rosenbluth weight of a particular chain conformation of length i , \mathcal{W}_i , exceeds the threshold, $\mathcal{W}_i > \mathcal{W}_i^{\max}$, then the single conformation is replaced by k copies. The partial Rosenbluth weight of every copy is set equal to \mathcal{W}_i/k . If, on the other hand, the partial Rosenbluth weight of a particular conformation, \mathcal{W}_i , is below the lower threshold, $\mathcal{W}_i < \mathcal{W}_i^{\min}$, then we “prune.” With a probability of 50% we delete the conformation. But if the conformation survives, we double its Rosenbluth weight. There is considerable freedom in the choice of \mathcal{W}_i^{\max} , \mathcal{W}_i^{\min} , and k . In fact, all of them can be chosen “on the fly” (as long as this choice does not depend on properties of conformations that have been grown from the same ancestor). A detailed discussion of the algorithm can be found in refs. [460,461].

The limitation of the recursive growth algorithm is that it is intrinsically a *static* Monte Carlo technique; every new configuration is generated from scratch. This is in contrast to *dynamic* (Markov-chain) MC schemes in which the basic trial move is an attempt to modify an existing configuration. Dynamic MC schemes are better suited for the simulations of many-particle systems than their static counterparts. The reason is simple: it is easy to *modify* a many-particle configuration to make other “acceptable” configurations (for instance, by displacing one particle over a small distance). In contrast, it is very difficult to generate such configurations from scratch. On the other hand, once a new

configuration is successfully generated in a static scheme, it is completely independent of all earlier configurations. In contrast, successive configurations in dynamic MC are strongly correlated.

CBMC is, in a sense, a hybrid scheme: it is a dynamic (Markov-chain) MC method. But the chain-regrowing step is more similar to a static MC scheme. However, in this step, it is less “smart” than the recursive algorithms discussed above, because it is rather “myopic.” The scheme looks only one step ahead. It may happen that we spend a lot of time growing a chain almost to completion, only to discover that there is simply no space left for the last few monomers. This problem can be alleviated by using a scanning method of the type introduced by Meirovitch [462]. This is basically a static, Rosenbluth-like method for generating polymer configurations. But, in contrast to the Rosenbluth scheme, the scanning method looks several steps ahead. If this approach is transferred naively to a configurational-bias Monte Carlo program, it would yield an enhanced generation of acceptable trial conformations, but the computational cost would rise steeply (exponentially) with the depth of the scan. This second drawback can be avoided by incorporating a recursive scanning method that cheaply eliminates doomed trial configurations, within a dynamic Monte Carlo scheme. In section 12.7 we discuss a dynamic MC algorithm (recoil growth), that is based on this approach.

Chapter 11

Long-ranged interactions

11.1 Introduction

The most time-consuming step in a simulation is the calculation of the potential energy (MC), or of the forces acting on all particles (MD). In the case of pairwise-additive interactions, the total time for such a calculation scales as the number of interacting pairs. Clearly, as the range of the interactions becomes longer, the number of pairs approaches $N(N - 1)/2$, where N is the number of particles in the system. The usual way to avoid this problem (see section 3.3.2.2) is to truncate the potential at some finite range r_c , and approximate the remaining interaction by a “tail correction” of the form

$$\mathcal{U}^{\text{tail}} \sim \frac{N\rho}{2} \int_{r_c}^{\infty} dr r^{d-1} u(r).$$

However, for interactions that decay as $1/r^d$ or slower (e.g., Coulomb or dipolar interactions), the tail corrections diverge.

Moreover, for such interactions, it is also not permissible to limit the calculation of the interactions between the nearest periodic image. Simply ignoring the long-ranged part of the potential has serious consequences (for a discussion, see ref. [463]). So, clearly, there is a problem. The more so as Coulomb and dipolar interactions between molecules are very common. Fortunately, efficient methods have been developed that can treat the long-ranged nature of the Coulomb interaction correctly.

When selecting an algorithm for a specific application, key factors to consider are accuracy and speed, not whether they are easy to explain. Modern algorithms to compute electrostatic interactions typically scale very favorably with the N , the number of particles in the system: either as $N \ln N$ or even as N . But some of the most powerful algorithms only outperform the simpler ones for fairly large system sizes. These are all considerations to keep in mind. A systematic comparison of scalable algorithms to compute long-ranged interactions can be found in a paper by Arnold et al. [464]. That paper describes many of the widely used variants of the algorithms that we discuss, but some of the more recent algorithms are not included in that review.

Before proceeding, we must discuss the fact that we will write the laws of electrostatics in a form that looks as if we are using non-SI units, but that is not the explanation. When, in section 3.3.2.5, we introduced reduced units, we

argued that it was convenient to express all observables in units of a characteristic energy ϵ , a characteristic length σ , and a characteristic mass m . Now we must introduce yet another characteristic unit, namely the unit of charge. For microscopic simulations, neither the Coulomb nor the, by now largely forgotten, *e.s.u.* are convenient units, but the absolute value e of the electron charge is. If we have two particles with charges Q_1, Q_2 at separation r in a medium with a relative dielectric constant ϵ_r , we can write their Coulomb interaction in SI units as

$$\Phi(r) = \frac{Q_1 Q_2}{4\pi \epsilon_0 \epsilon_r r},$$

where ϵ_0 denotes the dielectric permittivity of vacuum. But in a simulation, we wish to use dimensionless quantities so that we would write

$$\phi'(r) \equiv \frac{\Phi(r)}{\epsilon} = \frac{1}{\epsilon} \frac{e^2}{4\pi \epsilon_0 \epsilon_r \sigma} \frac{(Q_1/e)(Q_2/e)}{r/\sigma} = \frac{(\lambda_B/\sigma)}{(\epsilon/k_B T)} \frac{q_1 q_2}{r^*},$$

where r^* is the distance in units of σ , $q \equiv Q/e$, and λ_B is the Bjerrum length (see e.g., [59]). In terms of the reduced Bjerrum length $\lambda^* \equiv \lambda/\sigma$ and the reduced temperature $T^* \equiv k_B T/\epsilon$, we get

$$\phi'(r) = T^* \lambda^* \frac{q_1 q_2}{r^*}.$$

In what follows, we drop the asterisk on r^* . To make the notation in the remainder of this chapter compact, we go one step further and define $\phi(r) \equiv \phi'(r)/(T^* \lambda^*)$, so that Coulomb's law becomes

$$\phi(r) = \frac{q_1 q_2}{r}, \quad (11.1.1)$$

which *looks like* Coulomb's law in Gaussian units, but it is not. It is just a convenient way of expressing electrostatics in a dimensionless form with a minimum number of conversion factors. Note that once we have computed $\phi(r)$, we must still multiply the result by $T^* \lambda^*$ to obtain all energies in the same units.

Below we discuss a number of widely used methods to compute Coulomb interactions. It is easy to get buried under formalism when discussing algorithms to compute Coulomb interactions. Therefore, following the well-known rule *Tell'em what you're going tell'em*, we briefly outline what classes of algorithms we will discuss and why. Very briefly: we will only discuss different algorithms separately if they make use of a sufficiently different approach. In this spirit, we will discuss

1. The Ewald summation (section 11.2), which was the first to treat the complete Coulomb interaction as the sum of a short-range part and a long-ranged part that is evaluated by Fourier transformation.

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1 - f(r)}{r}, \quad (11.1.2)$$

with $f(r)$ some short-ranged function (in the case of the Ewald approach, $f(r)$ is a complementary error function). In the context of the Ewald sum, we also discuss:

- The corresponding expression for dipolar particles (section 11.2.1), and the calculation of the dielectric constant (section 11.2.1).
 - The fact that the nature of the boundary conditions at infinity can never be ignored (section 11.2.2).¹
 - Methods to optimize the choice of $f(r)$ in Eq. (11.1.2) (section 11.2.3).
2. Then we discuss Particle-Mesh methods that, although similar in spirit to the Ewald approach, are much faster because they use the Fast Fourier Transform of a discretized charge distribution to compute the long-range interactions (section 11.3).
 3. Next, we consider how, by smart truncation of the short-range part of the interaction, we can completely ignore the long-ranged part of the electrostatic potential and yet obtain a very good approximation to the complete expression (section 11.4).
 4. The so-called **Fast-Multipole Method (FMM)** is a very different approach, which we explain mainly in words. The **FMM** scales as N , which becomes important for large systems (section 11.5). Importantly, the **FMM** method was originally designed for two-dimensional systems, where most other methods do not perform well.
 5. Finally, we discuss two *local* algorithms that, although in principle rigorous, never compute any long-ranged interaction. We also mention one rather different algorithm that replaces infinite-ranged interactions in flat space with finite-ranged interactions on a hyper-sphere. (section 11.7).

The choice of the method to compute Coulomb interactions depends on a number of factors, such as system size and the desired accuracy, but also on which method is the fastest for a given application. Of course, the speed of a given implementation depends not just on the software but also on the hardware.

11.2 Ewald method

We start with a discussion of the Ewald-summation method, which has a long history, as it goes back to ... Ewald (1921) [466]. Ewald's method was initially used to estimate the electrostatic part of the cohesive energy of ionic crystals solids [467]. The method was first used in computer simulations by Brush, Sahlin, and Teller [468], but the current methodology started with a series of papers by De Leeuw, Perram, and Smith [469–471], to which we refer the reader for more details.

The computational effort required for the Ewald summation does not scale as N^2 , but it still increases faster than linear with the number of particles: at best as $\mathcal{O}(N^{3/2})$ —see section 11.2.3. As a consequence, the Ewald method, although

¹ A clear treatment of the boundary conditions at infinity can be found in ref. [465].

adequate for system sizes of $\mathcal{O}(10^3 - 10^4)$ particles, becomes less attractive for larger systems. The methods that we will discuss later have a more favorable scaling ($\mathcal{O}(N \log N)$ or even $\mathcal{O}(N)$). But these methods tend to have a larger overhead making them less attractive for smaller systems, whilst the methods that also work for small systems are only approximate.

Below we present a simplified discussion of the Ewald method [466] for computing long-ranged contributions to the potential energy in a system with periodic boundary conditions. A more rigorous derivation is given in the articles by De Leeuw et al. [469–471] mentioned above, and in a paper by Hansen [472].

In what follows, we limit our discussion to the three-dimensional case. For a discussion of two-dimensional Ewald sums, we refer to the reader ref. [21]. However, one aspect of dimensionality should be mentioned: Coulomb's law (Eq. (11.1.1)) is a three-dimensional law. Of course, we can embed 3d charges in a (quasi)-2d geometry, but the potential due to such charges does not satisfy the 2d Laplace equation. Rather, the potential that satisfies the 2d Laplace equation depends logarithmically on distance. For real charged particles, the true 2d equivalent of Coulomb's law is irrelevant. However, there are other interactions for which the 2d laws are relevant: for instance, the interactions between dislocations in a 2d crystal: in such cases, a true 2d equivalent of the Ewald method can sometimes be used.

Let us consider a system consisting of positively and negatively charged particles (q). These particles are assumed to be located in a cubic box with diameter L (and volume $V = L^3$). We assume periodic boundary conditions. The total number of particles in the fundamental simulation box (the unit cell) is N . We assume that the system as a whole is electrically neutral; that is, $\sum_i q_i = 0$. Moreover, we rely on the fact that a non-Coulombic short-range repulsion (Pauli exclusion principle) will prevent unlike point charges from getting arbitrarily close. We wish to compute the Coulomb contribution to the potential energy of this N -particle system,

$$\mathcal{U}_{\text{Coul}} = \frac{1}{2} \sum_{i=1}^N q_i \phi(r_i), \quad (11.2.1)$$

where $\phi(r_i)$ is the electrostatic potential at the position of ion i :

$$\phi(r_i) = \sum'_{j,\mathbf{n}} \frac{q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|}, \quad (11.2.2)$$

where the prime on the summation indicates that the sum is over all periodic images \mathbf{n} and over all particles j , except $j = i$ if $\mathbf{n} = \mathbf{0}$. In other words: particle i interacts with *all* its periodic images, but not with itself.

Eq. (11.2.2) cannot be used to compute the electrostatic energy in a simulation, because it contains a poorly converging sum; in fact, the sum is only conditionally convergent. To improve the convergence of the expression for the electrostatic potential energy, we rewrite the expression for the charge density.

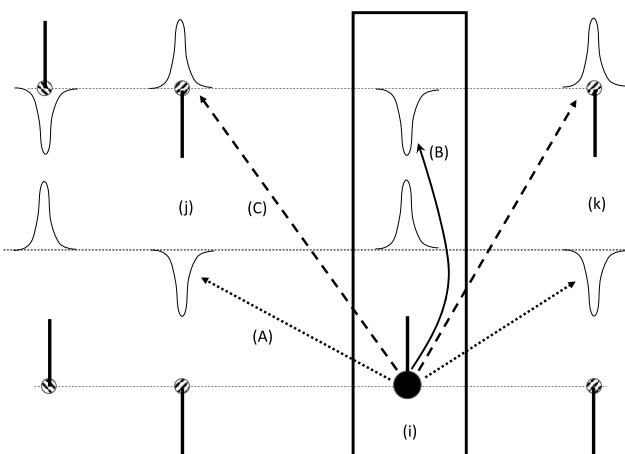


FIGURE 11.1 In the Ewald scheme to compute the electrostatic energy, we evaluate the electrostatic potential at the positions of all particles, due to the same set of point charges, represented as a set of screened charges minus the smoothly varying screening background. In the figure, particle (i) experiences the electrostatic potential due to a set of charge clouds (e.g., Gaussians) (A), plus the contribution due to a set of point charges, each embedded in a neutralizing counter-charge cloud (B). Finally, as contribution (B) over-counts the interaction of particle i with its own neutralizing charge cloud, we must subtract this contribution (C).

In Eq. (11.2.2) we have represented the charge density as a sum of point charges. The contribution to the electrostatic potential due to these point-charges decays as $1/r$. Now consider what happens if we assume that every particle i with charge q_i is surrounded by a diffuse charge distribution of the opposite sign, such that the total charge of this cloud exactly cancels q_i . In that case, the electrostatic potential of such a composite particle is due exclusively to the fraction of the point charge that is not screened. At large distances, this fraction goes to zero. How rapidly depends on the functional form of the screening charge distribution. In what follows, we shall assume a Gaussian distribution for the screening charge cloud.²

The contribution to the electrostatic potential at a point r_i due to a set of screened charges can be easily computed by direct summation, because the electrostatic potential due to a screened charge is a short-ranged function of r . However, our aim is to evaluate the potential due to *point* charges, not *screened* charges. Hence, we must correct for the fact that we have added a screening charge cloud to every particle. The different contributions to the charge density are shown schematically in Fig. 11.1. This compensating charge density varies smoothly in space. We wish to compute the electrostatic potential at the site of ion i . Of course, we should exclude the electrostatic interaction of the ion with itself. We have three contributions to the electrostatic potential: first of all, the

² The choice of Gaussian compensating charge clouds is simple and convenient, but other choices are possible —see [473]).

one due to the point charge q_i . Secondly, the one due to the (Gaussian) *screening* charge cloud with charge $-q_i$, and finally, the one due to the *compensating* charge cloud with charge q_i . In order to exclude Coulomb self-interactions, we should not include any of these three contributions to the electrostatic potential at the position of ion i . However, it turns out that it is convenient to retain the contribution due to the *compensating* charge distribution and correct for the resulting spurious interaction afterwards. The reason why we retain the compensating charge cloud for ion i is that, if we do so, the compensating charge distribution is not only a smoothly varying function, but it is also the same for all particles, and it is periodic. Such a function can be represented by a (rapidly converging) Fourier series, and this will turn out to be essential for the numerical implementation. Of course, in the end, we should correct for the inclusion of a spurious “self” interaction between ion i and the compensating charge cloud.

One point needs stressing: in the Ewald method, we compute the *electrostatic potential* by expressing the charge distribution as described above. However, when we compute the *electrostatic energy*, we use Eq. (11.2.1) to compute how this electrostatic potential interacts with a set of *point charges*.

Let us next consider the individual terms that contribute to the electrostatic energy. We assume that the compensating charge distribution surrounding an ion i is a Gaussian with width $\sqrt{2/\alpha}$:

$$\rho_{\text{Gauss}}(r) = -q_i (\alpha/\pi)^{\frac{3}{2}} \exp(-\alpha r^2).$$

The choice of α will be determined later by considerations of computational efficiency. We shall first evaluate the contribution to the Coulomb energy due to the continuous background charge, then the spurious “self” term, and finally the real-space contribution due to the screened charges.

Fourier transformation

This chapter relies heavily on the use of Fourier transforms. Here we summarize the basics of Fourier transformation in the context of electrostatics.

We already know Coulomb’s law in the form

$$\phi(\mathbf{r}) = \frac{z}{|\mathbf{r}|}. \quad (11.2.3)$$

The electrostatic potential at a point \mathbf{r} due to a collection of charges is then

$$\phi(\mathbf{r}) = \sum_{i=1}^N \frac{q_i}{|\mathbf{r} - \mathbf{r}_i|}.$$

However, this expression is of little use in simulations where the system is repeated periodically. In simulations, our aim is to compute the electrostatic energy of a system from knowledge of the charge distribution $\rho_P(\mathbf{r})$ made up of point

charges:

$$\rho_P(\mathbf{r}) = \sum_{i=1}^N q_i \delta(\mathbf{r} - \mathbf{r}_i), \quad (11.2.4)$$

where \mathbf{r}_i and q_i denote the position and the charge of particle i .

In order to relate the electrostatic potential to the charge density, we now use Poisson's equation for the electrostatic potential. For our choice of units, the Poisson equation reads:

$$-\nabla^2 \phi(\mathbf{r}) = 4\pi \rho_P(\mathbf{r}), \quad (11.2.5)$$

where $\phi(\mathbf{r})$ is the electrostatic potential at point \mathbf{r} .

For what follows, it is convenient to consider the Fourier transform of Poisson's equation. For convenience, we assume a system in a periodically repeated cubic box with diameter L and volume $V = L^d$. Any well-behaved function $f(\mathbf{r})$ with period L in all directions can be represented by a Fourier series:

$$f(\mathbf{r}) = \frac{1}{V} \sum_{\mathbf{k}=-\infty}^{\infty} \tilde{f}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}}, \quad (11.2.6)$$

where $\mathbf{k} = (2\pi/L)\mathbf{l}$ with $\mathbf{l} = (l_x, l_y, l_z)$ are the lattice vectors in Fourier space. The Fourier coefficients $\tilde{f}(\mathbf{k})$ are calculated using

$$\tilde{f}(\mathbf{k}) = \int_V d\mathbf{r} f(\mathbf{r}) e^{-i\mathbf{k}\cdot\mathbf{r}}. \quad (11.2.7)$$

In Fourier space Poisson's equation (11.2.5) reduces to:

$$\begin{aligned} -\nabla^2 \phi(\mathbf{r}) &= -\nabla^2 \left(\frac{1}{V} \sum_{\mathbf{k}} \tilde{\phi}(\mathbf{k}) e^{i\mathbf{r}\cdot\mathbf{k}} \right) \\ &= \frac{1}{V} \sum_{\mathbf{k}} k^2 \tilde{\phi}(\mathbf{k}) e^{i\mathbf{r}\cdot\mathbf{k}} \\ &= \frac{4\pi}{V} \sum_{\mathbf{k}} \tilde{\rho}(\mathbf{k}) e^{i\mathbf{r}\cdot\mathbf{k}}, \end{aligned} \quad (11.2.8)$$

where $\tilde{\rho}(\mathbf{k})$ in the last equality, denotes the Fourier transform of the charge density. The equality in Eq. (11.2.8) must hold for every Fourier component. Hence:

$$k^2 \tilde{\phi}(k) = 4\pi \tilde{\rho}(k). \quad (11.2.9)$$

To find the solution of Poisson's equation for a point charge of strength z at the origin, we have to perform the Fourier transform of a delta function:

$$\begin{aligned}\tilde{\rho}(\mathbf{k}) &= \int_V d\mathbf{r} z\delta(\mathbf{r})e^{-i\mathbf{k}\cdot\mathbf{r}} \\ &= z.\end{aligned}$$

This yields as solution for the Poisson equation

$$\tilde{\phi}(k) = \frac{4\pi z}{k^2}.$$

The solution for a unit charge is the so-called Green's function:

$$\tilde{g}(k) = \frac{4\pi}{k^2}. \quad (11.2.10)$$

For a collection of point charges, the charge density is given by Eq. (11.2.4), and we can write for the Fourier coefficients of the potential

$$\tilde{\phi}(k) = \tilde{g}(k)\tilde{\rho}(k)$$

with

$$\begin{aligned}\tilde{\rho}(k) &= \int_V d\mathbf{r} \sum_{i=1}^N q_i \delta(\mathbf{r} - \mathbf{r}_i) e^{-i\mathbf{k}\cdot\mathbf{r}} \\ &= \sum_{i=1}^N q_i e^{-i\mathbf{k}\cdot\mathbf{r}_i}.\end{aligned} \quad (11.2.11)$$

These equations show that in Fourier space the solution of Poisson's equation is simply obtained by multiplying $\tilde{\rho}(k)$ and $\tilde{g}(k)$ for all k vectors.

In what follows we will also use another property of the Fourier transform. If we have a function $f_1(x)$, which is the convolution (\star) of two other functions $f_2(x)$ and $f_3(x)$:

$$f_1(x) \equiv f_2(x) \star f_3(x) \equiv \int dx' f_2(x') f_3(x - x'),$$

then the Fourier coefficients of these functions are related by a simple multiplication:

$$\tilde{f}_1(k) = \tilde{f}_2(k)\tilde{f}_3(k).$$

For example, if we have a charge distribution that is “smeared out” around a sum of δ functions by replacing each δ -function centered at \mathbf{r}_i with a normalized

distribution function $\gamma(\mathbf{r} - \mathbf{r}_i)$, then we can write:

$$\rho(\mathbf{r}) = \sum_i q_i \gamma(\mathbf{r} - \mathbf{r}_i) = \int d\mathbf{r}' \gamma(\mathbf{r}') \rho_p(\mathbf{r} - \mathbf{r}'), \quad (11.2.12)$$

and the Poisson equation in Fourier space takes the form

$$\tilde{\phi}(k) = \tilde{g}(k) \tilde{\gamma}(k) \tilde{\rho}(k).$$

This result is convenient because it shows that in Fourier space the effect of smearing out a charge distribution amounts to a simple multiplication.

Fourier part of Ewald sum

We now apply the properties of the Poisson equation in Fourier form to compute the electrostatic potential at a point r_i due to a charge distribution $\rho_S(r)$ that consists of a periodic sum of Gaussians (interaction (A) in Fig. 11.1):

$$\rho_S(r) = \sum_{j=1}^N \sum_{\mathbf{n}} q_j (\alpha/\pi)^{\frac{3}{2}} \exp \left[-\alpha |\mathbf{r} - (\mathbf{r}_j + \mathbf{n}L)|^2 \right],$$

where the subscript S in ρ_S denotes the *smoothed* charge distribution (as opposed to the real (point)charge distribution $\rho(r)$). To compute the electrostatic potential $\phi_S(r)$ due to this charge distribution, we use the Fourier form of the Poisson equation:

$$k^2 \tilde{\phi}_S(k) = 4\pi \tilde{\rho}_S(k).$$

Fourier transforming the charge density ρ_S yields

$$\begin{aligned} \tilde{\rho}_S(\mathbf{k}) &= \int_V d\mathbf{r} \exp(-i\mathbf{k} \cdot \mathbf{r}) \rho_S(\mathbf{r}) \\ &= \int_V d\mathbf{r} \exp(-i\mathbf{k} \cdot \mathbf{r}) \sum_{j=1}^N \sum_{\mathbf{n}} q_j (\alpha/\pi)^{\frac{3}{2}} \exp \left[-\alpha |\mathbf{r} - (\mathbf{r}_j + \mathbf{n}L)|^2 \right] \\ &= \int_{\text{all space}} d\mathbf{r} \exp(-i\mathbf{k} \cdot \mathbf{r}) \sum_{j=1}^N q_j (\alpha/\pi)^{\frac{3}{2}} \exp \left[-\alpha |\mathbf{r} - \mathbf{r}_j|^2 \right] \\ &= \sum_{j=1}^N q_j \exp(-i\mathbf{k} \cdot \mathbf{r}_j) \exp \left(-k^2 / 4\alpha \right). \end{aligned} \quad (11.2.13)$$

If we now insert this expression in Poisson's equation, we obtain

$$\tilde{\phi}_S(k) = \frac{4\pi}{k^2} \sum_{j=1}^N q_j \exp(-i\mathbf{k} \cdot \mathbf{r}_j) \exp \left(-k^2 / 4\alpha \right). \quad (11.2.14)$$

For a neutral system, the term $\tilde{\phi}_S(k=0)$ is ill-defined, as both numerator and denominator vanish. Such behavior is a direct consequence of the conditional convergence of the Ewald sum.

In section 11.2.2 we will see that if the system has a net polarization, and if this polarization results in a depolarizing field, then the electrostatic energy contains a term for $\mathbf{k} = 0$. However, for the time being, we shall assume that the term with $\mathbf{k} = 0$ is equal to 0. As we shall see in section 11.2.2, this assumption is consistent with a situation where the periodic system is embedded in a medium with infinite dielectric constant.

We now compute the contribution to the potential energy due to ϕ_S , using Eq. (11.2.1). To this end, we first compute $\phi_S(r)$:

$$\begin{aligned}\phi_S(r) &= \frac{1}{V} \sum_{\mathbf{k} \neq 0} \tilde{\phi}_S(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{r}) \\ &= \sum_{\mathbf{k} \neq 0} \sum_{j=1}^N \frac{4\pi q_j}{k^2} \exp[i\mathbf{k} \cdot (\mathbf{r} - \mathbf{r}_j)] \exp(-k^2/4\alpha),\end{aligned}\quad (11.2.15)$$

and hence,

$$\begin{aligned}\mathcal{U}_S &\equiv \frac{1}{2} \sum_i q_i \phi_S(r_i) \\ &= \frac{1}{2} \sum_{\mathbf{k} \neq 0} \sum_{i,j=1}^N \frac{4\pi q_i q_j}{V k^2} \exp[i\mathbf{k} \cdot (\mathbf{r}_i - \mathbf{r}_j)] \exp(-k^2/4\alpha) \\ &= \frac{1}{2V} \sum_{\mathbf{k} \neq 0} \frac{4\pi}{k^2} |\tilde{\rho}(\mathbf{k})|^2 \exp(-k^2/4\alpha),\end{aligned}\quad (11.2.16)$$

where we have used the definition

$$\tilde{\rho}(\mathbf{k}) \equiv \sum_{i=1}^N q_i \exp(i\mathbf{k} \cdot \mathbf{r}_i).\quad (11.2.17)$$

Correction for self-interaction

The expression for the electrostatic potential energy given in Eq. (11.2.16) includes a term $(1/2)q_i \phi_{\text{self}}(r_i)$ that is due to the interaction of a point charge q_i with its own compensating charge cloud (see (A) in Fig. 11.1). As particles do not interact with themselves, this interaction is spurious and should be subtracted. We do this by evaluating the interaction of point charge i with its surrounding charge cloud (interaction (B) in Fig. 11.1).

To evaluate the correction to the spurious self-interaction, we must compute the electrostatic potential at the center of a Gaussian charge cloud. The charge

distribution that we have overcounted is

$$\rho_{\text{Gauss}}(r) = -q_i(\alpha/\pi)^{\frac{3}{2}} \exp(-\alpha r^2).$$

We can compute the electrostatic potential due to this charge distribution using Poisson's equation. Using the spherical symmetry of the Gaussian charge cloud, we can write Poisson's equation as

$$-\frac{1}{r} \frac{\partial^2 r \phi_{\text{Gauss}}(r)}{\partial r^2} = 4\pi \rho_{\text{Gauss}}(r)$$

or

$$-\frac{\partial^2 r \phi_{\text{Gauss}}(r)}{\partial r^2} = 4\pi r \rho_{\text{Gauss}}(r).$$

Partial integration yields

$$\begin{aligned} -\frac{\partial r \phi_{\text{Gauss}}(r)}{\partial r} &= \int_{\infty}^r dr \ 4\pi r \rho_{\text{Gauss}}(r) \\ &= 2\pi q_i (\alpha/\pi)^{\frac{3}{2}} \int_r^{\infty} dr^2 \ \exp(-\alpha r^2) \\ &= 2q_i (\alpha/\pi)^{\frac{1}{2}} \exp(-\alpha r^2). \end{aligned} \quad (11.2.18)$$

A second partial integration gives

$$\begin{aligned} r \phi_{\text{Gauss}}(r) &= -2q_i (\alpha/\pi)^{\frac{1}{2}} \int_0^r dr \ \exp(-\alpha r^2) \\ &= -q_i \operatorname{erf}(\sqrt{\alpha}r), \end{aligned} \quad (11.2.19)$$

where, in the last line, we have employed the definition of the error function: $\operatorname{erf}(x) \equiv (2/\sqrt{\pi}) \int_0^x \exp(-u^2) du$. Hence,

$$\phi_{\text{Gauss}}(r) = -\frac{q_i}{r} \operatorname{erf}(\sqrt{\alpha}r). \quad (11.2.20)$$

To compute the spurious self term to the potential energy, we must compute $\phi_{\text{Gauss}}(r)$ at $r = 0$. It is easy to verify that

$$\phi_{\text{Gauss}}(r = 0) = -2q_i (\alpha/\pi)^{\frac{1}{2}}.$$

Hence, the correction for the spurious contribution to the potential energy is

$$U_{\text{self}} = -\frac{1}{2} \sum_{i=1}^N q_i \phi_{\text{self}}(r_i)$$

$$= -(\alpha/\pi)^{\frac{1}{2}} \sum_{i=1}^N q_i^2. \quad (11.2.21)$$

The correction term $\mathcal{U}_{\text{self}}$ should be subtracted from the sum of the real-space and Fourier contributions to the Coulomb energy. Note that Eq. (11.2.21) does not depend on the particle positions. Hence, during a simulation, this term is constant, provided that the number of particles in the system is fixed and the values of all (partial) charges remain unchanged.

Real-space sum

Finally, we must compute the electrostatic energy due to all pair interactions between the individual point charges and the screened point charges of all other particles (interaction (C) in Fig. 11.1). Using the results of section 11.2, in particular Eq. (11.2.20), we can immediately write the (short-range) electrostatic potential due to a point charge q_i surrounded by a Gaussian with net charge $-q_i$:

$$\begin{aligned} \phi_{\text{short-range}}(r) &= \frac{q_i}{r} - \frac{q_i}{r} \operatorname{erf}(\sqrt{\alpha}r) \\ &= \frac{q_i}{r} \operatorname{erfc}(\sqrt{\alpha}r), \end{aligned} \quad (11.2.22)$$

where the last line defines the complementary error function $\operatorname{erfc}(x) \equiv 1 - \operatorname{erf}(x)$. The total contribution of the screened Coulomb interactions to the potential energy is then given by

$$\mathcal{U}_{\text{short-range}} = \frac{1}{2} \sum_{i \neq j}^N q_i q_j \operatorname{erfc}(\sqrt{\alpha} r_{ij}) / r_{ij}. \quad (11.2.23)$$

The total electrostatic contribution to the potential energy now becomes the sum of Eqs. (11.2.16), (11.2.21), and (11.2.23):

$$\begin{aligned} \mathcal{U}_{\text{Coul}} &= \frac{1}{2V} \sum_{\mathbf{k} \neq 0} \frac{4\pi}{k^2} |\tilde{\rho}(\mathbf{k})|^2 \exp(-k^2/4\alpha) \\ &\quad - (\alpha/\pi)^{\frac{1}{2}} \sum_{i=1}^N q_i^2 + \frac{1}{2} \sum_{i \neq j}^N \frac{q_i q_j \operatorname{erfc}(\sqrt{\alpha} r_{ij})}{r_{ij}}. \end{aligned} \quad (11.2.24)$$

11.2.1 Dipolar particles

Once we have the expression for the electrostatic potential energy of a periodic system of point charges, the corresponding expressions for the potential energy of a system containing dipolar molecules can be derived by differentiation. The only modification is that we must everywhere replace q_i by $-\boldsymbol{\mu}_i \cdot \nabla_i$, where $\boldsymbol{\mu}$

is the dipole moment. For example, the electrostatic energy of a dipolar system becomes

$$\begin{aligned} \mathcal{U}_{\text{dipolar}} = & \frac{1}{2V} \sum_{\mathbf{k} \neq 0} \frac{4\pi}{k^2} |\tilde{\mathbf{M}}(\mathbf{k})|^2 \exp(-k^2/4\alpha) \\ & - \frac{2\pi}{3} (\alpha/\pi)^{\frac{3}{2}} \sum_{i=1}^N \mu_i^2 \\ & + \frac{1}{2} \sum_{i \neq j}^N [(\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j) B(r_{ij}) - (\boldsymbol{\mu}_i \cdot \mathbf{r}_{ij})(\boldsymbol{\mu}_j \cdot \mathbf{r}_{ij}) C(r_{ij})], \quad (11.2.25) \end{aligned}$$

where

$$\begin{aligned} B(r) & \equiv \frac{\operatorname{erfc}(\sqrt{\alpha}r)}{r^3} + 2(\alpha/\pi)^{\frac{1}{2}} \frac{\exp(-\alpha r^2)}{r^2}, \\ C(r) & \equiv 3 \frac{\operatorname{erfc}(\sqrt{\alpha}r)}{r^5} + 2(\alpha/\pi)^{\frac{1}{2}} \left(2\alpha + 3/r^2\right) \frac{\exp(-\alpha r^2)}{r^2}, \end{aligned}$$

and

$$\tilde{\mathbf{M}}(\mathbf{k}) \equiv \sum_{i=1}^N i \boldsymbol{\mu}_i \cdot \mathbf{k} \exp(i\mathbf{k} \cdot \mathbf{r}_i).$$

Again, this expression applies to a situation where the periodic system is embedded in a material with infinite dielectric constant.

Dielectric constant

To derive an expression for the dielectric constant of a polar fluid, we consider the system shown in Fig. 11.2: a large spherical dielectric with radius a and dielectric constant ϵ (region I) surrounded by a much larger sphere with radius b and dielectric constant ϵ' (region II). The entire system is placed in vacuum (region III), and an external electric field \mathbf{E}_0 is applied. The potential at a given point in this system follows from the solution of the Poisson equation with the appropriate boundary conditions (continuity of the normal component of the displacement \mathbf{D} and tangential component of the electric field \mathbf{E}) at the two boundaries between regions I and II, and II and III. Solving the set of linear equations that impose the continuity of \mathbf{D}_\perp and \mathbf{E}_\parallel , and taking the limit $a \rightarrow \infty$, $b \rightarrow \infty$, $a/b \rightarrow 0$, we obtain the following expression for the electric field in region I,

$$\mathbf{E}_I = \frac{9\epsilon'}{(\epsilon' + 2)(2\epsilon' + \epsilon)} \mathbf{E}, \quad (11.2.26)$$

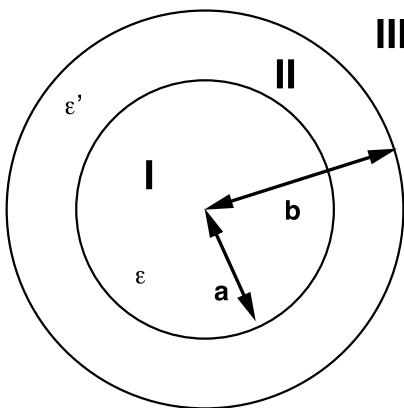


FIGURE 11.2 Spherical dielectric surrounded by a sphere.

which gives, for the polarization \mathbf{P} ,

$$\mathbf{P} \equiv \frac{\epsilon - 1}{4\pi} \mathbf{E}_I = \frac{9\epsilon'(\epsilon - 1)}{4\pi(\epsilon' + 2)(2\epsilon' + \epsilon)} \mathbf{E}_0. \quad (11.2.27)$$

In order to make contact with linear response theory, we should compute the polarization of the system as a function of the applied field inside region I , i.e., the electric field that would be present in this region in the absence of the particles. Using Eq. (11.2.26), it is easy to derive that the electrostatic field \mathbf{E}'_I that would be present in region I if it were empty is given by Eq. (11.2.26) with $\epsilon = 1$:

$$\mathbf{E}'_I = \frac{9\epsilon'}{(\epsilon' + 2)(2\epsilon' + 1)} \mathbf{E}_0.$$

The field \mathbf{E}'_I is uniform throughout region I . If we assume that the system is isotropic, we can write for the polarization

$$\begin{aligned} \langle \mathbf{P} \rangle &= \frac{1}{VQ} \int d\mathbf{r}^N \sum_{i=1}^N \boldsymbol{\mu}_i \exp \left[-\beta \left(\mathcal{H}_0 - \sum_{i=1}^N \boldsymbol{\mu}_i \cdot \mathbf{E}'_I \right) \right] \\ &= \frac{\beta}{3V} \left(\langle \mathbf{M}^2 \rangle - \langle \mathbf{M} \rangle^2 \right) \mathbf{E}'_I, \end{aligned} \quad (11.2.28)$$

where, in the second line, we have assumed that the response is linear.

Note that Eq. (11.2.28) describes the relation between the polarization $\langle \mathbf{P} \rangle$ and the electric field acting on the medium: the external field \mathbf{E}_0 has disappeared from this expression. This is important because the relation between $\langle \mathbf{P} \rangle$ and \mathbf{E}_0 depends on the shape of the dielectric medium, whereas the relation between $\langle \mathbf{P} \rangle$ and \mathbf{E}'_I is shape-independent.

Comparison of Eqs. (11.2.28) and (11.2.27) yields

$$\langle \mathbf{P} \rangle = \frac{1}{3} \beta \rho g_k \mu^2 \mathbf{E}'_I, \quad (11.2.29)$$

where the g_k is the Kirkwood factor, which is defined as

$$g_k \equiv \frac{1}{N \mu^2} \left(\langle \mathbf{M}^2 \rangle - \langle \mathbf{M} \rangle^2 \right),$$

where \mathbf{M} is the total dipole moment

$$\mathbf{M} = \sum_{i=1}^N \boldsymbol{\mu}_i.$$

Combining Eqs. (11.2.27) and (11.2.29) gives

$$\frac{(\epsilon - 1)(2\epsilon' + 1)}{(2\epsilon' + \epsilon)} = \frac{4}{3} \pi \beta \rho g_k \mu^2.$$

For a simulation with conducting boundary conditions ($\epsilon' \rightarrow \infty$), the expression for the dielectric constant becomes

$$\epsilon = 1 + \frac{4}{3} \pi \rho \beta g_k \mu^2. \quad (11.2.30)$$

This result shows that the fluctuations of the dipole moment depend on the dielectric constant of the *surrounding* medium. This, in turn, implies that, for a polar system, the Hamiltonian itself depends on the dielectric constant ϵ' of the surrounding medium, but not on the dielectric constant ϵ of the medium itself.

11.2.2 Boundary conditions

It may appear strange that the form of the potential energy of an infinite periodic system of ions or dipoles should depend on the nature of the boundary conditions at infinity.

However, for systems of charges or dipoles, this is a very real effect, which has a simple physical interpretation. To see this, consider the system shown in Fig. 11.2. The fluctuating dipole moment of the unit cell \mathbf{M} gives rise to a surface charge at the boundary of the sphere, which, in turn, is responsible for a homogeneous depolarizing field:

$$\mathbf{E} = -\frac{4\pi \mathbf{P}}{2\epsilon' + 1},$$

where $\mathbf{P} \equiv \mathbf{M}/V$. Now let us consider the reversible work per unit volume that must be performed against this depolarizing field to create the net polarization

P. Using

$$dw = -\mathbf{E}d\mathbf{P} = \frac{4\pi}{2\epsilon' + 1}\mathbf{P}d\mathbf{P},$$

we find that the total work needed to polarize a system of volume V equals

$$\mathcal{U}_{\text{pol}} = \frac{2\pi}{2\epsilon' + 1} P^2 V = \frac{2\pi}{2\epsilon' + 1} M^2 / V$$

or, using the explicit expression for the total dipole moment of the periodic box,

$$\mathcal{U}_{\text{pol}} = \frac{2\pi}{(2\epsilon' + 1)V} \left| \sum_{i=1}^N \mathbf{r}_i q_i \right|^2,$$

in the Coulomb case, and

$$\mathcal{U}_{\text{pol}} = \frac{2\pi}{(2\epsilon' + 1)V} \left| \sum_{i=1}^N \boldsymbol{\mu}_i \right|^2,$$

in the dipolar case. This contribution to the potential energy corresponds to the $\mathbf{k} = \mathbf{0}$ term that we have neglected thus far. It is permissible to ignore this term if the depolarizing field vanishes. This is the case if our periodic system is embedded in a medium with infinite dielectric constant (a conductor, $\epsilon' \rightarrow \infty$), which is what we have assumed throughout. Ballenegger [465] gives a unified discussion of the effect of differently shaped boundaries at infinity.

For simulations of ionic systems, it is essential to use such “conducting” (sometime called “tin-foil”) boundary conditions; for polar systems, it is merely advantageous. For a discussion of these subtle points, see [474]. A paper by Sprik explains how simulations can be carried out at constant \mathbf{D} or \mathbf{E} [475].

11.2.3 Accuracy and computational complexity

In the Ewald summation, the calculation of the energy is performed in two parts: the real-space part (11.2.22) and the part in Fourier space (11.2.16). For a given implementation, we have to choose the parameter α that characterizes the width of the Gaussian charge distributions, r_c the real-space cutoff distance, and k_c the cutoff in Fourier space. In fact, it is common to write k_c as $2\pi/Ln_c$, where n_c is a positive integer. The total number of Fourier components within this cutoff value is equal to $(4\pi/3)n_c^3$. The values of these parameters depend on the desired accuracy ϵ , that is, the root-mean-squared difference between the exact Coulombic energy and the results from the Ewald summation. Expressions for the cutoff errors in the Ewald summation method³ have been derived in

³ The accuracy is dependent on whether we focus on the energy (for Monte Carlo) or on the forces (for Molecular Dynamics).

[476,477]. For the energy, the standard deviation of the real-space cutoff error of the total energy is

$$\delta E_R \approx Q \left(\frac{r_c}{2L^3} \right)^{1/2} \frac{1}{\alpha r_c^2} \exp(-\alpha r_c^2) \quad (11.2.31)$$

and for the Fourier part of the total energy

$$\delta E_F \approx Q \frac{n_c^{1/2}}{L^2} \frac{\sqrt{\alpha}}{(\pi n_c/L)^2} \exp\left[-(\pi n_c/L)^2/\alpha\right], \quad (11.2.32)$$

where

$$Q = \sum_i q_i^2.$$

Note that for both the real-space part and the Fourier part, the strongest dependence of the estimated error on the parameters α , r_c , and n_c is through a function of the form $x^{-2} \exp(-x^2)$. We now impose that these two functions have the same value ϵ . The value of x for which $x^{-2} \exp(-x^2) = \epsilon$ we denote by s . Hence $\epsilon = s^{-2} \exp(-s^2)$. It then follows from Eq. (11.2.31) that

$$r_c = \frac{s}{\sqrt{\alpha}} \quad (11.2.33)$$

and from Eq. (11.2.32) we obtain

$$n_c = \frac{s L \sqrt{\alpha}}{\pi}. \quad (11.2.34)$$

If we insert these expressions for r_c and n_c back into the expressions (11.2.31) and (11.2.32), we find that both errors have the same functional form:

$$\delta E_R \approx Q \left(\frac{s}{\sqrt{\alpha} L^3} \right)^{1/2} \frac{\exp(-s^2)}{s^2}$$

and

$$\delta E_F \approx Q \left(\frac{s}{2\sqrt{\alpha} L^3} \right)^{1/2} \frac{\exp(-s^2)}{s^2}.$$

Hence, changing s affects both errors in the same way. We now estimate the computational effort involved in evaluating the Ewald sum. To this end, we write the total computational time as the sum of the total time in real space and the total time in Fourier space

$$\tau = \tau_R N_R + \tau_F N_F, \quad (11.2.35)$$

where τ_R is the time needed to evaluate the real part of the potential of a pair of particles and τ_F is the time needed to evaluate the Fourier part of the potential per particle and per k vector. N_R and N_F denote the number of times these terms need to be evaluated to determine the total energy or the force on the particles. If we assume a uniform distribution of particles, these two numbers follow from the estimates of r_c and n_c :

$$\begin{aligned} N_R &= \frac{4}{3}\pi \frac{s^3 N^2}{\alpha^{3/2} L^3} \\ N_F &= \frac{4}{3}\pi \frac{s^3 \alpha^{3/2} L^3 N}{\pi^3}. \end{aligned}$$

The value of α follows from minimization of Eq. (11.2.35)

$$\alpha = \left(\frac{\tau_R \pi^3 N}{\tau_F L^6} \right)^{\frac{1}{3}},$$

which yields for the time

$$\tau = \frac{8\sqrt{\tau_R \tau_F} N^{3/2} s^3}{3\sqrt{\pi}} = \mathcal{O}(N^{3/2}). \quad (11.2.36)$$

Note that, with the above expression for α , the parameters r_c and n_c follow from Eqs. (11.2.33) and (11.2.34), respectively, once we have specified the desired accuracy. To optimize the Ewald summation one has to make an estimate of τ_R/τ_F . This ratio depends on the details of the particular implementation of the Ewald summation and can be obtained from a short simulation.⁴

We conclude this section with a few comments concerning the implementation. First of all, when using Eq. (11.2.33) to relate r_c to α , one should make sure that $r_c \leq L/2$; otherwise the real part of the energy cannot be restricted to the particles in the box $\mathbf{n} = \mathbf{0}$.

A second practical point is the following: in most simulations, there are short-range interactions between the particles, in addition to the Coulomb interaction. Usually, these short-range interactions also have a cutoff radius. Clearly, it is convenient if the same cutoff radius can be used for the short-range interactions and for the real-space part of the Ewald summation. However, if this is done, the parameters of the Ewald summation need not have their optimum values.

11.3 Particle-mesh approaches

As discussed in section 11.2.3, the CPU time required for a fully optimized Ewald summation scales with the number of particles as $\mathcal{O}(N^{3/2})$. In many

⁴ A typical value of this ratio is $\tau_R/\tau_F = 3.6$ [478].

applications, we not only have the long-ranged interactions but short-range interactions as well. For such systems, it may be convenient to use the same cutoff radius for the real-space sum in the Ewald summation as for the short-range interactions. For a fixed cutoff, however, the calculation of the Fourier part of the Ewald summation scales as $\mathcal{O}(N^2)$, which makes the Ewald summation inefficient for large systems. Note that it is only the reciprocal-space part of the Ewald sum that suffers from this drawback. Clearly, it would be advantageous to have an approach that handles the Fourier part more efficiently. Several schemes for solving this problem have been proposed. They all exploit the fact that the Poisson equation can be solved more efficiently if the charges are distributed on a mesh with fixed spacing. One reason why “meshing” increases the computational efficiency is that we can then use the [Fast Fourier Transform \(FFT\)](#) [38] to compute the Fourier components of the charge density: the computational cost of an M -point [FFT](#) scales as $M \ln M$, and as M typically scales as N , the routine leads to $N \ln N$ scaling.

The efficiency and accuracy of such mesh-based algorithms depend strongly on the way in which the charges are attributed to mesh points. Below, we briefly discuss the basics of the particle-mesh approach.

The earliest particle-mesh scheme for molecular simulations was developed by Hockney and Eastwood [28]. The charges in the systems were interpolated on a grid to arrive at a discretized Poisson equation. In its simplest implementation, the particle-mesh method is fast, but not very accurate. The technique was subsequently improved by splitting the calculation into a short-range and a long-range contribution. In the spirit of the Ewald method, the short-range part is then calculated directly from the particle-particle interactions while the particle-mesh technique is used for the long-range contribution.

Below, we briefly discuss the particle-mesh methods and their relation to the Ewald-sum approach. As before, we do not discuss the most sophisticated particle-mesh approach, but a version that allows us to explain the physical idea behind the approach. Excellent and detailed reviews of particle-mesh methods and their performance exist, e.g., the reviews by Deserno and Holm [479] and by Arnold et al. [464].

A description of a “typical” particle-mesh method is reasonable because many variants of the particle-mesh methods, such as the [Particle Mesh Ewald \(PME\)](#) [480] or the [Smooth Particle Mesh Ewald \(SPME\)](#) [481], are similar in spirit, as they are all inspired by the original [Particle-Particle/Particle-Mesh \(PPPM\)](#) technique of Hockney and Eastwood [28]. In practice, the choice of the method depends on the application. For example, Monte Carlo simulations require an accurate estimate of the energy, while in Molecular Dynamics simulations, we need to compute the forces accurately. Some particle-mesh schemes are better suited to do one, and some to do the other.

Like the Ewald method, the idea behind the [PPPM](#) method is based on the splitting of the Coulomb potential a short-ranged and a long-ranged part

(Eq. (11.1.2)):

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1 - f(r)}{r}.$$

The idea of using a switching function is similar to the splitting of the Ewald summation into a short-range and a long-range part. Pollock and Glosli [482] found that different choices for $f(r)$ yield comparable results, although the efficiency of the method does depend strongly on a careful choice of this function. Darden et al. [480] showed that, if one uses the same Gaussian screening function as in the Ewald summation, the **PPPM** technique becomes very similar to the Ewald method.

It is instructive to recall the Fourier-space contribution of the energy:

$$\mathcal{U}_S = \frac{1}{2V} \sum_{\mathbf{k} \neq 0} \frac{4\pi}{k^2} |\tilde{\rho}(\mathbf{k})|^2 \exp(-k^2/4\alpha).$$

Following Deserno and Holm [479], we write the Fourier-space contribution as

$$\begin{aligned} \mathcal{U}_S &= \frac{1}{2} \sum_{i=1}^N q_i \left(\frac{1}{V} \sum_{k \neq 0} \tilde{g}(\mathbf{k}) \tilde{\gamma}(\mathbf{k}) \tilde{\rho}(\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{r}_i} \right) \\ &= \frac{1}{2} \sum_{i=1}^N q_i \phi^k(\mathbf{r}_i), \end{aligned} \quad (11.3.1)$$

where $\phi^k(\mathbf{r}_i)$ can be interpreted as the electrostatic potential due to the second term in Eq. (11.1.2):

$$\phi^k(\mathbf{r}_i) = \frac{1}{V} \sum_{\mathbf{k}} \tilde{g}(\mathbf{k}) \tilde{\gamma}(\mathbf{k} \neq 0) \tilde{\rho}(\mathbf{k}) \exp(i\mathbf{k} \cdot \mathbf{r}_i).$$

As a product in Fourier space corresponds to a convolution in real space, we see that the potential $\phi^k(\mathbf{r}_i)$ is due to the original charge distribution $\rho(x)$, convolved by a smearing function $\gamma(r)$. The Ewald summation is recovered if we choose a Gaussian smearing function, in which case $f(r)$ is given by an error function.

To evaluate the above expression for the Fourier part of the electrostatic energy using a discrete fast Fourier transform, we have to perform the following steps [479,483]:

1. Charge assignment: Up to this point, the charges in the system are not localized on lattice points. We now need a prescription to assign the charges to the grid points.
2. Solving Poisson's equation for our discrete charge distribution using a **FFT** technique (the Poisson equation on a lattice can also be solved efficiently, using a diffusion algorithm [484]).

3. Force assignment (in the case of MD): Once the electrostatic energy has been obtained from the solution of the Poisson equation, the forces have to be calculated and assigned back to the particles in our system.

At every stage, there are several options to choose from. Ref. [479] assesses the relative merits of the various options and their combinations. Below we give a brief summary of the main conclusions of [479].

To assign the charges of the system to a grid, a charge assignment function, $W(\mathbf{r})$, is introduced. For example, in a one-dimensional system, the fraction of a unit charge at position x assigned to a grid point at position x_p is given by $W(x_p - x)$. Hence, if we have a charge distribution $\rho(x) = \sum_i q_i \delta(x - x_i)$, then the charges at a grid point x_p are given by

$$\rho_M(x_p) = \frac{1}{h} \int_0^L dx W(x_p - x) \rho(x), \quad (11.3.2)$$

where L is the box diameter and h is the mesh spacing. The number of mesh points in one dimension, M , is equal to L/h . The factor $1/h$ ensures that ρ_M is a density. Many choices for the function $W(x)$ are possible, but some choices are better than others [479]. Obviously, $W(x)$ should be an even function and should be normalized such that the sum of the fractional charges equals the total charge of the system. Moreover, since the computational cost is proportional to the number of mesh points over which a single charge is distributed, a function with a small support decreases the computational cost. Of course, one would like to reduce the computational errors due to the discretization as much as possible. As a particle moves through the system, the function $W(x)$ should not yield abrupt changes in the fractional charges as it passes from one grid point to another.

A nice way to approach the charge assignment problem was described by Essmann et al. [481], who argue that the problem of discretizing the Fourier transform can be viewed as an interpolation procedure. Consider a single term in the (off-lattice) Fourier sum $q_i e^{-ik \cdot \mathbf{r}_i}$. This term cannot be used in a discrete Fourier transform, because \mathbf{r} rarely coincides with a mesh point. However, we can *interpolate* $e^{-ik \cdot \mathbf{r}_i}$ in terms of values of the complex exponential at mesh points. For convenience, consider a one-dimensional system. Moreover, let us assume that x varies between 0 and L and that there are M equidistant mesh points in this interval. Clearly, the particle coordinate x_i is located between mesh points $[Mx_i/L]$ and $[Mx_i/L] + 1$, where $[\dots]$ denotes the integer part of a real number. Let us denote the real number Mx_i/L by u_i . We can then write an order- $2p$ interpolation of the exponential as

$$e^{-ik_x x_i} \approx \sum_{j=-\infty}^{\infty} W_{2p}(u_i - j) e^{-ik_x Lj/M},$$

where the W_{2p} 's denote the interpolation coefficients. Strictly speaking, the sum over j contains only M terms. However, to account for the periodic boundary

conditions, we have written it as if $-\infty < j < \infty$. For an interpolation of order $2p$, only the $2p$ mesh point nearest to x_i contributes to the sum. For all other points, the weights W_{2p} vanish. We can now approximate the Fourier transform of the complete charge density as

$$\rho_k \approx \sum_{i=1}^N q_i \sum_{j=-\infty}^{\infty} W_{2p}(u_i - j) e^{-ik_x L j / M}.$$

This can be rewritten as

$$\rho_k \approx \sum_j e^{-ik_x L j / M} \sum_{i=1}^N q_i W_{2p}(u_i - j).$$

We can interpret the above expression as a *discrete* Fourier transform of a “meshed” charge density $\rho(j) = \sum_{i=1}^N q_i W_{2p}(u_i - j)$. This shows that the coefficients W_{2p} that were introduced to give a good interpolation of $e^{-ik_x x_i}$ end up as the charge-assignment coefficients that attribute off-lattice charges to a set of lattice points.

While the role of the coefficients W is now clear, there are still several choices possible. The most straightforward one is to use the conventional Lagrange interpolation method to approximate the exponential (see Darden et al. [480] and Petersen [477]). The Lagrange interpolation scheme is useful for Monte Carlo simulations, but less so for the Molecular Dynamics method. The reason is that although the Lagrangian coefficients are everywhere continuous, their derivative is not. This is problematic when we need to compute the forces acting on charged particles (the solution is that a separate interpolation must be used to compute the forces). To overcome this drawback of the Lagrangian interpolation scheme, Essmann et al. suggested the so-called **SPME** method [481]. The **SPME** scheme uses exponential Euler splines to interpolate complex exponentials. This approach results in weight functions W_{2p} that are $2p - 2$ times continuously differentiable. It should be stressed that we cannot automatically use the *continuum* version of Poisson’s equation in all interpolation schemes. In fact Eq. (11.2.10) is only consistent with the Lagrangian interpolation schemes. To minimize discretization errors, other schemes, such as the **SPME** method, require other forms of the Green’s function $\tilde{g}(k)$ (see ref. [479]).

In section 11.2.3, we discussed how the parameter α in the conventional Ewald sum method can be chosen such that it minimizes the numerical error in the energy (or in the forces). Petersen [477] has derived similar expressions for the **PME** method. Expressions that apply to the **PPPM** method [28] and the **SPME** scheme are discussed by Deserno and Holm [485]. In the case of the force computation, matters are complicated by the fact that, in a particle-mesh scheme, there are several inequivalent ways to compute the electrostatic forces acting on the particles. Some such schemes do not conserve momentum, and

others do —but at a cost. The choice of what is the “best” method, depends largely on the application [479].

This concludes our discussion of particle-mesh schemes. While we have tried to convey the spirit of these algorithms, we realize that this description is not sufficiently detailed to be of any help in the actual implementation of such an algorithm. We refer readers who are considering implementing one of the particle-mesh schemes to the articles of Essmann et al. [481], Deserno and Holm [479], Arnold et al. [464], and of course Hockney and Eastwood [28].

11.4 Damped truncation

The Coulomb potential is long-ranged and naive attempts to truncate it lead to very inaccurate results. However, as stressed by Wolf [486], the mathematical manipulations of the Ewald method do not clarify the physical origins of this problem. The failure of naive truncation may seem puzzling, as it had already been noted earlier [487] that the effective interactions in Coulomb fluids are short-ranged except, of course, for the fields that are due to boundary conditions at infinity. As shown in ref. [486], the reason why naive truncations of the Coulomb potential fail is that the volume within the cutoff radius is rarely charge neutral. This problem is not specific to the Ewald method: the same problem occurs with any representation of the Coulomb potential in the form given by Eq. (11.1.2)

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1-f(r)}{r}.$$

Wolf [486] showed that good estimates for the energy of a Coulomb system could be obtained, even when ignoring the long-ranged part and truncation $f(r)$ at some finite cutoff radius R_c , provided that the volume within the cut-off radius is made charge neutral. Charge-neutrality is achieved by placing a compensating charge on the surface of the sphere with radius R_c .

Wolf made the Ewald choice for $f(r)$, but of course, other choices can be, and have been, made. Here, we give Wolf’s expression for the electrostatic energy for a system with a “damped” Coulomb potential, truncated at R_c in the general form:

$$E^{\text{electrostatic}}(R_c) = \sum_{i < j}^N q_i q_j \left[\frac{f(r_{ij})}{r_{ij}} - \lim_{r_{ij} \rightarrow R_c} \frac{f(r_{ij})}{r_{ij}} \right] - E_{\text{self}}(R_c), \quad (11.4.1)$$

where the self-energy is a (known) constant term, and the term with $r_{ij} \rightarrow R_c$ accounts for the damped interaction with the compensating charge at R_c . The limit $r_{ij} \rightarrow R_c$ must be taken to ensure that the force due to the compensating charge is computed correctly and vanishes smoothly at R_c . A good overview can be found in ref. [21]. An extension to molecular systems is described in ref. [488].

Two final comments about the Wolf method: first of all, the use of a compensating charge to neutralize the volume around an ion is fine, unless the system really has regions that carry a net charge, giving rise to long-range fields (for example, a charge in a polar fluid). In such cases, the Wolf method will not account correctly for the Coulomb interaction at a distance from the ion larger than R_c . Secondly, as the Wolf method ignores long-ranged interactions, the method yields rather poor estimates of the liquid-vapor surface tension [489], unless rather large cutoff radii are used —but doing so, would remove the computational advantage of the method.

11.5 Fast-multipole methods

In a book about Molecular Simulation techniques, it is sometimes difficult to draw the line between algorithms that exploit the physics of a problem and those that use sophisticated applied mathematics to speed up a calculation, yet do not yield much additional physical insight. Fast-multipole methods are a case in point: they are undoubtedly hugely powerful tools of applied mathematics. But like, for instance, the Fast Fourier transform or the Cholesky decomposition in linear algebra, they are less likely to excite, in their own right, those who are primarily interested in the physical properties of classical many-body systems. As a consequence, it is difficult to find the right balance between too little and too much in discussing Fast Multipole Methods in the context of Molecular Simulations. Here we will describe the *idea* behind Fast Multipole Methods, but we will avoid the technical details. Those are discussed in the references mentioned in the text.

The **Fast-Multipole Method (FMM)** denotes a class of algorithms that evaluate Coulomb (or similarly long-ranged) interactions such that the computational effort scales as $\mathcal{O}(N)$, multiplied with a pre-factor that (certainly in 3d) can be rather large, plus a constant that accounts for the overhead. The **FMM** becomes competitive for large ($N > \mathcal{O}(10^5)$) systems, be it that the precise threshold depends on the desired accuracy.

The algorithm has a history dating back to Appel [490] and Barnes and Hut [491]. The fast-multipole approach as we now know it was developed by **Greengard and Rokhlin (GR)**, initially for a non-periodic, two-dimensional case [492], and later extended to 3d [493]. Schmidt and Lee extended the method to systems with periodic boundary conditions [494], which adds a bit to the initial overhead, but does not otherwise slow down the algorithm. Yoshii et al. [495] have shown that the **FMM** method can also be generalized to 3d problems with a slab geometry (periodic in the x and y directions, but not in the z direction). Ewald-like schemes typically become cumbersome for slab geometries (see SI section L.7). Other helpful descriptions of the FMM approach can be found in refs. [496,497].

In the FMM, the system (the “root”) is divided into cubic cells (“children”) that are then subdivided into smaller cubic cells (“grandchildren”) with half the

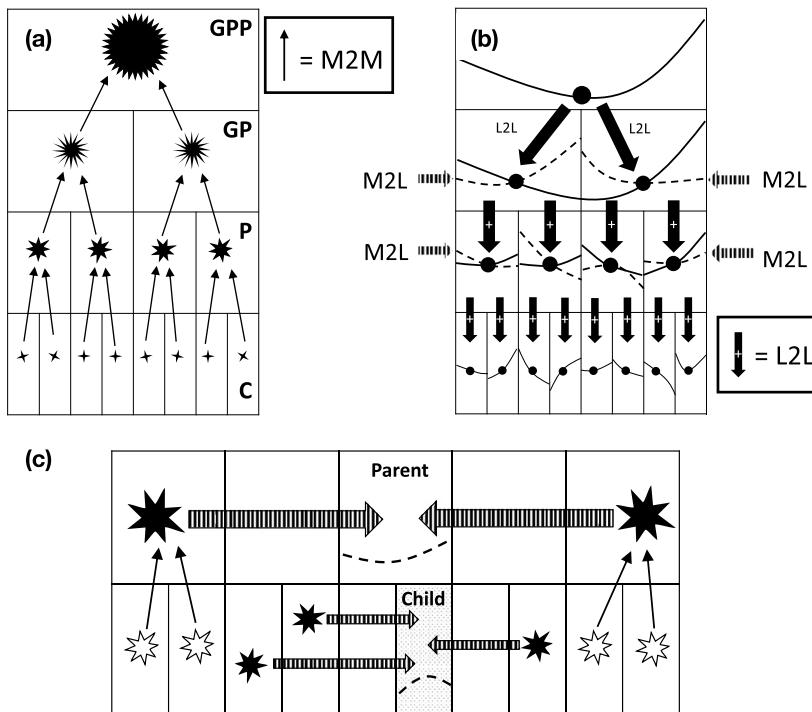


FIGURE 11.3 One-dimensional representation of the steps in the fast multipole method (FMM). (A) In the **M2M** step Multipoles in smaller cells are transformed and added to generate the Multipoles of the parent cells, which are then used to generate the multipoles of the grandparents, etc. (B) **L2L-step:** The Local potential in parent cells is transformed to generate the Local potential in the off-spring cells (black arrows). In addition, the Local potential in the off-spring also contains a Multipole-to-Local contribution (see (C)) due to multipoles in cells that are beyond the next-nearest neighbor, but not yet included at the parent level: dashed arrows. (C) **M2L** step. The Local potential due to Multipoles surrounding goldilocks cells (*not too far, not too close*). Multipoles that are further removed have been accounted for at the parent level. Those that are too close will be included in subsequent generations.

linear dimensions (see Fig. 11.3). This subdivision goes on until there are $\mathcal{O}(1)$ charges in the smallest cells. This procedure creates what is called an *octal tree* of cells. As the name suggests, the FMM is based on the fact that the electrostatic potential due to a group of charges inside a volume ΔV can be written as the sum of the contributions due to all the multipoles of the charge distribution inside that volume.

The crucial point is that an electrostatic potential in a vacuum satisfies the Laplace equation and that the solution of this equation at a point with spherical coordinates (r, θ, ϕ) , where r is the distance from the chosen origin, is of the form [498]⁵

⁵ This multipole expansion is for the 3d Coulomb potential. A similar expansion exists for the 2d Laplace equation. Note, however, $1/r$ is not a solution to the 2d Laplace equation.

$$\phi(r, \theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} \left(\frac{4\pi}{2\ell+1} \right) \left[A_{\ell m} r^{\ell} + B_{\ell m} r^{-(\ell+1)} \right] Y^{\ell m}(\theta, \phi),$$

where the $Y^{\ell m}$ denotes the spherical harmonics (unfortunately, different authors use different notation conventions). The coefficients $A_{\ell m}$ and $B_{\ell m}$ characterize the spatial distribution of the charges responsible for the potential. For instance, for a potential at distances r larger than the radius of the sphere that encloses all charges $i = 1, \dots, k$ causing the potential, the $B_{\ell m}$ are the multipole moments in spherical tensor notation:

$$M_{\ell m} = \sum_{i=1}^k q_i r_i^{\ell} Y_{\ell m}^*(\theta_i, \phi_i), \quad (11.5.1)$$

but there is an equivalent expression relating the $A_{\ell m}$ to the charges at a distance larger than r :

$$M'_{\ell m} = \sum_{i=1}^k q_i r_i^{-(\ell+1)} Y_{\ell m}^*(\theta_i, \phi_i). \quad (11.5.2)$$

In practice, the multipole expansion is truncated at some finite value of ℓ : the larger ℓ , the lower the speed but, the higher the accuracy of the algorithm.

The GR approach uses a polynomial expansion of the electrostatic potential inside a charge distribution similar to Eq. (11.5.2), but due to *multipoles*, rather than due to charges, at distances larger than r . We will not write down this expression (see ref. [496]).

This *local* polynomial expansion of the electrostatic potential can be used to compute the contribution to the potential inside a given cell due to the multipoles in cells that are far enough away to guarantee the convergence of the expansion. The simplest way to ensure convergence of the local expansion [496,499] is to include only the contribution to the local potential due to cells that are further away than the next-nearest neighbor of the central cell (next nearest because no charges responsible for this part of the local potential can be inside the cell for which we compute the local potential). Cells that are closer will be dealt with at the next level of refinement, where we split every cell into eight sub-cells.

The part of the local field that is due to multipoles that are well-removed from the local cell, is slowly varying and should be well-represented by a small number of terms, corresponding to low of powers of r . The crucial point is that the contribution to the coefficients of r^{ℓ} from the multipoles centered in the different surrounding cells (“not too close, not too far”), is given by a simple,⁶ fixed linear combination of these cell multipoles.

We do not have to compute the contribution to the local potential due to cells that are treated in the next level up (i.e., at the level of *parent* cells), where

⁶ The relation is simple only if you are into the construction of rotational invariants from products of spherical harmonics.

the same calculation is performed for the local potential due to multipoles in beyond-next-nearest-neighbor parent cells. But we have not yet explained how we compute the multipole moments of cells, their parents, their grandparents, etc.

Let us first focus on the highest resolution (when there are few, if any, charges in a cell). We use the terminology of Ref. [496] and call these cells “leaf” cells, as opposed to “root” or “branch” cells. We simply compute the multipole moments of a leaf cell using Eq. (11.5.1). From there on, we use the **GR** recursive relations described below to compute everything else. The key point is that, at this level, we are done with computing multipoles. What follows are only linear transformations with fixed coefficients.

The key steps underlying the **FMM** are:

1. There exists a “simple” linear transformation that allows us to move a set of multipoles centered on the middle of box ΔV to the center of its “parent” box, which (in 3d) contains 8 smaller cells. Note that the new multipole moments are not the same as the old ones: they are linear combinations (and they mix different orders of ℓ). We repeat the same *Multipole-to-Multipole* operation for all 8 “children” of a given parent box. We can then simply add the multipoles thus transformed and transported, and we have thereby obtained the multipole moments of the parent box.
2. We now repeat this procedure by combining 8 parent boxes into one “grand-parent” box, and so on. Note that we never have to recompute multipoles from charges: we just apply a fixed, recursive, linear operation.
3. Once we have all the multipole moments of all cells at all the different levels up to the root (the periodic unit cell), it is time to increase the resolution again to compute the multipole contributions to the local potentials.⁷ Once we have the local potential at level L , starting with $L = 0$ —the whole simulation box, we can carry out **GR**’s second set of linear transformations that transform and move the local potential at the center of the parent box to centers of its 8 off-spring boxes. Note that this *Local-to-Local* transformation takes care of all multipole fields at the scale of the parent box or larger, all the way up to $L = 0$.
4. However, now we still need to add the contribution due to the charge distribution surrounding parent cells that were closer than next nearest neighbors. As we are now one level down from the parent cells, we have a set of nearby (but not too nearby) offspring cells for which we know the multipoles. We can compute the *Multipole-to-Local* contribution of all these cell multipoles to the local potential in our target cell. We still miss all cells of the same size

⁷ Here, the situation with periodic boundary conditions is different from the case of a non-periodic charge-distribution: in the non-periodic case, we can only start using the multipole expansion of local potentials if we have cells that are beyond next nearest neighbors. However, for periodic boundary conditions, the potential due to the multipole moments of beyond-next-nearest-neighbor images of the periodic box must be computed. This stage requires an Ewald summation that is carried out only once at the beginning of the simulation. The boundary conditions at infinity enter at this stage.

that are (next)nearest neighbors. But, no problem, these are dealt with at the next refinement level.

5. Finally, arrive at the level of the “leaf” cells. There we compute the contribution to the local potential due to the beyond-next-nearest neighbor leaf cells that are not already included in the recursive calculation. Now the only thing that remains is computing the contribution to the local potential due to the charges that were not yet included in the multipole expansion. This contribution includes the electrostatic interactions between all charges inside the same leaf cell and the interaction with all charges in its immediate neighborhood. These contributions are computed explicitly using Coulomb’s law:

$$u_{\text{close}} = \sum_{\text{close}} \frac{q_i q_j}{r_{ij}}.$$

As we have already computed the expansion coefficients of the local electrostatic potential $\phi_L(\mathbf{r})$ in every “leaf” cell, computing the total electrostatic potential is now just a matter of adding $(1/2) \sum_i \phi_L(\mathbf{r}_i) q_i$.

The Fast Multipole Method has the unique feature that the computational effort scales as N . However, a potential drawback of the FMM method is that the changeover from the direct calculation of the Coulomb interactions with charges in the immediate vicinity, to a truncated multipole expansion for cells beyond the next nearest neighbor, may lead to discontinuities in the potential, which cause energy drift in [MD](#). Special regularization techniques are required to address this problem [[464,500,501](#)].

Moreover, the FMM comes with some overhead, and may not be the best solution for molecular simulations of homogeneous systems. Moreover, there are now many algorithms that scale as $N(a + b \ln N)$, but with b substantially smaller than a .

A comparison of a number of algorithms with good (N or $N \ln N$) scaling [[464](#)] (see Example 15) shows that the FMM approach is certainly very good, in particular for larger systems, but whether it is the best depends on many other factors (including the architecture of the computer used). For large, non-periodic systems many of the competitors to FMM lose their advantage and the choice becomes easier.

11.6 Methods that are suited for Monte Carlo simulations

The majority of the techniques described above are less suited for MC simulations that employ single-particle moves. The reason is that computing the long-ranged potential after moving a single charge requires computing the change in the Fourier components of the charge densities. This is inefficient in the standard implementation of the Ewald and [PME](#) methods, and also for computing the propagated multipoles and the local potentials due to a single charge in FMM. The approximate Wolf method does not suffer from this problem and is therefore also suited for Monte Carlo simulations.

Below, we describe two refreshingly different techniques that work either exclusively for MC 11.6.2, or are conceptually much simpler for MC than for MD 11.6.1.

11.6.1 Maxwell equations on a lattice

The **Maxwell Equations Method (MEM)** proposed by Maggs and coworkers [502–504] is rather different from the schemes mentioned previously as, in its simplest form, it does not compute the electrostatic potential at all, but minimizes the energy associated with the electrostatic field. The energy associated with an electric field \mathbf{E} can be written as:

$$\mathcal{U}_{\text{el}} = \frac{1}{8\pi} \int d\mathbf{r} E^2(\mathbf{r}), \quad (11.6.1)$$

where, as before, the use of the factor $1/(8\pi)$ does not imply that we use Gaussian units, but that we use reduced units, as was explained below Eq. (11.2.1). We also note that the electrostatic energy in Eq. (11.6.1) contains the *self-energy* which diverges for the case of a point particle. However, as the **MEM** method is implemented on a lattice, the self-energy remains finite. It still creates some problems, but these we will discuss later. We first focus on the simplest possible case of the MEM approach, namely one where particles have fixed charges and are constrained to lattice sites.

Our starting point is the relation between the charge density $\rho(\mathbf{r})$ and the divergence of the E -field:

$$\nabla \cdot \mathbf{E}(\mathbf{r}) = 4\pi\rho(\mathbf{r}). \quad (11.6.2)$$

We can write the E -field as the sum of two independent terms⁸ E_{tr} and E_{phi} : the first, the *transverse* field, is the rotation of some other vector field $\mathbf{Z}(\mathbf{r})$ and is therefore by construction divergence-free, the other is the gradient of a potential $\phi(\mathbf{r})$:

$$\mathbf{E}(\mathbf{r}) = -\nabla\phi + \nabla \times \mathbf{Z}(\mathbf{r}).$$

The total field energy is then

$$\mathcal{U}_{\text{el}} = \frac{1}{8\pi} \int d\mathbf{r} \left[(\nabla\phi)^2 + (\nabla \times \mathbf{Z})^2 \right], \quad (11.6.3)$$

where we note that the integral over the cross terms vanishes. In addition, we impose that divergence of the E -field satisfies Gauss's law, i.e. Eq. (11.6.2). The charges are located on the lattice points at the centers of these cells, and the

⁸ In general, \mathbf{E} may also contain a constant term \mathbf{E}_0 , due to the boundary conditions (section 11.2.2). For the MEM-algorithm, this term makes no difference, and we leave it out here.

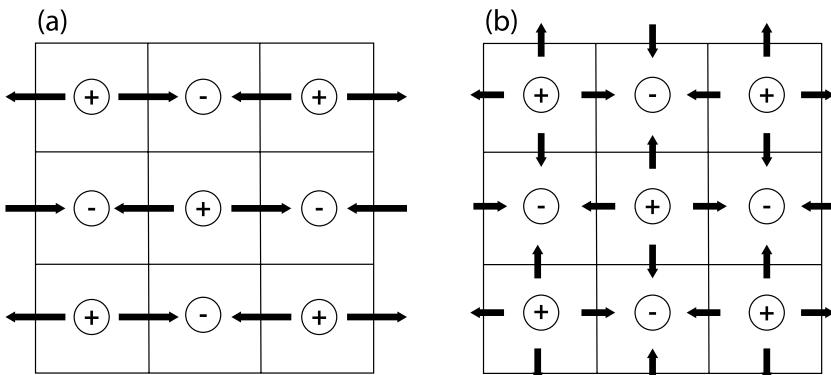


FIGURE 11.4 Simple example of two field distributions that both satisfy Gauss's law for a simple, square lattice of positive and negative charges. Fig. (a) shows an initial field distribution that has been constructed such that the divergence of E satisfies Gauss's law. However, the E fields are *not* equal to (minus) the gradient of the electrostatic potential. (b) shows a snapshot of the same lattice, but with an E -field that is equal to (minus) the gradient of the electrostatic potential. Note that situation (b) can be created from situation (a) by adding E fields that circulate around the plaquettes in a checkerboard manner, as sketched in Fig. 11.5(A). The net flux of the E -fields through the cell boundaries is the same in both figures (a) and (b), yet the field patterns are clearly quite different. This difference is due to the difference in the transverse fields.

electric fields live on the links between lattice points. Gauss law then has the form

$$\sum_{j \in \text{nn } i} E_{ij} = 4\pi Q_i ,$$

where the sum runs over the nearest neighbors of lattice point i .

At this stage, we do not yet know the fields. The procedure is now as follows: having placed the charges on some reasonable initial positions, we first generate an initial E -field that satisfies Eq. (11.6.2).⁹ Note that this E -field is not equal to $-\nabla\phi$. In fact, this original field is not rotation free (see Fig. 11.4).

Once the field has been initialized, we have two types of MC trial moves:

1. We attempt updates of the transverse part of the E -field (see Fig. 11.5A). That part of the field does not “see” the charges and hence does not depend on the particle positions. The role of these MC moves is only to sample all field configurations that are compatible with the given charge distribution. There is finite thermal energy associated with the transverse field fluctuations, but as this energy does not depend on the charges, it does not affect the Monte Carlo sampling of the positions of the charged particles. As the total field is the sum of $-\nabla\phi$ and $\nabla \times \mathbf{Z}(\mathbf{r})$, the part of the field that is not

⁹ One way of initializing the field is as follows [505]: We first apply the lattice version of Gauss's law to planes (say, perpendicular to x). Then to lines inside these planes, and finally to points inside these lines.

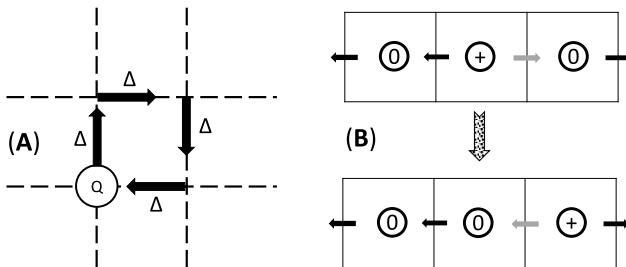


FIGURE 11.5 In its simplest form, the Maxwell Equations Method MC algorithm has two types of MC trial moves: (A) moves that change the rotation of the E -field, but not the divergence. In such a move a random amount Δ is added in a clockwise (or anti-clockwise) fashion to all links around a plaquette. The charges live on the vertices of the plaquettes. Hence, this move does not change $\nabla \cdot E$. (B) Particle moves: in a (charged) particle trial move, a charged particle is moved from a site to a neighboring site. The E -field (gray arrow) between the old and the new site is then changed such that the Gauss relation holds for the new charge geometry.

fluctuating is equal to $E_{\text{phi}} \equiv -\nabla\phi$. This part of the field does depend on the positions of the charges, and we change it by moving charges.

2. In a particle move, we attempt to displace a charge from its lattice site to a neighboring lattice site, and at the same time change the field on the link between the old and the new sites, such that Gauss's law is satisfied for the new position of the charge (see Fig. 11.5B).

The above discussion of the MEM MC algorithm is over-simplified. However, it shows that the algorithm is completely local: no long-ranged interactions are computed. Moreover, the computational effort associated with a single-particle MC move is $\mathcal{O}(1)$. However, in particular, at low temperatures, the acceptance of the trial moves can become rather low. For a discussion of this problem—and of ways to mitigate it—see ref. [504]. Another obvious limitation of the original MEM algorithm is that it deals with discrete charges that live on lattice sites.

Obviously, it would be attractive to apply the method to off-lattice models, in which case one would like to use a particle-mesh method to distribute the charges of a particle over neighboring lattice sites, using one of the particle-mesh schemes. However, now the self-energy of the electric field of a single charge creates problems: for a single charge moving in the space between lattice sites, the discretized electric-field energy depends on the position of the particle and favors positions in the middle of the lattice cells. This problem is not specific to MEM.

The MEM approach has been extended to Molecular Dynamics calculations (see refs. [464, 505]). These MD algorithms *emulate* a discretized version of the full Maxwell equations, but they are different, as the speed of light must be chosen much lower than the real speed of light. We will not discuss these MD algorithms (see, however, ref. [21]). The advantage of the Maggs method in MD is not as clear as it is in MC, as there are many, very efficient MD algorithms to deal with long-ranged forces.

11.6.2 Event-driven Monte Carlo approach

Another $\mathcal{O}(N)$ algorithm is based on the [Event Chain Monte Carlo \(ECMC\)](#) method described in Chapter 13, section 13.4.4.

We will not repeat the discussion of the [ECMC](#) method here, but just mention that by embedding each charge in a neutralizing background (either a neutralizing line charge or a neutralizing volume charge), the interactions between the charge that are subject to a trial move and the other charges are sufficiently short ranged to make use of the cell-veto method discussed in section 13.4.4. In fact, as discussed by Faulkner et al. [89], several choices are possible for treating Coulomb interactions in the context of Event-Chain MC. Although the [ECMC](#) approach does not eliminate the long-range nature of Coulomb forces, the expensive calculations need to be done only once, at the beginning of a simulation. For more details, we refer the reader to Ref. [89].

11.7 Hyper-sphere approach

Finally, we mention an early method to treat Coulomb interactions, due to Caillol and Levesque [506]. The approach of ref. [506] is based on the idea of embedding the system as a (hyper)sphere in a higher-dimensional space. As a consequence, the range of Coulomb interaction is limited by the diameter of the hyper-sphere.

Of course, curving space does change the properties of a system, in particular for crystalline solids, where the curvature of space is incompatible with a defect-free crystal. However, as shown in ref. [506], the method yields reasonable for liquids. We will not discuss the Caillol-Levesque method, but refer the reader to the original paper for more details.

Illustration 15 (Algorithms to calculate long-range interactions). As the speed of the computation of long-ranged forces is often a bottleneck in large-scale simulations, much effort has been invested in optimizing this part of the simulation code. In fact, there is a large number of algorithms (and acronyms), some of which we have discussed in this chapter.

Clearly, it is useful to know which method is best, and the answer is: “*that depends*”. First of all, as existing algorithms are improved all the time, and new algorithms keep being introduced, any comparison can only be a snapshot. But, more importantly, different conditions and requirements may change the relative ranking of algorithms. For instance, some simulations focus on homogeneous systems, and others may deal with large inhomogeneities, sharp boundaries, or spatial variations of the dielectric constant. In addition, what is best for MD may be less useful for (single-particle) MC. And then, there are factors such as the desired accuracy, the number of processors, and other machine or compiler-related issues.

Still, it is useful to give some impression of the performance of some of the better-known algorithms, as long as the reader is aware of the fact that other comparisons might give a somewhat different ranking.

In what follows, we briefly discuss the findings of a careful comparison by Arnold et al. [464] of half a dozen of the most competitive (or promising) algorithms. The comparison was published in 2013, and applies to MD simulations of homogeneous systems. A more focused review of Poisson solvers [134] was published in 2016. Still, at the time of the writing of the current edition of this book, we were not aware of comprehensive reviews that also cover more recent algorithms, such as the one mentioned in section 11.6.2.

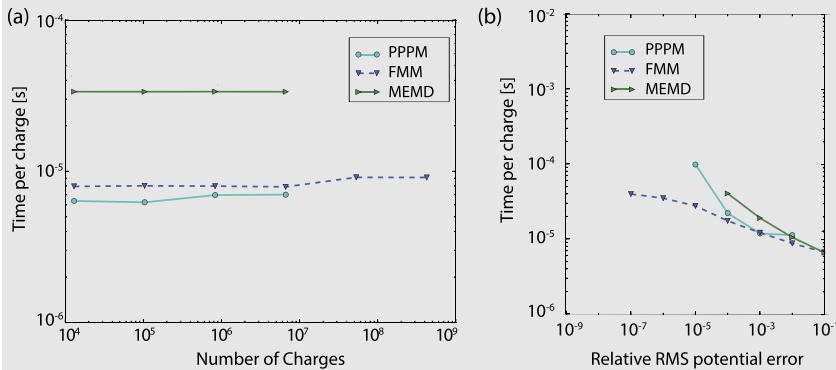


FIGURE 11.6 (a) Comparison of the wall-clock time per charge for a number of high-quality algorithms to compute Coulomb interactions in Molecular Dynamics simulations. The acronyms in the legend are briefly discussed in the text. They refer to: Particle-Particle/Particle-Mesh (PPPM), Fast-Multipole Method (FMM), MEMD refers to the molecular dynamics implementation of the Maxwell equation method. The most striking feature of this figure is that all these algorithms scale effectively as $\mathcal{O}(N)$, even the one that uses fast Fourier transforms, which scale as $\mathcal{O}(N \ln N)$. (b) Comparison of the wall-clock time per charge as a function of the RMS error in the potential energy. Figure based on the data of ref. [464].

With these caveats, let us look at some of the findings of ref. [464]. First of all, we would like to get an impression of how the speeds of the different algorithms compare. Such a comparison is shown in Fig. 11.6 (a). The MD simulation data shown in this figure refer to systems of 12,960 up to 424,673,280 BKS-silicon particles [507]. Details about the simulation parameters and the hardware used can be found in ref. [464].

The figure compares three algorithms, some of which we discussed in this chapter, namely: PPPM, a version of the “Particle-Particle-Particle-Mesh” method of ref. [28], FMM, a version of the “Fast Multipole Method” ref. [492, 493] and MEM, which is the MD version [505] of the Maxwell Equations Method discussed in section 11.6.1.

The most striking feature of Fig. 11.6 (a) is that all algorithms scale effectively linearly with particle size. To be more precise: the ones that include an

[FFT](#) step, should eventually scale as $N \ln N$. However, as the [FFT](#) part of the algorithm takes only a small fraction of the time, even for these rather large systems, the logarithmic corrections are barely visible.

Next, we look at the relative performance of the algorithms. For the accuracy shown in Fig. 11.6 (a), the [MEM](#) method seems somewhat slower than the other two. However, the conclusion about relative performance may depend on the nature of the system under study: systems that are very inhomogeneous (e.g., containing large regions of low density) tend to be better suited for fast-multipole methods, whereas the [MEM](#) algorithm is suited for [MC](#) simulations because the effect of single-particle moves is local.

Apart from these considerations, there is also the question of the desired accuracy. Arnold et al. made a comparison for the same set of algorithms, but now for a fixed system size ($N = 102,900$), and considered the cost of decreasing the relative root-mean-square error in the potential energy (see Fig. 11.6 (b)). For most simulations, a relative error of $\mathcal{O}(10^{-4})$ is acceptable. Hence, it is usually not of much interest to know how algorithms behave when the relative error is 10^{-7} or less (see ref. [464]).

Chapter 12

Configurational-bias Monte Carlo

Up to this point, we have barely addressed the fairly obvious question: what is the point of using the Monte Carlo technique in simulations? After all, Molecular Dynamics simulations can be used to study the static properties of many-body systems and, in addition, MD provides information about their dynamical behavior. Moreover, a standard MD simulation is computationally no more expensive than the corresponding MC simulation. Hence, it would seem tempting to conclude that the MC method is an elegant but outdated scheme.

As the reader may have guessed, we believe that there are good reasons to use MC rather than MD in certain cases. But we stress the phrase *in certain cases*. All other things being equal, MD is clearly the method of choice. Hence, if we use the Monte Carlo technique, we should always be prepared to justify our choice. Of course, the reasons may differ from case to case. Sometimes it is simply a matter of ease of programming: in MC simulations there is no need to compute forces. This is irrelevant if we work with pair potentials, but for many-body potentials, the evaluation of the forces may be nontrivial. Another possible reason is that we are dealing with a system that has no natural dynamics. For instance, this is the case in models with discrete degrees of freedom (e.g., Ising spins). And, indeed, for simulations of lattice models, MC is almost always the technique of choice. But even in off-lattice models with continuous degrees of freedom, it is sometimes better, or even essential, to use Monte Carlo sampling. Usually, the reason to choose the MC technique is that it allows us to perform *unphysical* trial moves, that is, moves that cannot occur in nature (and, therefore, have no counterpart in Molecular Dynamics) but are essential for the equilibration of the system.

This introduction is meant to place our discussion of Monte Carlo techniques for simulating complex fluids in a proper perspective: in most published simulations of complex (often macromolecular) fluids, Molecular Dynamics is used, and rightly so. The Monte Carlo techniques that we discuss here have been developed for situations where either MD cannot be used at all or the natural dynamics of the system is too slow to allow the system to equilibrate on the time scale of a simulation.

Examples of such simulations are Gibbs-ensemble and grand-canonical Monte Carlo simulations. Both techniques require the exchange of particles, either between a reservoir and the simulation box or between the two boxes. Such

particle exchanges are not related to any real dynamics and therefore require the use of Monte Carlo techniques. But, in the case of complex fluids, in particular, fluids consisting of chain molecules, the conventional Monte Carlo techniques for grand-canonical or Gibbs-ensemble simulations also fail. The reason is that, in the case of large molecules, the probability of acceptance of a random trial insertion in the simulation box is extremely small and hence the number of insertion attempts has to be made prohibitively large. For this reason, the early grand-canonical and Gibbs-ensemble simulations were limited to the study of adsorption and liquid-vapor phase equilibria of small molecules.

12.1 Biased sampling techniques

In this chapter, we discuss extensions of the standard Monte Carlo algorithm that allow us to overcome some of these limitations.¹ The main feature of these more sophisticated Monte Carlo trial moves is that they are no longer completely random: the moves are biased in such a way that the molecule to be inserted has an enhanced probability to “fit” into the existing configuration. In contrast, no information about the present configuration of the system is used in the generation of normal (unbiased) MC trial moves: that information is used only to accept or reject the move (see Chapters 3 and 6). Biasing a Monte Carlo trial move means that we are no longer working with a symmetric *a priori* transition matrix. To satisfy detailed balance, we therefore also should change the acceptance rules. Clearly, the price we pay for using configurationally biased MC trial moves is a greater complexity of our program. However, the reward is that, with the help of these techniques, we can sometimes speed up a calculation by many orders of magnitude. To illustrate this, we shall discuss examples of simulations that were made possible only through the use of biased sampling.

12.1.1 Beyond Metropolis

The general idea of biased sampling is best explained by considering a simple example. Let us assume that we have developed a Monte Carlo scheme that allows us to generate trial configurations with a probability that depends on the potential energy of that configuration:

$$\alpha(o \rightarrow n) = f[\mathcal{U}(n)].$$

For the reverse move, we have

$$\alpha(n \rightarrow o) = f[\mathcal{U}(o)].$$

Suppose we want to sample the N, V, T ensemble, which implies that we have to generate configurations with a Boltzmann distribution (6.2.1). Imposing de-

¹ Readers who are not familiar with the Rosenbluth scheme are advised to read section 10.2 first.

tailed balance (see section 6.1) yields, as a condition for the acceptance rule,

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{f[\mathcal{U}(o)]}{f[\mathcal{U}(n)]} \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}.$$

A possible acceptance rule that obeys this condition is

$$\text{acc}(o \rightarrow n) = \min\left(1, \frac{f[\mathcal{U}(o)]}{f[\mathcal{U}(n)]} \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\}\right). \quad (12.1.1)$$

This derivation shows that we can introduce an arbitrary biasing function $f(\mathcal{U})$ in the sampling scheme and generate a Boltzmann distribution of configurations, provided that the acceptance rule is modified in such a way that the bias is removed from the sampling scheme. Ideally, by biasing the probability to generate a trial conformation in the right way, we could make the term on the right-hand side of Eq. (12.1.1) always equal to unity. In that case, every trial move will be accepted. In section 13.4.2, we show that it is sometimes possible to achieve this ideal situation. However, in general, biased generation of trial moves is simply a technique for enhancing the acceptance of such moves without violating detailed balance.

We now give some examples of the use of non-Metropolis sampling techniques to demonstrate how they can be used to enhance the efficiency of a simulation.

12.1.2 Orientational bias

To perform a Monte Carlo simulation of molecules with an intermolecular potential that depends strongly on the relative molecular orientation (e.g., polar molecules, hydrogen-bond formers, liquid-crystal forming molecules), it is important to find a position that not only does not overlap with the other molecule but also has an acceptable orientation. If the probability of finding a suitable orientation by chance is very low, we can use biased trial moves to enhance the acceptance.

Algorithm

Let us consider a Monte Carlo trial move in which a randomly selected particle has to be moved and reoriented. We denote the old configuration by o and the trial configuration by n . We use standard random displacement for the translational parts of the move, but we bias the generation of trial orientations, as follows:

1. Move the center of mass of the molecule over a (small) random distance and determine all those interactions that do not depend on the orientations. These interactions are denoted by $u^{\text{pos}}(n)$. In practice, there may be several ways to separate the potential into orientation-dependent and orientation-independent parts.

2. Generate k trial orientations $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}$ and for each of these trial orientations, calculate the energy $u^{\text{or}}(\mathbf{b}_i)$.
3. We define the Rosenbluth² factor, W

$$W(n) = \sum_{j=1}^k \exp[-\beta u^{\text{or}}(\mathbf{b}_j)]. \quad (12.1.2)$$

Out of these k orientations, we select one, say, n , with a probability

$$p(\mathbf{b}_n) = \frac{\exp[-\beta u^{\text{or}}(\mathbf{b}_n)]}{\sum_{j=1}^k \exp[-\beta u^{\text{or}}(\mathbf{b}_j)]}. \quad (12.1.3)$$

4. For the old configuration, o , the part of the energy that does not depend on the orientation of the molecules is denoted by $u^{\text{pos}}(o)$. The orientation of the molecule in the old position is denoted by \mathbf{b}_o , and we generate $k - 1$ trial orientations denoted by $\mathbf{b}_2, \dots, \mathbf{b}_k$. Using these k orientations, we determine

$$W(o) = \exp[-\beta u^{\text{or}}(\mathbf{b}_o)] + \sum_{j=2}^k \exp[-\beta u^{\text{or}}(\mathbf{b}_j)]. \quad (12.1.4)$$

5. The move is accepted with a probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{W(n)}{W(o)} \exp\{-\beta[u^{\text{pos}}(n) - u^{\text{pos}}(o)]\} \right). \quad (12.1.5)$$

It is clear that Eq. (12.1.3) ensures that energetically favorable configurations are more likely to be generated. An example implementation of this scheme is shown in Algorithm 21. Next, we should demonstrate that the sampling scheme is correct.

Justification of algorithm

To show that the orientational-bias Monte Carlo scheme just described is correct, that is, generates configurations according to the desired distribution, it is convenient to consider lattice models and continuum models separately. For both cases, we assume that we work in the canonical ensemble, for which the distribution of configurations is given by Eq. (6.2.1)

$$\mathcal{N}(\mathbf{q}^N) \propto \exp[-\beta \mathcal{U}(\mathbf{q}^N)],$$

where $\mathcal{U}(\mathbf{q}^N)$ is the sum of orientational and non-orientational part of the energy:

$$\mathcal{U} = u^{\text{or}} + u^{\text{pos}}.$$

² Since this algorithm for biasing the orientation of the molecules is very similar to an algorithm developed by Rosenbluth and Rosenbluth in 1955 [438] for sampling configurations of polymers (see section 10.2), we refer to the factor W as the Rosenbluth factor.

Algorithm 21 (Orientational bias)

function orien_bias	Configurational-bias MC trial move to change orientation of molecule o
o = int($\mathcal{R} * npart$) + 1	Select a particle at random
sumw = 0	k, the number of trial directions, is arbitrary but fixed.
for $1 \leq j \leq k$ do	generate random trial direction
b(j) = ranor	calculate energy of trial orientation
eno = enero(x(o), o, b(j))	calculate Rosenbluth factor (12.1.2)
w(j) = exp(-beta * eno)	
sumw = sumw + w(j)	
enddo	
n = select(w, sumw)	select one of the orientations
bn = b(n)	n is the selected orientation
wn = sumw	Rosenbluth factor new orientation
sumw = 0	Next consider the old orientation.
for $1 \leq j \leq k$ do	consider k trial orientations
if j == 1 then	
b(j) = bu(o)	use actual orientation of particle o
else	
b(j) = ranor	generate a random orientation
endif	
eno = enero(x(o), b(j))	calculate energy of trial orientation j
sumw = sumw + exp(-beta * eno)	calculate Rosenbluth factor (12.1.4)
enddo	
wo = sumw	Rosenbluth factor old configuration
if $\mathcal{R} < wn/wo$ then	acceptance test (12.1.5)
bu(o) = bn	accepted
endif	
end function	

Specific Comments (for general comments, see p. 7)

1. The molecule originally has a position $x(o)$ and orientation $bu(o)$. In this example, we leave the position of the molecule unchanged.
2. The function **enero** computes the energy of a molecule o with position $x(o)$ and orientation b.
3. The function **ranor** generates a randomly oriented unit vector (Algorithm 38).
4. The function **select** selects one of the orientations with probability $p(n) = w(n) / \sum_j w(j)$ (see, Algorithm 37).

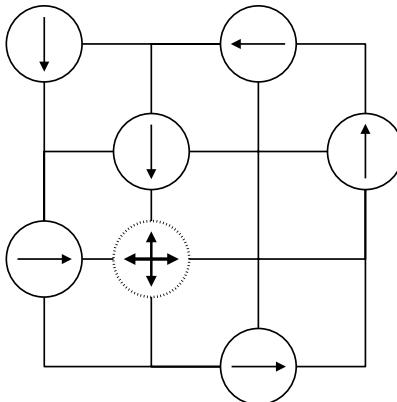


FIGURE 12.1 Lattice model in which the molecules can take four orientations (indicated by arrows, $k = 4$). The dotted circle indicates the trial position of the particle that we attempt to move.

We first consider a lattice model.

Lattice models

We assume that the molecules in our lattice model can have k discrete orientations (see Fig. 12.1). We impose the condition of detailed balance (6.1.1):

$$K(o \rightarrow n) = K(n \rightarrow o).$$

The flow of configurations o to n is (Eq. (6.1.2))

$$K(o \rightarrow n) = \mathcal{N}(o) \times \alpha(o \rightarrow n) \times \text{acc}(o \rightarrow n). \quad (12.1.6)$$

In the orientational-bias scheme, the probability of selecting conformation n is (see Eq. (12.1.3))

$$\alpha(o \rightarrow n) = \frac{\exp[-\beta u^{\text{or}}(n)]}{W(n)}.$$

Imposing detailed balance and substitution of the desired distribution for $\mathcal{N}(n)$ and $\mathcal{N}(o)$ imposes the following condition on the acceptance rules:

$$\begin{aligned} \frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} &= \frac{\exp[-\beta \mathcal{U}(n)]}{\exp[-\beta \mathcal{U}(o)]} \times \frac{\exp[-\beta u^{\text{or}}(o)]}{W(o)} \times \frac{W(n)}{\exp[-\beta u^{\text{or}}(n)]} \\ &= \frac{W(n)}{W(o)} \exp\{-\beta[u^{\text{pos}}(n) - u^{\text{pos}}(o)]\}. \end{aligned} \quad (12.1.7)$$

Acceptance rule (12.1.5) satisfies this condition. This demonstrates that for a lattice model detailed balance is fulfilled.

Continuum model

If the orientation of a molecule is described by a continuous variable, then there is an essential difference from the previous case. In the lattice model all the possible orientations can be considered explicitly, and the corresponding Rosenbluth factor can be calculated exactly. For the continuum case, we can never hope to sample *all* possible orientations. It is impossible to determine the exact Rosenbluth factor since an infinite number of orientations are possible.³ Hence, the scheme for lattice models, in which the Rosenbluth factor for all orientations is calculated, and cannot be used for a continuum model. A possible solution would be to use a large but finite number of trial directions. Surprisingly, this is not necessary. It is possible to devise a *rigorous* algorithm using an *arbitrary subset* of all possible trial directions. The answer we get does *not* depend on the number of trial directions we choose but the statistical accuracy does.

Let us consider the case in which we use a set of k trial orientations; this set is denoted by

$$\{\mathbf{b}\}_k = \{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k\}.$$

Conformation \mathbf{b}_n can be selected only if it belongs to the set $\{\mathbf{b}\}_k$. The set of all sets $\{\mathbf{b}\}_k$ that includes conformation n is denoted by

$$\mathcal{B}_n = \{\{\mathbf{b}\}_k \mid \mathbf{b}_n \in \{\mathbf{b}\}_k\}.$$

Every element of \mathcal{B}_n can be written as (\mathbf{b}_n, b^*) , where b^* is the set of $k - 1$ additional trial orientations. In the flow of configuration o to n , we have to consider the sum over all sets in \mathcal{B}_n

$$K(o \rightarrow n) = \mathcal{N}(o) \sum_{i \in \mathcal{B}_n} \alpha(o \rightarrow n, i) \times \text{acc}(o \rightarrow n, i), \quad (12.1.8)$$

in which the probability of generating configuration n and the acceptance depends on the particular set of trial orientations i .

Similarly, for the reverse move, we define the set \mathcal{B}_o

$$\mathcal{B}_0 = \{\{\mathbf{b}\}_k \mid \mathbf{b}_o \in \{\mathbf{b}\}_k\},$$

for which each element can be written as (\mathbf{b}_o, b'^*) . The expression for the reverse flow then becomes

$$K(n \rightarrow o) = \mathcal{N}(n) \sum_{j \in \mathcal{B}_o} \alpha(n \rightarrow o, j) \times \text{acc}(n \rightarrow o, j). \quad (12.1.9)$$

It should be stressed that infinitely many different sets of orientations include \mathbf{b}_n , and the same holds for sets that include \mathbf{b}_o . Moreover, the probability of selecting \mathbf{b}_n from such a set depends on the remainder of the set b^* (see Fig. 12.2). Hence, the acceptance probability must also depend on the sets b^* and b'^* .

³ In Example 16 we discuss a special case for which the Rosenbluth factor *can* be calculated exactly.

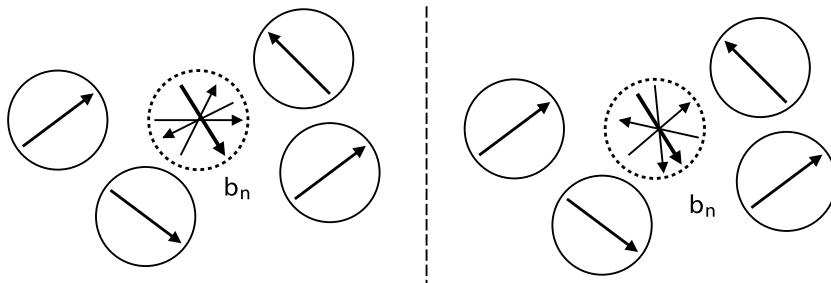


FIGURE 12.2 Continuum model in which the molecule can have an arbitrary orientation (indicated by arrows). The figure shows two different sets of four trial orientations that both include orientation \mathbf{b}_n .

Detailed balance is certainly obeyed if we impose a much stronger condition, “super-detailed balance,” which states that for every particular choice of the sets b^* and b'^* , detailed balance should be obeyed,

$$\begin{aligned} K(o \rightarrow n, b^*, b'^*) &= K(n \rightarrow o, b'^*, b^*), \\ \mathcal{N}(o) \alpha(o \rightarrow n, b^*, b'^*) \text{acc}(o \rightarrow n, b^*, b'^*) \\ &= \mathcal{N}(n) \alpha(n \rightarrow o, b'^*, b^*) \text{acc}(n \rightarrow o, b'^*, b^*), \end{aligned} \quad (12.1.10)$$

in which b^* and b'^* are two sets of $k - 1$ arbitrary additional trial orientations. It may seem strange that the sets b^* and b'^* show up on *both* sides of the equations. However, bear in mind that to decide on the acceptance of the forward move, one should generate both the set b^* that includes the new orientation *and* the set b'^* around the old orientation. Hence, the construction of a trial move includes both sets of trial orientations. As the probabilities of generating b^* and b'^* appear on both sides of the equations, they cancel each other. Moreover, the *a priori* probability of generating a random orientation \mathbf{b}_n in the forward move is equal to the *a priori* probability of generating \mathbf{b}_o in the reverse move. So these generation probabilities also cancel each other. This leads to a great simplification of the acceptance criterion. For the canonical ensemble, substitution of Eqs. (12.1.2) and (12.1.3) yields

$$\begin{aligned} \frac{\text{acc}(o \rightarrow n, b^*, b'^*)}{\text{acc}(n \rightarrow o, b'^*, b^*)} &= \frac{\exp[-\beta \mathcal{U}(n)]}{\exp[-\beta \mathcal{U}(o)]} \frac{\exp[-\beta u^{\text{or}}(o)]}{W(\mathbf{b}_o, b'^*)} \frac{W(\mathbf{b}_n, b^*)}{\exp[-\beta u^{\text{or}}(n)]} \\ &= \frac{W(\mathbf{b}_n, b^*)}{W(\mathbf{b}_o, b'^*)} \exp\{-\beta[u^{\text{pos}}(n) - u^{\text{pos}}(o)]\}. \end{aligned} \quad (12.1.11)$$

As acceptance rule (12.1.5) satisfies this condition, detailed balance is indeed obeyed.

Note that, in this demonstration, we did not have to assume that the number of trial orientations k had to be large. In fact, the result is *independent* of the number of trial orientations.

Illustration 16 (Dipoles embedded in spherical atoms). In systems with dipoles, the energy depends on the mutual orientation of the molecules and a bias in the sampling of the orientation can be useful. For models of dipoles embedded in an otherwise spherical particle (e.g., the dipolar hard-sphere fluid) the scheme of section 12.1.2 can be implemented elegantly as pointed out by Caillol [237]. In Eqs. (12.1.2) and (12.1.4), the Rosenbluth factor, W , is calculated by sampling k trial orientations. For a dipolar hard sphere (or any point dipole), we can calculate the Rosenbluth factors exactly once the electric field (\mathbf{E}) at the position of the inserted particle and that at the position of the old configuration are known:

$$\begin{aligned} W(\mathbf{r}) &= \int d\mathbf{b} \exp[-\beta\mu \cdot \mathbf{E}(\mathbf{r})] \\ &= \frac{\sinh[\beta|\mu||\mathbf{E}(\mathbf{r})|]}{\beta|\mu||\mathbf{E}(\mathbf{r})|}, \end{aligned}$$

where μ is the dipole moment of the molecule.^a A trial orientation can now be drawn directly from the distribution

$$p(\mathbf{r}, \omega) = \frac{\exp[-\beta\mu \cdot \mathbf{E}(\mathbf{r})]}{W(r)}.$$

^a In fact, there is a subtlety with this expression. It assumes that the component of the local electric field in the direction of the dipole does not depend on the orientation of the dipole. This seems obvious. But, in the case of an Ewald summation, where the long-range interaction of a molecule with its periodic images is represented by a Fourier sum, this condition is not quite satisfied.

12.2 Chain molecules

The sampling of equilibrium conformations of polymers is usually time consuming. The main reason is that the natural dynamics of polymers is dominated by topological constraints (chains cannot cross) and hence any algorithm based on the real motion of macromolecules will suffer from the same problem. For this reason, many “unphysical” Monte Carlo trial moves have been proposed to speed up the sampling of polymer conformations (see, e.g., [449]). An early Monte Carlo scheme to simulate polymers is the “reptation” method of Kron [508]. In this algorithm, a trial move consists of a Metropolis-style trial move to remove a monomer from one end of a linear chain and add it to the other end. The original reptation method could only be used for homopolymers. However, this limitation was overcome in the “wormhole”-move approach of Houdayer [509]. In Houdayer’s approach, reptation-style exchanges of monomers between the two ends of a chain are continued until the whole chain has been rebuilt in its original sequence.

In the present section we introduce the [Configurational-Bias Monte Carlo \(CBMC\)](#) scheme [444, 447, 510, 511]. This simulation technique can be used for

systems where it is less practical to change the conformation of a macromolecule by successive small steps.

12.2.1 Configurational-bias Monte Carlo

The starting point for the configurational-bias Monte Carlo technique is the scheme introduced by Rosenbluth and Rosenbluth in 1955 [438]. The Rosenbluth scheme itself was also designed as a method to sample polymer conformations.⁴ A drawback of the Rosenbluth scheme is, however, that it generates an unrepresentative sample of all polymer conformations; that is, the probability of generating a particular conformation using this scheme is *not* proportional to its Boltzmann weight. Rosenbluth and Rosenbluth corrected for this bias in the sampling of polymer conformations by introducing a conformation-dependent weight factor W . However, as was shown in detail by Batoulis and Kremer [450], this correction procedure, although correct in principle, in practice works only for relatively short chains (see Illustration 14).

The solution to this problem is to bias the Rosenbluth sampling in such a way that the correct (Boltzmann) distribution of chain conformations is recovered in a Monte Carlo sequence. In the configurational-bias scheme to be discussed next, the Rosenbluth weight is used to bias the *acceptance* of trial conformations generated by the Rosenbluth procedure. As we shall show, this guarantees that all chain conformations are generated *with the correct Boltzmann weight*.

12.2.2 Lattice models

Algorithm

The configurational-bias Monte Carlo algorithm consists of the following steps:

1. Generate a trial conformation using the Rosenbluth scheme (see Fig. 12.3, left) to grow the entire molecule, or part thereof, and compute its Rosenbluth weight $W(n)$.
2. “Retrace” the old conformation (see Fig. 12.3, right) and determine its Rosenbluth factor.
3. Accept the trial move with a probability

$$\text{acc}(o \rightarrow n) = \min[1, W(n)/W(o)]. \quad (12.2.1)$$

The generation of a trial conformation n of a polymer consisting of ℓ monomers is generated using an algorithm based on the method of Rosenbluth and Rosenbluth (see Fig. 12.3):

⁴ The Rosenbluth scheme is discussed in some detail in the context of a free energy calculation of a chain molecule in Chapter 10.

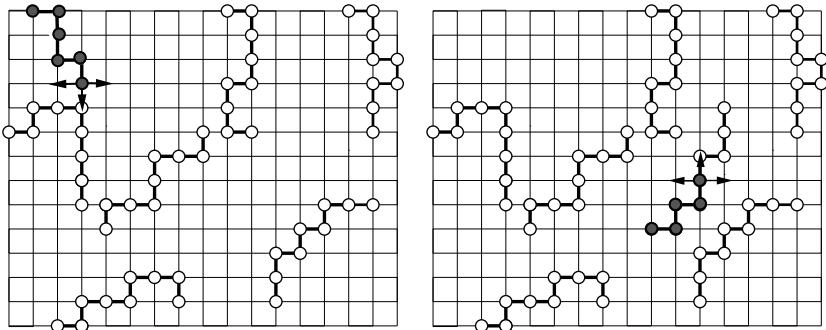


FIGURE 12.3 Sketch of the configurational-bias Monte Carlo scheme. The left figure shows the generation of a new conformation and the right figure shows the retracing of the old conformation. The arrows indicate the three trial positions.

1. The first atom is inserted at random, and its energy is denoted by $u_1(n)$, and⁵ $w_1(n) = k \exp[-\beta u_1(n)]$, where k is the coordination number of the lattice, for example, $k = 6$ for a simple cubic lattice.
2. For the next segment, with index i , there are k possible trial directions. The energy of trial direction j is denoted by $u_i(j)$. From the k possible directions, we select one, say, n , with a probability

$$p_i(n) = \frac{\exp[-\beta u_i(n)]}{w_i(n)}, \quad (12.2.2)$$

where $w_i(n)$ is defined as

$$w_i(n) = \sum_{j=1}^k \exp[-\beta u_i(j)]. \quad (12.2.3)$$

The interaction energy $u_i(j)$ includes all interactions of segment i with other molecules in the system and with segments 1 through $i - 1$ of the same molecule. It does not include the interactions with segments $i + 1$ to ℓ . Hence, the total energy of the chain is given by $\mathcal{U}(n) = \sum_{i=1}^{\ell} u_i(n)$.

3. Step 2 is repeated until the entire chain is grown and we can determine the Rosenbluth factor, W , of configuration n :

$$W(n) = \prod_{i=1}^{\ell} w_i(n). \quad (12.2.4)$$

Similarly, to determine the Rosenbluth factor of the old configuration, o , we use the following steps (see Fig. 12.3).

⁵ The factor k in the definition of the Rosenbluth weight of the first segment, strictly speaking, is unnecessary. We introduce it only here to make the subsequent notation more compact.

1. One of the chains is selected at random. This chain is denoted by o .
2. We measure the energy of the first monomer $u_1(o)$ and compute $w_1(o) = k \exp[-\beta u_1(o)]$.
3. To compute the Rosenbluth weight for the remainder of the chain, we determine the energy of monomer i at its actual position, and also the energy it would have had if been placed in any of the other $k - 1$ sites neighboring the actual position of monomer $i - 1$ (see Fig. 12.3). These energies are used to calculate

$$w_i(o) = \exp[-\beta u_i(o)] + \sum_{j=2}^k \exp[-\beta u_i(j)].$$

4. Once the entire chain has been retraced, we determine its Rosenbluth factor:

$$W(o) = \prod_{i=1}^{\ell} w_i(o). \quad (12.2.5)$$

Finally, the trial move from o to n is accepted with a probability given by

$$\text{acc}(o \rightarrow n) = \min[1, W(n)/W(o)]. \quad (12.2.6)$$

A schematic example of the implementation of this scheme is given in Algorithms 22 and 23. We now have to demonstrate that the acceptance rule (12.2.6) correctly removes the bias of generating new segments in the chain introduced by using Eq. (12.2.2).

Justification of the algorithm

The demonstration that this algorithm samples a *Boltzmann* distribution is similar to the one for the orientational-bias algorithm for lattice models (section 12.1.2).

The probability of generating a particular conformation n follows from the repetitive use of Eq. (12.2.2):

$$\alpha(o \rightarrow n) = \prod_{i=1}^{\ell} \frac{\exp[-\beta u_i(n)]}{w_i(n)} = \frac{\exp[-\beta \mathcal{U}(n)]}{W(n)}. \quad (12.2.7)$$

Similarly, for the reverse move,

$$\alpha(n \rightarrow o) = \frac{\exp[-\beta \mathcal{U}(o)]}{W(o)}. \quad (12.2.8)$$

The requirement of detailed balance (6.1.1) imposes the following condition on the acceptance criterion:

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{W(n)}{W(o)}. \quad (12.2.9)$$

Algorithm 22 (Basic configurational-bias Monte Carlo)

function CBMC	carry out CBMC trial move
new_conf=.false.	first retrace (part of) the old conf.
wo = grow (new_conf)	to calculate its Rosenbluth factor
new_conf=.true.	next consider the new configuration
wn = grow (new_conf)	grow (part of) a chain and calculate
if $\mathcal{R} < \text{wn}/\text{wo}$ then	the Rosenbluth factor of the new conf.
accept	acceptance test (12.2.6)
endif	accept and do bookkeeping
end function	

Specific Comments (for general comments, see p. 7)

1. This algorithm shows the basic structure of the configurational-bias Monte Carlo method. The details of the model are considered in the function **grow** (see Algorithm 23 for a polymer on a lattice).
2. The function **accept** takes care of the bookkeeping of the new configuration.

Clearly, the proposed acceptance criterion (12.2.6) satisfies this condition.

It should be stressed that the value of factor $W(o)$ depends on the direction in which the old configuration is retraced: if we start from monomer 1, we find a different numerical value for $W(o)$ than if we start from monomer ℓ . As a consequence the probability of such a move depends on the way the factor $W(o)$ has been calculated. Although such a dependence is at first sight counterintuitive, both ways of retracing the old conformation—starting with monomer 1 or with monomer ℓ —result in the correct distribution of states, as long as both ways occur with equal probability during the simulation. This is automatically satisfied in the case of linear chains of identical segments where the labeling of the terminal groups is completely arbitrary.

12.2.3 Off-lattice case

Next, we consider configurational-bias Monte Carlo for off-lattice systems. As with the orientational moves described in section 12.1.2, some aspects in a continuum version of configurational-bias Monte Carlo require special attention. In section 12.1.2 we already showed that it may be possible to develop a configurational-bias sampling scheme even when it is impossible to calculate the Rosenbluth factor exactly. For chain molecules, we can follow basically the same approach.

The other important point that we have to consider is the way in which trial conformations of a chain molecule are generated. In a lattice model, the num-

Algorithm 23 (Growing a chain on a lattice)

function grow(new_conf, w)	grow an ℓ -bead polymer on a lattice with coordination number k.
	the function returns the Rosenbluth factor w
if new_conf then	new_conf is a logical variable (see Notes)
xn(1)=R*box	insert the first monomer
else	
o=R*npart+1	select old chain at random
xn(1)=x(o,1)	
endif	
en = ener(xn(1),o)	calculate energy of 1st monomer
w=k*exp(-beta*en)	Rosenbluth factor first monomer
for 2 \leq i \leq ell do	
sumw=0	
for 1 \leq j \leq k do	consider the k trial directions
xt(j)=xn(i-1)+b(j)	determine trial position
en = ener(xt(j),o)	determine energy trial position j
w(j)=exp(-beta*en)	
sumw=sumw+w(j)	
enddo	
if new_conf then	select one of the trial position
n = select(w,sumw)	direction n is selected
xn(i)=xt(n)	
else	
xn(i)=x(o,i)	
endif	
w=w*sumw	update Rosenbluth factor
enddo	
end function	

Specific Comments (for general comments, see p. 7)

1. If new_conf == .true. generate a new configuration, if new_conf == .false. retrace an old one.
2. In a lattice model we consider all possible trial positions, denoted by the fixed set b(j). Therefore, for the old conformation, the actual position is automatically included.
3. The function select (Algorithm 37) selects one of the trial positions with probability $p(i) = w(i)/\sum_j w(j)$. The function ener calculates the energy of the monomer at the given position with the other polymers and the monomers of the chain that already have been grown.

ber of trial conformations is dictated by the lattice. In an off-lattice system, one could generate trial segments with orientations distributed uniformly on a unit sphere. However, for many models of interest, this procedure is not very efficient, in particular when there are strong intramolecular interactions (e.g., bending and torsion potentials). The efficiency of a configurational-bias Monte Carlo algorithm depends to a large extent on the method used for generating the trial orientations. For example, an isotropic distribution of trial directions is well-suited for completely flexible chains. In contrast, for a stiff chain (e.g., liquid-crystal forming polymer), such a trial position will almost always be rejected because of the intramolecular interactions.

Algorithm

From the preceding discussion, it follows that the intramolecular interactions should be taken into account in generating the set of trial conformations. Here, we consider the case of a flexible molecule *with* contributions to the internal energy due to bond bending and torsion. The fully flexible case then follows trivially. Consider a chain of ℓ linear segments, the potential energy of a given conformation \mathcal{U} has two contributions:

1. The *bonded potential energy* $\mathcal{U}^{\text{bond}}$ is equal to the sum of the contributions of the individual joints. A joint between segments i and $i + 1$ (say) has a potential energy u_i^{bond} that depends on the angle θ between the successive segments. For instance, $u_i^{\text{bond}}(\theta)$ could be of the form $u_i^{\text{bond}}(\theta) = k_\theta(\theta - \theta_0)^2$. For realistic models for polyatomic molecules, u_i^{bond} includes all local bonded potential energy changes due to the bending and torsion of the bond from atom $i - 1$ to atom i .
2. The *external potential energy* \mathcal{U}^{ext} accounts for all interactions with other molecules and for all the nonbonded intramolecular interactions. In addition, interactions with any external field that may be present are also included in \mathcal{U}^{ext} .

In what follows, we shall denote a chain in the absence of the external interactions as the *ideal* chain. Note that this is a purely fictitious concept, as real chains always have nonbonded intramolecular interactions.

To perform a configurational-bias Monte Carlo move, we apply the following “recipe” to construct a conformation of a chain of ℓ segments. The construction of chain conformations proceeds segment by segment. Let us consider the addition of one such segment. To be specific, let us assume that we have already grown $i - 1$ segments and are trying to add segment i . This is done in two steps. First, we generate a trial conformation n , next we consider the old conformation o . A trial conformation is generated as follows:

1. Generate a fixed number of, say, k trial segments. The orientations of the trial segments are distributed according to the Boltzmann weight associated with the bonded interactions of monomer i (u_i^{bond}). We denote this set of k

different trial segments by

$$\{\mathbf{b}\}_k = \{\mathbf{b}_1, \dots, \mathbf{b}_k\},$$

where the probability of generating a trial segment \mathbf{b} is given by

$$p_i^{\text{bond}}(\mathbf{b})d\mathbf{b} = \frac{\exp[-\beta u_i^{\text{bond}}(\mathbf{b})]d\mathbf{b}}{\int d\mathbf{b} \exp[-\beta u_i^{\text{bond}}(\mathbf{b})]} = C \exp[-\beta u_i^{\text{bond}}(\mathbf{b})]d\mathbf{b}. \quad (12.2.10)$$

2. For all k trial segments, we compute the external Boltzmann factors $\exp[-\beta u_i^{\text{ext}}(\mathbf{b}_i)]$, and out of these, we select one, denoted by n , with a probability

$$p_i^{\text{ext}}(\mathbf{b}_n) = \frac{\exp[-\beta u_i^{\text{ext}}(\mathbf{b}_n)]}{w_i^{\text{ext}}(n)}, \quad (12.2.11)$$

where we have defined

$$w_i^{\text{ext}}(n) = \sum_{j=1}^k \exp[-\beta u_i^{\text{ext}}(\mathbf{b}_j)]. \quad (12.2.12)$$

3. The selected segment n becomes the i th segment of the trial conformation of the chain.
4. When the entire chain is grown, we calculate the Rosenbluth factor of the chain:

$$W^{\text{ext}}(n) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(n), \quad (12.2.13)$$

where the Rosenbluth factor of the first monomer is defined by

$$w_1^{\text{ext}}(n) = k \exp[-\beta u_1^{\text{ext}}(\mathbf{r}_1)], \quad (12.2.14)$$

where \mathbf{r}_1 is the position of the first monomer.

For the old configuration, a similar procedure to calculate its Rosenbluth factor is used.

1. One of the chains is selected at random. This chain is denoted o .
2. The external energy of the first monomer is calculated. This energy involves only the external interactions. The Rosenbluth weight of this first monomer is given by

$$w_1^{\text{ext}}(o) = k \exp[-\beta u_1^{\text{ext}}(o)]. \quad (12.2.15)$$

3. The Rosenbluth factors of the other $\ell - 1$ segments are calculated as follows. We consider the calculation of the Rosenbluth factor of segment i . We generate a set of $k - 1$ orientations with a distribution prescribed by the bonded

interactions (12.2.10). These orientations, together with the actual bond between segment $i - 1$ and i , form the set of k orientations $(\mathbf{b}_o, \mathbf{b}'^*)$. These orientations are used to calculate the external Rosenbluth factor:

$$w_i^{\text{ext}}(o) = \sum_{j=1}^k \exp[-\beta u_i^{\text{ext}}(\mathbf{b}_j)]. \quad (12.2.16)$$

4. For the entire chain the Rosenbluth factor of the old conformation is defined by

$$W^{\text{ext}}(o) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(o). \quad (12.2.17)$$

After the new configuration has been generated and the Rosenbluth factor of the old configuration has been calculated, the move is accepted with a probability

$$\text{acc}(o \rightarrow n) = \min[1, W^{\text{ext}}(n)/W^{\text{ext}}(o)]. \quad (12.2.18)$$

We still have to show that this sampling scheme is correct.

Justification of algorithm

Comparison with the lattice version shows that for the off-lattice case, two aspects are different. First, for a model with continuous degrees of freedom, we cannot calculate the Rosenbluth factor exactly. This point has been discussed in detail in section 12.1.2 for the orientational-bias scheme. As in section 12.1.2, we impose super-detailed balance. Second, the way in which we generate trial conformations is different for off-lattice than for lattice models. In a lattice model there is no need to separate the interactions into bonded and external ones. We have to show that the way in which we treat bonded interactions does not perturb the sampling.

The probability of generating a chain of length ℓ is the product of the probability of generating a trial orientation (12.2.10) and the probability of selecting this orientation (12.2.11); for all monomers, this gives, as a probability of generating conformation n ,

$$\alpha(o \rightarrow n) = \prod_{i=1}^{\ell} p_i(o \rightarrow n) = \prod_{i=1}^{\ell} p_i^{\text{bond}}(n) p_i^{\text{ext}}(n). \quad (12.2.19)$$

In the following, we consider the expressions for one of the ℓ segments, to keep the equations simple. A given set of k trial orientations, which includes orientation n , is denoted by (\mathbf{b}_n, b^*) (see section 12.1.2). As before, we stress that the generation of the additional trial orientations (b'^*) around the old segment (\mathbf{b}_o) is an essential part of the *generation* of the trial move. We denote the probability

of generating the combined set b^*, b'^* by

$$\mathcal{P}^{\text{bond}}(b^*, b'^*).$$

Hence, the flow of configurations is given by

$$\begin{aligned} K(o \rightarrow n, b^*, b'^*) &= \mathcal{N}(o) \times \alpha(o \rightarrow n, b^*, b'^*) \times \text{acc}(o \rightarrow n, b^*, b'^*) \\ &= \exp[-\beta u(o)] \times C \exp[-\beta u^{\text{bond}}(n)] \times \frac{\exp[-\beta u^{\text{ext}}(n)]}{w^{\text{ext}}(\mathbf{b}_n, b^*)} \\ &\quad \times \text{acc}(o \rightarrow n, b^*, b'^*) \mathcal{P}^{\text{bond}}(b^*, b'^*). \end{aligned} \quad (12.2.20)$$

For the reverse move, we have

$$\begin{aligned} K(n \rightarrow o, b'^*, b^*) &= \mathcal{N}(n) \times \alpha(n \rightarrow o, b'^*, b^*) \times \text{acc}(n \rightarrow o, b'^*, b^*) \\ &= \exp[-\beta u(n)] \times C \exp[-\beta u^{\text{bond}}(o)] \times \frac{\exp[-\beta u^{\text{ext}}(o)]}{w^{\text{ext}}(\mathbf{b}_o, b'^*)} \\ &\quad \times \text{acc}(n \rightarrow o, b'^*, b^*) \mathcal{P}^{\text{bond}}(b^*, b'^*). \end{aligned} \quad (12.2.21)$$

Recall that the total energy of a monomer is the sum of the bonded and external contributions:

$$u(n) = u^{\text{bond}}(n) + u^{\text{ext}}(n).$$

We now impose super-detailed balance (12.1.10). The factors $\mathcal{P}^{\text{bond}}(b^*, b'^*)$ on both sides of the equation cancel each other, and we get the following simple criterion for the acceptance rule:

$$\frac{\text{acc}(o \rightarrow n, b^*, b'^*)}{\text{acc}(n \rightarrow o, b'^*, b^*)} = \frac{w^{\text{ext}}(\mathbf{b}_n, b^*)}{w^{\text{ext}}(\mathbf{b}_o, b'^*)}. \quad (12.2.22)$$

This demonstration was only for a single segment in a chain. For the entire chain, the corresponding acceptance criterion is obtained analogously. It is simply the product of the terms for all segments⁶:

$$\frac{\text{acc}[o \rightarrow n, (b_1^*, \dots, b_\ell^*)]}{\text{acc}[n \rightarrow o, (b'_1^*, \dots, b'_\ell^*)]} = \frac{\prod_{i=1}^\ell w_i^{\text{ext}}(\mathbf{b}_n, b^*)}{\prod_{i=1}^\ell w_i^{\text{ext}}(\mathbf{b}_o, b'^*)} = \frac{W[n, (b^*_1, \dots, b^*_\ell)]}{W[o, (b'^*_1, \dots, b'^*_\ell)]}. \quad (12.2.23)$$

And, indeed, our acceptance rule (12.2.18) satisfies this condition. The equation shows that, because the trial orientations are generated with a probability (12.2.10) prescribed by the bonded energy, this energy does *not* appear in the acceptance rules. In Case Study 19, a detailed discussion is given on the advantages of this approach. It is important to note that we do not need to know the normalization constant C of Eq. (12.2.10).

⁶ The computational efficiency may be enhanced by exploiting an early-rejection scheme similar to the one discussed in section 13.4.3

The basic structure of an algorithm for configurational-bias Monte Carlo for continuum models is very similar to the lattice version (Algorithm 22); the main difference is the way in which configurations are generated.

Example 20 (Equation of state of Lennard-Jones chains). To illustrate the configurational-bias Monte Carlo technique described in this section, we determine the equation of state of a system consisting of eight-bead chains of Lennard-Jones particles. The nonbonded interactions are described by a truncated and shifted Lennard-Jones potential. The potential is truncated at $R_c = 2.5\sigma$. The bonded interactions are described with a harmonic spring

$$u^{\text{vib}}(l) = \begin{cases} 0.5k_{\text{vib}}(l - 1)^2 & 0.5 \leq l \leq 1.5 \\ \infty & \text{otherwise} \end{cases},$$

where l is the bond length, the equilibrium bond length has been set to 1, and $k_{\text{vib}} = 400$.

The simulations are performed in cycles. In each cycle, we perform on average N_{dis} attempts to displace a particle, N_{cbmc} attempts to (partly) regrow a chain, and N_{vol} attempts to change the volume (only in the case of N, P, T simulations). If we regrow a chain, the configurational-bias Monte Carlo scheme is used. In this move we select at random the monomer from which we start to regrow. If this happens to be the first monomer, the entire molecule is regrown at a random position. For all the simulations, we used eight trial orientations. The lengths of trial bonds are generated with a probability prescribed by the bond-stretching potential (see Case Study 19).

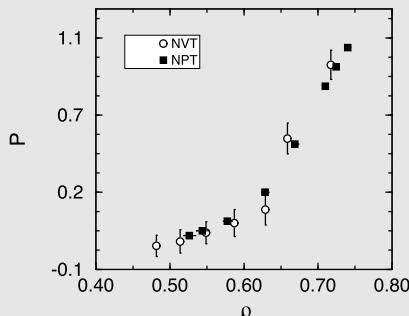


FIGURE 12.4 Equation of state of an eight-bead Lennard-Jones chain as obtained from N, V, T and N, P, T simulations using the configurational-bias Monte Carlo scheme. The simulations are performed with 50 chains at a temperature $T = 1.9$.

In Fig. 12.4 the equation of state as obtained from N, V, T simulations is compared with one obtained from N, P, T simulations. This isotherm is well above the critical temperature of the corresponding monomeric fluid ($T_c = 1.085$, see Fig. 3.3), but the critical temperature of the chain molecules is appreciably higher [512].

For more details, see SI (Case Study 18).

12.3 Generation of trial orientations

The efficient generation of good trial conformations is an essential aspect of the configurational-bias Monte Carlo scheme for continuum models with strong intramolecular interactions. For some models (for example, Gaussian chains) it is possible to generate this distribution directly. For an arbitrary model, we can use the acceptance-rejection technique [38] of generating the trial orientations.

Here, we show how a rejection technique can be used to generate trial positions efficiently. The number of trial directions in the CBMC scheme can be chosen at will. Often, the optimal number of trial directions is determined empirically. However, more systematic techniques exist to compute this optimal number [513].

12.3.1 Strong intramolecular interactions

Let us consider as an example a model of a molecule in which the bonded interactions include bond stretching, bond bending, and torsion. The external interactions are the nonbonded interactions. A united atom model of an alkane is a typical example of such a molecule.

The probability that we generate a trial configuration \mathbf{b} is given by, (see Eq. (12.2.10))

$$P(\mathbf{b})d\mathbf{b} = C \exp[-\beta u^{\text{bond}}(\mathbf{b})]d\mathbf{b}. \quad (12.3.1)$$

It is convenient to represent the position of an atom using the bond length r , bond angle θ , and torsional angle ϕ (see Fig. 12.5). With these coordinates the volume element $d\mathbf{b}$ is given by

$$d\mathbf{b} = r^2 dr d\cos\theta d\phi. \quad (12.3.2)$$

The bonded energy is the sum of the bond-stretching potential, the bond-bending potential, and the torsion potential:

$$u^{\text{bond}}(r, \theta, \phi) = u_{\text{vib}}(r) + u_{\text{bend}}(\theta) + u_{\text{tors}}(\phi). \quad (12.3.3)$$

Substitution of Eqs. (12.3.3) and (12.3.2) into Eq. (12.3.1) gives

$$\begin{aligned} P(\mathbf{b}) d\mathbf{b} &= P(r, \theta, \phi) r^2 dr d\cos\theta d\phi \\ &= C \exp[-\beta u_{\text{vib}}(r)] r^2 dr \times \exp[-\beta u_{\text{bend}}(\theta)] d\cos\theta \\ &\quad \times \exp[-\beta u_{\text{tors}}(\phi)] d\phi. \end{aligned} \quad (12.3.4)$$

Many models use a fixed bond length, in which case the first term in Eq. (12.3.4) is a constant.

Let us consider the molecule shown in Fig. 12.5. The first atom is placed at a random position and we now have to add the second atom. For convenience, it is assumed that the model has a fixed bond length. The second atom has no bonded

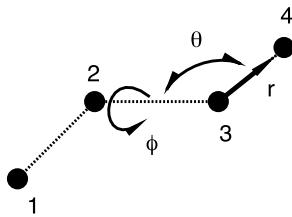


FIGURE 12.5 Schematic sketch of a part of a molecule.

interactions other than the constraints on the bond length. The distribution of trial orientations, Eq. (12.3.4), reduces to

$$P_2(\mathbf{b})d\mathbf{b} \propto d \cos \theta d\phi. \quad (12.3.5)$$

Hence, the trial orientations are randomly distributed on the surface of a sphere (such a distribution can be generated with Algorithm 38 in Appendix J).

For the third atom, the bonded energy contains the bond-bending energy as well. This gives, for the distribution of trial orientations,

$$P_3(\mathbf{b})d\mathbf{b} \propto \exp[-\beta u_{\text{bend}}(\theta)] d \cos \theta d\phi. \quad (12.3.6)$$

To generate k trial orientations distributed according to Eq. (12.3.6), we again generate a random vector on a unit sphere and determine the angle θ . This vector is accepted with a probability $\exp[-\beta u_{\text{bend}}(\theta)]$. If rejected, this procedure is repeated until a value of θ has been accepted. In [38], this acceptance-rejection method is shown to indeed give the desired distribution of trial orientations. In this way, k (or $k - 1$, for the old conformation) trial orientations are generated.

An alternative scheme would be to generate angle θ uniformly ($\theta \in [0, \pi]$) and to determine the bond-bending energy corresponding to this angle. This angle θ is accepted with a probability $\sin(\theta) \exp[-\beta u_{\text{bend}}(\theta)]$. If rejected, this procedure is repeated until a value of θ has been accepted. The selected value of θ is supplemented with a randomly selected angle ϕ . These two angles determine a new trial orientation.

For the fourth and higher atoms, the bonded energy includes both bond-bending and torsion energy. This gives, for Eq. (12.3.4),

$$p_l^{\text{bond}}(\mathbf{b})d\mathbf{b} \propto \exp[-\beta u_{\text{bend}}(\theta)] \exp[-\beta u_{\text{tors}}(\phi)] d \cos \theta d\phi. \quad (12.3.7)$$

We again generate a random vector on a sphere and calculate the bond-bending angle θ and torsion ϕ . These angles are accepted with a probability $\exp\{-\beta[u_{\text{bend}}(\theta) + u_{\text{tors}}(\phi)]\}$. If these angles are rejected, new vectors are generated until one gets accepted.

Again an alternative scheme is to determine first a bond-bending angle θ by generating θ uniformly on $[0, \pi]$ and calculating the bond-bending energy corresponding to this angle. This angle θ is then accepted with a probability

Algorithm 24 (Growing an “alkane”)

function grow(new_conf,w)	grow or retrace an “alkane” and calculate its Rosenbluth factor w
if new_conf == .true. then	new_conf =.true.: new conf.
ib=int($\mathcal{R} * \ell$)+1	start growth from position $ib \leq \ell$
ibnewconf=ib	store starting position
else	new_conf =.false.: old conf.
ib=ibnewconf	same starting position to regrow as used for the new configuration
endif	
for 1 ≤ i ≤ b-1 do	store positions that are not regrown
xn(i)=x(i)	
enddo	
w=1	
for ib ≤ i ≤ ℓ do	
if ib == 1 then	first atom
if new_conf == .true. then	generate random position
xt(1)= $\mathcal{R} * \text{box}$	
else	use old position
xt(1)=xn(1)	
endif	calculate (external) energy
eni = enerex(xt(1))	and Rosenbluth factor
w=k*exp(-beta*eni)	second and higher atoms
else	
sumw=0	
for 1 ≤ j ≤ k do	
if new_conf == .false.	actual position as trial orientation
& j == 1) then	
xt(1)=x(i)	generate trial position
else	(external) energy of this position
xt(j) = next_ci(xn,i)	
endif	
eni= enerex(xt(j))	
wt(j)= exp(-beta*eni)	
sumw=sumw+wt(j)	
enddo	
w=w*sumw	update Rosenbluth factor
... (continue)....	

$\sin(\theta) \exp[-\beta u_{\text{bend}}(\theta)]$. This procedure is continued until we have accepted an angle. Next we generate a torsion angle randomly on $[0, 2\pi]$ and accept this angle with a probability $\exp[-\beta u_{\text{tors}}(\phi)]$, again repeating this until a value has been accepted. In this scheme, the bond angle and torsion are generated indepen-

```

...(continue)....
if new_conf == .true. then
    n = select(wt,sumw)           select one of the trial orientations
    xn(i)=xt(n)
    xstore(i)=xt(n)               store selected configuration
    else                         for bookkeeping
        xn(i)=x(i)
    endif
    endif
end function

```

Specific Comments (for general comments, see p. 7)

1. This algorithm shows the CBMC growth/regrowth of a linear chain molecule (“alkane”) of ℓ united atoms.
2. Function **enerex** calculates the external energy of an atom at the given position, and function **select** selects one of the trial positions with probability $p(i) = w(i)/\sum_j w(j)$ (Algorithm 37).
3. function **next_ci**, with $i = 2, 3$ or n , adds the next atom to the chain as prescribed by the bonded interactions (Algorithms 25, 26, and 27 are, respectively, examples for a dimer (united-atom “ethane”), a trimer (“propane”), and a longer chain with bending and torsion potentials (“higher alkanes”).

Algorithm 25 (Add a single, randomly oriented bond)

```

function next_c2(xn,i)      generate a randomly oriented trial bond
                            starting from previous position xn(i-1)
     $\ell$  = bondl             generate bond length
    b = ranor               generate a randomly oriented unit vector
    xt(i)=xn(i-1)+ $\ell$ *b
end function

```

Specific Comments (for general comments, see p. 7)

1. The function **ranor** generates a randomly oriented unit vector (Algorithm 38), and the function **bondl** (Algorithm 39) generates a bond length prescribed by the bonded interactions.

dently, which can be an advantage in cases where the corresponding potentials are sharply peaked.

The acceptance-rejection technique is illustrated in Algorithms 24–27 for different n -alkanes. The methodology for dealing with branched hydrocarbons can be found in SI section L.8.1. For all-atom or explicit-hydrogen models of hydrocarbons, a different strategy is needed for which we refer the reader to the relevant literature [514,515].

Algorithm 26 (Trial conformation for a trimer)

<pre>function next_c3(xn,i) $\ell = \text{bondl}$ if i == 2 then xt = next_c2(xn,i) else if i == 3 then b= bonda(xn,i) xt=xn(2)+l*b endif end function</pre>	generate a trial position for i-th atom $xn(i-1)$ is position of the (i-1)th atom generate bond length second atom in chain use Algorithm 25 third atom generate orientation of the new position with desired bond angle
--	---

Specific Comments (for general comments, see p. 7)

1. A simple example of a trimer with a bond-angle potential is a united-atom model for propane.
2. The function **ranor** generates a random vector on a unit sphere (Algorithm 38). The function **bondl** (Algorithm 39) generates the bond length prescribed by the bonded interactions (for the second atom, only bond stretching). The function **bonda** generates a vector on a unit sphere with bond angle prescribed by the bond-bending potential (Algorithm 40).

Algorithm 27 (Generating conformation of chain with torsion potential)

<pre>function next_cn(xn,i) $\ell = \text{bondl}$ if i == 2 then xt = next_c2(xn,i) else if i == 3 then xt = next_c3(xn,i) else if i ≥ 4 then b = tors_bonda(xn,i) xt=xn(i-1)+l*b endif end function</pre>	generate a trial position for i-th atom position of atom (i-1) is known generate bond length second atom use Algorithm 25 third atom use Algorithm 26 fourth and higher atoms generate vector with prescribed bond and torsional angles
--	--

Specific Comments (for general comments, see p. 7)

1. The function **tors_bonda** (Algorithm 41) generates bond bending and a torsional angle prescribed by the corresponding potentials.

Example 21 (Generation of trial configurations of ideal chains). In section 12.2.3, we emphasized the importance of efficiently generating trial segments for molecules with strong intramolecular interactions. In this Example, we quantify this. We consider the following bead-spring model of a polymer. The nonbonded interactions are described with a Lennard-Jones potential and the bonded interactions with a harmonic spring:

$$u^{\text{vib}}(l) = \begin{cases} 0.5k_{\text{vib}}(l - 1)^2 & 0.5 \leq l \leq 1.5 \\ \infty & \text{otherwise} \end{cases},$$

where l is the bond length, the equilibrium bond length has been set to 1, and $k_{\text{vib}} = 400$. The bonded interaction is only the bond stretching. The external (nonbonded) interactions are the Lennard-Jones interactions. We consider the following two schemes of generating a set of trial positions:

1. Generate a random orientation with bond length uniformly distributed in the spherical shell between limits chosen such that they bracket all acceptable bond lengths. For instance, we could consider limits that correspond to a 50% stretching or compression of the bond. In that case, the probability of generating bond length l is given by

$$p_1(l) = \begin{cases} \propto C dl \propto l^2 dl & 0.5 \leq l \leq 1.5 \\ 0 & \text{otherwise} \end{cases}.$$

2. Generate a random orientation and the bond length prescribed by the bond-stretching potential (as described in Algorithm 25). The probability of generating bond length l with this scheme is

$$p_2(l) = \begin{cases} \propto C \exp[-\beta u^{\text{vib}}(l)] dl = C \exp[-\beta u^{\text{vib}}(l)] l^2 dl & 0.5 \leq l \leq 1.5 \\ 0 & \text{otherwise} \end{cases}.$$

Let us consider a case in which the system consists of ideal chains. Ideal chains are defined (see section 12.2.3) as chains having only *bonded* interactions.

Suppose we use method 1 to generate the set of k trial orientations with bond lengths l_1, \dots, l_k , then the Rosenbluth factor for atom i is given by

$$w_i(n) = \sum_{j=1}^k \exp[-\beta u^{\text{vib}}(l_j)].$$

The Rosenbluth factor of the entire chain is

$$W(n) = \prod_{i=1}^{\ell} w_i(n).$$

For the old conformation a similar procedure is used to calculate its Rosenbluth factor:

$$W(o) = \prod_{i=1}^{\ell} w_i(o).$$

In absence of external interactions the Rosenbluth factor of the first atom is defined to be $w_1 = k$.

In the second scheme, we generate the set of k trial orientations with a bond length distribution $p_2(l)$. If we use this scheme, we have to consider only the external interaction. Since, for an ideal chain, the external interactions are by definition 0, the Rosenbluth factor for each atom is given by

$$w_i^{\text{ext}}(n) = \sum_{j=1}^k \exp[-\beta u^{\text{ext}}(l_j)] = k,$$

and similarly, for the old conformation

$$w_l^{\text{ext}}(o) = k.$$

Hence, the Rosenbluth weight is the same for the new and the old conformations:

$$W^{\text{ext}}(n) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(n) = k^{\ell}$$

and

$$W^{\text{ext}}(o) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(o) = k^{\ell}.$$

The acceptance rule for the first scheme is

$$\text{acc}(o \rightarrow n) = \min[1, W(n)/W(o)]$$

and for the second scheme is

$$\text{acc}(o \rightarrow n) = \min[1, W^{\text{ext}}(n)/W^{\text{ext}}(o)] = 1.$$

Inspection of these acceptance rules shows that, in the second scheme, *all* configurations generated are accepted, whereas in the first scheme this probability depends on the bond-stretching energy and therefore will be less than 1. Hence, it is clearly useful to employ the second scheme.

To show that the results of schemes 1 and 2 are indeed equivalent, we compare the distribution of the bond length of the chain and the distribution of the radius of gyration in Fig. 12.6. The figure shows that the results for the two methods are indeed indistinguishable. The efficiency of the two methods, however, is very different. In Table 12.1, the difference in acceptance probability is given for some values of the bond-stretching force constant and various chain lengths. The table shows that if we use method 1 and generate a

uniformly distributed bond length, we need to use at least 10 trial orientations to have a reasonable acceptance for chains longer than 20 monomers. Note that the corresponding table for the second method has a 100% acceptance for all values of k independent of the chain length.

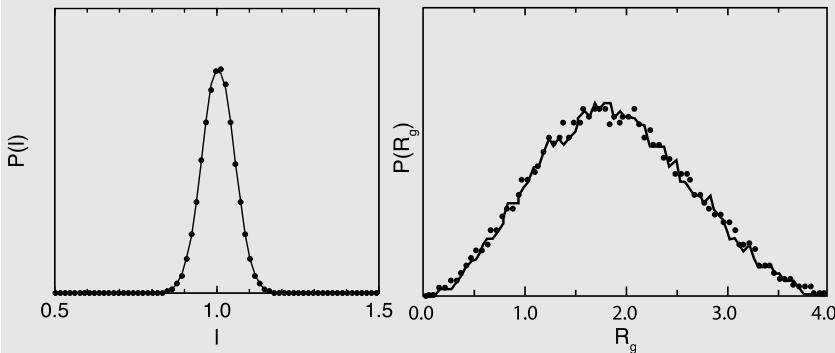


FIGURE 12.6 Comparison of methods 1 and 2 for the distribution of bond lengths l (left) and the distribution of the radius of gyration R_g (right). The solid lines represent the results for method 1, the dots for method 2 ($\ell = 5$ and $k = 5$).

TABLE 12.1 Probability of acceptance (%) for ideal chains using uniformly distributed bond lengths (method 1), where ℓ is the chain length, and k is the number of trial orientations. The value for the spring constant is $k_{\text{vib}} = 400$ (see [440]). For method 2, the acceptance would have been 100% for all values of k and ℓ .

k	$\ell = 5$	$\ell = 10$	$\ell = 20$	$\ell = 40$	$\ell = 80$	$\ell = 160$
1	0.6	$\ll 0.01$				
5	50	50	10	$\ll 0.01$	$\ll 0.01$	$\ll 0.01$
10	64	58	53	42	$\ll 0.01$	$\ll 0.01$
20	72	66	60	56	44	$\ll 0.01$
40	80	72	67	62	57	40
80	83	78	72	68	62	60

Most of the simulations, however, do not involve ideal chains but chains with external interactions. For chains with external interactions, the first method performs even worse. First of all, we generate the chains the same way as in the case of the ideal chains. The bonded interactions are the same and we need to generate at least the same number of trial directions to get a reasonable acceptance. In addition, if there are external interactions, we have to calculate the nonbonded interactions for *all* of these trial positions. The calculation of the nonbonded interactions takes most of the CPU time; yet, in the first method, most of the trial orientations are doomed to be rejected solely on the basis of the bonded energy. These two reasons make the second scheme much more attractive than the first.

The Fortran code to generate this Example can be found in the online-SI, Case Study 19.

12.4 Fixed endpoints

A drawback of the conventional configurational-bias Monte Carlo scheme is that it regrows a chain molecule, either partly or completely, starting from one of the endpoints. For dense systems, where only relatively short segments of the molecule can be regrown successfully, the configurational-bias Monte Carlo scheme reduces to the reptation scheme. This implies that the equilibration of the middle segments of a chain proceeds very slowly—for heteropolymers, where we must use wormhole rather than reptation moves [509], the situation is even worse. The same restriction applies to chain molecules that have either end rigidly anchored to a surface. Finally, conventional configurational-bias Monte Carlo cannot be applied at all to ring polymers.

In the present section, we discuss how the configurational-bias Monte Carlo scheme can be extended to include the sampling of chain conformations with fixed endpoints. With such a scheme it is possible to relax the interior of a chain as efficiently as the endpoints. Ring polymers can be considered special examples of chain molecules with fixed endpoints. Another interesting example that can be treated in the same way is the sampling of path integrals [516], but this falls outside the scope of this book. In SI section L.8.2, we discuss some alternative Monte Carlo techniques, such as concerted rotations and end-bridging Monte Carlo, which have been developed by Theodorou and co-workers [517].

12.4.1 Lattice models

Let us first consider configurational-bias Monte Carlo between fixed endpoints for a chain molecule on a simple cubic lattice. If we remove n segments of the molecule between two fixed endpoints \mathbf{r}_1 and \mathbf{r}_2 , we cannot simply regrow the molecule by the normal Rosenbluth scheme, because this does not ensure that a trial conformation starting at \mathbf{r}_1 will end at \mathbf{r}_2 . Clearly, we must bias our regrowth scheme in such a way that the trial conformation is forced to terminate at \mathbf{r}_2 . To achieve this, we use the following scheme. Suppose that we start our regrowth at position \mathbf{r}_1 . On a three-dimensional lattice, this coordinate is represented by three integer coordinates $\{k_1, l_1, m_1\}$. The final position is denoted by $\{k_2, l_2, m_2\}$. The total number of ideal (i.e., nonself-avoiding) random walks of length n between \mathbf{r}_1 and \mathbf{r}_2 is denoted by $\Omega(\mathbf{r}_1, \mathbf{r}_2; n)$. We can always compute the number of ideal random walks between fixed endpoints analytically as it is simply a finite sum of multinomial coefficients [518,519]. Let us next consider the growth of one segment, starting at \mathbf{r}_1 . In the original configurational-bias Monte Carlo scheme, we would consider all k possible trial directions. And we

would select one of these directions, say direction j , with a probability

$$P(j) = \frac{\exp[-\beta u^{\text{ext}}(j)]}{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(j')]} ,$$

where $u^{\text{ext}}(j)$ denotes the potential energy of trial segment j due to all other particles already in the system. In the present case, we use a different weight factor to select the trial segment, namely,

$$P(j) = \frac{\exp[-\beta u^{\text{ext}}(j)]\Omega(\mathbf{r}_1 + \Delta\mathbf{r}(j), \mathbf{r}_2; n - 1)}{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(j')]\Omega(\mathbf{r}_1 + \Delta\mathbf{r}(j'), \mathbf{r}_2; n - 1)}. \quad (12.4.1)$$

In other words, the probability of selecting a given trial direction is proportional to the number of ideal random walks of length $n - 1$ that start at the position of the trial segment and terminate at \mathbf{r}_2 . In this way, we *guarantee* that we generate only conformations that start at \mathbf{r}_1 and terminate at \mathbf{r}_2 . However, as before, we must correct for the bias that we have introduced. We do this by constructing a modified Rosenbluth weight W : $W = \prod_{i=1}^n w_i$ with

$$\begin{aligned} w_i &\equiv \frac{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(j')]\Omega[\mathbf{r}_i + \Delta\mathbf{r}(j'), \mathbf{r}_2; n - i]}{\sum_{j'=1}^k \Omega[\mathbf{r}_i + \Delta\mathbf{r}(j'), \mathbf{r}_2; n - i]} \\ &= \frac{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(j')]\Omega[\mathbf{r}_i + \Delta\mathbf{r}(j'), \mathbf{r}_2; n - i]}{\Omega[\mathbf{r}_i, \mathbf{r}_2; n - i + 1]}. \end{aligned} \quad (12.4.2)$$

If we now multiply the probability of generating a given trial conformation Γ with the Rosenbluth weight of that conformation, we find that

$$\begin{aligned} P_{\text{gen}}(\Gamma) \times W(\Gamma) &= \prod_{i=1}^n \left\{ \frac{\exp[-\beta u^{\text{ext}}(j)]\Omega[\mathbf{r}_i + \Delta\mathbf{r}(j), \mathbf{r}_2; n - i]}{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(j')]\Omega[\mathbf{r}_i + \Delta\mathbf{r}(j'), \mathbf{r}_2; n - i]} \right. \\ &\quad \times \left. \frac{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(j')]\Omega[\mathbf{r}_i + \Delta\mathbf{r}(j'), \mathbf{r}_2; n - i]}{\Omega[\mathbf{r}_i, \mathbf{r}_2; n - i + 1]} \right\} \\ &= \prod_{i=1}^n \left\{ \frac{\exp[-\beta u^{\text{ext}}(j)]\Omega[\mathbf{r}_i + \Delta\mathbf{r}(j), \mathbf{r}_2; n - i]}{\Omega[\mathbf{r}_i, \mathbf{r}_2; n - i + 1]} \right\}. \end{aligned} \quad (12.4.3)$$

The modified Rosenbluth weight has been chosen such that all but one of the factors involving the number of ideal conformations cancel each other:

$$\begin{aligned} P_{\text{gen}}(\Gamma) \times W(\Gamma) &= \prod_{i=1}^n \frac{\exp[-\beta u^{\text{ext}}(i)]}{\Omega(\mathbf{r}_1, \mathbf{r}_2; n)} \\ &= \frac{\exp[-\beta U^{\text{ext}}(\Gamma)]}{\Omega(\mathbf{r}_1, \mathbf{r}_2; n)}. \end{aligned} \quad (12.4.4)$$

The remaining factor Ω is the same for all conformations of length n that start at \mathbf{r}_1 and terminate at \mathbf{r}_2 ; hence, it drops out when we compute the relative probabilities of the old and new conformations. As before, the actual Monte Carlo scheme involves generating the trial conformation using the scheme indicated in Eq. (12.4.1) and accepting the new conformation with a probability given by

$$\text{acc}(o \rightarrow n) = \min [1, W(n)/W(o)]. \quad (12.4.5)$$

A total regrowth of a ring polymer of length ℓ can be accomplished by choosing $\mathbf{r}_1 = \mathbf{r}_2$ and $n = \ell$.

12.4.2 Fully flexible chain

Again, it is possible to extend configurational-bias Monte Carlo to sample chain conformations between fixed endpoints, using our knowledge of the exact expression for the number (or, more precisely, the probability density) of ideal (nonself-avoiding) conformations of n segments between fixed endpoints \mathbf{r}_1 and \mathbf{r}_2 . If we denote the probability density to find segment $i + 1$ at a distance r from segment i by $p_1(r)$, then we have the following recursion relation between the probability density of the end-to-end separation of chains of length n and $n + 1$:

$$P(\mathbf{r}_{12}; n + 1) = \int d\Delta P(\mathbf{r}_{12} - \Delta; n) p_1(\Delta). \quad (12.4.6)$$

From Eq. (12.4.6) and the fact that $p_1(r)$ is normalized, we immediately deduce the inverse relation:

$$P(\mathbf{r}_{12}; n) = \int d\Delta P(\mathbf{r}_{12} + \Delta; n + 1). \quad (12.4.7)$$

In the special case that all segments are of fixed length a , the expression for this probability density is [520]

$$P(\mathbf{r}_{12}; n) = \frac{\sum_{k=0}^{k \leq (n-r_{12}/a)/2} (-1)^k \binom{n}{k} (n - 2k - r_{12}/a)^{n-2}}{2^{n+1} (n-2)! \pi a^2 r_{12}}, \quad (12.4.8)$$

where $r_{12} \equiv |\mathbf{r}_1 - \mathbf{r}_2|$. This expression is valid for all $n > 1$. As before, we wish to modify the configurational-bias Monte Carlo sampling of conformations of a fully flexible chain in such a way that the chain is forced to terminate at \mathbf{r}_2 . There are two ways to do this. In one approach, we include the bias in the probability with which we generate trial directions; in the second, the bias is in the acceptance probability. In either case, our approach does not depend on the specific form of $p_1(r)$, but only on the existence of the recurrence relation (12.4.7).

In the first approach, we use the following scheme of generating the i th segment out of ℓ segments to be regrown. We generate k trial segments, all starting at the current trial position \mathbf{r} , such that the *a priori* probability of generating a

given trial direction (say, Γ_j) is proportional to the probability of having an ideal chain conformation of length $\ell - i$ between this trial segment and the final position \mathbf{r}_2 . Let us denote this *a priori* probability by $p_{\text{bond}}(\Gamma_j)$. By construction, $p_{\text{bond}}(\Gamma_j)$ is normalized. Using Eq. (12.4.7) we can easily derive an explicit expression for p_{bond} :

$$\begin{aligned} p_{\text{bond}}(\Gamma) &= \frac{p_1(\Gamma)P(\mathbf{r} + \Gamma - \mathbf{r}_2; \ell - i)}{\int d\Gamma' p_1(\Gamma')P(\mathbf{r} + \Gamma' - \mathbf{r}_2; \ell - i)} \\ &= \frac{p_1(\Gamma)P(\mathbf{r} + \Gamma - \mathbf{r}_2; \ell - i)}{P(\mathbf{r} - \mathbf{r}_2; \ell - i + 1)}. \end{aligned} \quad (12.4.9)$$

From here on, we treat the problem just like the sampling of a continuously deformable chain, described in section 12.2.3. That is, we select one of the k trial directions with a probability

$$P_{\text{sel}}(j) = \frac{\exp[-\beta u^{\text{ext}}(\Gamma_j)]}{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(\Gamma_{j'})]}.$$

The contribution to the total Rosenbluth weight of the set of k trial directions generated in step i is

$$w_i \equiv \frac{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}(\Gamma_{j'})]}{k}.$$

The overall probability of moving from the old conformation Γ_{old} to a new conformation Γ_{new} is proportional to the product of the probability of generating the new conformation and the ratio of the new to the old Rosenbluth weights. The condition of (super-)detailed balance requires that the product of the probability of generating the new conformation times the Rosenbluth weight of that conformation is (but for a factor that is the same for the old and new conformations) equal to the product of the Boltzmann weight of that conformation and the properly normalized probability of generating the corresponding ideal (i.e., noninteracting) conformation. If we write the expression for this product, we find that

$$\begin{aligned} &\prod_{i=1}^{\ell} P_{\text{gen}}[\Gamma_j(i)]w_i \\ &= \prod_{i=1}^{\ell} \left(\frac{p_1(\mathbf{r}_i - \mathbf{r}_{i-1})P(\mathbf{r}_i - \mathbf{r}_2; \ell - i)}{P(\mathbf{r}_{i-1} - \mathbf{r}_2; \ell - i + 1)} \right) \left(\frac{\exp[-\beta u^{\text{ext}}[\Gamma_j(i)]]}{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}[\Gamma_{j'}(i)]]} \right) \\ &\quad \times \left(\frac{\sum_{j'=1}^k \exp[-\beta u^{\text{ext}}[\Gamma_{j'}(i)]]}{k} \right) \\ &= \frac{\exp[-\beta U^{\text{ext}}(\Gamma_{\text{total}})] \prod_{i=1}^{\ell} p_1(\mathbf{r}_i - \mathbf{r}_{i-1})}{k^{\ell} P(\mathbf{r}_{12}; \ell)}. \end{aligned} \quad (12.4.10)$$

As the last line of this equation shows, the conformations are indeed generated with the correct statistical weight. In ref. [521] this scheme has been applied to simulate model homopolymers, random heteropolymers, and random copolymers consisting of up to 1000 Lennard-Jones beads. For molecules with strong intramolecular interactions, the present scheme will not work and other approaches are needed.

12.4.3 Strong intramolecular interactions

In the previous section, we have shown that we can use the configurational-bias Monte Carlo scheme to grow a chain of length n between two fixed endpoints \mathbf{r}_1 and \mathbf{r}_2 if we know the probability density of conformations of length n between these points. For the special case of a fully flexible chain, this probability distribution is known analytically. For chains with strong intramolecular interactions, such an analytical distribution is not known. Wick and Siepmann [522] and Chen and Escobedo [523] have shown that one can use an approximated distribution. Chen and Escobedo [523] estimate this distribution using a simulation of an isolated chain with bonded interactions only. Wick and Siepmann [522] proposed a scheme in which this estimated probability distribution is further refined during the simulation.

12.5 Beyond polymers

Thus far, the CBMC scheme has been presented exclusively as a method of generating polymer conformations. The method is more general than that. It can be used as a scheme to perform collective rearrangements of any set of labeled coordinates. In fact, the scheme can be used to carry out Monte Carlo moves to swap n small particles within a volume ΔV with one large particle that occupies the same (excluded) volume. This application of the CBMC scheme has been exploited by Biben et al. [524,525] to study mixtures of large and small hard spheres. Gibbs ensemble simulations of mixtures of spherical colloids and rod-like polymers were performed by Bolhuis and Frenkel [526] (see Example 17), using CBMC-style particle swaps and a closely related approach was employed by Dijkstra and co-workers to study phase separation [518,519] in mixtures of large and small hard-core particles on a lattice. An application of CBMC for improving the sampling of ionic solutions has been proposed by Shelley and Patey [527].

A different application of the CBMC ideas is used by Esselink et al. [528] to develop an algorithm to perform Monte Carlo moves in parallel. Parallel Monte Carlo appears to be a contradiction in terms, since the Monte Carlo procedure is an intrinsically sequential process. One has to know whether the current move is accepted or rejected before one can continue with the next move. The conventional way of introducing parallelism is to distribute the energy calculation over various processors or to farm out the calculation by performing separate

simulations over various processors. Although the last algorithm is extremely efficient and requires minimum skills to use a parallel computer, it is not a truly parallel algorithm. For example, farming out a calculation is not very efficient if the equilibration of the system takes a significant amount of CPU time. In the algorithm of Esselink et al. several trial positions are generated in parallel, and out of these trial positions, the one with the highest probability of being accepted is selected. This selection step introduces a bias that is removed by adjusting the acceptance rules. The generation of each trial move, which includes the calculation of the energy (or Rosenbluth factor in the case of chain molecules), is distributed over the various processors. Loyens et al. have used this approach to perform phase equilibrium calculations in parallel using the Gibbs ensemble technique [529].

An interesting application of this parallel scheme is the multiple-first-bead algorithm. In a conventional CBMC simulation one would have to grow an entire chain before one can reject a configuration that is “doomed” from the start because the very first bead has an unfavorable energy. If the chains are long growing them to the end before deciding on acceptance can be inefficient and it becomes advantageous to use a multiple-first-bead scheme [528].⁷ Instead of generating a single trial position for the first bead, k trial positions are generated. The energy of these beads, $u_1(j)$ with $j = 1, \dots, k$, is calculated, and one of these beads, say j , is selected using the Rosenbluth criterion:

$$P_{1st}(j) = \frac{\exp[-\beta u_1(j)]}{w_1}$$

where

$$w_1(n) = \sum_{i=1}^k \exp[-\beta u_1(i)].$$

Also for the old configuration, one should use a similar scheme to compute $w_1(o)$. For some moves, the same set of first beads used for the new configuration can be used to compute the Rosenbluth factor for the old configuration [530]. To ensure detailed balance the Rosenbluth factors associated with the multiple-first beads should be taken into account in the acceptance rules:

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{w_1(n)W(n)}{w_1(o)W(o)} \right),$$

where $W(n)$ and $W(o)$ are the (conventional) Rosenbluth factors of the new and the old configuration of the chain, respectively, excluding the contribution of the first segment. Vlugt et al. [531] have shown that a multiple-first-bead move can increase the efficiency of simulations of n-alkanes up to a factor of 3.

⁷ Note that the same problem is also addressed by the early-rejection method discussed in section 13.4.3

Another extension of the CBMC approach is the use of a dual-cutoff radius [531]. The idea is that usually a particular trial conformation is accepted not because it is energetically very favorable, but because its competitors are so unfavorable. This suggests that one can use a much cheaper potential to perform a prescreening of acceptable trial configurations in a CBMC move. Let us split the potential into a contribution that is cheap to compute and the expensive remainder:

$$U(r) = U^{\text{cheap}}(r) + \Delta U(r).$$

This can be done, for example, by splitting the potential into a long-range and short-range part. We can now use the cheap part in our CBMC scheme to generate trial configurations. The probability of generating a given configuration is then

$$P^{\text{cheap}}(n) = \frac{\exp[-\beta U^{\text{cheap}}(n)]}{W^{\text{cheap}}(n)}$$

and the move is accepted using

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{W^{\text{cheap}}(n)}{W^{\text{cheap}}(o)} \exp\{-\beta[\Delta U(n) - \Delta U(o)]\} \right).$$

In ref. [531] it is shown that this scheme obeys detailed balance. The advantage of this algorithm is that the expensive part of the energy calculation has to be performed only once and not for every trial segment. A typical application would be to include the Fourier part of an Ewald summation in ΔU . Many variations on this theme exist: one example is Hybrid MC (see section 13.3.1).

Illustration 17 (Mixtures of colloids and polymers). We have presented CBMC as a scheme for sampling conformations of chain molecules. However, the method is more general than that. It can be used to perform collective rearrangements of any set of labeled coordinates. For instance, the scheme can be used to carry out Monte Carlo moves to swap n small particles within a volume ΔV with one large particle that occupies the same (excluded) volume. This application of the CBMC scheme has been exploited by Biben [524] to study mixtures of large and small hard spheres. Gibbs ensemble simulations of mixtures of spherical colloids and rodlike polymers were performed in ref. [526] using CBMC-style particle swaps, and a closely related approach was employed by Dijkstra et al. [518,519] to study phase separation of mixtures of large and small hard-core particles on a lattice.

Below, we briefly discuss an example of such a CBMC scheme, related to the phase behavior of colloidal suspensions [526]. Examples of colloidal solutions are milk, paint, and mayonnaise. Since a single colloidal particle may contain more than 10^9 atoms, it is not practical to model such a particle as a collection of atoms. It is better to describe colloidal solutions using

coarse-grained models. For example, a suspension of sterically stabilized silica spheres in a nonpolar solvent can be described surprisingly accurately with a hard-sphere potential. Similar to the hard-sphere fluid, such a colloidal suspension has a “fluid-solid” transition but not a “liquid-gas” transition. To be more precise, the colloidal particles undergo a transition from a liquid-like arrangement to a crystalline structure. But in either case, the solvent remains liquid. In what follows, the terms “crystal,” “liquid,” and “gas” refer to the state of the colloidal particles in suspension. Experimentally, it is observed that a liquid-gas transition can be induced in a suspension of hard-sphere colloids by adding nonadsorbing polymers.

The addition of polymers induces an effective attraction between the colloidal particles. This attraction is not related to any change in the internal energy of the system but to an increase in entropy. It is not difficult to understand the origin of such entropic attractions. Let us assume that the polymers in solution do not interact with each other. This is never rigorously true, but for dilute solutions of long, thin molecules, it is a good first approximation. The translational entropy of N polymers in a volume V is then equal to that of N ideal-gas molecules occupying the same volume: $S_{\text{trans}}^{(0)} = \text{constant} + Nk_B \ln V$, where the constant accounts for all those contributions that do not depend on the volume V . In the absence of colloids, the volume accessible to the polymers is equal to V_0 , the volume of the container. Now suppose that we add one hard colloidal particle with radius R_c . As the polymers cannot penetrate the colloidal particle, such a colloid excludes the polymers from a spherical volume with radius $R_{\text{excl}} \equiv R_c + R_p$, where R_p is the effective radius of the polymer (for flexible polymers, R_p is on the order of the radius of gyration, and for rigid polymers, R_p is of order $\mathcal{O}(L)$, where L is the length of the polymer). Let us denote the volume excluded by one colloid by v_{excl}^c . Clearly, the entropy of N polymers in the system that contains one colloid is $S_{\text{trans}}^{(1)} = \text{constant} + Nk_B \ln(V_0 - v_{\text{excl}}^c)$. Now consider what happens if we have two colloidal spheres in the solution. Naively, one might think that the entropy of the polymer solution is now equal to $S_{\text{trans}}^{(2)} = \text{constant} + Nk_B \ln(V_0 - 2v_{\text{excl}}^c)$. However, this is only true if the two colloids are far apart. If they are touching, their exclusion zones overlap, and the total excluded volume $v_{\text{excl}}^{\text{pair}}$ is less than $2v_{\text{excl}}^c$. This implies that the entropy of the polymers is larger when the colloids are touching than when they are far apart. Therefore, we can lower the free energy of the polymer solution by bringing the colloids close together. And this is the origin of entropic attraction. The strength of the attraction can be tuned by changing the polymer concentration, and for sufficiently high polymer concentrations, the colloidal suspensions may undergo a “liquid-vapor” phase separation.

In the present example, we consider the phase behavior of a mixture of colloidal hard spheres and thin hard rods [526]. In principle, we can use Gibbs ensemble simulations to study the “vapor-liquid” coexistence in this mixture. However, a conventional Gibbs ensemble simulation is likely to fail as the transfer of a colloidal sphere from one simulation box to the other will, almost certainly, result in an overlap of the sphere with some of the rodlike

polymers. We can now use the CBMC approach to perform such a trial move with a higher chance of success. In this scheme, we perform the following steps:

1. Randomly select a sphere in one of the boxes and insert this sphere at a random position in the other box.
2. Remove all the rods that overlap with this sphere. These rods are inserted in the other box. The positions and orientations of the rods are chosen such that they intersect with the volume vacated by the colloid—but apart from that, they are random. Even though we have thus ensured that the rods are in, or near, the “cavity” left by the colloidal sphere, they are very likely to overlap with one or more of the remaining spheres. However, if one tries several orientations and positions of the rods and selects an acceptable configuration using the configurational-bias Monte Carlo scheme, one can strongly enhance the acceptance probability of such particle swaps.

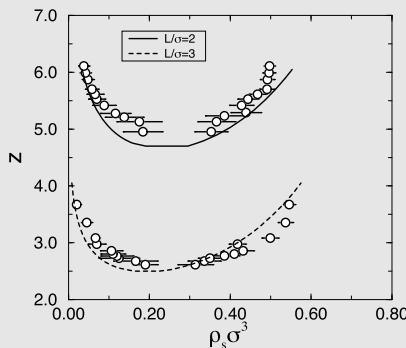


FIGURE 12.7 Coexistence curves for a mixture of hard spheres and thin rods [526]. The horizontal axis measures the density, and the vertical axis the fugacity ($= \exp(\beta\mu)$). L/σ is the ratio of the length of the rods to the diameter of the hard spheres.

The results of these Gibbs ensemble simulations are presented in Fig. 12.7. This figure shows that if one increases the fugacity (and thereby the concentration) of the rods, a demixing into a phase with a low density of spheres and a phase with a high density of spheres occurs. The longer the rods, the lower the concentration at which this demixing occurs. We stress once again that, in this system, only hard-core interactions between the particles exist. Therefore this demixing is driven by entropy alone.

12.6 Other ensembles

12.6.1 Grand-canonical ensemble

In Chapter 6, we introduced the grand-canonical ensemble in the context of simulations of systems in open contact with a reservoir. An essential ingredient of Monte Carlo simulations in this ensemble is the random insertion or removal of particles. Clearly, such simulations will be efficient only if there is

a reasonable acceptance probability of particle-insertion moves. In particular, for polyatomic molecules, this is usually a problem. Let us consider the system mentioned in Example 4, a grand-canonical ensemble simulation of the adsorption of molecules in the pores of a microporous material such as a zeolite. For single atoms, the probability that we find an arbitrary position that does not overlap with one of the atoms in the zeolite lattice is on the order 1 in 10^3 . For dimers, we have to find two positions that do not overlap, and if we assume that these positions are independent, the probability of success will be 1 in 10^6 . Clearly, for the long-chain molecules, the probability of a successful insertion is so low that to obtain a reasonable number of accepted insertions, the number of attempts needs to be prohibitively large. In the present section, we demonstrate how configurational-bias Monte Carlo technique can be used in the grand-canonical ensemble to make the exchange step of chain molecules more probable.

Algorithm

As in the general scheme of the configurational-bias Monte Carlo technique for off-lattice systems, we divide the potential energy of a given conformation into a bonded potential energy ($\mathcal{U}^{\text{bond}}$), which includes the local intramolecular interactions, and an *external* potential energy (\mathcal{U}^{ext}), which includes the intermolecular interactions and the nonbonded intramolecular interactions (see section 12.2.3). A chain that has only bonded interactions is defined as an ideal chain. Let us now consider the Monte Carlo trial moves for the insertion and removal of particles.

Particle insertion

To insert a particle into the system, we use the following steps:

1. For the first monomer, a random position is selected, and the energy of this monomer is calculated. This energy is denoted by $u_1^{\text{ext}}(n)$ and we define $w_1^{\text{ext}}(n) = k \exp[-\beta u_1^{\text{ext}}(n)]$ (as before, the factor k is introduced only to simplify the subsequent notation).
2. For the following monomers, a set of k trial positions is generated. We denote these positions by $\{\mathbf{b}\}_k = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k)$. This set of trial orientations is generated using the bonded part of the potential, which results in the following distribution for the i th monomer:

$$p_i^{\text{bond}}(\mathbf{b})d\mathbf{b} = C \exp[-\beta u_i^{\text{bond}}(\mathbf{b})]d\mathbf{b} \quad (12.6.1)$$

with

$$C^{-1} \equiv \int d\mathbf{b} \exp[-\beta u_i^{\text{bond}}(\mathbf{b})]. \quad (12.6.2)$$

Note that the way the trial orientations are generated depends on the type of monomer being added (see section 12.3). For each of these trial positions the

external energy, $u_i^{\text{ext}}(\mathbf{b}_j)$, is calculated, and one of these positions is selected with a probability

$$p_i^{\text{ext}}(\mathbf{b}_n) = \frac{\exp[-\beta u_i^{\text{ext}}(\mathbf{b}_n)]}{w_i^{\text{ext}}(n)}, \quad (12.6.3)$$

in which

$$w_i^{\text{ext}}(n) = \sum_{j=1}^k \exp[-\beta u_i^{\text{ext}}(\mathbf{b}_j)].$$

3. Step 2 is repeated until the entire alkane of length ℓ has been grown, and the normalized Rosenbluth factor can be calculated:

$$\mathcal{W}^{\text{ext}}(n) \equiv \frac{W^{\text{ext}}(n)}{k^\ell} = \prod_{i=1}^{\ell} \frac{w_i^{\text{ext}}(n)}{k}. \quad (12.6.4)$$

4. The new molecule is accepted with a probability

$$\text{acc}(N \rightarrow N + 1) = \min \left(1, \frac{q(T) \exp(\beta \mu^B) V}{(N + 1)} \mathcal{W}^{\text{ext}}(n) \right), \quad (12.6.5)$$

where μ^B is the chemical potential of a reservoir consisting of *ideal chain* molecules and $q(T)$ is the kinetic contribution to the molecular partition function (for atoms, $q(T) = 1/\Lambda^3$).

Particle removal

To remove a particle from the system, we use the following algorithm:

1. A particle, say, o , is selected at random, the energy of the first monomer is calculated and is denoted by $u_1^{\text{ext}}(o)$, and we determine $w_1^{\text{ext}}(o) = k \exp[-\beta u_1^{\text{ext}}(o)]$.
2. For the following segments of the chain, the external energy $u_i^{\text{ext}}(o)$ is calculated and a set of $k - 1$ trial orientations is generated with a probability given by Eq. (12.6.1). Using this set of orientations and the actual position, we calculate for monomer i :

$$w_i^{\text{ext}}(o) = \exp[-\beta u_i^{\text{ext}}(o)] + \sum_{j=2}^k \exp[-\beta u_i^{\text{ext}}(\mathbf{b}_j)].$$

3. After step 2 is repeated for all ℓ monomers and we compute for the entire molecule:

$$\mathcal{W}^{\text{ext}}(o) \equiv \frac{W^{\text{ext}}(o)}{k^\ell} = \prod_{i=1}^M \frac{w_i^{\text{ext}}(o)}{k}. \quad (12.6.6)$$

4. The selected molecule is removed with a probability

$$\text{acc}(N \rightarrow N - 1) = \min \left(1, \frac{N}{q(T)V \exp(\beta\mu^B)} \frac{1}{\mathcal{W}^{\text{ext}}(o)} \right). \quad (12.6.7)$$

We have defined μ^B as the chemical potential of a reservoir consisting of ideal chains. It is often convenient to use as a reference state the ideal gas of nonideal chains (i.e., chains that have both bonded and nonbonded intramolecular interactions). This results in a simple, temperature-dependent shift of the chemical potential:

$$\beta\mu^B \equiv \beta\mu_{\text{id.chain}} = \beta\mu_{\text{nonid.chain}} + \ln \langle \mathcal{W}^{\text{nonbonded}} \rangle, \quad (12.6.8)$$

where $\langle \mathcal{W}^{\text{nonbonded}} \rangle$ is the average Rosenbluth factor due to the nonbonded intramolecular interactions. This Rosenbluth factor has to be determined in a separate simulation of a single-chain molecule. For more details about reference states, see SI section L.14. In the same appendix, we also discuss the relation between the chemical potential and the imposed pressure (the latter quantity is needed when comparing with real experimental data). To show that the preceding algorithm does indeed yield the correct distribution, we have to demonstrate, as before, that detailed balance is satisfied. As the proof is very similar to those shown before, we will not reproduce it here. For more details, the reader is referred to [454].

Illustration 18 (Adsorption of alkanes in zeolites). In Illustration 4, grand-canonical simulations were used to determine the adsorption of methane in the zeolite silicalite. Using the scheme described in the present section, Smit and Maesen computed adsorption isotherms of the longer alkanes [532]. Adsorption isotherms are of interest since they may signal phase transitions, such as capillary condensation or wetting, of the fluid inside the pores [533]. Capillary condensation usually shows up as a step or rapid variation in the adsorption isotherm. It is often accompanied by hysteresis, but not always; for instance, experiments on flat substrates [534] found evidence for steps in the adsorption isotherm without noticeable hysteresis.

Since the pores of most zeolites are of molecular dimensions, adsorbed alkane molecules behave like a one-dimensional fluid. In a true one-dimensional system, phase transitions are not expected to occur. To the extent that zeolites behave as a one-dimensional medium, one, therefore, might expect that the adsorption isotherms of alkanes in zeolites exhibit no steps. If steps occur, they are usually attributed to capillary condensation in the exterior secondary pore system formed by the space between different crystals. For silicalite, adsorption isotherms have been determined for various *n*-alkanes, and, indeed, for the short-chain alkanes (methane–pentane), the isotherms ex-

hibit no steps. The same holds for decane. For hexane and heptane, however, steplike features are observed (for experimental details, see [532]).

In the simulations of Smit and Maesen [532], the alkane molecules are modeled with a united atom model; that is, CH_3 and CH_2 groups are considered as single interaction centers [535]. The zeolite is modeled as a rigid crystal and the zeolite-alkane interactions are assumed to be dominated by the interaction with the oxygen atoms and are described by a Lennard-Jones potential.

Fig. 12.8 compares the simulated adsorption isotherms of various alkanes in silicalite with experimental data. For butane, a smooth isotherm is observed, and the agreement between experiments and simulation is good. For hexane and heptane, the agreement is good at high pressures, but at low pressures, deviations indicate that the zeolite-alkane model may need to be refined. It is interesting to note that, for heptane, both the experiments and the simulations show a step at approximately half the loading. Since the simulations are performed on a perfect single crystal, this behavior must be due to a transition of the fluid inside the pores and cannot be attributed to the secondary pore system.

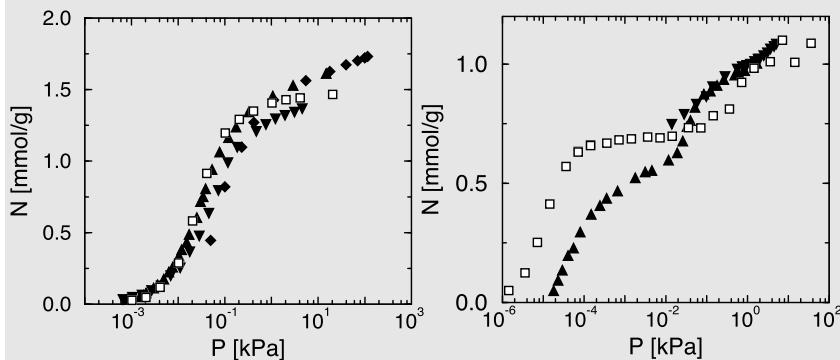


FIGURE 12.8 Adsorption isotherms of butane (left) and heptane (right); the closed symbols are experimental data, and the open symbols the results from simulations at $T = 298\text{ K}$.

Silicalite has two types of channels, straight and zigzag, which are connected via intersections. It so happens that the length of a hexane molecule is on the order of the length of the period of the zigzag channel. The simulations show that, at low chemical potential, the hexane molecules move freely in these channels, and the molecules will spend part of their time at the intersections. If a fraction of the intersections is occupied, other molecules cannot reside in the straight channels at the same time. At high pressures, almost all hexane molecules fit exactly into the zigzag channel. They no longer move freely and keep their noses and tails out of the intersection. In such a configuration, the entire straight channel can now be tightly packed with hexane molecules. This may explain the plateau in the adsorption isotherm; to fill the entire zeolite structure neatly, the hexane molecules located in zigzag

channels first have to be “frozen” in these channels. This “freezing” of the positions of the hexane molecules implies a loss of entropy and, therefore, will occur only if the pressure (or chemical potential) is sufficiently high to compensate for this loss. This also makes it clear why we do not observe a step for molecules shorter or longer than hexane or heptane. If the molecules are longer, they will always be partly in the intersection, and nothing can be gained by collective freezing in the zigzag channels. If the molecules are shorter than one period of the zigzag channel, a single molecule will not occupy an entire period, and a second molecule will enter, which results in a different type of packing. The interesting aspect is that after the simulations were published, this observation was confirmed by experiments [536].

Also, the adsorption behavior of mixtures of hydrocarbons has many surprising effects [537,538].

In SI section L.8.3 the combination of CBMC and the Gibbs ensemble is discussed.

12.7 Recoil growth

To find numerical schemes that are more efficient than CBMC, we should first understand why CBMC works better than a scheme that employs random trial moves. Suppose that we have a system with hard-core interactions and the probability of successfully inserting a monomer is a . If we assume that the insertion of an m -mer is equivalent to inserting m independent monomers, then the probability of a successful random insertion of an n -mer is

$$p_m^{\text{random}} \approx a^m.$$

For a dense system, $a \ll 1$, and therefore random insertion only works for very short chains. With the CBMC scheme we generate k trial orientations and our growing scheme fails if *all* of the k trial orientations result in an overlap. The probability that we grow a chain successfully is therefore

$$p_m^{\text{CBMC}} \approx a \left[1 - (1 - a)^k \right]^{m-1} = ab^{m-1}.$$

This crude estimate suggests that by increasing k , the number of trial orientations, we can make b arbitrarily close to 1 and hence obtain a reasonable insertion probability for any chain length and at any density. In practice, simply increasing k will not solve the problem. First of all, there is a practical limitation: increasing k increases the computational cost. More importantly, the assumption that the probability of a successful insertion of a monomer is equal and independent for each trial position is not correct. For instance, if we have grown into a “dead alley” where there is simply no space for an additional monomer (see Fig. 12.9), then no matter how often we try, the insertion will not be accepted. At high densities, such dead alleys are the main reason the CBMC method becomes

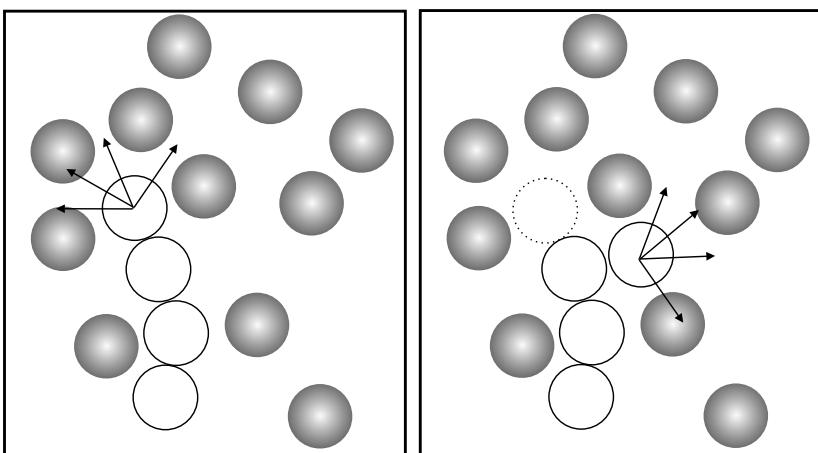


FIGURE 12.9 The configurational-bias Monte Carlo scheme fails if the molecule is trapped in a dead alley (left); irrespective of the number of trial orientations the CBMC scheme will never generate an acceptable conformation. In the recoil growth scheme (right) the algorithm “recoils” back to a previous monomer and attempts to regrow from there.

inefficient. This suggests that we need a computational scheme that allows us to escape from these dead alleys.

The **Recoil Growth (RG)** scheme is a dynamic Monte Carlo algorithm that was developed with the dead-alley problem in mind [539,540]. The algorithm is related to earlier static MC schemes due to Meirovitch [541] and Alexandrowicz and Wilding [542]. The basic strategy of the method is that it allows us to escape from a trap by “recoiling back” a few monomers and retrying the growth process using another trial orientation. In contrast, the CBMC scheme looks only one step ahead. Once a trial orientation has been selected, we cannot “deselect” it, even if it turns out to lead into a dead alley. The recoil growth scheme looks several monomers ahead to see whether traps are to be expected before a monomer is irrevocably added to the trial conformation (see Fig. 12.9). In this way we can alleviate (but not remove) the dead-alley problem. In principle, one could also do something similar with CBMC by adding a sequence of l monomers per step. However, as there are k possible directions for every monomer, this would involve computing k^l energies per group. Even though many of these trial monomers do not lead to acceptable conformations, we would still have to compute all interaction energies.

12.7.1 Algorithm

To explain the practical implementation of the RG algorithm, let us first consider a totally impractical, but conceptually simple scheme that will turn out to have the same net effect. Consider a chain of l monomers. We place the first monomer at a random position. Next, we generate k trial positions for the second

monomer. From each of these trial positions, we generate k trial positions for the third monomer. At this stage, we have generated k^2 “trimer” chains. We continue in the same manner until we have grown k^{l-1} chains of length l . Obviously, most of the conformations thus generated have a vanishing Boltzmann factor and are, therefore, irrelevant. However, some may have a reasonable Boltzmann weight and it is these conformations that we should like to find.

To simplify this search, we introduce a concept that plays an important role in the RG algorithm: we shall distinguish between trial directions that are “open” and those that are “closed.” To decide whether a given trial direction, say b , for monomer j is open, we compute its energy $u_j(b)$. The probability⁸ that trial position b is open is given by

$$p_j^{\text{open}}(b) = \min(1, \exp[-\beta u_j(b)]). \quad (12.7.1)$$

For hard-core interactions, the decision of whether a trial direction is open or closed is unambiguous, as $p_j^{\text{open}}(b)$ is either zero or one. For continuous interactions we compare $p_j^{\text{open}}(b)$ with a random number between 0 and 1. If the random number is less than $p_j^{\text{open}}(b)$, the direction is open; otherwise, it is closed. We now have a tree with k^{l-1} branches but many of these branches are “dead,” in the sense that they emerge from a “closed” monomer. Clearly, there is little point in exploring the remainder of a branch if it does not correspond to an “open” direction. This is where the RG algorithm comes in. Rather than generating a host of useless conformations, it generates them “on the fly.” In addition, the algorithm uses a cheap test to check if a given branch will “die” within a specified number of steps (this number is denoted by l_{\max}). The algorithm then randomly chooses among the available open branches. As we have only looked a distance l_{\max} ahead, it may still happen that we have picked a branch that is doomed. But the probability of ending up in such a dead alley is much lower than that in the CBMC scheme.

In practice, the recoil growth algorithm consists of two steps. The first step is to grow a new chain conformation using only “open” directions. The next step is to compute the weights of the new and the old conformations.

The following steps are involved in the generation of a new conformation:

1. The first monomer of a chain is placed at a random position. The energy of this monomer is calculated (u_1). The probability that this position is “open” is given by Eq. (12.7.1). If the position is closed we cannot continue growing the chain and we reject the trial conformation. If the first position is open, we continue with the next step.
2. A trial position b_{i+1} for monomer $i + 1$ is generated starting from monomer i . We compute the energy of this trial monomer $u_{i+1}(b)$ and, using Eq. (12.7.1),

⁸ This probability can be chosen in many alternative ways and may be used to optimize a simulation. However, the particular choice discussed here appears to work well for Lennard-Jones and hard-core potentials.

we decide whether this position is open or closed. If this direction is closed, we try another trial position, up to a maximum⁹ of k trial orientations. As soon as we find an open position we continue with step 3.

If not a single open trial position is found, we make a recoil step. The chain retracts one step to monomer $i - 1$ (if this monomer exists), and the unused directions (if any) from step 2, for $i - 1$, are explored. If all directions at level $i - 1$ are exhausted, we attempt to recoil to $i - 2$. The chain is allowed to recoil a total of l_{\max} steps, i.e., down to length $i - l_{\max} + 1$.

If, at the maximum recoil length, all trial directions are closed, the trial conformation is discarded.

3. We have now found an “open” trial position for monomer $i + 1$. At this point monomer $i - l_{\max}$ is permanently added in the new conformation; i.e., a recoil step will not reach this monomer anymore.
4. Steps 2 and 3 are repeated until the entire chain has been grown.

In the naive version of the algorithm sketched above, we can consider the above steps as a procedure for searching for an open branch on the existing tree. However, the **RG** procedure does this by generating the absolute minimum of trial directions compatible with the chosen recoil distance l_{\max} .

Once we have successfully generated a trial conformation, we have to decide on its acceptance. To this end, we have to compute the weights, $W(n)$ and $W(o)$, of the new and the old conformations, respectively. This part of the algorithm is more expensive. However, we only carry it out once we know for sure that we have successfully generated a trial conformation. In contrast, in **CBMC** it may happen that we spend much of our time computing the weight factor for a conformation that terminates in a dead alley.

In the **RG** scheme, the following algorithm is used to compute the weight of the new conformation:

1. Consider that we are at monomer position i (initially, of course, $i = 1$). In the previous stage of the algorithm, we have already found that at least one trial direction is available (namely, the one that is included in our new conformation). In addition, we may have found that a certain number of directions (say k_c) are closed—these are the ones that we tried but that died within l_{\max} steps. We still have to test the remaining $k_{\text{rest}} \equiv k - 1 - k_c$ directions. We randomly generate k_{rest} trial positions for monomer $i + 1$ and use the recoil growth algorithm to test whether at least one “feeler” of length l_{\max} can be grown in this direction (unless $i + l_{\max} > l$; in that case, we only continue until we have reached the end of the chain). Note that, again, we do *not* explore all possible branches. We only check if there is at least *one* open branch of length l_{\max} in each of the k_{rest} directions. If this is the case, we call that direction “available.” We denote the total number of available directions (including the one that corresponds to the direction that we had

⁹ The maximum number of trial orientation should be chosen in advance—and may depend on the index i —but is otherwise arbitrary.

found in the first stage of the algorithm) by m_i . In the next section, we shall derive that monomer i contributes a factor $w_i(n)$ to the weight of the chain, where $w_i(n)$ is given by

$$w_i(n) = \frac{m_i(n)}{p_i^{\text{open}}(n)}$$

and $p_i^{\text{open}}(n)$ is given by Eq. (12.7.1).

2. Repeat the previous step for all i from 1 to $l - 1$. The expression for the partial weight of the final monomer seems ambiguous, as $m_l(n)$ is not defined. An easy (and correct) solution is to choose $m_l(n) = 1$.
3. Next compute the weight for the entire chain:

$$W(n) = \prod_{i=1}^{\ell} w_i(n) = \prod_{i=1}^{\ell} \frac{m_i(n)}{p_i^{\text{open}}(n)}. \quad (12.7.2)$$

For the calculation of the weight of the old conformation, we use almost the same procedure. The difference is that, for the old conformation, we have to generate $k - 1$ additional directions for every monomer i . The weight is again related to the total number of directions that start from monomer i and that are “available,” i.e., that contain at least one open feeler of length l_{\max} :

$$W(o) = \prod_{i=1}^{\ell} w_i(o) = \prod_{i=1}^{\ell} \frac{m_i(o)}{p_i^{\text{open}}(o)}.$$

Finally, the new conformation is accepted with a probability:

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta U(n)]W(n) / \exp[-\beta U(o)]W(o)), \quad (12.7.3)$$

where $U(n)$ and $U(o)$ are the energies of the new and old conformations, respectively. In the next section, we demonstrate that this scheme generates a Boltzmann distribution of conformations. The justification of the recoil-growth algorithm can be found in the SI section L.9.

Example 22 (Recoil growth simulation of Lennard-Jones chains). To illustrate the recoil growth (RG) method, we make a comparison between this method and configurational-bias Monte Carlo (CBMC). Consider 20 Lennard-Jones chains of length 15. The monomer density is $\rho = 0.3$ at temperature $T = 6.0$. Two bonded monomers have a constant bond length of 1.0, while three successive particles have a constant bond angle of 2.0 radians.

In Fig. 12.10 the distribution of the end-to-end vector, R_E , of the chain is plotted. In this figure we compare the results from a CBMC and a RG. Since both methods generate a Boltzmann distribution of conformations, the results are identical (as they should be).

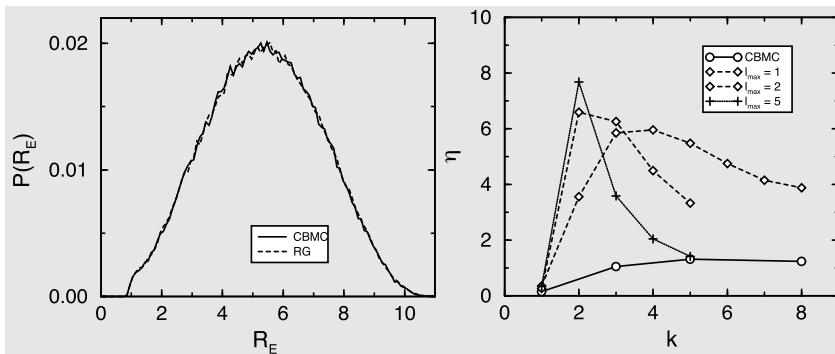


FIGURE 12.10 Comparison of configurational-bias Monte Carlo (CBMC) with recoil growth for the simulation of Lennard-Jones chains of length 15. The left figure gives the distribution of the end-to-end distance (R_E). In the right figure the efficiency (η) is a function of the number of trial directions (k) for different recoil lengths (l_{\max}) as well as for CBMC.

For this specific example, we have compared the efficiency, η , of the two methods. The efficiency is defined as the number of accepted trial moves per amount of CPU time. For CBMC we see that the efficiency increases as we increase k , the number of trial orientations, from 1 to 4. From 4 to 8 the efficiency is more or less constant, and above 8 a decrease in the efficiency is observed.

In the RG scheme we have two parameters to optimize: the number of trial orientations k and the recoil length l_{\max} . If we use only one trial orientation, recoiling is impossible, since there are no other trial orientations. If we use a recoil length of 1, the optimum number of trial orientations is 4 and for larger recoil lengths the optimum is reached with less trial orientations. Interestingly, the global optimum is 2 trial orientations and a recoil length of 3–5. In this regime, the increase in CPU time associated with a larger recoil length is compensated by a higher acceptance. In the present study, optimal RG was a factor 8 more efficient than optimal CBMC.

The Fortran code to generate this Example can be found in the online-SI, Case Study 20.

12.8 Questions and exercises

Question 25 (Biased CBMC). In a configurational-bias Monte Carlo simulation, trial positions are selected with a probability that is proportional to the Boltzmann factor of each trial segment. However, in principle, one can use another probability function [531] to select a trial segment. Suppose that the probability of selecting a trial segment i is proportional to

$$p_i \propto \exp[-\beta^* u_i]$$

in which $\beta^* \neq \beta$.

1. Derive the correct acceptance/rejection rule for this situation.
2. Derive an expression for the excess chemical potential when this modified CBMC method is used to generate configurations of test particles.
3. What happens if $\beta^* \rightarrow \infty$ and if $\beta^* \rightarrow 0$?

Exercise 16 (CBMC of a single chain). In this exercise, we will look at the properties of a single-chain molecule. We will compare various sampling schemes. Suppose that we have a chain molecule of length n in which there are the following interactions between beads:

- Two successive beads have a fixed bond length l . We will use $l = 1$.
- Three successive beads have a bond-bending interaction

$$U = \frac{1}{2}k_t (\theta - \theta_0)^2,$$

in which θ is the bond angle, θ_0 is the equilibrium bond angle, and k_t is a constant. We will use $\theta_0 = 2.0$ rad ($\approx 114.6^\circ$) and $k_t = 2.0$.

- Every pair of beads that is separated by more than two bonds has a soft repulsive interaction

$$U(r) = \begin{cases} \frac{A(r-r_{\text{cut}})^2}{r_{\text{cut}}^2} & r \leq r_{\text{cut}} \\ 0 & r > r_{\text{cut}} \end{cases},$$

in which r_{cut} is the cutoff radius (we will use $r_{\text{cut}} = 1.0$ and $A > 0$).

An interesting property of a chain molecule is the distribution of the end-to-end distance, which is the distance between the first and the last segments of the chain. There are several possible schemes for studying this property:

Dynamic schemes

In a dynamic scheme, a Markov chain of states is generated. The average of a property B is the average of B over the elements of the Markov chain

$$\langle B \rangle \approx \frac{\sum_{i=1}^{i=N} B_i}{N}.$$

In the limit $N \rightarrow \infty$ this expression becomes exact. Every new configuration is accepted or rejected using an acceptance criterion:

- When unbiased chains are generated:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min(1, \exp\{-\beta [U(\text{n}) - U(\text{o})]\}),$$

in which U is the total energy (soft repulsion and bond bending) of a chain.

- When configurational-bias Monte Carlo is used:

$$\text{acc}(\text{o} \rightarrow \text{n}) = \min\left(1, \frac{W(\text{n})}{W(\text{o})}\right),$$

in which

$$W = \frac{\prod_{i=2}^{i=n} \sum_{j=1}^{j=k} \exp[-\beta U(i, j)]}{k^{n-1}}.$$

In this equation, k is the number of trial positions and $U(i, j)$ is the energy of the j th trial position of the i th chain segment. The term $U(i, j)$ does not contain the bond-bending potential, because that potential has already been used to generate the trial positions.

Static schemes

In a static scheme, all configurations are generated independently. To obtain a canonical average, every configuration is weighted with a factor R

$$\langle B \rangle = \frac{\sum_{i=1}^{i=N} B_i \times R_i}{\sum_{i=1}^{i=N} R_i}.$$

For R_i we can write:

- When random chains are generated:

$$R_i = \exp[-\beta U_i].$$

Here, U_i is the total energy of the chain.

- When CBMC is used:

$$R_i = W. \quad (12.8.1)$$

1. On the book's website you can find a program for calculating chain properties using these four methods. However, some additional programming has to be done in the file *grow.f*, which contains a routine for growing a new chain using either CBMC or random insertion.
2. Compare the end-to-end distance distributions of the four methods. Which method has the best performance? Investigate how the efficiency of CBMC depends on the number of trial directions (k).
3. Investigate the influence of chain length on the end-to-end distance distribution. For which chain lengths do the four methods start to fail?
4. For high temperatures (and for low k_f and A), the end-to-end distance distribution looks like the distribution of a nonself-avoiding random walk. This means that the chain segments are randomly oriented and the segments are allowed to overlap. For the mean square end-to-end distance, we can write

$$\frac{\langle r^2 \rangle}{l^2} = \left\langle \left(\sum_{i=1}^{i=n} x_i^2 \right) + \left(\sum_{i=1}^{i=n} y_i^2 \right) + \left(\sum_{i=1}^{i=n} z_i^2 \right) \right\rangle,$$

in which (x_i, y_i, z_i) are the projections of each segment on the (x, y, z) axes

$$x_i = \sin(\theta_i) \cos(\phi_i)$$

$$y_i = \sin(\theta_i) \sin(\phi_i)$$

$$z_i = \cos(\theta_i).$$

This set of equations can be reduced to

$$\frac{\langle r^2 \rangle}{l^2} = n. \quad (12.8.2)$$

- Derive Eq. (12.8.2). Hint: the following equations will be very useful:

$$\begin{aligned} \cos^2(\theta_i) + \sin^2(\theta_i) &= 1 \\ \cos(\theta_i - \theta_j) &= \cos(\theta_i)\cos(\theta_j) + \sin(\theta_i)\sin(\theta_j) \\ \langle \cos(\theta_i - \theta_j) \rangle &= 0. \end{aligned}$$

The last equation holds because $\theta_i - \theta_j$ is uniformly distributed.

- Modify the program in such a way that $\langle r^2 \rangle$ is calculated for a nonself-avoiding random walk. Compare your results with the analytical solution.
- Does

$$\langle r^2 \rangle \propto n$$

hold for a chain with a potential energy function described in this exercise? Investigate the influence of A on the end-to-end distance distribution.

Chapter 13

Accelerating Monte Carlo sampling

The key advantage of Molecular Dynamics simulations is that they generate physically realistic trajectories of a classical many-body system. The key advantage of Monte Carlo simulations is that they do not. That is: MC simulations can explore configuration space in a way that is incompatible with Newton's equations of motion, or even with Brownian dynamics. Were it not for this feature, MC simulations of off-lattice systems would probably have died out long ago.

In this chapter, we discuss a variety of Monte Carlo algorithms that can greatly enhance the speed with which configuration space is explored, compared with the simple Metropolis-style MC simulations. However, partly because of the availability of powerful simulation packages, MD simulations are much more widely used than MC simulations, and many of the ideas that were first explored in the context of MC have subsequently been ported to MD, where they became more widely known than their MC parents. For a discussion of the ideas behind the algorithms, it is, however, more convenient to use the MC language, although we will mention some of the MD off-spring.

13.1 Sampling intensive variables

The Monte Carlo simulations that we considered thus far were designed to study the behavior of systems at constant NVT , NPT , or μVT . In such simulations, the relevant intensive thermodynamic variables (T , P , or μ) are kept fixed, while the microscopic configurations of the system are explored by performing Metropolis MC moves that change the particle coordinates. In addition, for NPT -MC, we carry out trial moves that change the volume V , and for μVT -MC simulations, we attempt moves that change the number of particles.

In the present chapter, we consider a different class of MC simulations where we still carry out particle moves, but in addition, we perform trial moves that change an intensive variable such as T , P , or μ (see Fig. 13.1). In fact, as we shall see below, this approach can be extended to cover other intensive parameters in the Hamiltonian of the system, e.g., parameters that determine the strength or the range of interactions between particles. This general class of simulations can be decomposed into two sub-classes: one in which we let a single system explore different values of the intensive variable (say, the temperature).

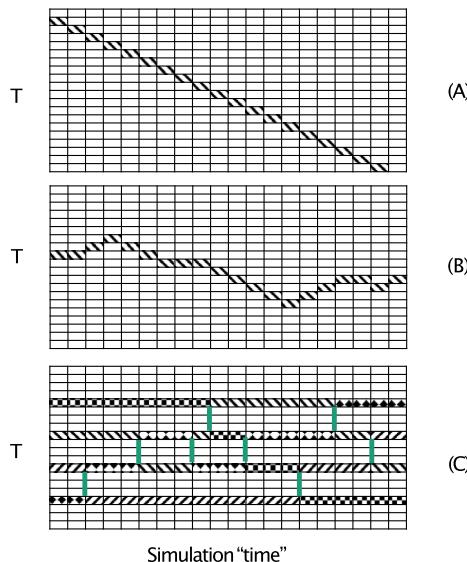


FIGURE 13.1 This figure compares three different Monte Carlo schemes to couple simulations carried out at different values of a control parameter (in this example: the temperature T). Panel (A) shows how, in a simulated annealing simulation (which is not really an equilibrium Monte Carlo scheme), the temperature is gradually lowered, allowing the system to find low-energy states that would be missed in an instantaneous quench. Panel (B) shows how, in an expanded-ensemble simulation, a random change of the temperature is a valid Monte-Carlo trial move. To ensure uniform sampling, such simulations require a judiciously chosen bias. Panel (C) shows an example of a parallel tempering simulation. Here we prepare n different systems (in the figure, only 4) at different initial temperatures. In addition to the regular Monte Carlo moves, we now also allow trial moves that swap the temperatures attributed to the different systems (identified by different patterns). Hence, at every stage in the simulations, we always have n systems at n distinct temperatures.

Such simulations are known under the name *Expanded-ensemble simulations*, sometimes referred to as “*extended-ensemble simulations*” [543], for reasons that shall become clear shortly. The second class of algorithms that allow us to sample intensive variables, considers the parallel evolution of n systems that are initially prepared at n different temperatures (say). In a trial move, we attempt to swap the temperatures of two adjacent systems. Algorithms of this type also have various names, among which parallel tempering and *replica exchange* are the most common. In what follows, we will mostly use the term **Parallel Tempering**.

Before discussing these algorithms, we should address one obvious question: *Why?* Or, more precisely: is there an advantage associated with expanding the move repertoire to include changes in intensive variables? The short answer is: it depends. For most MC simulations, performing trial moves in intensive variables only complicates the program, and does not add to the quality of the simulations. However, in some cases, expanded-ensemble simulations greatly

enhance the ability of a simulation to explore the accessible configuration space—typically, if different parts of the accessible configuration space are separated by barriers that are effectively insurmountable at the temperature (or pressure, or chemical potential) of interest.

To give an explicit example: suppose that we wish to sample an amorphous solid at low temperatures. We know that the particles in an amorphous solid can be arranged in many different ways. But at the low temperatures where the amorphous phase is mechanically stable, normal MC sampling will not allow the system to escape from the vicinity of its original arrangement. Clearly, if we were to heat the system, and then cool it down again, it would be easy to move from one amorphous structure to the next, but in general, such a heating-cooling protocol does not result in sampling states with their correct Boltzmann weight. Expanded-ensemble simulations allow us to implement heating-cooling protocols in such a way that all the states of the system that we sample (including at all intermediate temperatures) are *bona-fide* equilibrium states. Hence, all configurations that we sample can be used to compute observable properties. In contrast, if we would simply heat and cool the system, most of our simulation points would not correspond to equilibrium states of the system.

There is a nomenclature problem with expanded-ensemble simulations and related methods such as parallel tempering: these methods have been (re-)discovered many times in slightly different forms (see ref. [543]) —all correct, but also all very similar in spirit. We will not try to provide a taxonomy of the many algorithms that achieve correct sampling of intensive variables, but just distinguish the main classes.

For the sake of simplicity, we first describe the case where we include sampling of the temperature of a system. After that, we quickly generalize to the case where we allow variation in some other intensive parameter that characterizes the system.

13.1.1 Parallel tempering

The [Parallel Tempering \(PT\)](#) method for sampling intensive variables is closest in spirit to the original Markov-Chain MC method. This method has been invented and re-invented in various forms and with various names [544–548], and was also introduced in Molecular Dynamics simulations under the name *replica-exchange MD* [549]. Here, we introduce the PT algorithm, in the context of simulations that allow transitions between different temperatures. After that, we discuss the general case.

The simplest way of viewing the PT algorithm is by considering n simultaneous simulations of an N particle system. At this stage, we will assume that both V and N are constant, although sometimes it is more convenient to keep P or μ (but not both) fixed. The n parallel simulations differ in the values T_1, T_2, \dots, T_n that we have chosen for the imposed temperatures. At this stage, the choice of these temperatures is still arbitrary. Later, we shall see that the spacing between

adjacent temperatures should be chosen such that the energy fluctuations in adjacent systems still have some overlap.

Up to this point, we can consider the combined system simply as n noninteracting systems each at its own temperature T_i . The probability density to find system 1 in configuration $(\mathbf{r}^N)_1$, system 2 in configuration $(\mathbf{r}^N)_2$ etc. follows directly from Eq. (2.3.10):

$$P\left(\{\mathbf{r}^N\}_1, \{\mathbf{r}^N\}_2, \dots, \{\mathbf{r}^N\}_n\right) = \frac{\prod_{i=1}^n e^{-\beta_i U(\{\mathbf{r}^N\}_i)}}{\prod_{i=1}^n Z(N, V, T_i)}. \quad (13.1.1)$$

We could study this combined system with a standard MC simulation where only moves between two configurations at the same temperature are allowed, but clearly, such a simulation would not have any advantage over n separate simulations.

Now consider what would happen if we allowed a massive particle swap, where we would permute the configurations of systems at temperature T_i and T_j . Usually, we choose i random, and j adjacent to i . To guarantee microscopic reversibility, trial swaps for which $j < 1$ or $j > n$ should be rejected. Such a particle swap, which we denote by $(\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j) \rightarrow (\mathbf{j}, \beta_i), (\mathbf{i}, \beta_j)$, is a perfectly legitimate MC trial move. For such a trial move, the Boltzmann weight of the system as a whole would change. Denoting the configuration of system i by $\mathbf{i} = \{\mathbf{r}_i^N\}$, the condition for detailed balance reads

$$\begin{aligned} & \mathcal{N}(\mathbf{i}, \beta_i) \mathcal{N}(\mathbf{j}, \beta_j) \times \alpha [(\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j) \rightarrow (\mathbf{j}, \beta_i), (\mathbf{i}, \beta_j)] \\ & \quad \times \text{acc}[(\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j) \rightarrow (\mathbf{j}, \beta_i), (\mathbf{i}, \beta_j)] \\ & = \mathcal{N}(\mathbf{i}, \beta_j) \mathcal{N}(\mathbf{j}, \beta_i) \times \alpha [(\mathbf{i}, \beta_j), (\mathbf{j}, \beta_i) \rightarrow (\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j)] \\ & \quad \times \text{acc}[(\mathbf{i}, \beta_j), (\mathbf{j}, \beta_i) \rightarrow (\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j)]. \end{aligned}$$

If we perform the simulations in such a way that the *a priori* probability, α , of performing a particular swap move is equal for all conditions, we obtain as acceptance rule

$$\begin{aligned} & \frac{\text{acc}[(\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j) \rightarrow (\mathbf{j}, \beta_i), (\mathbf{i}, \beta_j)]}{\text{acc}[(\mathbf{i}, \beta_j), (\mathbf{j}, \beta_i) \rightarrow (\mathbf{i}, \beta_i), (\mathbf{j}, \beta_j)]} \\ & = \frac{\exp[-\beta_i U(\mathbf{j}) - \beta_j U(\mathbf{i})]}{\exp[-\beta_i U(\mathbf{i}) - \beta_j U(\mathbf{j})]} \\ & = \exp\{(\beta_i - \beta_j)[U(\mathbf{i}) - U(\mathbf{j})]\}, \end{aligned} \quad (13.1.2)$$

where we have defined $U(\mathbf{i}) \equiv U(\{\mathbf{r}^N\}_i)$.

It is important to note that, as we know the total energy of a configuration anyway, these swap moves are very inexpensive since they do not involve additional calculations.

With this kind of massive swap move, we can exchange configurations that were generated at different temperatures, while maintaining the equilibrium at

all temperatures T_i . In particular, we can swap configurations at lower temperatures where equilibration is slow, with those at higher temperatures, where normal MC sampling can adequately sample the configuration space.

Of course, in reality, we do not move particles from configuration i to j and *vice versa*. Rather, what we do is swap the (inverse) temperatures β_i and β_j . One could therefore interpret a parallel tempering trial move as an attempt to change (permute) the intensive variable T . However, when computing averages for a given temperature T_i , it is more convenient to think in terms of moves that change the configuration of a system at a constant temperature. With PT, interpreting trial moves in terms of changes of the intensive variable is only a matter of words. However, below we shall discuss extended ensembles, where no such ambiguity exists.

Example 23 (Parallel tempering of a single particle). As an illustration of the power of parallel tempering, we consider a single particle moving an external potential as shown in Fig. 13.2(left):

$$U(x) = \begin{cases} \infty & x < -2 \\ 1 \times (1 + \sin(2\pi x)) & -2 \leq x \leq -1.25 \\ 2 \times (1 + \sin(2\pi x)) & -1.25 \leq x \leq -0.25 \\ 3 \times (1 + \sin(2\pi x)) & -0.25 \leq x \leq 0.75 \\ 4 \times (1 + \sin(2\pi x)) & 0.75 \leq x \leq 1.75 \\ 5 \times (1 + \sin(2\pi x)) & 1.75 \leq x \leq 2 \\ \infty & x > 2 \end{cases} \quad (13.1.3)$$

We place the particle initially in the left-most potential energy well, and then we first use normal Metropolis MC at three different temperatures ($T = 0.05$, 0.3 and 2.0). At the lowest temperature ($T = 0.05$) the particle is effectively trapped in its initial potential-energy well during the entire simulation, whereas for the highest temperature ($T = 2.0$) it can explore all wells.

Next we apply parallel tempering, that is: we allow for temperature swaps between the three systems (see Fig. 13.2). Due to the temperature-swap moves, the systems now equilibrate rapidly at all three temperatures. The difference is particularly strong for the probability distribution at the lowest temperature.

In the present parallel-tempering simulation, we consider two types of trial moves:

1. Particle displacement: we randomly select one of the three temperatures and carry out a trial displacement Δ of a randomly selected particle at that temperature, choosing the trial displacement Δ for a random distribution, uniform between -0.1 and 0.1 . The acceptance of this trial displacement is determined by the conventional Metropolis MC rule

$$\text{acc}(o \rightarrow n) = \min \{1, \exp[-\beta(U(n) - U(o))]\}. \quad (13.1.4)$$

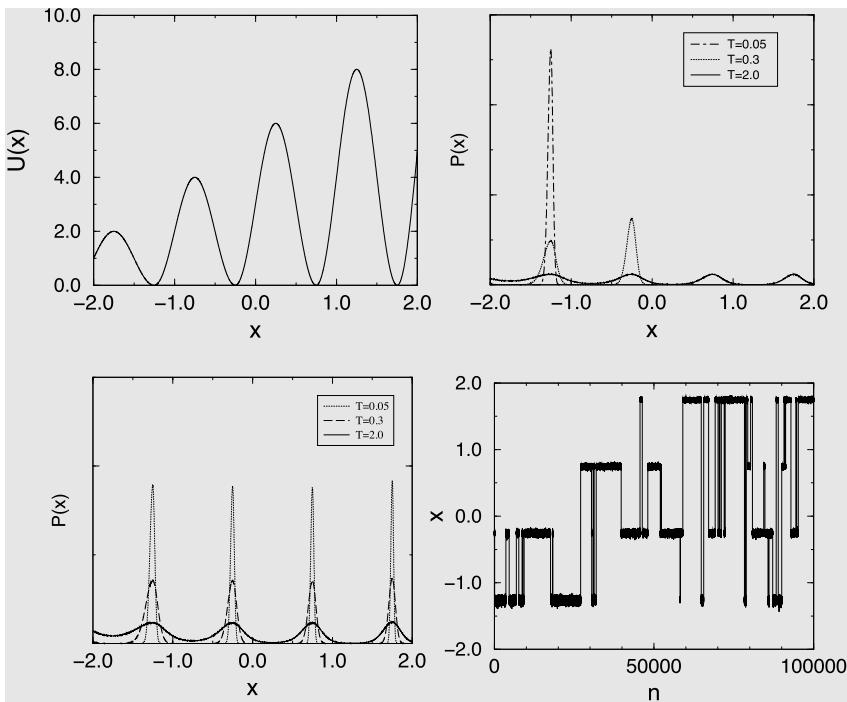


FIGURE 13.2 (Top left) Potential energy ($U(x)$) as a function of the position x . (Top right) Probability ($P(x)$) of finding a particle at position x for various temperatures (T) as obtained from ordinary Monte Carlo simulations and (bottom left) using parallel tempering. In the ordinary MC simulations, the lower-temperature systems are not (or barely) able to cross the energy barriers separating the wells. (bottom right) Position (x) as a function of the number of Monte Carlo trial moves (n) for $T = 0.05$.

2. Temperature swapping. The trial move consists of attempting to swap two randomly selected neighboring temperatures (T_i and T_j). Such a trial move is accepted with a probability given by Eq. (13.1.2)

$$\text{acc}(o \rightarrow n) = \min\left(1, \exp\left[(\beta_i - \beta_j) \times (U_j - U_i)\right]\right). \quad (13.1.5)$$

We are free to choose the relative rate of displacement and swap moves. In the present example, we used 10% swap moves and 90% particle displacements, as suggested in ref. [550]. Note, however, that other choices are possible.^a As can be seen in Fig. 13.2, parallel tempering results in a dramatic improvement in the sampling of configuration space at the lowest temperature.

For more details, see SI (Case Study 21).

^a There are even algorithms that use an infinite swap rate [551], including all possible permutations of the temperatures. However, such algorithms (see also [552]) do not scale well with the number of distinct temperatures.

One obvious question is: how many parallel simulations are needed in a PT run, and how should we choose the temperature spacing? On the one hand, we want to use as few simulations as possible, which means choosing the temperature spacing as large as possible. But if the temperature spacing becomes too large, the acceptance of the swap moves plummets. A good (but probably not optimal) rule of thumb is to choose the separation of neighboring temperatures such that the spacing in the average energy of adjacent systems is comparable to the root-mean-square fluctuations in the energy of these systems.

This rule allows us to make a quick estimate of the temperature spacing. Note that the difference in the average energy of two systems separated by a temperature interval ΔT is equal to $\Delta E \approx C_V \Delta T$, where $C_V(T)$ is the heat capacity of the system at temperature T and volume V . But the variance in the energy of the system is also related to the heat capacity (Eq. (5.1.7)):

$$k_B T^2 C_V = [\langle E^2 \rangle - \langle E \rangle^2].$$

Hence, our rule to estimate the optimal value of ΔT is

$$\sqrt{k_B T^2 C_V} \approx C_V \Delta T \quad (13.1.6)$$

or

$$\frac{\Delta T}{T} \approx \sqrt{k_B / C_V}. \quad (13.1.7)$$

As the heat capacity is an extensive variable (it scales with the number of particles N), Eq. (13.1.7) shows that $\Delta T/T$ scales as $1/\sqrt{N}$. Hence, parallel tempering becomes less efficient as N grows.

The parallel tempering approach is not limited to coupling simulations at different temperatures. We can use parallel tempering for any intensive control parameter—or any combination thereof. Obvious examples are the pressure P or the chemical potential μ (but not both)—see e.g., ref. [553]. But we can also consider PT simulations that couple systems with different potential-energy functions. To capture all these cases in a single notation, we can generalize the simple Boltzmann weight $w_B(\mathbf{i}; \beta_i) = \exp[-\beta_i U(\mathbf{i})]$ to $w(\mathbf{i}; \lambda_i)$, where λ_i stands for any set of intensive variables that we intend to vary. In this notation, the generalized expression for the acceptance probability of a trials move that swaps λ_i and λ_j is:

$$\begin{aligned} & \frac{\text{acc}[(\mathbf{i}, \lambda_i), (\mathbf{j}, \lambda_j)] \rightarrow (\mathbf{j}, \lambda_i), (\mathbf{i}, \lambda_j)]}{\text{acc}[(\mathbf{i}, \lambda_j), (\mathbf{j}, \lambda_i)] \rightarrow (\mathbf{i}, \lambda_i), (\mathbf{j}, \lambda_j)]} \\ &= \frac{w(\mathbf{i}; \lambda_j) w(\mathbf{j}; \lambda_i)}{w(\mathbf{i}; \lambda_i) w(\mathbf{j}; \lambda_j)}. \end{aligned} \quad (13.1.8)$$

Examples of parallel tempering simulations that sample a parameter in the Hamiltonian of a system are, for instance, the simulations of Yan and de Pablo

[554] of a polymer mixture: the intensive variables in these simulations were the temperature, and the length of a tagged polymer chain (see Illustration 19).

Illustration 19 (Parallel tempering and phase equilibria). Another example that illustrates the power of parallel tempering simulations is the study of liquid-vapor coexistence. Typically, if a fluid is heated above the critical temperature, the density is a monotonically increasing function of the chemical potential. However, below the critical temperature, the density of a macroscopic fluid sample will jump from a vapor-like value to a liquid-liquid value as the chemical potential crosses its value at coexistence (μ_{coex}). In a constant μVT simulation of a system containing only a few particles, coexistence will show up in a different way: for μ well below μ_{coex} , the density of a fluid will fluctuate around a vapor-like value, and for μ well above μ_{coex} , the fluid density will fluctuate around a liquid-like value. However, close to coexistence, the system should be able to exist in a vapor-like and a liquid-like state and precisely at coexistence, the areas under the vapor and liquid peaks in the density distribution should be equal, provided that the system can equilibrate. However, apart from simulations close to the critical point, where large density fluctuations carry a low free-energy cost [176,212], it is rare to observe transitions between liquid-like and vapor-like densities: at temperatures well below T_c , the free energy barrier separating the two states makes equilibration simply too slow.

This is where parallel tempering can make a difference. Yan and de Pablo [553] used simulations at a number of different points (in their case 18) in the $\mu - T$ plane, linking the low-temperature vapor and liquid branches of the fluid via a path that passes close to T_c (where large density fluctuations can occur). As the method of ref. [553] applies parallel-tempering swaps to both T and μ , Yan and de Pablo called this method hyper-parallel tempering.

Deviating a bit from the original presentation of ref. [553], we will use the *fugacity* f to characterize the chemical potential of a system. We recall that the fugacity of a fluid can be interpreted as the density of a hypothetical ideal gas of the same molecules at the same temperature that has the same chemical potential as the fluid:

$$\beta\mu(\beta, \rho) = \ln(\rho) + \beta\mu^{\text{ex}}(\rho, \beta) \equiv \ln f(\rho, \beta),$$

where we have dropped all “uninteresting” constants. Note that at low densities, $f \rightarrow \rho$, as it should.

In a parallel-tempering move, we exchange the fugacities and the temperatures of the two systems without changing the configuration of either system. The new configuration of system i is then characterized by $i (U_i, N_i, \beta_j, \ln f_j)$, while for system j we have $(U_j, N_j, \beta_i, \ln f_i)$. Such a trial move is accepted with probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \exp \left[(\ln f_j - \ln f_i)(N_i - N_j) - (\beta_j - \beta_i)(U_i - U_j) \right] \right).$$

Because these parallel tempering simulations connect equilibrium state points on the liquid and vapor side of the critical point, equilibration is no longer a problem, and a density distribution is obtained, which, close to coexistence, contains both the liquid and vapor peaks.

Of course, none of the state points studied in this parallel tempering calculation will be located exactly on the liquid-vapor coexistence curve. However, as we know the density histograms as a function of β and f , we can use the histogram-reweighting technique discussed in section 8.6.11. The histogram reweighting procedure allows us to estimate the density distribution for arbitrary points in the f, β -plane close to the points that were sampled. We can then locate those points in the f, β -plane for which the probability of finding the system at a liquid density is equal to the probability of finding it at a vapor density (i.e., the area under the two peaks in the reweighted histogram are equal).

As shown in ref. [553], the combined parallel temperature and histogram reweighting technique is very efficient in determining the coexistence curve of the Lennard-Jones fluid, provided that one has a well-chosen set of temperatures and chemical potentials. Using the approach of Wilding and Bruce [176,212], Yan and de Pablo also used the histogram-reweighting techniques to estimate the location of the liquid-vapor critical point.

A closely related example is the work of Bunker and Dünweg [555], who used the excluded volume of polymer chains as the intensive variable to be sampled: typically, the relaxation of polymer conformations in dense melts is very slow, but becomes much faster as the polymers interact less (see Illustration 20). Of course, if the scheme is only meant to equilibrate the melt of interacting polymers, the overhead involved in running additional “unphysical” simulations makes the scheme more expensive. If at all possible, PT should be used in situations where the information from all intermediate values of the intensive variable is of interest, as is for instance the case when PT is used in the context of computing free-energy landscapes (e.g., nucleation barriers [556]), and even when performing thermodynamic integration (section 8.4.1)—see ref. [155]. As such calculations require several simulations anyway, the sampling efficiency can be increased at no extra cost by adding parallel tempering moves.

Illustration 20 (Parallel tempering and polymers). In a Parallel Tempering (PT) simulation, one can also perform swaps between systems that have slightly different Hamiltonians. In some cases, this can be very useful. Suppose that system i has a temperature T and energy $U_i^{(i)} = \sum_{k>l}^N u^{(i)}(r_{kl})$ and system j a temperature T and energy $U_j^{(j)} = \sum_{k>l}^N u^{(j)}(r_{kl})$, where $u^{(i)}$ and $u^{(j)}$ are different potentials. As a parallel tempering move, we can swap systems i and j . We take the positions of the particles in system i and recompute the energy using the intermolecular potential of system j ; this energy is denoted by $U_i^{(j)}$.

In a similar way, we compute for system j the energy using the intermolecular potential of system i , $U_j^{(i)}$. The acceptance rule for such a move reads

$$\text{acc}(o \rightarrow n) = \min \left(1, \exp \left[-\beta(U_i^{(j)} - U_i^{(i)}) + (U_j^{(i)} + U_j^{(j)}) \right] \right).$$

This type of parallel tempering move can be combined with others involving the temperature, pressure, or chemical potential.

Bunker and Dünweg used “Hamiltonian” parallel tempering to simulate a long-chain polymer. The polymer was modeled using a bead-spring model with a purely repulsive Lennard-Jones interaction between the beads:

$$U^{\text{pol}}(r) = \begin{cases} A - Br^2 & r \leq r_{\text{PT}} \\ 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] & r_{\text{PT}} < r \leq 2^{1/6}\sigma, \\ 0 & r > 2^{1/6}\sigma \end{cases}$$

where A and B were chosen such that, for a given value of r_{PT} , the potential and its first derivative were continuous. We are interested in the properties of the model system with $r_{\text{PT}} = 0$. The other systems are simply added to facilitate equilibration. For instance, for $r_{\text{PT}} = 2^{1/6}\sigma$, the core repulsion vanishes, and polymer chains can pass through each other. In their parallel tempering scheme, Bunker and Dünweg simulated a number of systems with different values of r_{PT} . A drawback of this “Hamiltonian” parallel tempering scheme is that most of the simulation time is spent on systems that are *not* of physical interest. After all, we are only interested in the thermodynamic behavior of the system with $r_{\text{PT}} = 0$. However, as is discussed in ref. [555], the gain in sampling efficiency due to the use of the parallel tempering scheme is still sufficient to make the scheme competitive.

13.1.2 Expanded ensembles

The parallel tempering approach that we discussed in the previous section could still be viewed as a normal Monte Carlo scheme, if we chose to interpret the moves as particle swaps, rather than as swaps of the intensive variables. Now we consider simulation methods where the elementary trial move involves changing an intensive variable; in the following, we will take the temperature as an example.

Similar to parallel tempering, the idea behind *expanded-ensemble* simulations is to speed up the equilibration of a system by allowing excursions to other thermodynamic conditions where configuration space can be explored more efficiently, whilst maintaining thermodynamic equilibrium. Expanded-ensemble simulations and their predecessors have many inventors (see [543]), but the term “Expanded-ensemble method” seems to have been coined by Lyubartsev et al. [546].

There is a conceptual problem with such simulations: for normal Markov Chain MC moves, we can decide on the acceptance probability of a trial move by considering the ratio of the Boltzmann weight of the new and the old states. However, if we change (say) the temperature of a system, there is no natural way to compare probabilities. We cannot say that the probability of a given configuration will be higher or lower when we change the temperature. The underlying reason for this indeterminacy is that intensive variables are not properties of the system itself but of an external “reservoir”. Whilst this situation may seem to present a problem, it is in fact an advantage, because it offers us more freedom to tune the properties of expanded ensembles.

As before, we start from the expression for the probability distribution of a system at a given inverse temperature β :

$$P(\mathbf{r}^N) = \frac{e^{-\beta U(\mathbf{r}^N)}}{Z(N, V, T)}. \quad (13.1.9)$$

Next, we allow this system to take on a range of temperatures T_1, T_2, \dots, T_n . We now define a weighted, extended configurational integral $Z(N, V, \{T\})$ as

$$Z(N, V, \{T\}) \equiv \sum_{i=1}^n e^{\eta_i} Z(N, V, T_i). \quad (13.1.10)$$

The probability to find the system in volume V in configuration \mathbf{r}^N and temperature T_i is then given by:

$$P(\mathbf{r}^N; T_i) = \frac{e^{\eta_i} e^{-\beta_i U(\mathbf{r}^N)}}{\sum_{j=1}^n e^{\eta_j} Z(N, V, T_j)}. \quad (13.1.11)$$

Note that at this stage the choice of the weighting factors e^{η_i} is left open.

Having defined a probability distribution for a system that can be found at n different temperatures, we can construct the rules for trial moves that change the temperature. We follow the usual recipe where we require microscopic reversibility and impose detailed balance. Then the acceptance probability for a trial move that changes the inverse temperature from β_i to β_j is given by the condition for detailed balance:

$$\frac{\text{acc}(i \rightarrow j)}{\text{acc}(j \rightarrow i)} = \exp(\eta_j - \eta_i) \exp[-(\beta_j - \beta_i)U(\mathbf{r}^N)] \quad (13.1.12)$$

Note that if we compute some observable for the state with temperature T_i , we will obtain the normal Boltzmann average for that temperature. Hence, for every temperature, we perform equilibrium sampling.

Of course, as in the case of parallel tempering, we could sample other intensive variables using another form of the weight function (see Eq. (13.1.8)). We

will not go through the same steps here, as it adds nothing, as long as it is clear that in the discussion below, temperature stands for a wider class of variables.

The first question to address is: what values should we choose for the weighting factors e^{η_i} . It should be noted that the values chosen for the η_i do not affect the values of the equilibrium averages, only the efficiency with which states with different T_i are sampled.

The simplest (but not necessarily the best) choice would be the one that results in equal sampling of all temperatures T_1 through T_n . We note that the probability of finding the system in any configuration at temperature T_i is

$$P(T_i) = \frac{e^{\eta_i} Z(N, V, T_i)}{\sum_{j=1}^n e^{\eta_j} Z(N, V, T_j)} = \frac{e^{\eta_i - \beta_i F(N, V, T_i)}}{\sum_{j=1}^n e^{\eta_j - \beta_j F(N, V, T_j)}}. \quad (13.1.13)$$

Hence, if we were to choose the η_i such that $\eta_i = F(N, V, T_i) + \text{constant}$, then all T_i s would be equally likely. We do not have to make this choice, but clearly, we do not wish to use weights that would make certain temperatures orders of magnitude less likely than others, because then these temperatures would be under-sampled.

Of course, we do not know the free energy $F(N, V, T_i)$ *a priori*. However, we can make a good initial estimate (up to an unimportant constant) by using the fact that

$$\left(\frac{\partial \beta F(N, V, T)}{\partial \beta} \right)_{N, V} = E(N, V, T), \quad (13.1.14)$$

and we can compute $E(N, V, T)$ in a normal MC simulation. But the expanded-ensemble simulations provide us with an estimate of the free-energy difference between two states i and j . If we sample the equilibrium probabilities $P(T_i)$ and $P(T_j)$ that states i and j are populated, then we can estimate $F(N, V, T_i) - F(N, V, T_j)$ from

$$\frac{P(T_i)}{P(T_j)} = e^{(\eta_i - \eta_j)} e^{-\beta_i F(N, V, T_i) + \beta_j F(N, V, T_j)}. \quad (13.1.15)$$

However, populating all states in the expanded ensemble equally may not be an optimal choice. Often, we wish to maximize the rate at which the system “diffuses” between T_1 and T_n . More importantly, in the case where the intermediate states correspond to unphysical values of some other intensive parameter (such situations occur, for instance, if we consider the gradual insertion of a molecule in the system), we may only be interested in computing accurate averages for those parameter values that have physical meaning. Under those conditions, a more sensible optimization criterion might be to choose the set $\{\eta_i\}$ such that it maximizes the rate of “diffusion” between the different physically meaningful states. In such cases, it is better to choose the η_i such that the population of state i is proportional to $1/\sqrt{D_i}$, where D_i is the local “diffusion coefficient”

of the intensive variable [378,557,558]. Clearly, D_i will depend on the acceptance of trial moves that change the intensive variable, and on the spacing in the temperatures —and therefore D_i may vary strongly with i .

Escobedo and coworkers have compared different strategies to optimize the sampling in expanded ensembles, and we refer the reader to their work for more details [559–561].

13.2 Noise on noise

Thus far, we have assumed that once we know the coordinates of all particles in a system, we can directly evaluate the value of the potential energy. However, sometimes computing the potential energy itself requires a sampling process, and hence the resulting value of U is subject to statistical noise. Of course, the noise can be decreased by sampling longer, but surprisingly, that is not always necessary. Ceperley-Dewing [562] showed that under certain conditions, it is possible to perform Boltzmann sampling of configuration space, even if our estimate of the potential energy is subject to random statistical errors.

For convenience, we focus on the case of equal *a priori* probabilities for forward and reverse moves. In the absence of noise, the acceptance probability of a trial move from a state o to a new state n , which changes the potential energy by an amount $\Delta U = U(n) - U(o)$ is, as usual, given by

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta \Delta U]). \quad (13.2.1)$$

Now assume that there are statistical errors in the potential energy difference, i.e., we must decide the acceptance of a trial move from o to n based on our knowledge of $\delta = \Delta U + x$, where x describes the noise in our estimate of ΔU . The new stochastic variable δ follows some distribution $P(\delta)$, such that $\langle \delta \rangle = \Delta U$. The acceptance probability of trial moves will now depend on δ : $\text{acc}(\delta; o \rightarrow n)$. The average acceptance probability for trial moves between o and n is

$$\text{acc}'(o \rightarrow n) = \int d\delta P(\delta) \text{acc}(\delta; o \rightarrow n). \quad (13.2.2)$$

To obtain Boltzmann sampling, we require that

$$\frac{\text{acc}'(o \rightarrow n)}{\text{acc}'(n \rightarrow o)} = \exp[-\beta \Delta U], \quad (13.2.3)$$

which implies that

$$\int d\delta P(\delta) \text{acc}(\delta; o \rightarrow n) = \exp[-\beta \Delta U] \int d\delta P(\delta) \text{acc}(-\delta; n \rightarrow o). \quad (13.2.4)$$

In general, we do not know the true ΔU , nor do we know $P(\delta)$, and as a consequence, we cannot determine $\text{acc}(\delta; n \rightarrow o)$. However, Ceperley and Dewing

showed that if we assume that $P(\delta)$ is distributed normally around ΔU , with a variance σ^2 , then things simplify dramatically. The only additional information we need is a good estimate of σ^2 , which can be sampled directly. Under those circumstances, the choice

$$\text{acc}(\delta; o \rightarrow n) = \min \left(1, \exp \left[-\beta \delta - (\beta \sigma)^2 / 2 \right] \right) \quad (13.2.5)$$

will generate the correct Boltzmann distribution over all states sampled (as shown in ref. [562]). Of course, the assumption that *the same* normal distribution describes the variation in δ around ΔU for all pairs o and n is almost never exact (although often reasonable). Hence, when computing σ^2 , it is useful to check if the distribution $P(\delta)$ is normal.

Noise can affect Monte Carlo sampling in several ways. In the case discussed above, we considered the case where the energy function U was subject to random noise, in such a way that the average of U over many samples yielded to correct average value. In this case, $\langle w \rangle = \langle \exp(-\beta U) \rangle \neq w(\langle U \rangle) = \exp(-\beta \langle U \rangle)$. However, in some simulation problems, the weight function w itself is subject to random noise with zero mean. In such cases, other sampling schemes exist (see [563]).

13.3 Rejection-free Monte Carlo

The conventional wisdom is that if something sounds too good to be true, it probably is not true. However, the rejection-free Monte Carlo schemes that we discuss below are both real and powerful, be it that they usually involve some computational overhead.

13.3.1 Hybrid Monte Carlo

To illustrate that there is nothing strange about rejection-free MC, we first consider the case of Hybrid MC. In the context of molecular simulations, the Hybrid MC method is, as the name suggests, a hybrid between MC and MD [564]. That is: the trial moves are not random displacements of particles, but rather a collective displacement of particles generated by carrying out an MD simulation of length t . The underlying thought is that a “good” (time-reversible, symplectic) MD algorithm (e.g., Verlet) generates a perfectly valid MC trial move because 1) it is reversible and 2) it conserves volume in phase space, meaning that a volume element $d\Gamma \equiv d\mathbf{r}^N d\mathbf{p}^N$ is not changed during the time evolution of the system (see Chapter 4 and ref. [117]).

We can now start a trial MD run at point $\mathbf{r}^N(0)$. To define a trajectory, we must generate a set of initial momenta $\mathbf{p}^N(0)$ according to some (as yet unspecified) probability distribution $P(\mathbf{p}^N(0))$. Having specified $\Gamma(0)$, we run the MD algorithm for n steps (the choice of n can be made later). After n steps, the system will be in a state characterized by phase-space coordinates $\mathbf{r}^N(t)$, $\mathbf{p}^N(t)$.

If we wish to use MD as an MC trial move, it is essential that we impose microscopic reversibility, meaning that the reverse move must be possible. That implies that we must ensure that $P(\mathbf{p}^N(t)) \neq 0$. In addition, as the reverse move involves running the trajectory backward after flipping all momenta at time t , we impose that $P(\mathbf{p}^N)$ is invariant under the reversal of all momenta. In thermal equilibrium, the probability of finding the system with coordinates \mathbf{r}^N must be proportional to $e^{-\beta U(\mathbf{r}^N)}$. Denoting the configuration of the system at the initial (final) time ($t = 0$) by $\{\mathbf{i}\} \equiv \mathbf{r}^N(0)$, and $\{\mathbf{f}\} \equiv \mathbf{r}^N(t)$, the condition for detailed balance (Eq. (3.2.7)) reads

$$e^{-\beta U(\{\mathbf{i}\})} P(\mathbf{p}^N(0)) \text{acc}(i \rightarrow f) = e^{-\beta U(\{\mathbf{f}\})} P(-\mathbf{p}^N(t)) \text{acc}(f \rightarrow i). \quad (13.3.1)$$

From this, it follows that

$$\frac{\text{acc}(i \rightarrow f)}{\text{acc}(f \rightarrow i)} = e^{-\beta[U(\{\mathbf{f}\}) - U(\{\mathbf{i}\})]} \frac{P(\mathbf{p}^N(t))}{P(\mathbf{p}^N(0))}. \quad (13.3.2)$$

First, consider the case that the MD trial move is perfectly energy-conserving. If we denote the total energy $U + K$ by E , then the detailed balance condition implies

$$\frac{\text{acc}(i \rightarrow f)}{\text{acc}(f \rightarrow i)} = \frac{e^{+\beta K(\{\mathbf{f}\})} P(\mathbf{p}^N(t))}{e^{+\beta K(\{\mathbf{i}\})} P(\mathbf{p}^N(0))} \quad (13.3.3)$$

Clearly, if we would have chosen $P(\mathbf{p}^N)$ to be the Maxwell-Boltzmann distribution at inverse temperature β ($P(\mathbf{p}^N) \sim e^{-\beta K(\mathbf{p}^N)}$), then the acceptance ratio for forward and backward trial moves would be 1, which means that we can accept 100% of the trial moves. In reality, energy is not perfectly conserved in an MD simulation. If $E(t) - E(0) = \delta$, then the acceptance ratio of our trial moves would be equal to $\exp(-\beta\delta)$. Hence, in practice, we do not reach 100% acceptance of trial moves, but we can get close.

Of course, we could choose another form for $P(\mathbf{p}^N)$, but typically such a choice would only result in a lower acceptance of hybrid-MC trial moves. For more details on the Hybrid MC method and related techniques, see refs. [21, 565–567].

13.3.2 Kinetic Monte Carlo

In Chapter 3 we presented the Markov-Chain Monte Carlo method as a technique to compute the equilibrium properties of many-body systems. However, the original Monte Carlo method of Ulam (and Metropolis) found its first applications in the simulation of sequences of events (e.g., the interaction of neutrons with condensed matter). The philosophy behind the **Kinetic Monte Carlo** method builds on the original Ulam paper: an efficient, and rejection-free tool to model sequences of stochastic events. The **KMC** method was invented

many times over in various flavors, for instance, in the 1960s in the context of electron transport in semiconductors (see the 1983 review by Jacoboni and Reggiani [568]), in the context of Chemical Kinetics [569] (the so-called Gillespie algorithm) and in the formulation by Bortz, Kalos, and Lebowitz [570] as a method to perform Boltzmann sampling of systems with discrete degrees of freedom.

Rather than discussing all these flavors separately, we give a simplified presentation of the essentials. The key ingredient in KMC is that we consider a situation where different stochastic events labeled with an index α may happen with different rates r_α . What is essential is that we should be able to evaluate all r_α *a priori*, if we know the current state of the system. Let us denote the total rate that *something* happens by $R = \sum_\alpha r_\alpha$. Then the probability that the system evolves for a time t without any of these events taking place is $P_0(t) = \exp(-Rt)$. The probability density for the first event happening at time t is then

$$p_1(t) = R \exp(-Rt). \quad (13.3.4)$$

Our aim is to generate the time t_1 , the time of the next event, according to the distribution given by Eq. (13.3.4). This we can do by expressing t_1 as

$$t_1 = -(1/R) \ln x, \quad (13.3.5)$$

where x is a random number, uniformly distributed between zero and 1. Once we have generated t_1 , we still have to decide *which* event takes place. But, again, this is straightforward because the probability of observing event α is equal to r_α/R , and we know all r_α . If the number N_r of distinct processes is large, selecting one particular α_r has to be done efficiently, for instance, using Walker's Alias method [571]; a nice, non-mathematical explanation can be found in [572].

The next step is to carry out event α (e.g., a spin flip, or a scattering event) and to recompute all those r_α that may have changed. We then compute the time to the next event, a repeat the same steps. In the context of Boltzmann sampling, the events are (for instance) spin flips for spins with different arrangements of neighbors. To obtain the correct equilibrium sampling, we should choose the r_α proportional to the acceptance probability of a standard (e.g., Metropolis) MC trial move. For spin systems with a small number of states of the neighboring spins, we can easily classify each spin according to its environment.

When computing averages, the time intervals between successive events matter. For instance, if we denote the energy of the system during interval i , by E_i , then the energy estimate for a run of total length $\tau = t_1 + t_2 + \dots + t_n$ is

$$\overline{E} = \tau^{-1} \sum_i E_i t_i. \quad (13.3.6)$$

Note that, as this average depends on a ratio of times, the unit of time is unimportant.

As the algorithm of ref. [570] requires the (re)evaluation of all r_α at every event, the computational overhead is high. On the whole, the method is most useful to study the kinetics of systems where the time evolution is dominated by rare, unfavorable spin flips. In standard MC, most high-energy trial moves would be rejected. In contrast, with the method of ref. [570], all moves are accepted, but the system simply spends much less time in “high energy” than in “low-energy” configurations (see e.g., ref. [573]).

The KMC algorithm proposed by Gillespie [569] is widely used for modeling stochastic (bio)chemical reactions or similar processes. However, its applications fall mostly outside the scope of this book.

13.3.3 Sampling rejected moves

In standard Markov-Chain MC, a large fraction of all trial moves is rejected. This seems wasteful, because before rejecting a trial move, we have probed its final state. Yet, if we reject the move, we discard all the information that we have acquired, even if the final state had a fair chance of being accepted. Surprisingly, it is possible to perform MC sampling in such a way that even the information contained in rejected trial configurations is taken into account when computing thermal averages. Sampling of rejected moves was introduced by Ceperley and Kalos [574] in the context of quantum Monte-Carlo simulations and rediscovered in the context of classical MC simulations [575,576].

To derive the method to include the sampling of rejected states, we once again consider the probability of moves from state m to a trial state n (see Chapter 3). The Boltzmann weight of state m is denoted by $\mathcal{N}(m)$. The probability that a system will undergo a transition from state m to state n is $\pi(m \rightarrow n)$, which we shorten to π_{mn} . A valid choice for π_{mn} should maintain the equilibrium distribution [87]:

$$\sum_m \mathcal{N}(m) \pi_{mn} = \mathcal{N}(n). \quad (13.3.7)$$

Typically, we aim to compute averages of the form:

$$\langle A \rangle = \frac{\sum_n \mathcal{N}(n) a_n}{\sum_n \mathcal{N}(n)}. \quad (13.3.8)$$

We can now use Eq. (13.3.7) to write Eq. (13.3.8) as

$$\langle A \rangle = \frac{\sum_n \sum_m \mathcal{N}(m) \pi_{mn} a_n}{\sum_n \sum_m \mathcal{N}(m) \pi_{mn}} = \frac{\sum_m \mathcal{N}(m) \sum_n \pi_{mn} a_n}{\sum_m \mathcal{N}(m)}, \quad (13.3.9)$$

where, in the final equality, we have changed the order of summation and used the fact that $\sum_n \pi_{mn} = 1$. Hence, $\langle A \rangle$ can be rewritten as a Boltzmann average over all states m of the quantity $\sum_n \pi_{mn} a_n$. To estimate $\langle A \rangle$ we can use normal

Markov-chain sampling over the M visited states

$$\langle A \rangle = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \left[\sum_m \pi_{nm} a_m \right], \quad (13.3.10)$$

where the sum now runs over the visited states n . We now use the fact that in a Monte Carlo algorithm the transition probability π_{nm} is the product of two factors: the *a priori* probability α_{nm} to attempt a trial move to m , given that the system is initially in state n and the probability $\text{acc}(n \rightarrow m)$ to accept this trial move.

$$\pi_{nm} = \alpha_{nm} \times \text{acc}(n \rightarrow m). \quad (13.3.11)$$

With this definition, our estimate for $\langle A \rangle$ becomes

$$\langle A \rangle = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \left[\sum_m \alpha_{nm} \times \text{acc}(n \rightarrow m) a_m \right]. \quad (13.3.12)$$

This expression is still not very useful, because it would require us to compute a_m for *all* states m that can be reached in a single trial move from n . In what follows, we use the following shorthand notation:

$$\langle a \rangle_n \equiv \sum_m \alpha_{nm} \text{acc}(n \rightarrow m) a_m. \quad (13.3.13)$$

In simulations, we do not compute $\langle a \rangle_n$ explicitly, but use the fact that α_{nm} is a normalized probability. We can therefore estimate $\langle a \rangle_n$ by drawing, with a probability α_{nm} , trial moves from the total set of all moves starting at n :

$$\langle a \rangle_n = \langle \text{acc}(n \rightarrow m) a_m \rangle_{\alpha_{nm}}. \quad (13.3.14)$$

Our scheme to estimate $\langle A \rangle$ then reduces to the following sampling

$$\langle A \rangle = \lim_{M \rightarrow \infty} \frac{1}{M} \sum_{n=1}^M \left[\sum_{m'} \text{acc}(n \rightarrow m') a'_m \right] \quad (13.3.15)$$

where m' is the set of states generated with the probability α_{nm} . If we use an MC algorithm for which a trial move generates only a single candidate state m , then the expression for $\langle A \rangle$ becomes

$$\langle A \rangle = \frac{\sum_n [(1 - \text{acc}(n \rightarrow m')) a_n + \text{acc}(n \rightarrow m') a'_m]}{\sum_n}. \quad (13.3.16)$$

This average combines information about both the “accepted” and the rejected state of a trial move. Note that the Monte Carlo algorithm used to generate the

random walk among the states n need not be the same as the one corresponding to π_{nm} . For instance, we could use standard Metropolis to generate the random walk, but use the symmetric rule [577]

$$\pi_{mn} = \alpha_{mn} \frac{\mathcal{N}(n)}{\mathcal{N}(n) + \mathcal{N}(m)} \quad (13.3.17)$$

to sample the a_m 's, in which case it can be shown [578,579] that the statistical error is necessarily lower than when using standard Metropolis sampling. The advantage of sampling rejected moves becomes larger when trial moves can cheaply generate many possible final states [580].

13.4 Enhanced sampling by mapping

In section 6.3, we discussed volume-changing moves in the constant- NPT MC method. In these moves, we kept the scaled coordinates of all center-of-mass coordinates fixed, and hence the *real* coordinates of all particles would change when the volume was changed. This is not the only way we could carry out constant pressure MC. We could, for instance, have a system terminated by hard walls and change the volume by an amount ΔV , by carrying out trial displacements of a wall acting as a piston. In such moves, we could leave all real coordinates unchanged, but of course, if the piston would overlap with a particle, then the trial move would be rejected. Such rejections would occur, even if we would simulate an ideal gas at density ρ . The probability that a trial move yields a non-overlapping configuration, would decay as $\min[1, \exp(+\rho\Delta V)]$ (expansion never yields an overlap with the walls). But if no such overlap would be detected, then the acceptance probability would simply be $\min[1, \exp(-\beta P\Delta V)]$, which accounts for the free energy change of the reservoir. If the ideal gas pressure is equal to the reservoir pressure, then the acceptance of expansion and compression trial moves is symmetric in ΔV , and the average volume change after a move is zero, as it should be. In contrast, if, as in section 6.3, we use scaled coordinates, then all trial moves would be guaranteed to generate configurations without overlap with the walls. But then there is a factor $[(V + \Delta V)/V]^N$ in the acceptance probability (Eq. (6.3.12)). This factor is the ratio of the Jacobians of the transformation from unscaled to scaled coordinates.¹ It is a purely entropic term.

We show this example, because it shows that a) trial moves can sometimes be made more efficient by using a coordinate scaling and b) that if we use trial moves that change the mapping from a set of reference coordinates (s^N) to the real coordinates, then the ratio of the old and new Jacobians ends up in the acceptance rule.

Jarzynski [581] explored the more general problem of how a suitably chosen coordinate transformation can improve sampling, in the context of estimating

¹ As before, we write “Jacobian” as a shorthand for “the absolute value of the determinant of the Jacobian matrix.”

free-energy differences. To explain the gist of the method, consider that we have a well-characterized and easy-to-sample N -particle reference system (A) with dN coordinates denoted by q , a potential energy $\mathcal{U}_A(q)$, and a Boltzmann distribution $\rho_A(q)$, and we wish to sample a system B with a harder-to-sample potential energy function $\mathcal{U}_B(q')$ and Boltzmann distribution $\rho_B(q')$. The holy grail would be to construct a coordinate transformation \mathcal{T} from q to q' , such that our sampling of A automatically yields all points in B with the correct Boltzmann weight. The probability density generated by sampling A and transforming from q to q' is

$$\rho(q') \sim \frac{\exp(-\beta\mathcal{U}_A)}{|J|_T} \quad (13.4.1)$$

where $|J|_T \equiv \left| \frac{\partial q'}{\partial q} \right|$ is the Jacobian of the transformation from q to q' . Clearly, if we could construct a transformation \mathcal{T} such that $|J|_T = \rho_A(q)/\rho_B(q')$, then sampling A would immediately allow us to sample B with the correct Boltzmann weight. In general, constructing such a perfect transformation is not feasible, and hence much of the effort focuses on generating reasonably good transformations. (see however, section 13.4.1 below). But for illustrative purposes, it helps to consider a simple case where we can do the exact transformation.

Consider a single particle in a 1d box of length $L = 1$: this is our system A. If the potential energy in the box is flat, uniform random sampling of q between 0 and 1 yields the Boltzmann distribution of A. Next consider system B: a particle that can take all coordinates $0 < q' < \infty$, and that is subject to an external potential $\mathcal{U}_B(q') = \kappa q'$, with a corresponding Boltzmann distribution $\rho_B(q') \sim \exp(-\beta\kappa q')$. To achieve the correct sampling of B via a coordinate transformation we need

$$\frac{\partial q'}{\partial q} = C \exp(\beta\kappa q') \quad (13.4.2)$$

where C is a constant that will be fixed by the condition $\langle q' \rangle = (\beta\kappa)^{-1}$. It is easy to verify that Eq. (13.4.2) implies that [66]

$$q' = (\beta\kappa)^{-1} \ln(-q).$$

In this case, Metropolis sampling of the exponential distribution of system B would be slower than sampling the uniform distribution of system A. Jarzynski [581] showed that, for the general case where A and B can be complex many body systems, the free-energy difference ΔF between A and B is given by

$$\Delta F = -k_B T \ln \left\langle e^{-\beta(\mathcal{U}_B(q') - \mathcal{U}_A(q) - \ln J(q))} \right\rangle,$$

and hence a transformation with $\ln J(q) \approx \mathcal{U}_B(q') - \mathcal{U}_A(q)$ would speed up the estimation of free-energy differences, provided that A is easy to sample (no barriers, no traps).

The mapping method was reformulated by Kofke and collaborators (see e.g., [582]), for a wide range of sampling problems. But also in the Kofke approach, it is necessary to make physically plausible approximations to construct the mapping.

13.4.1 Machine learning and the rebirth of static Monte Carlo sampling

More recently, a number of groups (see Noé et al. [583], Wirnsberger et al. [584] and Gabrié et al. [585]), showed that there is considerable promise in using machine learning to create ‘good’ invertible mappings between q and q' defined in section 13.4, in situations where our intuition may not be good enough.

As explained in section 3.2.1, an ideal static MC code would generate independent points in configurations space with a probability proportional to their Boltzmann weights. A system of N particles in d dimensions has dN degrees of freedom; we can view the generation of a configuration as a complex, nonlinear mapping of dN random numbers $\mathcal{R}^{dN} = (\mathcal{R}_1, \dots, \mathcal{R}_{dN})$ to dN coordinates $\mathbf{r}^N = (r_1, \dots, r_{dN})$. A trivial example would be the generation of an ideal gas configuration in a cubic box with diameter L : a typical configuration can be generated by multiplying dN random scaled coordinates \mathbf{s}^N inside a unit hypercube, i.e., $0 \leq s_i < 1$, with the box-length L .

For interacting systems, the above mapping would be useless: almost all configurations would have a vanishing Boltzmann weight. The question is whether we can define some complex, nonlinear vector function \mathcal{F} , with $\mathbf{r}^N = \mathcal{F}(\mathcal{R}_1, \dots, \mathcal{R}_{dN})$ such that the resulting configurations appear with a frequency proportional to their Boltzmann weights. The answer to this question used to be “no”, but that situation is changing.

Before proceeding, we note that the mapping from \mathcal{R}^{dN} to \mathbf{r}^N should be bijective, that is: every set of dN random numbers should correspond to one and only one valid point in configuration space, and conversely. In section 13.4 we described MCMC trial moves that use the idea of mapping to generate trial configurations with a higher Boltzmann weight than those generated by random trial moves. Typically, such mappings are constructed “by hand”, using physical insight. But our physical insight is insufficient to design reasonable mappings that generate, in a static MC move, configurations with a probability close to their Boltzmann weight. This is where Machine Learning (ML) comes in: the idea is that we can train a neural net to perform such bijective transformations. Note that most machine-learned “mappings” are not at all bijective: typically, many inputs generate the same output, e.g., many pictures of the same person should identify the same individual. A popular class of methods for training a neural net to carry out a complex, bijective transformation is called “normalizing flows” [586].

We will not go into the ML aspect of the method. Just a remark on the terminology: the term “flow” is used because in practice the mapping is im-

plemented as a series of transformations. The original, normalized probability distribution of the random numbers (more commonly normal, rather than uniform), is mapped onto a (normalizable) probability distribution in the space of transformed coordinates.

To explain what happens next, it is convenient to assume that we map random numbers that are uniform in $0 \leq \mathcal{R} < 1$, to the coordinates of the system. The original, simple distribution is often referred to as the “latent space” distribution. In practical cases, the latent-space distribution is not uniform but normal. We denote the generated probability distribution in r^{dN} by $\mathcal{P}(r^{dN})$. Then

$$\mathcal{P}(\mathcal{R}^{dN}) d\mathcal{R}^{dN} = \mathcal{P}(r^{dN}) dr^{dN} = \mathcal{P}(r^{dN}) J(\mathcal{R}^{dN}, r^{dN}) d\mathcal{R}^{dN}, \quad (13.4.3)$$

where $\mathcal{P}(\mathcal{R}^{dN}) = 1$, and $J(\mathcal{R}^{dN}, r^{dN})$ denotes the Jacobian for the transformation from \mathcal{R}^{dN} to r^{dN} . We aim to find a mapping such that $\mathcal{P}(r^{dN})$ is proportional to the Boltzmann distribution $\exp[-\beta U(r^{dN})]$

$$\exp[-\beta U(r^{dN})] J(\mathcal{R}^{dN}, r^{dN}) = \text{constant}. \quad (13.4.4)$$

Without loss of generality, we take the constant to be equal to one. Then

$$\beta U(r^{dN}) = \ln J(\mathcal{R}^{dN}, r^{dN}). \quad (13.4.5)$$

In words: the trained transformation should map more densely onto regions with low potential energy (small Jacobian \leftrightarrow large Boltzmann factor) and less densely elsewhere. The [ML](#) aspect is then to train a neural net to find a transformation that approximately satisfies Eq. (13.4.5): we can then correct for the remaining discrepancy by re-weighting the sampled points with a factor $\exp[-\beta U(r^{dN})]/J(\mathcal{R}^{dN}, r^{dN})$.² If the training has been successful, the re-weighting factor should be close to one.

To train the neural network, we need to be able to generate a number of configurations of the system with a reasonable Boltzmann weight. These configurations may be generated in advance [583], or on the fly [585]. Also, the training works both ways: we train the neural net such that probable states of the latent distribution map onto states with a high Boltzmann weight. But conversely, we also train the network such that the inverse mapping from real space to the latent distribution, maps states with a high Boltzmann weight onto “likely” reference states. In practice, it is often convenient to train the network such that it minimizes the average of the backward and forward Kullback-Leibler divergences.

² Training of a neural network requires finding an extremum, rather than an equality. In this case, the quantity to optimize is the (forward or backward) Kullback-Leibler divergence [587] between the generated distribution and the target distribution. This divergence is at a minimum if the two distributions are equal. Normalization of the distributions is unimportant because the log of the normalizing factor is a constant.

There are two points to note. First, computing Jacobians by hand, in particular for a sequence of complex transformations, is no fun. However, this problem has been made much less daunting by the use of automatic differentiation (see [108]). Second: a Jacobian is the absolute value of a determinant. In general computing, an $M \times M$ determinant requires $\mathcal{O}(M^3)$ operations, which quickly becomes prohibitive. However, in some cases, such as when the matrix is triangular, computing the determinant is an $\mathcal{O}(M)$ operation. The following trick (called “coupling layers/flows”) can be used to create a mapping for which the Jacobian is guaranteed to be triangular: at every step in the sequence of transformations, part of the M coordinates, i.e., those with index $1 \leq i \leq k$ (k can be chosen) are just mapped onto themselves and are therefore independent of all other coordinates. As a result, the top k rows of the Jacobian matrix are only non-zero on the diagonal. The remaining coordinates ($k + 1 \leq i \leq M$), are transformed such that every new coordinate with index i is a function of the old coordinate with the same index, and of *all* the coordinates in the first set ($1 \leq i \leq k$). The function that carries out the transformation can be linear or non-linear and should be different for every coordinate; importantly, it depends on a set of parameters that can be “learned”. The net effect of the coupled-flows trick is that the resulting matrix is lower diagonal and computing it is an $\mathcal{O}(M)$ operation. The splitting of the coordinates in sets that will/will not be transformed, should be different at every step in the sequence of transformations. The total Jacobian is then simply the product of the partial Jacobians, or in practice: the log of the total Jacobian is the sum of the logs of the partial Jacobians.

Illustration 21 (Normalizing flows and static Monte Carlo sampling). Noé et al. [583] illustrated the potential of normalizing flows in the context of Molecular Simulations. Here we focus on two features of the normalizing-flows method that are highlighted in ref. [583]: 1. the ability to discover states with a high Boltzmann weight that are different from the training set, and 2. the ability to compute free-energy differences between two or more different structures of a given model, without the need to construct a transition between the two.

To illustrate the “discovery” of new structures, ref. [583] considered a simple example of a system with a low-dimensional Mueller-Brown potential-energy landscape [588] that contains two low-energy basins separated by a high-energy barrier.

The first stage of the simulations is to start by sampling a small number of states in one of the potential energy basins, and training the network such that states with large Boltzmann weights correspond to points close to the center of normal distribution in the latent space, and conversely that points that are generated with high probability in the latent space correspond to configurations with high Boltzmann weights in the real energy landscape.

As the simulation is trained on one basin only, the simulation will initially simply find more configurations in that basin. However, after a number of it-

erations, the network starts to find states in the other basin, which was not a part of the initial training set. After a sufficiently large number of iterations, the simulation recovers the complete Boltzmann distribution of this energy landscape (see Fig. 13.3). Note, however, that this is a simple, low-dimensional example. Gabrié et al. [585] argue that the discovery of “unknown” basins does not work in real-life problems with high-dimensional potential energy landscapes.

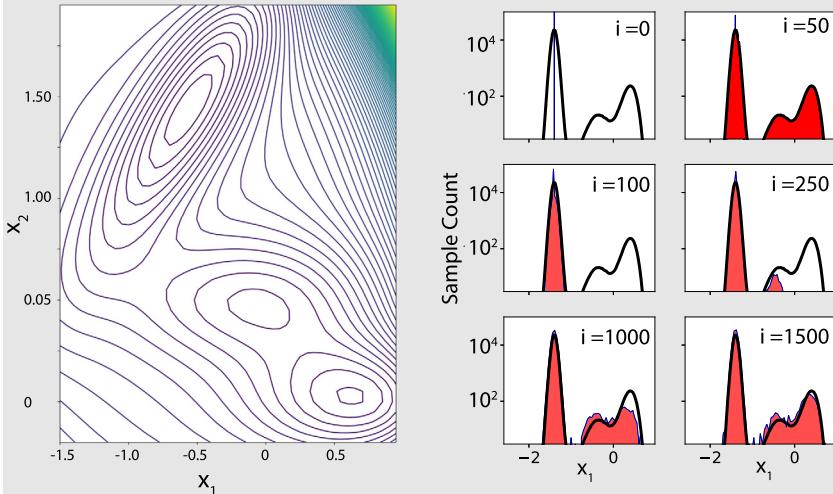


FIGURE 13.3 Example of the “discovery” of a basin in an energy landscape that was not included in the original training set of the mapping [583]. The potential energy “landscape” is given by a version of the Mueller potential (left figure: see [583]). In the legend, the number of iterations is denoted by I . Initially (right figure: top left) only the original (single) state in one basin is found. The original basin is explored during the next $\mathcal{O}(100)$ iterations. However, after that, also the other basins are found. After some 1500 iterations, the complete Boltzmann distribution of the original system is recovered. Note that this is achieved without having to cross the high energy barrier between the two basins along the x_2 direction.

Next, we consider the normalizing-flows approach to computing free-energy differences. The starting point is the following relation [581], expressing the free-energy difference $\Delta F \equiv F_B - F_A$ of two systems A and B related via a bijective mapping, as:

$$e^{-\beta \Delta F} = \frac{Z_B}{Z_A} = \left\langle e^{-\beta [\mathcal{U}_B - \mathcal{U}_A - k_B T \ln J]} \right\rangle_A , \quad (13.4.6)$$

where J is the Jacobian of the mapping from A to B . In our case, system A would be described by a dN dimensional normal distribution, and system B would be the system of interest. If we have a good mapping, we can obtain good estimates of the average in Eq. (13.4.6), and thereby obtain the free energy of system A . But this result is not limited to a single target system B : we could have several target systems (e.g., different crystal structures of

the same model [584]). In that case, Eq. (13.4.6) allows us to compute the free-energy difference between different structures. Note, however, that the free-energy calculations in refs. [583,589] do not use Eq. (13.4.6) itself, but variants based on the more sophisticated methods described in Chapter 8.

But we can also bias the mapping in such a way that we explore the (Landau) free energy of system *B* as a function of a given order parameter. Noe et al. [583] showed that the latter procedure yields free-energy profiles that are in good agreement with results obtained with conventional (e.g., umbrella sampling) methods (see Fig. 13.4).

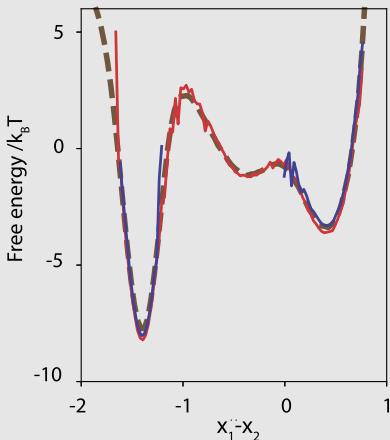


FIGURE 13.4 Free-energy profile along the diagonal $x_1 - x_2$ direction in the potential energy landscape of the Mueller potential (see Fig. 13.3). The figure compares the reference free-energy profile obtained using a normal (biased) simulations with the results obtained using an unbiased normalizing-flows, and with normalizing flows plus biasing. Note that, with the biasing, it is possible to recover the free-energy profile in the range that would be poorly sampled in the original normalizing-flows calculation. Figure based on the data of ref. [583].

Note, however, that the normalizing flow approach to Monte Carlo sampling is very much “work in progress”. For higher-dimensional problems (e.g., liquids), the mapping starting from a Gaussian distribution [583] tends to run into trouble: approaches that create mappings from a reference state that is more similar to the target state work better [584,585,590].

13.4.2 Cluster moves

The most spectacular examples of Monte Carlo schemes that reach 100% acceptance are those based on cluster moves. In this area, there have been important developments since the previous edition of this book. Below, we describe some of the key concepts behind these developments, without pretending to be anywhere near complete.

For didactical reasons, it is best to start with the earliest example of a cluster-move scheme that reaches 100% acceptance, namely the so-called [Swendsen-Wang](#) method [591].

13.4.2.1 Cluster moves on lattices

The central idea behind the [Swendsen-Wang](#) ([SW](#)) scheme and subsequent extensions and modifications is to generate trial configurations with a probability that is proportional to the Boltzmann weight of that configuration. As a result, the subsequent trial moves can be accepted with 100% probability. We use a somewhat simplified derivation of the [SW](#) for cluster moves based, again, on the condition for detailed balance. Consider an “old” configuration (labeled by a superscript o) and a “new” configuration (denoted by a superscript n). Detailed balance is satisfied if the following equality holds:

$$\begin{aligned} \mathcal{N}(o) P_{\text{Gen}}^o(\{\text{cluster}\}) P^{\{\text{cl}\}}(o \rightarrow n) \text{acc}(o \rightarrow n) \\ = \mathcal{N}(n) P_{\text{Gen}}^n(\{\text{cluster}\}) P^{\{\text{cl}\}}(n \rightarrow o) \text{acc}(n \rightarrow o), \end{aligned} \quad (13.4.7)$$

where $\mathcal{N}(o)$ is the Boltzmann weight of the old configuration, $P_{\text{Gen}}^o(\{\text{cluster}\})$ denotes the probability of generating a specific cluster, starting from the old configuration of the system. The term $P^{\{\text{cl}\}}(o \rightarrow n)$ is the probability of transforming the generated cluster from the old to the new situation. Finally, $\text{acc}(o \rightarrow n)$ is the acceptance probability of a given trial move. We can simplify Eq. (13.4.7) in two ways. First of all, we require that the *a priori* probability $P^{\{\text{cl}\}}(o \rightarrow n)$ be the same for the forward and reverse moves. Moreover, we wish to impose $P_{\text{acc}} = 1$ for both forward and reverse moves. This may not always be feasible. However, for the simple case that we discuss next, this is indeed possible. The detailed balance equation then becomes

$$\mathcal{N}(o) P_{\text{Gen}}^o(\{\text{cluster}\}) = \mathcal{N}(n) P_{\text{Gen}}^n(\{\text{cluster}\}) \quad (13.4.8)$$

or

$$\frac{P_{\text{Gen}}^o(\{\text{cluster}\})}{P_{\text{Gen}}^n(\{\text{cluster}\})} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp(-\beta \Delta \mathcal{U}), \quad (13.4.9)$$

where $\Delta \mathcal{U}$ is the difference in energy between the new and the old configurations. The challenge is to find a recipe for cluster generation that will satisfy Eq. (13.4.9). To illustrate how this works, we consider the Ising model. The extension to many other lattice models is straightforward.

Swendsen-Wang algorithm for Ising model

Consider a given configuration of the spin system (the dimensionality is unimportant), with N_p spin pairs parallel and N_a spin pairs anti-parallel. The total energy of that configuration is

$$\mathcal{U} = (N_a - N_p) J, \quad (13.4.10)$$

where J denotes the strength of the nearest-neighbor interaction. The Boltzmann weight of that configuration is

$$\mathcal{N}(o) = \exp[-\beta J(N_a - N_p)]/Z, \quad (13.4.11)$$

where Z is the partition function of the system. In general, Z is unknown, but that is unimportant. The only thing that matters is that Z is constant. Next, we construct clusters by creating bonds between spin pairs according to the following recipe:

- If nearest neighbors are *antiparallel*, they are not connected.
- If nearest neighbors are *parallel*, they are connected with probability p and disconnected with probability $(1 - p)$.

Here, it is assumed that J is positive. If J is negative (antiferromagnetic interaction), parallel spins are not connected, while antiparallel spins are connected with a probability p .

In the case that we consider, there are N_p parallel spin pairs. The probability that n_c of these are connected and $n_b = N_p - n_c$ are “broken” is

$$P_{\text{Gen}}^o(\{\text{cluster}\}) = p^{n_c} (1 - p)^{n_b}. \quad (13.4.12)$$

Note that this is the probability to connect (or break) a *specified* subset of all links between parallel spins. Once the connected bonds have been selected, we can define the clusters in the system. A cluster is a set of spins that is at least singly connected by bonds. Let us denote the number of such clusters by M . We now flip every one of the M clusters with 50% probability. After the cluster flipping, the number of parallel and antiparallel spin pairs will have changed, for example,

$$N_p(n) = N_p(o) + \Delta \quad (13.4.13)$$

and (hence)

$$N_a(n) = N_a(o) - \Delta. \quad (13.4.14)$$

Therefore, the total energy of the system will have changed by an amount $-2J\Delta$:

$$\mathcal{U}(n) = \mathcal{U}(o) - 2J\Delta. \quad (13.4.15)$$

Let us now consider the probability of making the *reverse* move. To do this, we should generate the *same* cluster structure, but now starting from a situation where there are $N_p + \Delta$ parallel spin pairs and $N_a - \Delta$ antiparallel pairs. As before, the bonds between antiparallel pairs are assumed to be broken (this is compatible with the same cluster structure). We also know that the new number of connected bonds, n'_c , must be equal to n_c , because the same number of connected bonds is required to generate the same cluster structure. The difference

appears when we consider how many of the bonds between parallel spins in the new configuration should be broken (n'_b). Using

$$N_p(n) = n'_c + n'_b = n_c + n'_b = N_p(o) + \Delta = n_c + n_b + \Delta, \quad (13.4.16)$$

we see that

$$n'_b = n_b + \Delta. \quad (13.4.17)$$

If we insert this in Eq. (13.4.9), we obtain

$$\begin{aligned} \frac{P_{\text{Gen}}^o(\{\text{cluster}\})}{P_{\text{Gen}}^n(\{\text{cluster}\})} &= \frac{p^{n_c}(1-p)^{n_b}}{p^{n_c}(1-p)^{n_b+\Delta}} = (1-p)^{-\Delta} \\ &= \frac{\mathcal{N}(n)}{\mathcal{N}(o)} = \exp(2\beta J \Delta). \end{aligned} \quad (13.4.18)$$

To satisfy this equation, we must have

$$p = 1 - \exp(-2\beta J), \quad (13.4.19)$$

which is the Swendsen-Wang rule.

Wolff algorithm

Most of the computational overhead of the Swendsen-Wang algorithm goes into decomposing the system into M clusters that can be flipped independently. In a sense, this is overkill, because the algorithm prepares M clusters that can be flipped in 2^M different ways, yet we only select one of these possible choices as our trial move. A more efficient approach to the cluster move problem was proposed by Wolff [592]: rather than constructing all clusters that could be flipped, the Wolff algorithm selects one spin at random, and then uses the above rules for defining bonds to construct a single cluster connected to the selected spin. There are bonds connecting all spins inside the cluster, yet none of the spins in the cluster is connected to any of the other spins. This single cluster is then flipped (with a 100% acceptance probability). Wolff clusters may be small, in particular, well above the critical point, in which case the algorithm will not speed up the decay of fluctuation much compared with single-spin moves, but this is not serious as constructing small clusters is cheap. As the system is cooled towards the critical temperature, Wolff clusters will typically grow —and hence more expensive to generate, but flipping them then contributes significantly to the decay of fluctuations.

13.4.2.2 Off-lattice cluster moves

An obvious question is if it is possible to formulate algorithms that generate rejection-free cluster moves of the Swendsen-Wang/Wolff type for off-lattice systems. As we explain below, the answer is: *Yes, but....*

Let us start with the *Yes*. Dress and Krauth [593] proposed a rejection-free algorithm to carry out large-scale cluster moves for hard-core systems. Such a move is constructed by considering the effect on the particle coordinates of a symmetry operation that is compatible with the periodic boundary conditions. We will not describe the original ([Swendsen-Wang](#)-like) approach of ref. [593], but the Wolff-like approach of ref. [594].

The approach of ref. [593] does not depend on dimensionality, and therefore we illustrate the method by considering a one-dimensional example: a fluid of N hard particles in a periodically repeated segment of length L . We now consider a permissible symmetry operation: in 1d, this would be inversion around a randomly chosen point, or a translation over a randomly chosen distance. However, for Wolff clusters in higher dimensions, we can also consider other point-symmetries, e.g., rotations around an axis (this cannot be done when the cluster move should map the *whole* periodic box onto itself). Whatever the move, the Wolff cluster should be sufficiently small such that, after the move, it cannot interact with any of the periodic images of the original cluster.

We first apply this symmetry operation (say, inversion) to a randomly chosen “seed” particle, say i : the result is that $x_i \rightarrow x'_i$, where x'_i is the position where particle i would end up after the inversion. Of course, there is now a finite chance that particle i at x'_i will overlap with one or more particles j . If that is the case, we apply the same symmetry operation to the coordinate(s) of particle(s) j . As we started from a configuration with no overlaps, particles i and j will not overlap if they have *both* undergone the symmetry operation. However, particle j may now overlap with a particle k , which is then also inverted. The sequence of moves might be described by the rule “*if you cannot avoid them, join them*”.

If the density of the fluid is not too high, there will come a point when no new overlaps are generated, say after we have inverted the coordinate of the n -th particle in our cluster. At that point, we can stop: we have moved a cluster of particles i, j, k, \dots, n without generating any overlaps. Therefore, we can accept this trial move.

The limitation of the method is that for a dense fluid, the cluster will percolate through the periodic boundaries and will contain most or all particles. In that case, the move would not contribute to the relaxation of structural fluctuations. However, the same problem is already present in the [Swendsen-Wang](#)/Wolff algorithm: at low enough temperatures, the cluster that is flipped contains (almost) all spins, in which case: *yes*, it is rejection-free, but *no*, it does not speed up the simulation.

Having formulated a rejection free cluster-algorithm for hard particles, it is logical to ask if we can do the same for particles with continuous interactions.

The answer is, again, yes. But to explain that approach, it is convenient to first discuss a seemingly unrelated method, namely the *early rejection* method.³

³ In earlier editions of this book, the early-rejection method was described in a different context. However, the approach has gained importance in the context of the algorithms described in this chapter.

13.4.3 Early rejection method

In section 3.4.1 we argued that the optimal acceptance probability of MC trial moves in hard-core systems should be lower than of trial moves in systems with continuous interactions. The reason is that in simulations of hard-core systems, we can reject a trial move as soon as a single overlap is detected. For continuous potentials, all interactions must be computed before a standard MC trial move can be accepted or rejected. As a consequence, it is on average cheaper to perform a trial move of a hard-core particle that results in rejection than in acceptance. This leads to the strange situation that it would be cheaper to perform a MC simulation of a hard-core model than of a corresponding model that has very steep but continuous repulsive interactions. Yet one would expect that, also for continuous potentials, it should be possible to reject trial moves that are almost certainly “doomed” if they incur a large potential-energy penalty from a near overlap.

Fortunately, we can use the idea behind the Swendsen-Wang scheme to formulate a criterium that allow us to reject “doomed” trial moves for particles with a continuous intermolecular potential at an early stage [595].

To see how this approach works, consider a trial displacement of particle i to a new position. We now construct “bonds” between this particle and a neighboring particle j with a probability

$$p_{\text{bond}}(i, j) = \max[0, 1 - \exp(-\beta \Delta u_{i,j})], \quad (13.4.20)$$

where $\Delta u_{i,j} = u^n(i, j) - u^o(i, j)$ is change in the interaction energy of particles i and j caused by the trial displacement of particle i . If j is not connected to i , it means that j will not block the move of i , and we proceed with the next neighbor k , and so on. But as soon as a bond is found between i and any of its neighbors, the single-particle move of i is blocked and we reject the trial move. Only if particle i is not bonded to any of its neighbors do we accept the trial move.

It is easy to show that this scheme satisfies detailed balance. The probability that we accept a move from the old to the new position is given by

$$\text{acc}(o \rightarrow n) = \prod_{j \neq i} [1 - p_{\text{bond}}(i, j)] = \exp \left[-\beta \sum_{j \neq i}^+ \Delta u_{i,j}(o \rightarrow n) \right], \quad (13.4.21)$$

where the summation is over particles j for which the $\Delta u(i, j)$ is positive. For the reverse move $n \rightarrow o$, we have

$$\text{acc}(n \rightarrow o) = \prod_{j \neq i} [1 - p_{\text{bond}}(i, j)] = \exp \left[-\beta \sum_{j \neq i}^+ \Delta u_{i,j}(n \rightarrow o) \right]. \quad (13.4.22)$$

The summation is over all particles j for which the reverse move causes an increase in energy. In addition, we can write

$$\Delta u_{i,j}(n \rightarrow o) = -\Delta u_{i,j}(o \rightarrow n), \quad (13.4.23)$$

which gives for the probability of accepting the reverse move $n \rightarrow o$

$$\text{acc}(n \rightarrow o) = \exp \left[+\beta \sum_{j \neq i}^- \Delta u_{i,j}(o \rightarrow n) \right], \quad (13.4.24)$$

where the summation is over all particle j for which the energy $u_{i,j}(o \rightarrow n)$ decreases. Detailed balance now implies that

$$\begin{aligned} \frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} &= \frac{\exp \left[-\beta \sum_{j \neq i}^+ \Delta u_{i,j}(o \rightarrow n) \right]}{\exp \left[+\beta \sum_{j \neq i}^- \Delta u_{i,j}(o \rightarrow n) \right]} \\ &= \exp \left[-\beta \sum_{j \neq i} \Delta u_{i,j}(o \rightarrow n) \right] = \frac{\mathcal{N}(n)}{\mathcal{N}(o)}, \end{aligned} \quad (13.4.25)$$

which demonstrates that detailed balance is indeed obeyed.

It is interesting to compare this scheme with the original Metropolis algorithm. In the bond formation scheme, it is possible that a move is rejected even when the total energy decreases. Hence, although this scheme yields a valid Monte Carlo algorithm, it is *not* equivalent to the Metropolis method. Eq. (13.4.20) ensures that, if a trial displacement puts particle i in a very unfavorable position where $\Delta u_{ij} \gg 0$, then it is very likely that a bond will form between these particles and hence the trial move can be rejected.

The early-rejection scheme is not limited to single particle moves. In fact, it is most useful when applied to complex many-particle moves, such as the cluster moves discussed below, or the [Configurational-Bias Monte Carlo](#) scheme discussed in Chapter 12. In the standard configurational-bias scheme, one has to “grow” an entire chain molecule and calculate its Rosenbluth weight, before a trial move can be accepted or rejected.

However, if one of the first segments during this growth was placed at an unfavorable position, such that the new configuration is “doomed”, then the early-rejection scheme could be used to avoid having to complete the growth of a new polymer configuration.

Early rejection and cluster moves

We can use the early-rejection method described above to recast the method of ref. [593] in a form suitable for systems with continuous interactions (see Liu and Luijten [594]).

Consider a cluster move of the type considered above. We can now again define “bonds”. If particle i moves from x_i to x'_i , its interaction with another particle j at x_j would change by an amount $\Delta u_{i,j} \equiv u_{i',j} - u_{i,j}$, where i' denotes the situation where particle i is located at x'_i . Clearly, if $\beta\Delta u_{i,j} \gg 1$, the move would be unfavorable, unless we also include particle j in the same cluster as i . We can now use Eq. (13.4.20) to describe the probability that particles i and j are “bound” which, in this case, means that they undergo the same cluster move. As before, we now continue adding particles j, k , etc. to the cluster until no more bonds are formed. We can then carry out the cluster move. As we have shown in Eq. (13.4.25), this cluster move satisfies detailed balance and would be rejection-free. The method of ref. [594] works well for particles with short-ranged interactions.

There is, however, a problem if we try to carry out cluster moves for particles with longer-ranged interactions because then we would have to compute $p_{\text{bond}}(i, j)$ for a large number of particle pairs.

In the case of longer-ranged interactions, we could, in principle, attempt hybrid cluster moves: we use the approach described above to construct a cluster of particles interacting with the pair-potential $u(r)$ truncated at some distance r_c . When attempting the cluster move, we then compute ΔU_{LR} , the change in the total potential energy due to all other (presumably weak) long-ranged interactions ($r > r_c$). We then accept or reject our cluster move with the Metropolis rule $\text{acc}(o \rightarrow n) = \min[1, \exp(-\beta\Delta U_{\text{LR}})]$, but of course, then the moves are no longer rejection free.

Virtual move Monte Carlo

The [Virtual-Move Monte Carlo \(VMMC\)](#) method of Whitelam et al. [596,597] is a cluster-move scheme that aims to make the motion of clusters resemble the diffusive motion of clusters in a solvent. The basic moves are cluster translation/rotation, but to make the dynamics more “realistic,” the clusters are given a diffusion constant that scales as their inverse mass, which is not the Stokes-Einstein result for approximately spherical particles, nor for most other cluster shapes. The algorithm is distinctly more complex than the basic cluster algorithm of ref. [594].

13.4.4 Beyond detailed-balance

Thus far, we have considered Monte Carlo algorithms that were constructed by imposing irreducibility and detailed balance (Eq. (3.2.7)). However, as argued in Chapter 3, valid MC algorithms *must* obey balance (Eq. (3.2.6)): detailed-balance is overkill.

Most currently used MC algorithms satisfy detailed balance, rather than balance, because proving that an algorithm satisfies balance is often subtle. However, a new class of MC algorithms has emerged, which do not satisfy detailed balance and, importantly, explore configuration space much more ef-

ficiently than the Metropolis algorithm. Below we discuss the so-called *Event Chain Monte Carlo* (ECMC) simulations, in the form introduced by Krauth and co-authors [89,598–601] and a related approach proposed by Peters and de With [602].

Event-chain Monte Carlo

To explain the concept of balance in the context of the [Event Chain Monte Carlo \(ECMC\)](#) method, we first consider the example of a 1d hard-core system discussed in ref. [603]. We consider N identical hard particles with diameter σ on a line segment of length L , subject to periodic boundary conditions. The coordinates of the particles are denoted by $x_1, x_2, \dots, x_i, \dots, x_N$. Because of the hard-core exclusion, two particles cannot come closer than a distance σ . We also assume that particles cannot cross. Now consider the situation in that we attempt to move a randomly chosen particle by a random distance δx drawn from a probability distribution $P(\delta x)$, with $P(\delta x) = 0$ for $\delta x < 0$. A necessary condition for an algorithm to maintain the Boltzmann distribution is that it satisfies the *balance* condition (Eq. (3.2.6)):

$$\sum_{o'} \mathcal{N}(o') \pi(o' \rightarrow n) = \mathcal{N}(n) \sum_o \pi(n \rightarrow o), \quad (13.4.26)$$

which is a weaker condition than the detailed-balance condition, given by equation (3.2.7). Note that in Eq. (13.4.26), the set of states feeding into a given state n need not be the same as the states that can be reached from state n . In the specific case of hard-core interactions, the Boltzmann weights of all permissible (non-overlapping) configurations are the same, hence in that case balance implies that

$$\sum_o \pi(o \rightarrow n) = 1, \quad (13.4.27)$$

where we have used the fact that the probabilities to carry out any of the permitted moves (including rejected moves) must add to one. In the case of N hard particles, we can for instance select any one of the particles with a probability $1/N$ and attempt to move in the positive x -direction (because we have chosen $\delta x > 0$). Such an algorithm clearly does not satisfy detailed balance because the reverse move ($\delta x < 0$) is excluded. A given trial move, say of particle i from position $x'_i \equiv x_i - \delta x$ to the (permissible) position x_i , will be accepted if x'_i was at least a distance σ to the right of $i - 1$. But if (and only if) this condition is *not* satisfied, then a trial move of the particle at x_{i-1} over a distance δx would be rejected because it would create an overlap of the particle at x_i (we count particles Modulo N). In that case, the move is rejected and all particles stay at the original configuration. However, in Monte Carlo terms, a rejected move is also a move (namely from o to o). Together, the probability of a move *into* the

configuration $x_1, x_2, \dots, x_i \dots, x_N$ is then

$$\frac{1}{N} \sum_i [\text{acc}(x'_i \rightarrow x_i) + (1 - \text{acc}(x_{i-1} \rightarrow x_i))] = 1 \quad (13.4.28)$$

because every term in the sum is equal to $1/N$. But, as explained above, every single term in the sum is equal to $1/N$. For the trial move $(x'_i \rightarrow x_i)$, the *old* configuration is $x_1, x_2, \dots, x_i - \delta x, \dots, x_N$, whereas for the trial move $(x_{i-1} \rightarrow x_i)$, the *old* configuration is equal to the *new* configuration $x_1, x_2, \dots, x_i \dots, x_N$. In the example discussed above, we considered only one size δx for the trial move. However, in order to ensure ergodicity, we need a distribution $p(\delta x)$ of step sizes, with $\lim_{\delta x \rightarrow 0} p(\delta x) \neq 0$.

As discussed in ref. [603], it can be shown in a number of cases that the irreversible MC algorithm samples configuration space more efficiently than the Metropolis algorithm. However, the irreversible MC algorithm can be made even more powerful by comparing it with another trick called “lifting”. To explain lifting, we will introduce it using somewhat less mathematical language than ref. [603].

Up to this point, we have assumed that the particle to be moved in an MC algorithm is selected at random (see section 3.3.1). However, this is not necessary to satisfy the balance condition. We could, for instance, attribute an additional label ω to the particles in the simulation, where ω can have the values 0 and 1. Trial moves will only be attempted for a particle i if $\omega_i = 1$. At any time only one particle has $\omega = 1$. We now add a rule to our MC algorithm: if the trial move of particle i is rejected because the move would lead to an overlap with particle j , then we set $\omega_j = 1$ and $\omega_i = 0$. Hence, in the next move, we will attempt to move particle j . Does this rule satisfy balance? In fact, it follows directly from the discussion above: the irreversible MC algorithm of a one-dimensional hard-core system. Consider that the “new” state is one where particle i at x_i is mobile ($\omega_i = 1$). As before, we simplify things by considering only trial moves of length $+\delta x$. How could the system have ended up in the state $x_i, \omega_i = 1$? There are now only two possibilities: particle i came from a position $x_i - \delta x$ and had $\omega_i = 1$, or a trial move of particle $i - 1$ with $\omega_{i-1} = 1$ was rejected, and the label $\omega = 1$ was moved to i . But we already know that the probability that the first move is accepted plus the probability that the second move is rejected must add up to one. Hence, this simple lifting algorithm satisfies balance.

We can now carry out the next trial move for the currently active particle, and so on. To ensure that the algorithm is ergodic and satisfies global balance, we should randomly select a new active particle after a given number of steps, say M . Of particular importance is the case where the $\delta x \rightarrow 0$ and $M \rightarrow \infty$ such that $M\delta x \equiv \lambda$ finite. In this case, the moves in the **ECMC** algorithm are like a relay race: a particle moves continuously until it hits another particle: it then stops and passes the “baton” (i.e., the label $\omega = 1$) to the particle that stopped it. This process continues until the total distance covered by the successive active

particles equals λ , after which a new active particle is chosen at random. The optimal choice of the distribution of λ -values is discussed in ref. [604].

Intuitively, it is easy to understand why this algorithm may lead to more rapid equilibration than the Metropolis algorithm: in the Metropolis algorithm, density heterogeneities decay by diffusion, which is slow in regions of high density. In contrast, in the **ECMC** algorithm, a particle that hits a high-density region can lead to a sound-like propagation through that region, so that another particle comes off at the other end of the high-density obstacle. Clearly, if we would also allow **ECMC** moves in the reverse direction, the motion would become more diffusive (although not as bad as Metropolis) and would lead to slower decay of density fluctuations. In higher dimensions, most of the above arguments remain unchanged, except that in that case, it is important to have unidirectional moves in all spatial dimensions.

The next advance in **ECMC** algorithms was the extension of the method of [598] to systems with continuous interactions (early attempts are described in [600,602]). In the case of continuous interactions, the rejection of a trial move of a particle i cannot be attributed uniquely to the interaction with a single other particle. The way to resolve this problem is described in some detail in refs. [599,603]: here we give a very condensed description.

The crucial point to note is that if we decompose the move of a particle i in infinitesimal steps, then for every step the probability to reject the move is non-zero as long as the interaction with another particle j *increases* (if the interaction energy with j decreases, then j cannot stop the motion of i). Of course, the energy increase due to an infinitesimal step is itself infinitesimal, and so is the rejection probability. However, we can now use Eq. (13.4.21) to compute the probability that the interaction with particle j could lead to the rejection of a move after a displacement s^* of particle i in the chosen direction $\hat{\mathbf{e}}$. It is

$$P_{\text{reject}}(i; j) = 1 - \exp \left(-\beta \int_0^{s_{ij}^*} ds \left(\frac{\partial u(\mathbf{r}_j - \mathbf{r}_i - s\hat{\mathbf{e}})}{\partial s} \right)^+ \right), \quad (13.4.29)$$

where $\mathbf{r}_j - \mathbf{r}_i \equiv \mathbf{r}_j - \mathbf{r}_i$ is the vectorial distance between i and j at the beginning of the move, and the superscript “+” indicates that we only consider those contributions for which the pair interaction is increasing with increasing s .

Using a slight generalization of Eq. (13.3.5) we can then generate values of s_{ij}^* according to a distribution that gives the correct rejection probability, by drawing a random number $0 < R \leq 1$ and evaluation the s_{ij}^* for which

$$\beta \int_0^{s_{ij}^*} ds \left(\frac{\partial u(\mathbf{r}_j - \mathbf{r}_i - s\hat{\mathbf{e}})}{\partial s} \right)^+ = -\ln R. \quad (13.4.30)$$

To evaluate the distance s_{ij}^* , the use to the parsimonious Metropolis algorithm 3.2.2 is advantageous.

For many simple forms of the intermolecular potential, Eq. (13.4.30) can be solved analytically (or very nearly so). Of course, it is possible that this equation has no solution (namely when the potential energy decreases, or when the integral is less than $-\ln R$ even for $s_{ij}^* \rightarrow \infty$). In such cases, particle j cannot block the move of i . The calculation of s_{ij}^* must be carried out for all potential interaction partners j of particle i . As explained below, in practice, the calculation usually can be done without explicitly considering particles j that are not very close to i and therefore unlikely to block the move of i .

Cell-veto method

An elegant way to avoid having to consider *all* pairs ij explicitly is the *cell-veto* method of ref. [601], explained in more detail in ref. [89]. In this approach, the system is divided up into cells that are chosen small enough that they will typically contain at most one particle.⁴ We consider the case that we have chosen to move particle i . We now explicitly consider all other particles j that are in the cells surrounding the cell of i . However, for all other, more distant, cells we do something else: we compute the maximum possible value of the “rejection rate”

$$q(\mathbf{r}_{S_i, S_j}) \equiv \left(\frac{\partial u(\mathbf{r}_{S_j} - \mathbf{r}_{S_i} - s\hat{\mathbf{e}})}{\partial s} \right)^+$$

for a particle on the surface (S_i) of \mathcal{C}_i , cell of i , interacting with another particle on the surface (S_j) of \mathcal{C}_j , the cell of j . Note that the real rejection rate for particle i anywhere in \mathcal{C}_i must be lower. To compute this maximum rejection rate, $q_{\max}(\mathcal{C}_i, \mathcal{C}_j)$, we must scan the surfaces of cells \mathcal{C}_i and \mathcal{C}_j to find the pair of points that maximizes the rejection rate, but we do this only once, at the beginning of the simulation. For a displacement of particle i by an amount s , the rejection probability is then bounded from above by

$$p_{\text{reject}}(s; \mathcal{C}_i, \mathcal{C}_j) = (1 - \exp[-sq_{\max}(\mathcal{C}_i, \mathcal{C}_j)]). \quad (13.4.31)$$

For every cell \mathcal{C}_j , we can now compute $q_{\max}(\mathcal{C}_i, \mathcal{C}_j)$, the upper limit to the rate at which a particle in cell \mathcal{C}_j can block the move of particle i .

Next, we can sum the maximum rejection rates of all these cells, to obtain

$$Q_{\max} \equiv \sum_{\mathcal{C}_j} q_{\max}(\mathcal{C}_i, \mathcal{C}_j),$$

the upper bound to the rejection rate of a move of particle i anywhere in cell \mathcal{C}_i (note that Q_{\max} is the same for all cells: it only depends on the form of the interaction potential and the lattice geometry). The upper bound to the cell rejection probability for a displacement over a distance s is then $1 - \exp[-sQ_{\max}]$.

⁴ Ref. [601] explains what to do if, nevertheless, a cell contains more than one particle.

We can now proceed as before: we use Eq. (13.4.30) to compute s_{ij}^* for all (nearby) particles j that are considered explicitly and the $s^*(\text{Cells})$ for the distance that the move might be blocked by any of the other particles. If the smallest s^* corresponds to any of the particles j considered explicitly, we carry out the displacement and make particle j active. However, if the smallest s^* is due to the Q_{\max} , then we have to select one of the cells (say k) with a probability

$$P(k) = \frac{q_{\max}(\mathcal{C}_i, \mathcal{C}_k)}{\sum_{\mathcal{C}_j} q_{\max}(\mathcal{C}_i, \mathcal{C}_j)}.$$

This cell-selection step can be carried out in $\mathcal{O}(1)$ time, using Walker's algorithm [571]. If there is a particle (say k) in the cell, we first use Eq. (13.4.29) to compute the real probability that this particle would block a move of length s^* of particle i . Particle i is then blocked by particle k probability

$$\frac{P_{\text{reject}}(i; k)}{q_{\max}(\mathcal{C}_i, \mathcal{C}_k)}.$$

If the further movement of i is rejected by k , then i is moved over a distance s^* and k becomes the active particle. However, if the move is *not* blocked by particle k , then particle i is moved over a distance s^* , *but remains the active one*. One special case is when s^* would move particle i out of its cell: in that case, i is only moved to (or better just across) the cell boundary, but it remains the active particle. One point should be stressed: for the cell-veto method all periodic images of a particle k should be “folded” into the image of k nearest to i . For details, see [89,601].

Of particular importance is the fact that the cell-veto method can be used to simulate systems with very long-ranged (e.g., Coulomb) interactions, where at every step, the interaction need only be computed for one particle pair (see section 11.6.2). For more details, we refer the reader to [89].

Chapter 14

Time-scale-separation problems in MD

Molecules are made up of atoms. Hence, one might expect that Molecular Dynamics simulations of molecules can be performed using the algorithms used to simulate nonbonded atomic systems, as long as it is legitimate to ignore the quantum nature of intramolecular motions. In practice, it is usually not advisable to integrate the equations of motion of internal modes in molecules, employing the same algorithms used for simulating unbonded atoms. The reason is that the characteristic time scales associated with intramolecular motions are typically 10–50 shorter than the typical decorrelation time of the translational velocity of the same molecule in a liquid.

In a Molecular Dynamics simulation, time steps should be chosen such that they are appreciably shorter than the shortest relevant time scale in the simulation. If we simulate the intramolecular dynamics of molecules explicitly, this implies that our time step should be shorter than the period of the highest-frequency intramolecular vibration. This condition would make the simulation of molecular substances very time-consuming. Several techniques for tackling this problem have been developed. Here, we will discuss three approaches: constraints, extended Lagrangians, and multiple-time-step simulations.

Multiple-time-scale Molecular Dynamics [117], is based on the observation that forces associated with a high-frequency intramolecular vibration can be integrated efficiently with a time step that is different from the one used to integrate the intermolecular vibrations. An alternative is to treat the bonds (and, sometimes, bond angles) in molecules as rigid. The Molecular Dynamics equations of motion are then solved under the constraint that the rigid bonds and bond angles do not change during our simulation. This procedure should eliminate the highest-frequency modes in the dynamics: the motion associated with the remaining degrees of freedom is presumably slower, and hence we can again use a long time step in our simulations. Below, we briefly explain how such constraints are implemented in a Molecular Dynamics simulation.

In addition, we illustrate how extended Lagrangians can be used in “on-the-fly” optimization problems. The foremost example of a technique that uses an extended Lagrangian for this purpose is the original Car-Parrinello “*ab initio*” MD method [605]. We shall not discuss this technique because quantum simulations fall outside the scope of this book. Rather, we shall illustrate the

application of the extended Lagrangian approach in the Car-Parrinello method to a purely classical optimization problem.

14.1 Constraints

Constraints on the classical equations of motion are best expressed in the language of Lagrangian dynamics (see [54] and Appendix A). To get a feel for the way in which constrained dynamics works, let us consider a simple example, namely a single particle that is constrained to move on the surface of a 3d sphere with radius d . The constraint is then of the form: $f(x, y, z) \equiv x^2 + y^2 + z^2 - d^2 = 0$.¹ The Lagrangian equations of motion for the unconstrained particle are (see Appendix A)

$$\frac{\partial}{\partial t} \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} = \frac{\partial \mathcal{L}}{\partial q}. \quad (14.1.1)$$

As the Lagrangian, \mathcal{L} , is equal to $\mathcal{K}_{\text{kin}} - \mathcal{U}_{\text{pot}}$, the equation of motion for the unconstrained particle is

$$m\ddot{\mathbf{q}} = -\frac{\partial \mathcal{U}}{\partial \mathbf{q}}.$$

Now, suppose that we start with a particle located on the surface $f(x, y, z) = 0$ and moving initially tangentially to the constraint surface, that is:

$$\dot{\mathbf{f}} = \dot{\mathbf{q}} \cdot \nabla f = 0.$$

Without any constraints, the particle would move away from the surface of the sphere, and its velocity would no longer be tangential to the constraint surface. To keep the particle on the constraint surface, we now apply a fictitious force (the constraint force) in such a way that the new velocity is again perpendicular to ∇f .

In a many-body simulation, the dynamics often must satisfy many constraints simultaneously (e.g., many bond lengths). Let us denote the functions describing these constraints by $\sigma_1, \sigma_2, \dots$. For instance, σ_1 may be a function that is equal to 0 when atoms i and j are at a fixed distance d_{ij} :

$$\sigma_1(\mathbf{r}_i, \mathbf{r}_j) = r_{ij}^2 - d_{ij}^2.$$

We now introduce a new Lagrangian \mathcal{L}' that contains all the constraints:

$$\mathcal{L}' = \mathcal{L} - \sum_{\alpha} \lambda_{\alpha} \sigma_{\alpha}(\mathbf{r}^N),$$

where α denotes the set of constraints and λ_{α} denotes a set of (as yet undetermined) Lagrange multipliers. The equations of motion corresponding to this new Lagrangian are

¹ A constraint that can be written as a relation between the particle coordinates of the form $f(\mathbf{r}^N) = 0$, is called a *holonomic* constraint.

$$\frac{\partial}{\partial t} \frac{\partial \mathcal{L}'}{\partial \dot{q}} = \frac{\partial \mathcal{L}'}{\partial q} \quad (14.1.2)$$

or

$$\begin{aligned} m_i \ddot{q}_i &= -\frac{\partial \mathcal{U}}{\partial q_i} - \sum_{\alpha} \lambda_{\alpha} \frac{\partial \sigma_{\alpha}}{\partial q_i} \\ &\equiv F_i + \sum_{\alpha} G_i(\alpha). \end{aligned} \quad (14.1.3)$$

The last line of this equation defines the constraint force \mathbf{G}_{α} . To solve for the set λ_{α} , we require that the second derivatives of all σ_{α} vanish (our initial conditions were chosen such that the first derivatives vanished):

$$\begin{aligned} \frac{\partial \dot{\sigma}_{\alpha}}{\partial t} &= \frac{\partial \dot{q} \nabla \sigma_{\alpha}}{\partial t} \\ &= \ddot{q} \nabla \sigma_{\alpha} + \dot{q} \dot{q} : \nabla \nabla \sigma_{\alpha} \\ &= 0. \end{aligned} \quad (14.1.4)$$

Using Eq. (14.1.3), we can rewrite this equation as

$$\begin{aligned} \frac{\partial \dot{\sigma}_{\alpha}}{\partial t} &= \sum_i \frac{1}{m_i} \left[F_i + \sum_{\beta} G_i(\beta) \right] \nabla_i \sigma_{\alpha} + \sum_{i,j} \dot{q}_i \dot{q}_j \nabla_i \nabla_j \sigma_{\alpha} \\ &= \sum_i \frac{1}{m_i} F_i \nabla_i \sigma_{\alpha} - \sum_i \frac{1}{m_i} \sum_{\beta} \lambda_{\beta} \nabla_i \sigma_{\beta} \nabla_i \sigma_{\alpha} + \sum_{i,j} \dot{q}_i \dot{q}_j \nabla_i \nabla_j \sigma_{\alpha} \\ &\equiv \mathcal{F}_{\alpha} - \Lambda_{\beta} M_{\alpha\beta} + \mathcal{T}_{\alpha} \\ &= 0. \end{aligned} \quad (14.1.5)$$

In the last line of Eq. (14.1.5), we have written the equation on the previous line in matrix notation. The formal solution of this equation is

$$\Lambda = M^{-1}(\mathcal{F} + \mathcal{T}). \quad (14.1.6)$$

This formal solution of the Lagrangian equations of motion in the presence of constraints is, unfortunately, of little practical use. The reason is that, in a simulation, we do not solve differential equations but difference equations. Hence there is little point in going through the (time-consuming) matrix inversion needed for the exact solution of the differential equation, because this procedure does not guarantee that the constraints also will be accurately satisfied in the solution of the difference equation.

Before proceeding, let us again consider the simple example of a particle moving on the surface of a sphere with radius d . In that case, we can write our constraint function σ as

$$\sigma = \frac{1}{2} (r^2 - d^2).$$

The factor 1/2 has been included to make the following equations simpler. The constraint force \mathbf{G} is equal to

$$\mathbf{G} = -\lambda \nabla \sigma = -\lambda \mathbf{r}.$$

To solve for λ , we impose $\ddot{\sigma} = 0$:

$$\begin{aligned}\partial_t \dot{\sigma} &= \partial_t (\dot{\mathbf{r}} \cdot \mathbf{r}) \\ &= (\ddot{\mathbf{r}} \cdot \mathbf{r}) + \dot{\mathbf{r}}^2 = 0.\end{aligned}\quad (14.1.7)$$

The Lagrangian equation of motion is

$$\begin{aligned}\ddot{\mathbf{r}} &= \frac{1}{m}(\mathbf{F} + \mathbf{G}) \\ &= \frac{1}{m}(\mathbf{F} - \lambda \mathbf{r}).\end{aligned}\quad (14.1.8)$$

For convenience, we assume that no external forces are acting on the particle ($\mathbf{F} = \mathbf{0}$). Combining Eqs. (14.1.7) and (14.1.8), we obtain

$$-\frac{\lambda}{m}r^2 + \dot{\mathbf{r}}^2 = 0. \quad (14.1.9)$$

Hence

$$\lambda = \frac{m\dot{\mathbf{r}}^2}{r^2},$$

and the constraint force \mathbf{G} is equal to

$$\mathbf{G} = -\lambda \mathbf{r} = -\frac{m\dot{\mathbf{r}}^2}{r^2} \mathbf{r}.$$

Recall that, on the surface of a sphere, the velocity $\dot{\mathbf{r}}$ is simply equal to ωr . Hence we can also write the constraint force as

$$\mathbf{G} = -m\omega^2 \mathbf{r},$$

which is the well-known expression for the centripetal force.

This simple example helps us to understand what goes wrong if we insert the preceding expression for the constraint force into an MD algorithm, e.g., in the Verlet scheme. In the absence of external forces, we would get the following algorithm for a particle on the surface of a sphere:

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) - \omega^2 \Delta t^2 \mathbf{r}(t).$$

How well is the constraint $r^2 = d^2$ satisfied? To get an impression, we work out the expression for r^2 after one time-step. Assuming that the constraint was

satisfied at $t = 0$ and at $t = -\Delta t$, we find that, at $t = \Delta t$,

$$\begin{aligned} r^2(t + \Delta t) &= d^2 \left\{ 5 + (\omega \Delta t)^4 - 4(\omega \Delta t)^2 + \cos(\omega \Delta t)[2(\omega \Delta t)^2 - 4] \right\} \\ &\approx d^2 \left[1 - \frac{(\omega \Delta t)^4}{6} + \mathcal{O}(\Delta t^6) \right]. \end{aligned}$$

At first sight, this looks reasonable, and the constraint violation is of order $(\Delta t)^4$, as is to be expected for the Verlet scheme. However, whereas for center-of-mass motion, we do not worry too much about errors of this order in the trajectories, we should worry in the case of constraints. In the case of translational motion, we argued that two trajectories that are initially close but subsequently diverge exponentially may still both be representative of the true trajectories of the particles in the system. However, if we find that, due to small errors in the integration of the equations of motion, the numerical trajectories diverge exponentially from the constraint surface, then we are in deep trouble. The conclusion is that we should not rely on our algorithm to satisfy the constraints (although, in fact, for the particle on a sphere, the Verlet algorithm performs remarkably well). We should construct our algorithm such that the constraints are rigorously obeyed.

The most straightforward solution to this problem is not to fix the Lagrange multiplier λ by the condition that the second derivative of the constraint vanishes but by the condition that the constraint is exactly satisfied after one time-step. In the case of the particle on a sphere, this approach would work as follows. The equation for the position at time $t + \Delta t$ in the presence of the constraint force is given by

$$\begin{aligned} \mathbf{r}(t + \Delta t) &= 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) - \frac{\lambda}{m}\mathbf{r}(t) \\ &= \mathbf{r}_u(t + \Delta t) - \frac{\lambda}{m}\mathbf{r}(t), \end{aligned}$$

where $r_u(t + \Delta t)$ denotes the new position of the particle in the absence of the constraint force. We now impose that the constraint $r^2 = d^2$ is satisfied at $t + \Delta t$:

$$\begin{aligned} d^2 &= \left[\mathbf{r}_u(t + \Delta t) - \frac{\lambda}{m}\mathbf{r}(t) \right]^2 \\ &= r_u^2(t + \Delta t) - \frac{2\lambda}{m}\mathbf{r}(t) \cdot \mathbf{r}_u(t + \Delta t) + \left[\frac{\lambda}{m}r(t) \right]^2. \end{aligned}$$

This expression is a quadratic equation in λ ,

$$\left(\frac{\lambda}{m}d \right)^2 - \frac{2\lambda}{m}\mathbf{r}(t) \cdot \mathbf{r}_u(t + \Delta t) + \mathbf{r}_u^2(t + \Delta t) - d^2 = 0,$$

and the solution is

$$\lambda = \frac{\mathbf{r}(t) \cdot \mathbf{r}_i(t + \Delta t) - \sqrt{[\mathbf{r}(t) \cdot \mathbf{r}_i(t + \Delta t)]^2 - d^2[r_i^2(t + \Delta t) - d^2]}}{d^2/m}.$$

For the trivial case of a particle on a spherical surface, this approach clearly will work. However, for a large number of constraints, it will become difficult, or even impossible, to solve the quadratic constraint equations analytically. Why this is so can be seen by considering the form of the Verlet algorithm in the presence of ℓ constraints:

$$\mathbf{r}_i^{\text{constrained}}(t + \Delta t) = \mathbf{r}_i^{\text{unconstrained}}(t) - \frac{\Delta t^2}{m_i} \sum_{k=1}^{\ell} \lambda_k \nabla_i \sigma_k(t). \quad (14.1.10)$$

If we satisfy the constraints at time $t + \Delta t$, then $\sigma_k^c(t + \Delta t) = 0$. But if the system would move along the unconstrained trajectory, the constraints would not be satisfied at $t + \Delta t$. We assume that we can perform a Taylor expansion of the constraints:

$$\begin{aligned} \sigma_k^c(t + \Delta t) &= \sigma_k^u(t + \Delta t) + \sum_{i=1}^N \left(\frac{\partial \sigma_k}{\partial \mathbf{r}_i} \right)_{r_i^u(t+\Delta t)} \cdot [\mathbf{r}_i^c(t + \Delta t) - \mathbf{r}_i^u(t + \Delta t)] \\ &\quad + \mathcal{O}(\Delta t^4). \end{aligned} \quad (14.1.11)$$

If we insert Eq. (14.1.10) for $r_i^u - r_i^c$ in Eq. (14.1.11), we get

$$\sigma_k^u(t + \Delta t) = \sum_{i=1}^N \frac{\Delta t^2}{m_i} \sum_{k'=1}^{\ell} \nabla_i \sigma_k(t + \Delta t) \nabla_i \sigma_{k'}(t) \lambda_{k'}. \quad (14.1.12)$$

We note that Eq. (14.1.12) has the structure of a matrix equation:

$$\boldsymbol{\sigma}^u(t + \Delta t) = \Delta t^2 \mathbf{M} \boldsymbol{\Lambda}. \quad (14.1.13)$$

By inverting the matrix, we can solve for the vector $\boldsymbol{\Lambda}$. However, as we had truncated the Taylor expansion in Eq. (14.1.11), we should then compute the σ 's at the corrected positions, and iterate the preceding equations until convergence is reached.

Although the approach sketched here will work, it is not computationally cheap because it requires a matrix inversion at every iteration. In practice, therefore, one often uses a simpler iterative scheme to satisfy the constraints. In this scheme, called *SHAKE* [606], the iterative scheme just sketched is not applied to all constraints simultaneously but to each constraint in succession. To be more precise, we use the Taylor expansion of Eq. (14.1.11) for σ_k , but then we approximate $r_i^c - r_i^u$ as

$$\mathbf{r}_i^c(t + \Delta t) - \mathbf{r}_i^u(t) \approx -\frac{\Delta t^2 \lambda_k}{m_i} \nabla_i \sigma_k(t). \quad (14.1.14)$$

If we insert Eq. (14.1.14) in Eq. (14.1.11), we get

$$\sigma_k^u(t + \Delta t) = \Delta t^2 \lambda_k \sum_{i=1}^N \frac{1}{m_i} \nabla_i \sigma_k(t + \Delta t) \nabla_i \sigma_k(t), \quad (14.1.15)$$

and hence our estimate for λ_k is

$$\lambda_k \Delta t^2 = \frac{\sigma_k^u(t + \Delta t)}{\sum_{i=1}^N \frac{1}{m_i} \nabla_i \sigma_k(t + \Delta t) \nabla_i \sigma_k(t)}. \quad (14.1.16)$$

In a simulation, we treat all constraints in succession during one cycle of the iteration and then repeat the process until all constraints have converged to within the desired accuracy. The above implementation of constrained dynamics is based on the normal (position) Verlet algorithm (Eq. (4.2.3)). Andersen [607] has shown how to impose constraints in MD simulations that use the Velocity Verlet algorithm (Eqs. (4.3.4) and (4.3.5)), and De Leeuw et al. [608] showed how the problem of constrained dynamics can be cast in Hamiltonian form.

Computing the derivatives of a constraint is not always enjoyable, in particular when the quantity that is being constrained is a complicated function of the coordinates of many particles, as can happen when constraining an order parameter (see section 15.2.1). In such cases, using Automatic Differentiation (see e.g., [108]) may become advantageous.

14.1.1 Constrained and unconstrained averages

Thus far, we have presented constrained dynamics as a convenient scheme for modeling the motion of molecules with stiff internal bonds. The advantage of using constrained dynamics was that we could use a longer time step in our Molecular Dynamics algorithm when the high-frequency vibrations associated with the stiff degrees of freedom were eliminated. However, somewhat surprisingly, the results of a constrained simulation depend on *how* the constraints are imposed: as was pointed out by Fixman [97], a simulation that uses hard constraints that are introduced in the Lagrangian equations of motion, does not yield the same averages as a simulation, where the constraints are represented by arbitrarily stiff but non-rigid bonds. Below we reproduce an argument by Van Kampen [98], showing that different expressions for the bond-angle distribution of a fully flexible trimer (see Fig. 14.1) are obtained in the case of hard and soft constraints.

We wish to fix the bond lengths r_{12} and r_{23} . We can do this in two ways. One is to impose the constraints $r_{12}^2 = d^2$ and $r_{23}^2 = d^2$ in the Lagrangian equations of motion of the trimer. The other is to link the atoms in the trimer by harmonic springs, such that

$$\mathcal{U}_{\text{Harmonic}} = \frac{\alpha}{2} \left[(r_{12} - d)^2 + (r_{23} - d)^2 \right].$$

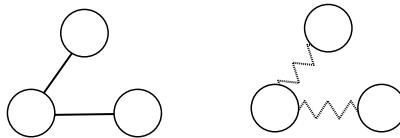


FIGURE 14.1 Symmetric trimer with bond length d and internal bond angle ψ . (Left) Bonds are represented by an infinitely stiff spring; (right) bonds are represented by hard constraints in the Lagrangian equations of motion of the trimer.

Intuitively, one might expect that the limit $\alpha \rightarrow \infty$ would be equivalent to dynamics with hard constraints, but this is not so. In fact, if we look at $P(\psi)$, the distribution of the internal angle ψ , we find that

$$\begin{aligned} P(\psi) &= c \sin \psi && \text{(Harmonic forces)} \\ P(\psi) &= c \sin \psi \sqrt{1 - (\cos \psi)^2 / 4} && \text{(Hard constraints).} \end{aligned} \quad (14.1.17)$$

Next, we sketch the origins of this difference in the behavior of “hard” and “soft” constraints. To this end, we start with the Lagrangian of the system, $\mathcal{L} = \mathcal{K} - \mathcal{U}$. Thus far, we had expressed the kinetic (\mathcal{K}) and potential (\mathcal{U}) energy of the system in terms of the Cartesian velocities and coordinates of the atoms. However, when we talk about bonds and bond angles or, for that matter, any other function of the coordinates that has to be kept constant, it is more convenient to use generalized coordinates, denoted by q . We choose our generalized coordinates such that every quantity we wish to constrain corresponds to a single generalized coordinate. We denote by q_H the set of generalized coordinates that describes the quantities that are effectively, or rigorously, fixed. The remaining soft coordinates are denoted by q_S . The potential energy function \mathcal{U} is a function of both q_H and q_S :

$$\mathcal{U}(q) = \mathcal{U}(q_H, q_S).$$

If we rigorously fix the hard coordinates such that $q_H = \sigma$, then the potential energy is a function of q_S , while it depends parametrically on σ :

$$\mathcal{U}_{\text{hard}}(q_S) = \mathcal{U}_{\text{soft}}(\sigma, q_S).$$

Let us now express the Lagrangian in terms of these generalized coordinates:

$$\begin{aligned} \mathcal{L} &= \frac{1}{2} \sum_{i=1}^N m_i \dot{r}_i^2 - \mathcal{U} \\ &= \frac{1}{2} \sum_{i=1}^N m_i \dot{q}_\alpha \frac{\partial \mathbf{r}_i}{\partial q_\alpha} \cdot \frac{\partial \mathbf{r}_i}{\partial q_\beta} \dot{q}_\beta - \mathcal{U} \\ &\equiv \frac{1}{2} \dot{\mathbf{q}} \cdot \mathbf{G} \cdot \dot{\mathbf{q}} - \mathcal{U}, \end{aligned} \quad (14.1.18)$$

where the last line of Eq. (14.1.18) defines the mass-weighted metric tensor G . We can now write the expression for the generalized momentum:

$$p_\alpha \equiv \frac{\partial \mathcal{L}}{\partial \dot{q}_\alpha} = G_{\alpha\beta} \dot{q}_\beta, \quad (14.1.19)$$

where summation of the repeated index β is implied. Next, we can write the Hamiltonian, \mathcal{H} as a function of generalized coordinates and momenta:

$$\mathcal{H} = \frac{1}{2} \mathbf{p} \cdot G^{-1} \cdot \mathbf{p} + U(\mathbf{q}).$$

Once we have the Hamiltonian, we can write an expression for the equilibrium phase space density that determines all thermal averages. Although one could write expressions for all averages in the microcanonical ensemble (constant N , V , E), this is, in fact, not very convenient. Hence, we shall consider canonical averages (constant N , V , T). It is straightforward to write the expression for the canonical distribution function in terms of the generalized coordinates and momenta:

$$\rho(\mathbf{p}, \mathbf{q}) = \frac{\exp[-\beta \mathcal{H}(\mathbf{p}, \mathbf{q})]}{Q_{NVT}} \quad (14.1.20)$$

with

$$Q_{NVT} = \int d\mathbf{p} d\mathbf{q} \exp[-\beta \mathcal{H}(\mathbf{p}, \mathbf{q})]. \quad (14.1.21)$$

The reason we can write Eq. (14.1.20) in this simple form is that the Jacobian of the transformation from Cartesian coordinates to generalized coordinates is 1. Let us now look at the canonical probability distribution function as a function of \mathbf{q} only:

$$\begin{aligned} \rho(\mathbf{q}) &= c \int d\mathbf{p} \exp\{-\beta[\mathbf{p} \cdot G^{-1} \cdot \mathbf{p}/2 + U(\mathbf{q})]\} \\ &= c' \exp[-\beta U(\mathbf{q})] \sqrt{|G|}, \end{aligned} \quad (14.1.22)$$

where $|G|$ denotes the (absolute value of the) determinant of G and c and c' are normalizing constants.

Thus far, we have not mentioned constraints. We have simply transformed the canonical distribution function from one set of phase space coordinates to another. Clearly, the answers will not depend on our choice of these coordinates. But now we introduce constraints. That is, in our Lagrangian (14.1.18) we remove the contribution to the kinetic energy due to the dynamics of the hard coordinates; that is, we set $\dot{\mathbf{q}}_H = \mathbf{0}$, and in the potential energy function, we replace the coordinates \mathbf{q}_H by the parameters σ . The Lagrangian for the system

with constraints is

$$\begin{aligned}\mathcal{L}^H &= \frac{1}{2} \sum_{i=1}^N m_i \dot{r}_i^2 - \mathcal{U} \\ &= \frac{1}{2} \sum_{i=1}^N m_i \dot{q}_\alpha^S \frac{\partial \mathbf{r}_i}{\partial q_\alpha^S} \cdot \frac{\partial \mathbf{r}_i}{\partial q_\beta^S} \dot{q}_\beta^S - \mathcal{U}(\mathbf{q}^S, \boldsymbol{\sigma}) \\ &\equiv \frac{1}{2} \dot{\mathbf{q}}^S \cdot \mathbf{G}_S \cdot \dot{\mathbf{q}}^S - \mathcal{U}(\mathbf{q}^S, \boldsymbol{\sigma}).\end{aligned}\quad (14.1.23)$$

Note that the number of variables has decreased from $3N$ to $3N - \ell$, where ℓ is the number of constraints. The Hamiltonian of the constrained system is

$$\mathcal{H}^H = \frac{1}{2} \mathbf{p}^S \cdot \mathbf{G}_S^{-1} \cdot \mathbf{p}^S + \mathcal{U}(\mathbf{q}^S, \boldsymbol{\sigma}),$$

where

$$p_\alpha^S \equiv \frac{\partial \mathcal{L}}{\partial \dot{q}_\alpha^S}. \quad (14.1.24)$$

As before, we can write the phase space density. In this case, it is most convenient to write this density directly as a function of the generalized coordinates and momenta:

$$\rho(\mathbf{p}^S, \mathbf{q}^S) = \frac{\exp[-\beta \mathcal{H}(\mathbf{p}^S, \mathbf{q}^S)]}{Q_{NVT}^S}. \quad (14.1.25)$$

Let us now write the probability density in coordinate space:

$$\begin{aligned}\rho(\mathbf{q}^S) &= a \int d\mathbf{p}^S \exp\{-\beta[\mathbf{p}^S \cdot \mathbf{G}_S \cdot \mathbf{p}^S / 2 + \mathcal{U}(\mathbf{q}^S, \boldsymbol{\sigma})]\} \\ &= a' \exp[-\beta \mathcal{U}(\mathbf{q}^S, \boldsymbol{\sigma})] \sqrt{|\mathbf{G}_S|},\end{aligned}\quad (14.1.26)$$

where a and a' are normalizing constants. Now compare this expression with the result that we would have obtained if we had applied very stiff springs to impose the constraints. In that case, we would have to use Eq. (14.1.22). For $\mathbf{q}^H = \boldsymbol{\sigma}$, Eq. (14.1.22) predicts

$$\rho(\mathbf{q}^S) = c' \exp[-\beta \mathcal{U}(\mathbf{q}^S, \boldsymbol{\sigma})] \sqrt{|\mathbf{G}|}, \quad (14.1.27)$$

and this is not the same result as given by Eq. (14.1.26). Ignoring constant factors, the ratio of the probabilities in the constrained and unconstrained system is given by

$$\frac{\rho(\mathbf{q}^S)}{\rho(\mathbf{q}^S, \mathbf{q}_H = \boldsymbol{\sigma})} = \sqrt{\frac{|\mathbf{G}_S|}{|\mathbf{G}|}}.$$

This implies that, if we do a simulation in a system with hard constraints, and we wish to predict average properties for the system with “stiff-spring” constraints, then we must compute a weighted average with a weight factor $\sqrt{|G|/|G_S|}$ to compensate for the bias in the distribution function of the constrained system. Fortunately, it is usually easier to compute the ratio $|G|/|G_S|$ than to compute $|G|$ and $|G_S|$ individually. To see this, consider the inverse of G

$$G_{\alpha\beta}^{-1} = \sum_{i=1}^N m_i^{-1} \frac{\partial q_\alpha}{\partial \mathbf{r}_i} \cdot \frac{\partial q_\beta}{\partial \mathbf{r}_i}.$$

It is easy to verify that this is indeed the inverse of G :

$$\begin{aligned} G_{\alpha\beta} G_{\beta\gamma}^{-1} &= \sum_{i,j=1}^N m_i \frac{\partial \mathbf{r}_i}{\partial q_\alpha} \frac{\partial \mathbf{r}_i}{\partial q_\beta} \frac{\partial q_\beta}{\partial \mathbf{r}_j} \frac{\partial q_\gamma}{\partial \mathbf{r}_j} m_j^{-1} \\ &= \sum_{i=1}^N \frac{\partial \mathbf{r}_i}{\partial q_\alpha} \frac{\partial q_\gamma}{\partial \mathbf{r}_i} \\ &= \delta_{\alpha\gamma}. \end{aligned} \quad (14.1.28)$$

Now, let us write both the matrices G and G^{-1} in block form

$$G = \left(\begin{array}{c|c} G_S & A_{SH} \\ \hline A_{HS} & A_{HH} \end{array} \right) \quad (14.1.29)$$

and

$$G^{-1} = \left(\begin{array}{c|c} B_{SS} & B_{SH} \\ \hline B_{HS} & H \end{array} \right), \quad (14.1.30)$$

where the subscripts S and H denote soft and hard coordinates, respectively. The submatrix H is simply that part of G^{-1} that is quadratic in the derivatives of the constraints:

$$H_{\alpha\beta} = \sum_{i=1}^N m_i^{-1} \frac{\partial \sigma_\alpha}{\partial \mathbf{r}_i} \frac{\partial \sigma_\beta}{\partial \mathbf{r}_i}.$$

Now we construct a matrix X as follows. We take the first $3N - \ell$ columns of G and we complete it with the last ℓ columns of the unit matrix:

$$X = \left(\begin{array}{c|c} G_S & 0 \\ \hline A_{HS} & I \end{array} \right). \quad (14.1.31)$$

From the block structure of X , it is obvious that the determinant of X is equal to the determinant of G_S . Next, we multiply X with GG^{-1} , that is, with the unit

matrix. Straightforward matrix multiplication shows that

$$GG^{-1}X = G \left(\begin{array}{c|c} I & B_{SH} \\ \hline 0 & H \end{array} \right). \quad (14.1.32)$$

Hence,

$$\begin{aligned} |X| &= |G_S| \\ &= |GG^{-1}X| \\ &= |G||H|. \end{aligned} \quad (14.1.33)$$

The final result is that

$$\frac{|G|}{|G_S|} = |H|. \quad (14.1.34)$$

We therefore can write the following relation between the coordinate space densities of the constrained and unconstrained systems:

$$\rho_{\text{flex}}(\mathbf{q}) = |H|^{-\frac{1}{2}} \rho_{\text{hard}}(\mathbf{q}). \quad (14.1.35)$$

The advantage of this expression is that we have expressed the ratio of the determinants of a $3N \times 3N$ matrix and a $3N - \ell \times 3N - \ell$ matrix, by the determinant of an $\ell \times \ell$ matrix. In many cases, this simplifies the calculation of the weight factor considerably.

As a practical example, let us consider the case of the flexible trimer, discussed at the beginning of this section. We have two constraints:

$$\begin{aligned} \sigma_1 &= r_{12}^2 - d^2 = 0 \\ \sigma_2 &= r_{23}^2 - d^2 = 0. \end{aligned}$$

If all three atoms have the same mass m , we can write $|H|$ as

$$|H| = \frac{1}{m} \begin{vmatrix} \sum_i \frac{\partial \sigma_1}{\partial \mathbf{r}_i} \frac{\partial \sigma_1}{\partial \mathbf{r}_i} & \sum_i \frac{\partial \sigma_1}{\partial \mathbf{r}_i} \frac{\partial \sigma_2}{\partial \mathbf{r}_i} \\ \sum_i \frac{\partial \sigma_2}{\partial \mathbf{r}_i} \frac{\partial \sigma_1}{\partial \mathbf{r}_i} & \sum_i \frac{\partial \sigma_2}{\partial \mathbf{r}_i} \frac{\partial \sigma_2}{\partial \mathbf{r}_i} \end{vmatrix}.$$

Inserting the expressions for σ_1 and σ_2 , we find that

$$|H| = \frac{2}{m} \begin{vmatrix} 2r_{12}^2 & -\mathbf{r}_{12} \cdot \mathbf{r}_{23} \\ -\mathbf{r}_{12} \cdot \mathbf{r}_{23} & 2r_{23}^2 \end{vmatrix}.$$

Using the fact that $r_{12}^2 = r_{23}^2 = d^2$, we get

$$|H| = \frac{8}{m} \left(r_{12}^2 r_{23}^2 - \frac{1}{4} (\mathbf{r}_{12} \cdot \mathbf{r}_{23})^2 \right)$$

$$= \frac{8d^4}{m} \left(1 - \frac{\cos^2 \psi}{4} \right). \quad (14.1.36)$$

Finally, we recover Eq. (14.1.17) for the ratio of the probability densities for the constrained and unconstrained systems:

$$\frac{\rho_{\text{flex}}}{\rho_{\text{hard}}} = |\mathcal{H}|^{\frac{1}{2}} = c \sqrt{1 - \frac{\cos^2 \psi}{4}}. \quad (14.1.37)$$

This ratio varies between 1 and 0.866, that is, at most some 15%. It should be noted that, in general, the ratio depends on the masses of the particles that participate in the constraints. For instance, if the middle atom of our trimer is much lighter than the two end atoms, then $\sqrt{|\mathcal{H}|}$ becomes $\sqrt{1 - \cos^2 \psi} = |\sin \psi|$ and the correction due to the presence of hard constraints is not small. However, to put things in perspective, we should add that, at least for bond-length constraints of the type most often used in Molecular Dynamics simulations, the effect of the hard constraints on the distribution functions appears to be relatively small.

One obvious question is: which description is correct? Somewhat depressingly, for intra-molecular bonds, the answer is “neither”. The reason is that very stiff bonds tend to have a high vibration frequency and cannot be described by classical mechanics. In other cases (for instance, constraints on order parameters), the answer is: both methods can be used, as long as they are not mixed.

14.1.2 Beyond bond constraints

The above discussion of constrained dynamics was fairly general, but it mostly focused on the situation where the constraints had a simple geometric interpretation in terms of bond lengths. There are many examples of constraints that do not have a simple geometric interpretation. A common example is when simulations are carried out under conditions where an order-parameter of reaction coordinate is kept fixed. This application is important in the context of studying the crossing of a free-energy barrier by Molecular Dynamics [609] (see Chapter 15). It is also useful when computing free-energy *differences*: as we change an order parameter that characterizes a system (say the total dipole moment) from Q_A to Q_B , the change in free-energy is equal to the reversible work that must be performed to change the order parameter against the conjugate constraint force $f(Q)$:

$$\Delta F = w_{\text{rev}} = - \int_{Q_A}^{Q_B} dQ f(q).$$

In the previous section, we saw that there were two not completely equivalent methods to impose constraints: 1) by including holonomic constraints of the type $\sigma = 0$ in the Lagrangian equation of motion or 2) by approximating the constraint by stiff, usually harmonic, term (a “restraint”) in the Hamiltonian:

$H = H_{\text{unconstrained}} + (1/2)\kappa\sigma^2$. In the second case, small oscillations along the restraint direction are still possible, and they have an associated kinetic energy. In fact, equipartition fixes the average kinetic energy at $1/2k_B T$ per restrained degree of freedom. And for harmonic restraints, the average potential energy associated with the restraint is also determined by equipartition. As such stiff harmonic restraints thermalize very slowly, it is often necessary to couple the constrained degrees of freedom to a separate thermostat. Moreover, it is often convenient to set the temperature of this thermostat low, to minimize the fluctuations in the constraints.

A prototypical example of constraints that have no simple geometric interpretation appeared in the original Car-Parrinello scheme for ab-initio Molecular Dynamics [605]. In standard ab-initio MD, the **Density Functional Theory (DFT)** estimate of the electronic energy should be at a minimum. In DFT, the electronic energy depends parametrically on the coefficients (e.g., plane-wave amplitudes) that characterize the Kohn-Sham orbitals. Clearly, these amplitudes have no simple geometrical interpretation, but they are fixed by the condition that the Kohn-Sham orbitals are orthonormal, that the energy is at a minimum, and that the integrated electronic density is constant.

In the early Car-Parrinello approach, part of these constraints —namely those that keep the Kohn-Sham orbitals orthonormal—were implemented by including them as holonomic constraints in the Lagrangian equations of motion. However, restraining the energy to be near a minimum was imposed by treating the plane-wave amplitudes as coordinates, with a (fictitious) mass and an associated momentum. As a result, the system is never exactly in its DFT ground state, but close. In this respect, Car and Parrinello’s method was similar to the methods used by Andersen [607] and Nosé [248]: it uses an *extended Lagrangian* rather than a holonomic constraint on the Lagrangian equations of motion. However, we stress that it is but one choice: the alternative is to use holonomic constraints [610].

Below, we briefly discuss the Car-Parrinello-style approach to approximate complex constraints “on the fly”, as this approach is widely used for classical applications. For more details on the Car-Parrinello method for electronic structure calculations, we refer the reader to the book by Marx and Hutter [611] and the many early reviews that have been written on the subject (see e.g., [612–614]).

14.2 On-the-fly optimization

In the Car-Parrinello method, the electronic density fluctuates around its optimal (adiabatic) value. Even though at every step the system is not exactly in its electronic ground state, the electrons do not exert a systematic drag force on the nuclei, hence the slower nuclear dynamics is still correct.

A close classical analog of “ab initio” Molecular Dynamics is the method developed by Löwen et al. [615,616] to simulate counterion screening in colloidal suspensions of polyelectrolytes. In the approach of [615], the counterions

are described by classical density-functional theory and an extended Lagrangian method is used to keep the free energy of the counterions close to its minimum.

Here we consider a somewhat simpler application of the Car-Parrinello approach to a classical system. As before, the aim of the method is to replace the iterative optimization procedure with an extended dynamical scheme. As a specific example, we consider a fluid of point-polarizable molecules. The molecules have a static charge distribution that we leave unspecified (for instance, we could be dealing with ions, dipoles, or quadrupoles). We denote the polarizability of the molecules by α . The total energy of this system is given by

$$\mathcal{U} = U_0 + U_{\text{pol}},$$

where U_0 is the part of the potential energy that does not involve polarization. The induction energy, U_{pol} , is given by [617]

$$U_{\text{pol}} = - \sum_i \mathbf{E}_i \cdot \boldsymbol{\mu}_i + \frac{1}{2\alpha} \sum_i (\boldsymbol{\mu}_i)^2,$$

where \mathbf{E}_i is the *local* electric field acting on particle i and $\boldsymbol{\mu}_i$ is the dipole induced on particle i by this electric field. Of course, the local field depends on the values of all other charges in the system. For instance, in the case of dipolar molecules,

$$\mathbf{E}_i = \mathbf{T}_{ij} \cdot \boldsymbol{\mu}_j^{\text{tot}},$$

where \mathbf{T}_{ij} is the dipole-dipole tensor and $\boldsymbol{\mu}_j^{\text{tot}}$ is the total (i.e., permanent plus induced) dipole moment of molecule j . We assume that the induced dipoles follow the nuclear motion adiabatically and that U_{pol} is always at its minimum. Minimizing U_{pol} with respect to the $\boldsymbol{\mu}_i$ yields

$$\boldsymbol{\mu}_i = \alpha \mathbf{E}_i. \quad (14.2.1)$$

Hence, to properly account for the molecular polarizability of an N -particle system, we would have to solve a set of $3N$ linear equations at every time step. If we solve this set of equations iteratively, we must make sure that the solution has fully converged, because otherwise, the local field will exert a systematic drag force on the induced dipoles and the system will fail to conserve energy.

Now let us consider the Car-Parrinello approach to this optimization problem. The application of this extended Lagrangian method to polarizable molecules has been proposed by Rahman and co-workers [618] and by Sprik and Klein [619]. A closely related approach was subsequently advocated by Wilson and Madden [620]. The basic idea is to treat the magnitude of the induced dipoles as additional dynamical variables in the Lagrangian:

$$\mathcal{L}(\mathbf{r}^N, \boldsymbol{\mu}^N) = \frac{1}{2} \sum_{i=1}^N m \dot{\mathbf{r}}_i^2 + \frac{1}{2} \sum_{i=1}^N M \dot{\boldsymbol{\mu}}_i^2 - \mathcal{U}, \quad (14.2.2)$$

where M is the mass associated with the motion of the dipoles. This Lagrangian yields the following equations of motion for the dipole moments:

$$M\ddot{\mu}_i \equiv \frac{\partial \mathcal{L}}{\partial \dot{\mu}_i} = -\frac{\mu_i}{\alpha} + \mathbf{E}_i.$$

The right-hand side of this equation can be considered as a generalized force that acts on the dipoles. In the limit that this force is exactly zero the iterative scheme is recovered. If the temperature associated with the kinetic energy of the dipoles is sufficiently low, the dipoles will fluctuate around their lowest-energy configuration. More importantly, there will be no systematic drag force on the dipoles, and hence the energy of the system will not drift.

To make sure that the induced dipoles are indeed close to their ground-state configuration, we should keep the temperature of the induced-dipole degrees of freedom low. Yet, at the same time, the dipoles should be able to adapt rapidly (adiabatically) to changes in the nuclear coordinates to ensure that the condition of minimum energy is maintained during the simulation. This implies that the masses associated with the induced dipoles should be small. In summary, we require that

$$\begin{aligned} T_\mu &\ll T_r \\ M &\ll m, \end{aligned}$$

where the temperature of the induced dipoles is defined as

$$T_\mu = \frac{1}{2} \sum_{i=1}^N M \dot{\mu}_i^2,$$

while the translational temperature is related in the usual way to the kinetic energy

$$T_r = \frac{1}{2} \sum_{i=1}^N m \dot{\mathbf{r}}_i^2.$$

The condition that the temperature of the induced dipoles should be much lower than the translational temperature seems to create a problem because, in an ordinary simulation, the coupling between induced-dipole moments and translational motion leads to heat exchange. This heat exchange will continue until the temperature of the induced dipoles equals the translational temperature. Hence, it would seem that we cannot fix the temperature of the induced dipoles independent of the translational temperature. However, here we can again make use of thermostats. Sprik and Klein [619] showed that one can use two separate Nosé-Hoover thermostats to impose the temperature of the positions and to impose the (low) temperature of the polarization [621]. The mass M associated with the induced dipoles should be chosen such that the relaxation time of the

polarization is on the same order of magnitude as the fastest relaxation in the liquid.

As mentioned above, the extended Lagrangian approach is but one way to address the problem of complex “non-geometric” constraints. An alternative approach has been proposed by Coretti et al. [622] and Bonella et al. [610]. Although the latter approach starts from the extended-Lagrangian picture, it is different in that it considers the limit in which the mass associated with the dynamics of the restrained variables goes to zero. In this limit, the restraint becomes a constraint, and the usual constraint techniques (e.g., SHAKE) are used to maintain the constraint. An obvious advantage of this approach that there is no need for thermostating the dynamics of the unphysical coordinates: they are rigorously constrained by the physical coordinates. It would seem that many of the applications that now use extended Lagrangians could be recast in the form that uses restraint dynamics with zero mass.

14.3 Multiple time-step approach

An alternative scheme for dealing with the high-frequency vibrational modes of polyatomic molecules is based on the Trotter expansion Liouville representation of the classical equations of motion (Eq. (4.3.18)). The idea here is not just to separate the propagation of the coordinates and the momenta, but also to decompose the propagation of the high-frequency modes into many shorter time steps, whilst maintaining a longer time step for the lower-frequency modes. To achieve this separation, we separate the force on a particle into two parts:

$$\mathbf{F} = \mathbf{F}_{\text{short}} + \mathbf{F}_{\text{long}}.$$

This division is arbitrary, but for our diatomic molecule we could divide the potential into the short-range interactions that are responsible for the bond vibration and the long-range attractive forces between the atoms. The idea is that on the time scale of the vibrations of the atoms, the long-range part of the potential hardly changes and therefore this “expensive potential” does not need to be updated as often as the “cheap” short-range part of the potential. This suggests using multiple time steps, a short time step for the vibration and a much longer one for the remainder of the interactions.

Martyna et al. [126] used the Liouville formalism to solve the equations of motion using multiple time steps. In our discussion of the approach, we consider the *NVE* ensemble. For details on how to use multiple time steps in other ensembles, we refer to [126]. Let us start with the simple case and derive the equations of motion for a single particle with force F . The Liouville operator (iL) for this system is Eq. (4.3.12):

$$\begin{aligned} iL &= iL_r + iL_p \\ &= \mathbf{v} \frac{\partial}{\partial \mathbf{r}} + \frac{\mathbf{F}}{m} \frac{\partial}{\partial \mathbf{v}}. \end{aligned}$$

The equations of motion follow from applying the Trotter formula (4.3.18) with time step Δt :

$$e^{iL\Delta t} \approx e^{iL_p\Delta t/2} e^{iL_r\Delta t} e^{iL_p\Delta t/2}.$$

The position and the velocity at time Δt follow from applying the Liouville operator under the initial condition $(\mathbf{r}(0), \mathbf{v}(0))$. As shown in section (4.3.4), $iL_r t$ corresponds to a shift in coordinates and $iL_p t$ to a shift in momenta. If we perform these operations in three steps, we obtain

$$\begin{aligned} & e^{iL\Delta t} f [\dot{\mathbf{r}}(0), \mathbf{r}(0)] \\ &= e^{iL_p\Delta t/2} e^{iL_r\Delta t} e^{iL_p\Delta t/2} f [\dot{\mathbf{r}}(0), \mathbf{r}(0)] \\ &= e^{iL_p\Delta t/2} e^{iL_r\Delta t} f [\dot{\mathbf{r}}(0) + \mathbf{F}(0)\Delta t/2m, \mathbf{r}(0)] \\ &= e^{iL_p\Delta t/2} f [\dot{\mathbf{r}}(0) + \mathbf{F}(0)\Delta t/2m, \mathbf{r}(0) + \dot{\mathbf{r}}(\Delta t/2)\Delta t] \\ &= f [\dot{\mathbf{r}}(0) + \mathbf{F}(0)\Delta t/2m + \mathbf{F}(\Delta t)\Delta t/2m, \mathbf{r}(0) + \dot{\mathbf{r}}(\Delta/2)\Delta t]. \end{aligned}$$

The equations of motion that follow are

$$\begin{aligned} \dot{\mathbf{r}}(\Delta t) &= \dot{\mathbf{r}}(0) + \frac{\Delta t}{2m} [\mathbf{F}(0) + \mathbf{F}(\Delta t)] \\ \mathbf{r}(\Delta t) &= \mathbf{r}(0) + \dot{\mathbf{r}}(\Delta/2)\Delta t, \end{aligned}$$

which the reader will recognize as the velocity Verlet equations (see section 4.3.4).

Example 24 (Multiple time step versus constraints). In this Example, we consider a system of diatomic Lennard-Jones molecules. We compare two models: the first model uses a fixed bond length l_0 between the two atoms of a molecule. In the second model, we use a bond-stretching potential given by

$$U_{\text{bond}}(l) = \frac{1}{2} k_b (l - l_0)^2,$$

where l is the distance between the two atoms in a molecule. In the simulations we used $k_b = 50000$ and $l_0 = 1$. In addition to the bond-stretching potential, all nonbonded atoms interact via a Lennard-Jones potential. The total number of diatomics was 125 and the box length 7.0 (in the usual reduced units). The Lennard-Jones potential was truncated at $r_c = 3.0$, while $T = 3.0$. The equations of motion are solved using bond constraints for the first model, while multiple time steps were used for the second model. All simulations were performed in the *NVE* ensemble.

It is interesting to compare the maximum time steps that can be used to solve the equations of motion for these two methods. As a measure of the accuracy with which the equations of motion are solved, we compute the average deviation of the initial energy, which is defined by Martyna et al.

[623] as

$$E = \frac{1}{N_{\text{step}}} \sum_{i=1}^{N_{\text{step}}} \left| \frac{E(i\Delta t) - E(0)}{E(0)} \right|,$$

in which $E(i)$ is the total energy at time i .

For the bond constraints we use the SHAKE algorithm [606] (see also section 14.1). In the SHAKE algorithm, the bond lengths are exactly fixed at l_0 using an iterative scheme. In Fig. 14.2 the energy fluctuations are shown as a function of the time step. Normally one tolerates a noise level in E of $\mathcal{O}(10^{-5})$, which would correspond to a time step of 2×10^{-4} for the first model. This should be compared with a single-time-step Molecular Dynamics simulation using the second model. A similar energy noise level can be obtained with a time step of 9×10^{-5} , which is a factor 2 smaller.

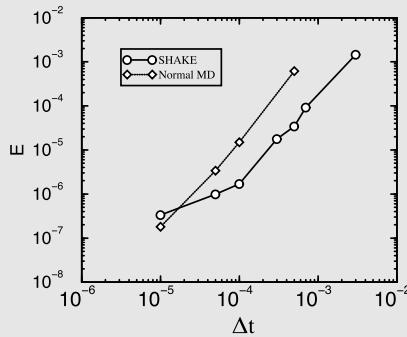


FIGURE 14.2 Comparison of the energy fluctuations as a function of the time step for a normal MD simulation with a harmonic bond potential and a constrained MD simulation with the SHAKE algorithm.

To apply the multiple-time-step algorithm, we have to separate the intermolecular force into a short-range and a long-range part. In the short-range part we include the bond-stretching potential and the short-range part of the Lennard-Jones potential. To make a split in the Lennard-Jones potential, we use a simple switching function $S(r)$:

$$\begin{aligned} U_{LJ}(r) &= U^{\text{short}}(r) + U^{\text{long}}(r) \\ U^{\text{short}}(r) &= S(r) \times U_{LJ}(r) \\ U^{\text{long}}(r) &= [1 - S(r)] U_{LJ}(r), \end{aligned}$$

where

$$S(r) = \begin{cases} 1 & 0 < r < r_c - \lambda \\ 1 + \gamma^2 (2\gamma - 3) & r_c - \lambda < r < r_m \\ 0 & r_m < r < r_c \end{cases}$$

and

$$\gamma = \frac{r - r_m + \lambda}{\lambda}. \quad (14.3.1)$$

In fact, there are other ways to split the total potential function [624,625]. We have chosen $\lambda = 0.3$ and $r_m = 1.7$. To save CPU time a list is made of all the atoms that are close to each other (see Appendix I for details); therefore the calculation of the short-range forces can be done very efficiently. For a noise level of 10^{-5} , one is able to use $\delta t = 10^{-4}$ and $n = 10$, giving $\Delta t = 10^{-3}$.

To compare the different algorithms in a consistent way, we compare in Fig. 14.3 the efficiency of the various techniques. The efficiency η is defined as the length of the simulation (time step times the number of integration steps) divided by the amount of CPU time that was used. In the figure, we have plotted η for all simulations from Fig. 14.2. For an energy noise level of 10^{-5} , the SHAKE algorithm is twice as efficient than normal MD ($n = 1$). This means that hardly any CPU time is spent in the SHAKE routine. However, the MTS algorithm is still two times faster ($n = 10$, $\delta t = 10^{-4}$) at the same efficiency.

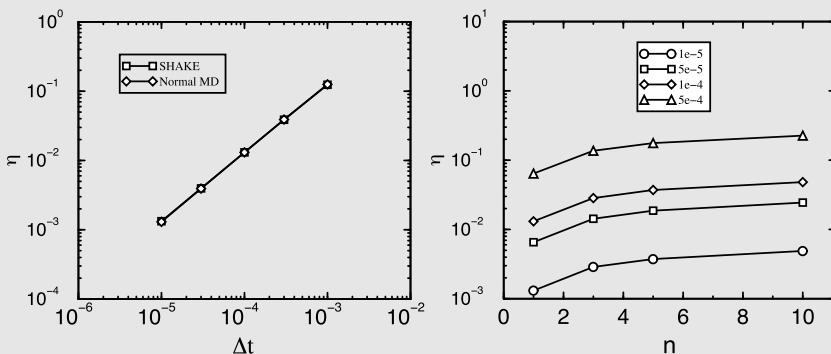


FIGURE 14.3 Comparison of the efficiency η for bond constraints (SHAKE) with normal molecular dynamics (left), and multiple times steps (right). The left figure gives the efficiency as a function of the time step and the right figure as a function of the number of small time steps n , $\Delta t = n\delta t$, where the value of δt is given in the symbol legend.

For more details, see SI (Case Study 22).

Let us now separate the Liouville operator iL_p into two parts:

$$iL_{\text{short}} = \frac{\mathbf{F}_{\text{short}}}{m} \frac{\partial}{\partial \mathbf{v}}$$

$$iL_{\text{long}} = \frac{\mathbf{F} - \mathbf{F}_{\text{short}}}{m} \frac{\partial}{\partial \mathbf{v}} = \frac{\mathbf{F}_{\text{long}}}{m} \frac{\partial}{\partial \mathbf{v}}.$$

We use a Trotter expansion with two time-steps: a long time step, Δt , and a short one, $\delta t = \Delta t/n$. The total Liouville operator then reads

$$e^{iL\Delta t} = e^{i(L_{\text{short}}+L_{\text{long}}+L_r)\Delta t}$$

Algorithm 28 (Multiple-time-step MD)

function multi(f _l ,f _s) vx=vx+0.5*delt*f _l for 1 ≤ it ≤ n do vx=vx+0.5*(delt/n)*f _s x=x+(delt/n)2*vx fs = force_short vx=vx+0.5*(delt/n)*f _s enddo f_l = force_long vx=vx+0.5*delt*f _l end function	input: f _l : long-range part of the force f _s : short-range part of the force velocity Verlet with time step Δt/2 loop for the short time steps velocity Verlet with short timestep Δt/n short-range forces all long-ranged forces velocity Verlet with time step Δt/2
--	---

Specific Comments (for general comments, see p. 7)

1. In the argument list of function call we have added f_l, f_s to indicate that in the velocity Verlet algorithm the force is remembered from the previous time step.
2. Function **force_short** determines the short-range forces. Since this involves a small number of particles, the calculation of these forces is much faster than **force_long** in which all interacting particles must be considered.

$$\approx e^{iL_{\text{long}}\Delta t/2} e^{i(L_{\text{short}}+L_r)\Delta t} e^{iL_{\text{long}}\Delta t/2}.$$

We can again apply a Trotter expansion for the terms iL_{long} and iL_r:

$$e^{iL\Delta t} = e^{iL_{\text{long}}\Delta t/2} \left[e^{iL_{\text{short}}\delta t/2n} e^{iL_r\delta t/n} e^{iL_{\text{short}}\delta t/2n} \right]^n e^{iL_{\text{long}}\Delta t/2}.$$

We apply this Liouville operator to the initial position and velocity. We first make a step using the expensive \mathbf{F}_{long}

$$e^{iL_{\text{long}}\Delta t/2} f [\dot{\mathbf{r}}(0), \mathbf{r}(0)] = f [\dot{\mathbf{r}}(0) + \mathbf{F}_{\text{long}}(0)\Delta t/2m, \mathbf{r}(0)],$$

followed by n small steps using the cheap $\mathbf{F}_{\text{short}}$ with the smaller time step, δt , or

$$\left[e^{iL_{\text{short}}\delta t/2n} e^{iL_r\delta t/n} e^{iL_{\text{short}}\delta t/2n} \right]^n f [\dot{\mathbf{r}}(0) + \mathbf{F}_{\text{long}}(0)\Delta t/2m, \mathbf{r}(0)],$$

and finally one more time-step of length Δt/2 with the expensive \mathbf{F}_{long} .

The result corresponds to solving the equations of motion using the velocity Verlet scheme using the force $\mathbf{F}_{\text{short}}$ with time step δt and initial conditions $\dot{\mathbf{r}}(0) + \mathbf{F}_{\text{long}}(0)\Delta t/2m, \mathbf{r}(0)$. By construction, this algorithm is time reversible.

In Algorithm 28 we illustrate how this **Multiple-Time-Step (MTS)** can be implemented.

Two applications of this algorithm are particularly important. One is the use of **MTS** algorithms to simulate the dynamics of molecules with stiff internal bonds. In Example 24 it is shown that this application of the **MTS** method is attractive, because it is competitive with constrained dynamics (see section 14.1), at least for the case that we considered. The second important area of application is as a time-saving device in the simulation of systems with computationally “expensive” potential-energy functions. Here the **MTS** method offers the possibility of carrying out many time steps with a “cheap” potential energy (e.g., an effective pair potential) and then performing the expensive correction every n th step. Procacci and Marchi have used this approach to reduce the computational costs associated with the long-range interactions of Coulombic systems [624,625], using **MTS** MD in combination with the Ewald summation (see Chapter 11) to reduce the CPU time for the calculation of long-range interactions.

Chapter 15

Rare events

Molecular Dynamics simulations can be used to probe the natural time evolution of classical many-body systems, typically on a time scale of 10^{-14} to 10^{-7} s: the upper limit depends on the computing power at our disposal and can even be several orders of magnitude higher, but then the computational cost becomes very high. The typical time window of 10^{-14} to 10^{-7} is adequate for studying many structural and dynamical properties, provided that the relevant fluctuations decay on a time scale that is appreciably shorter than 10^{-7} s. This is true for most equilibrium properties of simple liquids. It is also usually true for the dynamics associated with non-hydrodynamic modes. For hydrodynamic modes (typically, the modes that describe the diffusion or propagation of quantities that satisfy a conservation law, such as mass, momentum, or energy), the time scales can be much longer. But we still can use MD simulations to compute the transport coefficients that govern the hydrodynamic behavior by making use of the appropriate Green-Kubo relation. As explained in section 2.5.2, Green-Kubo relations allow us to express the hydrodynamic transport coefficients in terms of a time integral of a correlation function of a dynamical quantity that fluctuates on a microscopic time scale: for instance, the self-diffusion coefficient is equal to the integral of the velocity autocorrelation function.

Nevertheless, there are many dynamical phenomena that cannot be studied in this way. In this Chapter, we discuss one particularly important example, namely activated processes. Conventional MD simulations cannot be used to study activated processes. The reason is not that the relevant dynamics is slow, but rather that rare events happen infrequently, but when a rare event does take place, it usually happens quite quickly, i.e., on a time scale that can be followed by MD simulation. An example is the trans-to-gauche transition in an alkane: this process is infrequent if the barrier separating the two conformations is large compared to $k_B T$. Yet, once an unlikely fluctuation has driven the system to the top of the barrier, the actual barrier crossing is quick.

It turns out that in many cases, MD simulations can be used to compute the rate of such activated processes. Such calculations were first performed by Bennett in the context of diffusion in solids [626]. Subsequently, Chandler extended and generalized the approach to the calculation of reaction rates [58,627]. The basic idea behind the “Bennett-Chandler”-style MD simulations of rare events is that the rate at which a barrier crossing proceeds is determined by the product of a static term, namely the probability of finding the system at the top of the

barrier, and a dynamic term that describes the rate at which systems at the top of the barrier move to the other valley.

Since the mid-90s, techniques for computing the rate of rare events have undergone an explosive development: the number of papers on rare-event simulation techniques has increased by two orders of magnitude since the 1990s, and entire books have been written on the subject [34].

The aim of this chapter on Rare Events is to highlight the ideas behind the main developments, rather than to list them all. In this spirit, we pay attention to the basic physics of rare events, and we give simple examples of some of the rare-event techniques, at least those that have been successfully applied to complex problems. We add this qualification because some rare-event techniques that have been proposed in the literature have only been tested on barrier crossings in low-dimensional energy landscapes. Our focus on a set of representative techniques comes at a price: even less than in the rest of this book can we aim for completeness.

15.1 Theoretical background

By *Rare Events* we mean processes that happen on a timescale that is much longer than the natural timescale of the underlying dynamics. For instance, rare events in a Lennard-Jones system happen on timescales much longer than $t^* \equiv \sigma \sqrt{m/\epsilon}$. Unlike the decay of hydrodynamic modes, rare events are not simply slow: they are just infrequent, but when they happen they proceed rapidly. It is this separation of timescales that makes the numerical simulation of rare events challenging.

As a prototypical example of a physical phenomenon proceeding through rare events, we consider a unimolecular reaction $A \rightleftharpoons B$, in which species A is transformed into species B . If the rate-limiting step of this reaction is a (classical) barrier crossing, then Molecular Dynamics simulations can be used to compute the rate constant of such a reaction: Chandler's 1978 paper [627] explains under what conditions the use of rare-event techniques is justified. If the rate-limiting step is a tunneling event or the hopping from one potential-energy surface to another, the classical approach breaks down, and we should turn to quantum dynamical schemes that fall outside the scope of this book (see [48,628]).

Let us first look at the phenomenological description of unimolecular reactions. We denote the number density of species A and B by c_A and c_B , respectively. The phenomenological rate equations are

$$\frac{dc_A(t)}{dt} = -k_{A \rightarrow B} c_A(t) + k_{B \rightarrow A} c_B(t) \quad (15.1.1)$$

$$\frac{dc_B(t)}{dt} = +k_{A \rightarrow B} c_A(t) - k_{B \rightarrow A} c_B(t). \quad (15.1.2)$$

Clearly, as the number of molecules is constant in this conversion reaction, the total number density is conserved:

$$\frac{d[c_A(t) + c_B(t)]}{dt} = 0. \quad (15.1.3)$$

In equilibrium, all concentrations are time-independent, i.e., $\dot{c}_A = \dot{c}_B = 0$. This implies that

$$K \equiv \frac{\langle c_A \rangle}{\langle c_B \rangle} = \frac{k_{B \rightarrow A}}{k_{A \rightarrow B}}, \quad (15.1.4)$$

where K is the equilibrium constant of the reaction. Let us now consider what happens if we take a system at equilibrium, and apply a small perturbation, Δc_A , to the concentration of species A (and thereby of species B). We can write the rate equation that determines the decay of this perturbation as

$$\frac{d\Delta c_A(t)}{dt} = -k_{A \rightarrow B}\Delta c_A(t) - k_{B \rightarrow A}\Delta c_A(t),$$

where we have used Eqs. (15.1.3) and (15.1.4). The solution to this equation is

$$\Delta c_A(t) = \Delta c_A(0) \exp[-(k_{A \rightarrow B} + k_{B \rightarrow A})t] \equiv \Delta c_A(0) \exp(-t/\tau_R), \quad (15.1.5)$$

where we have defined the reaction time constant

$$\tau_R = (k_{B \rightarrow A} + k_{A \rightarrow B})^{-1} = k_{A \rightarrow B}^{-1} \left(1 + \frac{\langle c_A \rangle}{\langle c_B \rangle}\right)^{-1} = \frac{c_B}{k_{A \rightarrow B}}, \quad (15.1.6)$$

where we have assumed that the total concentration $c_A + c_B = 1$. With this normalization, c_A is simply the probability that a given molecule is in state A .

Thus far, we have discussed the reaction from a macroscopic, phenomenological point of view. Let us now look at the microscopics. We do this in the framework of linear response theory. First of all, we must have a microscopic description of the reaction. This means that we need a recipe that allows us to measure how far the reaction has progressed. In the case of diffusion over a barrier from one free energy minimum to another, we could use the fraction of the distance traveled as a reaction coordinate. In general, reaction coordinates may be complicated, nonlinear functions of the coordinates of all particles. It is convenient to think of the reaction coordinate q simply as a generalized coordinate of the type discussed in Chapter 14. In Fig. 15.1, we show a schematic drawing of the free energy surface of the system as a function of the reaction coordinate q . If we wish to change the equilibrium concentration of species A , we should apply an external perturbation that favors all states with $q < q^*$ relative to those with $q > q^*$.

By analogy to the discussion in section 2.5, we consider an external perturbation that changes the relative probabilities of finding species A and B . To

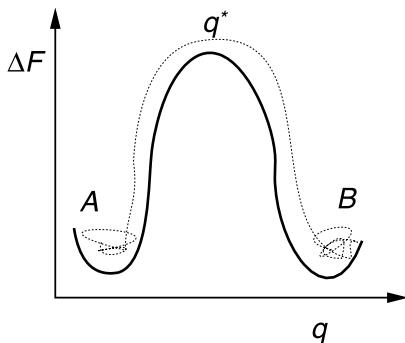


FIGURE 15.1 Schematic drawing of the free energy surface of a many-body system, as a function of the reaction coordinate q . For $q < q^*$, we have the reactant species A , for $q > q^*$, we have the product B . As will be discussed below, the choice of q^* is, to some extent, arbitrary. However, it is convenient to identify the value of the reaction coordinate at the top of the barrier with q^* .

achieve this, we add to the Hamiltonian a term that lowers the potential energy for $q < q^*$:

$$\mathcal{H} = \mathcal{H}_0 - \epsilon g_A(q - q^*), \quad (15.1.7)$$

where ϵ is a parameter that measures the strength of the perturbation. As we are interested in the linear response, we shall consider the limit $\epsilon \rightarrow 0$. The function $g_A(q - q^*)$ is chosen such that it is equal to 1 if the reaction coordinate q is in the range that corresponds to an equilibrium configuration of the “reactant,” while $g_A(q - q^*)$ should be equal to 0 for a typical “product” configuration. The traditional choice for g_A is a Heaviside θ -function:

$$g_A(q - q^*) = 1 - \theta(q - q^*) = \theta(q^* - q),$$

where $\theta(x) = 1$ for $x > 0$ and $\theta(x) = 0$ otherwise. In what follows, we shall consider the more general case that g_A is equal to the θ -function in the reactant and product domains. However, unlike θ , g_A varies smoothly from 1 to 0 in the region of the free energy barrier. For the sake of simplicity, we refer to the states A and B as “reactants” and “products” in the chemical sense of the word. However, in general, A and B can designate any pair of initial and final states that can interconvert by a barrier-crossing process.

Let us first consider the effect of a static perturbation of this type on the probability of finding the system in state A . We note that

$$\Delta c_A = \langle c_A \rangle_\epsilon - \langle c_A \rangle_0 = \langle g_A \rangle_\epsilon - \langle g_A \rangle_0.$$

Here we have used the fact that g_A is equal to 1 in the reactant basin. Hence, the average value of g_A is simply equal to the probability of finding the system in

state A . From Eq. (2.5.3) of section 2.5.1, we find immediately that

$$\frac{\partial \Delta c_A}{\partial \epsilon} = \beta \left(\langle g_A^2 \rangle_0 - \langle g_A \rangle_0^2 \right).$$

This equation can be simplified by noting that, outside the barrier region, g_A is either 1 or 0, and hence, $g_A^2(x) = g_A(x)$. In the barrier region, this equality need not hold—but those configurations hardly contribute to the equilibrium average. Hence,

$$\frac{\partial \Delta c_A}{\partial \epsilon} = \beta \left[\langle g_A \rangle_0 (1 - \langle g_A \rangle_0) \right] = \beta \langle c_A \rangle \langle c_B \rangle. \quad (15.1.8)$$

For what follows, it is convenient to define the function $g_B = 1 - g_A$. Clearly,

$$\langle g_B \rangle_0 = \langle (1 - g_A) \rangle_0 = \langle c_B \rangle_0.$$

Next, consider what happens if we suddenly switch off the perturbation at time $t = 0$. The concentration of A will relax to its equilibrium value as described in Eq. (2.5.8) and we find that, to first order in ϵ ,

$$\begin{aligned} \Delta c_A(t) &= \beta \epsilon \frac{\int d\Gamma \exp(-\beta \mathcal{H}_0) (g_A(0) - \langle g_A \rangle) \exp(i L_0 t) (g_A(0) - \langle g_A \rangle)}{\int d\Gamma \exp(-\beta \mathcal{H}_0)} \\ &= \beta \epsilon \langle \Delta g_A(0) \Delta g_A(t) \rangle. \end{aligned} \quad (15.1.9)$$

Finally, we can use Eq. (15.1.8) to eliminate ϵ from the above equation, and we find the following expression for the relaxation of an initial perturbation in the concentration of species A :

$$\Delta c_A(t) = \Delta c_A(0) \frac{\langle \Delta g_A(0) \Delta g_A(t) \rangle}{\langle c_A \rangle \langle c_B \rangle}. \quad (15.1.10)$$

If we compare this with the phenomenological expression, Eq. (15.1.5), we see that

$$\exp(-t/\tau_R) = \frac{\langle \Delta g_A(0) \Delta g_A(t) \rangle}{\langle c_A \rangle \langle c_B \rangle}. \quad (15.1.11)$$

Actually, we should be cautious with this identification. For very short times (i.e., times comparable to the average time that the system spends in the region of the barrier), we should not expect the autocorrelation function of the concentration fluctuations to decay exponentially. Only at times that are long compared to the typical barrier-crossing time should we expect Eq. (15.1.11) to hold. Let us assume that we are in this regime. Then we can obtain an expression for τ_R by differentiating Eq. (15.1.11):

$$-\tau_R^{-1} \exp(-t/\tau_R) = \frac{\langle g_A(0) \dot{g}_A(t) \rangle}{\langle c_A \rangle \langle c_B \rangle} = -\frac{\langle \dot{g}_A(0) g_A(t) \rangle}{\langle c_A \rangle \langle c_B \rangle}, \quad (15.1.12)$$

where we have dropped the Δ 's, because the time derivative of the equilibrium concentration vanishes. Hence, for times that are long compared to molecular times, but still very much shorter than τ_R , we can write

$$\tau_R^{-1} = \frac{\langle \dot{g}_A(0)g_A(t) \rangle}{\langle c_A \rangle \langle c_B \rangle} \quad (15.1.13)$$

or, if we recall Eq. (15.1.6) for the relation between $k_{A \rightarrow B}$ and τ_R , we find

$$k_{A \rightarrow B}(t) = \frac{\langle \dot{g}_A(0)g_A(t) \rangle}{\langle c_A \rangle}. \quad (15.1.14)$$

In this equation, the time dependence of $k_{A \rightarrow B}(t)$ is indicated explicitly. However, we recall that it is only the long-time plateau value of $k_{B \rightarrow A}(t)$ that enters into the phenomenological rate equation. Finally, we can re-express the correlation function in Eq. (15.1.14) by noting that

$$\dot{g}_A(q - q^*) = \dot{q} \frac{\partial g_A(q - q^*)}{\partial q} = -\dot{q} \frac{\partial g_B(q - q^*)}{\partial q} = -\dot{q}(\partial_q g_B),$$

where we use the notation $\partial_q \equiv \frac{\partial}{\partial q}$. Then:

$$k_{A \rightarrow B}(t) = \frac{\langle \dot{q}(\partial_q g_B)(0)g_B(t) \rangle}{\langle c_A \rangle}, \quad (15.1.15)$$

where we have used the fact that the equilibrium average $\langle \dot{q} \rangle$ is equal to zero. A particularly convenient form of Eq. (15.1.15) that we shall use in section 15.3 is

$$k_{A \rightarrow B}(t) = \int_0^\infty dt \frac{\langle \dot{q}(0)(\partial_q)g_B(0)\dot{q}(t)(\partial_q g_B)(t) \rangle}{\langle c_A \rangle}. \quad (15.1.16)$$

But first, we establish contact with the conventional “Bennett-Chandler” expression for the rate constant.

15.2 Bennett-Chandler approach

If we choose $g_A = \theta(q^* - q)$, and hence $g_B = \theta(q - q^*)$, then we can rewrite Eq. (15.1.15) in the following way:

$$\begin{aligned} k_{A \rightarrow B}(t) &= \frac{\langle \dot{q}\delta(q(0) - q^*)\theta(q(t) - q^*) \rangle}{\langle c_A \rangle} \\ &= \frac{\langle \dot{q}\delta(q^* - q(0))\theta(q(t) - q^*) \rangle}{\langle \theta(q^* - q) \rangle}. \end{aligned} \quad (15.2.1)$$

In this way, we have expressed the rate constant $k_{A \rightarrow B}$ exclusively in microscopic quantities that can be measured in a simulation. Next, we shall see

how this can be done. First, however, we establish the connection between Eq. (15.2.1) and the expression for the rate constant that follows from Eyring's **Transition State Theory (TST)**. To this end, consider $k_{A \rightarrow B}(t)$ in the limit $t \rightarrow 0+$:

$$\begin{aligned} \lim_{t \rightarrow 0+} k_{A \rightarrow B}(t) &= \frac{\langle \dot{q} \theta(q(0+) - q^*) \delta(q^* - q(0)) \rangle}{\langle \theta(q^* - q) \rangle} \\ &= \frac{\langle \dot{q} \theta(\dot{q}) \delta(q^* - q(0)) \rangle}{\langle \theta(q^* - q) \rangle}, \end{aligned} \quad (15.2.2)$$

where we have used the fact that $\theta(q(0+) - q^*) = 1$, if $\dot{q} > 0$, and 0 otherwise. In other words $\theta(q(0+) - q^*) = \theta(\dot{q})$. The expression on the last line of Eq. (15.2.2) is the classical transition state theory prediction for the rate constant, $k_{A \rightarrow B}^{TST}$.

It is useful to rewrite Eq. (15.2.1) as

$$k_{A \rightarrow B} = \frac{\langle \dot{q}(0) \delta(q^* - q(0)) \theta(q(t) - q^*) \rangle}{\langle \delta(q^* - q(0)) \rangle} \times \frac{\langle \delta(q^* - q) \rangle}{\langle \theta(q^* - q) \rangle}. \quad (15.2.3)$$

The first part on the right-hand side of Eq. (15.2.3) is a conditional average, namely the average of the product $\dot{q}(0)\theta(q(t) - q^*)$, given that $q(0) = q^*$. The second part expresses the probability density of finding the system at the top of the barrier, divided by the probability that the system is on the reactant side of the barrier. We denote this density with $P(q^*)$, where $P(q)$ is defined as

$$P(q) \equiv \frac{\langle \delta(q - q(\mathcal{X})) \rangle}{\langle \theta(q^* - q(\mathcal{X})) \rangle} = \frac{\int d\mathcal{X} \exp(-\beta U) \delta(q - q(\mathcal{X}))}{\int d\mathcal{X} \exp(-\beta U) \theta(q^* - q(\mathcal{X}))}. \quad (15.2.4)$$

It is inadvisable to use brute-force simulations to compute either term in Eq. (15.2.3). The reason is that if we study events that are rare, normal simulations will barely sample the top of the barrier, and would yield poor statistics on $k_{A \rightarrow B}$. The solution to this problem is either to use biased simulations of the type discussed in Chapter 8, or to use holonomic constraints as discussed in Chapter 14. As the two approaches are rather different, we discuss them separately.

Conceptually the easiest approach is to use umbrella sampling to compute first the probability distribution at the top of the barrier. In such a simulation, we would construct $P(q)$ over the entire range of q -values $q < q^*$, using multiple umbrella-sampling simulations (section 8.6.6) and a histogram reconstruction method, such as **MBAR** (see section 8.6.11). To probe $P(q)$ near the top of the barrier, we could use a narrow biasing potential well. This part of the simulation can be done with both MC and MD.

To compute the first term in Eq. (15.2.3), which is a conditional average, we would use configurations obtained from the umbrella sampling near the top of the barrier, to initiate MD simulations. In that case, we would generate the

velocities of the particles in the system according to a Maxwell distribution. With these initial velocities and positions, we run MD simulations to sample $\dot{q}(0)\theta(q(t) - q^*)$. Of course, the umbrella sampling will generate points very close to, but not exactly at q^* . In practice, this is rarely a problem.

15.2.1 Dealing with holonomic constraints (Blue-Moon ensemble)

The alternative to biased sampling is to compute the conditional average of $\dot{q}(0)\theta(q(t) - q^*)$ using a holonomic constraint $q(0) - q^* = 0$ for the initial conditions. With this constraint, we would then equilibrate the system using MD, and then use the points on this MD trajectory to start a large number of runs in which the system is allowed to evolve without any constraints.

However, when using holonomic constraints at $q = q^*$, we must be careful because in Chapter 14 we argued that holonomic constraints can change the equilibrium distribution:

$$\rho_{\text{unconstrained}}(\mathbf{q}) = |\mathcal{H}|^{-\frac{1}{2}} \rho_{\text{constrained}}(\mathbf{q}) \quad (15.2.5)$$

with

$$H_{\alpha\beta} = \sum_{i=1}^N m_i^{-1} \frac{\partial\sigma_\alpha}{\partial\mathbf{r}_i} \frac{\partial\sigma_\beta}{\partial\mathbf{r}_i}.$$

In the present case, we have one constraint.¹ Our constraint is

$$\sigma \equiv q^* - q = 0.$$

If q is a linear function of the Cartesian coordinates, there is no need to worry about the effect of the constraints on the distribution function, because $|\mathcal{H}|$ is a constant. However, in general, q is a nonlinear function of all other coordinates, and we should consider the effect of $|\mathcal{H}|$ on $\rho(\mathbf{q})$.

The hard constraint(s) will bias the initial distribution function:

$$\frac{\langle \dot{q}(0)\delta(q^* - q(0))\theta(q(t) - q^*) \rangle}{\langle \delta(q^* - q(0)) \rangle} = \frac{\left\langle |\mathcal{H}|^{-\frac{1}{2}} \dot{q}(0)\theta(q(t) - q^*) \right\rangle_c}{\left\langle |\mathcal{H}|^{-\frac{1}{2}} \right\rangle_c}, \quad (15.2.6)$$

where the subscript c indicates a constrained average over initial configurations.

Below, we derive an explicit expression for the variation of $P(q)$ with q , using constrained MD to reconstruct the complete $P(q)$ between the bottom and the top of the barrier, and in this way we can compute $P(q^*)$. In fact, rather than

¹ We assume, for the moment, that there are no other constraints in the system. For a discussion of the latter case, see e.g., ref. [609].

computing the derivative of $P(q)$ with respect to q , we differentiate $\ln P(q)$:

$$\frac{\partial \ln P(q)}{\partial q} = \frac{\int d\mathcal{X} \exp(-\beta\mathcal{U}) \delta(q - q(\mathcal{X}))/\partial q}{\int d\mathcal{X} \exp(-\beta\mathcal{U}) \delta(q - q(\mathcal{X}))}. \quad (15.2.7)$$

We can re-express the integral in the numerator by partial integration. To do this, we should first transform from the Cartesian coordinates \mathcal{X} to a set of generalized coordinates $\{\mathbf{Q}, q\}$ that includes the reaction coordinate q . We denote the Jacobian of the transformation from \mathcal{X} to $\{\mathbf{Q}, q\}$ by $|J|$. Now we carry out the partial integration

$$\begin{aligned} \frac{\partial \ln P(q)}{\partial q} &= \frac{\int d\mathbf{Q} dq |J| \exp(-\beta\mathcal{U}) \delta(q - q(\mathcal{X}))/\partial q}{\int d\mathcal{X} \exp(-\beta\mathcal{U}) \delta(q - q(\mathcal{X}))} \\ &= \frac{\int d\mathbf{Q} dq [\partial |J| \exp(-\beta\mathcal{U})/\partial q(\mathcal{X})] \delta(q - q(\mathcal{X}))}{\int d\mathcal{X} \exp(-\beta\mathcal{U}) \delta(q - q(\mathcal{X}))} \\ &= \frac{\int d\mathcal{X} [\partial(\ln(|J|) - \beta\mathcal{U})/\partial q(\mathcal{X})] \exp[-\beta\mathcal{U}] \delta(q - q(\mathcal{X}))}{\int d\mathcal{X} \exp(-\beta\mathcal{U}) \delta(q - q(\mathcal{X}))} \\ &= \frac{\langle [\partial(\ln(|J|) - \beta\mathcal{U})/\partial q(\mathcal{X})] \delta(q - q(\mathcal{X})) \rangle}{\langle \delta(q - q(\mathcal{X})) \rangle}, \end{aligned} \quad (15.2.8)$$

where, in the third line, we have transformed back to the original Cartesian coordinates. It should be noted that the computation of the Jacobian $|J|$ can be greatly simplified [629].

As the averages both in the numerator and in the denominator contain $\delta(q - q(\mathcal{X}))$, it is natural to express Eq. (15.2.8) in terms of constrained averages that can be computed conveniently in a constrained Molecular Dynamics simulation. Just as in Eq. (15.2.6), we must correct for the bias introduced by the hard constraint:

$$\frac{\partial \ln P(q)}{\partial q} = \frac{\left\langle |H|^{-\frac{1}{2}} \partial(\ln(|J|) - \beta\mathcal{U})/\partial q \right\rangle_c}{\left\langle |H|^{-\frac{1}{2}} \right\rangle_c}, \quad (15.2.9)$$

where the subscript c denotes averaging in an ensemble where $q(\mathcal{X})$ is constrained to be equal to q . If we integrate Eq. (15.2.9) from the bottom to the top of the barrier, we get

$$\ln \left(\frac{P(q^*)}{P(q = q_A)} \right) = \int_{q_A}^{q^*} dq \frac{\left\langle |H|^{-\frac{1}{2}} \partial(\ln(|J|) - \beta\mathcal{U})/\partial q \right\rangle_c}{\left\langle |H|^{-\frac{1}{2}} \right\rangle_c}. \quad (15.2.10)$$

In practice, this integration has to be carried out numerically.

By combining Eqs. (15.2.6), (15.2.4), and (15.2.10), we finally have an expression for the rate constant $k_{A \rightarrow B}$ that can be computed numerically.

It should be noted that, in the above expression, we have assumed that the reaction coordinate is the only quantity that will be constrained in the simulation. If there are more constraints, e.g., if we simulate a reaction in a polyatomic fluid, then the expression for $k_{A \rightarrow B}$ becomes a bit more complicated (see the article by Ciccotti [609], and references therein). The Blue-Moon technique can be painful to implement when the constraints are complicated many-body functions. In such cases, Automatic Differentiation (see e.g., [108]) may be an attractive option.

Bennett-Chandler epilogue

It is clear that the Bennett-Chandler expressions for systems with holonomic constraints are not particularly simple. The question is then: which technique to use when?

When writing a program from scratch, using holonomic constraints would be heroic, but the method is not better than a much simpler simulation that uses umbrella sampling or a similar technique. However, few programs are written from scratch. Moreover, there are many more MD codes available than MC codes. Hence, chances are that using the Blue-Moon ensemble in an existing MD code is just a matter of setting a parameter in the input file. In that case, the previous section should help the user understand what the Blue-Moon option does.

Irrespective of the method used, the expressions derived above for a unimolecular rate constant are not limited to chemical reactions: the same approach can be used to study any activated classical process, such as diffusion in solids, crystal nucleation, or transport through membranes.

Example 25 (Ideal gas particle over a barrier). To illustrate the “Bennett-Chandler” approach for calculating crossing rates, we consider an ideal gas particle moving in an external field. This particle is constrained to move on the dimensional potential surface shown in Fig. 15.2. This example is rather unphysical because the moving particle cannot dissipate its energy. As a consequence, the motion of the particle is purely ballistic. We assume that, far away on either side of the barrier, the particle can exchange energy with a thermal reservoir. Transition state theory predicts a crossing rate given by Eq. (15.2.2):

$$k_{A \rightarrow B}^{\text{TST}} = \frac{1}{2} |\dot{q}| \frac{\exp[-\beta u(q^*)]}{\int_{-\infty}^{q^*} dq \exp[-\beta u(q)]} = \sqrt{\frac{k_B T}{2\pi m}} \frac{\exp[-\beta u(q^*)]}{\int_{-\infty}^{q^*} dq \exp[-\beta u(q)]}. \quad (15.2.11)$$

If we choose the dividing surface q_1 (see Fig. 15.2) at the top of the barrier ($q_1 = q^*$) none of the particles that start off with a positive velocity will return to the reactant state. Hence, there is no recrossing of the barrier and transition state theory is exact for this system.

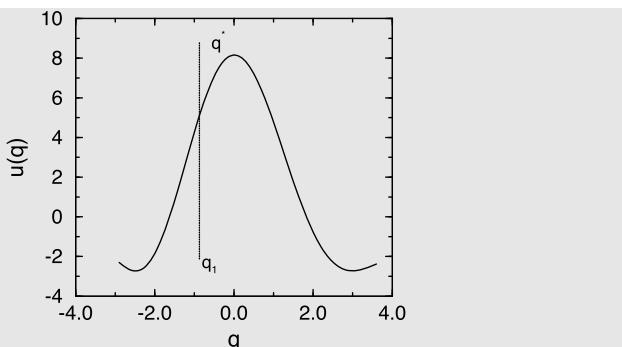


FIGURE 15.2 Potential-energy barrier for an ideal gas particle; if the particle has a position to the left of the dividing surface q_1 the particle is in state A (reactant). The region to the right of the barrier is designated as product B . The top of the barrier is denoted by q^* ($q^* = 0$).

Note that transition state theory (Eq. (15.2.11)) predicts a rate constant that depends on the location of the dividing surface. In contrast, the Bennett-Chandler expression for the crossing rate is independent of location of the dividing surface (as it should be). To see this, consider the situation that the dividing surface is chosen to be the left of the top of the barrier (i.e., at $q_1 < q^*$). The calculation of the crossing rate according to Eq. (15.2.3) proceeds in two steps. First we calculate the relative probability of finding a particle at the dividing surface. And then we need to compute the probability that a particle that starts with an initial velocity \dot{q} from this dividing surface will, in fact, cross the barrier. The advantage of the present example is that this probability can be computed explicitly. According to Eq. (15.2.4), the relative probability of finding a particle at q_1 is given by

$$\frac{\langle \delta(q - q_1) \rangle}{\langle \theta(q_1 - q) \rangle} = \frac{\exp[-\beta u(q_1)]}{\int_{-\infty}^{q_1} dq \exp[-\beta u(q)]}.$$

If the dividing surface is not at the top of the barrier, then the probability of finding a particle will be higher at q_1 than at q^* , but the fraction of the number of particles that actually cross the barrier will be less than predicted by transition state theory. It is convenient to introduce the time-dependent transmission coefficient $\kappa(t)$, defined as the ratio

$$\kappa(t) \equiv \frac{k_{A \rightarrow B}(t)}{k_{A \rightarrow B}^{\text{TST}}} = \frac{\langle \dot{q}(0) \delta(q(0) - q_1) \theta(q(t) - q_1) \rangle}{0.5 \langle |\dot{q}(0)| \rangle}.$$

The behavior of $\kappa(t)$ is shown in Fig. 15.3 for various choices of q_1 . The figure shows that for $t \rightarrow 0$ $\kappa(t) = 1$, and that for different values of q_1 we get different plateau values. The reason $\kappa(t)$ decays from its initial value is that particles that start off with too little kinetic energy cannot cross the barrier and recross the dividing surface (q_1). The plateau value of $\kappa(t)$ provides us with the correction that has to be applied to the crossing rate predicted by

transition state theory. Hence, we see that as we change q_1 , the probability of finding a particle at q_1 goes up, and the transmission coefficient goes down. But, as can be seen from Fig. 15.3, the actual crossing rate (which is proportional to the product of these two terms) is independent of q_1 , as it should be. Now consider the case that $q_1 > q^*$. In that case, all particles starting with positive \dot{q} will continue to the product side. But now there is also a fraction of the particles with negative \dot{q} that will proceed to the product side. These events will give a negative contribution to κ . And the net result is that the transmission coefficient will again be less than predicted by transition state theory. Hence, the important thing is not if a trajectory ends up on the product side, but if it starts on the reactant side and proceeds to the product side. In a simulation, it is therefore convenient always to compute trajectories in pairs: for every trajectory starting from a given initial configuration with a velocity \dot{q} , we also compute the time-reversed trajectory, i.e., the one starting from the same configuration with a velocity $-\dot{q}$. If both trajectories end up on the same side of the barrier then their total contribution to the transmission coefficient is clearly zero. Only if the forward and time-reversed trajectories end up on different sides of the barrier, do we get a contribution to κ . In the present (ballistic) case, this contribution is always positive. But in general, this contribution can also be negative (namely, if the initial velocity at the top of the barrier is not in the direction where the particle ends up).

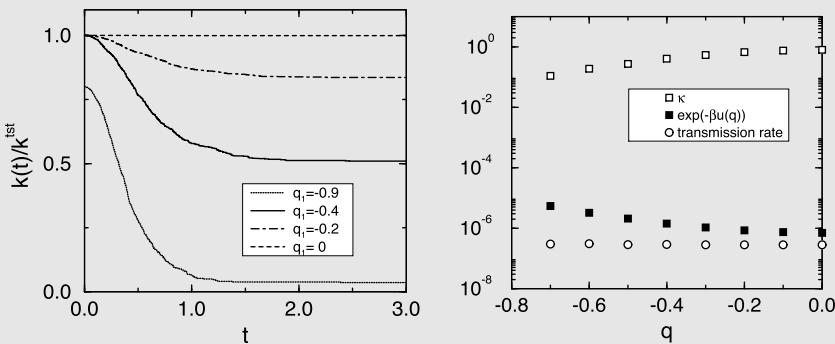


FIGURE 15.3 Barrier recrossing: the left figure gives the transmission coefficient as a function of time for different values of q_1 . The right-hand figure shows, in a single plot, the probability density of finding the system at $q = q_1$ (solid squares), the transmission coefficient κ (open squares), and the overall crossing rate (open circles), all plotted as a function of the location of the dividing surface. Note that the overall crossing rate is independent of the choice of the dividing surface.

We chose this simple ballistic barrier-crossing problem because we can easily show explicitly that the transmission rate is independent^a of the location of q_1 . We start with the observation that the sum of the kinetic and potential energies of a particle that crosses the dividing surface q_1 is constant. Only those particles that have sufficient kinetic energy can cross the barrier. We can easily compute the long-time limit of $\langle \dot{q}(0)\theta(q(t) - q_1) \rangle$:

$$\begin{aligned}\langle \dot{q}(0)\theta(q(\infty) - q_1) \rangle &= \sqrt{\frac{m\beta}{2\pi}} \int_{v_\epsilon}^{\infty} dv v \exp(-\beta mv^2/2) \\ &= \sqrt{\frac{1}{2\pi m\beta}} \exp\left(-\frac{1}{2}\beta mv_\epsilon^2\right),\end{aligned}$$

where v_ϵ is the minimum velocity needed to achieve a successful crossing. v_ϵ is given by

$$\frac{1}{2}mv_\epsilon^2 + u(q_1) = u(q^*).$$

It then follows that

$$\langle \dot{q}(0)\theta(q(\infty) - q_1) \rangle = \sqrt{\frac{1}{2\pi m\beta}} \exp\{-\beta[u(q^*) - u(q_1)]\}.$$

This term exactly compensates the Boltzmann factor, $\exp(-\beta u(q_1))$, associated with the probability of finding a particle at q_1 . Hence, we have shown that the overall crossing rate is given by Eq. (15.2.11), independent of the choice of q_1 .

The reader may wonder why it is important to have an expression for the rate constant that is independent of the precise location of the dividing surface. The reason is that, although it is straightforward to find the top of the barrier in a one-dimensional system, the precise location of the saddle point in a reaction pathway of a many-dimensional system is usually difficult to determine. With the Bennett-Chandler approach it is not necessary to know the exact location of the saddle point. Still, it is worth trying to get a reasonable estimate, as the statistical accuracy of the results is best if the dividing surface is chosen close to the true saddle point.

The nice feature of the Bennett-Chandler expression for barrier-crossing rates is that it allows us to compute rate constants under conditions where barrier recrossings are important, for instance, if the motion over the top of the barrier is more diffusive than ballistic. Examples of such systems are the cyclohexane interconversion in a solvent [631] and the diffusion of nitrogen in an argon crystal [632].

For more details, see SI (Case Study 23).

^a The general proof that the long-time limit of the crossing rate is independent of the location of the dividing surface was given by Miller [630].

15.3 Diffusive barrier crossing

In the previous section, we described the Bennett-Chandler expression for the rate of activated processes. This expression is widely used in numerical simulation. However, even though the expression is correct for arbitrary barrier crossings, provided that the barrier is much larger than $k_B T$, it is not always computationally efficient. To see this, consider the expression for the transmis-

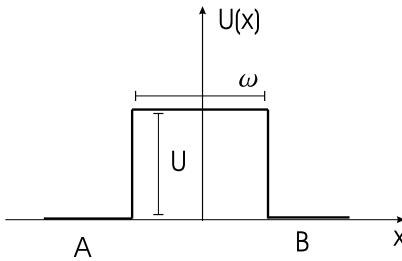


FIGURE 15.4 Simple model for a diffuse barrier crossing: a square barrier of height U and width ω that separates two macroscopic states, A and B .

sion coefficient, κ

$$\kappa(t) \equiv \frac{k_{A \rightarrow B}(t)}{k_{A \rightarrow B}^{TST}} = \frac{\langle \dot{q}(0) \delta(q(0) - q^*) \theta(q(t) - q^*) \rangle}{0.5 \langle |\dot{q}(0)| \rangle}. \quad (15.3.1)$$

Clearly, if $\kappa \rightarrow 1$, we can use **Transition State Theory (TST)** to compute the crossing rate, once we know the barrier height. Hence, the only regime where Eq. (15.3.1) is of interest is when there are appreciable corrections to **TST**, i.e., when $\kappa \ll 1$. However, precisely in this regime, the numerical calculation of κ , using Eq. (15.3.1), is plagued by slow transient behavior and large statistical errors.

To illustrate this point, let us consider a simple example: a square barrier of height U and width ω that separates two (meta)stable states, A and B (see Fig. 15.4). For simplicity, we assume that, in equilibrium, the two states have the same probability, $P_{\text{eq}} \approx 0.5$ (the population of the barrier region is negligible). Moreover, we assume that the motion in the barrier region is diffusive. For this simple geometry, it is easy to write down the diffusion equation. This equation follows from the continuity equation

$$\frac{\partial \rho(q, t)}{\partial t} = -\frac{\partial}{\partial q} J(q, t), \quad (15.3.2)$$

which relates the local density $\rho(q, t)$ at point q and time t to the flux density $J(q, t)$. In addition, we have the constitutive equation for the diffusional flux in an external field

$$J(q, t) = -D \left[\beta \frac{\partial U(q)}{\partial q} \rho(q, t) + \frac{\partial \rho(q, t)}{\partial q} \right], \quad (15.3.3)$$

where $U(q)$ is the external potential. Combining this with the continuity equation, we obtain the Smoluchowski equation [633],

$$\frac{\partial \rho(q, t)}{\partial t} = \frac{\partial}{\partial q} D \left[\beta \frac{\partial U}{\partial q}(q) \rho(q, t) + \frac{\partial \rho(q, t)}{\partial q} \right]$$

$$= \frac{\partial}{\partial q} D e^{-\beta U} \frac{\partial}{\partial q} e^{+\beta U} \rho(q, t), \quad (15.3.4)$$

where D is the diffusion constant of the system. In steady state $\dot{\rho} = 0$, and the diffusional flux J is constant.

We now use the fact that we have assumed a flat barrier, implying that $\partial U/\partial q = 0$ for $-\omega/2 < q < \omega/2$. We also will assume that D is independent of q . It then follows from Eq. (15.3.3) that the probability distribution at the top of the barrier must be a linear function of the reaction coordinate, q :

$$\rho^{\text{st}}(q) = aq + b \quad \text{for } -\omega/2 < q < \omega/2. \quad (15.3.5)$$

The constants a and b are determined from boundary conditions. In equilibrium, $a = 0$ and $b = \rho_{\text{eq}} \exp(-\beta U)$, where ρ_{eq} is the density in states A and B. Suppose we increase the probability density in state A from its equilibrium value by an amount $\rho_{\text{eq}}\delta/2$, and decrease the probability density of state B by the same amount, then the system is no longer in equilibrium. If the barrier is high enough, the resulting flux will be very small and the probabilities of states A and B will barely change with time. In this case, the stationary probability distribution at the top of the barrier is

$$\rho^{\text{st}}(q) = e^{-\beta U} \rho_{\text{eq}} \left[1 - q \frac{\delta}{\omega} \right], \quad (15.3.6)$$

and the flux

$$J^{\text{st}} = D \frac{\rho_{\text{eq}} \delta}{\omega} e^{-\beta U}. \quad (15.3.7)$$

As expected, the flux decreases exponentially with increasing barrier height. The probability density at the top of the barrier is given by Eq. (15.3.6) if, and only if, the flux has reached its stationary value.

Now consider expression for the rate. We rewrite Eq. (15.3.1) as

$$\kappa(t) \equiv \frac{\langle \theta(q^* - q(0)) \dot{q}(t) \delta(q(t) - q^*) \rangle}{0.5 \langle |\dot{q}(0)| \rangle}. \quad (15.3.8)$$

Apart from a constant factor, $\kappa(t)$ is the flux through the transition state, q^* ($= 0$), due to a step function probability profile at $t = 0$. As this step function differs from the linear steady-state profile, the resulting flux will depend on time. We are interested in the plateau value of $\kappa(t)$ after the initial transient regime. The usual assumption is that this transient regime extends over typical “molecular” time scales. However, in the present case it is easy to show that the approach of $\kappa(t)$ to its plateau value can be quite slow. For times $t \ll \omega^2/D$, we can combine Eqs. (15.3.4) and (15.3.2) to yield

$$\frac{\partial J(q, t)}{\partial t} \approx D \frac{\partial^2 J(q, t)}{\partial q^2}, \quad (15.3.9)$$

with the solution

$$J(q, t) \approx \epsilon D e^{-\beta U} \rho_{\text{eq}} \sqrt{\frac{1}{2\pi Dt}} \exp\left[-\frac{(q - q^*)^2}{2Dt}\right]. \quad (15.3.10)$$

We then find that $J(q^*, t)$ decays as $1/\sqrt{t}$ for times $t \ll \omega^2/D$. This means that the approach to the stationary state is very slow. But, more importantly, in the case of diffusive barrier crossings, the transmission coefficient κ is typically quite small. Below, we will give an estimate for κ , but at this stage we just note that small values of κ cannot be determined accurately using Eq. (15.3.8). To see this, consider the expression for the transmission coefficient:

$$\kappa = \frac{2}{\langle |\dot{q}| \rangle_{\text{eq}}} \langle \dot{q}(0) \theta(q(t) - q^*) \rangle_{q(0)=q^*}. \quad (15.3.11)$$

In a computer simulation, we put the system initially at q^* and let it evolve. We then compute $\theta(q(t) - q^*)$ for times that are long enough for Eq. (15.3.8) to have reached a plateau value. We repeat this procedure for n independent trajectories, and then estimate κ as

$$\kappa_{\text{est}} = \frac{2}{n \langle |\dot{q}| \rangle} \sum_{i=1}^n [\dot{q}(0) \theta(q(t) - q^*)]_i. \quad (15.3.12)$$

The statistical error in κ_{est} is given by

$$\sigma_\kappa^2 = \langle (\kappa_{\text{est}} - \langle \kappa \rangle)^2 \rangle. \quad (15.3.13)$$

Taking into account that the trajectories are uncorrelated and assuming that the average in Eq. (15.3.13) can be factorized as if \dot{q} and $\theta(q(t) - q^*)$ were Gaussian variables, we get

$$\sigma_\kappa^2 = \frac{4}{n \langle |\dot{q}| \rangle^2} \langle \dot{q}^2 \rangle \langle \theta^2 \rangle + \frac{1}{n} \kappa^2. \quad (15.3.14)$$

To estimate the variance we make use of the fact that $\langle \theta^2 \rangle \approx 0.5$ and

$$4 \frac{\langle \dot{q}^2 \rangle}{\langle |\dot{q}| \rangle^2} \sim \mathcal{O}(1). \quad (15.3.15)$$

As we consider the case that the transmission coefficient is much less than one small, the second contribution in Eq. (15.3.14) can be ignored. We then obtain,

$$\sigma_\kappa^2 \sim \frac{1}{n}, \quad (15.3.16)$$

and the relative error is

$$\frac{\sigma_\kappa}{\kappa} \sim \frac{1}{\kappa \sqrt{n}}. \quad (15.3.17)$$

This shows that, even for a transmission coefficient as large as 0.1, we would need to follow about 10^4 trajectories in order to get an accuracy of only 10%. The reason why the statistical error is so large is that we use the θ -function to detect transitions from A to B. In a diffusive barrier-crossing process, where re-crossings of the transition state are frequent, the time evolution of this θ -function resembles a random telegraph signal.

The above analysis suggests that the Bennett-Chandler approach becomes inefficient for systems with low transmission coefficients because: 1) the BC scheme prepares the system in a state that is not close to the steady-state situation, and 2) the BC scheme employs the “noisy” θ -function to detect whether the system is in state *B*.

The obvious question is whether we can do better. Below, we show that this is indeed possible. First of all, we shall go back to Eq. (15.1.7) and try to devise a perturbation that prepares the system immediately close to the steady state. Secondly, we shall construct a more continuous “detector” function for measuring the concentration of state B. Below, we shall not discuss the general case, but explain the basic ideas in the context of our simple square-barrier model. We refer the reader to the literature [629] for a more general discussion.

As discussed above, the steady-state probability profile at the top of the barrier is a linear function of the reaction coordinate. Hence, if we set up a perturbation that has this shape, rather than a step function, we would eliminate the problem of the slow, diffusive approach to the steady-state crossing rate. Let us therefore replace the θ -function perturbation by a function $g(q)$ chosen such that $g(q) = \theta(q^* - q)$ outside the barrier region, while inside the barrier region² $g(q) = 1/2 - q/\omega$.

The change in the equilibrium concentration profile due to this perturbation is

$$\Delta\rho(q) = -e^{-\beta U} \rho_{eq} \frac{q\beta\epsilon}{\omega}. \quad (15.3.18)$$

But, with the identification $\delta = \beta\epsilon$, this is precisely the (linear) concentration profile that corresponds to the steady state. Hence, with this perturbation, we have suppressed the initial transient. However, if we still use a θ -function to detect whether the system is in state B, the numerical results will still be noisy. So the second step is to replace the “detector” function for state B by $1 - g(q)$. Note that outside the barrier region $g(q) = \theta(q^* - q)$. Hence, replacing θ with g makes a negligible difference in our estimate of the concentration of B. Let us next consider the effect of this choice of the perturbation g on the statistical accuracy for the transmission coefficient κ . We start from Eq. (15.1.16) for the crossing rate

$$k_{A \rightarrow B} = \int_0^\infty dt \frac{\langle \dot{q}(0)(\partial_q g_B(0))\dot{q}(t)(\partial_q g_B(t)) \rangle}{\langle c_A \rangle}.$$

² Note that a perturbation that is everywhere constant does not change the equilibrium distribution. Hence, to compute the change in the concentration profile, we can focus on $g(q) = 1/2 - q/\omega$.

Now, $g_B = 1 - g_A = \frac{1}{2} + q/\omega$ inside the barrier, and zero elsewhere. Inside the barrier region, we have at all times

$$\partial_q g_B = \frac{1}{\omega}$$

and hence

$$k_{A \rightarrow B} = \frac{1}{\omega^2} \int_0^\infty dt \frac{\langle \dot{q}(0)\dot{q}(t) \rangle^*}{\langle c_A \rangle},$$

where the asterisk indicates the condition that both $\dot{q}(0)$ and $\dot{q}(t)$ should be within the barrier region. If the velocity correlations decay on a time scale that is much shorter than the time it takes to diffuse across the barrier, then we can write

$$\langle \dot{q}(0)\dot{q}(t) \rangle^* \approx \langle \dot{q}(0)\dot{q}(t) \rangle \omega \exp(-\beta U) \rho_{\text{eq}}.$$

The transition state theory expression for $k_{A \rightarrow B}$ is

$$k_{A \rightarrow B}^{TST} = 0.5 \langle |\dot{q}| \rangle \frac{\exp(-\beta U) \rho_{\text{eq}}}{\langle c_A \rangle}.$$

We then obtain the following expression for the transmission coefficient κ :

$$\kappa = \frac{2}{\omega \langle |\dot{q}| \rangle} \int_0^\infty dt \langle \dot{q}(0)\dot{q}(t) \rangle.$$

Making use of the Green-Kubo relation

$$D = \int_0^\infty dt \langle \dot{q}(0)\dot{q}(t) \rangle, \quad (15.3.19)$$

we obtain

$$\kappa = \frac{2D}{\omega \langle |\dot{q}| \rangle}.$$

As D is of order $\mathcal{O}(\langle |\dot{q}| \rangle \lambda)$, where λ is the mean-free path, we immediately see that

$$\kappa \sim \frac{\lambda}{\omega};$$

i.e., the transmission coefficient is approximately equal to the ratio of the mean-free path to the barrier width.

Next, we consider the statistical accuracy of our new estimate for κ

$$\kappa_{\text{est}} = \frac{2}{\omega \langle |\dot{q}| \rangle n} \sum_{i=1}^n \int_0^t dt' (\dot{q}(0)\dot{q}(t'))_i, \quad (15.3.20)$$

where we must remember that in all of the n trajectories considered the system is initially at the top of the barrier. Following essentially the same reasoning that led to Eq. (15.3.14) we now get

$$\begin{aligned} \langle (\Delta\kappa_{\text{est}})^2 \rangle &= \frac{4}{\omega^2 \langle |\dot{q}| \rangle^2 n} \left\{ \int_0^t dt' \int_0^t dt'' \langle \dot{q}(0) \dot{q}(t') \dot{q}(0) \dot{q}(t'') \rangle \right. \\ &\quad \left. - \left[\int_0^t dt' \langle \dot{q}(0) \dot{q}(t') \rangle \right]^2 \right\}. \end{aligned}$$

If we assume, as before, that \dot{q} behaves as a Gaussian variable, then

$$\langle (\Delta\kappa_{\text{est}})^2 \rangle = \frac{4}{\omega^2 \langle |\dot{q}| \rangle^2 n} \left[\langle \dot{q}^2 \rangle t \int_0^t dt' \langle \dot{q}(0) \dot{q}(t') \rangle + D^2 \right]. \quad (15.3.21)$$

We consider the limit $t \rightarrow \infty$. In that limit $D \ll \langle \dot{q}^2 \rangle t$ and hence

$$\langle (\Delta\kappa_{\text{est}})^2 \rangle \sim \frac{4}{\omega^2 \langle |\dot{q}| \rangle^2 n} \langle \dot{q}^2 \rangle Dt. \quad (15.3.22)$$

The relative error in the computation of the transmission coefficient is now

$$\frac{\langle (\Delta\kappa_{\text{est}})^2 \rangle^{1/2}}{\kappa} \sim \sqrt{\frac{\langle \dot{q}^2 \rangle t}{Dn}}. \quad (15.3.23)$$

From the Green-Kubo relation Eq. (15.3.19) we see that the diffusion constant D is equal to $\langle \dot{q}^2 \rangle \tau_c$, where τ_c is the decay time for velocity fluctuations. Hence,

$$\frac{\langle (\Delta\kappa_{\text{est}})^2 \rangle^{1/2}}{\kappa} \sim \sqrt{\frac{t}{n\tau_c}}. \quad (15.3.24)$$

Typically, there is not much point in computing the correlation function $\langle \dot{q}(0) \dot{q}(t) \rangle$ for times much larger than τ_c . Hence, the relative error in κ is simply $1/\sqrt{n}$. If we compare this expression for the statistical accuracy in κ with that obtained in the Bennett-Chandler scheme Eq. (15.3.17)

$$\left[\frac{\langle (\Delta\kappa_{\text{est}})^2 \rangle^{1/2}}{\kappa} \right]_{\text{Bennett-Chandler}} = \frac{1}{\kappa\sqrt{n}},$$

we conclude that, by a judicious choice of the scheme to compute κ , we have decreased the statistical error—for a given number of trajectories—by a factor κ . This implies that the present scheme is also applicable in the diffusive regime where $\kappa \ll 1$. Moreover, by suppressing the transient behavior, we have substantially reduced the time to compute a single barrier-crossing trajectory. The

additional gain due to the suppression of transients is of order

$$\frac{\tau_{\text{diff}}}{\tau_c} = \frac{\omega^2}{D\tau_c} = \left(\frac{\omega}{\lambda}\right)^2 \approx \frac{1}{\kappa^2}.$$

Hence, the overall gain in speed is of order $1/\kappa^4$. Of course, the present analysis is based on a highly simplified example. A discussion of the application of the present method to more realistic diffusive barrier-crossing problems is discussed in detail in ref. [629].

15.4 Path-sampling techniques

In the previous sections, we introduced the concept of reaction coordinates: functions of the coordinates of the system that measure the progress of a system from a “reactant” state A to a “product” state B . We use quotation marks to indicate that the transition between states A and B need not be a chemical reaction.

Even though it may be easy to distinguish states A and B —for instance, one might be a liquid, the other a crystal—there is no such thing as *the* reaction coordinate: many functions can measure the progress from A to B and, as we have seen in Example 25, the result of a simulation should be independent of this choice of order parameter. But, from a computational point of view, some reaction coordinates are better than others: making a good choice often decides whether a given rate calculation is feasible or not. In less extreme cases, the Bennett-Chandler approach and related techniques may still work, but the transmission coefficient for the barrier crossing may become very small, making the computational estimate of the crossing rate expensive. For example, if a chemical reaction in a liquid involves the complex reorganization of the solvent, the bottleneck of the reaction may be very different from the bottleneck for the same reaction in the gas phase. In such cases, it is dangerous to rely on our intuition about the choice of the reaction coordinate.

Clearly, it would be attractive to have computational techniques that make no *a priori* assumption about the reaction coordinate. This is where path-sampling techniques enter the game. Below, we discuss the ideas behind some of the methods. We stress that our discussion is necessarily incomplete. More comprehensive reviews can be found in refs. [34,634,635].

Before discussing path-sampling techniques to study the pathway and rate of rare processes, it is useful to ask the question: what do we want to know? It is not the individual pathways from reactant to product, nor is it necessarily the location of the transition state or its higher-dimensional analogs. Hummer [636], and in more detail, Vanden-Eijnden and co-workers (see e.g., [635,637]) have argued that the focus of our attention should be on the bundles of reactive paths. Together, these bundles carry the largest reactive flux, hopefully, localized in one, or a few, “tubes” in configuration space, or in the space spanned by some suitably guessed generalized coordinates. Identifying the dominant microscopic

pathway of a rare event is interesting in its own right, even if we are not primarily interested in rate calculations.

Some of the techniques that we discuss below can be viewed as special cases of **Transition Path Theory (TPT)** as formulated in refs. [635,637]. However, **TPT** is phrased assumed that the dynamics can be described by stochastic differential equations, such as the Langevin equation or its overdamped limit. Hence, Newtonian dynamics falls strictly speaking outside the scope of **TPT**. Also, in contrast to Forward-Flux Sampling (see section 15.5), the original implementation of **TPT** assumed that the stationary distribution of the system (Boltzmann or other) is known, which is usually not the case for driven systems (see, however, ref. [638]).

15.4.1 Transition-path sampling

Transition-Path Sampling (TPS) is a technique that was introduced by Chandler and co-workers [639–643], with the explicit aim of computing the rates of rare events without making *a priori* assumptions about the reaction coordinate. The development of the **TPS** approach was inspired by earlier work of Pratt [644], who introduced the concept of path sampling. Below we give a brief description of **TPS**. A more in-depth review can be found in refs. [645,646].

An important feature of the original version of **TPS** is that it was developed to compute the rate of rare events in systems evolving according to deterministic (e.g., Newtonian) dynamics. Although this may seem to be the simplest situation to consider, it is in many respects the hardest. As we shall see later, a bit of stochasticity makes path sampling easier.

To quantify the progress of the transition from state *A* to state *B* in an *N*-particle system, **TPS** defines a time correlation function $C(t)$ that measures the conditional probability that a trajectory starting in *A* at time $t = 0$, has arrived in *B* at time t . We call such a trajectory a *reactive* path. The probability that a path is reactive, is given by:

$$C(t) = \frac{\langle h_A(x_0)h_B(x_t) \rangle}{\langle h_A \rangle} \approx \langle h_B \rangle [1 - \exp(-t/\tau_R)], \quad (15.4.1)$$

where x_t denotes the phase-space coordinates $\{\mathbf{p}^N, \mathbf{r}^N\}$ of the *N* particles at time t , and h_A and h_B are “oracle” functions that return a value 1 if the system is in state *A* (*B*) and return 0 otherwise:

$$h_{A,B}(x) = \begin{cases} 1 & \text{if } x \text{ in A,B} \\ 0 & \text{otherwise.} \end{cases} \quad (15.4.2)$$

Note that the functions $h_{A,B}$ are not reaction coordinates: they just allow us to specify if the system is in the initial (final) state of the transition.

As before, we assume that transitions from *A* to *B* are rare events, meaning that $\tau_R \gg \tau_{\text{micro}}$, where τ_{micro} is the typical duration of a successful crossing

from A to B . As in Eq. (15.1.6) the rate constant follows from the time derivative $\dot{C}(t)$

$$k(t) = \dot{C}(t), \quad (15.4.3)$$

which reaches a plateau value for times $\tau_{\text{micro}} \ll t \ll \tau_R$. The correlation function $C(t)$ defined in Eq. (15.4.1) differs slightly from the quantity introduced in Eq. (15.2.1), but both expressions lead to the same value of $k(t)$.

15.4.1.1 Path ensemble

The crucial step in **TPS** and related techniques is the introduction of a *path ensemble*, which is an important extension of the usual ensemble of *points* in phase space. The path ensemble allows us to define averages over trajectories: just as we can relate a free energy to the partition function of a domain in phase space, so we will be able to associate a “free energy” to the ensemble of all paths that connect two regions in phase space in a given time t . In introducing the path ensemble, we make no assumptions about the nature of the dynamics (Newtonian, Langevin, Brownian), although **TPS** was designed for deterministic, Newtonian dynamics.

To stay close to the notation used in much of the literature on path sampling, we will use the symbol $x(\mathcal{T})$ to denote a trajectory of the system in time interval \mathcal{T} . In the case of Newtonian dynamics, x is the phase-space coordinate $\{\mathbf{p}^N, \mathbf{r}^N\}$, but for Brownian dynamics, it would just be the configuration-space coordinate $\{\mathbf{r}^N\}$.

In simulations, time is discretized with a time-step Δt . However, in theoretical analyses, we can consider the limit $\Delta t \rightarrow 0$. By discretizing time, we have created discrete *time slices* in x -space. A trajectory is then a sequence of x -values for successive time slices. The dynamics of the system determines the probability of a given trajectory. For instance, if we start at x_0 at time t_0 , then the probability to follow a particular trajectory $\{x_0, x_{\Delta t}, x_{2\Delta t}, \dots, x_{\mathcal{T}}\}$ is determined by the (Boltzmann) probability $\mathcal{N}(x_0)$ of finding the system at x_0 at time $t = 0$ multiplied by the probability $\mathcal{P}[x(\mathcal{T})]$ that a path will go from x_0 to $x_{\mathcal{T}}$ via $x_{\Delta t}, x_{2\Delta t}, x_{3\Delta t}$, and so on, is given by

$$\mathcal{P}[x(\mathcal{T})] = \mathcal{N}(x_0) \prod_{i=0}^{(\mathcal{T}/\Delta t)-1} \pi(x_{i\Delta t} \rightarrow x_{(i+1)\Delta t}), \quad (15.4.4)$$

where $\pi(x_{i\Delta t} \rightarrow x_{(i+1)\Delta t})$ denotes the probability that a system that is at $x_{i\Delta t}$ will arrive after one time step at $x_{(i+1)\Delta t}$. For deterministic (Newtonian) dynamics, this notation is a bit of an overkill as the path is fixed once we have specified x_0 . However, for stochastic dynamics, $\pi(x \rightarrow x')$ is a (normalized) distribution.

With this notation, the ensemble averages in Eq. (15.4.1) can be written as an integration over the initial conditions weighted with the equilibrium distribution

$\mathcal{N}(x_0)$. We first note that we can write Eq. (15.4.1) as

$$C(t) = \frac{\int dx_0 \mathcal{N}(x_0) h_A(x_0) h_B(x_t)}{\int dx_0 \mathcal{N}(x_0) h_A(x_0)}. \quad (15.4.5)$$

Using Eq. (15.4.4), we can write Eq. (15.4.5) as an average over the path probability $\mathcal{P}[x(\mathcal{T})]$:

$$C(t) = \frac{\int \mathcal{D}x(t) h_A(x_0) \mathcal{P}[x(t)] h_B(x_t)}{\int \mathcal{D}x(t) h_A(x_0) \mathcal{P}[x(t)]}, \quad (15.4.6)$$

where $\int \mathcal{D}x(t)$ denotes a path-integral, i.e., an integral over all paths between x_0 and x_t [647], and $\mathcal{D}x(t) \equiv \prod_i dx_i \Delta t$.

Note that the form of Eq. (15.4.6) resembles the ratio of two partition functions:

$$C(t) = \langle h_B(t) \rangle_{A,t} = \frac{Z_{AB}(t)}{Z_A(t)}, \quad (15.4.7)$$

where

$$Z_A = \int \mathcal{D}x(t) h_A(x_0) \mathcal{P}[x(t)]$$

is the partition function of all paths of length t that start in A at $t = 0$, while

$$Z_{AB} = \int \mathcal{D}x(t) h_A(x_0) \mathcal{P}[x(t)] h_B(x_t)$$

is the partition functions of all paths of length t that start in A at $t = 0$ and end up in B at time t . The fact that $C(t)$ can be expressed as the ratio of path-ensemble partition functions is exploited in TPS, because it allows us to evaluate Eq. (15.4.6) using the free-energy techniques discussed in Chapter 8. Below, we sketch how the analogy between free energy calculations and the evaluation of partition functions in the path ensemble can be used to compute rates of rare events. Only after that will we discuss how Monte-Carlo sampling can be used to generate trial reactive paths, starting from existing reactive paths between A and B . The only thing we need to know for now is that path-sampling algorithms attempt trial moves that generate trajectories that are at least locally close to an existing reactive trajectory. A trial trajectory is only accepted if it is also reactive (i.e., linking A and B in time t), and has a high-enough Boltzmann weight.

Of course, simulating an entire trajectory to see if it ends up in B and originates in A makes path sampling expensive: the larger t , the more expensive the test. This preliminary information is useful to appreciate the trick developed in ref. [642] to reduce the cost.

15.4.1.2 Computing rates

To compute reaction rates with TPS, we start from the expression for the rate $k(t)$

$$k(t) = \dot{C}(t) = \frac{\langle h_A(x_0)\dot{h}_B(x_t) \rangle}{\langle h_A \rangle}. \quad (15.4.8)$$

For $\tau_{\text{micro}} \ll t \ll \tau_R$, $k(t)$ reaches a plateau value, which defines the reaction rate. Hence, to compute rates, we must evaluate $\dot{C}(t)$.

In principle, we could compute $C(t)$ from a large number of “ordinary” MD simulations, thereby generating an ensemble of paths of length t that start at A and then counting the fraction that happens to be in B at time t . However, since we consider a situation where transitions from A to B are rare, the fraction of all paths that end in B is very small, and the brute-force approach would be prohibitively expensive. We, therefore, need computational approaches that do not waste most time on trajectories that are not going from A to B .

For what follows, we will assume that we can construct some, not necessarily representative, reactive paths that link A to B in time t . We can then subsequently apply a path sampling algorithm, which, starting from these initial paths, allows us to generate an equilibrium ensemble of reactive paths. How this is done will be discussed section 15.4.2.

Looking at Eq. (15.4.8), one might be inclined to compute $C(t)$ for a range of times t , find the regime where $C(t)$ is linear in t , and then obtain $\dot{C}(t)$ by numerical differentiation. Later, we shall see that this calculation can be simplified, but let us first focus on the calculation of $C(t)$.

As a preliminary step, we need an order parameter $\lambda(x)$ that allows us to distinguish states A and B . For instance, state B could correspond to a range of λ values between limits λ_{\min}^B and λ_{\max}^B :

$$x \in B \quad \text{if} \quad \lambda_{\min}^B < \lambda(x) < \lambda_{\max}^B, \quad (15.4.9)$$

and similarly for A . An example may help: we could consider a particle crossing a one-dimensional potential-energy barrier. In that case, λ could be simply the coordinate that measures the progress of the particles across the barrier region. The above example is a bit too simple, because the coordinate of the particle not only distinguishes A from B , but it is also a good reaction coordinate. In general, however, order parameters that are good enough to distinguish between A and B , may not be useful reaction coordinates. But a good reaction coordinate is always a good order parameter.

The next step is to link the computation of $C(t)$ and umbrella sampling (section 8.6.6). In umbrella sampling, we can force a system to explore an order-parameter range where the free energy is large (probability is low), by sampling the probability distribution in a sequence of partially overlapping windows, where the windows are chosen narrow enough that the probability distribution can be sampled over the entire window, without further biasing. By matching

the unnormalized probability distributions within these windows, we can reconstruct the overall probability distribution along the path between the states on either side of the barrier (see Example 11 in section 8.6.11).

In path sampling, we can do something similar to compute the probability that a path that starts in A at $t = 0$ ends up in a window with value λ at time t . Note that here we do not yet assume that λ is in the range corresponding to B .

Let us define $P(\lambda, t)$ as the probability density of finding the system with order parameter $\lambda = \lambda(x_t)$ at time t starting from A at $t = 0$:

$$P(\lambda, t) \equiv \frac{\int dx_0 \mathcal{N}(x_0) h_A(x_0) \delta[\lambda - \lambda(x_t)]}{\int dx_0 \mathcal{N}(x_0) h_A(x_0)}. \quad (15.4.10)$$

Unless t is very large, most paths of length t go from A to A . However, a fraction will end up with a λ -value adjacent to, but outside the range of A : we will label this window as window 1. Sampling many paths that start from equilibrium configurations in A , we can populate window 1. From this simulation, we can compute the fraction of all paths that start in A and end up in window 1. However, as we have made successive windows overlap, some of these paths also end in the next window (window 2). Now we use the path-sampling techniques of section 15.4.2 to use the paths that end up in window 2 in time t , to generate many more paths that end up in that window: this condition means that we reject all trial paths to end up outside window 2. But, of course, window 2 partially overlaps with window 3. And so, little by little, we construct paths that go in time t from A all the way to the range of λ -values that defines B . We note that this procedure generates reactive paths from A to B . However, they need not be representative: for instance, if the time t is not in the “plateau” regime.

In principle, we can use the window approach to compute the correlation function $C(t)$ for various values of t , check if it is linear in t , and compute $\dot{C}(t)$ in the linear regime. However, this procedure would be rather time-consuming. Dellago et al. [642] proposed a more efficient method to compute $C(t)$. To derive the approach of ref. [642], we note that $C(t)$ can be rewritten as (see Eq. (15.4.7)):

$$\begin{aligned} C(t) &= \langle h_B(t) \rangle_{A,t} = \frac{\langle h_A(x_0) h_B(x_t) \rangle}{\langle h_A(x_0) \rangle} \\ &= \frac{\langle h_A(x_0) h_B(x_t) \rangle}{\langle h_A(x_0) h_B(x'_t) \rangle} \times \frac{\langle h_A(x_0) h_B(x'_t) \rangle}{\langle h_A(x_0) \rangle} \\ &= \frac{\langle h_B(t) \rangle_{A,t}}{\langle h_B(t') \rangle_{A,t'}} \times C(t'). \end{aligned} \quad (15.4.11)$$

In words: $C(t)$ can be expressed as the product of the probability of finding a particle in B at time t' , multiplied by a correction factor equal to the ratio of the number of paths that end in B at times t and t' . The calculation of this correction would seem to be as complex as the calculation of $C(t)$. However, we can cast this term in a more convenient form. To this end, we define a slightly different

path ensemble, namely the ensemble of paths that have visited B at least once in the time interval $t \in [0; \mathcal{T}]$. An ensemble average in this ensemble can be written as

$$\langle h_B(x_t) \rangle_{A, H_B(\mathcal{T})} = \frac{\int dx_0 \mathcal{N}(x_0) h_A(x_0) H_B(x_{\mathcal{T}}) h_B(x_t)}{\int dx_0 \mathcal{N}(x_0) h_A(x_0) H_B(x_{\mathcal{T}})}, \quad (15.4.12)$$

where $H_B(\mathcal{T})$ is a characteristic function that has the value 1 if, somewhere in the interval $[0, \mathcal{T}]$, the path has visited B ; otherwise this function is 0. The difference with the previous ensemble is that a path does not have to end in B . We can use the fact that for all trajectories with length $t \in [0; \mathcal{T}]$ we have

$$h_B(x_t) = h_B(x_t) H_B(x_0, x_{\mathcal{T}}),$$

because if $h_B(x_t) = 1$, the $H_B(x_0, x_{\mathcal{T}})$ must be unity too, and if $h_B(x_t) = 0$, then both sides of the equation are zero. If we substitute this relation into Eq. (15.4.11), we obtain

$$\begin{aligned} C(t) &= \frac{\langle h_A(x_0) h_B(x_t) H_B(x_{\mathcal{T}}) \rangle}{\langle h_A(x_0) H_B(x_{\mathcal{T}}) \rangle} \times \frac{\langle h_A(x_0) H_B(x_{\mathcal{T}}) \rangle}{\langle h_A(x_0) h_B(x'_t) H_B(x_{\mathcal{T}}) \rangle} \times C(t') \\ &= \frac{\langle h_B(x_t) \rangle_{A, H_B(\mathcal{T})}}{\langle h_B(x'_t) \rangle_{A, H_B(\mathcal{T})}} \times C(t'). \end{aligned} \quad (15.4.13)$$

In this equation, we have rewritten the correction factor in terms of two ensemble averages. The nice feature of this ensemble average is that both averages can be obtained from a single path-sampling simulation of paths with length \mathcal{T} . For the rate constant $k(t)$ we then have

$$\begin{aligned} k(t) &= \frac{dC(t)}{dt} \\ &= C(t') \frac{1}{\langle h_B(x'_t) \rangle_{A, H_B(\mathcal{T})}} \langle \dot{h}_B(x_t) \rangle_{A, H_B(\mathcal{T})} \\ &\equiv \eta(t, t') C(t'). \end{aligned} \quad (15.4.14)$$

Note that in reactions where recrossings from B to A are common, $\dot{h}_B(x_t)$ in this equation can have both positive and negative contributions. Using Eq. (15.4.14), the calculation of the rate constant can be decomposed into two steps. First, we calculate $\eta(t, t')$ using the path ensemble as defined by Eq. (15.4.12). The second step is the calculation of $C(t')$ using umbrella sampling in the path ensemble.

It may be helpful to clarify the idea behind Eq. (15.4.13) by using an approximate human analogy: suppose that we wish to compute the rate at which people receive an Oscar for acting. In this case, state A corresponds to the overwhelming majority of people who have never even tried, or who have failed, and B corresponds to all living recipients. In between, there is a range of people

who tried acting but in the end, did not quite get there. The brute force approach would be to look at a large number of life histories, and determine the rate at which people win Oscars at the age of fifty (say) —this is our time t . Of course, the numbers are extremely small and the paths are long, making the umbrella sampling expensive. Hence, this approach would not be efficient.

Now suppose that, instead, we consider the biographies of people who have won an Oscar within a period of, say, ninety years since their birth (this is our time \mathcal{T}). Some of them may still be alive after ninety years and populate state B , others may have moved on. We can then find how many of them received the Oscar at age thirty (time t'). That number tells us little about the probability to have an Oscar at age thirty, if we do not know how many have failed to get there. So next, we compute the probability $C(30)$ that an average person is in possession of an Oscar at age thirty by using the equivalent of the umbrella sampling on life histories (path sampling) of people who try acting (in front of the mirror (=window 1), at a birthday party (window 2), at school (3), etc., all the way up to playing an Oscar-winning role. Finally, from our collection of biographies of Oscar winners, we compute the ratio R between the number of people who have received the Oscar by age fifty and those who already have it by age thirty. We then multiply R with $C(30)$ to obtain $C(50)$. To compute the rate, we must differentiate $C(t)$. To this end, we check our collection of biographies to find out how many people have won the Oscar between the ages of 50 and 51.

So, why this complicated route? The main reason is that we save much time by restricting ourselves to biographies of Oscar winners, except in the case of people who win an Oscar at the age of thirty: there we use umbrella sampling, but we use only relatively short life histories. It is not important that at age thirty we are not yet in the steady-state regime (if such a thing exists for Oscars): Eq. (15.4.13) is valid for any initial time. An illustration of the above procedure to compute crossing rates is discussed in Example 26.

Example 26 (Single particle in a two-dimensional potential well). To illustrate the path sampling method, consider a system containing a single particle in the following simple two-dimensional potential [641]:

$$\begin{aligned} V(x, y) = & \left(4(1 - x^2 - y^2)^2 + 2(x^2 - 2)^2 + ((x + y)^2 - 1)^2 \right. \\ & \left. + ((x - y)^2 - 1)^2 - 2 \right) / 6. \end{aligned} \quad (15.4.15)$$

Note that $V(x, y) = V(-x, y) = V(x, -y)$. Fig. 15.5 shows that this potential consists of two stable regions around the points $(-1, 0)$, which we call A, and $(1, 0)$, which we call region B. To be more specific, all points within a distance of 0.7 from $(-1, 0)$ or $(1, 0)$ are defined to be in region A or B, respectively. At a temperature of $T = 0.1$ transitions from A to B are rare events.

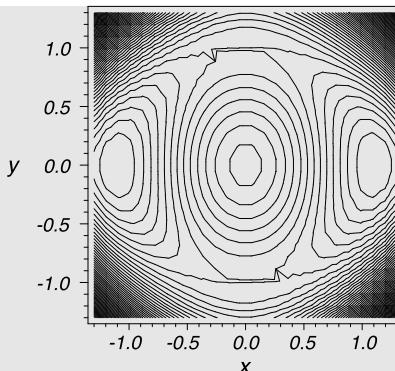


FIGURE 15.5 Contour plot of the function $V(x, y)$ defined by Eq. (15.4.15). The two minima are at $(-1, 0)$, A, and $(0, 1)$, B. These minima are separated by a potential energy barrier.

To compute the rate of transitions from A to B we used path ensemble simulations. The initial distribution $\mathcal{N}(x_0)$ was chosen to be canonical, i.e.,

$$\mathcal{N}(x_0) \propto \exp[-\beta \mathcal{H}(x_0)].$$

A trajectory was generated using standard Molecular Dynamics simulations (see Chapter 4). The equations of motion were integrated using the velocity-Verlet algorithm with a time step of 0.002.

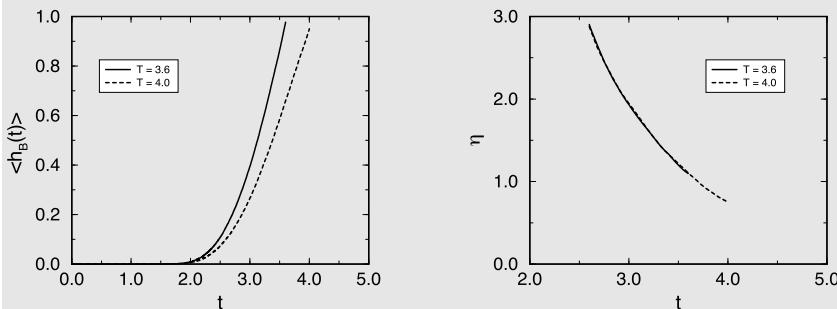


FIGURE 15.6 $h_B(t)$ (left) and $\eta(t)$ (right) as a function of time for various values of the total path length \mathcal{T} .

The first step was the calculation of the coefficient $\eta(t, t')$. This involves the computation of the path ensemble averages $\langle h_B(x_t) \rangle_{A, H_B(\mathcal{T})}$ for various times t . The result of such a simulation is shown in Fig. 15.6 for $\mathcal{T} = 4.0$ and $\mathcal{T} = 3.6$. An important question is whether the time \mathcal{T} is long enough. Since we are interested in the plateau of $k(t)$, the function $\langle h_B(x_t) \rangle_{A, H_B(\mathcal{T})}$ must have become a straight line for large values of t . If this function does not show a straight line, the value of \mathcal{T} was probably too short, the process is not a rare event, or the process cannot be described by a single hopping rate. The consistency of the simulations can be tested by comparing the results with a

simulation using a shorter (or longer, but this is more expensive) \mathcal{T} . Fig. 15.6 shows that the results of the two simulations are consistent.

The next step is the calculation of the correlation function $C(t)$. For the calculation of $P(\lambda, t)$, we have defined the order parameter λ as the distance from point B:

$$\lambda = 1 - \frac{|\mathbf{r} - \mathbf{r}_B|}{|\mathbf{r}_A - \mathbf{r}_B|}, \quad (15.4.16)$$

in which $\mathbf{r}_B = (1, 0)$. In this way, the region B is defined by $0.65 < \lambda \leq 1$ and the whole phase space is represented by $(-\infty, 1]$. In Fig. 15.7 (left), we have plotted $P(\lambda, i, t = 3.0)$ as a function of λ for different slices i . Recombining the slices leads to Fig. 15.7 (right). The value of $C(t = 3.0)$ can be obtained by integrating over region B:

$$C(t) = \int_B d\lambda P(\lambda, t). \quad (15.4.17)$$

Combining the results gives for the total crossing rate

$$k = \eta(t) C(t). \quad (15.4.18)$$

Using $t = 3.0$ leads to $\eta(3.0) = 1.94$, $C(3.0) = 4.0 \times 10^{-6}$, and $k = 8.0 \times 10^{-6}$.

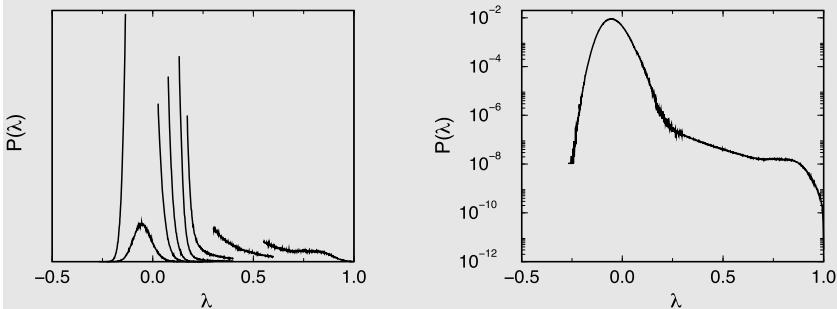


FIGURE 15.7 (Left) $P(\lambda, i, t = 3.0)$ for all slices i . (Right) $P(\lambda, t = 3.0)$ when all slices i are combined. The units on the y axis are such that $\int_{-\infty}^1 d\lambda P(\lambda, t) = 1$.

For more details, see SI (Case Study 24).

15.4.2 Path sampling Monte Carlo

In principle, **Transition-Path Sampling (TPS)** only requires information on the reactant and product state: the reaction pathway is a result of the simulation, not an input. Analyzing the ensemble of reactive paths may therefore help us identify a suitable reaction coordinate. Of course, the approach assumes that path sampling really probes the equilibrium path ensemble, which should be independent of how we created the first reactive paths.

To start a path-sampling simulation, we must first identify some path or paths that connect the reactant and product. Creating the first barrier-crossing paths in path sampling is like creating an initial configuration in a normal simulation, which may be valid but highly atypical: the first paths should connect A and B in time t , but they may be atypical. Typical paths are then created by path sampling, during the subsequent equilibration stage.

The fact that we can start with atypical paths makes it attractive to generate initial paths using a procedure that is cheap, rather than realistic. One way is to generate a barrier-crossing path at an unrealistically high temperature. However, there is no general “best” recipe to generate the first paths. Of course, it is also possible to use umbrella sampling, to generate the first paths that go from A to B . This procedure is expensive, but it is more likely to generate typical reactive paths.

Up to this point, we have not described the algorithm to sample reactive paths that start in A and, after a specified time, end up in B or, in the case of umbrella sampling, that end up in the i -th window along the coordinate λ that is used to distinguish A and B .

To emphasize the relation with umbrella sampling, we define “window potentials” $W_i(x_t)$ that allow us to obtain good statistics on paths that start in A and end at time t in window i between $\lambda^{\min}[i]$ and $\lambda^{\max}[i]$;

$$W_i(x_t) = \begin{cases} \infty & \lambda(x_t) < \lambda^{\min}[i] \\ 0 & \lambda^{\min}[i] < \lambda(x_t) < \lambda^{\max}[i] \\ \infty & \lambda(x_t) > \lambda^{\max}[i] \end{cases}. \quad (15.4.19)$$

It is necessary that neighboring windows overlap. To carry out path sampling, we use Monte Carlo trial moves that change an entire trajectory, rather than just a point in phase space. As with earlier MC algorithms, we use the criterion of detailed balance to construct valid path-sampling schemes.

In the present section, we focus on systems that evolve in time according to deterministic, Newtonian dynamics. However, transition path sampling can also be used in cases where the time evolution is more stochastic, e.g., if it is described by Brownian or by Langevin dynamics. Different dynamics require different path-sampling trial moves. We refer the reader to refs. [635,645] for a description of path-sampling moves in cases where the dynamics is stochastic.

Let us consider a path-sampling trial move. The old path is denoted by o and the new path by n . We distinguish two path ensembles: one ensemble comprises the collections of paths that start in A and are of length t . This ensemble is used to compute the correlation function $C(t)$.

The window potential (15.4.19) imposes the constraint that at time t the order parameter $\lambda(x_t)$ should lie within a pre-specified window. The path probability distribution for this ensemble is

$$\mathcal{N}(A, W) \propto \mathcal{N}(x_0) h_A(x_o) \exp[-W(x_t)]. \quad (15.4.20)$$

The second path ensemble, defined by Eq. (15.4.12), is the collection of paths of length T that start in A and that visit B at some time in the interval $t \in [0; T]$. This ensemble is used to sample the correction factor $\eta(t, t')$. The probability distribution for this ensemble is

$$\mathcal{N}(A, H_B) \propto \mathcal{N}(x_0) h_A(x_0) H_B(x_T). \quad (15.4.21)$$

The ratio of the acceptance probabilities for the forward and reverse path-sampling trial moves follows from the condition of detailed balance (see section 6.1):

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\mathcal{N}(n)\alpha(n \rightarrow o)}{\mathcal{N}(o)\alpha(o \rightarrow n)}, \quad (15.4.22)$$

where $\alpha(o \rightarrow n)$ is the *a priori* probability of generating path n from path o , and $\mathcal{N}(n)$ the desired probability distribution, i.e., Eqs. (15.4.20) or (15.4.21). In the Monte Carlo moves that are discussed below we consider trial moves for which the *a priori* probability of generating path n from o is equal to the probability of generating o from n . When we impose this equal *a priori* probability for forward and reverse path-sampling moves, the acceptance rules reduce to

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\mathcal{N}(n)}{\mathcal{N}(o)}. \quad (15.4.23)$$

Let us now consider the most important path-sampling moves. At first sight, it might seem simplest to generate a new path by making a small change in the initial condition x_0 and then perform a Molecular Dynamics simulation of length T or t , and use the acceptance rule to accept or reject this new path. However, because of the extreme sensitivity of MD trajectories to small changes in the initial conditions, this method will almost certainly fail for all but the shortest paths, or more precisely, for the windows nearest to A . A small change in the initial conditions of a path that visits B , will almost certainly result in a non-reactive path.

Shooting moves

A better strategy to generate a new path is to make a change in the x at some time t' , somewhere in the middle of the trajectory and integrate forwards to compute x_t and backward in time to compute x_0 . Since both A and B are stable states, these points will “attract” paths and therefore such shooting moves in the middle of a path are more likely to result in a path n that is also reactive.

Changing $x(t')$ will change the total energy of the trajectory, which in Newtonian dynamics, is conserved. Hence, we can split the acceptance of a path sampling move in a first “cheap” check on the change of the Boltzmann factor, and only if that step is accepted, carry out the forward and reverse integrations to verify that the new path still has a non-zero weight (Eq. (15.4.20) or Eq. (15.4.21)). Usually, we consider path “shooting” moves where the Jacobian

for the change from x_o to x_n equals one. For cases where this Jacobian is not equal to one, it should be included in the acceptance criterion (see [645]).

Shifting moves

We have seen that it is unlikely to generate a reactive path by making a small change in the initial positions or velocities. However, we can also change a trajectory by shifting initial and final times t_0 and t by the same amount Δt , where Δt can be positive or negative. For such a shifting move, we do not have to recompute the complete path, but only the increment i.e., we should perform a Molecular Dynamics simulation from t to $t + \Delta t$ if Δt is positive, and from t_0 to $t_0 + \Delta t$ if Δt is negative. Note, however, that shifting moves are, in practice, not ergodic. To improve the sampling, shifting moves should therefore be combined with shooting moves. Still, shifting moves are useful to improve the statistics.

As the above discussion of shooting and shifting moves makes clear, such trial moves tend to generate paths that can be reached from the original path without crossing a high free-energy barrier in path space. Yet, it is in general not guaranteed that a sequence of small “diffusive” steps in path space will be able to sample all relevant reactive trajectories, in particular when the most important reactive trajectories follow a path that is qualitatively different from the path that was used to initialize the path sampling.

15.4.3 Beyond transition-path sampling

Among the techniques to compute the rate of rare events, TPS occupies a special position, as it avoids some of the assumptions that earlier techniques made. For instance, unlike the Bennett-Chandler method, it does not assume that all co-ordinates orthogonal to the reaction coordinate equilibrate quickly. Moreover, TPS does not assume any knowledge of a reaction coordinate characterizing the progress of the transition: rather, TPS relies on the presumed “ergodicity” of path sampling to find the most important reactive trajectories. Another important feature of TPS is that it provides an unbiased estimate of the dominant reaction pathway: often, it is more important to know *how* a reaction proceeds, than *how fast*.

However, in its original form, TPS is computationally expensive. The high cost is partly due to the fact that paths must be followed during a time t that is long enough to ensure that $C(t)$, the probability that the system has crossed from A to B , scales linearly with t . Inevitably, this means that the transition paths spend a considerable fraction of the time in the basins of A and B , which is not wrong but uninformative.

15.4.4 Transition-interface sampling

To reduce the high cost of TPS, van Erp et al. introduced the so-called **Transition Interface Sampling (TIS)** method [648], which, like TPS, is a path-sampling method, but based on a rather different philosophy. First of all, unlike TPS,

TIS introduces an order parameter Q that measures the progress of the system from A to B . The set of points that correspond to a constant value of Q defines a hyper-surface in configuration space.³ Jumping ahead a little, we note that different terms have been introduced to denote these constant- Q hyper-surfaces. Faradjian and Elber [649], and also Haji-Akbari [650], use the term “milestones”, whereas Van Erp, Moroni, and Bolhuis [648], and also Allen et al. [651,652], use the term “interfaces”. In what follows, we shall use the term “interfaces” as it is more reminiscent of the hyper-surfaces in configuration space, noting that these interfaces are, of course, not physical interfaces that separate two phases.

The key innovation of **TIS** is that it separates the calculation of a transition rate into two factors: one is the rate at which trajectories arrive at an interface (a Q -hyper-surface) close to A , and the second measures the probability that a trajectory that has crossed this first interface will continue all the way to B rather than return to A . In practice, this second probability is decomposed into the product of probabilities that trajectories that have crossed interface i will proceed to interface $i + 1$, rather than return to A . Note that this factorization of the transition rate is very different from the approach used in **TPS**.

Computing the rate at which trajectories cross the first transition interface is straightforward: provided that this interface is not too far removed from A , the rate at which trajectories arrive at this interface from A can be computed in a normal MD simulation. The second step, computing the probability that a trajectory that has reached interface i will go on to $i + 1$, is computed using path-sampling techniques similar to **TPS**. An important advantage of **TIS** is that it focuses on trajectories that are in the process of crossing from A to B : no time is wasted computing the time evolution of a trajectory before it has left A or after it has arrived in B . In addition, **TIS** uses several other tricks that enhance its efficiency. For these, and other details, we refer the reader to ref. [648]. One important feature that **TIS** shares with **TPS** is that, during the path sampling, the acceptance of a trial move in trajectory space depends on the stationary distribution (e.g., the Boltzmann distribution) with which paths are generated.

15.5 Forward-flux sampling

In the spirit of **Transition Interface Sampling (TIS)**, **Forward-Flux Sampling (FFS)** [651–654] also aims to speed up the calculation of the transition rate from A to B by decomposing the barrier-crossing process into a sequence of crossings of intermediate interfaces defined by monotonically increasing Q -values. In **FFS**, a simulation is first run in the “reactant” state A , and configurations are collected that correspond to crossings of the first interface in the direction of increasing Q . This collection of configurations is used to initiate new simulations

³ In general, Q can also depend on the momenta, in which case a constant value of Q defines a hyper-surface in phase space.

that are continued either until the next interface is reached, or until the trajectory returns to A . This results in a new collection of configurations at the next interface that can be used to initiate simulations to the subsequent interface (or back to A), etc.

The crossing rate is then computed (as in [TIS](#)) as the product of the rate at which trajectories first arrive at the interface nearest to A , and the probability that such a trajectory will then continue to reach B without first returning to A . As in TIS, this probability is computed as the product of probabilities that a trajectory will proceed from interface i to $i + 1$ without returning to A , for all interfaces between A and B . Like [TIS](#) [648,655] and the “milestoning” approach of ref. [649], [FFS](#) does require the concept of a reaction coordinate, although the coordinate Q need not reproduce an actual pathway.⁴

[FFS](#) differs from other path-sampling techniques discussed before (and from [TPT](#)) in that it does not require prior knowledge of the phase-space density. Hence, [FFS](#) dispenses with the need to know the stationary distribution from which trajectories are generated. In fact, the earliest application of [FFS](#) considers rare switching events in driven biochemical networks: Boltzmann sampling would not even be possible in such systems. An advantage of [FFS](#) is that, unlike [TIS](#) (or [TPS](#)), it does not require propagating paths backwards in time to check whether they did indeed originate in A . The advantages of [FFS](#) mentioned above come at a cost: [FFS](#) will only work if the time evolution of the system has a stochastic component – it will not work for purely Newtonian dynamics. The reason why [FFS](#) needs stochasticity is simple: it makes it possible to generate new trajectories that are identical to a given parent trajectory from state A up to a given interface i , and different beyond that point. As a consequence, there is no need to back-propagate trajectories in time in order to check if they originated in A : they always do. This feature makes [FFS](#) particularly simple to implement or add to an existing code, and that is, in itself, a great advantage (see [34], p. 533).

15.5.1 Jumpy forward-flux sampling

The original Forward-Flux sampling approach was based on the assumption that if a system makes a transition characterized by the change in an order-parameter Q from Q_A to Q_B , then all intermediate values of the order parameter must also be visited on the way. It is this assumption that made it possible to decompose a transition in sub-steps: first from A to the first interface on the way from A to B , characterized by the order-parameter value Q_1 , then to Q_2 and so on, until Q_B .

However, this assumption is not always justified: sometimes, an order parameter may skip over one or more interfaces between A and B . An example is

⁴ We note that the milestoning approach of ref. [649] also considers the propagation of trajectories from one interface to the next, but it resembles the Bennett-Chandler method in that it assumes that the distribution of phase space points at the interfaces is equal to the stationary distribution of states at any given interface. For more details, see ref. [649].

nucleation and growth by cluster aggregation: if our reaction coordinate is the number of particles in a nucleus, then the attachment of a multi-particle cluster to a growing nucleus will result in a discontinuous jump in the number of particles in that nucleus. Such events are not captured by the original FFS scheme.

To address this issue, Haji-Akbari [650,656] has introduced a “jumpy” FFS scheme, that allows for the skipping of interfaces. In this algorithm, one should keep track of all distinct sequences of visited interfaces on the way from A to B . If we have chosen M distinct interfaces, then there are $2^M - 1$ possible sequences of visited interfaces between A and B , because every interface may, or may not, be visited: however, if the trajectory does not visit a single interface, then it has failed. In practice, the number of sequences sampled in a simulation is much less than $2^M - 1$.

However, apart from that extension —and an improved method for choosing the location of the interfaces —the scheme of ref. [650] has the same structure as FFS. As jumpy-FFS includes trajectories that would be missed by standard FFS, it yields higher (sometimes much higher [657]) estimates of the transition rate. Unlike the original FFS, trajectories generated with the approach of ref. [650] can deal with the situation where the barrier crossing evolves via states in between interfaces. For instance, if in a simulation of nucleation rates, cluster sizes of 10 and 20 are interfaces, then we might find in a simulation that the system jumps from cluster size 5 to 15. Nevertheless, these intermediate points can be used as starting points for launching new trajectories, as they have the correct statistical weight.

15.5.2 Transition-path theory

As mentioned in the introduction to path-sampling techniques, deterministic dynamics is arguably the most difficult case to tackle. As soon the dynamics generating trajectories has some stochasticity, path sampling becomes easier, and forward-flux sampling discussed in section 15.5 is a case in point.

Transition Path Theory (TPT) [635] provides a general framework for modeling rare events in systems that evolve in time according to a stochastic differential equation, such as the Langevin equation, or its overdamped limit (Brownian dynamics). The key observation of TPT is that associated with the stochastic differential equation for the dynamics. There is a partial differential equation (Fokker-Planck or, more precisely, Smoluchowski) that describes the evolution of the probability density of reactive paths.⁵ Associated with the Fokker-Planck equation, there is another equation (the *backward* Kolmogorov equation), the solution of which is the so-called committor $C(x)$. The value of $C(x)$ equals the probability that a trajectory starting in phase-space coordinate x located in the

⁵ There is some confusion about nomenclature here. We use the following convention: we use the name *Fokker-Planck* (or “forward-Kolmogorov”) equation to describe the equation that results from the lowest order expansion of the Chapman-Kolmogorov master equation for processes with diffusion and drift. If, in addition, we impose detailed balance, we obtain the Smoluchowski equation.

transition region between A and B , will end up in B before it ends up in A . The transition state (or actually, surface) is then the collection of all points x from where the system is equally likely to proceed to B as to return to A : in other words, at the transition surface $C(x) = 0.5$.

In the theory of rare events, the committor is a crucial quantity. In fact, as discussed in Illustration 22, P_B , a quantity closely related to $C(x)$ is often used in path-sampling simulations to test the quality of a postulated reaction coordinate. P_B is computed by launching many trajectories from points that have the same value of the reaction coordinate: some go to A , some to B . Note that different points in configuration space may have the same value of the reaction coordinate. For a good reaction coordinate, all points at the transition surface will have the same committor: 0.5. However, if we had chosen a poor reaction coordinate, the committor distribution for some points could be peaked close to one, and for other points, it may peak close to zero: this means that these points are not at all transition states. Some are inside the basin of attraction of A , others in the basin of attraction of B . Hence, computing the distribution of P_B is a way to test the quality of our reaction coordinate.

Of course, the above description of the committor immediately implies that the committor itself is the ideal reaction coordinate ... if only we could compute it easily.

TPT tells us, in principle, how to compute the committor. However, in practice, we can only compute the committor analytically in very simple cases (See Appendix G). In practical implementations of TPT, the calculation focuses on finding the minimum (free) energy path from A to B and, with this information, to compute how probability flows from A to B . In this picture, individual trajectories are no longer important: what TPT aims to achieve is to construct the path (string/tube) of the dominant probability flux from A to B , and from knowledge of this flux, to compute the reaction rate. A full description of the practical implementation of TPT is beyond the scope of this chapter, where we focus on trajectory-based simulations.

Illustration 22 (Ion pair dissociation). The dissociation of a Na^+Cl^- pair in water is an example of an activated process. It is of particular interest to understand the effect of the water molecules on the dynamics of this process.

As a first guess, one can use the ionic separation as a reaction coordinate:

$$r_{\text{ion}} \equiv |\mathbf{r}_{\text{Na}^+} - \mathbf{r}_{\text{Cl}^-}| .$$

The free energy as a function of this reaction coordinate is shown schematically in Fig. 15.8. Once we have computed the free energy barrier, we could, in principle, use the Bennett-Chandler approach to compute the reaction rate (see section 15.2). However, for this system, one would observe a very small transmission coefficient, which suggests that the chosen reaction coordinate does not provide an adequate description of the dynamics of this reaction.

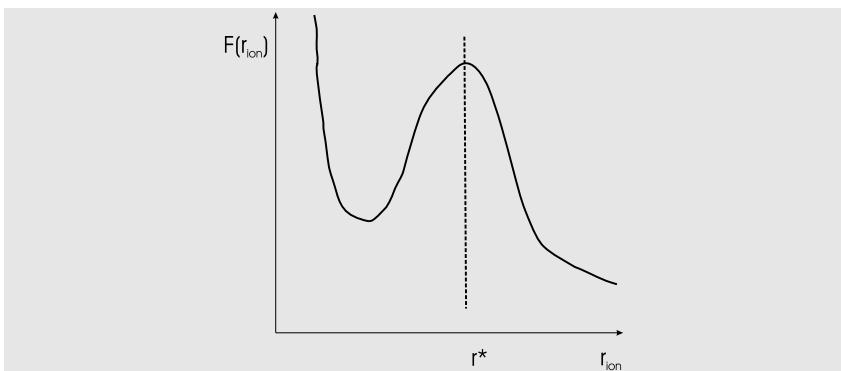


FIGURE 15.8 Free energy as a function of the ionic separation r_{ion} .

Fig. 15.9 explains how an unfortunate choice of the reaction coordinate may result in a low transmission coefficient in the Bennett-Chandler expression for the rate constant. But even if the reaction coordinate is well chosen, we may still get a low transmission coefficient. If the free energy landscape of the dissociation reaction looks like Fig. 15.9(a), the progress of the reaction would correlate directly with the ionic separation. However, it could still be that the system exhibits diffusive behavior near the transition state. If this is the case, one would obtain better statistics using the diffusive barrier crossing method described in section 15.3.

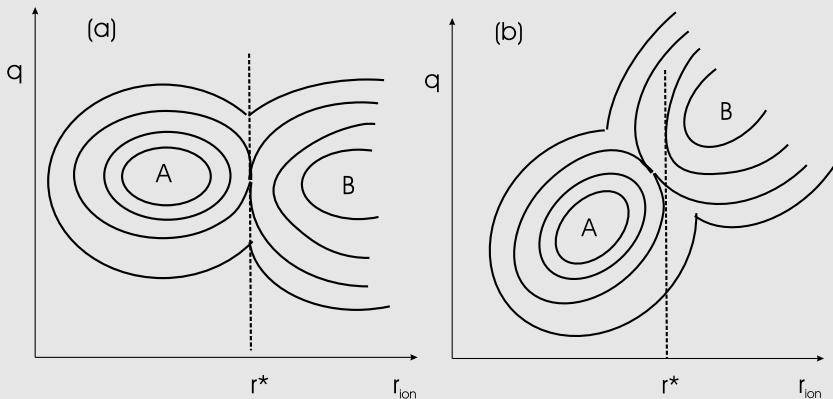


FIGURE 15.9 Two possible scenarios for the ion dissociation; the two figures show the two-dimensional free energy landscape in a contour plot. A is the stable associated state and state B the dissociated state, the dotted line corresponds to the dividing surface as defined by the maximum of the free energy profile (see Fig. 15.8), r_{ion} is the reaction coordinate while q represents all other degrees of freedom. In (a) one sees that the dividing surface nicely separates the two stable basins, while in (b) a point of r^* “belongs” either to the A basin or to the “ B ” basin.

Another possible scenario is shown in Fig. 15.9(b). Here, we have a situation in which the ionic separation is not a good reaction coordinate. Unlike the situation in Fig. 15.9(a), the dividing surface does not discriminate between the two stable states. Apparently, there is another relevant coordinate (denoted by q) that is an (as yet unknown) function of the positions of the solvent particles.

Since we do not know what this additional order parameter is, this is an ideal case to use [Transition-Path Sampling \(TPS\)](#), as in TPS we do not have to make an *a priori* choice of reaction coordinate.

Geissler et al. [658] used transition path sampling to generate some 10^3 reaction paths for this process. These paths were subsequently analyzed to obtain the *transition state ensemble*. This is the ensemble of configurations on the reaction paths that have the following “transition-state” property: half of the trajectories that are initiated at configurations that belong to this ensemble end up on the product side, and the other half on the reactant side. Although all trajectories originate from the same configuration, they have different initial velocities (drawn from the appropriate Maxwellian distribution). In general, P_B , the probability that a trajectory starting from an arbitrary configuration will end up in the state B is different from 0.5.

Geissler et al. showed that $P_B \approx 0$ for most configurations that contained fivefold-coordinated sodium ions. Conversely, $P_B \approx 1$, for configurations with sixfold-coordinated sodium ions. For the Cl^- ion, no such effect was found. This indicates that, in order to reach the transition state from the associated state, water molecules have to enter into the first solvation shell of the sodium ions. The water coordination of the Na^+ ion was the order parameter that was missing in the simple analysis. This example illustrates how [TPS](#) can be used to elucidate unknown “reaction” mechanisms.

15.5.3 Mean first-passage times

Our discussion of (non-jumpy) Forward-Flux Sampling provides a natural introduction to another method to study rare events, based on the distribution of [Mean First-Passage Times \(MFPT\)](#), even though the [MFPT](#) formalism in the theory of stochastic processes predates [FFS](#) by many years (see e.g., [67,659]).

The first step in [FFS](#) and [TIS](#), involves computing the rate at which a system that was originally in the “reactant” state A crosses the interface nearest to A (interface “1”). Suppose that we have started L independent trajectories in state A : for every trajectory i , we can measure the time interval $t_i^{(1)}$ until it first arrives at interface 1. The rate at which trajectories arrive at interface 1 is

$$\begin{aligned} k_{A \rightarrow 1} &= \frac{L}{\sum_{i=1}^L t_i^{(A \rightarrow 1)}} \\ &\equiv \frac{1}{\tau_{\text{MFPT}}^{(A \rightarrow 1)}}, \end{aligned} \quad (15.5.1)$$

where

$$\tau_{\text{MFPT}}^{(A \rightarrow 1)} = \frac{\sum_{i=1}^L t_i^{(A \rightarrow 1)}}{L} \quad (15.5.2)$$

is the average time it takes a trajectory to arrive at 1: this is the *mean first passage time* for trajectories that arrive at 1 from A. The times t_i in Eq. (15.5.2) account for the *total* time between one arrival at interface 1 and the next: the system spends most of that time in A.

Note that $k_{A \rightarrow 1}$ in Eq. (15.5.1) is a peculiar rate. It is the net *forward* rate: the reverse rate from 1 to A is not included. So $k_{A \rightarrow 1}$ is the rate that one would obtain if there would be an absorbing boundary at 1. Similarly, we can compute $\tau_{\text{MFPT}}^{(A \rightarrow 2)}$ the mean first passage time from A to interface 2, even though this is not what we compute in FFS. To compute $\tau_{\text{MFPT}}^{(A \rightarrow 2)}$, we could collect all trajectories that first arrive at 1 and then proceed to 2 without first returning to A. But note that there may be many trajectories that arrive at 1, but then return to A rather than proceed to 2. We have to include the duration of all these failed trajectories in the time to travel for A to 2. Hence, for rare events, the MFPT to travel from A to 2 can be much longer than the MFPT to travel to 1, when 2 is ‘uphill’ from 1. The forward rate of going from A to 2 is

$$k_{A \rightarrow 2} = \frac{1}{\tau_{\text{MFPT}}^{(A \rightarrow 2)}}. \quad (15.5.3)$$

Clearly, we can repeat this procedure for every subsequent interface, up to B. The forward rate of going from A to B is then

$$k_{A \rightarrow B} = \frac{1}{\tau_{\text{MFPT}}^{(A \rightarrow B)}}. \quad (15.5.4)$$

Eq. (15.5.4) shows that the forward rate of a process is directly related to the mean first passage time. Of course, the backward rate can be obtained by computing the mean passage time from B to A. For the relation between overall reaction rates and forward/backward rates, see Eq. (15.1.6).

In practice, MFPTs are often used to compute rates in cases where we can observe barrier crossings in brute-force simulations, but it should be clear that we can also compute the MFPT with FFS, be it that there is little to be gained.

The added advantage of the MFPT method is that, under certain conditions, it allows us to compute both the rate of a barrier crossing and the shape of the free energy barrier [660]. This approach to reconstruct the free energy barrier from MFPT data can be used when the barrier crossing is diffusive, and time-evolution of the probability density is governed by the Smoluchowski equation, or a discretized version thereof, as in the Becker-Döring theory of nucleation. The expression for the MFPT in the case of a crossing of a one-dimensional

free-energy barrier $F(q)$ between A and B :

$$\tau_{\text{MFPT}}^{(A \rightarrow B)} = \int_{q_a}^{q_B} dq \frac{e^{\beta F(q)}}{D(q)} \int_{-\infty}^q dq' e^{-\beta F(q')} , \quad (15.5.5)$$

where D is the diffusion coefficient, which may depend on q . For a derivation of this expression, and for a clear general discussion of the mean first-passage time approach, we refer the reader to ref. [34]. Additional details can be found in refs. [661,662].

To understand the physics behind Eq. (15.5.5), we give a simple derivation but approximate expression of the MFPT. We start with Eq. (15.5.4), which relates the forward flux to the MFPT. We assume that barrier crossing is diffusive and governed by the Smoluchowski equation [633] for the evolution of the probability density $\rho(q, t)$, written as⁶

$$\left(\frac{\partial \rho(q, t)}{\partial t} \right) = \nabla D(q) \rho_{\text{eq}}(q) \nabla (\rho(q, t) / \rho_{\text{eq}}(q)) \equiv -\nabla J \quad (15.5.6)$$

where $\rho_{\text{eq}}(q)$ is the normalized “equilibrium” probability density of the system in reactant state A , i.e., the distribution that would have resulted if the escape from A would be blocked, and J denotes the probability flux. ρ_{eq} is normalized and assumed to be small (but non-zero) outside A , which implies that

$$\rho_{\text{eq}}(q) \approx \frac{e^{-\beta F(q)}}{\int_{\text{basin } A} dq e^{-\beta F(q)}} . \quad (15.5.7)$$

We consider the case of slow barrier crossings, in which case we reach a steady state long before basin A has been depleted. In a steady state, the flux J is constant

$$J = -D(q) \rho_{\text{eq}} \nabla (\rho(q, t) / \rho_{\text{eq}}) \quad (15.5.8)$$

or

$$\int_A^B dq \frac{J}{D(q) \rho_{\text{eq}}} = - \int dq (\rho(q, t) / \rho_{\text{eq}}) = 1 , \quad (15.5.9)$$

where the last equality follows from the fact that in A , $\rho(q, t) \approx \rho_{\text{eq}}$ and in B $\rho(q, t) \approx 0$. Using Eq. (15.5.7), we can then rewrite Eq. (15.5.9) as

$$\int_{\text{basin } A} dq e^{-\beta F(q)} \int_A^B dq \frac{e^{\beta F(q)}}{D(q)} = J^{-1} = \tau_{\text{MFPT}}^{(A \rightarrow B)} , \quad (15.5.10)$$

which is basically Kramers’ result for the rate of a diffusive escape over a barrier [663].

⁶ The Smoluchowski equation can be viewed as a special case of the Fokker-Planck equation, with the added constraint of detailed balance —see e.g., [34].

Eq. (15.5.10) shows that, for barriers much larger than $k_B T$, the value of $\tau_{\text{MFPT}}^{(A \rightarrow B)}$ is dominated by the value of $D(q)$ in q -range close to the top of the barrier. The more general equation (15.5.5) can be used to compute MFPTs for interfaces at values of q between A and B , and the variation of $\tau_{\text{MFPT}}^{(A \rightarrow q)}$ with q can be used to reconstruct the free energy barrier [660].

On the whole, the MFPT approach is most useful for studying barrier-crossings that are sufficiently frequent to allow good statistics to be collected in brute-force simulations. It is not a good technique to study very rare events.

Illustration 23 (Transition-path sampling with parallel tempering). Transition-Path Sampling (TPS) is a technique that allows us to compute the rate of a barrier-crossing process without *a priori* knowledge of the reaction coordinate or the transition state. However, when there are many distinct pathways that lead from one stable state to another, it can be difficult to sample all possible pathways within the time scale of a single simulation. Vlugt and Smit [664] have shown that parallel tempering (see section 13.1.1) can be used to speed up the sampling of transition pathways that are separated by high free energy barriers.

The objective of TPS is to sample all relevant transition paths within a single simulation. This becomes difficult when different transition paths lead to distinct saddle points in the free energy surface. To be more precise, problems arise when the (free) energy barrier between two saddle points is much higher than $k_B T$ (see Fig. 15.10). Of course, the sampling problem would be much less serious if one could work at much higher temperatures where the transition path can cross the barriers separating the saddle points.

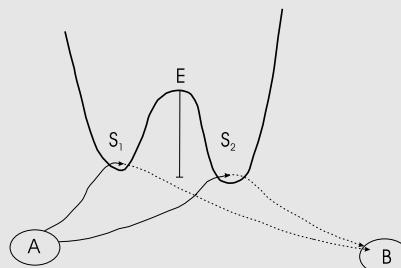


FIGURE 15.10 Schematic representation of two different transition paths from state A to state B . A and B are two stable states separated by a free energy barrier. There are two dynamical pathways for the system to go from A to B , one path crosses the barrier via saddle point S_1 while the other path crosses via saddle point S_2 . If the (free) energy barrier E between the two paths is much larger than $k_B T$, it is unlikely that path 1 will evolve to path 2 in a single transition path simulation. Note that the energy barrier between two paths (E) is not the same as the energy barrier along the path.

Parallel tempering exploits the possibility of generating “transitions” between different saddle points at high temperatures, to improve the sampling

efficiency at low temperatures. As was already discussed in section 13.1.1, parallel tempering can be used to switch between systems at various temperatures. As an illustration of this combined parallel tempering and transition path sampling approach, we consider the example discussed in ref. [664]: a two-dimensional system containing a linear chain of 15 repulsive Lennard-Jones particles. The nonbonded interactions are given by

$$u_{\text{rep}}(r) = \begin{cases} 1 + 4[r^{-12} - r^{-6}] & r \leq r_{\text{rep}} \\ 0 & r > r_{\text{rep}} \end{cases}, \quad (15.5.11)$$

in which r is the distance between two particles and $r_{\text{rep}} \equiv 2^{1/6}$ (we have used the Lennard-Jones σ as our unit of length). Neighboring particles i and j in the chain interact through a double-well potential:

$$u_{\text{dw}}(r_{ij}) = h \left[1 - \frac{(r_{ij} - w - r_{\text{rep}})^2}{w^2} \right]^2, \quad (15.5.12)$$

where r_{ij} denotes the distance between the neighboring particles i and j . This potential has two equivalent minima, one for $r_{ij} = r_{\text{rep}}$ and the other for $r_{ij} = r_{\text{rep}} + 2w$ ($w > 0$).

The chain can have a compact state, A , if all bonds are in the first minimum, or an extended state, B , if all bonds are in the second minimum. We wish to express the transition rate from A to B as the sum of the rates of the contributions due to all distinct transition paths. Clearly, there are many (in this case 14!) distinct pathways that lead from the compact state to the fully extended state (e.g., first stretch bond 2–3, then 1–2, then 6–7, etc.). Without Parallel Tempering (PT) it would take a prohibitively long time to obtain a representative sampling of all transition paths at low temperatures. In the present case, the number of distinct reaction paths is too large to be sampled even *with* parallel tempering. However, for many problems, the number of relevant paths is small, and the present approach can be used to compute the rate constant. For more details, the reader is referred to ref. [664].

15.6 Searching for the saddle point

In section 15.4.1, we described a general procedure for finding barrier-crossing rates. In principle, this procedure should work even if we have no knowledge about the reaction coordinate. However, in practice, the calculations may become very time-consuming. For this reason, a variety of techniques that aim to identify the relevant reaction coordinate have been proposed. Below, we briefly describe some of the methods that have been proposed. More details (and more references) can be found in ref. [48]. The methods for searching for a reaction path, which we discuss, have been designed for situations where the free energy barrier separating “reactant” and “product” is energy-dominated. This is often the case, but certainly not always (see, e.g., [556]).

One such energy-based scheme is the so-called **Nudged Elastic Band (NEB)** method proposed by Jonsson and collaborators [665]. This method aims to find the lowest-energy path to the saddle point that separates the reactant basin from the product basin. The **NEB** method assumes that both the reactant and product states are known. A number of replicas of the original system are now prepared. These replicas are initially located equidistantly along the “linear” path from reactant to product. The position of this originally linear path is now relaxed to find the reaction path. This is achieved as follows: the replicas are connected by harmonic springs that tend to keep them equally spaced—this is the “elastic band.” In addition, all replicas experience the gradient of the intermolecular potential that tends to drive them to a minimum of the potential energy. However, these gradient forces are only allowed to act perpendicular to the local tangent. Conversely, the elastic band forces are only allowed to act along the local tangent. As a consequence, the intermolecular forces move the elastic band laterally until the transverse forces vanish (i.e., when it is a minimum energy path) while the longitudinal forces prevent all replicas from collapsing into the reactant or product state. For more details and further refinements, see [666–668].

A technique that is similar in spirit, but very different in execution, to the **NEB** method is the activation-relaxation technique developed by Barkema and Mousseau [669]. This method also aims to find the lowest energy path to the saddle point. Unlike the **NEB** method, this scheme does not make use of any *a priori* knowledge of the product basin. To find the saddle point, the system is forced to move “uphill” against the potential energy gradient. However, if we would simply let the system move in a direction opposite to the force that acts on it, we would reach a potential energy *maximum*, rather than a saddle point. Hence, in the method of [669], the force that acts on the system is only inverted (and then only fractionally) along the vector in configuration space that connects the position of the initial energy minimum (i.e., the lowest energy initial state) with the present position of the system. In all other directions, the original forces keep acting on the system. The aim of this procedure is to force the system to stay close to the lowest energy trajectory towards the saddle point. Often there is more than one saddle point. In that case, the initial displacement of the system from the bottom of the reactant basin will determine which saddle point will be reached. Note that we cannot tell *a priori* whether the saddle point that is found will indeed be the relevant one. The true transition state can only be found by attempting many different initial displacements, and by computing the energy and (in the case of a quasi-harmonic energy landscape) the entropy of the saddle point.

The only reason special techniques are needed to simulate activated processes is simply that rare events are . . . rare. If, somehow, one could artificially increase the frequency of rare events in a controlled way, this would allow us to use standard simulation techniques to study activated processes. Voter and collaborators [670,671] have explored this route. The idea behind the approach of Voter is that the rate of activated processes can be increased either by artificially

lowering the energy difference between the top of the barrier and the reactant basin (“hyperdynamics” [670]) or by increasing the temperature (“temperature-accelerated dynamics” [671]).

The trick is to apply these modifications in such a way that it is possible to correct for the effect that they have on the crossing rate. In both schemes, the essence of the correction is that the rate k_i^B , at which the system crosses a point i in the biased system, is higher than the corresponding rate k_i^U in the unbiased system. To recover the unbiased rate, the biased rate should be multiplied by a factor

$$\frac{P_{\text{Boltzmann}}^U(i)}{P_{\text{Boltzmann}}^B(i)},$$

where $P_{\text{Boltzmann}}^U$ ($P_{\text{Boltzmann}}^B$) is the unbiased (biased) Boltzmann weight of configuration i . For more details, the reader is referred to refs. [670,671]. To recover the unbiased rate, the biased rate should be multiplied by a factor

$$\frac{P_{\text{Boltzmann}}^U(i)}{P_{\text{Boltzmann}}^B(i)},$$

where $P_{\text{Boltzmann}}^U$ ($P_{\text{Boltzmann}}^B$) is the unbiased (biased) Boltzmann weight of configuration i .

An additional “linear” speed-up of the rate calculations can be achieved by performing n barrier-crossing calculations in parallel [672]. Although this approach does not reduce the total amount of CPU time required, it does reduce the wall-clock time of the simulation. For more details, the reader is referred to refs. [670–672].

15.7 Epilogue

After this long but far from an exhaustive review of techniques to predict the pathway and rate of rare events, it is fair to ask if any of these techniques is guaranteed to find the dominant path by which a rare transition proceeds in reality. It may come as something of a let-down to the reader that the answer is: “Past performance is no guarantee for future results”. Just as free-energy calculations typically compare the relative stabilities of structures that we have identified, so rare-event techniques can only compute the rates for pathways that we have guessed or found. Yet, intuition and guessing are becoming less important in this context as [Machine Learning \(ML\)](#), particularly the use of auto-encoders, is transforming this field. ML can help to identify the number and approximate functional form of the collective variables that best describe, e.g., reaction pathways [673] or transition states [674]. Undoubtedly, much more is to come. Hence, we cannot do much more than signaling these developments and refer the reader to the emerging literature.

Chapter 16

Mesoscopic fluid models

This book describes classical molecular-simulation techniques. The focus on *classical* simulations means that we may use the results of quantum chemical calculations of the properties and interactions of atoms and molecules, but we do not discuss the quantum calculations as such. Neither do we use molecular simulations to study macroscopic phenomena where the particulate nature of atoms and molecules is irrelevant, such as in large-scale flow phenomena, or in the macroscopic description of the mechanical properties of solids. Molecular simulations can be used to predict the properties of such materials, e.g., the viscosity, thermal conductivity, or, in the case of solids, the elastic constants (Appendix F.4). But that is where the role of molecular simulations stops: we will not consider computational techniques to model a waterfall or a buckling beam.

However, in practice, the separation between the micro and the macro world is not always clean. There are many examples of physical phenomena that require a combination of a macroscopic description with a model that accounts for the particulate nature of matter. Examples are two-phase flows near interfaces or the propagation of a fracture tip in a solid.

Another important example where micro and macro meet is the study of colloidal suspensions. Colloids are mesoscopic particles with typical sizes of 10 nm–1 μm . These particles consist of millions, or even billions, of atoms, and these bodies are, therefore, best described by continuum mechanics. However, colloids undergo thermal motion, and a collection of colloids behaves in many ways like a collection of giant atoms: they are subject to the laws of Statistical Physics, and their properties can be studied by molecular simulations. Yet, such simulations will not focus on the atoms in the colloids, but rather treat the colloids themselves as the basic building blocks of a suspension.

The situation becomes more complex when we consider the solvent in which these colloids move: clearly, it would be prohibitively expensive to model the millions or billions of molecules of the solvent. Yet, we cannot ignore the solvent because, without a solvent, the colloids would move ballistically, whereas, in reality, the viscous drag of the solvent quickly kills off any ballistic motion of the colloids. The solvent is also driving the thermal motion of the colloids. In addition, the solvent is responsible for hydrodynamic interactions between the colloids, i.e., the phenomenon that the motion of one colloid creates a (transient) flow field that affects the motion of other colloids.

It is distinctly unattractive to use the continuum (Navier-)Stokes equation to compute all hydrodynamic interactions in a dense suspension of colloids, particularly in confined geometries. This is why it is common to use a simplified, particle-based description of the solvent: not to reproduce the molecular properties of the real solvent, but to account in the simplest way possible, both for the thermal motion of the colloids and for their hydrodynamic interactions in arbitrary geometries. We stress that the above considerations for the introduction of a coarse-grained model of the solvent apply only to the *dynamics* of colloids. For the static properties, hydrodynamics plays no role. Hence, for structural properties, implicit solvent models can be fine, provided that they adequately account for the effect of the solvent on the effective intermolecular interactions between solute particles.

The latter point is well illustrated by the use of the hard-sphere model to describe colloidal suspensions. Although the hard-sphere model was introduced as the simplest possible model of an atomic liquid [18], it became clear in the 1980s that it provided a rather realistic description of the static properties of suspensions of hard, spherical colloids (see, e.g., [209]).

However, the dynamics of Alder and Wainwright's hard-sphere model was Newtonian, which mean that the motion of the spheres in between collisions is purely ballistic. Such a model cannot account for the hydrodynamic effects of the solvent. Yet, although we cannot ignore the existence of the solvent, the Brownian motion of a hard spherical colloid depends on the molecular properties of the solvent almost exclusively through the solvent viscosity, even for dense suspensions. In addition, Brownian diffusion depends on temperature, a quantity that is completely insensitive to molecular details. Hence, if we are interested in the dynamics of suspensions, any solvent model that can reproduce the thermal motion and the viscosity of the solvent will do. The above comments highlight the need for constructing particle-based solvent models, which are as simple as possible, yet conserve momentum and obey Boltzmann statistics.

The very simplest such model, namely the ideal gas, will not do, because ideal gas particles never collide with each other: as a consequence, flow in such a system will not evolve according to the Navier-Stokes equation. It is, however, possible to develop models that have the static properties of an ideal gas, yet undergo collisions, with the result that the system behaves like a hydrodynamic fluid. Two models, in particular, belong to this category. One method was introduced by Malevanets and Kapral in 1999 [675,676]. The method is now widely known under the name **Stochastic Rotation Dynamics (SRD)**, which is a particular case of a wider class of models that are grouped under the name **Multi-particle Collision Dynamics (MPC)** [677].

We will briefly mention another method to describe a solvent with thermal fluctuations, namely the Lattice-Boltzmann method. However, on this topic, we will be brief because it is rather different from the particle-based methods described in the rest of this book. Moreover, Succi has written an excellent book devoted exclusively to the Lattice-Boltzmann method [678].

16.1 Dissipative-particle dynamics

The **Dissipative Particle Dynamics (DPD)** method was introduced by Hoogerbrugge and Koelman [679,680] as a cheap, particle-based model for a solvent. The key feature of the **DPD** method is that, in addition to conservative forces between the particles, it includes momentum-conserving friction forces and random forces between pairs of particles:

$$\mathbf{F}_i = \sum_{j \neq i} \left[\mathbf{f}_i^C(\mathbf{r}_{ij}) + \mathbf{f}_i^D(\mathbf{r}_{ij}, \mathbf{v}_{ij}) + \mathbf{f}_i^R(\mathbf{r}_{ij}) \right]. \quad (16.1.1)$$

In most applications, the conservative forces acting on a given particle i can be written as a sum of pair-forces $\mathbf{f}_i^C = \sum_{j \neq i} \mathbf{f}_{ij}^C$, although versions of DPD with many-body forces do exist [681]. The dissipative force \mathbf{f}_{ij}^D is purely frictional: it depends both on the positions of the particles i and j , and on their relative velocities:

$$\mathbf{f}_{ij}^D(\mathbf{r}_{ij}, \mathbf{v}_{ij}) = -\gamma \omega^D(r_{ij})(\mathbf{v}_{ij} \cdot \hat{\mathbf{r}}_{ij})\hat{\mathbf{r}}_{ij}, \quad (16.1.2)$$

where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ and $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, and $\hat{\mathbf{r}}_{ij}$ is the unit vector in the direction of \mathbf{r}_{ij} . γ is a coefficient controlling the strength of the frictional force between the **DPD** particles. $\omega^D(r_{ij})$ describes the variation of the friction coefficient with distance.

The random force $\mathbf{f}_i^R(\mathbf{r}_{ij})$ is of the form:

$$\mathbf{f}_i^R(\mathbf{r}_{ij}) = \sigma \omega^R(r_{ij}) \xi_{ij} \hat{\mathbf{r}}_{ij}. \quad (16.1.3)$$

σ determines the magnitude of the random pair force between the DPD particles. ξ_{ij} is a random variable with Gaussian distribution¹ and unit variance and $\xi_{ij} = \xi_{ji}$, while $\omega^R(r_{ij})$ describes the variation of the random force with distance.

The functions $\omega^R(r)$ and $\omega^D(r)$ cannot be chosen independently. Español and Warren [683] showed that, in order to ensure Boltzmann sampling of the configurations and velocities of the system, the following relation must be satisfied [683]:

$$\omega^D(r_{ij}) = \left[\omega^R(r_{ij}) \right]^2, \quad (16.1.4)$$

which shows that $\omega^D(r_{ij})$ cannot be negative. In the spirit of the Langevin equation, γ and σ are related to the temperature according to

$$\sigma^2 = 2k_B T \gamma. \quad (16.1.5)$$

We derive Eqs. (16.1.4) and (16.1.5) in SI section L.13. However, the simple derivation of the relation between friction and random momentum transfer (Eqs. (7.1.6) through (7.1.8)) applies just as well to the present case, because the

¹ Groot and Warren [682] found that a uniform distribution with unit variance gave similar results.

relation should hold for any value $\omega^D(r)$. However, as the time step in a DPD simulation is of finite length, we should be careful about where in the time step we compute the friction. This issue is briefly discussed in section 16.1.1. Possible choices for the functional form of r -dependence of $\omega^R(r)$ will be discussed in Example 27.

At first sight, DPD method bears a strong resemblance to Langevin dynamics (section 7.1.1.3), as both schemes employ a combination of random and dissipative forces. However, in Langevin dynamics, the frictional and random forces do not conserve momentum. In DPD, both the frictional and random forces do not change the center-of-mass velocity of a pair of particles, and hence the algorithm conserves momentum. Momentum conservation is necessary for recovering the correct “hydrodynamic” (Navier-Stokes) behavior on sufficiently large length and time scales. An alternative approach for imposing both momentum conservation and proper Boltzmann sampling has been proposed by Lowe [258] (see section 7.1.1.2).

We still have to show that, on sufficiently large lengths and time scales, the DPD fluid obeys the Navier-Stokes equation of hydrodynamics. At present, there exists no rigorous demonstration that this is true for an arbitrary DPD fluid.² However, all existing numerical studies suggest that, in the limit where the integration time step $\delta t \rightarrow 0$, the large-scale behavior of the DPD fluid is described by the Navier-Stokes equation. The kinetic theory for the transport properties of DPD fluids [684–687] supports this conclusion. One interesting limit of the DPD model is the “dissipative ideal gas”, i.e., a DPD fluid without the conservative forces. The static properties of this fluid are those of an ideal gas, yet its transport behavior is that of a viscous fluid.

The advantage of DPD over conventional (atomistic) MD is that it uses a coarse-grained model with a soft intermolecular potential (see SI, Case Study 25). Such models are computationally cheap because they allow rather large time steps. The applicability of soft potentials in coarse-grained models is not limited to simple fluids: we can use the same approach to construct coarse-grained models of the building blocks of complex liquids. However, if we are only interested in the static properties of a complex liquid, we could have used standard MC or MD on a model with the same conservative forces, but without dissipation. The real advantage of DPD shows up when we try to model the dynamics of complex liquids.

16.1.1 DPD implementation

A DPD simulation can easily be implemented in any working Molecular Dynamics program. The only subtlety is in the integration of the equations of

² In MD simulations, many seemingly obvious properties cannot be proven rigorously. One would expect the Navier-Stokes picture to hold if momentum and mass are conserved, and if all fluctuations of non-conserved quantities decay on microscopic length and timescales.

motion. As the forces between the particles depend on their relative velocities, the standard velocity-Verlet scheme cannot be used.

In their original publication, Hoogerbrugge and Koelman [679] used an Euler-type algorithm to integrate the equations of motion. However, Marsh and Yeomans found that, with such an algorithm, the effective equilibrium temperature depends on the time step that is used in a DPD simulation [688]. Only in the limit of the time step approaching zero was the correct equilibrium temperature recovered. A similar result was obtained by Groot and Warren [682] using a modified velocity-Verlet algorithm. There is, however, an important feature missing in these algorithms. If we compare the DPD integration schemes with those used in a Molecular Dynamics simulation, all “good” MD schemes are intrinsically time-reversible while the above DPD schemes are not. Pagonabarraga et al. [689] argued that time reversibility is also important in a DPD simulation, since only with a time-reversible integration scheme can detailed balance be obeyed. In the Leap-Frog scheme (see section 4.3.3) the velocities are updated using

$$v(t + \Delta t/2) = v(t - \Delta t/2) + \Delta t \frac{f(t)}{m}, \quad (16.1.6)$$

and the positions using

$$r(t + \Delta t) = r(t) + \Delta t v(t + \Delta t/2). \quad (16.1.7)$$

In DPD, the force at time t depends on the velocities at time t . The velocity at time t is approximated by

$$v(t) = \frac{v(t + \Delta t/2) + v(t - \Delta t/2)}{2}.$$

This implies that the term $v(t + \Delta t/2)$ appears on both sides of Eq. (16.1.6). In the scheme of Pagonabarraga et al., these equations were solved self-consistently; i.e., the value for $v(t + \Delta t/2)$ calculated from Eq. (16.1.6) has to be the same as the value for $v(t + \Delta t/2)$ used to calculate the force at time t . This implies that we have to perform several iterations before the equations of motion can be solved. This self-consistent scheme implies that the equations are solved in such a way that time reversibility is preserved.

To see how time reversibility is related to detailed balance, we consider a single DPD step as a step in a Monte Carlo simulation. If we have only conservative forces, DPD is identical to standard Monte Carlo. In fact, we can use the hybrid Monte Carlo scheme (see section 13.3.1) for the DPD particles. For hybrid Monte Carlo, a time-reversible algorithm must be used to integrate the equations of motion. In the case of hybrid Monte Carlo, detailed balance implies that if we reverse the velocities, the particles should return to their original positions. If this is not the case, detailed balance is not obeyed. If we use a non-iterative scheme to solve the equations of motion in our DPD scheme, the

velocity that we calculate at time t is not consistent with the velocity that is used to compute the force at this time. Hence, if we reverse the velocities, the particles do not return to their original positions, and detailed balance is not obeyed. In Example 27, the DPD method is illustrated with a few examples. In the DPD approach, the forces due to individual solvent molecules are lumped together to yield effective friction and a fluctuating force between moving fluid elements. While this approach does not provide a correct atomistic description of molecular motion, it has the advantage that it does reproduce the correct hydrodynamic behavior on long length and time scales.

Although the idea behind DPD is simple, the integration of the equations of motion, which contain velocity-dependent forces and position-dependent friction, requires care. The problem of position-dependent friction is discussed in refs. [250,682]. Different DPD algorithms are compared in ref. [690].

Example 27 (Dissipative particle dynamics). To illustrate the DPD technique, we have simulated a system of two components (1 and 2). The conservative force is a soft repulsive force given by

$$\mathbf{f}_{ij}^C = \begin{cases} a_{ij} (1 - r_{ij}) \hat{\mathbf{r}}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases}, \quad (16.1.8)$$

in which $r_{ij} = \|\mathbf{r}_{ij}\|$ and r_c is the cutoff radius of the potential. The random forces are given by Eq. (16.1.3) and the dissipative forces by Eq. (16.1.2). The total force on a particle equals the sum of the individual forces:

$$\mathbf{f}_i = \sum_{i \neq j} (\mathbf{f}_{ij}^C + \mathbf{f}_{ij}^S + \mathbf{f}_{ij}^R + \mathbf{f}_{ij}^D). \quad (16.1.9)$$

To obtain a canonical distribution, we use

$$\sigma^2 = 2\gamma k_B T$$

$$w^D(r_{ij}) = [w^R(r_{ij})]^2.$$

A simple but useful choice is [682]

$$w^D(r_{ij}) = \begin{cases} (1 - r_{ij}/r_c)^2 & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases}$$

with $r_c = 1$. The simulations were performed with $\rho = 3.0$ and $\sigma = 1.5$. We have chosen $a_{ii} = a_{jj} = 25$ and $a_{ij,i \neq j} = 30$. This system will separate into two phases. In the example shown in Fig. 16.1, we have chosen the z -direction perpendicular to the interface. The left part of Fig. 16.1 shows typical density profiles of the two components. In Fig. 16.1 (right), we have plotted the concentration of one of the components in the coexisting phases.

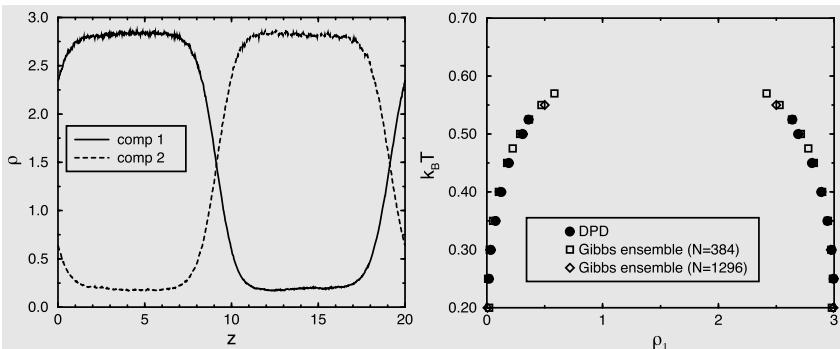


FIGURE 16.1 (Left) Density profile for $k_B T = 0.45$. (Right) Phase diagram as calculated using DPD and Gibbs ensemble simulations. Both techniques result in the same phase diagram, but the Gibbs ensemble technique needs less particles due to the absence of a surface. In the DPD simulations, we have used a box of $10 \times 10 \times 20$ (in units of r_c^3). The time step of the integration was $\Delta t = 0.03$.

Since we can write down a Hamiltonian for a DPD system, we can also perform standard Monte Carlo simulations [691]. For example, we can also use a Gibbs ensemble simulation (see section 6.6) to compute the phase diagram. As expected, Fig. 16.1 shows that both techniques give identical results. Of course, due to the presence of an interface, one needs many more particles in such a DPD simulation.

Thermodynamic quantities are calculated using only the conservative force. The pressure of the system is calculated using

$$p = \rho k_B T + \frac{1}{3V} \sum_{i>j} \left\langle \mathbf{r}_{ij} \cdot \mathbf{f}_{ij}^C \right\rangle.$$

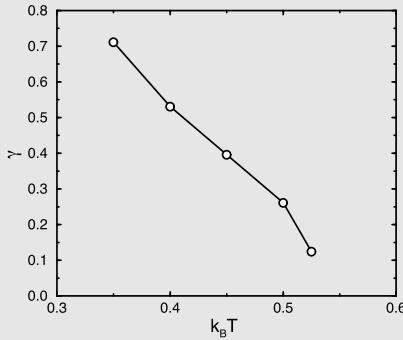


FIGURE 16.2 Surface tension as a function of temperature.

When the DPD fluid undergoes phase separation, two slabs containing the different fluid phases form, and we can compute the associated interfacial tension, using the techniques described in Section 5.1.6. Fig. 16.2 shows this

interfacial tension γ as a function of temperature. Clearly, γ decreases with increasing temperature and should vanish at the critical point. As the critical point is approached, the driving force for the formation of a surface (surface tension) becomes very low, and it becomes difficult, therefore, to form a stable interface in a small system.

For more details, see SI (Case Study 25).

16.1.2 Smoothed dissipative-particle dynamics

The name dissipative particle dynamics implies that during a simulation energy is dissipated. The fact that energy is not conserved in a DPD simulation can be a disadvantage for some applications. For example, if we have a system in which there is a temperature gradient, such a temperature gradient can only be sustained artificially in a DPD simulation [685]. An early solution to this problem was to introduce an additional variable characterizing the internal energy in a DPD simulation [692,693], which corresponds to every DPD particle carrying its own internal-energy “reservoir.” This reservoir absorbs or releases the energy that would normally go into the internal degrees of freedom of the group of molecules that are represented by a single DPD particle. During a collision, the energy of this reservoir can increase or decrease.

Subsequently, Español and Revenga [694] proposed a generic “top-down” approach to treat transport processes in mesoscopic fluid models, based on the ideas of Smoothed Particle Hydrodynamics [695–697]. In this context, top-down means that the model takes the concept of DPD particles as fluid elements seriously, and attributes to them a (constant) mass m , entropy S , and volume V . The energy of these fluid elements is related to the basic thermodynamic parameters through an equation of state of the form $E_i = E(m, S_i, V_i)$, which can be specified by the user to match the properties of the system being modeled. The fact that E depends on time only through the time dependence of m_i , S_i , and V_i implies that local thermodynamic equilibrium is assumed. In the **Smoothed Dissipative Particle Dynamics (SDPD)** approach of ref. [694], the volume of the fluid element is time-dependent, as it depends on the local density around ρ_i around a particle i : $V_i \equiv 1/\rho_i$

$$\rho_i = \int d\mathbf{r} \rho(\mathbf{r}) W(\mathbf{r} - \mathbf{r}_i),$$

where $\rho(\mathbf{r}) \equiv \sum_j \delta(\mathbf{r} - \mathbf{r}_j)$ and $W(\mathbf{r} - \mathbf{r}_i)$ is a normalized weight function:

$$\int d\mathbf{r} W(\mathbf{r} - \mathbf{r}_i) = 1.$$

Note that the density around particle i includes i itself: hence $\rho_i > 0$. A convenient choice for W [694] is given in Appendix H.

In addition, the entropy production in a volume element is given by the standard (macroscopic) equations that follow from irreversible thermodynamics [57], which describe the entropy production due to the (bulk and shear) viscous dissipation and heat flux. Again, the parameters determining these terms can be chosen from the macroscopic transport equations (subject to some constraints). Note that in this approach the entropy flux appears naturally as the advective term that describes how the entropy S_i of the particles is transported.

The important feature of SDPD is that it yields a closed set of equations that describes flow and dissipation and, importantly, like DPD also the effect of thermal noise—so that SDPD describes the dynamics of a hydrodynamic fluid subject to thermal fluctuations—and capable of driving the thermal motion of solutes. The great advantage of SDPD is that it can account for a much wider range of transport phenomena (including thermal) than DPD, and that the parameters can be chosen such that the system can be designed to mimic a liquid with known transport properties and a known equation of state, at least in the limit where discretization errors become unimportant.

However, when the aim of the simulation is to obtain a reasonable, coarse-grained description of a complex, molecular fluid, then the SDPD approach is less suited because all parameters in the model are essentially macroscopic and do not reflect any details of the microscopic intermolecular interactions. For some practical details, we refer the reader to a brief description in Appendix H. For a more extensive discussion, we refer to the paper by Español and Revenga [694] and to the comparative review by Español and Warren [690].

16.2 Multi-particle collision dynamics

The term **Multi-particle Collision Dynamics (MPC)** refers to a particularly simple class of mesoscopic fluid models. In MPC models, particles are propagated ballistically during a fixed time step, and then undergo a momentum and energy-conserving collision with other particles in their neighborhood, defined by cells in a periodic lattice.

A predecessor of the MPC scheme is the so-called **Direct Simulation Monte Carlo Method (DSMC)** method of Bird [698]. This method was developed to simulate the flow of dilute gases. In the DSMC method, particles move ballistically between collisions. Collisions are then carried out stochastically, that is: the number of collisions per time step is fixed by the known collision frequency at the specified density and temperature. To select collision partners, space is divided into cells. Collision partners are then randomly selected pairs within one cell. The precise collision dynamics depends on the molecular model that is used. However, in all cases, collisions conserve linear momentum and energy. For more details on the DSMC method, see refs. [699, 700], and [701].

Malevanets and Kapral [676] built on the DSMC idea to make a simple model of the solvent in a complex liquid [676]. The method of ref. [676] mixes a DSMC-style dynamics and normal MD, as the forces between the macromolecules, and between solute and solvent are taken into account explicitly:

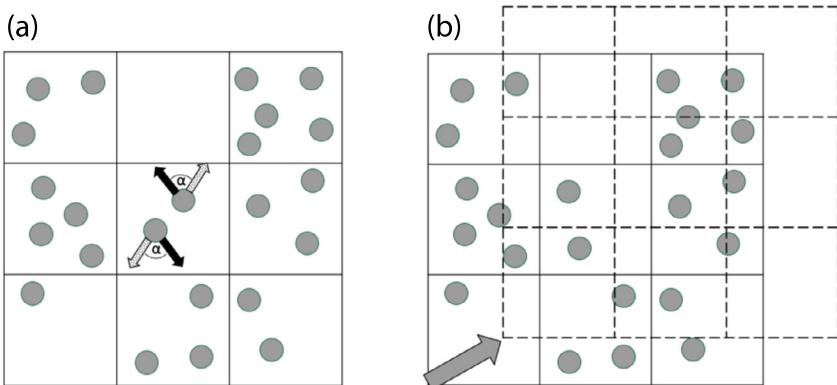


FIGURE 16.3 (a) In an **SRD** simulation, velocities of all particles in a cell, relative to the cell center-of-mass velocity are computed (black arrows in the central cell) and then uniformly rotated over an arbitrary angle α . The new relative velocities are indicated in gray. The uniform rotation of the relative velocities changes neither the total momentum nor the total kinetic energy of the particles in a given cell. (b) To ensure Galilean invariance, the mesh that defines the cell locations is shifted by a random vector (gray arrow) before every multi-particle collision. **SRD** collisions are then carried out between particles in the same new cells (dashed borders).

Molecular Dynamics is used to update the solvent and solute positions and momenta. The solvent-solvent interactions are represented by ballistic propagation interrupted by stochastic collisions: space is divided into cells, and solvent-solvent collisions take place inside these cells. However, unlike **DSMC**, ref. [676] uses many-body collisions between the solvent particles. During such collisions, the motion of the center of mass of the particles in a cell is left unchanged, but a uniform rotation over an angle α around a randomly chosen axis, is applied to the relative velocities of all particles within a given cell (see Fig. 16.3(a)). The rotations carried out in different cells are uncorrelated. As the rotation does not change the magnitude of the relative velocities, the total energy is conserved, while the conservation of the center-of-mass velocity takes care of the momentum conservation. These conservation laws are satisfied irrespective of the length of the **SRD** time steps. However, the transport properties of the **SRD** fluid do depend on the time step and on the average number of particles per cell —typically between 3 and 20 [677].

The **SRD** algorithm does not conserve angular momentum. Usually, this is not a problem, but in some cases, such as two-phase flow in a Couette cell [677], it can lead to unphysical results. If the mesh of the collision cells is kept fixed, the simulations will break Galilean invariance, in particular, if the mean-free path is less than the cell size. To restore Galilean invariance, Ihle and Kroll [702] proposed to apply a random shift to the mesh of collision cells before every collision (see Fig. 16.3(b)).

There exist analytical expressions for the transport properties of the **SRD** gas, which are quite accurate for both high and low densities (see [677]). The

values of the transport coefficients depend on density, temperature, time step, cell size, and on the value of the rotation angle α (see [677]). As the **SRD** gas is ideal, its equation of state is simply the ideal gas law. However, modifications of the **SRD** model that mimic non-ideal gases have been proposed (see e.g., [703]).

A simple way to couple the dynamics of embedded objects, such as polymers, to the solvent is to include the update of the velocity of the monomers in the multi-particle collision step [676]. It is advisable to choose the monomeric mass equal to the average mass of solvent particles in a cell. After the **SRD** collision step, the internal dynamics of the polymer is updated using normal MD. The above approach is less suited for larger objects, such as colloids. In that case, it is better to describe the large particle as a hard object with walls that impose non-slip boundary conditions on the solvent.

To model fluid flow in complex geometries, we must be able to implement boundary conditions at the wall: usually non-slip. This involves two modifications of the algorithm: first of all, **SRD** particles that hit a wall during their ballistic propagation are “bounced back” elastically in the direction where they came from. But if the boundaries run through cells of the mesh, the multi-particle collision step is also changed: for every collision, virtual **SRD** particles with a Maxwell-Boltzmann velocity distribution with zero average momentum are placed inside the part of a cell that is inside the wall. The number of these virtual particles is chosen such that the total number of particles in that cell equals the average [704]. For implementation details and a description of many variants of the **MPC** approach, see ref. [677].

16.3 Lattice-Boltzmann method

The **Lattice-Boltzmann (LB)** method [705–707] is not a particle-based method, although it can be viewed as a pre-averaged version of a lattice-gas cellular automaton model of a fluid [708], which is a (discrete) particle-based model.

The reason for including the **LB** method in this brief description of fluid models for mesoscopic, particle-based simulations is that if thermal fluctuations are added to the LB model of a fluid, it can account for both hydrodynamic interactions and the Brownian motion of dissolved particles. Although LB simulations have now moved away from the underlying **Cellular-Automaton (CA)** picture, it is still helpful to refer to that picture to explain the physical meaning of the LB method.

In a **CA** model, particles live on links connecting different lattice sites. Every lattice point is connected by z links to neighboring lattice points, and possibly also to itself. In two dimensions, z is usually equal to 6 (on a triangular lattice), and in $3d$, a commonly used model has $z = 19$ (an **FCC** lattice with 12 nearest neighbors, 6 next nearest neighbors, and one self-connection). In a single time step, particles move along the link from their current lattice site to the corresponding link on the neighboring lattice site. “Rest particles” stay where they are.

The propagation step is carried out for all particles simultaneously. The next step is the collision step. During collisions, the total number of particles and the total momentum (and, in certain models, the total energy) on a given lattice site is maintained, but apart from this constraint, particles can change their velocity, which is equal to the length of the link divided by the time step. There is considerable freedom in selecting the collision rules, as long as they maintain the full symmetry of the lattice. Moreover, the links must be chosen such that the viscosity of the lattice fluid, which is a fourth-rank tensor, can be made isotropic.

Cellular Automata models can mimic hydrodynamic behavior. However, CAs are “noisy” and suffer from a number of other practical drawbacks. Moreover, by construction, the lattice model lacks Galilean invariance. The [Lattice-Boltzmann \(LB\)](#) method was devised to overcome some (but not all) of the problems of lattice gas cellular automata.

In its most naive version, one can view the LB model as a pre-averaged version of a lattice gas cellular automaton [705]. In this pre-averaging, the number of particles on a given link is replaced by the particle density on that link. Note that the particle number is either zero or one, but the density is a real number. In addition, the resulting equations are greatly simplified if the collision operator (i.e., the function that describes how the post-collision state of a lattice point depends on the pre-collision state) is linearized in the deviation of the particle densities from their (local) equilibrium value. Finally, there is no need to restrict the [LB](#) collision operators to forms that can be derived from an underlying cellular automaton [706,707]. It is, however, essential that the collision operator satisfies the conservation laws and the symmetries of the original model.

For the simulation of complex flows, the LB method is much more efficient than the original cellular automaton model. However, one aspect is missing in the LB approach, namely the intrinsic fluctuations that result from the discreteness in the number of particles. This problem was resolved by Adhikari et al. [709] who showed how fluctuations could be included in a consistent way to the LB models, such that all conservation laws are obeyed, and the fluctuations have all the characteristics of *bona-fide* thermal fluctuations.

The fluctuating LB model of ref. [709] has all the features (hydrodynamic interactions, thermal fluctuations) needed to act as a cheap model of a solvent in a mesoscopic model of a (macro-molecular) solution [710]. Of course, the correct implementation of the boundary conditions between solvent and suspended particles, requires special care —but for these aspects we refer the reader to a review by Ladd [711]. For other aspects of the LB method, the standard work on the subject is the book by Succi [678].

Appendix A

Lagrangian and Hamiltonian equations of motion

Knowledge of Newton's equations of motion is sufficient to understand the basis of the Molecular Dynamics method. However, many of the more advanced simulation techniques make use of the Lagrangian or Hamiltonian formulations of classical mechanics. Here, we briefly sketch the relation between these different approaches (see also [712]). For a more detailed and more rigorous description of classical mechanics, the reader is referred to the book by Goldstein [54].

A.1 Action

The Lagrangian formulation of classical mechanics is based on a variational principle. The actual trajectory followed by a classical system in a time interval $\{t_b, t_e\}$, between an initial position x_b and a final position x_e , is the one for which the action, S , is an extremum (usually, a minimum). The classical action S for an arbitrary trajectory is defined as the time integral of the difference between the kinetic energy \mathcal{K} and the potential energy \mathcal{U} of the system, computed along that trajectory:

$$S = \int_{t_b}^{t_e} dt [\mathcal{K} - \mathcal{U}]. \quad (\text{A.1.1})$$

Before considering the general Lagrangian equations of motion that follow from this extremum principle, let us first consider a few simple examples.

The first case is that of a single particle that moves in the absence of an external potential, i.e., $\mathcal{U} = 0$. As the particle has to move from x_b to x_e in a time interval $t_e - t_b$, we already know its average velocity: v_{av} . If the particle would always move with this average velocity, it would follow a straight trajectory that we denote by $\bar{x}(t)$. Let us denote the true trajectory of the particle by $x(t) = \bar{x}(t) + \eta(t)$, where $\eta(t)$ is, as yet, unknown. Then the velocity of the particle is the sum of the average velocity v_{av} and the deviation from it, $\dot{\eta}(t)$:

$$v(t) = v_{\text{av}} + \dot{\eta}(t).$$

By construction,

$$\int dt \dot{\eta}(t) = 0.$$

In the present example, the potential energy is always zero, and hence the action S is determined by the time integral of the kinetic energy:

$$S = \frac{1}{2}m \int dt [v_{av} + \dot{\eta}(t)]^2 = S_{av} + \frac{1}{2}m \int dt \dot{\eta}^2(t).$$

Since the last term can never be less than zero, the action has its minimum if $\dot{\eta}(t) = 0$ for all t . In other words, we recover the well-known result that, in the absence of external forces, the particle moves with constant velocity. This is Newton's first law.

Next, consider a particle moving in a one-dimensional potential $U(x)$. In that case, the action is

$$S = \int_{t_b}^{t_e} dt \left[\frac{1}{2}m \left(\frac{dx(t)}{dt} \right)^2 - U(x) \right].$$

An arbitrary path, $x(t)$, can be written as the sum of the actual path that a classical particle will follow, $\bar{x}(t)$, plus the deviation from this path $\eta(t)$:

$$x(t) = \bar{x}(t) + \eta(t).$$

As before, we impose the initial and final positions of the particle and hence $\eta(t_b) = \eta(t_e) = 0$. For paths $x(t)$ that are close to the actual path, we can expand the action in powers of the (small) quantity $\eta(t)$. Actually, as $\eta(t)$ is itself a function of t , such an expansion is called a *functional expansion*. The action is extremal if the leading (linear) terms in this functional expansion vanish. Let us now consider the functional expansion of the action around the action of the true path to linear order in $\eta(t)$:

$$\begin{aligned} S &= \int_{t_b}^{t_e} dt \frac{1}{2}m \left(\frac{d\bar{x}(t)}{dt} + \frac{d\eta(t)}{dt} \right)^2 - U[\bar{x}(t) + \eta(t)] \\ &\approx \int_{t_b}^{t_e} dt \frac{1}{2}m \left[\left(\frac{d\bar{x}(t)}{dt} \right)^2 + 2 \frac{d\bar{x}(t)}{dt} \frac{d\eta(t)}{dt} \right] - \left[U(\bar{x}(t)) + \frac{\partial U(\bar{x})}{\partial x} \eta(t) \right] \\ &= \bar{S} + \int_{t_b}^{t_e} dt \left[m \frac{d\bar{x}(t)}{dt} \frac{d\eta(t)}{dt} - \frac{\partial U(\bar{x})}{\partial x} \eta(t) \right] \\ &= \bar{S} + m \frac{d\bar{x}(t)}{dt} \eta(t) \Big|_{t_b}^{t_e} - \int_{t_b}^{t_e} dt \left[m \frac{d^2 \bar{x}(t)}{dt^2} + \frac{\partial U(\bar{x})}{\partial x} \right] \eta(t), \end{aligned}$$

where the last step has been obtained via partial integration. Since by definition, $\eta(t) = 0$ at the boundaries, the second term on the right-hand side vanishes. The action has its extremum if the integrand in the last line of the above equation vanishes for arbitrary $\eta(t)$. This condition can be satisfied if and only if

$$m \frac{d^2 \bar{x}(t)}{dt^2} = -\frac{\partial U(\bar{x})}{\partial x}, \quad (\text{A.1.2})$$

which is Newton's second law. In other words, Newton's equations of motion can be derived from the statement that a particle follows a path for which the action is an extremum.

A.2 Lagrangian

There would be little point in introducing this alternative expression of the laws of classical mechanics if it did not allow us to do more than simply rederive $F = ma$. In fact, the Lagrangian formulation of classical mechanics turns out to be very powerful. For one thing, the Lagrangian approach makes it easy to derive equations of motion in non-Cartesian coordinate frames. Suppose that we wish to use some generalized coordinates q instead of the Cartesian coordinate x . For example, consider a pendulum of length l in a uniform gravitational field. The angle that the pendulum makes with the vertical (i.e., with the direction of the gravitational field) can be used to specify its orientation. Since the path that the pendulum follows is clearly independent of the coordinates that we happen to use to specify its state, the action, S , should be the same:

$$S = \int dt \mathcal{L}(x, \dot{x}) = \int dt \mathcal{L}(q, \dot{q}), \quad (\text{A.2.1})$$

where the quantity \mathcal{L} is called the Lagrangian. The Lagrangian is defined as the kinetic energy minus the potential energy¹:

$$\mathcal{L} \equiv \mathcal{K}(\dot{q}) - \mathcal{U}(q). \quad (\text{A.2.2})$$

We again introduce our actual path $\bar{q}(t)$ and the deviation $\eta(t)$ from it:

$$\begin{aligned} q(t) &= \bar{q}(t) + \eta(t) \\ \dot{q}(t) &= \dot{\bar{q}}(t) + \dot{\eta}(t). \end{aligned}$$

We can write for the Lagrangian, \mathcal{L} ,

$$\mathcal{L}(q, \dot{q}) = \mathcal{L}(\bar{q}, \dot{\bar{q}}) + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial \dot{q}} \dot{\eta}(t) + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial q} \eta(t).$$

As in the previous section, we use the functional expansion of S in powers of $\eta(t)$ to derive an expression for the classical path. To this end, we substitute the Lagrangian in the expression for the action (A.2.1). Next, we write a possible path of the particle as the sum of the actual path and a correction $\eta(t)$. As before, we use partial integration and use the fact that $\eta(t)$ vanishes at the boundaries of the integration. It then follows that the action has an extremum if

$$\int dt \left[-\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial \dot{q}} \right) + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial q} \right] \eta(t) = 0, \quad (\text{A.2.3})$$

¹ The correct definition is more restrictive; see [54] for more details.

which is satisfied for arbitrary $\eta(t)$ if and only if

$$\left[-\frac{d}{dt} \left(\frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial \dot{q}} \right) + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial q} \right] = 0. \quad (\text{A.2.4})$$

This is the Lagrangian equation of motion. To cast this equation of motion in a more familiar form, we introduce the generalized momentum p associated with the generalized coordinate q :

$$p \equiv \frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}}. \quad (\text{A.2.5})$$

Substitution of this expression into Eq. (A.2.4) yields

$$\dot{p} = \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q}. \quad (\text{A.2.6})$$

As the above formulation is valid for any coordinate system, it should certainly hold for Cartesian coordinates. In these coordinates the Lagrangian reads

$$\mathcal{L}(x, \dot{x}) = \frac{1}{2}m\dot{x}^2 - U(x).$$

The momentum associated with x is

$$p_x = \frac{\partial \mathcal{L}(x, \dot{x})}{\partial \dot{x}} = m\dot{x}$$

and the equation of motion is

$$m\ddot{x} = -\frac{\partial U(x)}{\partial x},$$

which is indeed the result we would obtain from Newton's equation of motion.

Illustration 24 (A pendulum in a gravitational field). Consider a simple pendulum of length l with mass m (see Fig. A.1). A uniform gravitational field is acting on the pendulum and the potential energy is a simple function of the angle θ that the pendulum makes with the vertical:

$$U(\theta) = mgl[1 - \cos(\theta)].$$

We wish to express the equations of motion in terms of the generalized coordinate θ . The Lagrangian, \mathcal{L} , is

$$\begin{aligned} \mathcal{L} &= \mathcal{K} - \mathcal{U} = \frac{1}{2}m[\dot{x}^2(t) + \dot{y}^2(t)] - U(\theta) \\ &= \frac{ml^2}{2}\dot{\theta}^2 - U(\theta). \end{aligned}$$

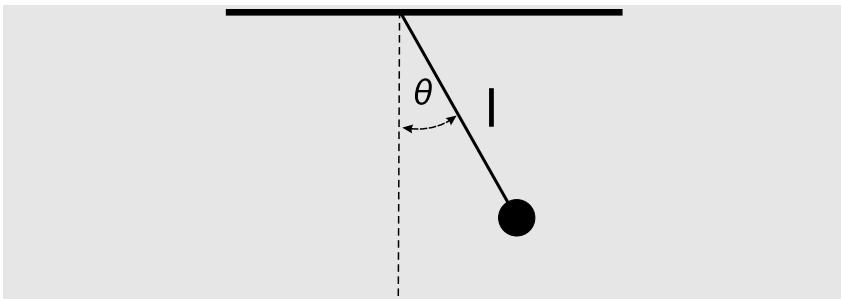


FIGURE A.1 A simple pendulum of length l with mass m .

The generalized momentum is defined as

$$p_\theta = \frac{\partial \mathcal{L}}{\partial \dot{q}} = ml^2 \dot{\theta}$$

and the equation of motion follows from Eq. (A.2.6)

$$\dot{p}_\theta = -\frac{\partial U(\theta)}{\partial \theta}$$

or

$$\ddot{\theta} = -\frac{1}{ml^2} \frac{\partial U(\theta)}{\partial \theta}.$$

A.3 Hamiltonian

Using the Lagrangian, we obtain equations of motion in terms of q and \dot{q} . Often, it is convenient to express the equations of motion in terms of q and its conjugate momentum p . To do this, we can perform a Legendre transformation²:

$$\mathcal{H}(q, p) \equiv p\dot{q} - \mathcal{L}(q, \dot{q}, t). \quad (\text{A.3.1})$$

² In thermodynamics, Legendre transforms are used to derive various thermodynamic potentials. For example, the energy E is a natural function of the entropy S and volume V : $E = E(S, V)$, i.e., in these variables, E is a thermodynamic potential. In most practical applications, it is more convenient to have the temperature T rather than the entropy S as the independent variable. Since the temperature is the variable conjugate to the entropy ($\partial E / \partial S = T$), we can perform a Legendre transform to remove the S dependence:

$$A \equiv E - TS,$$

yielding

$$dA = dE - d(TS) = -SdT - pdV.$$

For historical reasons, the Legendre transform linking the Lagrangian to the Hamiltonian has the opposite sign.

This equation defines the *Hamiltonian* \mathcal{H} of the system. As \mathcal{H} is a function of q , p , and, in general, also of t , it is clear that we can write an infinitesimal variation of \mathcal{H} as

$$d\mathcal{H}(q, p) = \frac{\partial \mathcal{H}}{\partial p} dp + \frac{\partial \mathcal{H}}{\partial q} dq + \frac{\partial \mathcal{H}}{\partial t} dt. \quad (\text{A.3.2})$$

But, using the definition of \mathcal{H} , we can also write

$$\begin{aligned} d\mathcal{H}(q, p) &= d(p\dot{q}) - d\mathcal{L}(q, \dot{q}) \\ &= pd\dot{q} + \dot{q}dp - \left[\frac{\partial \mathcal{L}}{\partial q} dq + \frac{\partial \mathcal{L}}{\partial \dot{q}} d\dot{q} + \frac{\partial \mathcal{L}}{\partial t} dt \right] \\ &= pd\dot{q} + \dot{q}dp - \dot{p}dq - pd\dot{q} - \frac{\partial \mathcal{L}}{\partial t} dt \\ &= \dot{q}dp - \dot{p}dq - \frac{\partial \mathcal{L}}{\partial t} dt, \end{aligned}$$

where we have used the definitions of p and \dot{p} , Eqs. (A.2.5) and (A.2.6), respectively. It then follows directly that

$$\frac{\partial \mathcal{H}}{\partial p} = \dot{q} \quad (\text{A.3.3})$$

$$\frac{\partial \mathcal{H}}{\partial q} = -\dot{p}. \quad (\text{A.3.4})$$

These are the desired equations of motion in terms of q , p . For most systems that we consider in this book, the Lagrangian does not explicitly depend on time. Under those circumstances, the Hamiltonian is conserved. This follows directly from the equations of motion:

$$\begin{aligned} \frac{d\mathcal{H}(q, p)}{dt} &= \frac{\partial \mathcal{H}}{\partial p} \dot{p} + \frac{\partial \mathcal{H}}{\partial q} \dot{q} \\ &= -\frac{\partial \mathcal{H}}{\partial p} \frac{\partial \mathcal{H}}{\partial q} + \frac{\partial \mathcal{H}}{\partial q} \frac{\partial \mathcal{H}}{\partial p} \\ &= 0. \end{aligned}$$

This conservation law expresses the fact that, in a closed system, the total energy is conserved. In Cartesian coordinates, the Hamiltonian can be written as

$$\begin{aligned} \mathcal{H}(x, p_x) &= \dot{x}p_x - \mathcal{L}(x, \dot{x}) \\ &= m\dot{x}^2 - \frac{1}{2}m\dot{x}^2 + U(x) \\ &= \frac{1}{2m}p_x^2 + U(x), \end{aligned}$$

and the Hamiltonian equations of motion reduce to Newton's equations

$$\begin{aligned}\dot{x} &= \frac{\partial \mathcal{H}}{\partial p_x} = \frac{p_x}{m} \\ \dot{p}_x &= -\frac{\partial \mathcal{H}}{\partial x} = -\frac{\partial U(x)}{\partial x}.\end{aligned}$$

The Hamiltonian equations of motion are two first-order differential equations —one for p and one for q . In contrast, the Lagrangian formalism yields a single second-order equation. However, both formalisms yield identical results. The choice between the two is dictated by considerations of mathematical convenience.

Example 28 (A pendulum in a gravitational field: Part II). We consider again the simple pendulum in a uniform gravitational field, introduced in Example 24:

$$U(\theta) = mgl[1 - \cos(\theta)],$$

where θ is the angle that the pendulum makes with the vertical and g the gravitational acceleration.

In Example 24 we have derived the equations of motion from the Lagrangian in terms of a second-order differential equation in θ . Now we will use Hamilton's formulation.

The Lagrangian is

$$\mathcal{L}(\theta, \dot{\theta}) = \mathcal{U}_K - \mathcal{U}_P = \frac{ml^2}{2}\dot{\theta}^2 - U(\theta).$$

The Lagrangian depends on the variables θ and $\dot{\theta}$ and in the Hamiltonian language we want to express the equations of motion in terms of θ and its conjugate momentum p_θ . This conjugate momentum is defined by Eq. (A.2.5)

$$p_\theta \equiv \frac{\partial \mathcal{L}(\theta, \dot{\theta})}{\partial \dot{\theta}} = ml^2\dot{\theta}.$$

The Hamiltonian follows from the Legendre transformation (A.3.1)

$$\begin{aligned}\mathcal{H} &= p_\theta \dot{\theta} - \mathcal{L}(\theta, \dot{\theta}) \\ &= \frac{p_\theta}{2ml^2} + U(\theta) \\ &= \frac{1}{2}ml^2\dot{\theta}^2 + U(\theta),\end{aligned}$$

which is, of course, equal to the total energy of the pendulum.

The equations of motion follow from Eqs. (A.3.3) and (A.3.4):

$$\dot{\theta} = \frac{\partial \mathcal{H}}{\partial p_\theta} = \frac{P_\theta}{ml^2}$$

$$\dot{p}_\theta = -\frac{\partial \mathcal{H}}{\partial \theta} = -\frac{dU(\theta)}{d\theta},$$

which are the desired equations of motion in terms of two first-order differential equations.

A.4 Hamilton dynamics and statistical mechanics

The choice between the Hamiltonian and Lagrangian formulations of classical mechanics is determined by considerations of convenience. One example of a case where the Lagrangian formalism is more convenient is in the derivation of the equations of motion of a system with constraints (see section 14.1). On the other hand, the Hamiltonian expressions are to be used when establishing the connection with statistical mechanics (see Chapter 2).

A.4.1 Canonical transformation

In the Hamiltonian formulation, the generalized coordinates and momenta are independent variables. One can therefore introduce a transformation of both variables simultaneously. For example, the transformation of the coordinates q, p to Q, P is denoted by

$$\begin{aligned} Q &= Q(q, p) \\ P &= P(q, p) \end{aligned} \tag{A.4.1}$$

and the inverse transformation, Q, P into q, p , by

$$\begin{aligned} q &= q(Q, P) \\ p &= p(Q, P). \end{aligned} \tag{A.4.2}$$

Obviously, the value of any function of the phase-space coordinates is unaffected by the coordinate transformation. In the case of the Hamiltonian, this implies that

$$\mathcal{H}(q, p) \equiv \mathcal{H}[Q(p, q), P(q, p)] \equiv \mathcal{H}'(Q, P). \tag{A.4.3}$$

In general, the equations of motion in the new coordinates are not of the canonical form, unless the coordinate transformation is *canonical*.³ If the coordinate transformation is canonical, the equations of motion for the new phase-space coordinates Q, P are

$$\dot{Q} = \left(\frac{\partial \mathcal{H}'(Q, P)}{\partial P} \right) \tag{A.4.4}$$

³ As we assume that time does not appear explicitly in these equations, we are defining a so-called restricted canonical transformation.

$$\dot{P} = - \left(\frac{\partial \mathcal{H}'(Q, P)}{\partial Q} \right). \quad (\text{A.4.5})$$

From Eq. (A.4.1) and the Hamilton equations of motion for the coordinates q , p , it follows that

$$\begin{aligned}\dot{Q} &= \left(\frac{\partial Q(q, p)}{\partial q} \right) \dot{q} + \left(\frac{\partial Q(q, p)}{\partial p} \right) \dot{p} \\ &= \left(\frac{\partial Q(q, p)}{\partial q} \right) \left(\frac{\partial \mathcal{H}(q, p)}{\partial p} \right) - \left(\frac{\partial Q(q, p)}{\partial p} \right) \left(\frac{\partial \mathcal{H}(q, p)}{\partial q} \right).\end{aligned}$$

Using Eq. (A.4.3), we can write

$$\left(\frac{\partial \mathcal{H}'(Q, P)}{\partial P} \right) = \left(\frac{\partial \mathcal{H}(q, p)}{\partial p} \right) \left(\frac{\partial p(P, Q)}{\partial P} \right) + \left(\frac{\partial \mathcal{H}(q, p)}{\partial q} \right) \left(\frac{\partial q(P, Q)}{\partial P} \right).$$

This equation can only be equal to expression (A.4.4) for \dot{Q} if

$$\left(\frac{\partial Q(q, p)}{\partial q} \right) = \left(\frac{\partial p(Q, P)}{\partial P} \right) \text{ and } \left(\frac{\partial Q(q, p)}{\partial p} \right) = - \left(\frac{\partial q(Q, P)}{\partial P} \right). \quad (\text{A.4.6})$$

Similarly, we can start with \dot{P} , and derive two other conditions:

$$\left(\frac{\partial P(q, p)}{\partial q} \right) = - \left(\frac{\partial p(Q, P)}{\partial Q} \right) \text{ and } \left(\frac{\partial P(q, p)}{\partial p} \right) = \left(\frac{\partial q(Q, P)}{\partial Q} \right). \quad (\text{A.4.7})$$

These two equations define the condition for a canonical transformation.

A.4.2 Symplectic condition

We can express the above conditions for a canonical transformation in a single equation, by using a matrix notation. Let Γ be a $2dN$ -dimensional vector containing the generalized coordinates q_i and momenta p_i of the N particles in d dimensions (see Section 2.5.1). Hamilton's equations of motion (A.3.3) and (A.3.4) can be written as

$$\dot{\Gamma} = \omega \frac{\partial \mathcal{H}}{\partial \Gamma}, \quad (\text{A.4.8})$$

where ω is an antisymmetric matrix defined as

$$\omega = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$

In a similar way, we can define ξ to be the $2N$ -dimensional vector containing the generalized coordinates Q_i and P_i . Using the matrix notation the transfor-

mation (A.4.1) from Q, P to q, p is written as

$$\xi = \xi(\Gamma).$$

For the time derivatives of ξ , we can write

$$\dot{\xi} = M\dot{\Gamma},$$

where M is the Jacobian matrix of the transformation. The elements of this matrix are

$$M_{ij} = \frac{\partial \xi_i}{\partial \Gamma_j}. \quad (\text{A.4.9})$$

We can write, using Eq. (A.4.8), for the time derivatives of ξ

$$\dot{\xi} = M\omega \frac{\partial \mathcal{H}}{\partial \Gamma}. \quad (\text{A.4.10})$$

In a similar way, we can define the inverse transformation (A.4.2)

$$\Gamma = \Gamma(\xi).$$

Since

$$\mathcal{H}(p, q) = \mathcal{H}(P, Q),$$

we can write

$$\frac{\partial \mathcal{H}(\Gamma)}{\partial \Gamma_i} = \sum_j \frac{\partial \mathcal{H}(\xi)}{\partial \xi_j} \frac{\partial \xi_j}{\partial \Gamma_i}. \quad (\text{A.4.11})$$

If we define the transposed matrix⁴ of M as defined in Eq. (A.4.9),

$$\tilde{M}_{ij} = \frac{\partial \xi_j}{\partial \Gamma_i}.$$

This allows us to rewrite Eq. (A.4.11) in matrix notation as

$$\frac{\partial \mathcal{H}(\Gamma)}{\partial \Gamma} = \tilde{M} \frac{\partial \mathcal{H}(\xi)}{\partial \xi}. \quad (\text{A.4.12})$$

If we combine Eqs. (A.4.10) and (A.4.12), we have

$$\dot{\xi} = M\omega \tilde{M} \frac{\partial \mathcal{H}}{\partial \xi}.$$

This expression for the equations of motion is valid for any set of variables ξ that are being transformed (independently of time) from the set Γ . Such a

⁴ One can obtain the transposed matrix of a given matrix A by interchanging rows and columns, i.e., $\tilde{a}_{ij} = a_{ji}$.

transformation is canonical if the equations of motion in the new coordinates have the canonical form:

$$\dot{\xi} = \omega \frac{\partial \mathcal{H}}{\partial \xi}.$$

This can only be the case if M satisfies the condition

$$M\tilde{\omega}M = \omega. \quad (\text{A.4.13})$$

This condition is often called the *symplectic condition*. A matrix M that satisfies this condition is called a *symplectic matrix*.⁵

A.4.3 Statistical mechanics

Using the symplectic notation for a canonical transformation, we consider the implications for statistical mechanics. In the microcanonical ensemble, the classical partition function Ω of a three-dimensional atomic system is defined as

$$\Omega_{N,V,E} = \frac{1}{h^{3N} N!} \int d\mathbf{p}^N d\mathbf{q}^N \delta(\mathcal{H}(p, q) - E), \quad (\text{A.4.14})$$

where h is Planck's constant and the delta-function restricts the integration to the hypersurface in phase space defined by $\mathcal{H}(p, q) = E$. We can re-express this integral in terms of other phase-space coordinates, but then we have to take into account that a volume element in the two coordinate sets needs not to be the same. The volume element associated with Γ is

$$d\Gamma = dq_1 \dots dq_N dp_1 \dots dp_N$$

and to ξ

$$d\xi = dQ_1 \dots dQ_N dP_1 \dots dP_N.$$

These two volume elements are related via the Jacobian matrix of the transformation matrix

$$d\Gamma = |\text{Det}(M)| d\xi. \quad (\text{A.4.15})$$

This equation shows that, in general, a coordinate transformation will result in the appearance of a Jacobian in the partition function:

$$\Omega_{N,V,E} = \frac{1}{h^{3N} N!} \int d\mathbf{P}^N d\mathbf{Q}^N |\text{Det}(M)| \delta(\mathcal{H}'(P, Q) - E). \quad (\text{A.4.16})$$

⁵ To see that this condition is identical to Eqs. (A.4.6) and (A.4.7), we have to multiply this equation from the right with the inverse matrix of \tilde{M} :

$$M\omega = \omega \tilde{M}^{-1}.$$

When computing ensemble averages in coordinate systems other than the original Cartesian one, the Jacobian of the transformation, M , may be different from one, and should be taken into account. In what follows, we denote the Jacobian $|\text{Det}(M)|$ by the symbol ω .

For a canonical transformation, i.e., obeys condition (A.4.13), the absolute value of the Jacobian is one. To derive this result, we take the determinant of both sides of the symplectic condition (A.4.13)

$$\begin{aligned}\text{Det}(M\tilde{\omega}M) &= \text{Det}(\omega) \\ \text{Det}^2(M)\text{Det}(\omega) &= \text{Det}(\omega).\end{aligned}$$

This equation can only be true if the determinant of M is ± 1 , which implies that for a canonical transformation the absolute value of the Jacobian associated with this transformation must be one.

The natural time evolution in phase space of a classical system may be considered as a coordinate transformation:

$$\Gamma(t_0) \rightarrow \Gamma(t).$$

One important property of a Hamiltonian system is that the natural time evolution corresponds to a symplectic coordinate transformation. We can consider the transformation from $\Gamma(t_0)$ to $\Gamma(t)$ as a sequence of infinitesimal transformations with time step δt . Suppose that we define the evolution of the coordinates during the time interval δt as a transformation of coordinates from Γ to ξ :

$$\begin{aligned}\xi &= \xi(\Gamma) \\ &= \Gamma(t + \delta t) \\ &= \Gamma(t) + \dot{\Gamma}(t)\delta t.\end{aligned}$$

The Jacobian of this transformation is

$$\begin{aligned}M &\equiv \frac{\partial \xi}{\partial \Gamma} \\ &= 1 + \delta t \frac{\partial}{\partial \Gamma} \left(\omega \frac{\partial \mathcal{H}}{\partial \Gamma} \right) \\ &= 1 + \delta t \omega \frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma},\end{aligned}$$

where

$$\left(\frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma} \right)_{ij} = \frac{\partial^2 \mathcal{H}}{\partial \Gamma_i \partial \Gamma_j}.$$

Taking into account that ω is an antisymmetric matrix, we can write for the transpose of the matrix M :

$$\tilde{M} = 1 - \frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma} \omega.$$

Substitution of this expression for the Jacobian into the symplectic condition (A.4.13) yields (to first order in δt)

$$\begin{aligned} M\omega\tilde{M} &= \left(1 + \delta t \omega \frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma}\right) \omega \left(1 - \delta t \frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma} \omega\right) \\ &\approx \omega + \delta t \omega \frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma} \omega - \omega \delta t \frac{\partial^2 \mathcal{H}}{\partial \Gamma \partial \Gamma} \omega \\ &= \omega. \end{aligned}$$

Hence the symplectic condition holds for the evolution of Γ during an infinitesimal time interval. As we can consider the evolution of Γ during a finite interval, as a sequence of canonical transformations of infinitesimal steps, the total time evolution also satisfies the symplectic condition.

One may view the Hamiltonian as the generator of a canonical transformation acting on all points in phase space. As the Jacobian of a canonical transformation is equal to 1, the size of a volume element in phase space does not change during the natural time evolution of a Hamiltonian system. Moreover, the density $f(q(t), p(t))$ around any point in phase space also remains constant during the time evolution. To see this, consider a volume V in phase space bounded by a surface S . During time evolution, the surface moves, and so do all points inside the surface. However, a point cannot cross the surface. The reason is simple: if two trajectories in phase space crossed, it would imply that two trajectories start from the same phase-space point. But this is impossible, as it would mean that a trajectory starting from this point is not uniquely specified by its initial conditions. Hence, the number of phase-space points inside any volume does not change in time. As the volume itself is also constant, this implies that the phase-space density (i.e., the number of points per unit volume) is constant. In other words: the phase-space density of a Hamiltonian system behaves like an incompressible fluid:

$$\frac{df}{dt} = 0. \quad (\text{A.4.17})$$

While the exact solution of Hamilton's equations of motion will satisfy the incompressibility condition, discrete, numerical schemes will—in general—violate it. As before, we can consider any numerical MD algorithm (e.g., Verlet, velocity Verlet, ...) as a transformation from $(q(t), p(t))$ to $(q(t + \Delta t), q(t + \Delta t))$. We can then compute the Jacobian of this transformation, and

check whether it is equal to 1 (see sections 4.3 and 4.3.4). For all “good” algorithms to solve Newton’s equations of motion, the Jacobian of the transformation from $(q(t), p(t))$ to $(q(t + \Delta t), p(t + \Delta t))$ is equal to 1 —such algorithms are said to be “area-preserving.” It should be noted that the symplectic condition implies more than just the area-preserving properties. Unfortunately, these other consequences do not have such a simple intuitive interpretation. When we say that it is desirable that an algorithm be symplectic, we mean more than that it should be area-preserving —it should really satisfy the symplectic condition. Fortunately, in many cases, the symplectic nature of an algorithm is easy to demonstrate by making use of the fact that *any* set of classical Hamiltonian equations of motion satisfies the symplectic condition. An algorithm that can be written as a sequence of exact time evolutions generated by simple Hamiltonians is, therefore, necessarily symplectic. An example is the Verlet algorithm. As discussed in section 4.3.4, this algorithm can be viewed as a sequence of exact propagations using either the kinetic part of the Hamiltonian or the potential part. Either propagation satisfies the symplectic condition. Hence the Verlet algorithm as a whole is symplectic. For an accessible discussion of symplectic dynamics in general, see ref. [713]. A discussion of symplectic integrators for Molecular Dynamics simulations can be found in ref. [714].

Appendix B

Non-Hamiltonian dynamics

A systematic procedure for extending the techniques of classical statistical mechanics to non-Hamiltonian systems was proposed by Tuckerman et al. [267, 715, 716]. In the present appendix, we sketch the general approach for analyzing extended Lagrangian systems. We will, however, skip most of the derivations. For a more complete and more rigorous derivation, using the mathematical techniques of differential geometry, the reader is referred to the original references.

In general, the dynamics that results from solving non-Hamiltonian equations of motion is not area-preserving. As we have seen in Appendix A.4.3, solving the equations of motion can be considered as a coordinate transformation from the phase-space coordinates at time t_0 to those at time t . If the system is Hamiltonian, the time evolution of the system will change the shape of an infinitesimal volume element in phase space, but not its volume $d\Gamma$. In contrast, for a non-Hamiltonian system, we have to take into account the Jacobian of the transformation associated with the evolution of $d\Gamma(t_0) \rightarrow d\Gamma(t)$:

$$d\Gamma_t = J(\Gamma_t; \Gamma_0) d\Gamma_0,$$

where the subscript 0 indicates the phase-space volume at $t = t_0$ and J is the determinant of the Jacobian matrix M of the transformation. For convenience, we choose $t_0 = 0$.

The motion in phase space of a Hamiltonian system is similar to that of an incompressible liquid: in time the volume of this “liquid” does not change. In contrast, a non-Hamiltonian system is compressible. This compressibility must be taken into account when considering the generalization of the Liouville theorem to non-Hamiltonian systems.

The compressibility can be derived from the time dependence of the Jacobian

$$\frac{dJ(\Gamma_t; \Gamma_0)}{dt} = \kappa(\Gamma_t, t) J(\Gamma_t; \Gamma_0) \quad (\text{B.0.1})$$

in which $\kappa(\Gamma_t, t)$, the *phase space compressibility* of the dynamical system, is defined:

$$\kappa(\Gamma_t, t) \equiv \nabla_{\Gamma} \cdot \dot{\Gamma}. \quad (\text{B.0.2})$$

Eq. (B.0.1) has as formal solution

$$J(\Gamma_t; \Gamma_0) = \exp \left(\int_0^t \kappa(\Gamma_s, s) ds \right).$$

If we define $w(\Gamma_t, t)$ as the primitive function of $\kappa(\Gamma_t, t)$, then we can write

$$\begin{aligned} J(\Gamma_t; \Gamma_0) &= \exp [w(\Gamma_t, t) - w(\Gamma_0, 0)] \\ &\equiv \frac{\sqrt{g(\Gamma_0, 0)}}{\sqrt{g(\Gamma_t, t)}}, \end{aligned}$$

where the last line defines the quantity \sqrt{g} . Recall that

$$\begin{aligned} d\Gamma_t &= J(\Gamma_t; \Gamma_0) d\Gamma_0 \\ &= \frac{\sqrt{g(\Gamma_0, 0)}}{\sqrt{g(\Gamma_t, t)}} d\Gamma_0. \end{aligned}$$

Hence,

$$\sqrt{g(\Gamma_t, t)} d\Gamma_t = \sqrt{g(\Gamma_0, 0)} d\Gamma_0,$$

which defines an invariant measure in phase space. This result can be used to derive a new form of the Liouville equation for non-Hamiltonian systems. The important point here is that the phase-space distribution, $f(\Gamma)$, the function in which we are interested, which gives the probability density in phase space, should be kept separate from the phase-space metric, \sqrt{g} , which ensures that the volume of phase space of a non-Hamiltonian system is invariant under time evolution,

$$\frac{\partial f \sqrt{g}}{\partial t} + \nabla \cdot (f \sqrt{g} \dot{\Gamma}) = 0. \quad (\text{B.0.3})$$

The expression corresponding to an ensemble average is

$$\langle A \rangle = \frac{\int d\Gamma \sqrt{g(\Gamma)} A(\Gamma) f(\Gamma)}{\int d\Gamma \sqrt{g(\Gamma)} f(\Gamma)}. \quad (\text{B.0.4})$$

Assuming that there are n_c conservation laws, $\Lambda_k(\Gamma') = C_k$ for $k = 1, \dots, n_c$, the partition function of the non-Hamiltonian system is given by

$$\Omega(C_1, \dots, C_n) = \int d\Gamma' \sqrt{g(\Gamma')} \prod_{k=1}^{n_c} \delta(\Lambda_k(\Gamma') - C_k). \quad (\text{B.0.5})$$

In many applications, one obtains the correct (NVT or NPT) partition function from the above ‘‘microcanonical’’ partition function, by carrying out the integration over the unphysical variables that have been introduced to represent the effect of a thermostat or barostat. In order to do this properly, it is essential

to identify *all* conservation laws. Moreover, it is useful to eliminate from the analysis all those coordinates that are linearly dependent on other variables and variables that are “driven.” Variables are called “driven” when they do not influence the time evolution of (and are not coupled through a conservation law) the physical variables of interest in the system, even though their own time evolution may depend on these last variables.

Appendix C

Kirkwood-Buff relations

C.1 Structure factor for mixtures

In section 5.1.7.1 we discussed the relation between the structure factor $S(\mathbf{q})$, Eq. (5.1.40), of a one-component system and the mean-squared value of the Fourier transform of the particle density (Eq. (5.1.38)):

$$\rho(\mathbf{q}) = \sum_{i=1}^N e^{i\mathbf{q}\cdot\mathbf{r}_i} = \int_V d\mathbf{r} \rho(\mathbf{r}) e^{i\mathbf{q}\cdot\mathbf{r}}.$$

For an n -component system, we can generalize this relation to yield expressions for the partial structure factors $S_{ab}(\mathbf{q})$ that measure the cross-correlations between fluctuations in the Fourier transforms of the densities of species a and b :

$$\begin{aligned} S_{ab}(\mathbf{q}) &= \frac{1}{\sqrt{\langle N_a \rangle \langle N_b \rangle}} \langle \delta\rho_a(\mathbf{q}) \delta\rho_b(-\mathbf{q}) \rangle \\ &= \frac{1}{\sqrt{\langle N_a \rangle \langle N_b \rangle}} \int_V \int_V d\mathbf{r} d\mathbf{r}' \langle \delta\rho_a(\mathbf{r}) \delta\rho_b(\mathbf{r}') \rangle e^{i\mathbf{q}\cdot(\mathbf{r}-\mathbf{r}')} , \end{aligned} \quad (\text{C.1.1})$$

where $\delta\rho(\mathbf{q})_i$ denotes the fluctuation of $\rho(\mathbf{q})$ around its average value. If we take the limit $q \rightarrow 0$ in the second half of Eq. (C.1.1), we get:

$$\lim_{q \rightarrow 0} S_{ab}(\mathbf{q}) = \frac{\langle \Delta N_a \Delta N_b \rangle}{\sqrt{\langle N_a \rangle \langle N_b \rangle}} \quad (\text{C.1.2})$$

where ΔN_a denotes the fluctuation in the total number of particles of species a in the system. Following Kirkwood and Buff [717], we show that Eq. (C.1.2) has a direct thermodynamic interpretation, which provides a powerful route to determine the composition-dependence of chemical potentials in solution. For a multi-component system, we can generalize the expression for the Grand-Canonical partition, Ξ , Eq. (2.3.19) to

$$\Xi(\{\mu\}, V, T) \equiv \sum_{N_1, N_2, \dots, N_n=0}^{\infty} \prod_{a=1}^n \exp(\beta\mu_a N_a) e^{-\beta F(\{N\}, V, T)}, \quad (\text{C.1.3})$$

where $\{\mu\}$ denotes $\mu_1, \mu_2, \dots, \mu_n$ and $\{N\}$ stands for N_1, N_2, \dots, N_n . The Grand Potential, $\Phi = \Phi(\{\mu\}, V, T)$ is given by

$$\Phi = -k_B T \ln \Xi(\{\mu\}, V, T).$$

From Eq. (C.1.3), it then follows that

$$\left(\frac{\partial \Phi}{\partial \mu_a} \right) = -\langle N_a \rangle \quad (\text{C.1.4})$$

and

$$\left(\frac{\partial^2 \Phi}{\partial \mu_a \partial \mu_b} \right) = - \left(\frac{\partial N_a}{\partial \mu_b} \right) = -\beta \langle \Delta N_a \Delta N_b \rangle_{\{\mu\}, V, T}. \quad (\text{C.1.5})$$

Comparing Eq. (C.1.5) with Eq. (C.1.2) indicates that there is a close relation between the behavior of the structure factors $S_{ab}(q)$ in the limit $q \rightarrow 0$, and the thermodynamic derivatives of the Grand Potential.

Kirkwood and Buff were the first to propose these relations [717]. However, they also expressed their results in terms of integrals over the radial distribution functions $g_{ab}(r)$, and many simulation studies use the $g(r)$ -approach to compute the composition-dependence of chemical potential. The $g(r)$ -based relations are correct in principle, but as explained below Eq. (5.1.42), they are very dangerous to use in simulations on small (and not even very small) systems. It is, therefore, better to stick with Eq. (C.1.2) [718].¹

We still have to explain why the relations Eq. (C.1.5) are important. The key reason is that they allow us to compute the variation with the composition of the chemical potentials of the various species in a multi-component mixture under conditions where particle-insertion methods (see section 8.5.1) fail.

We first note that at constant T and V :

$$d\Phi = - \sum_{r=1}^n N_r d\mu_r$$

and, hence

$$\left(\frac{\partial \Phi}{\partial N_a} \right)_{T, V, N'} = - \sum_{r=1}^n \langle N_r \rangle \left(\frac{\partial \mu_r}{\partial N_a} \right)_{T, V, N'},$$

where N' and μ' denote the set of all $\{N\}$ and $\{\mu\}$ that are not being varied in that particular differentiation (i.e., we use the same symbol for different sets). It then follows that

$$\begin{aligned} \left(\frac{\partial^2 \Phi}{\partial N_a \partial N_b} \right)_{T, V, N'} &= \sum_{r,s=1}^n \left(\frac{\partial^2 \Phi}{\partial \mu_r \partial \mu_s} \right)_{T, V, \mu'} \left(\frac{\partial \mu_r}{\partial N_a} \right)_{T, V, N'} \left(\frac{\partial \mu_s}{\partial N_b} \right)_{T, V, N'} \\ &\quad - \sum_{r=1}^n \langle N_r \rangle \left(\frac{\partial^2 \mu_r}{\partial N_a \partial N_b} \right)_{T, V, N'}. \end{aligned} \quad (\text{C.1.6})$$

¹ Ref. [719] describes an alternative approach to exploit the Kirkwood-Buff relations in systems where the total numbers of molecules of each species are fixed.

From the Gibbs-Duhem equation ($d\Phi_{V,T} = -\sum_r N_r d\mu_r$) it follows that

$$\left(\frac{\partial^2 \Phi}{\partial N_a \partial N_b} \right)_{T,V,N'} = - \left(\frac{\partial \mu_a}{\partial N_b} \right)_{T,V,N'} - \sum_{r=1}^n N_r \left(\frac{\partial^2 \mu_r}{\partial N_a \partial N_b} \right)_{T,V,N'}. \quad (\text{C.1.7})$$

Combining Eqs. (C.1.6) and (C.1.7), we get

$$\left(\frac{\partial \mu_a}{\partial N_b} \right)_{T,V,N'} = - \sum_{r,s=1}^n \left(\frac{\partial^2 \Phi}{\partial \mu_r \partial \mu_s} \right)_{T,V,\mu'} \left(\frac{\partial \mu_r}{\partial N_a} \right)_{T,V,N'} \left(\frac{\partial \mu_s}{\partial N_b} \right)_{T,V,N'}. \quad (\text{C.1.8})$$

Note that Eq. (C.1.8) has the form of a matrix equation

$$A_{ab} = A_{ar} B_{rs} A_{sb},$$

where

$$A_{ab} = A_{ba} = \left(\frac{\partial \mu_a}{\partial N_b} \right)_{T,V,N'},$$

and

$$B_{rs} = B_{sr} = \beta \langle \Delta N_r \Delta N_s \rangle_{\{\mu\}, V, T}.$$

But then, $\mathbf{A} = \mathbf{B}^{-1}$ or, in compact notations,

$$\beta \left(\frac{\partial \mu_r}{\partial N_b} \right)_{T,V,N'} \langle \Delta N_b \Delta N_s \rangle_{T,V,\mu} = \delta_{rs}, \quad (\text{C.1.9})$$

where δ_{rs} is the Kronecker δ -function. In other words, once we have computed the matrix of cross-correlations in the number fluctuations, then we know the variation with the composition of the chemical potentials of all species. Note that the A -matrices refer to an ensemble where T , V , and N' are fixed, whereas B expresses the fluctuations in an ensemble at constant μ , V , T .

C.2 Kirkwood-Buff in simulations

To compute the phase behavior of mixtures, we need to know the variation of the chemical potentials with composition. For dense liquids, we cannot compute μ_i using the particle-insertion method (8.5.1), or using grand-canonical simulations (6.5) because the probability of successful insertions becomes very small. It is under such conditions that Eq. (C.1.2) becomes useful: if the number of particles of the different species is fixed, $S_{ab}(\mathbf{q} = 0)$ vanishes identically, but we can compute the limit $S_{ab}(\mathbf{q} = 0)$ for $q \rightarrow 0$, and this limit is well defined and, apart from possible finite size effects, equal to the desired value.

Appendix D

Non-equilibrium thermodynamics

D.1 Entropy production

When a system relaxes from a non-equilibrium situation, entropy is produced. Irreversible thermodynamics, as formulated by Onsager [55,56], establishes relations between the different contributions to the entropy production. As a starting point, we, therefore, need an expression for the entropy production. The “canonical” derivation for the expression of the entropy production is given in the book on *Non-equilibrium Thermodynamics* by De Groot and Mazur [57]. However, the derivation of the expression for entropy production in ref. [57], whilst complete, is a bit daunting. Here, we have opted for a different route: it is less complete, but it is quick, and it makes it easier to understand what is going on.

Let us first briefly look at the entropy generation that takes place if we bring two sub-systems (I and II) of a closed system into contact. The subsystems can exchange energy and particles. We ignore the fact that the medium may flow—and that is the price we have to pay for keeping things simple. In reality, viscous flow is an important non-equilibrium phenomenon, and the way to include it in the expression for the entropy production is described in ref. [57].

The change of the entropy, S of system I due to an infinitesimal exchange would be:

$$dS^I = \frac{1}{T^I} dU^I - \sum_i \frac{\mu_i^I}{T^I} dN_i^I. \quad (\text{D.1.1})$$

As the total system is closed, we have $dU^I = -dU^{II}$, and $dN_i^{II} = -dN_i^I$. Hence, the total change in entropy is

$$dS^{\text{total}} = dS^I + dS^{II} = \left[\frac{1}{T^I} - \frac{1}{T^{II}} \right] dU^I - \sum_i \left[\frac{\mu_i^I}{T^I} - \frac{\mu_i^{II}}{T^{II}} \right] dN_i^I. \quad (\text{D.1.2})$$

This equation shows that entropy is produced if there is an energy flux in a system with a gradient in $1/T$ or if there is a particle flux in a system with a gradient in μ_i/T . Now consider the case that we have a thin slab of material with width dx and cross-section A and that heat flows into the system at temperature

$T(x)$ and out of it at temperature $T(x + dx)$. We denote the amount of heat transported per unit of time by \dot{Q} . The heat flux is then $\mathbf{j}_q = \dot{Q}/A$.

Similarly, the number of particles of type i flowing through the system is \dot{N}_i and the particles flux of species i is $\mathbf{j}_i = \dot{N}_i/A$. Note that we assume that the system as a whole is at rest and, therefore, the sum of all particle fluxes must vanish.¹

We can then write the rate of entropy production $\dot{\mathcal{S}}$ as

$$\dot{\mathcal{S}} = \left(\frac{\partial 1/T(x)}{\partial x} \right) dx A \mathbf{j}_q - \sum_i \left(\frac{\partial \mu_i(x)/T(x)}{\partial x} \right) dx A \mathbf{j}_i , \quad (\text{D.1.3})$$

or, in terms of σ , the entropy production per unit volume:

$$\sigma = \left(\frac{\partial 1/T(x)}{\partial x} \right) \mathbf{j}_q - \sum_i \left(\frac{\partial \mu_i(x)/T(x)}{\partial x} \right) \mathbf{j}_i . \quad (\text{D.1.4})$$

We can generalize this slightly by dropping the assumption that the transport is one-dimensional, and we can combine the gradient of the chemical potential with other forces \mathbf{F}_i acting on species i , due to gradients in external potentials. We then obtain

$$\sigma = \mathbf{J}_q \cdot \nabla \frac{1}{T} - \frac{1}{T} \sum_i \mathbf{J}_i \cdot \left(T \nabla \frac{\mu_i}{T} - \mathbf{F}_i \right) . \quad (\text{D.1.5})$$

For reasons to be discussed below, it is convenient to separate the gradient of μ_i/T into a part that depends on the temperature gradient and a part that does not. If the chemical potential depends on temperature and pressure, we can use

$$\left(\frac{\partial \beta \mu_i}{\partial \beta} \right)_{P, \{N_j\}} = h_i , \quad (\text{D.1.6})$$

where $\beta \equiv 1/k_B T$, and

$$\sigma = \left(\mathbf{J}_q - \sum_i \mathbf{J}_i h_i \right) \cdot \nabla \frac{1}{T} - \frac{1}{T} \sum_i \mathbf{J}_i \cdot [(\nabla \mu_i)_T - \mathbf{F}_i] \quad (\text{D.1.7})$$

and this allows us to define the ‘non-diffusive’, irreversible heat flow \mathbf{J}'_q

$$\mathbf{J}'_q \equiv \mathbf{J}_q - \sum_i \mathbf{J}_i h_i . \quad (\text{D.1.8})$$

¹ The most elegant choice for the particle fluxes would be the mass flux, in which case the condition that the sum of fluxes must vanish means that the center of mass is stationary. However, in practice, it makes little difference which fluxes we use. We will use number densities. The thing to bear in mind is that the definition of the chemical potential (e.g., per particle, per unit mass, or per mole) has to be consistent with the choice of the fluxes.

In what follows, we shall use the symbol \mathbf{J}_h to denote the enthalpy flux:

$$\mathbf{J}_h \equiv \sum_i \mathbf{J}_i h_i . \quad (\text{D.1.9})$$

D.1.1 Enthalpy fluxes

Why is it important to subtract the enthalpy fluxes? First of all: as de Groot and Mazur state, heat fluxes in mixtures are not uniquely defined. The same holds for enthalpy fluxes. The value of the enthalpy flux depends on where we choose the zero of the energy of the particles. We could, for instance, include the $m_i c^2$ associated with the rest-mass of particle i . This is not as silly as it seems because if we are, for instance, pumping UF_6 or a similar nuclear fuel, the energy flux does take the rest of the mass into account.

The key point is that the “internal energy” choice makes a huge difference in what we mean by “transferring a particle at constant energy.” If we move particle i from reservoir 1 to reservoir 2, without moving energy, then removing particle i from reservoir 1 will lead to a large increase in the entropy (because the energy that used to be in the rest mass of particle i is now taken up by the bath of other particles), and conversely, the entropy of system 2 will decrease very substantially to compensate for the energy gain associated with the introduction of particle i . It is convenient to view this process *as if* we allow particle i to retain its enthalpy but then add an enthalpy flux from 2 to 1 that exactly compensates for this. Schematically, the problem of transferring a particle at constant energy is sketched in Fig. D.1.

In particular, we could consider the case where, in addition, there is a heat flux $\sum_i \mathbf{J}_i h_i$ from 1 to 2. This would correspond to the case that we do not force the particle to extract energy when it leaves 1 and to add energy when it arrives in 2. In this particular case, the contribution of diffusive enthalpy transport to the net ‘irreversible’ heat flow \mathbf{J}'_q is zero. Importantly, if we thus allow particles to travel with their associated enthalpy, the entire problem of the reference state disappears. Of course, in reality, there still will be a heat flux, but this is the heat associated with thermal motion and intermolecular interactions.

To summarize: the rate of entropy production is given by

$$\sigma = \mathbf{J}'_q \cdot \nabla \frac{1}{T} - \frac{1}{T} \sum_i \mathbf{J}_i \cdot [(\nabla \mu_i)_T - \mathbf{F}_i] \quad (\text{D.1.10})$$

where the irreversible heat flow depends neither on the enthalpy carried by the particle fluxes \mathbf{J}_i .

D.2 Fluctuations

In the previous sections, we argued that the equilibrium state of a system corresponds to the state with the largest number of microscopic realizations. More-



FIGURE D.1 Transferring particles from system 1 at temperature T_1 to system 2 at temperature T_2 , without changing the energy of either system, requires that every particle leaves *all* energy (even kinetic) behind in 1 and acquires its new energy at the expense of system 2.

over, we argued that we could link this probabilistic picture to the experimental observation that the entropy of a closed system is at a maximum if we identify the entropy S with $k_B \ln \Omega$:

$$S = k_B \ln \Omega . \quad (\text{D.2.1})$$

We now assume that the entropy is a unique function of n linearly independent extensive variables, denoted by $\{A_1, A_2, \dots, A_n\}$. In equilibrium, the entropy of an isolated system must be at a maximum. If A_i denotes a non-conserved quantity (e.g., the degree of crystallinity), then the second law implies that, at equilibrium,

$$\frac{\partial S_t}{\partial A_i} = 0 , \quad (\text{D.2.2})$$

where S_t denotes the entropy of the entire system. However, if A_i denotes a conserved quantity, then it cannot change in a closed system. In that case, we can consider the entropy variation as an amount dA_i is transferred from subsystem

1 to subsystem 2. Then

$$\frac{\partial S_t}{\partial A_i^{(2)}} = \frac{\partial S_2}{\partial A_i^{(2)}} - \frac{\partial S_1}{\partial A_i^{(1)}} = 0, \quad (\text{D.2.3})$$

because $dA_i^{(2)} = -dA_i^{(1)}$. More generally, if we consider m subsystems, there are $m - 1$ independent variables $A_i^{(n)}$ if A is conserved, and m if A is not conserved. For the conserved quantities, Eq. (D.2.3) simply expresses the equality of T , P/T , or μ_i/T in the two subsystems. So, S does not vary to linear order in A_i . However, to quadratic order, S does vary with A_i .² Hence, to quadratic order, we can write

$$S = S_0 + \frac{1}{2} \sum_{i,j} \frac{\partial^2 S_t}{\partial A_i \partial A_j} \alpha_i \alpha_j, \quad (\text{D.2.4})$$

where $\alpha_i \equiv A_i - A_i^0$. To keep our notation aligned with De Groot and Mazur [57], we write

$$g_{ij} \equiv -\frac{\partial^2 S_t}{\partial A_i \partial A_j}. \quad (\text{D.2.5})$$

If the α_i are linearly independent, g_{ij} is a symmetric, positive definite matrix. The probability to find a system in a state away from (but close to) the most probable state and characterized by the variables $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ is

$$P(\{\alpha_1, \alpha_2, \dots, \alpha_k\}) \sim \exp \left(-\frac{1}{2k_B} \sum_{i,j} g_{ij} \alpha_i \alpha_j \right). \quad (\text{D.2.6})$$

As the second law states that a system will evolve from a less probable to a more probable state, we can define the “driving force” that causes the system to return to its most probable state. With every variable α_i , there is an associated driving force X_i given by

$$X_i = -\frac{1}{2} \frac{\partial \sum_{i,j} g_{ij} \alpha_i \alpha_j}{\partial \alpha_i} = -\sum_j g_{ij} \alpha_j. \quad (\text{D.2.7})$$

To define the driving forces X_i , we made use of the assumption of *local* thermodynamic equilibrium, meaning that locally the relations between the entropy S and the basic extensive thermodynamic quantities U , V , and M_i (the mass of component i) are the same as during a quasi-static process. We do not know *a priori* how quickly a system returns to equilibrium. A separate set of constitutive

² We assume that S is an analytical function of the A_i . This may seem reasonable, but it need not always be true.

equations is needed to describe the relation between the driving forces X_j and the “fluxes” $J_i \equiv \dot{\alpha}_i$. We assume that, to lowest order, these relations are of the form:

$$J_i = \sum_j L_{ij} X_j . \quad (\text{D.2.8})$$

At this stage, we know nothing about the transport coefficients L_{ij} . Onsager [55] assumed that the law describing the rate of decay of a variable α_i to its equilibrium value (zero) is valid for arbitrarily small α_i and therefore also describes the rate at which spontaneous fluctuations around equilibrium decay to their average value. This “Onsager Regression Hypothesis” provides a link between the macroscopic transport coefficient and the microscopic dynamics of a system in equilibrium. The regression hypothesis can be viewed as a generalization of Einstein’s assumption that diffusive transport can be treated as a macroscopic manifestation of Brownian motion.

From Eq. (D.2.4) it is easy to derive an expression for the entropy production \dot{S} :

$$\dot{S} = - \sum_{i,j} g_{ij} \dot{\alpha}_i \alpha_j = \sum_i X_i \cdot \dot{\alpha}_i = \sum_{ij} L_{ij} X_i X_j . \quad (\text{D.2.9})$$

D.3 Onsager reciprocal relations

To make contact with the decay of fluctuations in equilibrium, we now show that the equilibrium fluctuations α_i are only correlated to the fluctuations in the *conjugate* force X_i

$$\langle \alpha_i X_j \rangle = -k_B \delta_{ij} \quad (\text{D.3.1})$$

where k_B is Boltzmann’s constant and δ_{ij} is the Kronecker delta. Equation (D.3.1) follows from the fact that

$$\begin{aligned} \langle \alpha_i X_j \rangle &= k_B \int d\{\alpha\} \alpha_i \frac{\partial P(\{\alpha\})}{\partial \alpha_j} \\ &= -k_B \int d\{\alpha\} \frac{\partial \alpha_i}{\partial \alpha_j} P(\{\alpha\}) \\ &= -k_B \delta_{ij} , \end{aligned} \quad (\text{D.3.2})$$

where $\{\alpha\}$ denotes the set $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$. Using $J_i = \sum_j L_{ij} X_j$, it then follows that

$$\langle \alpha_j(t) J_i(t) \rangle = -k_B L_{ij} . \quad (\text{D.3.3})$$

Equation (D.3.3) allows us to derive the Onsager reciprocal relations. But before we do so, note that microscopically, Eq. (D.3.3) is a bit strange, because a fluctuation at time t will only result in a flux for $t > 0$. In fact, as α_j and J_j

have different time-reversal symmetries, the equal-time product actually vanishes. The quantity that is non-zero is $\langle \alpha_j(t) J_i(t + \epsilon) \rangle$. We will come back to this point later. For the time being, we continue with Eq. (D.3.3) and rewrite it as:

$$\int_0^\infty dt \langle \alpha_j(0) \dot{J}_i(t) \rangle = -k_B L_{ij}, \quad (\text{D.3.4})$$

where we have made used the fact that $\langle \alpha_j(0) \dot{J}_i(0) \rangle$ vanishes. Next, using time invariance,

$$\int_0^\infty dt \langle \dot{\alpha}_j(0) J_i(t) \rangle = +k_B L_{ij}, \quad (\text{D.3.5})$$

and, using the fact that $J_j = \dot{\alpha}_j$:

$$\int_0^\infty dt \langle J_j(0) J_i(t) \rangle = k_B L_{ij}. \quad (\text{D.3.6})$$

The important thing is that (classically):

$$\langle J_j(0) J_i(t) \rangle = \langle J_i(t) J_j(0) \rangle = \langle J_i(0) J_j(-t) \rangle.$$

Here, and in what follows, we will limit the discussion to fluxes that all have the same time-reversal symmetry, in which case $\langle J_j(0) J_i(t) \rangle = \langle J_i(0) J_j(t) \rangle$, from which it then follows that

$$L_{ij} = L_{ji}. \quad (\text{D.3.7})$$

Appendix E

Non-equilibrium work and detailed balance

The relation between free-energy differences and (non-equilibrium) work described in section 8.7 holds for all protocols or equations of motion that are markovian and satisfy detailed balance for every step.

To see what that means, consider a process where work is performed by changing a control parameter Λ . For instance, Λ might be the system's volume, or it might be a parameter in the Hamiltonian of the system. We denote the original/final value of Λ as Λ_0/Λ_K (we use the index K to keep the subsequent notation consistent).

The protocol to change Λ from Λ_0 to Λ_K can be decomposed in two types of elementary steps: during the first, all phase-space coordinates of the system (Γ) are kept fixed, and Λ is changed by an amount $\Delta\Lambda_i$, where i labels the step; if there are K such steps, then $i = \{1, 2, \dots, K\}$. Note that steps that only change Λ are deterministic. However, there is also a second type of step in the protocol, namely one where Λ is kept fixed, but the system is allowed to evolve by its natural dynamics, exchanging energy with a thermostat. Examples of the evolution at constant Λ are sequences of one or more Monte Carlo moves, or of one or more time steps in a constant-temperature MD simulation.¹ In the language of Crooks [389], we denote the energy exchanged with the “reservoir” (i.e., the thermostat) by Q : it can be interpreted as the heat delivered to the system by the reservoir. Because of detailed balance, the ratio of the probability of the system to evolve at constant Λ (say Λ_i) from phase-space coordinate Γ to Γ' to the probability of the reverse move, is given by

$$\frac{P(\Gamma \rightarrow \Gamma'; \Lambda_i)}{P(\Gamma' \rightarrow \Gamma; \Lambda_i)} = e^{-\beta \Delta E(\Lambda_i)}, \quad (\text{E.1.1})$$

where $\Delta E(\Lambda_i) \equiv E(\Gamma'; \Lambda_i) - E(\Gamma; \Lambda_i)$.

When we change Λ at constant Γ , we perform work on the system. We denote the work associated with a change Λ from Λ_{i-1} to Λ_i at fixed Γ_i by w_i . The total work W done on the system as Λ is increased from Λ_0 to Λ_K , is then equal to $W = \sum_{i=1}^K w_i$.

¹ The argument can be generalized to the case where we also exchange volume or particles with a reservoir, but here we consider the simplest case.

If the time evolution of the system is Markovian, we can write the probability to evolve from Γ_0 at Λ_0 to Γ_K at Λ_K as

$$\prod_{i=0}^{K-1} [P(\Gamma_i \rightarrow \Gamma_{i+1}; \Lambda_i) \times 1] .$$

We included a factor one in every step to indicate that changing Λ at constant Γ is deterministic. In what follows, we leave out this trivial factor. We can also write down the probability to evolve from Γ_K at Λ_K to Γ_0 at Λ_0 along the same path, as

$$\prod_{i=K-1}^0 P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i) .$$

Because of Eq. (E.1.1), we have

$$\prod_{i=0}^{K-1} P(\Gamma_i \rightarrow \Gamma_{i+1}; \Lambda_i) = \prod_{i=K-1}^0 \left[P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i) e^{-\beta \Delta E(\Lambda_i)} \right]. \quad (\text{E.1.2})$$

Note that $\prod_{i=K-1}^0 e^{-\beta \Delta E(\Lambda_i)}$ is equal to $e^{-\beta Q(\{\Gamma\})}$, where Q is the total energy transferred from the reservoir for a sequence of states $\{\Gamma\} \equiv \Gamma_0, \Gamma_1, \dots, \Gamma_K$. Note that Q is *not* equal to $E(\Gamma_K; \Lambda_K) - E(\Gamma_0; \Lambda_0)$ because the energy of the system is also changed by performing work:

$$E(\Gamma_K; \Lambda_K) - E(\Gamma_0; \Lambda_0) = W(\{\Gamma\}) + Q(\{\Gamma\}), \quad (\text{E.1.3})$$

which can be viewed as a microscopic version of the First Law of Thermodynamics. Note, in particular, that W and Q are path-dependent, but their sum is not. If we sample initial conditions from a Boltzmann distribution, then we can express the average of $e^{-\beta W}$ as

$$\overline{e^{-\beta W}} = \sum_{\Gamma_0, \dots, \Gamma_K} P_B(\Gamma_0, \Lambda_0) \prod_{i=0}^{K-1} P(\Gamma_i \rightarrow \Gamma_{i+1}; \Lambda_i) e^{-\beta W(\{\Gamma\})}, \quad (\text{E.1.4})$$

where $P_B(\Gamma_0, \Lambda_0) = \exp(-\beta[E(\Gamma_0; \Lambda_0) - F(\Lambda_0)])$, and the Helmholtz free energy F is, as usual, given by

$$\beta F(\Lambda) = -\ln \sum_{\Gamma} e^{-\beta [E(\Gamma; \Lambda)]}. \quad (\text{E.1.5})$$

Using Eq. (E.1.2), we can write

$$\frac{\prod_{i=0}^{K-1} P(\Gamma_i \rightarrow \Gamma_{i+1}; \Lambda_i)}{\prod_{i=K-1}^0 P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i)} = e^{-\beta Q(\{\Gamma\})} \quad (\text{E.1.6})$$

or, using the definition of the Boltzmann weights:

$$\frac{P_B(\Gamma_0, \Lambda_0) \prod_{i=0}^{K-1} P(\Gamma_i \rightarrow \Gamma_{i+1}; \Lambda_i)}{P_B(\Gamma_K, \Lambda_K) \prod_{i=K-1}^0 P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i)} = e^{-\beta Q(\{\Gamma\})} e^{+\beta[\Delta E - \Delta F]}, \quad (\text{E.1.7})$$

where $\Delta E \equiv E(\Gamma_K; \Lambda_K) - E(\Gamma_0; \Lambda_0)$, and $\Delta F \equiv F(\Lambda_K) - F(\Lambda_0)$. We can then write:

$$\begin{aligned} & P_B(\Gamma_0, \Lambda_0) \prod_{i=0}^{K-1} P(\Gamma_i \rightarrow \Gamma_{i+1}; \Lambda_i) e^{-\beta W(\{\Gamma\})} \\ &= P_B(\Gamma_K, \Lambda_K) \prod_{i=K-1}^0 P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i) e^{-\beta[Q(\{\Gamma\}) + W(\{\Gamma\}) - \Delta E]} e^{-\beta \Delta F} \\ &= P_B(\Gamma_K, \Lambda_K) \prod_{i=K-1}^0 P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i) e^{-\beta \Delta F}, \end{aligned} \quad (\text{E.1.8})$$

where we have used Eq. (E.1.3). It then follows that Eq. (E.1.4) can be written as

$$\overline{e^{-\beta W}} = e^{-\beta \Delta F} \sum_{\Gamma_0, \dots, \Gamma_K} \prod_{i=K-1}^0 P(\Gamma_{i+1} \rightarrow \Gamma_i; \Lambda_i) = e^{-\beta \Delta F}, \quad (\text{E.1.9})$$

where the last equality follows from the fact that all transition probabilities are normalized.

Appendix F

Linear response: examples

F.1 Dissipation

Many experimental techniques probe the dynamics of a many-body system by measuring the absorption of some externally applied field (e.g., visible light, infrared radiation, microwave radiation). Linear response theory allows us to establish a simple relation between the absorption spectrum and the Fourier transform of a time-correlation function. To see this, let us again consider an external field that is coupled to a dynamical variable $A(\mathbf{p}^N, \mathbf{q}^N)$. The time-dependent Hamiltonian, \mathcal{H} , of the system is

$$\mathcal{H}(t) = \mathcal{H}_0 - f(t)A(\mathbf{p}^N, \mathbf{q}^N).$$

Note that the only quantity explicitly time-dependent is $f(t)$. As the Hamiltonian depends on time, the total energy E of the system also changes with time:

$$E(t) = \langle \mathcal{H}(t) \rangle.$$

Let us compute the average rate of change of the energy of the system. This is the amount of energy absorbed (or emitted) by the system, which per unit of time reads:

$$\begin{aligned} \frac{\partial E}{\partial t} &= \left\langle \frac{d\mathcal{H}}{dt} \right\rangle \\ &= \left\langle \sum_i \left(\dot{\mathbf{q}}_i \frac{\partial \mathcal{H}}{\partial \mathbf{q}_i} + \dot{\mathbf{p}}_i \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i} \right) + \frac{\partial \mathcal{H}}{\partial t} \right\rangle. \end{aligned} \quad (\text{F.1.1})$$

But, from Hamilton's equations of motion, we have

$$\dot{\mathbf{q}}_i = \frac{\partial \mathcal{H}}{\partial \mathbf{p}_i}$$

and

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{H}}{\partial \mathbf{q}_i}.$$

As a consequence, Eq. (F.1.1) simplifies to

$$\frac{\partial E}{\partial t} = \left\langle \frac{\partial \mathcal{H}}{\partial t} \right\rangle$$

608 Linear response: examples

$$\begin{aligned}
&= - \left\langle \dot{f}(t) A(\mathbf{p}^N, \mathbf{q}^N) \right\rangle \\
&= - \dot{f}(t) \langle A(t) \rangle.
\end{aligned} \tag{F.1.2}$$

Note, however, that $\langle A(t) \rangle$ itself is the response to the applied field f :

$$\langle A(t) \rangle = \int_{-\infty}^{\infty} dt' \chi_{AA}(t-t') f(t').$$

Let us now consider the situation where $f(t)$ is a periodic field with frequency ω (e.g., monochromatic light). In that case, we can write $f(t)$ as

$$f(t) = \operatorname{Re} f_\omega e^{i\omega t}$$

and

$$\dot{f}(t) = \frac{i\omega}{2} \left(f_\omega e^{i\omega t} - f_\omega^* e^{-i\omega t} \right).$$

The average rate of energy dissipation is

$$\begin{aligned}
\frac{\partial E}{\partial t} &= - \dot{f}(t) \langle A(t) \rangle \\
&= - \dot{f}(t) \int_{-\infty}^{\infty} dt' \chi_{AA}(t-t') f(t').
\end{aligned} \tag{F.1.3}$$

For a periodic field, we have

$$\begin{aligned}
\int_{-\infty}^{\infty} dt' \chi_{AA}(t-t') f(t') &= \frac{f_\omega e^{i\omega t}}{2} \int_{-\infty}^{\infty} dt' \chi_{AA}(t-t') e^{i\omega(t'-t)} \\
&\quad + \frac{f_\omega^* e^{-i\omega t}}{2} \int_{-\infty}^{\infty} dt' \chi_{AA}(t-t') e^{-i\omega(t'-t)} \\
&= \pi \left[f_\omega e^{i\omega t} \chi_{AA}(\omega) + f_\omega^* e^{-i\omega t} \chi_{AA}(-\omega) \right],
\end{aligned} \tag{F.1.4}$$

where

$$\chi_{AA}(\omega) \equiv \frac{1}{2\pi} \int_0^{\infty} dt \chi_{AA}(t) e^{-i\omega t}.$$

To compute \dot{E} , the rate of change of the energy, we must average $\langle \partial \mathcal{H} / \partial t \rangle$ over one period, $T (= 2\pi/\omega)$, of the field:

$$\begin{aligned}
\dot{E} &= \frac{-\pi}{2T} \int_0^T dt \left\{ i\omega (f_\omega e^{i\omega t} - f_\omega^* e^{-i\omega t}) \right. \\
&\quad \times \left. \left[f_\omega e^{i\omega t} \chi_{AA}(\omega) + f_\omega^* e^{-i\omega t} \chi_{AA}(-\omega) \right] \right\} \\
&= -\pi \omega |f_\omega|^2 \frac{\chi_{AA}(\omega) - \chi_{AA}(-\omega)}{2i}
\end{aligned}$$

$$= -\pi\omega |f_\omega|^2 \operatorname{Im}[\chi_{AA}(\omega)]. \quad (\text{F.1.5})$$

We use the relation between $\chi_{AA}(t)$ and the autocorrelation function (2.5.17) of A :

$$\chi_{AA}(\omega) = \frac{1}{2\pi} \int_0^\infty dt e^{-i\omega t} [-\beta \langle A(0) \dot{A}(t) \rangle].$$

The imaginary part of $\chi_{AA}(\omega)$ is given by

$$\begin{aligned} \operatorname{Im}[\chi_{AA}(\omega)] &= \frac{\beta}{2\pi} \int_0^\infty dt \sin(\omega t) \langle A(0) \dot{A}(t) \rangle \\ &= -\frac{\beta}{4\pi} \int_{-\infty}^\infty dt \omega \cos(\omega t) \langle A(0) A(t) \rangle. \end{aligned} \quad (\text{F.1.6})$$

Finally, we obtain

$$\dot{E} = |f_\omega|^2 \frac{\beta\omega^2}{4} \int_{-\infty}^\infty dt \cos(\omega t) \langle A(0) A(t) \rangle. \quad (\text{F.1.7})$$

Hence, from knowledge of the autocorrelation function of the quantity that couples with the applied field, we can compute the shape of the absorption spectrum. This relation was derived assuming classical dynamics and therefore is valid only if $\hbar\omega \ll k_B T$. However, it is also possible to derive a quantum-mechanical version of linear response theory that is valid for arbitrary frequencies (see, e.g., [53]).

To give a specific example, let us compute the shape of the absorption spectrum of a dilute gas of polar molecules. In that case, the relevant correlation function is the dipole (μ) autocorrelation function:

$$\langle M_x(0) M_x(t) \rangle = \frac{N}{3} \langle \boldsymbol{\mu}(0) \cdot \boldsymbol{\mu}(t) \rangle.$$

For molecules that rotate almost freely (*almost*, otherwise there would be no dissipation), $\boldsymbol{\mu}(0) \cdot \boldsymbol{\mu}(t)$ depends on time, because each molecule rotates. For a molecule with a rotation frequency ω , we have

$$\boldsymbol{\mu}(0) \cdot \boldsymbol{\mu}(t) = \mu^2 \cos(\omega t),$$

and for an assembly of molecules with a thermal distribution of rotational velocities $P(\omega)$, we have

$$\langle \boldsymbol{\mu}(0) \cdot \boldsymbol{\mu}(t) \rangle = \mu^2 \int d\omega P(\omega) \cos(\omega t).$$

The rate of absorption of radiation is then given by

$$\dot{E} = \frac{\pi\beta\omega^2 N \mu^2}{12} P(\omega) |f_\omega|^2. \quad (\text{F.1.8})$$

610 Linear response: examples

For more details about the relation between spectroscopic properties and time correlation functions, the reader is referred to the article by Madden in [44].

F.2 Electrical conductivity

In the derivation of linear response theory in Chapter 2, we assumed that we prepare the system in an equilibrium state with the perturbation *on* and then allow the system to relax to a new equilibrium state with the perturbation *off*. However, this will not always work. Consider, for instance, electrical conductivity. In that case, the perturbation is an electrical field that will cause a current to flow in the system. Hence, the state in which we prepared the system with the field on is *not* an equilibrium state but a steady *non-equilibrium* state. The same holds, for instance, for a system under steady shear. It would seem that, in such circumstances, one cannot use the framework of linear response theory in its simplest form to derive transport coefficients such as the electrical conductivity σ_e or the viscosity η . Fortunately, things are not quite as bad as that. Consider, for example, electrical conductivity. Indeed, if we put a conducting system in an external field, we will generate a non-equilibrium steady state. However, what we *can* is perturbing the system by switching on a weak, uniform *vector potential* \mathbf{A} . The Hamiltonian of the system with the vector potential switched on is

$$\mathcal{H}' = \sum_{i=1}^N \frac{1}{2m_i} \left(\mathbf{p}_i - \frac{e_i}{c} \mathbf{A} \right)^2 + \mathcal{U}_{\text{pot}}. \quad (\text{F.2.1})$$

The system described by this Hamiltonian satisfies the same equations of motion as the unperturbed system (\mathbf{A} is a gauge field), and the system will be in an equilibrium state at $t = 0$. We then abruptly switch off the vector potential. From electrodynamics, we know that a time-dependent vector potential generates an electric field:

$$\mathbf{E} = -\frac{1}{c} \dot{\mathbf{A}}. \quad (\text{F.2.2})$$

In the present case, the electrical field will be an infinitesimal δ spike at $t = 0$:

$$\mathbf{E}(t) = \frac{1}{c} \mathbf{A} \delta(t). \quad (\text{F.2.3})$$

We can compute the current that results in the standard way. We note that we can write \mathcal{H}' in Eq. (F.2.1) as

$$\begin{aligned} \mathcal{H}' &= \mathcal{H}_0 - \sum_{i=1}^N \frac{e_i}{cm_i} \mathbf{p}_i \cdot \mathbf{A} + \mathcal{O}(A^2) \\ &= \mathcal{H}_0 - \frac{\mathbf{A}}{c} \int d\mathbf{r} \sum_{i=1}^N \frac{e_i}{m_i} \mathbf{p}_i \delta(\mathbf{r}_i - \mathbf{r}) \end{aligned}$$

$$= \mathcal{H}_0 - \frac{\mathbf{A}}{c} \int d\mathbf{r} \mathbf{j}(\mathbf{r}), \quad (\text{F.2.4})$$

where $\mathbf{j}(\mathbf{r})$ denotes the current density at point \mathbf{r} . The average current density at time t due to the perturbation is given by

$$\langle \mathbf{j}(t) \rangle = \frac{\mathbf{A}}{cV k_B T} \int d\mathbf{r} d\mathbf{r}' \langle \mathbf{j}(\mathbf{r}, 0) \mathbf{j}(\mathbf{r}', t) \rangle. \quad (\text{F.2.5})$$

The phenomenological expression for the current response to an applied δ -function electric field spike is (see Eq. (2.5.14))

$$\begin{aligned} \langle \mathbf{j}(t) \rangle &= \int_{-\infty}^t dt' \sigma(t - t') E(t') \\ &= \sigma(t) \frac{A}{c}. \end{aligned} \quad (\text{F.2.6})$$

From this, it immediately follows that

$$\sigma(t) = \frac{1}{V k_B T} \int d\mathbf{r} d\mathbf{r}' \langle \mathbf{j}(\mathbf{r}, 0) \mathbf{j}(\mathbf{r}', t) \rangle. \quad (\text{F.2.7})$$

The dc conductivity is then given by

$$\sigma(\omega = 0) = \frac{1}{V k_B T} \int_0^\infty dt \int d\mathbf{r} d\mathbf{r}' \langle \mathbf{j}(\mathbf{r}, 0) \mathbf{j}(\mathbf{r}', t) \rangle. \quad (\text{F.2.8})$$

F.3 Viscosity

The corresponding linear response expression for the viscosity seems more subtle because shear usually is not interpreted in terms of an external field acting on all molecules. Still, we can use, by analogy to the electrical conductivity case, a canonical transformation, the time derivative that corresponds to uniform shear. To achieve this, we consider a system of N particles with coordinates \mathbf{r}^N and Hamiltonian

$$\mathcal{H}_0 = \sum_{i=1}^N p_i^2 / (2m_i) + \mathcal{U}(\mathbf{r}^N). \quad (\text{F.3.1})$$

Now consider another system described by a set of coordinates \mathbf{r}'^N related to \mathbf{r}^N by a linear transformation:

$$\mathbf{r}'_i = \mathbf{h} \mathbf{r}_i. \quad (\text{F.3.2})$$

The Hamiltonian for the new system can be written as

$$\mathcal{H}_1 = \sum_{i=1}^N \frac{1}{2m_i} \mathbf{p}'_i \cdot \mathbf{G}^{-1} \cdot \mathbf{p}'_i + \mathcal{U}(\mathbf{r}'^N), \quad (\text{F.3.3})$$

where G , the metric tensor, is defined as

$$G \equiv h^T \cdot h. \quad (\text{F.3.4})$$

We assume that h differs infinitesimally from the unit matrix I :

$$h = I + \epsilon. \quad (\text{F.3.5})$$

In the case that we are interested in the effect of uniform shear, for instance, we could choose $\epsilon_{xy} = \epsilon$, while all other elements of $\epsilon_{\alpha\beta}$ are 0. Now consider the case that we equilibrate the system with Hamiltonian \mathcal{H}_1 , and at time $t = 0$, we switch off the infinitesimal deformation ϵ . This means that, at $t = 0$, the system experiences a δ -function spike in the shear rate

$$\frac{\partial v_x}{\partial y} = -\epsilon \delta(t). \quad (\text{F.3.6})$$

We can compute the time-dependent response of the shear stress, $\sigma_{xy}(t)$, to the sudden change from \mathcal{H}_1 to \mathcal{H}_0 :

$$\langle \sigma_{xy}(t) \rangle = -\epsilon \frac{1}{V k_B T} \langle \sigma_{xy}(0) \sigma_{xy}(t) \rangle. \quad (\text{F.3.7})$$

By combining Eqs. (F.3.6) and (F.3.7) with Eq. (2.5.14), we immediately see that the steady-state stress, σ_{xy} , that results from a steady shear is given by

$$\sigma_{xy} = \frac{\partial v_x}{\partial y} \times \frac{1}{V k_B T} \int_0^\infty dt \langle \sigma_{xy}(0) \sigma_{xy}(t) \rangle, \quad (\text{F.3.8})$$

and the resulting expression for the shear viscosity η is

$$\eta = \frac{1}{V k_B T} \int_0^\infty dt \langle \sigma_{xy}(0) \sigma_{xy}(t) \rangle. \quad (\text{F.3.9})$$

F.4 Elastic constants

A liquid flows under the influence of shear forces. A solid does not. Rather, any small deformation of a solid induces an elastic response (stress) that counteracts it. This elastic stress is proportional to the applied deformation (strain). The constants of proportionality between stress and strain (to be defined more precisely below) are called the elastic constants. Below we discuss how to measure these constants by computer simulation. For the sake of simplicity, we limit the discussion to crystals on isotropic (hydrostatic) pressure.

When considering the effect of the strain on the free energy of a solid, it is essential to introduce the so-called *Lagrangian strain tensor* (see, e.g., [182]).¹

¹ Confronted with the finiteness of the Greek alphabet, we use the symbol η to denote the Lagrangian strain tensor. This symbol can easily be confused with the scalar viscosity η .

The reason is that, on a local scale, all changes in free energy are due to changes in the *distances* between the particles that make up the solid. And the quantity that measures this change is precisely the Lagrangian strain. We start with the relation between new and old coordinates due to an elastic deformation:

$$\mathbf{r}' = (1 + \boldsymbol{\epsilon})\mathbf{r}, \quad (\text{F.4.1})$$

where

$$\epsilon_{\alpha\beta} \equiv \frac{\partial u_\alpha}{\partial x_\beta} \quad (\text{F.4.2})$$

is the (conventional) strain tensor. It measures the variation of the displacement field \mathbf{u} with the original coordinate \mathbf{r} . Due to the strain, the distance r_{ij} separating two points i and j in the solid is changed. The new squared distance is then related to the old distance by

$$\begin{aligned} r_{ij}^2 &= \mathbf{r}_{ij} \left(1 + \boldsymbol{\epsilon}^T\right) (1 + \boldsymbol{\epsilon}) \mathbf{r}_{ij} \\ &= \mathbf{r}_{ij} \left(1 + \boldsymbol{\epsilon}^T + \boldsymbol{\epsilon} + \boldsymbol{\epsilon}^T \boldsymbol{\epsilon}\right) \mathbf{r}_{ij} \\ &\equiv \mathbf{r}_{ij} (1 + 2\eta) \mathbf{r}_{ij}. \end{aligned}$$

This defines the Lagrangian strain tensor η . The new volume V' of the system is related to the original volume V_0 by

$$V' = V_0 \det(1 + \boldsymbol{\epsilon}) \quad (\text{F.4.3})$$

or

$$V' = V_0 \sqrt{\det(1 + 2\eta)}. \quad (\text{F.4.4})$$

We now expand the Helmholtz free-energy (F) per unit of (undeformed) volume (V) in powers of the Lagrangian strain parameters η :

$$\begin{aligned} F(\eta)/V &= V^{-1} \left[F(0) + \frac{\partial F}{\partial \eta_{\alpha\beta}} \eta_{\alpha\beta} + \frac{1}{2} \frac{\partial^2 F}{\partial \eta_{\alpha\beta} \partial \eta_{\gamma\delta}} \eta_{\alpha\beta} \eta_{\gamma\delta} + \dots \right] \\ &= V^{-1} F(0) + C_{\alpha\beta}^{(1)} \eta_{\alpha\beta} + \frac{1}{2} C_{\alpha\beta\gamma\delta}^{(2)} \eta_{\alpha\beta} \eta_{\gamma\delta} + \dots \quad (\text{F.4.5}) \end{aligned}$$

This equation defines the (second-order) elastic constants $C_{\alpha\beta\gamma\delta}^{(2)}$. To compute the elastic constants numerically, we need a *microscopic* expression for the η -dependence of F . To derive such a relation, we must consider in detail what a deformation of the system does to the partition function. Let us first consider the deformed system. The partition function of this system (ignoring constants, such as h^{-3N}) is equal to

$$Q(\eta) = \int d\mathbf{p}^N d\mathbf{r}^N \exp \left[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) \right]. \quad (\text{F.4.6})$$

614 Linear response: examples

This partition function depends on the deformation through the boundary conditions of the integral over the coordinates. This is not very convenient when computing derivatives with respect to the strain. Therefore, we first express the partition function of the deformed system in terms of coordinates and momenta of the original, undeformed system. We can express the coordinates (\mathbf{r}_i) and velocities ($\dot{\mathbf{r}}_i$) in this system in terms of the strain tensor $h \equiv (1 + \epsilon)$, and the original coordinates ($\mathbf{r}_{0,i}$) and velocities ($\dot{\mathbf{r}}_{0,i}$):

$$\begin{aligned}\mathbf{r}_i &= h\mathbf{r}_{0,i} \\ \dot{\mathbf{r}}_i &= h\dot{\mathbf{r}}_{0,i}\end{aligned}\quad (\text{F.4.7})$$

The kinetic energy, $\mathcal{K} = \sum \frac{1}{2}m_i \dot{\mathbf{r}}_i^2$, can be written as

$$\begin{aligned}\mathcal{K} &= \sum \frac{1}{2}m_i \dot{\mathbf{r}}_i^2 \\ &= \sum \frac{1}{2}m_i \dot{\mathbf{r}}_{0,i} (h^T h) \dot{\mathbf{r}}_{0,i} \equiv \sum \frac{1}{2}m_i \dot{\mathbf{r}}_{0,i} \cdot G \cdot \dot{\mathbf{r}}_{0,i},\end{aligned}\quad (\text{F.4.8})$$

where $h^T = (1 + \epsilon^T)$ is the transverse of h and $G = h^T h$ is the metric tensor. From the definition of h it follows that $G = (1 + 2\eta)$. We can now write down the generalized momentum $\mathbf{p}_{0,i}$ conjugate to the coordinate $\mathbf{r}_{0,i}$ (see Appendix A):

$$p_{0,i}^\alpha = \left(\frac{\partial \mathcal{K}}{\partial \dot{r}_{0,i}^\alpha} \right) = m_i G_{\alpha\beta} \dot{r}_{0,i}^\beta \quad (\text{F.4.9})$$

and hence

$$\begin{aligned}\mathcal{K} &= \sum \frac{1}{2}m_i \dot{\mathbf{r}}_{0,i} \cdot G \cdot \dot{\mathbf{r}}_{0,i} \\ &= \sum \frac{1}{2m_i} \mathbf{p}_{0,i} \cdot G^{-1} \cdot \mathbf{p}_{0,i} \\ &= \sum \frac{1}{2m_i} \mathbf{p}_{0,i} \cdot (1 + 2\eta)^{-1} \cdot \mathbf{p}_{0,i}.\end{aligned}\quad (\text{F.4.10})$$

As

$$\begin{aligned}\mathbf{p}_i &= m_i \dot{\mathbf{r}}_i = m_i h \dot{\mathbf{r}}_{0,i} = (h^T)^{-1} \mathbf{p}_{0,i} \\ \mathbf{r}_i &= h\mathbf{r}_{0,i},\end{aligned}\quad (\text{F.4.11})$$

the Jacobian of the transformation between $\{\mathbf{p}^N, \mathbf{r}^N\}$ and $\{\mathbf{p}_0^N, \mathbf{r}_0^N\}$ is equal to 1. Hence, we can write

$$\begin{aligned}Q(\eta) &= \int d\mathbf{p}^N d\mathbf{r}^N \exp \left[-\beta \mathcal{H}(\mathbf{p}^N, \mathbf{r}^N) \right] \\ &= \int d\mathbf{p}_0^N d\mathbf{r}_0^N \exp \left\{ -\beta \left[\sum \frac{1}{2m_i} \mathbf{p}_{0,i} \cdot (1 + 2\eta)^{-1} \cdot \mathbf{p}_{0,i} + \mathcal{U}(\mathbf{r}_0^N; \eta) \right] \right\}.\end{aligned}\quad (\text{F.4.12})$$

Now the dependence of $Q(\boldsymbol{\eta})$ on $\boldsymbol{\eta}$ is only contained in the Hamiltonian. We can now explicitly carry out the differentiation with respect to $\boldsymbol{\eta}$. Using

$$\begin{aligned} \left(\frac{\partial U}{\partial \eta_{\alpha\beta}} \right) &= \sum_{i < j} \left(\frac{\partial U}{\partial r_{ij}^2} \right) \left(\frac{\partial r_{ij}^2}{\partial \eta_{\alpha\beta}} \right) = \sum_{i < j} \left(\frac{\partial U}{\partial r_{ij}} \right) \frac{r_{0,ij}^\alpha r_{0,ij}^\beta}{r_{ij}} \\ &= \left(\mathbf{h}^{-1} \sum_{i < j} \left(\frac{\partial U}{\partial r_{ij}} \right) \frac{\mathbf{r}_{ij} \mathbf{r}_{ij}}{r_{ij}} (\mathbf{h}^T)^{-1} \right)_{\alpha\beta} \end{aligned} \quad (\text{F.4.13})$$

and

$$\begin{aligned} \sum \frac{1}{2m_i} \mathbf{p}_{0,i} \cdot \left(\frac{\partial G^{-1}}{\partial \eta_{\alpha\beta}} \right) \cdot \mathbf{p}_{0,i} &= - \sum \frac{1}{m_i} \left(\mathbf{p}_{0,i} \cdot G^{-1} \right)_\alpha \left(G^{-1} \cdot \mathbf{p}_{0,i} \right)_\beta \\ &= - \left(\mathbf{h}^{-1} \left(\sum \frac{1}{m_i} \mathbf{p}\mathbf{p} \right) (\mathbf{h}^T)^{-1} \right)_{\alpha\beta}, \end{aligned} \quad (\text{F.4.14})$$

we obtain

$$\left(\frac{\partial F}{\partial \eta_{\alpha\beta}} \right) = - \left(\mathbf{h}^{-1} \left(\sum \frac{1}{m_i} \mathbf{p}\mathbf{p} + \sum_{j < i} \mathbf{r}_{ij} \mathbf{f}_{ij} \right) (\mathbf{h}^T)^{-1} \right)_{\alpha\beta}. \quad (\text{F.4.15})$$

From this, it follows immediately that

$$\begin{aligned} C_{\alpha\beta}^{(1)} &\equiv \left(\frac{\partial F}{\partial \eta_{\alpha\beta}} \right) \\ &= \frac{V'}{V} \left[(1 + \boldsymbol{\epsilon})^{-1} \boldsymbol{\sigma} (1 + \boldsymbol{\epsilon}^T)^{-1} \right]_{\alpha\beta} \\ &= \sqrt{\det(1 + 2\boldsymbol{\eta})} \left[(1 + \boldsymbol{\epsilon})^{-1} \boldsymbol{\sigma} (1 + \boldsymbol{\epsilon}^T)^{-1} \right]_{\alpha\beta}, \end{aligned} \quad (\text{F.4.16})$$

where

$$\sigma_{\gamma\delta} = - \frac{1}{V'} \sum_i \left[\frac{p_{i\gamma} p_{i\delta}}{m_i} + \sum_{j < i} r_{ij\gamma} f_{ij\delta} \right] \quad (\text{F.4.17})$$

denotes the microscopic stress in the deformed system. Note that $\boldsymbol{\sigma}$ can be measured in a simulation, while $\boldsymbol{\epsilon}$ is fixed by the applied strain. For an undeformed system $C^{(1)}$ is simply equal to $-P$, where P is the hydrostatic pressure. From Eq. (F.4.16), it also follows that the constant of proportionality between the stress σ_{ij} and the linear strain ϵ_{rs} is given by

$$\left(\frac{\partial \sigma_{\alpha\beta}}{\partial \epsilon_{\gamma\delta}} \right) = (\sigma_{\alpha\delta} \delta_{\beta\gamma} + \sigma_{\beta\delta} \delta_{\alpha\gamma} - \sigma_{\alpha\beta} \delta_{\gamma\delta}) + C_{\alpha\beta\gamma\delta}^{(2)}. \quad (\text{F.4.18})$$

616 Linear response: examples

To determine the second-order elastic constants, $C_{\alpha\beta\gamma\delta}^{(2)}$, we must determine the initial linear dependence of $C_{\alpha\beta}^{(1)}$ on $\eta_{\gamma\delta}$. This technique for measuring the elastic constants is simple and also quite accurate (see, e.g., [720]). However, several computations are needed to measure the different elastic constants. The lower the symmetry of the crystal, the more calculations are needed. This can be avoided by directly considering the microscopic expression for the $C_{\alpha\beta\gamma\delta}^{(2)}$. Such an expression was derived by Squire et al. [721]:

$$C_{\alpha\beta\gamma\delta}^{(2)} = -\frac{1}{V k_B T} \langle \Delta\sigma_{\alpha\beta} \Delta\sigma_{\gamma\delta} \rangle + 2\rho k_B T (\delta_{\alpha\gamma} \delta_{\beta\delta} + \delta_{\alpha\delta} \delta_{\beta\gamma}) \\ + 4 \sum_{i < j, k < l} \left\langle \frac{\partial^2 U}{\partial r_{ij}^2 \partial r_{kl}^2} r_{ij\alpha} r_{ij\beta} r_{kl\gamma} r_{kl\delta} \right\rangle. \quad (\text{F.4.19})$$

Using Eq. (F.4.19), it takes only a single simulation to measure *all* elastic constants. Unfortunately, the statistical errors in evaluating this fluctuation expression are usually larger than those obtained when computing Eq. (F.4.16). The problems with statistics appear even worse in constant-stress MD simulations on the same system, where the elastic compliances (rather than the moduli) are determined from fluctuations in the box shape [722]. Eq. (F.4.19) can only be used if the intermolecular potentials are everywhere continuous. However, a fluctuation expression that works for hard-core systems has been developed by Farago and Kantor [723]. More details about the numerical evaluation of elastic constants can be found in [721], [436], [723], and [724], while the general framework of elastic properties at finite temperature is discussed in [182].

Appendix G

Committor for 1d diffusive barrier crossing

In section 15.4 we discussed the concept of the committer that quantifies the fraction of trajectories in a certain part of configuration space that will go on to reach the product state B .

Here, we give a very simple example of such a calculation for the case of a diffusive crossing of a one-dimensional barrier of width L . Mathematicians will cringe at the derivation below because it is clumsy, and the final result can be obtained in one line (literally¹) using the backward Kolmogorov equation in the theory of stochastic differential equations [637].

G.1 1d diffusive barrier crossing

Consider a process whereby a system diffusively crosses a one-dimensional barrier of constant height $\epsilon \ll k_B T$ and width L . The diffusion constant of the system is assumed to be constant and equal to D . We denote the coordinate of the system by x . The barrier is in the x -range $0 \leq x < L$. We call region $x < 0$ the “reactant” region A . The product region (B) corresponds to $x > L$.

What is the probability that a system at point x_0 in the barrier region will “commit” to state B ? To answer this question, we must solve the 1d diffusion equation in the barrier region and compute the integrated flux to B . If $\epsilon/k_B T$ is large enough, all systems arriving at $x = L$ will proceed to B , and all systems arriving at $x = 0$ will proceed to A . We can therefore treat both $x = 0$ and $x = L$ as absorbing boundaries, meaning that the concentration $c(x)$ always vanishes at these boundaries.

The diffusion problem is best solved by Fourier transforming. The diffusion equation that we must solve is of the form:

$$\dot{c}(x, t) = D \left(\frac{\partial^2 c(x, t)}{\partial x^2} \right). \quad (\text{G.1.1})$$

¹ The one line is $\left(\frac{\partial^2 C(x)}{\partial x^2} \right) = 0$, where the committer $C(x)$ satisfies the boundary conditions $C(0) = 0$ and $C(L) = 1$. But to get there would have required a mini-course on stochastic differential equations, an investment well worth the effort but beyond the scope of this book.

618 Committor for 1d diffusive barrier crossing

We expand $c(x, t)$ in the region $0 < x \leq L$ in Fourier modes. As the concentration must vanish at the boundaries, we only consider sine components²:

$$c(x, t) = \sum_{n=1}^{\infty} a_n(t) \sin(n\pi x/L), \quad (\text{G.1.2})$$

and hence

$$a_n(t) = \frac{2}{L} \int_0^L dx c(x, t) \sin(n\pi x/L). \quad (\text{G.1.3})$$

In what follows, we will use the notation $q_n \equiv n\pi/l$. We assume that the concentration profile at $t = 0$ is a delta-function at x_0 :

$$c(x, 0) = \delta(x - x_0), \quad (\text{G.1.4})$$

then

$$a_n(0) = \frac{2}{L} \sin(q_n x_0). \quad (\text{G.1.5})$$

In Fourier components, the diffusion equation reads:

$$\dot{a}_n(t) = -D q_n^2 a_n(t) \quad (\text{G.1.6})$$

and hence:

$$a_n(t) = a_n(0) \exp(-D q_n^2 t). \quad (\text{G.1.7})$$

G.2 Computing the committor

The probability current at $x = L$ is

$$j(L, t) = -D \left(\frac{\partial c(x, t)}{\partial x} \right)_L = -D \sum_{n=1}^{\infty} a_n(t) q_n \cos(q_n L). \quad (\text{G.2.1})$$

The integrated flux to B is the probability that a system originally at x_0 will reach the absorbing boundary at L . In other words, it is the committor $C(x_0)$, given by:

$$C(x_0) \equiv \int_0^{\infty} dt j(L, t) = -\frac{2}{L} \sum_{n=1}^{\infty} \sin(q_n x_0) \frac{\cos(q_n L)}{q_n}. \quad (\text{G.2.2})$$

Rather than solving this equation, we consider the derivative

$$\left(\frac{\partial C(x_0)}{\partial x_0} \right) = -\frac{2}{L} \sum_{n=1}^{\infty} \cos(q_n x_0) \cos(q_n L), \quad (\text{G.2.3})$$

² The sine expansion applies to a periodic, anti-symmetric ($f(-x) = -f(x)$) function with period $2L$. We are only interested in the behavior for $0 \leq x < L$.

but the right-hand side is nothing else than the Fourier transform of $-\delta(L - x_0)$, excluding the $n = 0$ part.³ Therefore, we can write $\left(\frac{\partial C(x_0)}{\partial x_0}\right)$ as

$$\left(\frac{\partial C(x_0)}{\partial x_0}\right) = \frac{1}{L} - \delta(L - x_0). \quad (\text{G.2.4})$$

As $\delta(L - x_0) = 0$ for $x_0 \neq L$, the value of $\left(\frac{\partial C(x_0)}{\partial x_0}\right)$ equals $1/L$ for $x_0 < L$. Integrating the above equation with respect to x_0 , and using the fact that $C(0) = 0$, we get

$$C(x_0) = \frac{x_0}{L}. \quad (\text{G.2.5})$$

In other words, the value of the committer for this simple barrier crossing equals x_0/L . It varies from 0 at $x_0 = 0$ to 1 at $x_0 = L$. This is just an example that shows how committors may be computed if we know the solution of the Fokker-Planck equation of the problem.

³ The cosine expansion applies to a periodic, symmetric ($f(-x) = f(x)$) function with period $2L$. Again, we are only interested in the behavior for $0 \leq x < L$.

Appendix H

Smoothed dissipative particle dynamics

H.1 Navier-Stokes equation and Fourier's law

We briefly summarize the most important relations that underlie the [Smoothed Dissipative Particle Dynamics \(SDPD\)](#) method of Espa  ol and Revenga [694]. A description of [SDPD](#) fits in this book on particle-based simulations because it provides a particle-based picture of transport in fluids that is compatible with non-equilibrium thermodynamics [57] in that it correctly accounts for entropy production. Moreover, like [DPD](#), it correctly accounts for thermal fluctuations.

As indicated in the main text, the method, which is an extension of the smoothed-particle hydrodynamics method of the 1970s [695–697], aims to represent a continuous hydrodynamic flow field by a set of particle trajectories following equations of motion that, on a large enough scale, mimic the Navier-Stokes equations of viscous flow, combined with those for heat conduction.

The energy of a fluid “particle” i is related to the basic thermodynamic parameters through an equation of state of the form $E_i = E(m, S_i, V_i)$, which remains to be specified. Every particle has a temperature T_i and a pressure P_i given by:

$$T_i = \left(\frac{\partial E}{\partial S_i} \right)$$

and

$$P_i = - \left(\frac{\partial E}{\partial V_i} \right).$$

In the absence of thermal fluctuations, macroscopic transport is described by the Navier-Stokes equation:

$$m\rho \frac{d\mathbf{v}}{dt} = -\nabla P + \eta \nabla^2 \mathbf{v} + (\zeta + \eta/3) \nabla \nabla \cdot \mathbf{v}, \quad (\text{H.1.1})$$

complemented with the (simplified) form of the expression for entropy production due to viscous flow and heat transport

$$T\rho \frac{ds}{dt} = \phi + \kappa \nabla^2 T \quad (\text{H.1.2})$$

where

$$\phi = 2\eta \bar{\nabla} \mathbf{v} : \bar{\nabla} \mathbf{v} + \zeta (\nabla \cdot \mathbf{v})^2$$

and finally, the mass-conservation law

$$\frac{d\rho}{dt} = -\rho \nabla \cdot \mathbf{v}. \quad (\text{H.1.3})$$

In the above equations, ρ stands for the number density: to be consistent with the notation elsewhere in this book, we use a slightly different notation from ref. [694]. \mathbf{v} denotes the flow velocity, P the hydrostatic pressure, η the shear viscosity, ζ the bulk viscosity, s the entropy per particle and κ the thermal conductivity.

H.2 Discretized SDPD equations

The discretized form of Eqs. (H.1.1) through (H.1.3) is:

$$m \dot{\mathbf{v}}_i = -\frac{(\nabla P)_i}{\rho_i} + \frac{\eta}{\rho_i} (\nabla^2 \mathbf{v})_i + \frac{(\zeta + \eta/3)}{\rho_i} (\nabla \nabla \cdot \mathbf{v})_i, \quad (\text{H.2.1})$$

$$T_i \dot{S}_i = \frac{\phi + \kappa (\nabla^2 T)_i}{\rho_i} \quad (\text{H.2.2})$$

and

$$\dot{\rho}_i = -\rho_i (\nabla \cdot \mathbf{v})_i, \quad (\text{H.2.3})$$

where we now use the local expression for density, pressure velocity gradients, and entropy. However, to link the continuum version of the Navier-Stokes equation, we now must give a prescription for computing gradient terms in the vicinity of particle i .

To make this link between the Navier-Stokes equation and the particle-based description, Español and Revenga postulated the following relation between the volume V_i of a particle i to the local density ρ_i around that particle:

$$V_i \equiv 1/\rho_i$$

with

$$\rho_i \equiv \int d\mathbf{r} \rho(\mathbf{r}) W(\mathbf{r} - \mathbf{r}_i),$$

where $\rho(\mathbf{r}) \equiv \sum_j \delta(\mathbf{r} - \mathbf{r}_j)$ and $W(\mathbf{r} - \mathbf{r}_i)$ is a normalized weight function:

$$\int d\mathbf{r} W(\mathbf{r} - \mathbf{r}_i) = 1.$$

Note that the density around particle i includes i itself: hence $\rho_i > 0$. Various choices for the functions form or $W(r)$ are possible; it is convenient to choose a

form for W that has a finite support, i.e., for which $W(r) = 0$ for $r > h$, where h remains to be chosen. Ref. [694] used:

$$\begin{aligned} W(r) &= \frac{105}{16\pi h^3} \left(1 + 3\frac{r}{h}\right) \left(1 - \frac{r}{h}\right)^3 && \text{for } r \leq h \\ &= 0 && \text{otherwise .} \end{aligned}$$

In what follows, it is not $W(r)$ itself that enters into the equations, but a quantity $F(r)$ defined through the relation:

$$\nabla W(r) = -\mathbf{r}F(r) .$$

For the choice of ref. [694],

$$F(r) = \frac{315}{4\pi h^5} (1 - r/h)^2 .$$

For $F(r_i - r_j)$ we will use the shorthand notation F_{ij} , and similarly $T_{ij} \equiv T_i - T_j$, $\mathbf{r}_{ij} \equiv (\mathbf{r}_i - \mathbf{r}_j)$, $\mathbf{v}_{ij} \equiv (\mathbf{v}_i - \mathbf{v}_j)$ and $\hat{\mathbf{e}}_{ij} \equiv \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$.

It then follows (after some slightly tedious algebra —see ref. [694]) that

$$\begin{aligned} m\dot{\mathbf{v}}_i &= \sum_j \left[\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right] F_{ij} \mathbf{r}_{ij} - \left(\frac{5\eta}{3} - \zeta \right) \sum_j \frac{F_{ij}}{\rho_i \rho_j} \mathbf{v}_{ij} \\ &\quad - 5 \left(\frac{\eta}{3} + \zeta \right) \sum_j \frac{F_{ij}}{\rho_i \rho_j} \hat{\mathbf{e}}_{ij} \hat{\mathbf{e}}_{ij} \cdot \mathbf{v}_{ij} . \end{aligned} \quad (\text{H.2.4})$$

Eq. (H.2.4), together with

$$\mathbf{r}_i = \mathbf{v}_i ,$$

provides an explicit expression for the time evolution of the positions of the fluid “particles.” Importantly, the equation satisfies Newton’s third law.

We still need to give the discretized form of the entropy production:

$$T_i \dot{S}_i = (\phi)_i - 2\kappa \sum_j \frac{F_{ij}}{\rho_i \rho_j} T_{ij} , \quad (\text{H.2.5})$$

where

$$(\phi)_i = \frac{1}{2} \left[\left(\frac{5\eta}{3} - \zeta \right) \sum_j \frac{F_{ij}}{\rho_i \rho_j} \mathbf{v}_{ij}^2 + 5 \left(\frac{\eta}{3} + \zeta \right) \sum_j \frac{F_{ij}}{\rho_i \rho_j} (\hat{\mathbf{e}}_{ij} \cdot \mathbf{v}_{ij})^2 \right] .$$

To solve these equations, we need an equation of state to relate the pressure to the density at each position: this is our relation $E_i = E(m, S_i, V_i)$, which we can choose to reproduce the properties of the fluid under consideration.

The only thing that remains is to add the noise term, to account for thermal fluctuations in the fluid. This involves adding random noise of the correct amplitude to the entropy of the particles and their velocities in such a way that energy conservation and momentum conservation are satisfied. For details, we refer the reader to ref. [694].

Appendix I

Saving CPU time

The energy or force calculation is the most time-consuming part of almost all Molecular Dynamics and Monte Carlo simulations. If we consider a model system with pairwise additive interactions (as is done in many molecular simulations), we have to consider the contribution to the force on particle i , by all its neighbors. If we do not truncate the interactions, this implies that, for a system of N particles, we must evaluate $N(N - 1)/2$ pair interactions. And even if we do truncate the potential, we still would have to compute all $N(N - 1)/2$ pair distances to describe which pairs can interact. This implies that, if we use no tricks, the time needed for evaluating the energy scales as N^2 . There exist efficient techniques for speeding up the evaluation of both short-range and long-range interactions in such a way that the computing time scales as $N^{3/2}$, rather than N^2 . The techniques for the long-range interactions were discussed in Chapter 11; here, we discuss some of the techniques used for the short-range interactions. These techniques are:

1. Verlet list
2. Cell (or linked) list
3. Combination of Verlet and cell lists

I.1 Verlet list

If we simulate a large system and use a cutoff that is smaller than the simulation box, many particles do not contribute to the energy of a particle i . It is advantageous, therefore, to exclude the particles that do not interact from the expensive energy calculation. Verlet [14] developed a bookkeeping technique, commonly referred to as the Verlet list or neighbor list, which is illustrated in Fig. I.1. In this method, a second cutoff radius $r_v > r_c$ is introduced, and before we calculate the interactions, a list is made (the Verlet list) of all particles within a radius r_v of particle i . In the subsequent calculation of the interactions, only those particles in this list have to be considered. Until now, we have not saved any CPU time. We gain such time when we next calculate the interactions; if the maximum displacement of the particles is less than $r_v - r_c$, then we have to consider only the particles in the Verlet list of particle i . This is a calculation of order N . As soon as one of the particles is displaced more than $r_v - r_c$, we have to update the Verlet list. The latter operation is of order N^2 , and although this step is not performed each time an interaction is calculated, it will dominate for a very large number of particles.

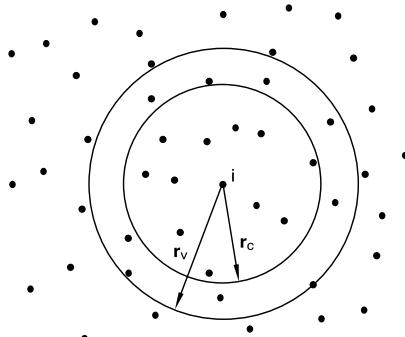


FIGURE I.1 The Verlet list: a particle i interacts with those particles within the cutoff radius r_c ; the Verlet list contains all the particles within a sphere with radius $r_v > r_c$.

The Verlet list can be used for both Molecular Dynamics and Monte Carlo simulations. However, there are some small differences in the implementation. For example, in a Molecular Dynamics simulation, the force on all particles is calculated simultaneously. It is sufficient, therefore, to have a Verlet list with half the number of particles for each particle as long as the interaction $i-j$ is accounted for in either the list of particle i or that of j . In a Monte Carlo simulation, each particle is considered separately. Therefore, it is convenient to have for each particle the complete list. Algorithm 29 shows the use of the Verlet list in a Monte Carlo simulation.

Bekker et al. have developed an elegant extension of the Verlet list for systems with periodic boundary conditions [725]. To calculate the force or potential energy of particle i , one has to locate the nearest image of the particles in the Verlet list of particle j (see, Algorithm 30). Bekker et al. have shown that this nearest image calculation in the inner loop of an MD or MC simulation can be avoided.

In a periodic system, the total force on particle i can be written as

$$\mathbf{F}_i = \sum_{j=1}^N \sum_{k=-13}^{13} ' \mathbf{F}_{i(j,k)},$$

where the prime denotes that the summation is performed over the nearest image of particle j in the central box ($k = 0$) or in one of its 26 periodic images. Here, (j,k) denotes the periodic image of particle j in box k . Box k is defined by the integer numbers n_x, n_y, n_z :

$$k = 9n_x + 3n_y + n_z$$

and

$$\mathbf{t}_k = n_x \mathbf{L}_x + n_y \mathbf{L}_y + n_z \mathbf{L}_z,$$

Algorithm 29 (Make use of Verlet list in MC trial move)

function mcmove_verlet	attempts to displace a particle using a Verlet list
o=int(\mathcal{R} *npart)+1	select a particle at random
if $ x(o)-xv(o) > (rv-rc)/2$ then	check to make a new list
new_vlist	
endif	
eno = en_vlist(o,x(o))	energy old configuration
xn=x(o)+($\mathcal{R}-0.5$)*delx	random displacement
if $ xn-xv(o) > (rv-rc)/2$ then	check to make a new list
new_vlist	
endif	
enn = en_vlist(o,xn)	energy new configuration
arg=exp(-beta*(enn-eno))	
if $\mathcal{R} < arg$ then	accepted: replace $x(o)$ by xn
$x(o)=xn$	
endif	
end if	

Specific Comments (for general comments, see p. 7)

1. The MC algorithm is based on Algorithm 2.
2. function **new_vlist** makes the Verlet list (see Algorithm 30) and function **en_vlist** calculates the energy of a particle at the given position using the Verlet list (see Algorithm 31).

where \mathbf{t}_k is the translation vector of the central box to its periodic image k . A particle in the central box is denoted by $(i,0) = i$. Using this notation, we can write, for the interaction between particles i and j ,

$$\mathbf{F}_{i(j,k)} = \mathbf{F}_{(i,-k)j} = -\mathbf{F}_{(j,k)i} = -\mathbf{F}_{j(i,-k)}.$$

We can write

$$\mathbf{F}_i = \sum_{j=1}^N \sum_{k=-13}^{13} {}' \mathbf{F}_{(i,k)j}.$$

The importance of this seemingly trivial result is that the summation is over all particles j in the *central* box with the nearest image of particle i . The difference between the two approaches is shown in Fig. I.2.

This method is implemented using different Verlet lists for each periodic image of particle i . These lists contain only those particles that interact with particle i and are in the central box. If these lists are used, it is not necessary to use the nearest image operation during the calculation of the force or energy. In [725] the use of these lists is shown to speed up an MD simulation by a factor

Algorithm 30 (Making a Verlet list)

```

function new_vlist
for 1 ≤ i ≤ npart do makes a new Verlet list
    nlist(i)=0 initialize list
    xv(i)=x(i)
    store position of particles
enddo
for 1 ≤ i ≤ npart-1 do
    for i+1 ≤ j ≤ npart do
        xr=x(i)-x(j)
        xr=xr-box*round(xr/box) nearest image
        if |xr| < rv then
            nlist(i)=nlist(i)+1 Test if particle j is within
            nlist(j)=nlist(j)+1 range rv of Verlet List
            list(i,nlist(i))=j If so, add to lists
            list(j,nlist(j))=i
        endif
    enddo
enddo
end function

```

Specific Comments (for general comments, see p. 7)

- If j is within rv (> rc) from i, then the 2d array element list(i,icount)=j. The total number of particles within distance rv from i is given by nlist(i). The array xv(i) contains the position of particle i at the moment that the list is made.*
- The example above is for a Verlet list for MC: it stores j as a neighbor of i and i as a neighbor of j. For MD, every particle pair enters only once.*
- We must update the Verlet list if any particle has moved more than a distance (rv-rc)/2 from the position where it was when the list was last made.*
- Clearly, there is a trade off: the smaller rv, the cheaper it is to make the list, but the more often it has to be updated.*

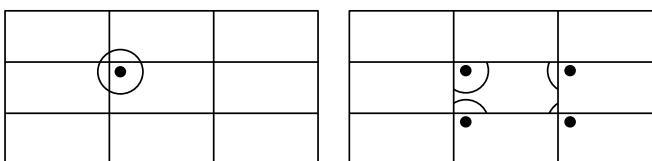


FIGURE 1.2 Verlet lists: (left) conventional approach in which each particle has a Verlet list; (right) the approach of Bekker et al. in which each periodic image of a particle has its own Verlet list that contains only those particles in the central box.

1.5. In addition, Bekker et al. have shown that a similar trick can be used to take the calculation of the virial (pressure) out of the inner loop.

Algorithm 31 (Calculating the energy using a Verlet list)

<pre>function en_vlist(i,xi) en=0 for 1 ≤ jj ≤ nlist(i) do j=list(i,jj) xij=xi-x(j) xij= (xij-box*round(xij/box) en=en+enij(xij) enddo end function</pre>	calculates interaction energy of particle i with the Verlet list loop over the particles in the list next particle in the list nearest image distance enij is pair potential of pair ij
---	--

Specific Comment (for general comments, see p. 7)

1. Array $\text{list}(i,\text{itel})$ and nlist are made in Algorithm 30 and enij (not specified) returns the pair potential energy of particles i and j at xi and $\text{x}(j)$.

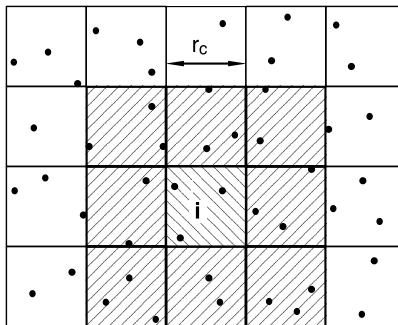


FIGURE I.3 The cell list: the simulation cell is divided into cells of size $r_c \times r_c$; a particle i interacts with those particles in the same cell or neighboring cells (in 2d there are 9 cells; and in 3d, 27 cells).

I.2 Cell lists

An algorithm that scales with N is the cell list or linked-list method [28]. The idea of the cell list is illustrated in Fig. I.3. The simulation box is divided into cells with a size equal to or slightly larger than the cutoff radius r_c ; each particle in a given cell interacts with only those particles in the same or neighboring cells. Since the allocation of a particle to a cell is an operation that scales with N and the total number of cells that needs to be considered for the calculation of the interaction is independent of the system size, the cell list method scales as N . Algorithm 33 shows how a cell list can be used in a Monte Carlo simulation.

Algorithm 32 (Making a cell list)

function new_nlist(rc)	make linked cell list for pair interactions with cut-off distance rc
$rn=box/int(box/rc)$	determine diameter of cells: $rn \geq rc$ box is the simulation box diameter
for $0 \leq icel \leq ncel-1$ do	set head of chain to 0 for each cell
$hoc(icel)=0$	
enddo	
for $1 \leq i \leq npart$ do	loop over the particles
$icel=int(x(i)%rn)$	determine cell number
$ll(i)=hoc(icel)$	link list the head of chain of cell $icel$
$hoc(icel)=i$	make particle i the head of chain
end function	

Specific Comment (for general comments, see p. 7)

1. This algorithm sets up a 1D linked-list. All particles are attributed to “their” cell. The latest particle (say i) added to a cell is referred to as the head of chain and stored in the array $hoc(icel)$. Particle i replaces the previous head of chain, but is linked to it via the link-list array $ll(i)$. Every particle points to (at most) one other particle. If $ll(i) = 0$, then there are no particles in the cell, beyond i . (chain).
2. The desired (optimum) cell size is rc , and rn ($> rc$) is the closest size that fits in the box.

I.3 Combining the Verlet and cell lists

It is instructive to compare the efficiency of the Verlet list and cell list in more detail. In three dimensions, the number of particles for which the distance needs to be calculated in the Verlet list is given by

$$n_v = \frac{4}{3}\pi\rho r_v^3;$$

for the cell list, the corresponding number is

$$n_l = 27\rho r_c^3.$$

If we use typical values for the parameters in these equations (Lennard-Jones potential with $r_c = 2.5\sigma$ and $r_v = 2.7\sigma$), we find that n_l is five times larger than n_v . As a consequence, in the Verlet scheme, the number of pair distances that needs to be calculated is 16 times less than in the cell list.

The observation that the Verlet scheme is more efficient in evaluating the interactions motivated Auerbach et al. [726] to use a combination of the two

Algorithm 33 (Calculating the energy using a cell list)

function enlist(<i>i, xi, en</i>)	calculates energy using a linked-cell list
<i>en</i> =0	
<i>icel</i> =int(<i>xi</i> % <i>rn</i>)	determine the cell number
for -1 ≤ <i>jn</i> ≤ 1 do	loop over neighbor cells (1d) triple loop in 3d
<i>jcel</i> =(<i>icel</i> + <i>jn</i>)% <i>ncel</i>	0≤ <i>jcell</i> ≤ <i>ncel</i> -1
<i>j</i> = <i>hoc(jcel)</i>	head of chain of cell <i>jcel</i>
while <i>j</i> ≠ 0 do	
if <i>i</i> ≠ <i>j</i> then	
<i>xij</i> = <i>xi</i> - <i>x(j)</i>	nearest image distance
<i>rij</i> = (xij-box*round(xij/box)	enij is pair potential of pair <i>ij</i>
<i>en</i> = <i>en</i> + enij(rij)	
endif	
<i>j</i> = <i>ll(j)</i>	next particle in the list
enddo	
enddo	
end function	

Specific Comment (for general comments, see p. 7)

1. Array *ll(i)* and *hoc(icel)* are constructed in Algorithm 32; *enij* is a function that returns the pair energy of particles *i* and *j* at distance *rij*.
2. *jn* = -1,0,+1 designates the cell to the left of *icel*, *icel* and the cell to the right.

lists: use a cell list to construct a Verlet list. The use of the cell list removes the main disadvantage of the Verlet list for a large number of particles—scales as N^2 —but keeps the advantage of an efficient energy calculation. An implementation of this method in a Monte Carlo simulation is shown in Algorithm 34.

I.4 Efficiency

The first question that arises is when to use which method. This depends very strongly on the details of the systems. In any event, we always start with a scheme as simple as possible, hence no tricks at all. Although the algorithm scales as N^2 , it is straightforward to implement, and therefore the probability of programming errors is relatively small. In addition, we should take into account how often the program will be used.

The use of the Verlet list becomes advantageous if the number of particles in the list is significantly less than the total number of particles; in three dimen-

Algorithm 34 (Combination of Verlet and cell lists)

function mcmove_clist	displace a particle using a combined list
o=int($\mathcal{R} \times n_{\text{part}}$)+1	select a particle at random
if $(x(o)-xv(o)) > rv - rc$ then	need to make new list ?
new_clist	
endif	
eno= en_vlist (o,x(o))	energy old configuration
xn=x(o)+($\mathcal{R} - 0.5$) $\times \Delta x$	random displacement
if $(xn-xv(o)) > rv - rc$ then	need to make new list ?
new_clist	
endif	
enn = en_vlist (o,xn)	energy new configuration
arg=exp(-beta*(enn-eno))	
if $\mathcal{R} < arg$ then	
x(o)=xn	accepted: replace $x(o)$ by xn
endif	
end function	

Specific Comments (for general comments, see p. 7)

1. The algorithm is based on Algorithm 29.
2. Function **new_clist** creates a Verlet list using a cell list (see Algorithm 35) and function **en_vlist** calculates the energy of a particle at the given position using the Verlet list (see Algorithm 31).

sions, this means

$$n_v = \frac{4}{3}\pi r_v^3 \rho \ll N.$$

If we substitute some typical values for a Lennard-Jones potential ($r_v = 2.7\sigma$ and $\rho = 0.8\sigma^{-3}$), we find $n_v \approx 66$, which means that only if the number of particles in the box is more than 100 does it make sense to use a Verlet list.

To see when to use one of the other techniques, we have to analyze the algorithms in somewhat more detail. If we use no tricks, the amount of CPU time to calculate the total energy is given by

$$\tau = cN(N-1)/2.$$

The constant gives the required CPU time for an energy calculation between a pair of particles. If we use the Verlet list, the CPU time is

$$\tau_v = cn_v N + \frac{c_v}{n_u} N^2,$$

Algorithm 35 (Making a Verlet list using a cell list)

function new_clist	
	makes a new Verlet list
new_nlist(rv)	using a cell list
for $1 \leq i \leq npart$ do	first make the cell list
$nlist(i)=0$	initialize Verlet list
$xv(i)=x(i)$	store particle positions
enddo	
for $1 \leq i \leq npart$ do	determine cell number
$icel=\text{int}(x(i)\%rn)$	loop over neighbor cells (1D)
for $-1 \leq jn \leq 1$ do	$0 \leq jcell \leq ncel - 1$
$jcel=(icel+jn)\%ncel$	head of chain of cell jcel
$j=hoc(jcel)$	
while $j \neq 0$ do	nearest image distance
if $i \neq j$ then	add to the Verlet lists
$xr=x(i)-x(j)$	
$xr=xr-\text{box}*\text{round}(xr/\text{box})$	
if $ xr < rv$ then	
$nlist(i)=nlist(i)+1$	
$nlist(j)=nlist(j)+1$	
$list(i,nlist(i))=j$	
$list(j,nlist(j))=i$	
endif	
endif	
$j=ll(j)$	next particle in the cell list
enddo	
enddo	
end function	

Specific Comments (for general comments, see p. 7)

1. Array $list(i,itel)$ is the Verlet list of particle i . The number of particles in the Verlet list of particle i is given by $nlist(i)$. The array $xv(i)$ contains the position of the particles at the moment the list is made, and is used in Algorithm 29 to test if new list should be made).
2. Function **new_nlist** makes a cell list (Algorithm 32). The cell size rn should be no less than the range of the Verlet list (rv), which in turn should be no less than the interaction range rc .

where the first term arises from the calculation of the interactions and the second term from the update of the Verlet list, which is done every n_u^{th} cycle.

The cell list scales with N , and the CPU time can be split into two contributions: one that accounts for the calculation of the energy and the other for the

making of the list,

$$\tau_l = cn_l N + c_l N.$$

If we use a combination of the two lists, the total CPU time becomes

$$\tau_c = cn_v N + \frac{c_l}{n_u} N.$$

The way to proceed is to perform some test simulations to estimate the various constants, and from the equations, it will become clear which technique is preferred. In Example 29, we have made such an estimate for a simulation of the Lennard-Jones fluid.

Example 29 (Comparison of schemes for the Lennard-Jones fluid). It is instructive to make a detailed comparison of the various schemes to save CPU time for the Lennard-Jones fluid. We compare the following schemes:

1. Verlet list
2. Cell list
3. Combination of Verlet and cell lists
4. Simple N^2 algorithm

We have used the program of Case Study 1 as a starting point. At this point it is important to note that we have not tried to optimize the parameters (such as the Verlet radius) for the various methods; we have simply taken some reasonable values.

For the Verlet list (and for the combination of Verlet and cell lists) it is important that the maximum displacement be smaller than twice the difference between the Verlet radius and cutoff radius. For the cutoff radius we have used $r_c = 2.5\sigma$, and for the Verlet radius $r_v = 3.0\sigma$. This limits the maximum displacement to $\Delta_x = 0.25\sigma$ and implies for the Lennard-Jones fluid that, if we want to use a optimum acceptance of 50%, we can use the Verlet method only for densities larger than $\rho > 0.6\sigma^{-3}$. For smaller densities, the optimum displacement is larger than 0.25. Note that this density dependence does not exist in a Molecular Dynamics simulation. In a Molecular Dynamics simulation, the maximum displacement is determined by the integration scheme and therefore is independent of density. This makes the Verlet method much more appropriate for a Molecular Dynamics simulation than for a Monte Carlo simulation. Only at high densities does it make sense to use the Verlet list.

The cell list method is advantageous only if the number of cells is larger than 3 in at least one direction. For the Lennard-Jones fluid this means that, if the number of particles is 400, the density should be lower than $\rho < 0.5\sigma^{-3}$. An important advantage of the cell list over the Verlet list is that this list can also be used for moves in which a particle is given a random position.

From these arguments it is clear that, if the number of particles is smaller than 200–500, the simple N^2 algorithm is the best choice. If the number of particles is significantly larger and the density is low, the cell list method is probably more efficient. At high density, all methods can be efficient and we have to make a detailed comparison.

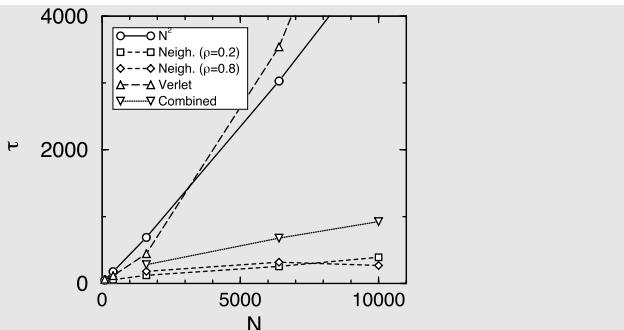


FIGURE I.4 Comparison of various schemes to calculate the energy: τ is in arbitrary units and N is the number of particles. As a test case the Lennard-Jones fluid is used. The temperature was $T^* = 2$ and per cycle the number of attempts to displace a particle was set to 100 for all systems. The lines serve to guide the eye.

To test these conclusions about the N dependence of the CPU time of the various methods, we have performed several simulations with a fixed number of Monte Carlo cycles. For the simple N^2 algorithm the CPU time per attempt is

$$\tau_{N^2} = cN,$$

where c is the CPU time required to calculate one interaction. This implies that the total amount of CPU time is independent of the density. For a calculation of the total energy, we have to do this calculation N times, which gives the scaling of N^2 . Fig. I.4 shows that indeed for the Lennard-Jones fluid, the τ_{N^2} increases linearly with the number of particles.

If we use the cell list, the CPU time will be

$$\tau_l = cV_l\rho + c_l p_l N,$$

where V_l is the total volume of the cells that contribute to the interaction (in three dimensions, $V_l = 27r_c^3$), c_l is the amount of CPU time required to make a cell list, and p_l is the probability that a new list has to be made. Fig. I.4 shows that the use of a cell list reduces the CPU time for 10,000 particles with a factor 18. Interestingly, the CPU time does not increase with increasing density. We would expect an increase since the number of particles that contribute to the interaction of a particle i increases with density. However, the second contribution to $\tau_{\text{Neigh.}}$ (p_l) is the probability that a new list has to be made, depends on the maximum displacement, which decreases when the density increases. Therefore, this last term will contribute less at higher densities.

For the Verlet scheme the CPU time is

$$\tau_v = cV_v\rho + c_v p_v N^2,$$

where V_v is the volume of the Verlet sphere (in three dimensions, $V_v = 4\pi r_v^3/3$), c_v is the amount of CPU time required to make the Verlet-list, and p_v

636 Saving CPU time

is the probability that a new list has to be made. Fig. 1.4 shows that this scheme is not very efficient. The N^2 operation dominates the calculation. Note that we use a program in which a new list for all particles has to be made as soon as one of the particles has moved more than $(r_v - r_c)/2$; with some more bookkeeping it is possible to make a much more efficient program, in which a new list is made for only the particle that has moved out of the list.

The combination of the cell and Verlet lists removes the N^2 dependence of the simple Verlet algorithm. The CPU time is given by

$$\tau_c = c V_v \rho + c_v p_v c_l N.$$

Fig. 1.4 shows that indeed the N^2 dependence is removed, but the resulting scheme is not more efficient than the cell list alone.

This case study demonstrates that it is not simple to give a general recipe for which method to use. Depending on the conditions and number of particles, different algorithms are optimal. It is important to note that for a Molecular Dynamics simulation the conclusions may be different.

For more details, see SI (Case Study 26).

Appendix J

Some general purpose algorithms

This appendix describes a few algorithms that are used in the text.

J.1 Gaussian distribution

Algorithm 36 (Gaussian distribution)

If R_1 and R_2 are two random numbers distributed uniformly over the interval $\{0, 1\}$, then X_g , given by

$$X_g = X_{\text{avg}} + \sigma \sqrt{-\ln(R_1)} \cos(2\pi R_2)$$

has a Gaussian distribution with average X_{avg} and variance σ^2

Comment *The above algorithm is but one example. It is simple, but not necessarily the fastest.*

J.2 Selection of trial orientations

Algorithm 37 (Selection of trial orientations)

In the configurational bias MC method, we often need to select the next bond direction from a set of k trial directions. Below, we assume that the (Boltzmann) weights $w(n)$ of the individual trial directions are known.

<pre> function select(w, sum_w) selects a trial orientation n with prob. $p(n) = w(n) / \sum_j w(j)$ $ws = \mathcal{R} * \text{sum}_w$ $cumw = w(1)$ $n = 1$ while $cumw < ws$ do $n = n + 1$ $cumw = cumw + w(n)$ enddo end function</pre>	The function returns n , the index of the selected trial position
---	--

Specific Comments (for general comments, see p. 7)

1. For large values of k bisection [38] can be more efficient.

J.3 Generate random vector on a sphere

Algorithm 38 (Random vector on a unit sphere)

<pre> function ranor ransq=2. do while (ransq.ge.1.0) ran1=1.-2.*\mathcal{R} ran2=1.-2.*\mathcal{R} ran3=1.-2.*\mathcal{R} ransq=ran1*ran1+ran2*ran2 +ran3*ran3 enddo or = 1.0/\sqrt{ransq} bx=ran1*or by=ran2*or bz=ran3*or end function</pre>	generates a 3d random unit vector with components bx, by, bz Continue until the vector is inside unit sphere
---	---

Specific Comment (for general comments, see p. 7)

1. The above algorithm is but one example. It is simple, but not necessarily the fastest.

J.4 Generate bond length

Algorithm 39 (Generate bond length with harmonic springs (3d))

```

function bondl
    returns bond length  $\ell$ 
    assume harmonic springs.
    Spring constant  $k_v$ .
     $\ell_0$ : bond length at  $T = 0$ 

     $\alpha = k_v/(k_B T)$ 
     $\ell_M = (\ell_0/2) * (1 + \sqrt{1 + 8/(\alpha\ell_0^2)})$  Position maximum at  $T$ 
    ready=.false.
    while ready == .false. do
         $\ell = \text{gauss}(\alpha, \ell_M)$  generate  $\ell$  with a Gaussian distr.
        aux =  $2 * [-(\ell/\ell_M - 1) + \ln(\ell/\ell_M)]$  auxiliary quantity
        if  $\mathcal{R} \leq \exp(-\text{aux})$  then accepted ?
            ready=.true.
        endif
    enddo rejection step
end function

```

Specific Comments (for general comments, see p. 7)

1. The bond length has the following distribution:

$$p(\ell) \propto \exp[-\beta 0.5k_v(\ell - \ell_0)^2] d\ell \propto \ell^2 \exp[-\beta 0.5k_v(\ell - \ell_0)^2] d\ell.$$

2. We make use of the fact that $x - 1 \geq \ln x$.

3. $\text{gauss}(\alpha, \ell_M)$ is a 1d normal distribution, see section J.1.

The distribution of bond lengths of linear, harmonic molecules in 3d is close to a Gaussian distribution and can be generated starting from the Box-Muller algorithm [66].

Consider a linear molecule that can vibrate along its axis. The force constant for the vibration is denoted by κ . For a fixed orientation, the equilibrium bond-length is denoted by ℓ_0 . We denote the inverse temperature by β . We ignore coupling between rotation and vibration.

Under these circumstances, the length-distribution of the molecule is

$$P(\ell) \sim \ell^2 \exp[-0.5\beta\kappa(\ell - \ell_0)^2] = \exp[-0.5\beta\kappa(\ell - \ell_0)^2 + 2\ln\ell]. \quad (\text{J.4.1})$$

For what follows, the normalization is unimportant. We cannot sample this distribution directly. However, we can use the rejection method. First, we determine the location ℓ_M of the maximum of $\ln P(\ell)$. We then get:

$$\beta\kappa(\ell_M - \ell_0) - 2/\ell_M = 0. \quad (\text{J.4.2})$$

640 Some general purpose algorithms

Denoting $\beta\kappa$ by α we get

$$\ell_M^2 - \ell_M \ell_0 - 2/\alpha = 0, \quad (\text{J.4.3})$$

or

$$\ell_M = 0.5(\ell_0 + \sqrt{\ell_0^2 + 8/\alpha}). \quad (\text{J.4.4})$$

We can approximate Eq. (J.4.1) by a Gaussian around ℓ_0 :

$$P'(\ell) \sim \exp[-0.5\alpha(\ell - \ell_M)^2]. \quad (\text{J.4.5})$$

$P'(\ell)$ is related to $P(\ell)$ through

$$P(\ell)/P'(\ell) = \exp\left[2\ln(\ell/\ell_M) - \frac{2(\ell - \ell_M)}{\ell_M}\right] \leq 1.$$

We can therefore draw values of ℓ from $P'(\ell)$ and then reject values of ℓ if

$$\mathcal{R} > \exp\left[2\ln(\ell/\ell_M) - \frac{2(\ell - \ell_M)}{\ell_M}\right].$$

J.5 Generate bond angle

Algorithm 40 (Generate bond angle)

<pre> function bonda(xn,i) ready=.false. while ready == .false. do b = ranor dx1x2=xn(i-1)-xn(i-2) u12=dx1x2/ dx1x2 theta=acos(b*dx1x2) bu = ubb(theta) if $\mathcal{R} < \exp(-\beta\kappa bu)$ then ready=.true. endif endif end function</pre>	generate bond orientation vector b with Boltzmann probability given by the bond-bending potential unit vector on a sphere vector $\mathbf{r}_{21} = \mathbf{r}_{i-1} - \mathbf{r}_{i-2}$ normalize vector bending angle $\theta = \arccos(\hat{\mathbf{u}}_{12} \cdot \hat{\mathbf{b}})$ bond-bending energy rejection test
--	---

Specific Comment (for general comments, see p. 7)

1. This algorithm uses a naive rejection scheme to generate a Boltzmann distribution of orientations $\hat{\mathbf{b}}$. Function **ranor** generates a random vector on a unit sphere (Algorithm 38). The function **ubb** (not specified) gives the bond-bending energy for the given angle.

J.6 Generate bond and torsion angle

Algorithm 41 (Generate bond and torsion angle)

<pre> function tors_bonda(<i>xn,i</i>) ready=.false. while ready == .false. do b = ranor dx1x2=xn(i-1)-xn(i-2) dx1x2=dx1x2/ dx1x2 dx2x3=xn(i-2)-xn(i-3) dx2x3=dx2x3/ dx2x3 theta=acos(b * dx1x2) ubb=ubb(theta) xx1=b × dx1x2 xx2=dx1x2 × dx2x3 [...normalize xx1 and xx2] phi=acos(xx1 * xx2) utors=utors(phi) usum=ubb+utors if $\mathcal{R} < \exp(-\beta * \text{usum})$ then ready=.true. endif enddo end function</pre>	generate a unit vector with orientational Boltzmann distri- bution determined by torsion and bond-bending potentials generate random unit vector $\hat{\mathbf{b}}$ vector $\mathbf{r}_{21} = \mathbf{r}_{i-1} - \mathbf{r}_{i-2}$ normalize \mathbf{r}_{12} : $\hat{\mathbf{u}}_{12} \equiv \mathbf{r}_{12}/ \mathbf{r}_{12} $ vector $\mathbf{r}_{23} = \mathbf{r}_{i-2} - \mathbf{r}_{i-3}$ normalize \mathbf{r}_{23} : $\hat{\mathbf{u}}_{23} \equiv \mathbf{r}_{23}/ \mathbf{r}_{23} $ bending angle $\theta = \arccos(\hat{\mathbf{u}}_{12} \cdot \hat{\mathbf{b}})$ bond-bending energy cross product : $\mathbf{xx1} = \hat{\mathbf{b}} \times \hat{\mathbf{u}}_{12}$ cross product: $\mathbf{xx2} = \hat{\mathbf{u}}_{12} \times \hat{\mathbf{u}}_{23}$ torsion angle $\phi = \arccos(\hat{\mathbf{xx1}} \cdot \hat{\mathbf{xx2}})$ determine torsion energy rejection test
---	---

Specific Comments (for general comments, see p. 7)

1. This algorithm uses a naive rejection scheme to generate a Boltzmann distribution of orientations $\hat{\mathbf{b}}$.
2. In the literature, various definitions of the torsion angle are used.
3. The function **ranor** generates a vector uniformly on a unit sphere (Algorithm 38), and the function **ubb** (not specified) gives the bond-bending energy for the given angle θ . The function **utors** (also not specified) gives the torsion energy for the dihedral angle ϕ .

Appendix K

Errata

This page is optimistically left empty, but will be updated if it turns out that our optimism was unwarranted.

Please send your suggestions/corrections by email to berend.smit@epfl.ch and df246@cam.ac.uk.

Appendix L

Miscellaneous methods

L.1 Higher-order integration schemes

The basic idea behind the predictor-corrector algorithms is to use information about the position and its first n derivatives at time t to arrive at a prediction for the position and its first n derivatives at time $t + \Delta t$. We then compute the forces (and thereby the accelerations) at the predicted positions. And then we find that these accelerations are *not* equal to the values that we had predicted. So we adjust our predictions for the accelerations to match the facts. But we do more than that. On the basis of the observed discrepancy between the predicted and observed accelerations, we also try to improve our estimate of the positions and the remaining $n - 1$ derivatives. This is the “corrector” part of the predictor-corrector algorithm. The precise “recipe” used in applying this correction is a compromise between accuracy and stability. Here, we shall simply show a specific example of a predictor-corrector algorithm, without attempting to justify the form of the corrector part.

Consider the Taylor expansion of the coordinate of a given particle at time $t + \Delta t$:

$$r(t + \Delta t) = r(t) + \Delta t \frac{\partial r}{\partial t} + \frac{\Delta t^2}{2!} \frac{\partial^2 r}{\partial t^2} + \frac{\Delta t^3}{3!} \frac{\partial^3 r}{\partial t^3} + \dots$$

Using the notation

$$\begin{aligned}x_0(t) &\equiv r(t) \\x_1(t) &\equiv \Delta t \frac{\partial r}{\partial t} \\x_2(t) &\equiv \frac{\Delta t^2}{2!} \frac{\partial^2 r}{\partial t^2} \\x_3(t) &\equiv \frac{\Delta t^3}{3!} \frac{\partial^3 r}{\partial t^3},\end{aligned}$$

we can write the following *predictions* for $x_0(t + \Delta t)$ through $x_3(t + \Delta t)$:

$$\begin{aligned}x_0(t + \Delta t) &= x_0(t) + x_1(t) + x_2(t) + x_3(t) \\x_1(t + \Delta t) &= x_1(t) + 2x_2(t) + 3x_3(t) \\x_2(t + \Delta t) &= x_2(t) + 3x_3(t) \\x_3(t + \Delta t) &= x_3(t).\end{aligned}$$

e4 Miscellaneous methods

Now that we have $x_0(t + \Delta t)$, we can compute the forces at the predicted position, and thus compute the corrected value for $x_2(t + \Delta t)$. We denote the difference between $x_2^{\text{corrected}}$ and $x_2^{\text{predicted}}$ by Δx_2 :

$$\Delta x_2 \equiv x_2^{\text{corrected}} - x_2^{\text{predicted}}.$$

We now estimate “corrected” values for x_0 through x_3 , as follows:

$$x_n^{\text{corrected}} = x_n^{\text{predicted}} + C_n \Delta x_2, \quad (\text{L.1.1})$$

where the C_n are constants fixed for a given order algorithm. As indicated, the values for C_n are such that they yield an optimal compromise between the accuracy and the stability of the algorithm. For instance, for a fifth-order predictor-corrector algorithm (i.e., one that uses x_0 through x_4), the values for C_n are

$$\begin{aligned} C_0 &= \frac{19}{120} \\ C_1 &= \frac{3}{4} \\ C_2 &= 1 && \text{(of course)} \\ C_3 &= \frac{1}{2} \\ C_4 &= \frac{1}{12}. \end{aligned}$$

One may iterate the predictor and corrector steps to self-consistency. However, there is little point in doing so because (1) every iteration requires a force calculation. One would be better off spending the same computer time to run with a *shorter* time step and only one iteration because (2) even if we iterate the predictor-corrector algorithm to convergence, we still do not get the *exact* trajectory: the error is still of order Δt^n for an n th-order algorithm. This is why we gain more accuracy by going to a shorter time step than by iterating to convergence at a fixed value of Δt .

L.2 Surface tension via the pressure tensor

If we have an interface in our system we can compute the interfacial tension, γ , from the pressure tensor. In a homogeneous system at equilibrium, the thermodynamic pressure is constant and equal in all directions. For an inhomogeneous system, mechanical equilibrium requires that the component of the pressure tensor normal to the interface is constant throughout the system. The components tangential to the interface can vary in the interfacial region, but must be equal to the normal component in the bulk liquids.

For an inhomogeneous fluid there is no unambiguous way to compute the normal p_n and tangential p_t components of the pressure tensor [727–729]. Here, we have used the Kirkwood-Buff convention [730] for expressing the stress tensor in a system of particles with pairwise-additive interactions. The system is divided into N_{sl} equal slabs parallel to the x, y plane. The local normal ($p_n(k)$) and tangential ($p_t(k)$) components of the pressure tensor are given by [143]

$$p_n(k) = k_B T \langle \rho(k) \rangle - \frac{1}{V_{sl}} \left\langle \sum_{(i,j)}^{(k)} \frac{z_{ij}^2}{r_{ij}} \frac{dU(r_{ij})}{dr} \right\rangle, \quad (\text{L.2.1})$$

and

$$p_t(k) = k_B T \langle \rho(k) \rangle - \frac{1}{2V_{sl}} \left\langle \sum_{(i,j)}^{(k)} \frac{x_{ij}^2 + y_{ij}^2}{r_{ij}} \frac{dU(r_{ij})}{dr} \right\rangle, \quad (\text{L.2.2})$$

where $\langle \rho(k) \rangle$ is the average density in slab k , $V_{sl} = L_x L_y L_z / N_{sl}$ is the volume of a slab, $U(r)$ is the intermolecular potential from which the conservative forces can be derived. $\sum_{(i,j)}^{(k)}$ means that the summation runs over all pairs of particles i, j for which the slab k (partially) contains the line that connects the particles i and j . Slab k gets a contribution $1/N_o$ from a given pair (i, j) , where N_o is the total number of slabs which intersect this line.

It can be shown that, even though the stress tensor is not unique, the definition of the interfacial tension γ is free from ambiguities [729]. The interfacial tension can be calculated by integrating the difference between the normal and tangential components of the pressure tensor across the interface. In the case of our system with two interfaces, γ reads

$$\gamma = \frac{1}{2} \int_o^{L_z} dz [p_n(z) - p_t(z)]. \quad (\text{L.2.3})$$

The factor $\frac{1}{2}$ corrects for the fact that, in a system with periodic boundary conditions, interfaces necessarily come in pairs.

L.3 Micro-canonical Monte Carlo

Most experimental observations are performed at constant N, P, T ; sometimes at constant μ, V, T ; and occasionally at constant N, V, T .

Experiments at constant N, V, E exist (e.g., in adiabatic calorimetry) but are rare, and it is fair to say that microcanonical MC simulations of dense liquids or solids are equally rare. In fact, the first microcanonical Monte Carlo method was suggested by Creutz [172], in the context of lattice gauge theory MC simulations. The microcanonical Monte Carlo method does not use random numbers to determine the acceptance of a move, but it does use random numbers to generate trial moves.

The constant NVE MC algorithm uses the following procedure. We start with the system in a configuration \mathbf{q}^N . We denote the potential energy for this state by $U(\mathbf{q}^N)$. We now fix the total energy of the system at a value $E > U$. The excess energy E_D (the D stands for “demon”) is equal to $E - U$ is carried by an additional degree of freedom. E_D must be non-negative. Now we start our Monte Carlo run.

1. After each trial move from an “old” configuration (o) to a trial configuration (n) , we compute the change in potential energy of the system,

$$\Delta U = U(\mathbf{q}^N(n)) - U(\mathbf{q}^N(o)).$$

2. If $\Delta U < 0$, we accept the move and increase the energy carried by the demon by $|\Delta U|$. If $\Delta U > 0$, we test if the demon carries enough energy to make up the difference. Otherwise, we reject the trial move.

Note that no random numbers were used in this decision. Once the system has equilibrated, the probability density to find the demon with an energy E_D is given by the Boltzmann distribution:

$$\mathcal{N}(E_D) = (k_B T)^{-1} \exp(-E_D/k_B T),$$

where T is the temperature that the system reaches after equilibration. Hence, the demon acts as a thermometer. In that sense, the demon energy plays the same role as the kinetic energy in a micro-canonical MD simulation.

Micro-canonical Monte Carlo is rarely used to simulate molecular systems but it finds many other applications in cases where canonical MC simulations would fail, for instance in the study of systems interacting through gravity, as their potential energy is not bounded from below.

L.4 Details of the Gibbs “ensemble”

The introduction of a new ensemble brings up the question of whether it is a “proper ensemble”; that is, does it yield the same results as the conventional ensembles? To prove it does, we use the partition function (6.6.1) as derived in section 6.6.2 to define a free energy. This free energy is used to show that, in the thermodynamic limit, the Gibbs ensemble and the canonical ensemble are equivalent. This proof gives considerable insight into why the method works. Before we proceed, we first list a few basic results for the free energy in the canonical ensemble.

L.4.1 Free energy of the Gibbs ensemble

L.4.1.1 Basic definitions and results for the canonical ensemble

Consider a system of N particles in a volume V and temperature T (canonical ensemble). The partition function is defined as (see Ruelle [731])

$$Q(N, V, T) \equiv \frac{1}{\Lambda^{3N} N!} \int_V d\mathbf{r}^N \exp[-\beta \mathcal{U}(N)]. \quad (\text{L.4.1})$$

The free energy density is defined in the thermodynamic limit by

$$f(\rho) \equiv \lim_{\substack{V \rightarrow \infty \\ N/V = \rho}} f_V(\rho) \equiv \lim_{\substack{V \rightarrow \infty \\ N/V = \rho}} -\frac{1}{\beta V} \ln Q_{N,V,T},$$

where $\rho = N/V$ is the density of the system. For a finite number of particles we can write

$$Q(N, V, T) = \exp \{-\beta V [f(\rho) + o(V)]\}, \quad (\text{L.4.2})$$

where $g(V) = o(V)$ means $g(V)/V$ approaches 0 as $V \rightarrow \infty$. With this free energy, we can derive some interesting properties of a canonical system in the thermodynamic limit.

For example, it can be shown that this free energy is a convex function of the density ρ [731]:

$$f(x\rho_1 + (1-x)\rho_2) \leq xf(\rho_1) + (1-x)f(\rho_2), \quad (\text{L.4.3})$$

for every ρ_1 , ρ_2 , and x where $0 \leq x \leq 1$. The equality holds in the case of a first-order transition, if $\rho_g \leq \rho_1 \leq \rho_2 \leq \rho_l$, where ρ_g , ρ_l denote the density of coexisting gas and liquid phases, respectively.

Another interesting result, which plays a central role on the following pages, is the well-known saddle point theorem [732] (also called the *steepest descent method*). This theorem is based on the observation that, for a macroscopic system (N very large) in equilibrium, the probability that the free energy density deviates from its minimum value is extremely small. Therefore, when we calculate for such a system an ensemble average, we have to take into account only those contributions where the free energy has its minimum value. Assume that $Q(N, V, T)$ can be written as

$$Q(N, V, T) \equiv \int da_1, \dots, da_m \exp [-\beta V (f_m(a_1, \dots, a_m) + o(V))],$$

where a_1, \dots, a_m are variables that characterize the thermodynamic state of the system. Furthermore, define

$$f(\rho) \equiv \min_{a_1, \dots, a_m} f_m(a_1, \dots, a_m)$$

and assume that $f_m(a_1, \dots, a_m)$ and the term $o(V)$ satisfy a few technical conditions [732], which hold for most statistical mechanics systems. The saddle point theorem states that, in the thermodynamic limit, the free energy of the system is equal to this minimum value $f(\rho)$ or

$$\lim_{\substack{V \rightarrow \infty \\ N/V = \rho}} -\frac{1}{\beta V} \ln Q(N, V, T) = f(\rho). \quad (\text{L.4.4})$$

e8 Miscellaneous methods

Moreover, this saddle point theorem can also be used to calculate the ensemble average of a quantity A :

$$\begin{aligned} \langle A(a_1, \dots, a_m) \rangle_V &\equiv \frac{1}{Q(N, V, T)} \int da_1, \dots, da_m \\ &\times A(a_1, \dots, a_m) \exp \{-\beta V [f_m(a_1, \dots, a_m) + o(V)]\}. \end{aligned} \quad (\text{L.4.5})$$

In the thermodynamic limit, this ensemble average again has contributions only from those configurations where $f_m(a_1, \dots, a_m)$ has its minimum value. Let us define S as the collection of these minima:

$$S = \left\{ y_1, \dots, y_m \left| f_m(y_1, \dots, y_m) = \min_{a_1, \dots, a_m} f_m(a_1, \dots, a_m) \right. \right\}.$$

We now can state the saddle point theorem in a convenient form by introducing a function $G(a_1, \dots, a_m) \geq 0$ with support on the surface S and normalization

$$\int_S da_1, \dots, da_m G(a_1, \dots, a_m) = 1,$$

such that, for an arbitrary function A ,

$$\begin{aligned} \langle A(a_1, \dots, a_m) \rangle &\equiv \lim_{V \rightarrow \infty} \langle A(a_1, \dots, a_m) \rangle_V \\ &= \int_S da_1, \dots, da_m G(a_1, \dots, a_m) A(a_1, \dots, a_m). \end{aligned} \quad (\text{L.4.6})$$

L.4.1.2 The free energy density in the Gibbs ensemble

The Gibbs ensemble is introduced in section 6.6.2 as an N, V, T ensemble to which an additional degree of freedom is added: the system is divided into two subsystems that have *no* interaction with each other. We can rewrite the partition function of the canonical ensemble (L.4.1):

$$\begin{aligned} Q(N, V, T) &= \frac{1}{\Lambda^{3N} N!} \sum_{n_1=0}^N \binom{N}{n_1} \int_0^V dV_1 \int d\mathbf{r}_1^{n_1} \int d\mathbf{r}_2^{N-n_1} \exp \{-\beta [\mathcal{U}(n_1) \\ &+ \mathcal{U}(N - n_1) + \text{interactions between the two volumes}]\}. \end{aligned} \quad (\text{L.4.7})$$

The difference between this equation and the partition function of the Gibbs ensemble (6.6.1) is that, in Eq. (L.4.7), we have interactions between the subsystems. In the case of short-range interactions, the last term in the exponent of Eq. (L.4.7) is proportional to a surface term. This already suggests that both ensembles should behave similarly in many respects. We work out these ideas more rigorously in the following pages.

In the usual way, we define, as a free energy in the Gibbs ensemble,

$$\bar{f}(\rho) \equiv \lim_{\substack{V \rightarrow \infty \\ N/V = \rho}} -\frac{1}{\beta V} \ln \bar{Q}_{N,V,T}. \quad (\text{L.4.8})$$

In the partition function of the Gibbs ensemble (6.6.1), we can substitute Eq. (L.4.1):

$$\bar{Q}(N, V, T) = \sum_{n_1=0}^N \int_0^V dV_1 Q(n_1, V_1, T) Q(N - n_1, V - V_1, T).$$

Introducing $x = N_1/N$ and $y = V_1/V$, and assuming that the number of particles is very large, we can then write

$$\bar{Q}(N, V, T) = NV \int_0^1 dx \int_0^1 dy \bar{Q}_N(x, y),$$

where

$$\begin{aligned} \bar{Q}_N(x, y) &= Q_{xN, yV, T} Q_{(1-x)N, (1-y)V, T} \\ &= \exp \left\{ -\beta V \left[y f \left(\frac{x}{y} \rho \right) + (1-y) f \left(\frac{1-x}{1-y} \rho \right) + o(V) \right] \right\}. \end{aligned}$$

Note that, in this equation, $f(\rho)$ is the free energy of a canonical system. So, we can apply the saddle point theorem of the previous section (L.4.4) to calculate the free energy density of the Gibbs ensemble $\bar{f}(\rho)$

$$\bar{f}(\rho) = \min_{\substack{0 \leq x \leq 1 \\ 0 \leq y \leq 1}} \left[y f \left(\frac{x}{y} \rho \right) + (1-y) f \left(\frac{1-x}{1-y} \rho \right) \right] \equiv \min_{\substack{0 \leq x \leq 1 \\ 0 \leq y \leq 1}} \bar{f}(x, y).$$

We now have to find the surface S on which the function $\bar{f}(x, y)$ reaches its minimum. For this, we can use that $f(\rho)$ is a convex function of the density (L.4.3). This gives, for $\bar{f}(x, y)$,

$$\bar{f}(x, y) \geq f \left(y \frac{x}{y} \rho + (1-y) \frac{1-x}{1-y} \rho \right) = f(\rho). \quad (\text{L.4.9})$$

We first consider the case where there is only one phase. For this case any combination of x and y that results in densities ρ_1 and ρ_2 in the subsystems different from ρ will give a higher free energy. So, the equality in Eq. (L.4.9) holds only if

$$\frac{x}{y} \rho = \frac{1-x}{1-y} \rho, \quad \text{or } x = y.$$

e10 Miscellaneous methods

Thus, when there is only one phase, the free energy of the Gibbs ensemble has its minimum value (in the thermodynamic limit) when both boxes have a density equal to the equilibrium density of the canonical ensemble. Therefore, the surface S is given by

$$S = \{(x, y) | x = y\}.$$

Second, we consider the case of a first-order phase transition. Let ρ be such that $\rho_l \leq \rho \leq \rho_g$, and let us choose x and y such that

$$\rho_g \leq \frac{x}{y}\rho \equiv \rho_3 \leq \rho_l \quad \text{and} \quad \rho_g \leq \frac{1-x}{1-y}\rho \equiv \rho_4 \leq \rho_l. \quad (\text{L.4.10})$$

For this case the equality in Eq. (L.4.3) holds, and we can write, for $\bar{f}(x, y)$,

$$\begin{aligned} \bar{f}(x, y) &= yf(\rho_3) + (1-y)f(\rho_4) \\ &= f(y\rho_3 + (1-y)\rho_4). \end{aligned} \quad (\text{L.4.11})$$

Note that

$$(y\rho_3 + (1-y)\rho_4) = \rho, \quad (\text{L.4.12})$$

which gives

$$\bar{f}(x, y) = f(\rho). \quad (\text{L.4.13})$$

It can be shown that, if x, y do not satisfy Eq. (L.4.10),

$$\bar{f}(x, y) > f(\rho).$$

Therefore, the surface S in the case of a first-order phase transition is given by

$$S = \left\{ (x, y) \left| \rho_g \leq \frac{x}{y}\rho \leq \rho_l, \quad \rho_g \leq \frac{1-x}{1-y}\rho \leq \rho_l \right. \right\}. \quad (\text{L.4.14})$$

This result shows that, in the case of a first-order transition, the (bulk) free energy of the Gibbs ensemble has its minimum value (in the thermodynamic limit) for all values of x, y where there is vapor-liquid coexistence in *both* boxes.

Eqs. (L.4.9) and (L.4.13) show that, in the thermodynamic limit, the free energy of the Gibbs ensemble is equal to the free energy of the canonical ensemble. To calculate an ensemble average, it remains to determine the function $G(x, y)$ using Eq. (L.4.6).

In the case of a pure phase $G(x, y)$ needs to be of the form

$$G(x, y) = g(x) \delta(x - y). \quad (\text{L.4.15})$$

It is shown in the Appendix of [220] that, for an ideal gas, $g(x) = 1$. We expect that the same holds for an interacting gas. Fig. L.1 shows a probability plot in

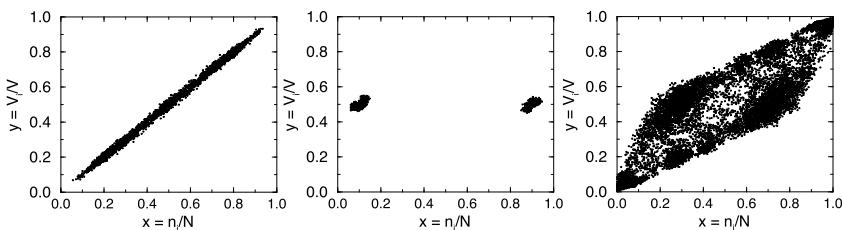


FIGURE L.1 Probability plot in the x , y plane ($x = n_1/N$, $y = V_1/V$ and $x = (N - n_1)/N$, $y = (V - V_1)/V$) for a Lennard-Jones fluid at various temperatures: (left) high temperature ($T = 10$), (middle) well below the critical temperature ($T = 1.15$), and (right) slightly below the critical temperature ($T = 1.30$).

the x , y plane for a simulation of a finite system at high temperature. This figure shows that $x \approx y$.

In the case of two phases, we will show that the system will split up into a liquid phase, with density, ρ_l , in one box, and a vapor phase, with density, ρ_g , in the other.

Until now we have ignored surface effects, which arise from the presence of a liquid-vapor interface in the boxes. When the density in one box is between the vapor and liquid density the system will form droplets of gas or liquid. The interfacial free energy associated with these droplets has (in the thermodynamic limit) a negligible contribution to the bulk free energy of the Gibbs ensemble. Nevertheless, this surface-free energy is the driving force that causes the system to separate into a homogeneous liquid in one box and a homogeneous vapor phase in the other.

These surface effects are taken into account in the next significant term in the expression for the free energy (L.4.2), which is the term due to the surface tension. This gives, for the partition function,

$$Q(N, V, T) = \exp \{-\beta [Vf(\rho) + \gamma A + o(A)]\}, \quad (\text{L.4.16})$$

where A denotes the area of the interface and γ denotes the interfacial tension. For three-dimensional systems, in general this area will be proportional to $V^{2/3}$. Using this form of the partition function for the Gibbs ensemble, Eq. (L.4.5) can be written as

$$\begin{aligned} & \langle A(x, y) \rangle_V \\ &= \frac{\iint dx dy A(x, y) \exp \{-\beta [Vf(x, y) + \gamma V^{2/3}a(x, y) + o(V^{2/3})]\}}{Q(N, V, T)}, \end{aligned} \quad (\text{L.4.17})$$

where $a(x, y)$ is a function of the order of unity.

We know from the saddle point theorem that the most important contribution to the integrals comes from the region S , defined by Eq. (L.4.14). Thus,

e12 Miscellaneous methods

$$\begin{aligned} & \langle A(x, y) \rangle_V \\ & \approx \frac{\iint_S dx dy A(x, y) \exp \{-\beta [Vf(x, y) + \gamma V^{2/3}a(x, y) + o(V^{2/3})]\}}{\iint_S dx dy \exp \{-\beta [Vf(x, y) + \gamma V^{2/3}a(x, y) + o(V^{2/3})]\}} \\ & = \frac{\iint_S dx dy A(x, y) \exp \{-\beta [\gamma V^{2/3}a(x, y) + o(V^{2/3})]\}}{\iint_S dx dy \exp \{-\beta [\gamma V^{2/3}a(x, y) + o(V^{2/3})]\}} \end{aligned} \quad (\text{L.4.18})$$

and applying the saddle point theorem again

$$\langle A(x, y) \rangle_V \approx \frac{\iint_{S_A} dx dy A(x, y) \exp [-\beta \gamma V^{2/3}a(x, y) + o(V^{2/3})]}{\iint_{S_A} dx dy \exp [-\beta \gamma V^{2/3}a(x, y) + o(V^{2/3})]} \quad (\text{L.4.19})$$

and

$$\lim_{V \rightarrow \infty} \langle A(x, y) \rangle_V = \iint_{S_A} dx dy G(x, y) A(x, y), \quad (\text{L.4.20})$$

where the surface S_A is now given by

$$S_A = \left\{ (x, y) \left| a(x, y) = \min_{\bar{x}, \bar{y}} a(\bar{x}, \bar{y}) \right. \right\}. \quad (\text{L.4.21})$$

In the infinite system it is easily seen that the area of the interface is 0, if box 1 contains only gas (liquid) and box 2 only liquid (gas). Therefore, the surface S_A contains only two points, which correspond to the vapor and liquid densities:

$$S_A = \left\{ (x, y) \left| \frac{x}{y} = \rho_l \text{ and } \frac{1-x}{1-y} = \rho_g \text{ or } \frac{x}{y} = \rho_g \text{ and } \frac{1-x}{1-y} = \rho_l \right. \right\}. \quad (\text{L.4.22})$$

It is straightforward to show that this surface gives, for $G(x, y)$,

$$\begin{aligned} G(x, y) &= \frac{1}{2} \delta \left(x - \frac{\rho_g}{\rho} \frac{\rho - \rho_g}{\rho_l - \rho_g} \right) \delta \left(y - \frac{\rho - \rho_g}{\rho_l - \rho_g} \right) \\ &\quad + \frac{1}{2} \delta \left(x - \frac{\rho_l}{\rho} \frac{\rho_l - \rho}{\rho_l - \rho_g} \right) \delta \left(y - \frac{\rho_l - \rho}{\rho_l - \rho_g} \right). \end{aligned} \quad (\text{L.4.23})$$

We have shown more formally that the free energy density for the Gibbs ensemble, as defined by Eq. (L.4.8), becomes identical to the free energy density of the canonical ensemble. Furthermore, it is shown that, at high temperatures, $x = y$; that is, the densities in the two subsystems of the Gibbs ensemble are equal and equal to the density in the canonical ensemble (see Fig. L.1).

In the case of a first-order phase transition, if surface terms would be unimportant, then x and y are restricted to the area defined by Eq. (L.4.14):

$$\rho_g \leq \frac{x}{y} \rho \equiv \rho_3 \leq \rho_l \quad \text{and} \quad \rho_g \leq \frac{1-x}{1-y} \rho \equiv \rho_4 \leq \rho_l. \quad (\text{L.4.24})$$

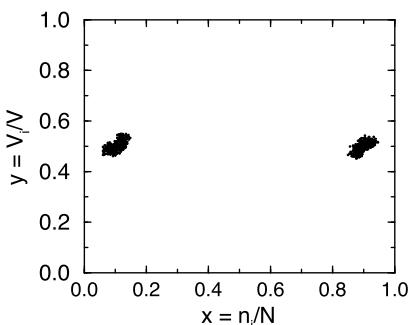


FIGURE L.2 Probability plot in the $x - y$ plane of a successful simulation of a Lennard-Jones fluid well below the critical temperature ($T = 1.15$ and $N = 500$).

If we take surface effects into account, this surface (Eq. (L.4.24)) reduces to two points in the x, y plane. The densities of these points correspond to the density of the gas or liquid phase in the canonical ensemble.

It is interesting to compare this with the results of an actual simulation of a finite system. In Fig. L.1, the results are shown for a simulation at a temperature well below the critical point. Under such conditions, the surface reduces to two points. This should be compared to the results of a simulation close to the critical point (Fig. L.1). Under such conditions the interfacial tension is very small and we see that the simulation samples the entire surface S . Note that due to the finite size of this system, fluctuations are also possible in which the density of a subsystem becomes greater or smaller than the density of the liquid or gas phase.

L.4.2 Graphical analysis of simulation results

In Appendix L.4, we describe a graphical technique for analyzing the results of a Gibbs ensemble simulation. In this scheme, the fraction of all particles (n_i/N) in box i is plotted versus the fraction of the total volume (v_i/V) taken up by this box. In the x - y plane, where $x = n_i/N$ and $y = V_i/V$, every dot represents a point sampled in the simulation.

In the thermodynamic limit, only two points in the x - y plane are sampled; namely, those that correspond to the coexisting liquid and gas density (see Appendix L.4). For a finite system, we expect to observe fluctuations around these points. Fig. L.2 shows an x - y plot for a simulation of two-phase coexistence well below the critical temperature. The fact that the simulation results cluster around the two points that correspond to the coexisting liquid and vapor indicates that the system was well equilibrated. If a simulation in the Gibbs ensemble is performed far below the critical temperature, it is in general no problem to analyze the results. After the equilibration, it becomes clear which of the boxes contains the vapor phase and which the liquid phase. The densities of the coexisting phases can simply be obtained by sampling the densities at regular intervals.

e14 Miscellaneous methods

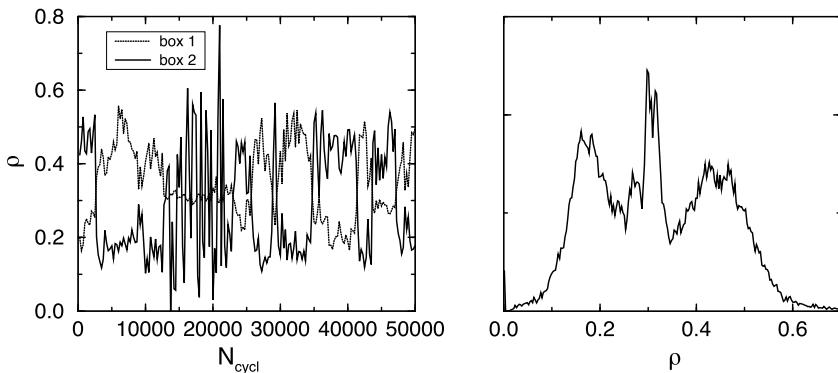


FIGURE L.3 Density in the two boxes in a Gibbs ensemble simulation close to the critical temperature. The left figure shows the evolution of the density of the two boxes during a simulation. The right figure gives the corresponding probability density. The simulations were performed on a Lennard-Jones fluid with $N = 256$ at $T = 1.30$.

When estimating the accuracy of the simulation one should be careful since the “measured” densities are not sampled independently: in estimating the standard deviations of the results one should take this into account (this aspect is discussed in more detail in Appendix A of [733]).

Close to the critical point, however, it is possible that the boxes continuously change “identity” during a simulation. In Fig. L.3 the evolution of the density in such a simulation close to the critical point is shown. In such a system, the average density in any one of the two boxes will tend to the overall density (N/V). In those circumstances, it is more convenient to construct a histogram of the probability density $P(\rho)$ to observe a density ρ in either box. Even when the boxes change identity during the simulation, the maxima of $P(\rho)$ are still well defined. And, as shown in Appendix L.4, in the thermodynamic limit, the two maxima of $P(\rho)$ correspond to coexisting vapor and liquid densities, except precisely at the critical point. (For a discussion of the critical behavior of $P(\rho)$, see the article by Allen and Tildesley [46].) Because $P(\rho)$ is obtained by sampling the density in both boxes, the results are not influenced when the boxes change identity. In Fig. L.3 an example of such a density distribution is shown. In this particular example, the simulation was carried out rather close to the critical point. Under those conditions, the interpretation of the density histogram is complicated because interfaces may form in both boxes. As a consequence, three peaks are observed; the two outside peaks correspond to the coexisting liquid and gas phase. A simple model that accounts for the existence of the middle peak is discussed in [220].

Example 30 (Finite-size effects in the Gibbs ensemble). Most Gibbs-ensemble simulations are performed on relatively small systems ($64 \leq N \leq 500$). One

therefore would expect to see significant finite-size effects, in particular, close to the critical point. Indeed, in simulations of a system of 100 Ising spins on a lattice,^a phase coexistence is observed at temperatures as much as 25% above the critical temperature of the infinite system. In contrast to what is found in lattice gases, the first Gibbs-ensemble studies of the phase diagram of the Lennard-Jones fluid (in two and three dimensions) [77,211,213,220] did not show significant finite-size effects. This striking difference with the lattice models motivated Mon and Binder [734] to investigate the finite-size effects in the Gibbs ensemble for the two-dimensional Ising model in detail. For the two-dimensional Ising model the critical exponents and critical temperature are known exactly. Mon and Binder determined for various system sizes L the order parameter $M_L(T)$ (see Eq. (6.6.13)):

$$M_L(T) = \frac{\rho_l(T) - \rho_c}{\rho_c} = (1 - T/T_c)^\beta,$$

where $\rho_l(T)$ is the density of the liquid phase, ρ_c and T_c are the critical density and temperature, respectively, and β is the critical exponent.

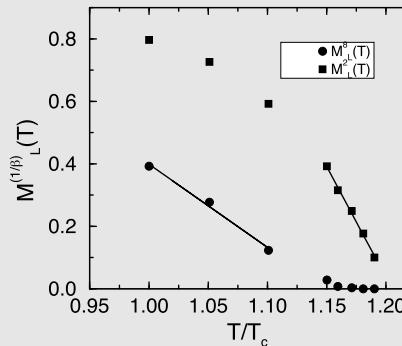


FIGURE L.4 Finite-size effects in a Gibbs-ensemble simulation of the two-dimensional Ising model. Order parameter $M_L^{1/\beta}(T)$ for $L = 10$ (i.e., $L \times L = 100$ spins) versus T/T_c , where T_c is the exact critical temperature for the infinite system. The lines are fitted through the points. The simulation data are taken from [734].

The results of the simulations of Mon and Binder are shown in Fig. L.4, in which the order parameter $M_L(T)$ is plotted as $M_L^{1/\beta}(T)$ versus T/T_c . Such a plot of the order parameter allows us to determine the effective critical exponent of the system. If the system behaves classically, the critical exponent has the mean field value $\beta = 1/2$ and we would expect a linear behavior of $M_L^2(T)$. On the other hand, if the system shows nonclassical behavior, with exponent $\beta = 1/8$, we would expect a straight line for $M_L^8(T)$. Fig. L.4 shows that, away from the critical point, the temperature dependence of the order parameter is best described with an exponent $\beta = 1/8$. Closer to the critical point, the mean field exponent $\beta = 1/2$ fits the data better. This behavior is as expected. Away from the critical point the system can accommodate all relevant

vant fluctuations and exhibits nonclassical behavior. But close to the critical point the system is too small to accommodate all fluctuations and, as a consequence, mean field behavior is observed. In addition, Fig. L.4 shows that we still can observe vapor-liquid coexistence at temperatures 20% above the critical temperature of the infinite system, which implies significant finite-size effects. The study of Mon and Binder shows that, in a lattice model of a fluid, finite-size effects on the liquid-vapor coexistence curve are very pronounced. It is important to note that, in this lattice version of the Gibbs ensemble, we do not change the volume and therefore fewer fluctuations are possible than in the off-lattice version.

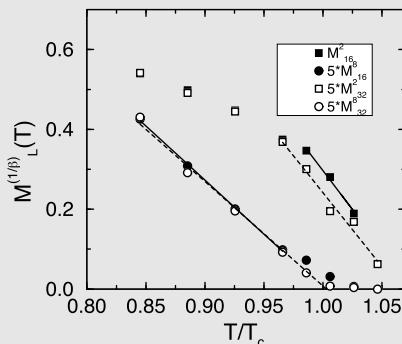


FIGURE L.5 Finite-size effects in the liquid-vapor coexistence curve of the two-dimensional Lennard-Jones fluid (truncated potential $r_c = 5.0\sigma$) studied by Gibbs-ensemble simulation. The order parameter M corresponds to the density difference between the coexisting liquid and vapor phases. The figure shows $M_L^{1/\beta}(T)$ versus T/T_c for various system sizes L . T_c is the estimated critical temperature for the infinite system ($T_c = 0.497 \pm 0.003$). The simulation data are taken from [222].

The striking differences between the findings of Mon and Binder and the results of the early simulations of the Lennard-Jones fluid motivated Panagiotopoulos to reinvestigate in some detail the finite-size effects of Gibbs-ensemble simulations of the two- and three-dimensional Lennard-Jones fluid [222]. The results of the simulations of Panagiotopoulos are shown in Fig. L.5. For the Lennard-Jones fluid, the order parameter is defined as

$$M_L(T) = \rho_l - \rho_g.$$

The results for the two-dimensional Lennard-Jones fluid are qualitatively similar to the results of Mon and Binder. At low temperatures, Ising-like behavior is observed and close to the critical point mean-field-like behavior. An important difference is the magnitude of the finite-size effects. Fig. L.5 shows that, for the two system sizes, the results are very similar; the finite size effects are at most 5%. In addition, Fig. L.5 also indicates why the initial Gibbs-ensemble studies on the Lennard-Jones fluids did not show significant finite-size effects. All these studies used Eqs. (6.6.12) and (6.6.13) to determine the critical point.

If we use these equations we implicitly assume nonclassical behavior up to the critical point. In Fig. L.5, this corresponds to extrapolating the lines, fitted to the data point, for $\beta = 1/8$. Extrapolation of these lines to $M_L^8(T) = 0$ gives a critical point that is not only independent of this system size but also very close to the true critical point of the infinite system.

For the three-dimensional Lennard-Jones fluid Panagiotopoulos did not observe a crossover from Ising-like to mean field behavior in the temperature regime that could be studied conveniently in the Gibbs ensemble ($T < 0.98T_c$). Also for liquid-liquid equilibria for the square well fluid, Recht and Panagiotopoulos [735] and de Miguel et al. [246] did not observe such a crossover. Moreover, for the three-dimensional Lennard-Jones fluid, the finite-size effects were negligible away from T_c and very small close to T_c .

^a The Ising model is equivalent to a lattice-gas model of a fluid. The latter model is the simplest that exhibits a liquid-vapor transition.

L.4.3 Chemical potential in the Gibbs ensemble

One of the steps in the Gibbs ensemble involves the insertion of a particle in one of the boxes. During this step, the energy of this particle has to be calculated (see section 6.6.3). Since this energy corresponds to the energy of a test particle, we can use the Widom insertion method [736] to calculate the chemical potential without additional costs [213]. At this point it is important to note that the Gibbs method requires no computation of the chemical potentials. However, to test whether the system under consideration has reached equilibrium or for comparison with other results, it is important to calculate the chemical potential of the individual phases correctly. The original Widom expression is valid only in the N, V, T ensemble and can be modified for applications in other ensembles (see section 8.5.1). Here we derive an expression for the chemical potential for the Gibbs ensemble. We restrict ourselves to temperatures sufficiently far below the critical temperature that the two boxes, after equilibration, do not change identity. For the more general case, we refer to [217].

If we rescale the coordinates of the particles with the box length, the partition function for the Gibbs ensemble (6.6.1) becomes

$$\begin{aligned} \bar{Q}_{N,V,T} &\equiv \frac{1}{V\Lambda^{3N}N!} \sum_{n_1=0}^N \binom{N}{n_1} \int_0^V dV_1 V_1^{n_1} (V - V_1)^{N-n_1} \\ &\quad \times \int d\mathbf{s}_1^{n_1} \exp[-\beta\mathcal{U}_1(n_1)] \int d\mathbf{s}_2^{N-n_1} \exp[-\beta\mathcal{U}_2(N-n_1)] \\ &= \sum_{n_1=0}^N \int_0^V dV_1 V_1^{n_1} (V - V_1)^{N-n_1} Q_1(n_1, V_1) Q_2(N - n_1, V - V_1), \end{aligned} \tag{L.4.25}$$

where $\mathbf{s} = \mathbf{r}/L$ is the scaled coordinates of a particle, L is the box length of the subsystem in which the particle is located, and $Q_i(n_i, V_i)$ is the partition of the canonical ensemble (see also section L.4.1.2).

The chemical potential of box 1 can be defined as

$$\begin{aligned}\mu_1 \equiv -k_B T \ln \sum_{n_1=0}^N \int_0^V dV_1 V_1^{n_1} (V - V_1)^{N-n_1} \\ \times \left[\frac{Q_1(n_1 + 1, V_1)}{Q_1(n_1, V_1)} \right] Q_2(N - n_1, V - V_1).\end{aligned}\quad (\text{L.4.26})$$

For the ratio of the partition functions of box 1, we can write

$$\begin{aligned}\frac{Q_1(n_1 + 1, V_1)}{Q_1(n_1, V_1)} &= \frac{V_1}{(n_1 + 1)\Lambda^3} \frac{\int d\mathbf{s}_1^{n_1+1} \exp[-\beta\mathcal{U}_1(n_1 + 1)]}{\int d\mathbf{s}_1^{n_1} \exp[-\beta\mathcal{U}_1(n_1)]} \\ &= \frac{V_1}{(n_1 + 1)\Lambda^3} \frac{\int d\mathbf{s}_1^{n_1} \exp[-\beta\Delta\mathcal{U}_1^+] \exp[-\beta\mathcal{U}_1(n_1)]}{\int d\mathbf{s}_1^{n_1} \exp[-\beta\mathcal{U}_1(n_1)]},\end{aligned}\quad (\text{L.4.27})$$

in which we have used the notation

$$\mathcal{U}_1(n_1 + 1) = \Delta\mathcal{U}_1^+ + \mathcal{U}_1(n_1),$$

where $\Delta\mathcal{U}_1^+$ is the test particle energy of a (ghost) particle in box 1. We can write Eq. (L.4.26) as an ensemble average restricted to box 1:

$$\mu_1 = -k_B T \ln \frac{1}{\Lambda^3} \left\langle \frac{V_1}{n_1 + 1} \exp[-\beta\Delta\mathcal{U}_1^+] \right\rangle_{\text{Gibbs, box 1}}, \quad (\text{L.4.28})$$

where $\langle \cdots \rangle_{\text{Gibbs, box } i}$ denotes an ensemble average in the Gibbs ensemble restricted to box i (note that this ensemble average is well defined if the boxes do not change identity during a simulation [217]).

L.4.4 Algorithms of the Gibbs ensemble

L.5 Multi-canonical ensemble method

In the multi-canonical ensemble method, we extend our ensemble by using the energy space [350]. In the previous sections, we used different tricks to obtain information on parts of phase space that are rarely sampled in conventional simulations. The idea of a multi-canonical ensemble is to ensure that all energies are sampled uniformly. The probability of finding a system with a particular energy is given by

$$\mathcal{P}(U) = \mathcal{N}(U)w(U),$$

Algorithm 42 (Attempt to change the volume in the Gibbs ensemble)

function mvvol	trial volume change of box 1/2
vol=box1**3	at constant total volume
vo2=v-vol	volume box 1 (diameter box1)
enol=toterg(vol,1)	...and 2
eno2=toterg(vo2,2)	energy old conf. box 1
lnvn=log(vol/vol2)+	and 2
+ (\mathcal{R} -0.5)*vmax	random walk in $\ln V_1 / V_2$
v1n=v*exp(lnvn)/(1+exp(lnvn))	new volume box 1 and 2
v2n=v-v1n	
box1n=v1n**(1/3)	new box length box 1
box2n=v2n**(1/3)	new box length box 2
for $1 \leq i \leq npart$ do	determine which box
if ibox(i) == 1 then	
fact=box1n/box1o	
else	
fact=box2n/box2o	
endif	
x(i)=x(i)*fact	rescale positions
enddo	
en1n=toterg(v1n,1)	total energy box 1
en2n=toterg(v2n,2)	total energy box 2
arg1=-beta*(en1n-enlo)	
+(npbox(1)+1)*log(v1n/v1o)	appropriate weight function
arg2=-beta*(en2n-en2o)	acceptance rule (6.6.10)
+(npbox(2)+1)*log(v2n/v2o)	
if $\mathcal{R} > \exp(arg1+arg2)$ then	REJECTED
do $i=1, npart$	determine which box
if ibox(i)== 1 then	
fact=box1o/box1n	
else	
fact=box2o/box2n	
endif	
x(i)=x(i)*fact	restore old configuration
enddo	
endif	
end function	

where $\mathcal{N}(U)$ is the density of states and $w(U)$ the particular weight in the ensemble. If we use $w(U) = \mathcal{C}/\mathcal{N}(U)$, we have a flat distribution of energies.

In practice, we do not know the density of states so we need to find this iteratively. For example, if we start with an ordinary N, V, T simulation, and

Specific Comments (for general comments, see p. 7)

1. The term `ibox(i) = 1` indicates that particle i is in box 1; `npart = npbox(1) + npbox(2)` where `npbox(i)` gives the number of particles in box i .
2. In this algorithm we perform a random walk in $\ln V$ and we use acceptance rule (6.6.10).
3. The function `toterg` is not shown explicitly: it is similar to Algorithm 5 and calculates the total energy of one of the two boxes. It requires information about the identity of the box (1 or 2) and the volume of these boxes (here assumed to be cubic).
In most cases the energy of the old configuration is known, and therefore it is not necessary to determine this energy at the beginning of the volume step.

we can make a histogram of the occurrence of a particular value of the energy $H_j(U)$, we can use this to improve our bias potential [737]:

$$\mathcal{U}_{j+1}^{\text{bias}}(U) = \mathcal{U}_j^{\text{bias}}(U) + \frac{1}{\beta} [\ln \bar{\mathcal{H}}_j - \ln \mathcal{H}_j]$$

where $\bar{\mathcal{H}}_j$ is the average value of histogram. Those energies that occur more than the average get an unfavorable bias, a higher energy, and those that occur lower than the average a favorable bias. If the histogram is flat, the bias potential does not change in the next iteration. To see how this can be used to compute a free-energy (difference) in the canonical ensemble we have:

$$\begin{aligned} \mathcal{P}(U) &= \frac{\int d\Gamma^N \delta(U(\Gamma^N) - U) \exp(-\beta U(\Gamma^N))}{\int d\Gamma^N \exp(-\beta U(\Gamma^N))} \\ &= \exp(-\beta U) \frac{\int d\Gamma^N \delta(U(\Gamma^N) - U)}{\mathcal{C}_0} \end{aligned}$$

and in the multi-canonical ensemble, we have

$$\begin{aligned} \mathcal{P}^{\text{MulCan}}(U) &= \frac{\int d\Gamma^N \delta(U(\Gamma^N) - U) \exp[-\beta(U(\Gamma^N) + \mathcal{U}^{\text{bias}}(U(\Gamma^N)))]}{\int d\Gamma^N \exp[-\beta(U(\Gamma^N) + \mathcal{U}^{\text{bias}}(U(\Gamma^N)))]} \\ &= \exp[-\beta(U + \mathcal{U}^{\text{bias}}(U))] \frac{\int d\Gamma^N \delta(U(\Gamma^N) - U)}{\mathcal{C}_{\text{MulCan}}} \end{aligned}$$

Or, for the ratio of these two probabilities:

$$\frac{\mathcal{P}(U)}{\mathcal{P}^{\text{MulCan}}(U)} = C \frac{\exp(-\beta U)}{\exp[-\beta(U + \mathcal{U}^{\text{bias}}(U))]} = C \exp[\beta \mathcal{U}^{\text{bias}}(U)].$$

Algorithm 43 (Attempt to swap a particle between the two boxes)

<pre> function mcsswap if $\mathcal{R} < 0.5$ then in=1 out=2 else in=2 out=1 endif xn=$\mathcal{R} \cdot \text{box}(in)$ enn = ener(xn, in) w(in)=w(in)+vol(in)* + $\exp(-\beta \cdot enn) / (\text{npbox}(in)+1)$ if ($\text{npbox}(out) == 0$) return ido=0 while ido \neq out do o=int(npert$\cdot \mathcal{R}$)+1 ido=ibox(o) end while eno= ener(x(o), out) arg=$\exp(-\beta \cdot (enn - eno) +$ + $\log(vol(out) * (\text{npbox}(in)+1) /$ + $(vol(in) * \text{npbox}(out))) / \beta)$ if $\mathcal{R} < arg$ then x(o)=xn ibox(o)=in nbox(out)=npbox(out)-1 nbox(in)=npbox(in)+1 endif end function</pre>	attempts to swap a particle between the two boxes which box to add or remove new particle at a random position energy new particle in box in update chemical potential (L4.28) if box empty return find a particle to be removed energy particle o in box out acceptance rule (6.6.11) add new particle to box in
--	---

Specific Comments (for general comments, see p. 7)

- 1.** $\text{ener}(x, ib)$ calculates the energy of a particle at position x in box ib . It carries a box label and is slightly different from the function called in Algorithm 2.
- 2.** We specify an additional argument for the function **ener**: the index of the box where we attempt to insert (**in**) or remove (**out**) a particle.
- 3.** The acceptance rule (6.6.11) is used in this algorithm.
- 4.** We also compute the Boltzmann factor associated with the random insertion of a particle, at virtually no added cost. At the end of the simulation, the excess chemical potential can be calculated from $w(\text{box})$ using $\mu_{\text{box}} = -\ln \langle w_{\text{box}} \rangle / \beta$, where $\langle w_{\text{box}} \rangle$ is the average of w_{box} accumulated during the run.

If we achieve an exactly flat distribution, $\mathcal{P}^{\text{MulCan}}(U) = C$, we have for the free energy:

$$-\beta F(U) = \ln \mathcal{P}(U) + C = \beta \mathcal{U}^{\text{bias}}(U).$$

This illustrates that in order to get a perfectly flat distribution, we need to apply a biasing potential that is the inverse of the free energy. In practice, we will never get a perfectly flat distribution, and the biasing potential becomes equivalent to the umbrella sampling technique, which we discuss in section 8.6.6.

L.6 Nosé-Hoover dynamics

L.6.1 Nosé-Hoover dynamics equations of motion

We now apply the methods of non-Hamiltonian dynamics to analyze the Nosé-Hoover algorithm and the Nosé-Hoover chains that are discussed in section 7.1.2.1 and Appendix L.6.1.2 Our discussion of these algorithms is only intended as an illustration that there exist systematic techniques for predicting the phase-space density generated by a particular non-Hamiltonian dynamics scheme. While we show a few simple examples, we refer the reader to ref. [267] for a more comprehensive discussion.

L.6.1.1 The Nosé-Hoover algorithm

In section 7.1.2.1 we showed that the Nosé-Hoover algorithm generates non-Hamiltonian dynamics. The Nosé-Hoover equations can be written as

$$\begin{aligned}\dot{\mathbf{r}}_i &= \mathbf{p}_i / m_i \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \frac{p_\xi}{Q} \mathbf{p}_i \\ \dot{\xi} &= p_\xi / Q \\ \dot{p}_\xi &= \sum_i \mathbf{p}_i^2 / m_i - k_B T L,\end{aligned}$$

where L is a parameter that has to be determined to generate the canonical distribution.

Implementation

In section 7.1.2.1, we showed how the introduction of an additional dynamical variable (s) in the Lagrangian could be used to perform MD simulations subject to a thermodynamic constraint (in this case, constant temperature). We stress that the importance of such extended-Lagrangian techniques transcends the specific application. In addition, the problems encountered in the numerical implementation of the Nosé scheme are representative of a wider class of algorithms (namely, those where forces depend explicitly on velocities). It is for this

reason that we discuss the numerical implementation of the Nosé thermostat in some detail (see also SI L.6.2).

The Nosé equations of motion can be written in terms of virtual variables or real variables. In a simulation based on the virtual variables, the real-time step is fluctuating, which complicates the sampling of averages. For this reason, it is better to use the real-variable formulation.

It is now common to use the Nosé scheme in the formulation of Hoover [257,738], who showed that the equations derived by Nosé could be written in a form that resembles Newtonian dynamics.

In Eqs. (7.1.27), (7.1.28), and (7.1.29), the variables s' , p'_s , and Q occur only as $s' p'_s / Q$. To simplify these equations, we can introduce the thermodynamic friction coefficient $\xi = s' p'_s / Q$. The equations of motion then become (dropping the primes on \mathbf{p}) and using dots to denote time derivatives)

$$\dot{\mathbf{r}}_i = \mathbf{p}_i / m_i \quad (\text{L.6.1})$$

$$\dot{\mathbf{p}}_i = -\frac{\partial \mathcal{U}(\mathbf{r}^N)}{\partial \mathbf{r}_i} - \xi \mathbf{p}_i \quad (\text{L.6.2})$$

$$\dot{\xi} = \left(\sum_i \mathbf{p}_i^2 / m_i - \frac{L}{\beta} \right) / Q \quad (\text{L.6.3})$$

$$\dot{s}/s = \frac{d \ln s}{dt} = \xi. \quad (\text{L.6.4})$$

Note that the last equation is redundant, since Eqs. (L.6.1)–(L.6.3) form a closed set. However, if we solve the equations of motion for s as well, we can use Eq. (7.1.30) as a diagnostic tool, since H' must be conserved during the simulation. In terms of the variables used in Eqs. (L.6.1)–(L.6.4), H reads

$$H_{\text{Nose}} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + \mathcal{U}(\mathbf{r}^N) + \frac{\xi^2 Q}{2} + L \frac{\ln s}{\beta}. \quad (\text{L.6.5})$$

As we use the real-variable formulation in this set of equations, we have to take $L = dN$.

An important implication of the Nosé equations is that in the Lagrangian (7.1.10), a logarithmic term ($\ln s$) is needed to have the correct scaling of time. Any scheme that does not rescale time will fail to recover the canonical ensemble.

Hoover [738] demonstrated that the equations of motion (L.6.1)–(L.6.3) are unique, in the sense that other equations of the same form cannot lead to a canonical distribution. In SI L.6.2, we discuss an efficient way of implementing the Nosé-Hoover scheme.

The equations of motion of the Nosé-Hoover scheme cannot be derived from a Hamiltonian. This implies that one cannot use the standard methods (see Appendix A) to make the time averages obtained with this dynamics map onto

ensemble averages of the standard form. In Appendix B, we discuss how one can analyze such non-Hamiltonian dynamics. The result of this analysis is that the conventional Nosé-Hoover algorithm only generates the correct distribution if there is a *single* constant of motion. Normally, the total energy defined by H_{Nose} , see Eq. (L.6.5), is conserved. But the existence of other conserved quantities creates a problem. This implies that one should not have any other conserved quantity, such as the total momentum. Momentum conservation is not an issue in simulations of systems confined by walls, or subject to another external potential that breaks momentum conservation.

However, if we simulate a system without external forces, momentum is conserved. Under those conditions, the Nosé-Hoover scheme can still be correct, provided that the center of mass remains fixed. This condition can be fulfilled if we set the initial velocity of the center of mass to zero. However, if we simulate systems with momentum conservation and a non-zero velocity of the center of mass, or if we have additional conserved quantities, then we must go beyond the simple NH algorithm and use chains of Nosé-Hoover thermostats [267,269] to obtain the correct canonical distribution.

L.6.1.2 Nosé-Hoover chains

The equations of motion for a system of N particles coupled with M Nosé-Hoover chains are given (in real variables, hence $L = dN$) by

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} \quad (\text{L.6.6})$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \frac{p_{\xi_1}}{Q_1} \mathbf{p}_i \quad (\text{L.6.7})$$

$$\dot{\xi}_k = \frac{p_{\xi_k}}{Q_k} \quad k = 1, \dots, M \quad (\text{L.6.8})$$

$$\dot{p}_{\xi_1} = \left(\sum_i \frac{p_i^2}{m_i} - Lk_B T \right) - \frac{p_{\xi_2}}{Q_2} p_{\xi_1} \quad (\text{L.6.9})$$

$$\dot{p}_{\xi_k} = \left[\frac{p_{\xi_{k-1}}^2}{Q_{k-1}} - k_B T \right] - \frac{p_{\xi_{k+1}}}{Q_{k+1}} p_{\xi_k} \quad (\text{L.6.10})$$

$$\dot{p}_{\xi_M} = \left[\frac{p_{\xi_{M-1}}^2}{Q_{M-1}} - k_B T \right]. \quad (\text{L.6.11})$$

For these equations of motion, the conserved energy is

$$H_{\text{NHC}} = \mathcal{H}(\mathbf{r}, \mathbf{p}) + \sum_{k=1}^M \frac{p_{\xi_k}^2}{2Q_k} + Lk_B T \xi_1 + \sum_{k=2}^M k_B T \xi_k. \quad (\text{L.6.12})$$

We can use this conserved quantity to check the integration scheme. It is important to note that the additional $M - 1$ equations of motion form a simple one-

dimensional chain and therefore are relatively simple to implement. In SI L.6.2, we describe an algorithm for a system with a Nosé-Hoover chain thermostat.

Example 31 (Nosé-Hoover chain for harmonic oscillator). The harmonic oscillator is the obvious model system on which we test the Nosé-Hoover chain thermostat. If we use a chain of two coupling parameters, the equations of motion are

$$\begin{aligned}\dot{r} &= v \\ \dot{v} &= -r - \xi_1 v \\ \dot{\xi}_1 &= \frac{v^2 - T}{Q_1} - \xi_1 \xi_2 \\ \dot{\xi}_2 &= \frac{Q_1 \xi_1^2 - T}{Q_2}.\end{aligned}$$

A typical trajectory generated with the Nosé-Hoover chains is shown in Fig. L.6. The distribution of the velocity and position of the oscillator are also shown in Fig. L.6. Comparison with the results obtained using the Andersen thermostat (see Case Study 13) shows that the Nosé-Hoover chains do generate a canonical distribution, even for the harmonic oscillator.

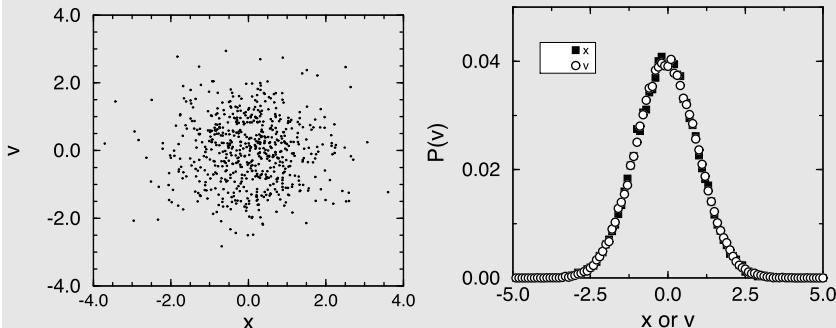


FIGURE L.6 Test of the phase space trajectory of a harmonic oscillator, coupled to a Nosé-Hoover chain thermostat. The left-hand side of the figure shows part of a trajectory: the dots correspond to consecutive points separated by 10,000 time steps. The right-hand side shows the distributions of velocity and position. Due to our choice of units, both distributions should be Gaussians of equal width.

The Fortran code to generate this Example can be found in the online-SI, Case Study 14.

To analyze the dynamics of this system, we have to determine the conservation laws and the non-driven variables. Let us consider first the case in which

e26 Miscellaneous methods

we assume that we only have conservation of energy, viz. Eq. (7.1.13)

$$\begin{aligned} H_{\text{Nose}} &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + \mathcal{H}(\mathbf{r}^N) + \frac{p_\xi^2}{2Q} + Lk_B T \xi \\ &= \mathcal{H}(\mathbf{r}, \mathbf{p}) + \frac{p_\xi^2}{2Q} + Lk_B T \xi = C_1, \end{aligned}$$

where $\mathcal{H}(\mathbf{r}, \mathbf{p})$ is the physical Hamiltonian. If we use $\Gamma = (\mathbf{r}^N, \mathbf{p}^N, \xi, p_\xi)$, the phase-space compressibility of this system can be written as

$$\begin{aligned} \kappa(\Gamma) &= \nabla_\Gamma \cdot \dot{\Gamma} \\ &= \sum_i \nabla_{\mathbf{r}_i} \cdot \dot{\mathbf{r}}_i + \sum_i \nabla_{\mathbf{p}_i} \cdot \dot{\mathbf{p}}_i + \nabla_\xi \cdot \dot{\xi} + \nabla_{p_\xi} \cdot \dot{p}_\xi \\ &= \sum_i \nabla_{\mathbf{p}_i} \cdot \dot{\mathbf{p}}_i \\ &= -dNp_\xi/Q = -dN\dot{\xi}. \end{aligned}$$

Hence, it follows that the metric \sqrt{g} is given by

$$\sqrt{g} = \exp \left[- \int \kappa dt \right] = \exp(dN\xi).$$

Substitution of this metric in the expression for the partition function gives:

$$\begin{aligned} \Omega_T(N, V, C_1) &= \int d^N \mathbf{p} \int d^N \mathbf{r} \int dp_\xi \int d\xi \\ &\quad \times \exp(dN\xi) \delta \left(\mathcal{H}(\mathbf{r}, \mathbf{p}) + \frac{p_\xi^2}{2Q} + Lk_B T \xi - C_1 \right). \end{aligned}$$

In this expression, the integration over ξ and p_ξ can be performed analytically. Because of the δ -function, integration over ξ gives as condition

$$\xi = \frac{1}{Lk_B T} \left(C_1 - \mathcal{H}(\mathbf{r}, \mathbf{p}) - \frac{p_\xi^2}{2Q} \right).$$

Substitution of this condition in the partition function gives

$$\begin{aligned} \Omega_T(N, V, C_1) &= \frac{1}{Lk_B T} \int d^N \mathbf{p} \int d^N \mathbf{r} \int dp_\xi \\ &\quad \times \exp \left[\frac{dN}{Lk_B T} \left(C_1 - \mathcal{H}(\mathbf{r}, \mathbf{p}) - \frac{p_\xi^2}{2Q} \right) \right] \\ &= \frac{\exp(dNC_1/Lk_B T)}{Lk_B T} \int dp_\xi \exp \left(-dNp_\xi^2/2QL \right) \end{aligned}$$

$$\times \int d^N \mathbf{p} \int d^N \mathbf{r} \exp(-\beta dN \mathcal{H}(\mathbf{r}, \mathbf{p})/L) \\ \propto Q(N, V, T),$$

where the last equality only holds provided that we choose L equal to dN . The integration over p_ξ yields a constant prefactor that has no physical importance. In section 7.1.2.1, we derived a similar result for the Nosé equations in terms of its real variables. The present demonstration that the Nosé-Hoover equations lead to a canonical distribution is completely different from Nosé's original argument, yet the end result is the same. Note, however, that we have assumed that there is only a single conservation law, viz. conservation of H_{Nose} . In general, there will be more conserved quantities. For instance, if we consider a system in the absence of external forces, then the total linear momentum is also conserved. This will affect the phase-space distribution. In the Nosé-Hoover dynamics, conservation of total momentum reads

$$\frac{d\mathbf{P}e^\xi}{dt} = e^\xi (\dot{\mathbf{P}} + \mathbf{P}\dot{\xi}) = e^\xi \left(\dot{\mathbf{P}} + \mathbf{P} \frac{p_\xi}{Q} \right) \\ = e^\xi \left(\sum_i \left(\mathbf{F}_i - \frac{p_\xi}{Q} \mathbf{p}_i \right) + \mathbf{P} \frac{p_\xi}{Q} \right) \\ = 0,$$

and hence

$$\mathbf{P}e^\xi = \mathbf{K}, \quad (\text{L.6.13})$$

where $\mathbf{P} = \sum_i \mathbf{p}_i$ is the center-of-mass momentum of the system and \mathbf{K} is an arbitrary constant vector.

To continue the analysis, we should eliminate the driven variables from our system. The center-of-mass position, \mathbf{R} , and momentum, \mathbf{P} , have no influence on the other variables. We can eliminate these by considering the positions and momenta relative to the center of mass of the system, \mathbf{r}' and \mathbf{p}' , respectively. Note, however, that the magnitude of the center-of-mass momentum is coupled to the other variables through a conservation law and cannot be eliminated from the analysis. The components of the center-of-mass momentum \mathbf{P} are linearly dependent.¹ Therefore of the d components, only one component can be chosen independently, or we can take as independent variable $P = (\sum_\alpha P_\alpha^2)^{1/2}$.

¹ To see this, consider the components of Eq. (L.6.13):

$$\frac{P_x}{K_x} = \frac{P_y}{K_y} = \frac{P_z}{K_z} = e^\xi,$$

which shows that only one of the components is independent.

e28 Miscellaneous methods

We now have to perform a transformation of our systems to the variables $\{\mathbf{p}', \mathbf{P}, \mathbf{r}', \mathbf{R}\}$; the resulting equations of motion are

$$\begin{aligned}\dot{\mathbf{r}}'_i &= \mathbf{p}'_i / m'_i \\ \dot{\mathbf{p}}'_i &= \mathbf{F}'_i - \frac{p_\xi}{Q} \mathbf{p}'_i \\ \dot{P}_i &= -\frac{p_\xi}{Q} P_i \\ \dot{\xi} &= \frac{p_\xi}{Q} \\ \dot{p}_\xi &= \sum_i^{N-1} \frac{\mathbf{p}'_i^2}{m'_i} + \frac{P^2}{M} - k_B T L.\end{aligned}$$

The equations of motion have two² conservation laws:

$$\begin{aligned}\mathcal{H}(\mathbf{p}', \mathbf{r}', P) + \frac{p_\xi^2}{2Q} + L k_B T \xi &= C_1 \\ P \exp(\xi) &= C_2.\end{aligned}$$

In the first conservation law, we have used

$$\mathcal{H}(\mathbf{r}', \mathbf{p}', P) = \sum_{i=1}^{N-1} \frac{\mathbf{p}'_i^2}{2m'_i} + \frac{P^2}{2M} + \mathcal{U}(\mathbf{r}'^N) = \mathcal{H}(\mathbf{r}, \mathbf{p}).$$

To compute the partition function, we have to determine the metric from the compressibility:

$$\begin{aligned}\kappa &= \sum_i^{N-1} \nabla_{\mathbf{r}'_i} \cdot \dot{\mathbf{r}}'_i + \sum_i^{N-1} \nabla_{\mathbf{p}'_i} \cdot \dot{\mathbf{p}}'_i + \nabla_P \cdot \dot{P} + \nabla_\xi \cdot \dot{\xi} + \nabla_{p_\xi} \cdot \dot{p}_\xi \\ &= -[d(N-1) + 1] \dot{\xi}.\end{aligned}$$

From which the metric follows directly:

$$\sqrt{g} = \exp\{[d(N-1) + 1] \xi\}.$$

The partition function Ω contains two δ -functions that express the two conservation laws:

$$\Omega_T(N, V, C_1, C_2) = \int d^{N-1} \mathbf{p}' \int d^{N-1} \mathbf{r}' \int dP \int dp_\xi \int d\xi$$

² Because we have replaced the d center-of-mass momenta components by a single variable P only one conservation law for the momenta is left.

$$\times \exp \{[d(N-1)+1]\xi\} \delta \left(\mathcal{H}(\mathbf{r}', \mathbf{p}', P) + \frac{p_\xi^2}{2Q} + Lk_B T \xi - C_1 \right) \\ \delta(e^\xi P - C_2).$$

The second δ -function imposes that

$$\xi = \ln(C_2/P).$$

Hence, integrating over ξ yields

$$\Omega_T(N, V, C_1, C_2) = \frac{1}{C_2} \int d^{N-1} \mathbf{p}' \int d^{N-1} \mathbf{r}' \int dP \int dp_\xi \\ \times \left(\frac{C_2}{P} \right)^{d(N-1)+1} \delta \left(\mathcal{H}(\mathbf{r}', \mathbf{p}', P) + \frac{p_\xi^2}{2Q} + Lk_B T \ln(C_2/P) - C_1 \right).$$

The remaining δ -function fixes p_ξ :

$$p_\xi = \{2Q [C_1 - \mathcal{H}(\mathbf{r}', \mathbf{p}', P) - Lk_B T \ln(C_2/P)]\}^{-\frac{1}{2}}.$$

Integration over p_ξ then results in the following expression for Ω :

$$\Omega_T(N, V, C_1, C_2) = \frac{\sqrt{2Q}}{C_2} \int d^{N-1} \mathbf{p}' \int d^{N-1} \mathbf{r}' \\ \times \int dP \left(\frac{C_2}{P} \right)^{d(N-1)+1} [C_1 - \mathcal{H}(\mathbf{r}', \mathbf{p}', P) - Lk_B T \ln(C_2/P)]^{-\frac{1}{2}}.$$

Note that this is *not* the partition function for a canonical ensemble. This problem was first pointed out by Cho et al. [739]. Only in the case that $C_2 = 0$ can the conventional Nosé-Hoover scheme generate a canonical distribution. If $C_2 = 0$ the partition function reads

$$\Omega_T(N, V, C_1, 0) = \int d^{N-1} \mathbf{p}' \int d^{N-1} \mathbf{r}' \int dP \int dp_\xi \int d\xi \\ \times \exp \{[d(N-1)+1]\xi\} \delta \left(\mathcal{H}(\mathbf{r}', \mathbf{p}', P) + \frac{p_\xi^2}{2Q} + Lk_B T \xi - C_1 \right) \\ \times \delta(e^\xi P - 0).$$

The δ -function imposes that $P = 0$. Integration over P then yields

$$\Omega_T(N, V, C_1, 0) = \int d^{N-1} \mathbf{p}' \int d^{N-1} \mathbf{r}' \int dp_\xi \int d\xi \\ \times \exp \{[d(N-1)+1]\xi\} \exp(-\xi)$$

$$\times \delta \left(\mathcal{H}(\mathbf{r}', \mathbf{p}') + \frac{p_\xi^2}{2Q} + Lk_B T \xi - C_1 \right).$$

The other δ -function fixes ξ

$$\xi = -\frac{\beta}{L} \left[\mathcal{H}(\mathbf{r}', \mathbf{p}') + \frac{p_\xi^2}{2Q} - C_1 \right],$$

and we finally obtain

$$\begin{aligned} \Omega_T(N, V, C_1, 0) &= \frac{\exp [\beta d(N-1)C_1/L]}{Lk_B T} \\ &\quad \times \int dp_\xi \exp \left[-\beta d(N-1)p_\xi^2/(2QL) \right] \\ &\quad \times \int d^{N-1}\mathbf{p}' \int d^{N-1}\mathbf{r}' \exp \left[-\beta \frac{d(N-1)}{L} \mathcal{H}(\mathbf{r}', \mathbf{p}') \right] \\ &\propto \int d^{N-1}\mathbf{p}' \int d^{N-1}\mathbf{r}' \exp \left[-\beta \frac{d(N-1)}{L} \mathcal{H}(\mathbf{r}', \mathbf{p}') \right]. \end{aligned}$$

Clearly, if we choose $L = d(N-1)$, then the correct canonical partition function is recovered. In practice, most conventional Nosé-Hoover simulations are performed with a fixed center of mass and therefore obey the condition $P = 0$.

6.6.1.3 The NPT ensemble

For the N, P, T ensemble the equations of motion are (see also section 7.2)

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \frac{p_\epsilon}{W} \mathbf{r}_i \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \left(1 + \frac{1}{N} \right) \frac{p_\epsilon}{W} \mathbf{p}_i - \frac{p_{\xi_1}}{Q_1} \mathbf{p}_i \\ \dot{V} &= dV \frac{p_\epsilon}{W} \\ \dot{P}_{\text{int}} &= dV (P_{\text{int}} - P_{\text{ext}}) + \frac{1}{N} \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} - \frac{p_{\xi_1}}{Q_1} p_\epsilon \\ \dot{\xi}_k &= \frac{p_{\xi_k}}{Q_k} \quad \text{for } k = 1, \dots, M \\ \dot{p}_{\xi_1} &= \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} + \frac{p_\epsilon^2}{W} - (dN+1)k_B T - \frac{p_{\xi_2}}{Q_2} p_{\xi_1} \\ \dot{p}_{\xi_k} &= \frac{p_{\xi_{k-1}}^2}{Q_{k-1}} - k_B T - \frac{p_{\xi_{k+1}}}{Q_{k+1}} p_{\xi_k} \quad \text{for } k = 2, \dots, M-1 \end{aligned}$$

$$\dot{p}_{\xi_M} = \frac{p_{\xi_{M-1}}^2}{Q_{M-1}} - k_B T.$$

To analyze the dynamics of this system, we have to consider two cases. In the case in which the sum of the forces is zero, $\sum_i \mathbf{F}_i = \mathbf{0}$, implies that we have additional conservation laws. The second case, $\sum_i \mathbf{F}_i \neq \mathbf{0}$, has only one conserved quantity; conservation of “energy”:

$$H' - C_1 = \mathcal{H}(\mathbf{p}, \mathbf{r}) + \frac{p_\epsilon^2}{W} + \sum_{k=1}^M \frac{p_{\xi_k}^2}{Q_k} + (dN + 1)k_B T \xi_1 + k_B T \xi_c + P_{\text{ext}} V - C_1 \\ = 0,$$

where ξ_c is defined as the center of the thermostat

$$\xi_c = \sum_{k=2}^M \xi_k.$$

We first consider the case $\sum_i \mathbf{F}_i \neq \mathbf{0}$. To analyze its dynamics we have to compute the compressibility. The independent variables are³ $\Gamma = \mathbf{p}^N, \mathbf{r}^N, \xi_1, \xi_c, p_{\xi_1}, p_{\xi_c}, V, p_\epsilon$:

$$\kappa = \nabla \cdot \Gamma = -(dN + 1) \frac{p_{\xi_1}}{Q_1} - \frac{p_{\xi_c}}{Q_c},$$

which gives a phase-space metric

$$\sqrt{g} = \exp[(dN + 1)\xi_1 + \xi_c].$$

We can now write for the partition function

$$\Omega_{T, P_{\text{ext}}}(N, C_1) = \int dV \int d^N \mathbf{p} \int d^N \mathbf{r} \int dp_{\xi_1} \int d\xi_1 \\ \times \int dp_{\xi_c} \int d\xi_c \int dp_\epsilon \exp[(dN + 1)\xi_1 + \xi_c] \delta(H' - C_1).$$

The delta function gives a condition for ξ_1

$$\xi_1 = \frac{1}{(dN + 1)k_B T} \left(C_1 - \mathcal{H}(\mathbf{p}, \mathbf{r}) - \frac{p_\epsilon^2}{W} - \sum_{k=1}^M \frac{p_{\xi_k}^2}{Q_k} - k_B T \xi_c - P_{\text{ext}} V \right).$$

Substitution of this expression into the partition function gives

$$\Omega_{T, P_{\text{ext}}}(N, C_1) = \frac{\exp(\beta C_1)}{(dN + 1)k_B T} \int dV \int d^N \mathbf{p} \int d^N \mathbf{r} \int dp_{\xi_1} \int d\xi_1$$

³ A Nosé-Hoover chain of length M has two independent variables, we use ξ_1 and ξ_c .

$$\begin{aligned}
& \times \int dp_{\xi_c} \int d\xi_c \int dp_\epsilon \\
& \times \exp \left[-\beta \left(\mathcal{H}(\mathbf{p}, \mathbf{r}) + \frac{p_\epsilon^2}{W} + \sum_{k=1}^M \frac{p_{\xi_k}^2}{Q_k} + P_{\text{ext}} V \right) \right] \\
& \propto \int dV \exp(-\beta P_{\text{ext}} V) \int d^N \mathbf{p} \int d^N \mathbf{r} \exp[-\beta \mathcal{H}(\mathbf{p}, \mathbf{r})].
\end{aligned}$$

The integration over ξ_c gives a constant, which can be infinite but has no physical importance. This demonstrates that the desired distribution is generated.

At this point, we would like to emphasize that the original Nosé-Hoover algorithm does not generate this distribution. The reason is that the metric for this algorithm generates an additional $1/V$ term in the partition function. With the algorithm of Martyna et al., this term is removed. This point is explained in detail in ref. [267].

For the case $\sum_i \mathbf{F}_i = \mathbf{0}$, we have as additional conservation laws for the total momentum \mathbf{P}

$$\mathbf{P} \exp[(1 + 1/N)\epsilon + \xi_1] = \mathbf{K}.$$

Similar to the N, V, T ensemble, the components of P are linearly dependent, and the center-of-mass coordinates have to be eliminated from the analysis. This results in a set of equations of motion in coordinates relative to the center of mass. The details of this proof can be found in ref. [267]. Similar to the N, V, T ensemble, if we use $\mathbf{K} = \mathbf{0}$, we generate an $(N - 1)PT$ ensemble.

L.6.2 Nosé-Hoover algorithms

As discussed in section L.6.1.1, it is advantageous to implement the Nosé thermostat using the formulation of Hoover, Eqs. (L.6.1)–(L.6.4). Since the velocity also appears on the right-hand side of Eq. (L.6.2), this scheme cannot be implemented directly into the velocity Verlet algorithm (see also section 4.3). To see this, consider a standard constant- N, V, E simulation, for which the velocity Verlet algorithm is of the form

$$\begin{aligned}
r(t + \Delta t) &= r(t) + v(t)\Delta t + f(t)\Delta t^2/(2m) \\
v(t + \Delta t) &= v(t) + \frac{f(t + \Delta t) + f(t)}{2m}\Delta t.
\end{aligned}$$

When we use this scheme for the Nosé-Hoover equations of motion, we obtain for the positions and velocities

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + [f_i(t)/m_i - \xi(t)v_i(t)]\Delta t^2/2 \quad (\text{L.6.14})$$

$$\begin{aligned}
v_i(t + \Delta t) &= v_i(t) + [f_i(t + \Delta t)/m_i - \xi(t + \Delta t)v_i(t + \Delta t) \\
&\quad + f_i(t)/m_i - \xi(t)v_i(t)]\Delta t/2.
\end{aligned} \quad (\text{L.6.15})$$

The first step of the velocity Verlet algorithm can be carried out without difficulty. In the second step, we first update the velocity, using the old “forces” to the intermediate value $v(t + \Delta t/2) \equiv v'$. And then we must use the new “forces” to update v' :

$$v_i(t + \Delta t) = v'_i + [f_i(t + \Delta t)/m_i - \xi(t + \Delta t)v_i(t + \Delta t)]\Delta t/2. \quad (\text{L.6.16})$$

In these equations $v_i(t + \Delta t)$ appears on the right- and left-hand sides; therefore, these equations cannot be integrated exactly.⁴ For this reason the Nosé-Hoover method is usually implemented using a predictor-corrector scheme or solved iteratively [623]. This has a disadvantage that the solution is no longer time reversible. Martyna et al. [126] have developed a set of explicit reversible integrators using the Liouville approach (see section 4.3.4) for this type of extended systems.

L.6.2.1 Canonical ensemble

For M chains, the Nosé-Hoover equations of motion are given by (see also section L.6.1.2)

$$\begin{aligned} \dot{\mathbf{r}}_i &= \mathbf{p}_i/m_i \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \frac{p_{\xi_1}}{Q_1}\mathbf{p}_i \\ \dot{\xi}_k &= \frac{p_{\xi_k}}{Q_k} \quad k = 1, \dots, M \\ \dot{p}_{\xi_1} &= \left(\sum_i p_i^2/m_i - Lk_B T \right) - \frac{p_{\xi_2}}{Q_2} p_{\xi_1} \\ \dot{p}_{\xi_k} &= \left[\frac{p_{\xi_{k-1}}^2}{Q_{k-1}} - k_B T \right] - \frac{p_{\xi_{k+1}}}{Q_{k+1}} p_{\xi_k} \\ \dot{p}_{\xi_M} &= \left[\frac{p_{\xi_{M-1}}^2}{Q_{M-1}} - k_B T \right]. \end{aligned}$$

The Liouville operator, $\text{i}L$, for the equations of motion is defined as (see section 4.3.4)

$$\text{i}L \equiv \dot{\boldsymbol{\eta}} \frac{\partial}{\partial \boldsymbol{\eta}}$$

with $\boldsymbol{\eta} = (\mathbf{r}^N, \mathbf{p}^N, \xi^M, p_\xi^M)$. Using the equations of motion, $\mathbf{p}_i = m_i \mathbf{v}_i$, and $p_{\xi_k} = Q_k v_{\xi_k}$, we obtain as Liouville operator for the Nosé-Hoover chains

$$\text{i}L_{\text{NHC}} = \sum_{i=1}^N \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} + \sum_{i=1}^N \left[\frac{\mathbf{F}_i(\mathbf{r}_i)}{m_i} \right] \cdot \nabla_{\mathbf{v}_i} - \sum_{i=1}^N v_{\xi_1} \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} + \sum_{k=1}^M v_{\xi_k} \frac{\partial}{\partial \xi_k}$$

⁴ For the harmonic oscillator, it is possible to find an analytic solution (see Case Study 13).

e34 Miscellaneous methods

$$+ \sum_{k=1}^{M-1} (G_k - v_{\xi_k} v_{\xi_{k+1}}) \frac{\partial}{\partial v_{\xi_k}} + G_M \frac{\partial}{\partial v_{\xi_M}}$$

with

$$\begin{aligned} G_1 &= \frac{1}{Q_1} \left(\sum_{i=1}^N m_i \mathbf{v}_i^2 - L k_B T \right) \\ G_k &= \frac{1}{Q_k} \left(Q_{k-1} v_{\xi_{k-1}}^2 - k_B T \right). \end{aligned}$$

As explained in section 4.3.4, the Liouville equation combined with the Trotter formula is a powerful technique for deriving a time-reversible algorithm for solving the equations of motion numerically. Here we will use this technique to derive such a scheme for the Nosé-Hoover thermostats. We use a simplified version; a more complete description can be found in ref. [126].

We have to make an intelligent separation of the Liouville operator. The first step is to separate the part of the Liouville operator that only involves the positions (iL_r) and the velocities (iL_v) from the parts that involve the Nosé-Hoover thermostats (iL_C):

$$iL_{\text{NHC}} = iL_r + iL_v + iL_C$$

with

$$\begin{aligned} iL_r &= \sum_{i=1}^N \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} \\ iL_v &= \sum_{i=1}^N \frac{\mathbf{F}_i(\mathbf{r}_i)}{m_i} \cdot \nabla_{\mathbf{v}_i} \\ iL_C &= \sum_{k=1}^M v_{\xi_k} \frac{\partial}{\partial \xi_k} - \sum_{i=1}^N v_{\xi_1} \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} \\ &\quad + \sum_{k=1}^{M-1} (G_k - v_{\xi_k} v_{\xi_{k+1}}) \frac{\partial}{\partial v_{\xi_k}} + G_M \frac{\partial}{\partial v_{\xi_M}}. \end{aligned}$$

There are several ways to factorize iL_{NHC} using the Trotter formula; we follow the one used by Martyna et al. [126]:

$$e^{(iL\Delta t)} = e^{(iL_C\Delta t/2)} e^{(iL_v\Delta t/2)} e^{(iL_r\Delta t)} e^{(iL_v\Delta t/2)} e^{(iL_C\Delta t/2)} + \mathcal{O}(\Delta t^3). \quad (\text{L.6.17})$$

The Nosé-Hoover chain part L_C has to be further factorized. Here, we will do this for a chain of length $M = 2$; the more general case is discussed in ref. [126].

The Nosé-Hoover part of the Liouville operator for this chain length can be separated into five terms:

$$iL_C = iL_\xi + iL_{C_v} + iL_{G1} + iL_{v\xi_1} + iL_{G2},$$

where the terms are defined as

$$\begin{aligned} iL_\xi &\equiv \sum_{k=1}^2 v_{\xi_k} \frac{\partial}{\partial \xi_k} \\ iL_{C_v} &\equiv - \sum_{i=1}^N v_{\xi_1} \mathbf{v}_i \cdot \nabla \mathbf{v}_i \\ iL_{G1} &\equiv G_1 \frac{\partial}{\partial v_{\xi_1}} \\ iL_{v\xi_1} &\equiv - (v_{\xi_1} v_{\xi_2}) \frac{\partial}{\partial v_{\xi_1}} \\ iL_{G2} &\equiv G_2 \frac{\partial}{\partial v_{\xi_2}}. \end{aligned}$$

The factorization for the Trotter equation that we use is⁵

$$\begin{aligned} e^{(iL_C \Delta t / 2)} &= e^{(iL_{G2} \Delta t / 4)} e^{(iL_{v\xi_1} \Delta t / 4 + iL_{G1} \Delta t / 4)} \\ &\quad \times e^{(iL_\xi \Delta t / 2)} e^{(iL_{C_v} \Delta t / 2)} e^{(iL_{G1} \Delta t / 4 + iL_{v\xi_1} \Delta t / 4)} e^{(iL_{G2} \Delta t / 4)} \\ &= e^{(iL_{G2} \Delta t / 4)} \left[e^{(iL_{v\xi_1} \Delta t / 8)} e^{(iL_{G1} \Delta t / 4)} e^{(iL_{v\xi_1} \Delta t / 8)} \right] \\ &\quad \times e^{(iL_\xi \Delta t / 2)} e^{(iL_{C_v} \Delta t / 2)} \\ &\quad \times \left[e^{(iL_{v\xi_1} \Delta t / 8)} e^{(iL_{G1} \Delta t / 4)} e^{(iL_{v\xi_1} \Delta t / 8)} \right] e^{(iL_{G2} \Delta t / 4)}. \quad (\text{L.6.18}) \end{aligned}$$

Our numerical algorithm is now fully defined by Eqs. (L.6.17) and (L.6.18). This seemingly complicated set of equations is actually relatively easy to implement in a simulation.

To see how the implementation works, we need to know how each operator works on our coordinates $\eta = (\mathbf{r}^N, \mathbf{v}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2})$. If we start at $t = 0$ with initial condition η , the position at time $t = \Delta t$ follows from

$$e^{iL_{\text{NHC}} \Delta t} f[\mathbf{r}^N, \mathbf{v}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2}].$$

Because of the Trotter expansion, we can apply each term in iL_{NHC} , sequentially. For example, if we let the first term of the Liouville operator, iL_{G2} , act on

⁵ The second factorization, indicated by $[\dots]$, is used to avoid a hyperbolic sine function, which has a possible singularity. See ref. [126] for details.

the initial state η ,

$$\begin{aligned} & \exp\left(\frac{\Delta t}{4}G_2\frac{\partial}{\partial v_{\xi_2}}\right)f\left[\mathbf{r}^N, \mathbf{p}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2}\right] \\ &= \sum_{n=0}^{\infty} \frac{(G_2\Delta t/4)^n}{n!} \frac{\partial^n}{\partial v_{\xi_2}^n} f\left[\mathbf{r}^N, \mathbf{p}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2}\right] \\ &= f\left[\mathbf{r}^N, \mathbf{p}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2} + G_2\Delta t/4\right]. \end{aligned}$$

This shows that the effect of iL_{G2} is to shift v_{ξ_2} without affecting the other coordinates. This gives as transformation rule for this operator:

$$e^{(iL_{G2}\Delta t/4)} : \quad v_{\xi_2} \rightarrow v_{\xi_2} + G_2\Delta t/4. \quad (\text{L.6.19})$$

The operators ($iL_{v\xi_1}$ and iL_{Cv}) are of the form $\exp(ax\partial/\partial x)$; such operators give a scaling of the x coordinate:

$$\begin{aligned} \exp\left(ax\frac{\partial}{\partial x}\right)f(x) &= \exp\left(a\frac{\partial}{\partial \ln(x)}\right)f\{\exp[\ln(x)]\} \\ &= f\{\exp[\ln(x) + a]\} = f[x \exp(a)] \end{aligned}$$

If we apply this result⁶ to $iL_{v\xi_1}$, we obtain for this operator

$$\begin{aligned} & \exp\left(-\frac{\Delta t}{8}v_{\xi_2}v_{\xi_1}\frac{\partial}{\partial v_{\xi_1}}\right)f\left[\mathbf{r}^N, \mathbf{p}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2}\right] \\ &= f\left[\mathbf{r}^N, \mathbf{p}^N, \xi_1, \exp\left(-\frac{\Delta t}{8}v_{\xi_2}\right)v_{\xi_1}, \xi_2, v_{\xi_2}\right], \end{aligned}$$

giving the transformation rule

$$e^{(iL_{v\xi_1}\Delta t/8)} : \quad v_{\xi_1} \rightarrow \exp[-v_{\xi_2}\Delta t/8]v_{\xi_1}. \quad (\text{L.6.20})$$

In a similar way, we can derive for the other terms

$$e^{(iL_{G1}\Delta t/4)} : \quad v_{\xi_1} \rightarrow v_{\xi_1} + G_1\Delta t/4 \quad (\text{L.6.21})$$

$$e^{(iL_\xi\Delta t/2)} : \quad \xi_1 \rightarrow \xi_1 - v_{\xi_1}\Delta t/2 \quad (\text{L.6.22})$$

⁶ This can be generalized, giving the identity

$$\begin{aligned} \exp\left(a\frac{\partial}{\partial g(x)}\right)f(x) &= \exp\left(a\frac{\partial}{\partial g(x)}\right)f\left\{g^{-1}[g(x)]\right\} \\ &= \exp\left(a\frac{\partial}{\partial y}\right)f\left[g^{-1}(y)\right] \\ &= f\left\{g^{-1}[y + a]\right\} = f\left\{g^{-1}[g(x) + a]\right\}. \end{aligned}$$

$$\xi_2 \rightarrow \xi_2 - v_{\xi_2} \Delta t / 2 \quad (\text{L.6.23})$$

$$e^{(iL_C v \Delta t / 2)} : \quad v_i \rightarrow \exp[-v_{\xi_1} \Delta t / 2] v_i. \quad (\text{L.6.24})$$

Finally, the transformation rules that are associated with iL_v and iL_r are similar to the velocity Verlet algorithm, i.e.,

$$e^{(iL_v \Delta t / 2)} : \quad \mathbf{v}_i \rightarrow \mathbf{v}_i + \mathbf{F}_i \Delta t / (2m) \quad (\text{L.6.25})$$

$$e^{(iL_r \Delta t)} : \quad \mathbf{r}_i \rightarrow \mathbf{r}_i + \mathbf{v}_i \Delta t. \quad (\text{L.6.26})$$

With these transformation rules (L.6.19)–(L.6.26) we can write down our numerical algorithm by subsequently applying the transformation rules according to the order defined by Eqs. (L.6.17) and (L.6.18). If we start with initial coordinate $\eta(0) = (\mathbf{r}^N, \mathbf{v}^N, \xi_1, v_{\xi_1}, \xi_2, v_{\xi_2})$, we have to apply first $e^{iL_C} \eta$. Since this operator is further factorized according to Eq. (L.6.18), the first step in our algorithm is to apply $e^{(iL_G \Delta t / 4)}$. According to transformation rule (L.6.19) applying this operator on η gives as new state

$$v_{\xi_2}(\Delta t / 4) = v_{\xi_2} + G_2 \Delta t / 4.$$

The output of this rule is the new state on which we apply the next operator in Eq. (L.6.18), $iL_{v\xi_1}$, with transformation rule (L.6.22):

$$v_{\xi_1}(\Delta t / 8) = \exp[-v_{\xi_2}(\Delta t / 4) \Delta t / 8] v_{\xi_1}.$$

The next step is to apply iL_{G1} , followed by again $iL_{v\xi_1}$, etc. In this way, we continue to apply all operators on the output of the previous step.

Applying the Nosé-Hoover part of the Liouville operator changes ξ_k , v_{ξ_k} , and \mathbf{v}_i . The other two Liouville operators change \mathbf{v}_i and \mathbf{r}_i . This makes it convenient to separate the algorithm into two parts in which the positions and velocities of the particles and the Nosé-Hoover chains are considered separately. An example of a possible implementation is shown in Algorithm 44. The updates of the velocities and positions, i.e., the inner terms in Eq. (L.6.17), is done with the velocity-Verlet algorithm (Eq. (7)).

L.6.2.2 The isothermal-isobaric ensemble

Similar to the canonical ensemble, we can derive a time-reversible integration scheme for simulation in the NPT ensemble. The equations of motions are given by expressions (L.6.27)–(L.6.34):

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} + \frac{p_\epsilon}{W} \mathbf{r}_i \quad (\text{L.6.27})$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \left(1 + \frac{d}{dN}\right) \frac{p_\epsilon}{W} \mathbf{p}_i - \frac{p_{\xi_1}}{Q_1} \mathbf{p}_i \quad (\text{L.6.28})$$

$$\dot{V} = dV p_\epsilon / W \quad (\text{L.6.29})$$

Algorithm 44 (Equations of motion: Nosé-Hoover)

```

function integrate      integrate equations of motion
                        for Nosé-Hoover thermostat
chain( $K$ )
vel_verlet( $K$ )
chain( $K$ )
end function

```

Specific Comments (for general comments, see p. 7)

1. This function solves the equations of motion for a single time step Δt using the Trotter expansion (Eqs. (L.6.17) and (L.6.18)).
2. In function **chain** we apply $e^{iL_C \Delta t / 4}$ to the current state (see Algorithm 45).
3. In function **pos_vel** we apply $e^{(iL_r + iL_p) \Delta t}$ to the current state (see Algorithm 7).
4. K is the total kinetic energy: it is modified to impose a canonical distribution on K .

$$\dot{p}_\epsilon = dV(P_{\text{int}} - P_{\text{ext}}) + \frac{1}{N} \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} - \frac{p_{\xi_1}}{Q_1} p_\epsilon \quad (\text{L.6.30})$$

$$\dot{\xi}_k = \frac{p_{\xi_k}}{Q_k} \quad \text{for } k = 1, \dots, M \quad (\text{L.6.31})$$

$$\dot{p}_{\xi_1} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} + \frac{p_\epsilon^2}{W} - (dN + 1)k_B T - \frac{p_{\xi_2}}{Q_2} p_{\xi_1} \quad (\text{L.6.32})$$

$$\dot{p}_{\xi_k} = \frac{p_{\xi_{k-1}}^2}{Q_{k-1}} - k_B T - \frac{p_{\xi_{k+1}}}{Q_{k+1}} p_{\xi_k} \quad \text{for } k = 2, \dots, M-1 \quad (\text{L.6.33})$$

$$\dot{p}_{\xi_M} = \frac{p_{\xi_{M-1}}^2}{Q_{M-1}} - k_B T. \quad (\text{L.6.34})$$

To derive a time-reversible numerical integration scheme to solve the equations of motion, we use again the Liouville approach.

A state is characterized by the variables $\eta = (\mathbf{r}^N, \mathbf{p}^N, \epsilon, p_\epsilon, \xi^M, p_\xi^M)$. The Liouville operator is defined by

$$iL_{NPT} \equiv \dot{\eta} \frac{\partial}{\partial \eta}.$$

For a chain of length $M = 2$, using $\mathbf{p}_i = m_i \mathbf{v}_i (\neq m_i \dot{\mathbf{r}}_i)$, $p_{\xi_k} = Q_k v_{\xi_k}$, $\epsilon = (\ln V)/d$, and $p_\eta = W v_\eta$, the Liouville operator for these equations of motion can be written as

$$iL_{NPT} = iL_r + iL_v + iL_{CP},$$

Algorithm 45 (Equations of motion of NH chain)

function chain(K)	apply Eq. (L.6.18) to the current position
G2=(Q1*vxi1*vxi1-temp)	
vxi2=vxi2+G2*delt4	Update v_{ξ_2} using Eq. (L.6.19)
vxi1=vxi1*exp(-vxi2*delt8)	Update v_{ξ_1} using Eq. (L.6.20)
G1=(2*K-L*temp)/Q1	
vxi1=vxi1+G1*delt4	Update v_{ξ_1} using Eq. (L.6.21)
vxi1=vxi1*exp(-vxi2*delt8)	Update v_{ξ_1} using Eq. (L.6.20)
xil=xil+vxi1*delt2	Update ξ_1 using Eq. (L.6.22)
xi2=xi2+vxi2*delt2	Update ξ_2 using Eq. (L.6.23)
s=exp(-vxi1*delt2)	Scale factor in Eq. (L.6.24)
for $1 \leq i \leq npart$ do	
v(i)=s*v(i)	rescale \mathbf{v}_i using Eq. (L.6.24)
enddo	
K=K*s*s	rescale kinetic energy
vxi1=vxi1*exp(-vxi2*delt8)	Update v_{ξ_1} using Eq. (L.6.20)
G1=(2*K-L*temp)/Q1	
vxi1=vxi1+G1*delt4	Update v_{ξ_1} using Eq. (L.6.21)
vxi1=vxi1*exp(-vxi2*delt8)	Update v_{ξ_1} using Eq. (L.6.20)
G2=(Q1*vxi1*vxi1-temp)/Q2	
vxi2=vxi2+G2*delt4	Update v_{ξ_2} using Eq. (L.6.19)
end function	

Specific Comments (for general comments, see p. 7)

1. In this function temp is the imposed temperature, $delt = \Delta t$, $delt2 = \Delta t/2$, $delt4 = \Delta t/4$, and $delt8 = \Delta t/8$.
2. K is the total kinetic energy.

in which we define the operators

$$\begin{aligned} iL_r &= \sum_{i=1}^N (\mathbf{v}_i + v_\epsilon \mathbf{r}_i) \cdot \nabla_{\mathbf{r}_i} + v_\epsilon \frac{\partial}{\partial \epsilon} \\ iL_v &= \sum_{i=1}^N \left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i} \right] \cdot \nabla_{\mathbf{v}_i} \\ iL_{CP} &= - \sum_{i=1}^N v_{\xi_1} \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} + \sum_{k=1}^M v_{\xi_k} \frac{\partial}{\partial \xi_k} + \sum_{k=1}^{M-1} (G_k - v_{\xi_k} v_{\xi_{k+1}}) \frac{\partial}{\partial v_{\xi_k}} \end{aligned}$$

e40 Miscellaneous methods

$$+ G_M \frac{\partial}{\partial v_{\xi_M}} - \left(1 + \frac{1}{N}\right) \sum_{i=1}^N v_\epsilon \mathbf{v}_i \cdot \nabla \mathbf{v}_i + (G_\epsilon - v_\epsilon v_{\xi_1}) \frac{\partial}{\partial v_\epsilon}$$

with

$$\begin{aligned} G_1 &= \frac{1}{Q} \left[\sum_{i=1}^N m_i \mathbf{v}_i^2 + W v_\epsilon^2 - (N_f + 1) k_B T \right] \\ G_k &= \frac{1}{Q_k} \left(Q_{k-1} v_{\xi_{k-1}}^2 - k_B T \right) \\ G_\epsilon &= \frac{1}{W} \left[\left(1 + \frac{1}{N}\right) \sum_{i=1}^N m_i \mathbf{v}_i^2 + \sum_{i=1}^N \mathbf{r}_i \cdot \mathbf{F}_i(\mathbf{r}) m_i - dV \frac{\partial U(\mathbf{r}, V)}{\partial V} - dP_{\text{ext}} V \right]. \end{aligned}$$

An appropriate Trotter equation for the equations of motion is [126]

$$e^{(iL_{NPT}\Delta t)} = e^{(iL_{CP}\Delta t/2)} e^{(iL_v\Delta t/2)} e^{(iL_r\Delta t)} e^{(iL_v\Delta t/2)} e^{(iL_{CP}\Delta t/2)} + \mathcal{O}(\Delta t^3). \quad (\text{L.6.35})$$

The operator iL_{CP} has to be further factorized:

$$iL_{CP} = iL_\xi + iL_{C_v} + iL_{G_\epsilon} + iL_{v_\epsilon} + iL_{G1} + iL_{v_{\xi_1}} + iL_{G2},$$

where the terms are defined as

$$\begin{aligned} iL_\xi &\equiv \sum_{k=1}^2 v_{\xi_k} \frac{\partial}{\partial \xi_k} \\ iL_{C_v} &\equiv - \sum_{i=1}^N \left[v_{\xi_1} + \left(1 + \frac{d}{dN}\right) \right] \mathbf{v}_i \cdot \nabla \mathbf{v}_i \\ iL_{G_\epsilon} &\equiv G_\epsilon \frac{\partial}{\partial v_\epsilon} \\ iL_{v_\epsilon} &\equiv -(v_{\xi_1} v_\epsilon) \frac{\partial}{\partial v_\epsilon} \\ iL_{G1} &\equiv G_1 \frac{\partial}{\partial v_{\xi_1}} \\ iL_{v_{\xi_1}} &\equiv -(v_{\xi_1} v_{\xi_2}) \frac{\partial}{\partial v_{\xi_1}} \\ iL_{G2} &\equiv G_2 \frac{\partial}{\partial v_{\xi_2}}. \end{aligned}$$

The Trotter expansion of the term iL_C is

$$e^{(iL_{CP}\Delta t/2)} = e^{(iL_{G2}\Delta t/4 + iL_{v_{\xi_1}}\Delta t/4)} e^{(iL_{G1}\Delta t/4)} e^{(iL_{G_\epsilon}\Delta t/4 + iL_{v_\epsilon}\Delta t/4)}$$

$$\begin{aligned}
& \times e^{(iL_\xi \Delta t / 2)} e^{(iL_{Cv} \Delta t / 2)} \\
& \times e^{(iL_{v\epsilon} \Delta t / 4 + iL_{G\epsilon} \Delta t / 4)} e^{(iL_{G1} \Delta t / 4 + iL_{v\xi_1} \Delta t / 4)} e^{(iL_{G2} \Delta t / 4)} \\
& = e^{(iL_{G2} \Delta t / 4)} \left[e^{(iL_{v\xi_1} \Delta t / 8)} e^{(iL_{G1} \Delta t / 4)} e^{(iL_{v\xi_1} \Delta t / 8)} \right] \\
& \quad \times \left[e^{(iL_{v\epsilon} \Delta t / 8)} e^{(iL_{G\epsilon} \Delta t / 4)} e^{(iL_{v\epsilon} \Delta t / 8)} \right] \\
& \quad \times e^{(iL_\xi \Delta t / 2)} e^{(iL_{Cv} \Delta t / 2)} \left[e^{(iL_{v\epsilon} \Delta t / 8)} e^{(iL_{G\epsilon} \Delta t / 4)} e^{(iL_{v\epsilon} \Delta t / 8)} \right] \\
& \quad \times \left[e^{(iL_{v\xi_1} \Delta t / 8)} e^{(iL_{G1} \Delta t / 4)} e^{(iL_{v\xi_1} \Delta t / 8)} \right] e^{(iL_{G2} \Delta t / 4)}. \quad (\text{L.6.36})
\end{aligned}$$

Similar to the *NVT* version, the transformation rules of the various operators can be derived and translated into an algorithm. Such an algorithm is presented in ref. [126].

L.7 Ewald summation in a slab geometry

In Chapter 11, we discussed the treatment of long-range interactions in three-dimensional systems. For some applications, one is interested in a system that is finite in one dimension and infinite in the other two dimensions. Examples of such systems are fluids adsorbed in slit-like pores or monolayers of surfactants.

Special techniques are required to compute long-range interactions in such inhomogeneous systems. The most straightforward solution would be to use the same approach as for the three-dimensional Ewald summation, but restrict the reciprocal-space sum to vectors in the x , y directions [740,741]. The energy we wish to calculate is

$$U_{\text{Coul}} = \frac{1}{2} \sum_{i,j=1}^N \sum'_{\mathbf{n}} \frac{q_i q_j}{|\mathbf{r}_{ij} + \mathbf{n}|},$$

where the summation over $\mathbf{n} = (L_x n_x, L_y n_y, 0)$ indicates that periodicity is only imposed in the x , y directions. As in the ordinary Ewald summation, the prime indicates that for cell $(0, 0, 0)$ the terms $i = j$ should be omitted. We have a two-dimensional periodicity in the x , y directions for which we can use the Fourier representation. The resulting expression for the energy is [742]

$$\begin{aligned}
U_{\text{Coul}} &= \frac{1}{2} \sum_{i,j=1}^N q_i q_j \left[\sum'_{\mathbf{n}} \frac{\text{erfc}(\alpha |\mathbf{r}_{ij} + \mathbf{n}|)}{|\mathbf{r}_{ij} + \mathbf{n}|} + \frac{\pi}{L^2} \sum_{h>0} \cos(\mathbf{h} \cdot \mathbf{r}_{ij}) F(h, z_{ij}, \alpha) \right. \\
&\quad \left. - g(z_{ij}, \alpha) \right] - \frac{\alpha}{\sqrt{\pi}} \sum_{i=1}^M q_i^2,
\end{aligned} \quad (\text{L.7.1})$$

where $\mathbf{h} \equiv (2\pi m_x/L_x, 2\pi m_y/L_y, 0)$ denotes a reciprocal lattice vector, z_{ij} is the distance between two particles in the z direction, and α is the screening

parameter. The function $F(h, z_{ij}, \alpha)$

$$F(h, z, \alpha) = \frac{\exp(hz)\operatorname{erfc}[\alpha z + h/(2\alpha)] + \exp(-hz)\operatorname{erfc}[-\alpha z + h/(2\alpha)]}{2h} \quad (\text{L.7.2})$$

corrects for the inhomogeneity in the non-periodic direction. If the system is truly two-dimensional, this term takes a simpler form. The function $g(z, \alpha)$

$$g(z, \alpha) = z \operatorname{erf}(\alpha z) + \exp[-(z\alpha)^2]/(\alpha\sqrt{\pi}) \quad (\text{L.7.3})$$

is an additional self-term of charge interactions in the central cell that must be subtracted from the reciprocal-space sum. In a neutral system with all particles in the plane $z = 0$ this term disappears. The last term in Eq. (L.7.1) is the same self-term that appears in the normal Ewald summation (11.2.21). The details of the derivation can be found in refs. [740–744].

From a computational point of view, Eq. (L.7.1) is inconvenient. Unlike the three-dimensional case, the double sum over the particles in the Fourier part of Eq. (L.7.1) can, in general, not be expressed in terms of the square of a single sum. This makes the calculation much more expensive than its three-dimensional counterpart. Several methods have been developed to increase the efficiency of the evaluation of the Ewald sum for slab geometries. Spohr [745] showed that the calculation can be made more efficient by the use of a look-up table combined with an interpolation scheme and the long-distance limit given by Eq. (L.7.6).

Hautman and Klein [746] considered the case in which the deviation of the charge distribution from a purely two-dimensional system is small. For such a system one can introduce a Taylor expansion in z , to separate the in-plane contributions x, y in $1/\sqrt{x^2 + y^2 + z^2}$ from the out-of-plane contributions. Using this approach, Hautman and Klein derived an expression in which the Fourier contribution can again be expressed in terms of sums over single particles. However, unless the ratio $z/\sqrt{x^2 + y^2} \ll 1$, the Taylor expansion converges very poorly. Therefore the applicability of this method is limited to systems in which all charges are close to a single plane. An example of such a system would be a self-assembled monolayer in which only the head groups carry a charge [746].

An obvious idea would be to use the three-dimensional Ewald summation by placing a slab of vacuum in between the periodic images (see Fig. L.8). Spohr has shown [745], however, that even with a slab that is four times the distance between the charges one does not obtain the correct limiting behavior (see Example 25). The reason is that a periodically repeated slab behaves like a stack of parallel plate capacitors. If the slab has a net dipole moment, then there will be spurious electric fields between the periodic images of the slab. More importantly, the usual assumption that the system is embedded in a conducting sphere does not correctly account for the depolarizing field that prevails in a system with a (periodic) slab geometry. Yeh and Berkowitz [747] have shown

that one can add a correction term to obtain the correct limiting behavior in the limit of an infinitely thin slab. In the limit of an infinitely thin slab in the z direction, the force on a charge q_i due to the depolarizing field is given by [748]

$$F_z = -\frac{4\pi q_i}{V} M_z, \quad (\text{L.7.4})$$

and the total electrostatic energy due to this field is

$$U_c = -\frac{2\pi}{V} M_z^2, \quad (\text{L.7.5})$$

where M_z is the net dipole moment of the simulation cell in the z direction

$$M_z = \sum_{i=1}^N q_i z_i.$$

If the slab is not infinitely thin compared to the box dimensions, higher-order correction terms have to be added. However, Yeh and Berkowitz [747] have shown that the lowest-order correction is sufficient if the spacing between the periodically repeated slabs is three to five times larger than the thickness of the slab (see also Crozier et al. [749]).

Illustration 25 (Ewald in slab). To illustrate the difficulties that arise when using the Ewald approach for computing long-range forces in a slab geometry, Spohr and co-workers [745,749] considered a simple example of two point charges: a charge $+q$ at $(0, 0, z)$ and a charge $-q$ at $(0, 0, 0)$. The system is finite in the z direction and periodic in the x , y directions with box sizes L_x and L_y (see Fig. L.7). Because of the periodic boundary conditions, the system forms two “sheets” of opposite charge.

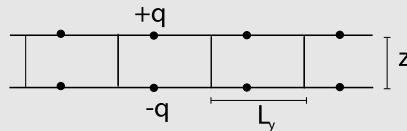


FIGURE L.7 A system containing two point charges at positions $(0, 0, 0)$ and $(0, 0, z)$; because of the periodic boundary conditions in the x and y directions, two oppositely charged “sheets” are formed. There are no periodic boundary conditions in the z direction.

In the limit $z \rightarrow \infty$, the distance between the periodic images of the charge is small compared to the distance between the sheets. We can therefore assume a uniform charge density $q/(L_x L_y)$ on each sheet. In this limit, the force acting between the two particles is given by

$$F_z = \frac{2\pi q^2}{L_x L_y}. \quad (\text{L.7.6})$$

It is instructive to compare the various methods to compute the long-range interactions in this geometry. The true forces are given by the two-dimensional Ewald summation, and we can compare the following methods:

- *Two-Dimensional Ewald summation* is the “exact” solution to this problem.
- *Bare Coulomb Potential*, we simply assume that the periodic images do not exist (or give a zero contribution). The resulting forces follow Coulomb law.
- *Truncated and Shifted Coulomb Potential*, in this method, it is assumed that beyond $r_c = 9$ the potential is zero. To remove the discontinuity at $r = r_c$ the potential is shifted as well (see section 3.3.2.2).
- *Three-Dimensional Ewald Summation*, in this approximation a layer of vacuum is added. The total system (vacuum plus slab) is seen as a normal periodic three-dimensional system (see Fig. L.8) for which the three-dimensional Ewald summation (see Eq. (11.2.24)) is used. To study the effect of the thickness of the slab of vacuum, two systems are considered, one with $L_z = 3L_x$ and a larger one with $L_z = 5L_x$.

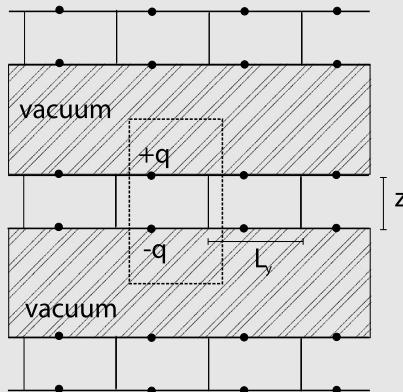


FIGURE L.8 The system of Fig. L.7 artificially made periodic in the z direction by adding a slab of vacuum.

- *3-Dimensional Ewald Summation with Correction Term*, this method is similar to the previous one; i.e., the normal three-dimensional Ewald summation is used with an additional slab of vacuum, except that now we correct for the spurious dipolar interactions.

In Fig. L.9 we compare the various approximations with the true two-dimensional solution. The bare Coulomb potential and the shifted and truncated Coulomb potential both give a zero force in the limit $z \rightarrow \infty$ and therefore do not lead to the correct limiting behavior. Although the three-dimensional Ewald summation gives a better approximation of the correct

solution, it still has the incorrect limiting behavior for both a small and a large added slab of vacuum. The corrected three-dimensional Ewald summation, however, does reproduce the correct solution, for both a slab of vacuum of $3L_x$ and that of $5L_x$.

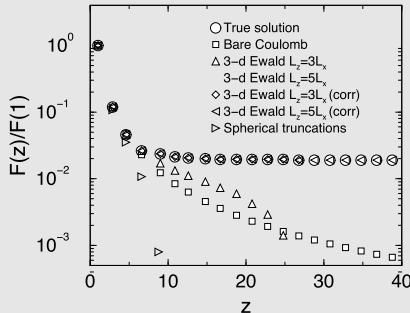


FIGURE L.9 Comparison of various methods for approximating the long-range interaction for two charges of the slab geometry shown in Fig. L.7.

L.8 Special configurational-bias Monte Carlo cases

L.8.1 Generation of branched molecules

The generation of trial configurations for branched alkanes requires some care. Naively, one might think that it is easiest to grow a branched alkane atom by atom. However, at the branchpoint we have to be careful. Suppose we have grown the backbone shown in Fig. L.10 and we now have to add the branches b_A and b_B . The total bond-bending potential has three contributions, given by

$$u_{\text{bend}} = u_{\text{bend}}(c_1, c_2, b_A) + u_{\text{bend}}(c_1, c_2, b_B) + u_{\text{bend}}(b_A, c_2, b_B).$$

Vlugt [750] pointed out that, because of the term $u_{\text{bend}}(b_A, c_2, b_B)$, it is better not to generate the positions of b_A and b_B independently. Suppose that we would try to do this anyway. We would then generate the first trial position, b_A , according to

$$P(b_A) \propto \exp[-\beta u_{\text{bend}}(c_1, c_2, b_A)],$$

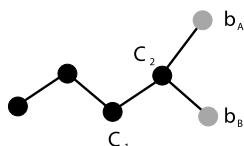


FIGURE L.10 Growth of a branched alkane.

e46 Miscellaneous methods

next, we would generate the second trial position, b_B , using

$$P(b_B|b_A) \propto \exp \{-\beta [u_{\text{bend}}(c_1, c_2, b_B) + u_{\text{bend}}(b_A, c_2, b_B)]\},$$

where $P(b_B|b_A)$ denotes the probability of generating b_B for a given position of segment b_A . However, if we would generate both positions at the same time, then the probability is given by

$$\begin{aligned} P(b_A, b_B) \\ \propto \exp \{-\beta [u_{\text{bend}}(c_1, c_2, b_A) + u_{\text{bend}}(c_1, c_2, b_B) + u_{\text{bend}}(b_A, c_2, b_B)]\}. \end{aligned}$$

The two schemes are only equivalent if

$$P(b_A, b_B) = P(b_B|b_A)P(b_A).$$

In general, this equality does not hold. To see this, compare the probability of generating configuration b_A for the two schemes. This probability is obtained by integrating over all orientations b_B . If both chains are inserted at the same time, we find that

$$\begin{aligned} P(b_A) &= \int db_B P(b_A, b_B) \\ &\propto \exp [-\beta u_{\text{bend}}(c_1, c_2, b_A)] \\ &\quad \times \int db_B \exp \{-\beta [u_{\text{bend}}(c_1, c_2, b_A) + u_{\text{bend}}(b_A, c_2, b_B)]\}. \end{aligned}$$

For the sequential scheme, we would have obtained

$$\begin{aligned} P(b_A) &= \int db_B P(b_B|b_A)P(b_A) \\ &= P(b_A) \\ &\propto \exp [-\beta u_{\text{bend}}(c_1, c_2, b_A)] \end{aligned}$$

as, in this scheme, segment b_A is inserted before segment b_B . Therefore the probability $P(b_A)$ cannot depend on b_B .

We can now easily see that if we use a model in which the two branches are equivalent, for example, isobutane, the sequential scheme does not generate equivalent *a priori* distributions for the two branches. Of course, the generation of trial segments is but one step in the [CBMC](#) scheme. Any bias introduced at this stage can be removed by incorporating the ratio of the true and the biased distributions in the acceptance criterion. However, the resulting algorithm may be inefficient. Vlugt et al. [538] have shown that simply ignoring the bias introduced by the “sequential” scheme will result in small, but noticeable, errors in the distribution of the bond angles.

The insertion of two segments at the same time is less efficient than sequential insertion, several strategies have been proposed to increase the efficiency of the simultaneous generation of branches.

For molecules in which the bond-bending potential has three contributions (as in the example above), the simplest scheme is to generate two random vectors on a sphere and use the conventional rejection scheme to generate configurations with a probability proportional to their Boltzmann weight [751]. One can also use this approach for more complex potentials that include torsion. If the random generation of trial directions becomes inefficient, it may be replaced by a simple Monte Carlo scheme [538].

For some intramolecular potential, it may even be necessary to add more than two atoms at the same time to ensure a proper *a priori* distribution of added segments. In fact, for some molecules that have multiple torsional angles, such as 2,3-dimethylbutane, this approach would imply that all atoms have to be added at the same time. To avoid such many-particle insertions, Martin and Siepmann [752] developed a scheme similar to the multiple-first-bead algorithm (see section 12.5).

The idea is to use a random insertion to generate several trial positions and to use a **CBMC** scheme to select acceptable candidates using the internal energies only. These configurations that are distributed according to the correct intramolecular Boltzmann weight will subsequently be used in another **CBMC** scheme that involves more expensive external energy calculations.

To see how this approach works, assume that we have a model with internal interactions given by u^{int} . A single segment is added using the following steps:

1. First generate a set of n_{int} random trial positions and for each position compute the internal energy, $U^{\text{int}}(i)$, and calculate the Rosenbluth factor, W associated with this *internal energy*

$$W^{\text{int}}(n) = \sum_{j=1}^{n_{\text{int}}} \exp[-\beta U^{\text{int}}(j)].$$

A possible orientation is then selected using

$$P^{\text{int}}(j) = \frac{\exp[-\beta U^{\text{int}}(j)]}{W^{\text{int}}(n)}.$$

2. Step 1 is repeated to generate k trial positions, which are then fed into the conventional **CBMC** scheme to compute the Rosenbluth factor, W , using the external potential $W^{\text{ext}}(n)$.
3. A similar method is used for the old configuration, giving $W^{\text{int}}(o)$ and $W^{\text{ext}}(o)$.
4. A move is accepted using

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{W^{\text{int}}(n) W^{\text{ext}}(n)}{W^{\text{int}}(o) W^{\text{ext}}(o)} \right).$$

Depending on the details of the potential, further refinements are possible. One can, for instance, separate the bond-bending potential and the torsion potential. This would imply three nested CBMC steps giving three different Rosenbluth factors. For more details see [752].

L.8.2 Rebridging Monte Carlo

If we model a realistic polymer or peptide we have to include bond-bending and torsional potentials. Suppose that we rotate in the interior of a polymer a randomly selected torsional angle by an amount $\Delta\phi$. If we would keep all other torsional angles of the remainder of the chain fixed, a tiny change of this torsional angle would lead to a large displacement of the last atom of the chain. If, on the other hand, one would only displace the neighboring atoms, the intramolecular interactions of the chain would increase significantly, again limiting the maximum rotation. We would like to ensure that the rotation affects only a small part of the interior of the chain and that it results in a redistribution of atoms that does not result in a large increase in the intramolecular energy. *Concerted rotation* and *rebridging* and *end-bridging* Monte Carlo are schemes that have been developed by Theodorou and co-workers [99,517,753] to perform such Monte Carlo moves.

In Fig. L.11 the rebridging problem is sketched schematically. Suppose we give the atoms 1 and 5 a new position by a random rotation of the driver angles ϕ_0 and ϕ_7 . Assume that all bond lengths and bond angles have a prescribed value, for example, their equilibrium value or any other specified value. The rebridging problem is to find all possible conformations of the trimer consisting of the atoms 2, 3, and 4 that rebridge the new positions of atoms 1 and 5 given the constraints of the prescribed bond lengths and angles. Wu and Deem [754] have shown that for the rebridging problem an analytical solution exists and that the maximum number of solutions is strictly limited to 16. Alternatively, in refs. [517,753] it is shown how to numerically locate *all* solutions of the rebridging problem.

Suppose that we have all solutions to the rebridging problem, either by the analytical solution of Wu and Deem or by the numerical scheme of Theodorou and co-workers. The next step is to use this in a Monte Carlo scheme. The scheme that we discuss here is only valid for the interior segments of a polymer. For the ends of the chains, a slightly different scheme has to be used [99]:

1. The present conformation of the polymer is denoted by σ . We generate the new configuration of the polymer, n , by selecting an atom and a direction (forward or backward) at random. This defines the atom's pair 1 and 5. These atoms are given new positions $1'$ and $5'$ by performing a random rotation around bonds $-1, 0$ and $6, 7$, respectively (see Fig. L.11).
2. Solve the rebridging problem to locate all possible conformations of the trimer that bridge the new positions of atoms $1'$ and $5'$. The total number of conformations is denoted by N_n and out of these we select one confor-

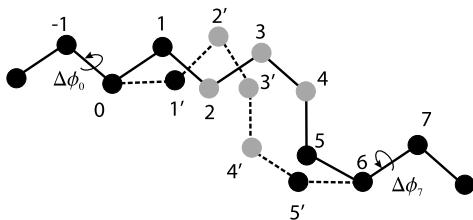


FIGURE L.11 Schematic drawing of the rebridging Monte Carlo scheme. Suppose we give the atoms 1 and 5 a new position, for example, by rotating around the -1 , 0 and 6 , 7 bonds by angles $\Delta\phi_0$ and $\Delta\phi_7$, respectively. If we would not change the positions of the trimer, consisting of the gray atoms 2 , 3 , and 4 , the intramolecular energy would increase significantly. The rebridging problem is to find a new conformation of the trimer with the same bond length and bond angle as the old conformation that bridges the new positions $1'$ and $5'$.

mation, say n , at random.⁷ If no such conformation is found the move is rejected.

3. For the old conformation, we also locate all possible conformation, i.e., we solve the rebridging problem to locate the conformations of the trimer that bridge the old positions of atoms 1 and 5 . This number of conformation is denoted by N_o .
4. In the rebridging scheme, we use a dihedral angle ϕ to generate a new configuration. This implies a temporary change of coordinate system; a Jacobian is associated with this change. In general, this Jacobian is not equal to 1. This Jacobian has to be taken into account in the acceptance rules [99]. The equations for this Jacobian can be found in refs. [99,753,754]. Here, we assume that these determinants for the old and new conformation have been calculated and are denoted by $J(o)$ and $J(n)$, respectively.
5. Of the new and old conformations the energies are calculated, $U(o)$ and $U(n)$, respectively.
6. The new conformation is accepted with a probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{\exp[-\beta U(n)] J(n)/N_n}{\exp[-\beta U(o)] J(o)/N_o} \right).$$

In refs. [99,753] the proof is given that this rebridging scheme obeys detailed balance and samples a Boltzmann distribution.

The reason it is important to find *all* solutions to the rebridging problem is to ensure detailed balance. Suppose that the determinants of the Jacobians are one and the energies are zero, then without the terms $1/N_n$ and $1/N_o$ the acceptance probability would be one for all possible new conformations. Suppose that we have a single solution for the new conformation, $N_n = 1$, and for the old conformation $N_o = 2$. Without the correction, we would violate detailed balance since

⁷ One could use the configurational-bias Monte Carlo scheme as an alternative for the random selection.

the *a priori* probability of the reverse move is only a half. Pant and Theodorou [753] have developed an alternative scheme in which one has to find only a single rebridging conformation, which is the first solution of their numerical scheme. To ensure detailed balance one has to check that the old conformation should also be the first solution to which the numerical scheme converges.

One can also use the rebridging scheme to connect atoms of *different* chains. The idea of end-bridging Monte Carlo is to alter the connectivity of the chain by bridging atoms from different chains. The simplest form is to rebridge a chain end to an interior segment of another chain with a trimer. Such an end-bridging Monte Carlo move induces a very large jump in configuration space. An important aspect of such an end-bridging move is, however, that it alters the chain lengths of the two chains. Therefore, such a move cannot be used if it is important to keep the chain length fixed. However, in most practical applications of polymers, one does not have a single chain length but a distribution of chain lengths. Pant and Theodorou [753] have shown that the resulting chain length distribution resembles a truncated Gaussian distribution.

One can envision a chain length distribution as a mixture of a very large number of components, each component characterized by its chain length l . Imposing the chain length distribution is equivalent to imposing the chemical potentials of the various components. This suggests that we could combine these end-bridging moves with the *semigrand ensemble* simulation technique (see section 6.5.4) to determine whether a change of the polymer length should be accepted.

In principle, one can use two rebridging moves for the interior segments of two chains. This would allow us to perform moves in which the total chain length remains constant. Whether in practice such a scheme will work depends on the probability that two segments of different chains with the *same* number of end segments connected to it are sufficiently close to each other.

Tests show that the rebridging method is very efficient for polymer melts with chain lengths up to C_{30} . For chains up to C_{70} rebridging Monte Carlo still samples the local structure efficiently, but fails to sample chain characteristics at larger length scales such as the end-to-end vector. End-bridging Monte Carlo effectively relaxes chains up to C_{500} [517]. Another important application of rebridging Monte Carlo is the possibility of simulating cyclic molecules. This application is illustrated by Wu and Deem in their study of cis/trans isomerization of proline-containing cyclic peptides [754,755].

L.8.3 Gibbs-ensemble simulations

In section 6.6, the Gibbs-ensemble technique was introduced as an efficient tool for simulating vapor-liquid phase equilibria. One of the Monte Carlo steps in the Gibbs-ensemble technique is the transfer of molecules between the liquid phase and gas phase. For long-chain molecules, this step, if carried out completely randomly, results in a prohibitively low acceptance of particle exchanges. Therefore, the Gibbs-ensemble technique used to be limited to systems containing

atoms or small molecules. However, by combining the Gibbs-ensemble method with configurational-bias Monte Carlo, the method can be made to work for much longer chain molecules.

Algorithm

Let us consider a continuum system with strong intramolecular interactions. In section 12.2.3 it is shown that for such a system, it is convenient to separate the potential energy into two contributions: the bonded intramolecular energy (U^{bond}) and the “external” energy (U^{ext}) that contains the intermolecular interactions and the nonbonded intramolecular interactions. As in the original implementation of the Gibbs ensemble, we attempt to exchange a molecule between the two boxes. However, while in section 6.6.1 the molecules were inserted at random, we now use the following procedure to grow a molecule atom by atom in a randomly selected box. Let us assume this is box 1 with volume V_1 . The number of particles in this box is denoted by n_1 .

1. The first atom is inserted at a random position, and the (external) energy $u_1^{\text{ext}}(n)$ is calculated together with

$$w_1^{\text{ext}}(n) = k \exp[-\beta u_1^{\text{ext}}(n)]. \quad (\text{L.8.1})$$

2. To insert the next atom i , k trial orientations are generated. The set of k trial orientations is denoted by $\{\mathbf{b}\}_k = \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_k$. These orientations are not generated at random but with a probability that is a function of the bonded part of the intramolecular energy:

$$p_i^{\text{bond}}(\mathbf{b}_n) = C \exp[-\beta u_i^{\text{bond}}(\mathbf{b}_n)]. \quad (\text{L.8.2})$$

Of each of these trial orientations the external energy is calculated $u_i^{\text{ext}}(\mathbf{b}_j)$ together with the factor

$$w_i^{\text{ext}}(n) = \sum_{j=1}^k \exp[-\beta u_i^{\text{ext}}(\mathbf{b}_j)]. \quad (\text{L.8.3})$$

Out of these k trial positions, we select one with probability

$$p_i^{\text{ext}}(\mathbf{b}_n) = \frac{\exp[-\beta u_i^{\text{ext}}(\mathbf{b}_i)]}{w_i^{\text{ext}}(n)}. \quad (\text{L.8.4})$$

3. Step 2 is repeated $\ell - 1$ times until the entire molecule is grown and the Rosenbluth factor, W of the molecule can be calculated:

$$W^{\text{ext}}(n) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(n). \quad (\text{L.8.5})$$

e52 Miscellaneous methods

For the other box, we select a molecule at random and determine its Rosenbluth factor, using the following procedure:

1. A particle is selected at random.
2. The (external) energy of the first atom is determined $u_1^{\text{ext}}(o)$ together with

$$w_1^{\text{ext}}(o) = k \exp[-\beta u_1^{\text{ext}}(o)]. \quad (\text{L.8.6})$$

3. For the next atom, $k - 1$ trial orientations are generated with a probability given by Eq. (L.8.2). These trial orientations, together with the actual position of atom i (\mathbf{b}_o), form the set $\{\mathbf{b}'\}_k$ for which we determine the factor

$$w_i^{\text{ext}}(o) = \exp[-\beta u_i^{\text{ext}}(o)] + \sum_{j=2}^k \exp[-\beta u_i^{\text{ext}}(\mathbf{b}'_j)]. \quad (\text{L.8.7})$$

4. Step 2 is repeated $\ell - 1$ times until we have retraced the entire chain and its Rosenbluth factor can be calculated:

$$W^{\text{ext}}(o) = \prod_{l=1}^{\ell} w_l^{\text{ext}}(o). \quad (\text{L.8.8})$$

We then accept this move with probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{V_1(N - n_1)}{(V - V_1)(n_1 + 1)} \frac{W^{\text{ext}}(n)}{W^{\text{ext}}(o)} \right). \quad (\text{L.8.9})$$

The proof of the validity of this algorithm, again, is very similar to those shown earlier in this chapter. We, therefore, refer the interested reader to [512, 756, 757]. The combination of the Gibbs-ensemble technique with the configurational-bias Monte Carlo method has been used to determine the vapor-liquid coexistence curve of chains of Lennard-Jones beads [512, 756] and alkanes [535, 757–759]. In Illustration 26, an application of this method is described.

Illustration 26 (Critical properties of alkanes). Alkanes are thermally unstable above approximately 650 K, which makes an experimental determination of the critical point of alkanes longer than decane (C_{10}) extremely difficult. The longer alkanes, however, are present in mixtures of practical importance for the petrochemical industry. In these mixtures, the number of components can be so large that it is not practical to determine all phase diagrams experimentally. One, therefore, has to rely on predictions made by equations of state. The parameters of these equations of state are directly related to the critical properties of the pure components. Therefore, the critical properties of the long-chain alkanes are used in the design of petrochemical processes, even if they are unstable close to the critical point [760]. Unfortunately, experimen-

tal data are scarce and contradictory, and one has to rely on semi-empirical methods to estimate the critical properties [760].

Siepmann et al. [535,757] have used the combination of the Gibbs ensemble technique and **Configurational-Bias Monte Carlo (CBMC)** to simulate vapor-liquid equilibria of the *n*-alkanes under conditions where experiments are not (yet) feasible. Phase diagrams are very sensitive to the choice of interaction potentials. Most available models for alkanes have been obtained by fitting simulation data to experimental properties of the liquid under standard conditions. In Fig. L.12, the vapor-liquid curve of octane, as predicted by some of these models, is compared with experimental data. This figure shows that the models of [761,762], which give nearly identical liquid properties, yield estimates of the critical temperature of octane that differ by 100 K. Siepmann et al. [535,757] used these vapor-liquid equilibrium data to improve the existing models.

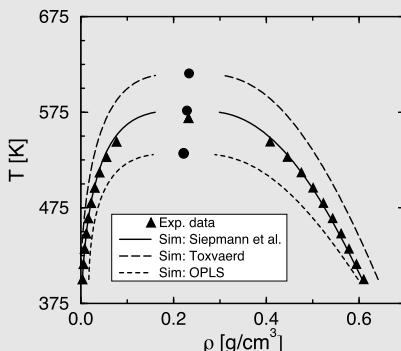


FIGURE L.12 The vapor-liquid curve of octane: comparison of Gibbs ensemble simulations using the so-called OPLS model of Jorgensen and co-workers [761], the model of Toxvaerd [762], and the model of Siepmann et al. [535,757].

In Fig. L.13, the critical temperatures and densities as predicted by the model of Siepmann et al. are plotted versus the carbon number. The simulations reproduce the experimental critical points very well. There is considerable disagreement, however, between the various experimental estimates of the critical densities. Much of our current knowledge of the critical properties of the higher alkanes is based on extrapolations of fits of the experimental data up to C₈. The most commonly used extrapolations assume that the critical density is a monotonically increasing function of the carbon number, approaching a limiting value for the very long alkanes [760,763]. In contrast to these expectations, the experimental data of Anselme et al. [764] indicate that the critical density has a maximum for C₈ and then decreases monotonically. The data of Steele (as reported in [763]), however, do not give any evidence for such a maximum (see Fig. L.13). The simulations indicate the same trend as that observed by Anselme et al. In this context, it is interesting to note that Mooij et al. [512], Sheng et al. [765], and Escobedo and de Pablo

[766] used Monte Carlo simulations to study the vapor-liquid curve of a polymeric bead-spring model for various chain lengths. These studies also show a decrease of the critical density as a function of chain length. Such a decrease of the critical density with chain length is a general feature of long-chain molecules, as was already pointed out by Flory.

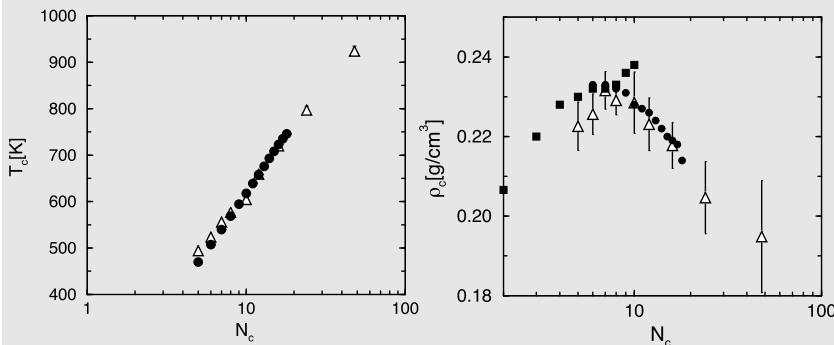


FIGURE L.13 Critical temperature T_c (left) and density ρ_c (right) as a function of carbon number N_c . The open symbols are the simulation data, and the closed symbols are experimental data.

The Gibbs ensemble technique makes it possible to compute efficiently the liquid-vapor coexistence curve of realistic models for molecular fluids. This makes it possible to optimize the parameters of the model to yield an accurate description of the entire coexistence curve rather than a single state point. It is likely, but not inevitable, that a model that describes the phase behavior correctly will also yield reasonable estimates of other properties, such as viscosity or diffusivity. Mondello and Grest have shown that this is indeed true for the diffusion coefficient of linear hydrocarbons [767,768], while Cochran, Cummings, and co-workers [769,770] found the same for the viscosity. The hydrocarbon model that was used in these studies had been optimized to reproduce experimental vapor-liquid coexistence data [535,757]. Improved force fields have since been proposed for linear alkanes [514,771,772], branched alkanes [752], alkenes [773,774], alkylbenzenes [773], and alcohols [775,776].

L.9 Recoil growth: justification of the method

The best way to arrive at the acceptance rule for the recoil growth scheme is to pretend that we actually carry out the naive brute-force calculation where we first generate the tree of *all* k^{l-1} trial conformations. We denote this tree by T_n and the *a priori* probability for generating this tree by $P_T(T_n)$. Next we test which links are “open” or “closed.” The decision whether a monomer direction is “open” or “closed” is made on the basis of the probabilities equation (12.7.1) and we denote the probability that we have a particular set O_n

of “open” monomers (and all others “closed”) by $P_O(O_n|T_n)$. Let us note the number of “open” monomers in this set by $N(O_n)$ and the number of “closed” monomers by $N(C_n)$. It is easy to see that the probability of generating this particular set is given by

$$P_O(O_n|T_n) = \prod_{j=1}^{N(O_n)} p_j^{\text{open}}(b) \prod_{k=1}^{N(C_n)} (1 - p_k^{\text{open}}(b)).$$

Finally we try to select one completely open conformation by randomly selecting, at every step, one of the “available” trial directions, i.e., a direction that is connected to (at least) one feeler that does not “die” within l_{\max} steps. At every step, there are $m_i(n)$ such directions. Hence the probability of selecting a given direction is simply $1/m_i(n)$ and the total probability that a specific conformation will be selected on the previously generated tree of all possible conformations is

$$P_S(n|O_n) = \prod_{i=1}^{l-1} \frac{1}{m_i(n)}$$

if all m_i are non-zero, and

$$P_S(n|O_n) = 0$$

otherwise. The fact that the algorithm leaves out many redundant steps (*viz.* generating the “doomed” branches or checking if there is more than one open feeler in a given direction) is irrelevant for the acceptance rule. The overall probability that we generate a trial conformation n on the set O_n , $P_S(n|O_n)$, in a tree T_n is

$$P_T(T_n) \times P_O(O_n|T_n) \times P_S(n|O_n). \quad (\text{L.9.1})$$

In order to compute the acceptance probability of a trial move, we should consider the reverse move where the *old* conformation is generated. By analogy to the forward case, this probability is given by

$$P_T(T_o) \times P_O(O_o|T_o) \times P_S(o|O_o). \quad (\text{L.9.2})$$

We wish our MC scheme to obey detailed balance. However, just as in the **CBMC** case, it is easier to impose the stronger condition of *super-detailed balance*. This implies that, in the forward move, we also should consider the probability of generating a complete tree of possible conformations around the “old” conformation and the probability that a subset of all monomers on this tree is “open.” We denote the probability of generating this tree by $P'_T(T'_o)$, where the prime indicates that this is the probability of generating all branches of the old tree, except the already existing old conformation. Clearly

$$P_T(T_o) = P'_T(T'_o) \times P_{\text{gen}}(o), \quad (\text{L.9.3})$$

where $P_{\text{gen}}(o)$ denotes the probability of generating the old conformation. As in the CBMC scheme, we can include strong intramolecular interactions in the generation of these trial monomers (see section 12.3). $P_{\text{gen}}(o)$ will then be of the form (see section 12.3)

$$P_{\text{gen}}(o) = \left(\prod_{i=1}^l p_i^{\text{bond}}(b_o) \right). \quad (\text{L.9.4})$$

Similarly, we have to consider the probability $P'_O(O'_o|T'_o)$ that a set O'_o on this tree is “open.” Again, the prime indicates that we should not include the old conformation itself. Again, it is easy to see that

$$P_O(O_o|T_o) = \left(\prod_{i=1}^l p_i^{\text{open}}(b_o) \right) P'_O(O'_o|T'_o). \quad (\text{L.9.5})$$

The *a priori* probability of generating a trial move from o to n is then given by

$$\begin{aligned} \alpha(o \rightarrow n|T_n, O_n, T_o, O_o) \\ = P_T(T_n) \times P_O(O_n|T_n) \times P_S(n|O_n) \times P'_T(T'_o) \times P'_O(O'_o|T'_o). \end{aligned} \quad (\text{L.9.6})$$

For the reverse move $n \rightarrow o$, we can derive a similar expression:

$$\begin{aligned} \alpha(n \rightarrow o|T_n, O_n, T_o, O_o) \\ = P_T(T_o) \times P_O(O_o|T_o) \times P_S(o|O_o) \times P'_T(T'_n) \times P'_O(O'_n|T'_n). \end{aligned} \quad (\text{L.9.7})$$

In these equations, we have used the notation $(o \rightarrow n|T_n, O_n, T_o, O_o)$ to indicate that we consider a transition from o to n (or vice versa) for a *given* set of “embedding” conformations. Clearly, there are many different trees and sets of open orientations that include the same conformations n and o .

Our super-detailed balance condition now becomes

$$\begin{aligned} \mathcal{N}(o) \times \alpha(o \rightarrow n|T_n, O_n, T_o, O_o) \text{acc}(o \rightarrow n|T_n, O_n, T_o, O_o) \\ = \mathcal{N}(n) \times \alpha(n \rightarrow o|T_n, O_n, T_o, O_o) \text{acc}(n \rightarrow o|T_n, O_n, T_o, O_o). \end{aligned} \quad (\text{L.9.8})$$

All terms in this equation are known, except the acceptance probabilities. We now derive an expression for the ratio $\text{acc}(o \rightarrow n|T_n, O_n, T_o, O_o)/\text{acc}(n \rightarrow o|T_n, O_n, T_o, O_o)$. To this end, we insert Eqs. (L.9.3) and (L.9.5) (and the corresponding expressions for $P'_T(T'_n)$ and $P'_O(O'_n|T'_n)$) into our super-detailed balance condition equation (L.9.8). This leads to a huge simplification as there is a complete cancellation of all probabilities for generating “open” or “closed” monomers that do *not* belong to the new (or the old) conformation. What remains is

$$\mathcal{N}(o) \times P_{\text{gen}}(n) \left(\prod_{i=1}^l \frac{p_i^{\text{open}}(b_n)}{m_i(n)} \right) \text{acc}(o \rightarrow n|T_n, O_n, T_o, O_o)$$

$$= \mathcal{N}(n) \times P_{\text{gen}}(o) \left(\prod_{i=1}^l \frac{p_i^{\text{open}}(b_o)}{m_i(o)} \right) \text{acc}(n \rightarrow o | T_n, O_n, T_o, O_o). \quad (\text{L.9.9})$$

In order to simplify the notation, we shall assume that the trial directions are uniformly distributed, i.e., see Eq. (L.9.4), $p^{\text{bond}} = \text{constant}$. From Eq. (L.9.4) it then follows that $P_{\text{gen}}(n)$ and $P_{\text{gen}}(o)$ are identical constants.

Our expression for the ratio of the acceptance probabilities then becomes

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\mathcal{N}(n) \prod_{i=1}^l p_i^{\text{open}}(o) / m_i(o)}{\mathcal{N}(o) \prod_{i=1}^l p_i^{\text{open}}(n) / m_i(n)}, \quad (\text{L.9.10})$$

where we have dropped the indices T_n, O_n, \dots . Using the definitions of $W(n)$ and $W(o)$ (Eq. (12.7.2) and below),

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \frac{\mathcal{N}(n) W(n)}{\mathcal{N}(o) W(o)}. \quad (\text{L.9.11})$$

This is precisely the acceptance rule given by Eq. (12.7.3). This concludes our “derivation” of the recoil growth scheme. The obvious question is: how well does it perform? A comparison between CBMC and the RG algorithm was made by Consta et al. [540], who studied the behavior of Lennard-Jones chains in solution. The simulations showed that for relatively short chains ($\ell = 10$) at a density of $\rho = 0.2$, the recoil growth scheme was a factor of 1.5 faster than CBMC. For higher densities $\rho = 0.4$ and longer chains $\ell = 40$ the gain could be as large as a factor 25. This illustrates the fact that the recoil scheme is still efficient, under conditions where CBMC is likely to fail. For still higher densities or still longer chains, the relative advantage of RG would be even larger. However, the bad news is that, under those conditions, *both* schemes become very inefficient.

While the recoil growth scheme is a powerful alternative to CBMC, the RG strategy is not very useful for computing chemical potentials (see [540]). More efficient schemes for computing the chemical potential are the recursive sampling scheme and the Pruning-Enriched Rosenbluth Method (PERM) (see Chapter 10).

L.10 Overlapping distribution for polymers

Let us first consider how the basic idea behind the overlapping distribution method can be applied to the Rosenbluth insertion scheme. The simplest approach would be to consider the histogram of the potential energy change on addition or removal of a chain molecule (see section 8.6.1). However, for chain molecules, this approach differs from the original Shing-Gubbins approach in that it has little, if any, diagnostic value. For instance, if we consider the chemical potential of hard-core chain molecules, the distributions of ΔU will always

overlap (namely, at $\Delta\mathcal{U} = 0$), even in the regime where the method cannot be trusted. Here, we shall describe an overlapping distribution method based on histograms of Rosenbluth weights [455]. This method will prove to be a useful diagnostic tool.

Consider again a model with internal potential energy u_{int} and external potential energy u_{ext} . In what follows, we shall compare two systems. The first, denoted by 0, contains N chain molecules ($N \geq 0$). The second system, denoted by 1, contains $N + 1$ chain molecules. In addition, both systems may contain a fixed number of other (solvent) molecules. Let us first consider system 1. Around every segment j of a particular chain molecule (say, i), we can generate $k - 1$ trial directions according to an internal probability distribution given by Eq. (10.2.19). Note that the set does *not* include the actual orientation of segment j . We denote this set of trial orientations by

$$\{\boldsymbol{\gamma}_{\text{rest}}(j)\} \equiv \prod_{j'=1}^{k-1} \{\boldsymbol{\gamma}\}_{j'},$$

where the subscript rest indicates that this set excludes the actual segment j . The probability of generating this set of trial directions is given by $P_{\text{rest}}(j)$, given by Eq. (10.2.19). Having thus constructed an umbrella of trial directions around every segment $1 \leq j \leq \ell$, we can compute the Rosenbluth weight \mathcal{W}_i of molecule i . Clearly, \mathcal{W}_i depends on all coordinates of the remaining N molecules (for convenience, we assume that we are dealing with a neat liquid), on the position \mathbf{r}_i and conformation $\boldsymbol{\Gamma}_i$ of molecule i , and on the ℓ sets of $k - 1$ trial directions:

$$\{\boldsymbol{\Gamma}_{\text{rest}}\} \equiv \prod_{j=1}^{\ell} \{\boldsymbol{\gamma}\}_{\text{rest}}(j).$$

We now define a quantity x through

$$x \equiv \ln \mathcal{W}_i(\mathbf{Q}^{N+1}, \{\boldsymbol{\Gamma}_{\text{rest}}\}),$$

where we use \mathbf{Q} to denote the translational coordinates \mathbf{r} and conformational coordinates $\boldsymbol{\Gamma}$ of a molecule. Next, consider the expression for the probability density of x , $p_1(x)$:

$$p_1(x) = \frac{\int d\mathbf{Q}^{N+1} d\{\boldsymbol{\Gamma}_{\text{rest}}\} \exp[-\beta\mathcal{U}(\mathbf{Q}^{N+1})] \prod_{j=1}^{\ell} P_{\text{rest}}(j) \delta(x - \ln \mathcal{W}_i)}{Z_{N+1}},$$

where

$$Z_{N+1} = \int d\mathbf{Q}^{N+1} d\{\boldsymbol{\Gamma}_{\text{rest}}\} \exp[-\beta\mathcal{U}(\mathbf{Q}^{N+1})] \prod_{j=1}^{\ell} P_{\text{rest}}(j)$$

$$= \int \cdots \int d\mathbf{Q}^{N+1} \exp \left[-\beta \mathcal{U}(\mathbf{Q}^{N+1}) \right].$$

The second line of this equation follows from the fact that all $P_{\text{int}}(j)$ are normalized. We shall now try to relate $p_1(x)$ to an average in system 0 (i.e., the system containing only N chain molecules). To this end, we write $\mathcal{U}(\mathbf{Q}^{N+1})$ as $\mathcal{U}(\mathbf{Q}^N) = u_{\text{ex}}(\mathbf{Q}^N, \mathbf{Q}_i) + u_{\text{int}}(\mathbf{Q}_i)$. Second, we use the fact that

$$\exp[-\beta u_{\text{int}}(i)] = Z_{\text{id}} \times \prod_{j=1}^{\ell} P_{\text{int}}(j),$$

where

$$Z_{\text{id}} \equiv \int d\Gamma_i \prod_{j=1}^{\ell} \exp[-\beta u_{\text{int}}(j)].$$

Our expression for $p_1(x)$ now becomes

$$\begin{aligned} p_1(x) &= \frac{Z_{\text{id}}}{Z_{N+1}} \int d\mathbf{Q}^N d\mathbf{r}_i d\{\Gamma_{\text{trial}}\} \exp \left[-\beta \mathcal{U}(\mathbf{Q}^N) \right] \\ &\quad \times \prod_{j=1}^{\ell} P_{\text{trial}}(j) \exp[-\beta u_{\text{ex}}(j)] \delta(x - \ln \mathcal{W}_i). \end{aligned}$$

We use the symbol $\{\Gamma_{\text{trial}}\}$ to denote the set of *all* trial segments, that is, the “umbrella” of trial directions around all segments of the chain molecule, *plus the segments themselves*. Next, every term $\exp(-\beta u_{\text{ex}}(j))$ is multiplied and divided by Z_j , defined as

$$Z_j \equiv \sum_{j'=1}^k \exp(-\beta u_{\text{ext}}(j')).$$

This allows us to write, for $p_1(x)$,

$$\begin{aligned} p_1(x) &= \frac{V Z_{\text{id}}}{Z_{N+1}} \\ &\quad \times \sum_{\text{trials}} \int d\mathbf{Q}^N d\mathbf{s}_i d\{\Gamma_{\text{trial}}\} \exp \left[-\beta \mathcal{U}(\mathbf{Q}^N) \right] P_{\text{sel}}(\mathbf{Q}_i) \mathcal{W}_i \delta(x - \ln \mathcal{W}_i), \end{aligned}$$

where we have transformed from real coordinates \mathbf{r}_i to scaled coordinates \mathbf{s}_i by factoring out V , the volume of the system. Here, $P_{\text{sel}}(\mathbf{Q}_i)$ denotes the probability of selecting the actual conformation of the molecule from the given set of trial segments according to the rule given in Eq. (10.2.20). Finally, we multiply and

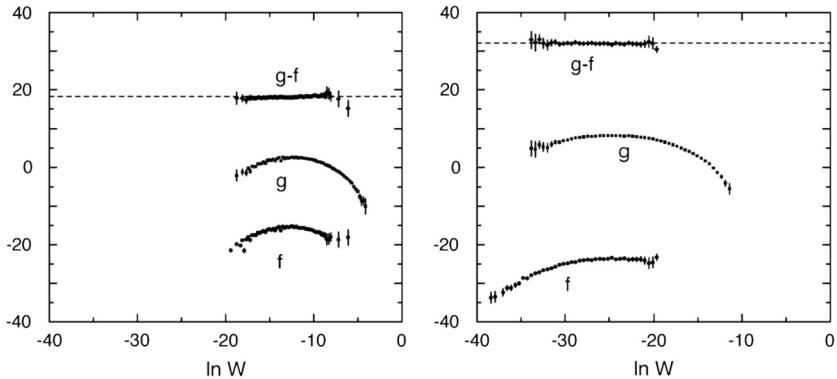


FIGURE L.14 The functions $f(\ln \mathcal{W}) \equiv p_0(\ln \mathcal{W}) + \frac{1}{2} \ln \mathcal{W}$, $g(\ln \mathcal{W}) \equiv p_1(\ln \mathcal{W}) - \frac{1}{2} \ln \mathcal{W}$ for fully flexible chains of hard sphere of length (left) $\ell = 8$ and (right) $\ell = 14$ in a hard-sphere fluid at density $\rho\sigma^3 = 0.4$. Note that the overlap between the distributions decreases as the chains become longer. The difference $g(\ln \mathcal{W}) - f(\ln \mathcal{W})$ is the overlapping distribution estimated for $\beta\mu^{\text{ex}}$. For the sake of comparison, we also show the value for $\beta\mu^{\text{ex}}$, obtained using the Rosenbluth test particle insertion method (dashed lines).

divide by Z_N and employ the fact that the δ function ensures that $\mathcal{W}_i = \exp(x)$:

$$p_1(x) = e^x \left(\frac{V Z_{\text{id}} Z_N}{Z_{N+1}} \right) \times \frac{\sum_{\text{trials}} \int d\mathbf{Q}^N d\mathbf{s}_i d\{\Gamma_{\text{trial}}\} \exp(-\beta U(\mathbf{Q}^N)) P_{\text{sel}}(\mathbf{Q}_i) \delta(x - \ln \mathcal{W}_i)}{Z_N}.$$

Finally, we obtain

$$p_1(x) = e^x \frac{V Z_{\text{id}} Z_N}{Z_{N+1}} p_0(x)$$

or

$$\ln p_1(x) = x + \beta\mu_{\text{ex}} + \ln p_0(x).$$

Hence, by constructing a histogram of $\ln w$ both in system 0 (with N chains) and in system 1 (with $N + 1$ chains), we can derive the excess chemical potential of the chain molecules by studying $\ln p_1(x) - \ln p_0(x)$. As in the original Bennett/Shing-Gubbins scheme [341–343], the method works only if there is a range of x values where we have good statistics on both $p_1(x)$ and $p_0(x)$. The advantage of this overlapping distribution scheme over the simple Rosenbluth particle insertion method is that, with the present method, sampling problems for long chains will manifest themselves as a breakdown of the overlap of p_0 and p_1 . Fig. L.14 shows an example of an application of this overlapping distribution method to hard-sphere polymers.

L.11 Hybrid Monte Carlo

In Molecular Dynamics simulations, all particle coordinates are updated simultaneously. In conventional MC simulations, only a few coordinates are changed in a trial move. As a consequence, collective molecular motions are not well represented by Monte Carlo, and this may adversely affect the rate of equilibration. The advantage of Monte Carlo is that, unlike MD, we can carry out unphysical moves. Moreover, in MC the system is not constrained to move on a hypersurface where some Hamiltonian is conserved. The time step in Molecular Dynamics is limited by the need to conserve energy. Clearly, no such constraint applies to Monte Carlo. For this reason, many authors have attempted to combine the natural dynamics of MD with the large jumps in configuration space possible in MC. The book by Allen and Tildesley [21] describes a number of such techniques (force-bias MC, Langevin Dynamics, smart Monte Carlo) that basically work by including some or all information about the intermolecular forces in the construction of a collective MC trial move.

One scheme that uses MD to generate Monte Carlo trial moves is the so-called hybrid Monte Carlo scheme [564]. At first sight, the advantage of mixing MC and MD is not obvious. However, the criteria for what constitutes a good Monte Carlo trial move are more tolerant than the specifications of a good Molecular Dynamics time step. In particular, one can take a time step that is too long for MD. Energy will not be conserved in such a trial move. However, as long as one uses an algorithm that is time reversible and area-preserving (i.e., that conserves volume in phase space), such collective moves can be used as a Monte Carlo trial move. Fortunately, a systematic way now exists to construct time-reversible, area-preserving MD algorithms, using the multiple-time-step MD scheme of Tuckerman et al. [117]. The usual Metropolis algorithm can then be used to decide on the acceptance or rejection of the move (see, e.g., [565,566]). For every trial move, the particle velocities are chosen at random from a Maxwell distribution. In fact, it is often advantageous to construct a trial move that consists of a sequence of MD steps. The reason is that, due to the randomization of the velocities, the diffusion constant of the system becomes quite low if the velocities are randomized well before the natural decay of the velocity autocorrelation function.

Yet one cannot make the time step for a single hybrid MC move too long, because then the acceptance would become very small. As a consequence, the performance of hybrid MC is not dramatically better than that of the corresponding Molecular Dynamics. Moreover, the acceptance probability of hybrid MC moves of constant length decreases with the system size, because the root-mean-square error in the energy increases with $N^{1/2}$. MD does not suffer from a similar problem. That is to say, the noise in the total energy increases with N , but the stability of the MD algorithm does not deteriorate. Hence, for very large systems, MD will always win. For more normal system sizes, hybrid MC may be advantageous.

It is also interesting to use hybrid MC on models that have an expensive (many-body) potential energy function that may, to a first approximation, be modeled using a cheap (pair) potential. We could then perform a sequence of MD steps, using the cheap potential. At the end of this collective (MD) trial move, we would accept or reject the resulting configuration by applying the Metropolis criterion with the *true* potential energy function. Many variations of this scheme exist. In any event, the hybrid Monte Carlo method is a scheme that requires fine tuning [566].

Forrest and Suter [567] have devised a hybrid MC scheme that samples polymer conformations, using fictitious dynamics of the generalized coordinates. This scheme leads to an improved sampling of the polymer conformations compared to normal MD. The interesting feature of the dynamics used in ref. [567] is that it uses a Hamiltonian that has the same potential energy function as the original polymer model. In contrast, the kinetic part of the Hamiltonian is adjusted to speed up conformational changes.

L.12 General cluster moves

In general, it is not possible to design clusters such that trial moves are always accepted. However, it is often convenient to perform clustering to enhance the acceptance of trial moves. For instance, in molecular systems with very strong short-range attractions, trial moves that pull apart two neighboring particles are very likely to be rejected. It is preferable therefore to include trial moves that attempt to displace the tightly bound particles as a single cluster. To do this, we have to specify a rule for generating clusters. Let us assume that we have such a rule that tells us that particles i and j belong to a single cluster with probability $p(i, j)$ and are disconnected with a probability $1 - p(i, j)$. Here, $p(i, j)$ depends on the state (relative distance, orientation, spin, etc.) of particles i and j . Moreover, we require that $p(i, j)$ be unchanged in a cluster move if both i and j belong to the cluster, and also if neither particle belongs to the cluster. For instance, $p(i, j)$ could depend on the current distance of i and j only. If we denote the potential energy of the old (new) configuration by \mathcal{U}_0 (\mathcal{U}_1), the detailed balance condition requires that

$$\begin{aligned} & \exp(-\beta\mathcal{U}_o) \prod_{kl} [1 - p^f(k, l)] \text{acc}(o \rightarrow n) \\ &= \exp(-\beta\mathcal{U}_n) \prod_{kl} [1 - p^r(k, l)] \text{acc}(n \rightarrow o), \end{aligned} \quad (\text{L.12.1})$$

where k denotes a particle *in* the cluster and l a particle outside it. The superscripts f and r denote forward and reverse moves. In writing Eq. (L.12.1), we have assumed that the probability of forming bonds completely within, or completely outside, the cluster is the same for forward and reverse moves. From Eq. (L.12.1), we derive an expression for the ratio of the acceptance probabili-

ties:

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} \prod_{kl} \frac{1 - p^r(k, l)}{1 - p^f(k, l)}. \quad (\text{L.12.2})$$

Clearly, many choices for p_{kl} are possible. A particularly simple form was chosen by Wu et al. [777], who assumed that $p(i, j) = 1$ for r_{ij} less than a critical distance r_c and $p(i, j) = 0$ beyond that distance (see Example 27). Note that the acceptance rule in Eq. (L.12.1) guarantees that two particles that did not belong to the same cluster in the old configuration will not end up at a distance less than r_c .

Illustration 27 (Micelle formation). Surfactants are amphiphilic molecules that consist of two chemically distinct parts. If these two parts cannot be connected, then one part will preferentially dissolve in another solvent compared to the other part. The most common case is that one part of the molecule is water soluble and the other oil soluble. But, as the two parts of the molecule are connected, the dissolution of surfactant molecules in a pure solvent (say, water) causes frustration, as the oil-soluble part of the molecule is dragged along into the water phase. Beyond a certain critical concentration, the molecules resolve this frustration by self-assembling into micelles. Micelles are (often spherical) aggregates of surfactant molecules in which the surfactants are organized in such a way that the hydrophilic heads point toward the water phase and the hydrophobic tails toward the interior of the micelle. It is of considerable interest to study the equilibrium properties of such a micellar solution. However, Molecular Dynamics simulations on model micelles have shown that the micelles move on a time scale that is long compared to the time it takes individual surfactant molecules to move [778]. Using conventional MC, rather than MD, does not improve the situation: it is relatively easy to move a single surfactant, but it takes many displacements of single surfactants to achieve an appreciable rearrangement of the micelles in the system. Yet, to sample the equilibrium properties of micellar solutions, the micelles must be able to move over distances that are long compared to their own diameter, they must be able to exchange surfactant molecules and they must even be able to merge or break up. As a consequence, standard simulations of micellar self-assembly are very slow.

Wu et al. [777] have used cluster moves to speed up the simulation of micellar solutions. They use a cluster MC scheme that makes it possible to displace entire micelles with respect to each other. The specific model for a surfactant solution that Wu et al. studied was based on a lattice model proposed by Stillinger [779]. In this model, the description of surfactants is highly simplified: the hydrophilic and hydrophobic groups are considered independent (unbonded) particles. The constraint that the head and tail of a surfactant be physically linked is translated into an electrostatic attraction between these groups. The magnitude of these effective charges depends on density and tem-

perature. A typical configuration is shown in Fig. L.15. In such a system it is natural to cluster the molecules in a micelle and subsequently, move entire micelles. The cluster criterion used by Wu et al. is

$$p(i, j) = \begin{cases} 1 & \text{if } r_{ij} < r_c \\ 0 & \text{if } r_{ij} > r_c \end{cases}$$

Wu et al. used $r_c = 1$, which implies that two particles belong to the same cluster if they are on neighboring lattice sites.

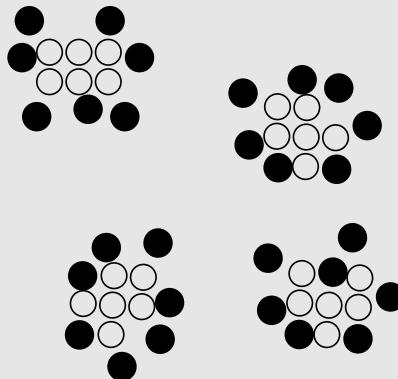


FIGURE L.15 Snapshot of a typical configuration of the surfactant model of Wu et al. [777], in which the hydrophilic and hydrophobic parts of surfactant molecules are represented by charges (black head and white tail). Under appropriate conditions, the surfactants self-assemble into micelles.

In the first step of the algorithm, the clusters are constructed using the preceding criterion. Subsequently, a cluster is selected at random and given a random displacement. It is instructive to consider the case in which we would use the ordinary acceptance rules to move the cluster; that is,

$$\text{acc}(o \rightarrow n) = \min[1, \exp(-\beta \Delta U)].$$

Fig. L.16 shows such a cluster move. The first step (top) is the construction of the cluster, followed by a displacement of one of the clusters (middle). If we accept this move with the probability given by the preceding equation, we would violate microscopic reversibility. Since we have moved the clusters in such a way that they touch each other, in the next step, these two clusters would be considered a single cluster. It then will be impossible to separate them to retrieve the initial configuration.

If we use the correct acceptance rule, Eq. (L.12.2),

$$\text{acc}(o \rightarrow n) = \min \left[1, \exp(-\beta \Delta U) \prod_{kl} \frac{1 - p^{\text{new}}(k, l)}{1 - p^{\text{old}}(k, l)} \right],$$

then this move would be rejected because $p^{\text{new}}(k, l) = 1$. Since these cluster moves do not change the configuration of the particles in a cluster and do not change the total number of particles in a cluster, it is important to combine these cluster moves with single particle moves, or use a cluster criterion $p(i, j)$ that allows the number of particles in a cluster to change.

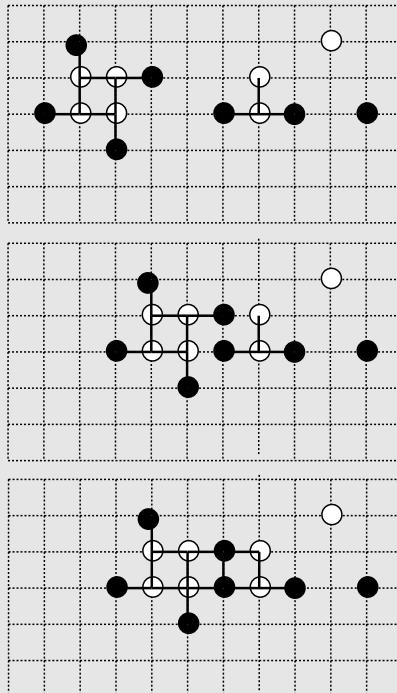


FIGURE L.16 Violation of detailed balance in a cluster move; the top figure shows the four clusters in the system; in the middle figure one of the clusters is given a random displacement, which brings this cluster into contact with another cluster; and the bottom figure shows that the new configuration has only three clusters if the moves have been accepted.

Orkoulas and Panagiotopoulos have used such cluster moves to simulate the vapor-liquid coexistence curve of the restricted primitive model of an ionic fluid [780,781].

L.13 Boltzmann-sampling with dissipative particle dynamics

The original version of DPD [257] did not generate a Boltzmann distribution. Español and Warren [683] corrected this problem. In the analysis of ref. [683], the time evolution generated by a DPD algorithm is written in the form of a Fokker-Planck equation (see e.g., [67]):

$$\partial_t \mathcal{N}(r, p; t) = \mathcal{L}_C \mathcal{N}(r, p; t) + \mathcal{L}_D \mathcal{N}(r, p; t), \quad (\text{L.13.1})$$

where \mathcal{L}_C is the usual Liouville operator of a Hamiltonian system interacting with conservative forces \mathbf{F}^C ,

$$\mathcal{L}_C \equiv - \left[\sum_i \frac{\mathbf{p}_i}{m} \frac{\partial}{\partial \mathbf{r}_i} + \sum_{i,j} \mathbf{f}_{ij}^C \frac{\partial}{\partial \mathbf{p}_i} \right], \quad (\text{L.13.2})$$

and the operator \mathcal{L}_D takes into account the effects of the dissipative and random forces:

$$\mathcal{L}_D \equiv \sum_{i,j} \hat{\mathbf{r}}_{ij} \frac{\partial}{\partial \mathbf{p}_i} \left[\gamma \omega_D(r_{ij}) (\hat{\mathbf{r}}_{ij} \cdot \mathbf{v}_{ij}) + \sigma^2 \omega_R^2(r_{ij}) \hat{\mathbf{r}}_{ij} \left(\frac{\partial}{\partial \mathbf{p}_i} - \frac{\partial}{\partial \mathbf{p}_j} \right) \right]. \quad (\text{L.13.3})$$

The derivation of these equations uses techniques developed for stochastic differential equations. The advantage of casting the DPD equations in a Fokker-Planck form is that the steady-state solution of Eq. (L.13.1) then corresponds to

$$\partial_t \mathcal{N}_{\text{eq}}(r, p; t) = 0.$$

To make the connection with statistical mechanics, the steady-state solution should correspond to the canonical distribution:

$$\begin{aligned} \mathcal{N}_{\text{eq}}(r, p; t) &= \frac{1}{Q_{NVT}} \exp[-\beta \mathcal{H}(r, p)] \\ &= \frac{1}{Q_{NVT}} \exp \left[-\beta \left(\sum_i p_i / 2m_i + V(r) \right) \right], \end{aligned}$$

where $V(r)$ is the potential that gives rise to the conservative forces, $\mathcal{H}(r, p)$ is the Hamiltonian, and Q_{NVT} is the partition function of the NVT ensemble. By construction, this Boltzmann distribution satisfies

$$\mathcal{L}_C \mathcal{N}_{\text{eq}}(r, p; t) = 0.$$

We, therefore, need to ensure that

$$\mathcal{L}_D \mathcal{N}_{\text{eq}}(r, p; t) = 0.$$

This is achieved by imposing that

$$\omega_R^2(r) = \omega_D(r) \quad \text{and} \quad \sigma^2 = 2k_B T \gamma,$$

which are Eqs. (16.1.4) and (16.1.5), i.e., the choice made by Español and Warren [683].

L.14 Reference states

L.14.1 Grand-canonical ensemble simulation

In a grand-canonical ensemble simulation, we impose the temperature and chemical potential. Experimentally, however, usually, the pressure rather than the chemical potential of the reservoir is fixed. To compare the experimental data with the simulation results it is necessary therefore to determine the pressure that corresponds to a given value of the chemical potential and temperature of our reservoir.

L.14.1.1 Preliminaries

The partition function of a system with N atoms in the N, V, T -ensemble is given by

$$Q(N, V, T) = \frac{V^N}{\Lambda^{3N} N!} \int d\mathbf{s}^N \exp[-\beta \mathcal{U}(\mathbf{s}^N)], \quad (\text{L.14.1})$$

where \mathbf{s}^N are the scaled coordinates of the N particles. The free energy is related to the partition function via

$$F = -\frac{1}{\beta} \ln Q(N, V, T),$$

which gives us for the chemical potential

$$\mu \equiv \frac{\partial F}{\partial N} = -\frac{1}{\beta} \ln [Q(N+1, V, T)/Q(N, V, T)]. \quad (\text{L.14.2})$$

For a system consisting of N molecules with each molecule having M atoms, the partition function is

$$Q(N, M, V, T) = \frac{q(T)^N V^N}{N!} \prod_{i=1}^M \int d\mathbf{s}_i^N \exp[-\beta \mathcal{U}(\mathbf{s}_i^N)], \quad (\text{L.14.3})$$

where $q(T)$ is the part of the partition function of a molecule that contains the integration over momenta (for an atom, $q(T)$ is simply Λ^{-3}) and the \mathbf{s}^N are the Cartesian coordinates of atoms in the molecule. It should be stressed that, in writing Eq. (L.14.3), we are making the assumption that there are no “hard” constraints on these intramolecular coordinates. In the presence of hard constraints, the integral in Eq. (L.14.3) would contain a Jacobian (see section 14.1).

L.14.1.2 Ideal gas

In the limit of zero density, any system will behave as an ideal gas. In this limit, only the intramolecular interactions contribute to the total potential energy

$$\mathcal{U} \approx \sum_{i=1}^N \mathcal{U}^{\text{intra}}(i).$$

For a system consisting of noninteracting atoms, the partition function (L.14.1) reduces to

$$Q_{\text{IG}}(N, V, T) = \frac{V^N}{\Lambda^{3N} N!}. \quad (\text{L.14.4})$$

We can write, for the chemical potential of such an ideal gas of atoms,

$$\mu_{\text{id.gas}} = \mu_{\text{id.gas}}^0 + k_B T \ln \rho, \quad (\text{L.14.5})$$

with the chemical potential of the reference state defined by

$$\mu_{\text{id.gas}}^0 \equiv k_B T \ln \Lambda^3. \quad (\text{L.14.6})$$

In the case of gas of noninteracting molecules, the partition function (L.14.3) reduces to

$$Q_{\text{id.gas}}(N, M, V, T) = \frac{q(T)^N V^N}{N!} \left\{ \prod_{i=1}^M \int d\mathbf{s}_i \exp[-\beta \mathcal{U}^{\text{intra}}(\mathbf{s}_i)] \right\}^N. \quad (\text{L.14.7})$$

Substitution into Eq. (L.14.2) yields, for the chemical potential,

$$\mu_{\text{id.gas}} = \mu_{\text{id.gas}}^0 + k_B T \ln \rho, \quad (\text{L.14.8})$$

where the reference chemical potential is defined as

$$\begin{aligned} \beta \mu_{\text{id.gas}}^0 &\equiv -\ln q(T) + \beta \mu_{\text{intra}}^0 \\ &= -\ln q(T) - \ln \left[\prod_{i=1}^M \int d\mathbf{s}_i \exp[-\beta \mathcal{U}^{\text{intra}}(\mathbf{s}_i)] \right]. \end{aligned} \quad (\text{L.14.9})$$

Note that $\mu_{\text{id.gas}}^0$ depends only on temperature. At any given temperature, it simply acts as a constant shift of the chemical potential that has no effect on the observable thermodynamic properties of the system.

L.14.1.3 Grand-canonical simulations

In a grand-canonical simulation, we use the following acceptance rules (see section 6.5.2). In addition, we have

$$\text{acc}(N \rightarrow N + 1) = \min \left[1, \frac{Vq(T) \exp(-\beta\mu_{\text{intra}}^0)}{(N+1)} \exp\{\beta[\mu^B - \mathcal{U}(N+1) + \mathcal{U}(N)]\} \right].$$

For the removal of a particle, we have

$$\text{acc}(N \rightarrow N - 1) = \min \left[1, \frac{N}{q(T) \exp(-\beta\mu_{\text{intra}}^0)V} \exp\{-\beta[\mu^B + \mathcal{U}(N-1) - \mathcal{U}(N)]\} \right].$$

These equations are based on the idea that particles are exchanged with a reservoir containing the same molecules at the same chemical potential, the only difference being that, in the reservoir, the molecules do not interact. In practical cases (e.g., adsorption), this means that we have a dense phase in equilibrium with a dilute vapor. And, whereas the absolute chemical potential of the vapor is of little interest, the absolute pressure is clearly an important quantity. The pressure in the reservoir is related to the chemical potential through

$$\begin{aligned} \beta\mu^B &\equiv \beta\mu_{\text{id,gas}}^0 + \ln(\rho) \\ &= \beta\mu_{\text{id,gas}}^0 + \ln(\beta P_{\text{id,gas}}). \end{aligned} \quad (\text{L.14.10})$$

Substitution of this expression in the acceptance rules yields

$$\text{acc}(N \rightarrow N + 1) = \min \left[1, \frac{V\beta P_{\text{id,gas}}}{(N+1)} \exp\{-\beta[\mathcal{U}(N+1) - \mathcal{U}(N)]\} \right] \quad (\text{L.14.11})$$

for the addition of a particle and a similar expression for particle removal. In other words, if the experimental conditions are such that the system of interest is in equilibrium with a reservoir that behaves like an ideal gas, then only the pressure of this effectively ideal gas enters into the acceptance rules for trial moves. All information about the reference state drops out (as expected).

If the pressure in the reservoir is too high for the ideal gas law to hold, we have to use an equation of state to relate the chemical potential of the reservoir to its pressure:

$$\beta\mu_B = \beta\mu_{\text{id,gas}}^0 + \ln(\beta P\phi), \quad (\text{L.14.12})$$

where ϕ is the fugacity coefficient of the fluid in the reservoir. The fugacity coefficient can be computed directly from the equation of state of the vapor in the reservoir. It is important to note that this fugacity coefficient is a function of the temperature and pressure. In summary, for a non-ideal gas, we should replace $P_{\text{id,gas}}$ in the acceptance rule (L.14.11) by $P\phi$.

Appendix M

Miscellaneous examples

M.1 Gibbs ensemble for dense liquids

Example 32 (Dense liquids). At high densities, the number of exchange steps can become very large and the simulation requires a significant amount of CPU time. This problem occurs also in conventional grand-canonical Monte Carlo simulations. Various methods, which are used to extend simulations in the grand-canonical ensemble to higher densities, can also be used in the Gibbs ensemble. An example of such a technique is the so-called excluded volume map sampling. This technique, based on the ideas of Deitrick et al. [782] and Mezei [188], has been applied to the Gibbs ensemble by Stapleton and Panagiotopoulos [783]. Before calculating the energy of the particle that has to be inserted, a map is made of the receiving subsystem, by dividing this subsystem into small boxes that can contain at most one particle. Each box carries a label that indicates whether it is empty or contains a particle. This map can then be used as a lookup table to check whether there is “space” for the particle to be inserted. If such a space is not available, the trial configuration can be rejected immediately. When using the excluded-volume map, some additional bookkeeping is needed to guarantee detailed balance (see [188] for further details).

M.2 Free energy of a nitrogen crystal

Example 33 (Free energy of a nitrogen crystal). For an orientationally disordered solid, we can use an approach similar to that used for atomic crystals. Let us consider an orientationally disordered molecular solid. We transform this solid into a state of known free energy in two stages [404,407]. First we couple the molecules in the solid with harmonic springs to their lattice sites. But in contrast to the method described earlier, we leave the original intramolecular interactions unaffected. Subsequently, we expand this “interacting Einstein crystal” to zero density (see Fig. M.1). Due to the coupling to the lattice, the crystal cannot melt on expansion but keeps its original structure. In the low-density limit, all intermolecular interactions vanish and the system behaves as an ideal Einstein crystal. This scheme of calculating an absolute free energy is referred to as the *lattice-coupling-expansion method* [404,407].

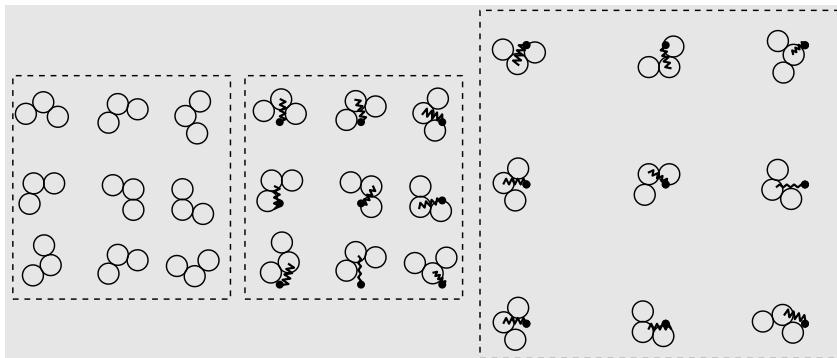


FIGURE M.1 Schematic drawing of the lattice-coupling-expansion method for calculating the free energy of a molecular solid: the first step is the coupling to an Einstein crystal (denoted by the black dots) and the second step is the expansion to zero density.

During the first stage of the thermodynamic integration, the potential energy function $\tilde{\mathcal{U}}_l$ contains both the original intermolecular potential and the harmonic coupling to the lattice:

$$\tilde{\mathcal{U}}_l(\mathbf{r}^N, \Omega^N; \lambda) = \mathcal{U}(\mathbf{r}^N, \Omega^N) + \lambda \sum_{i=1}^N \alpha(\mathbf{r}_i - \mathbf{r}_{0,i})^2, \quad (\text{M.2.1})$$

where Ω_i denotes the orientation of particle i , \mathbf{r}_i its center-of-mass position, and $\mathbf{r}_{0,i}$ the lattice site of particle i . For convenience, we have assumed that all lattice sites are equivalent. We therefore use the same value of the coupling constant α for all sites. In most molecular solids, several nonequivalent molecules may be in a unit cell. In that case different coupling constants may be chosen for all distinct lattice sites.

The change in free energy associated with switching on the harmonic springs is given by Eq. (8.4.8):

$$\Delta F_l = F(\lambda = 1) - F_{\text{mol sol}} = \int_0^1 d\lambda \left\langle \sum_{i=1}^N \alpha(\mathbf{r}_i - \mathbf{r}_{0,i})^2 \right\rangle_\lambda. \quad (\text{M.2.2})$$

It is reasonable to expect that the integrand in Eq. (M.2.2) is a smooth function of λ , as the mean-squared displacement decreases monotonically with increasing λ .

During the second stage of the thermodynamic integration, all molecules remain harmonically coupled to their (Einstein) lattice sites, but this reference lattice is expanded uniformly to zero density. In what follows, we assume for convenience that the intermolecular potential is pairwise additive:

$$\mathcal{U}(\mathbf{r}^N, \Omega^N) = \sum_{i < j}^N \mathcal{U}_{\text{pair}}(r_{ij}, \Omega_i, \Omega_j),$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between the centers of mass of the molecules i and j . When we expand the system uniformly with a factor γ , the coordinates of lattice sites are given by $\gamma \mathbf{r}_{0,i}$. When the lattice is expanded, the intermolecular interactions between the molecules change. This is best seen by expressing the center-of-mass coordinate of particle i as

$$\mathbf{r}_i = \gamma \mathbf{r}_{0,i} + \Delta \mathbf{r}_i,$$

where $\Delta \mathbf{r}_i$ is the position with respect to a lattice site of the expanded Einstein crystal. For the distance between two particles, we can write

$$r_{ij} = |\gamma(\mathbf{r}_{0,i} - \mathbf{r}_{0,j}) + \Delta \mathbf{r}_i - \Delta \mathbf{r}_j|. \quad (\text{M.2.3})$$

In terms of these coordinates, the potential energy of the expanded crystal is

$$\tilde{\mathcal{U}}_{\text{II}}(\mathbf{r}^N, \Omega^N; \gamma) = \sum_{i < j}^N \mathcal{U}_{\text{pair}}(r_{ij}, \Omega_i, \Omega_j) + \sum_{i=1}^N \alpha(\Delta \mathbf{r}_i)^2. \quad (\text{M.2.4})$$

Note that only the *intermolecular* interactions depend on γ (through r_{ij}). The derivative of the free energy with respect to γ is

$$\left\langle \frac{\partial \tilde{\mathcal{U}}}{\partial \gamma} \right\rangle = \left\langle \sum_{i < j}^N \frac{\partial \mathcal{U}_{\text{pair}}(r_{ij}, \Omega_i, \Omega_j)}{\partial \gamma} \right\rangle = \left\langle \sum_{i < j}^N \frac{\partial \mathcal{U}_{\text{pair}}(r_{ij}, \Omega_i, \Omega_j)}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial \gamma} \right\rangle.$$

Only at this stage do we make use of the assumption that the intermolecular potential is pairwise additive. We stress, however, that the assumption is not essential—it just yields a simpler form for the final expression.

From Eq. (M.2.3) it follows that

$$\frac{\partial r_{ij}}{\partial \gamma} = \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}^0}{r_{ij}}.$$

The change in free energy due to the uniform expansion is

$$\begin{aligned} \Delta F_{\text{II}} &= F_{\text{Ein}}^{\text{mol}} - F(\lambda = 1, \gamma = 1) \\ &= \int_1^\infty d\gamma \left\langle \frac{\partial \tilde{\mathcal{U}}}{\partial \gamma} \right\rangle \\ &= \int_1^\infty d\gamma \left\langle \sum_{i < j}^N \frac{\partial \mathcal{U}_{\text{pair}}(r_{ij}, \Omega_i, \Omega_j)}{\partial r_{ij}} \frac{\mathbf{r}_{ij} \cdot \mathbf{r}_{ij}^0}{r_{ij}} \right\rangle. \end{aligned} \quad (\text{M.2.5})$$

Note that if, in the preceding equation, we replace \mathbf{r}_{ij}^0 by r_{ij} , the expression in angular brackets reduces to the virial. We therefore refer to the integrand in Eq. (M.2.5) as a modified virial.

Finally, we must evaluate the free energy of the reference state: an Einstein crystal consisting of noninteracting molecules. As the intramolecular contribution to the free energy is a constant that depends only on temperature, we

e74 Miscellaneous examples

shall ignore it. The expression for the total free energy of the molecular solid then becomes

$$F_{\text{mol sol}} = F_{\text{Ein}}^{\text{mol}} - \Delta F_{\text{I}} - \Delta F_{\text{II}}. \quad (\text{M.2.6})$$

For molecular solids with (partial) orientational order, a similar scheme can be used. We transform the solid by imposing a coupling of the centers of mass of the molecules together with a coupling of the *ordered* degrees of freedom. With the combined coupling the solid is expanded. The details of this scheme depend on the nature of the orientational order.

M.3 Zeolite structure solution

Example 34 (Zeolite structure solution). Falcioni and Deem [550] used parallel tempering as a method to find zeolite structures. Zeolites are inorganic crystals with a known chemical structure. The interesting aspect of these materials is that there are about 118 different known zeolite structures. But many more zeolite structures are theoretically possible. If one makes a new zeolite, one often has a powder and it is much more time consuming, if possible at all, to make a sufficiently large crystal from which the crystal structure can be determined.

The problem that Falcioni and Deem were interested in was finding the zeolite structure that corresponds best to an experimental powder diffraction pattern. Experimentally, it is well known that in zeolites the angles between three successive T-sites (positions of the Si atoms in an all-silica zeolite) are clustered around 109.5° . Falcioni and Deem assumed that any new zeolite structure should have angles with a similar distribution. In a similar way, rules were deduced for the average bond distance and other properties of the structures. These rules were translated into a pseudo-Hamiltonian; this Hamiltonian has a high value for those crystal structures that do not resemble the characteristics of a zeolite and a low value for those crystals that have zeolite-like characteristics. An additional term was added to this structural pseudo-Hamiltonian that indicated the difference between the experimental and the calculated X-ray powder pattern of the new crystal structure. Falcioni and Deem then assumed that the unknown zeolite structure corresponds to the structure that minimizes this combined Hamiltonian. Various Monte Carlo rules were invented to change the crystal structure, and a new structure was accepted or rejected by using this pseudo-Hamiltonian. Since this pseudo-Hamiltonian has many local minima, parallel tempering was used to find the optimal structure.

To test their method, Falcioni and Deem used the information of 32 zeolite structures to generate the pseudo-Hamiltonian. This information was sufficient to solve the structure of the remainder of all known zeolite structures.

Appendix N

Supporting information for case studies

In the previous editions, we provided Fortran code to carry out the simulations that were discussed in the Case Studies. In the current edition, we provide some Jupiter-Notebooks instead.

Most of the old case studies are now presented as Examples in the new edition. For those that are interested in running the FORTRAN code, we list all the case studies from the second edition here.

N.1 Equation of state of the Lennard-Jones fluid-I

Case Study 1 (Equation of state of the Lennard-Jones fluid-I). One important application of molecular simulation is computing the phase diagram of a given model system. In several chapters, we discuss some of the techniques that have been developed to study phase transitions, including direct-coexistence simulations. However, direct-coexistence simulations may suffer from hysteresis, which may make them less suited for locating phase transitions. In the present Case Study, we illustrate some of the problems that occur when we use standard Monte Carlo simulation to determine a phase diagram. As an example, we focus on the vapor-liquid curve of the Lennard-Jones fluid. Of course, as was already mentioned in section 3.3.2.2, the phase behavior is quite sensitive to the detailed form of the intermolecular potential that is used. In the Case Study, we approximate the full Lennard-Jones potential as follows:

$$u(r) = \begin{cases} u_{\text{LJ}}(r) & r \leq r_c \\ 0 & r > r_c, \end{cases}$$

where the cutoff radius r_c is set to half the box length. The contribution of the particles beyond this cutoff is estimated with the usual tail corrections; that is, for the energy

$$u^{\text{tail}} = \frac{8}{3}\pi\rho \left[\frac{1}{3} \left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right]$$

and for the pressure

$$P^{\text{tail}} = \frac{16}{3}\pi\rho^2 \left[\frac{2}{3} \left(\frac{1}{r_c} \right)^9 - \left(\frac{1}{r_c} \right)^3 \right].$$

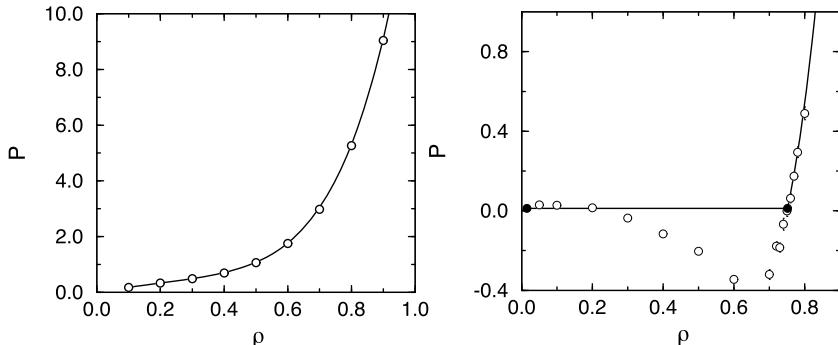


FIGURE N.1 Equation of state of the Lennard-Jones fluid. (Left) Isotherm at $T = 2.0$. (Right) Isotherm below the critical temperature ($T = 0.9$); the horizontal line is the saturated vapor pressure, and the filled circles indicate the densities of the coexisting vapor and liquid phases. The solid curve represents the equation of state of Johnson et al. [83], and the circles are the results of the simulations ($N = 500$). The horizontal line in the right-hand figure corresponds to the coexistence pressure obtained by applying the Maxwell construction to the Johnson equation of state. The Maxwell construction exploits the fact that the equality of the chemical potentials of the coexisting phases implies that $P_{\text{coex}} = \int_{V_{\text{liq}}}^{V_{\text{vap}}} dV P(V)/(V_{\text{vap}} - V_{\text{liq}})$. The statistical errors in the numerical data are smaller than the symbol sizes.

The equation of state of the Lennard-Jones fluid has been investigated by many groups using Molecular Dynamics or Monte Carlo simulations starting with the work of Wood and Parker [69]. The first systematic study of the equation of state of the Lennard-Jones fluid was reported by Verlet [14]. Subsequently, many more studies have been published. In 1979, the data available at that time were compiled by Nicolas et al. [82] into an accurate equation of state. This equation has been refitted by Johnson et al. [83] using the best data then available. In the present study, we compare our numerical results with the equation of state by Johnson et al.

We performed several simulations using Algorithms 1 and 2. During the simulations we determined the energy per particle and the pressure. The pressure was calculated using the virial, W

$$P = \frac{\rho}{\beta} + \frac{W}{V}, \quad (\text{N.1.1})$$

where the virial is defined by

$$W = \frac{1}{3} \sum_i \sum_{j>i} \mathbf{f}(\mathbf{r}_{ij}) \cdot \mathbf{r}_{ij}, \quad (\text{N.1.2})$$

where $\mathbf{f}(\mathbf{r}_{ij})$ is the intermolecular force. Fig. N.1 (left) compares the pressure as obtained from a simulation above the critical temperature with the equation of state of Johnson et al. [83]. The agreement is excellent (as is to be expected).

Fig. N.1 (right) shows a typical isotherm below the critical temperature. If we cool the system below the critical temperature, we should expect to observe

vapor-liquid coexistence. However, standard Monte Carlo or Molecular Dynamics simulations of small model systems are not suited to study the coexistence between two phases. Using the Johnson equation of state, we predict how the pressure of a macroscopic Lennard-Jones system would behave in the two-phase region (see Fig. N.1). For densities inside the coexistence region, the pressure is expected to be constant and equal to the saturated vapor pressure. If we now perform a Monte Carlo simulation of a finite system (500 LJ particles), we find that the computed pressure is not at all constant in the coexistence region (see Fig. N.1). In fact we observe that, over a wide density range, the simulated system is metastable and may even have a negative pressure. The reason is that, in a finite system, a relatively important free-energy cost is associated with the creation of a liquid-vapor interface. So much so that, for sufficiently small systems, it is favorable for the system not to phase separate at all [85]. Clearly, these problems will be most severe for small systems and in cases where the interfacial free energy is large. For this reason, standard *NVT*-simulations are not recommended to determine the vapor-liquid coexistence curve or, for that matter, any strong first-order phase transition in small systems.

To determine the liquid-vapor coexistence curve, we should determine the equation of state for a large number of state points outside the coexistence region. These data can then be fitted to an analytical equation of state. With this equation of state, we can determine the vapor-liquid curve (this is the procedure used by Nicolas et al. [82] and Johnson et al. [83]).

Of course, if we simulate a system consisting of a sufficiently large number of particles, it is possible to simulate a liquid phase in coexistence with its vapor. However, such simulations are time-consuming, because Monte Carlo moves are not very efficient at equilibrating the pressure in a two-phase system.

N.2 Importance of detailed balance

Case Study 2 (Importance of detailed balance). Monte Carlo simulations aim to sample points in configuration space according to their Boltzmann weights. If the system is ergodic, imposing detailed balance is sufficient, but not necessary to ensure Boltzmann sampling. In other words: some sampling schemes that violate detailed balance may nevertheless lead to correct Boltzmann sampling (see Section 13.4.4). However, one should avoid non-detailed-balance schemes, unless one can prove that they satisfy the more general *balance* condition. In the present Example, we show that systematic errors result when using a scheme that does not satisfy the *balance* condition.

Consider an ordinary *N, V, T* move; a new trial position is generated by giving a randomly selected particle, say i , a random displacement:

$$x_n(i) = x_o(i) + \Delta_x(\mathcal{R} - 0.5),$$

where Δ_x is twice the maximum displacement. We now make a seemingly small change in the algorithm and generate a new position using

$$x_n(i) = x_o(i) + \Delta_x(\mathcal{R} - 0.0) \quad \text{wrong!}$$

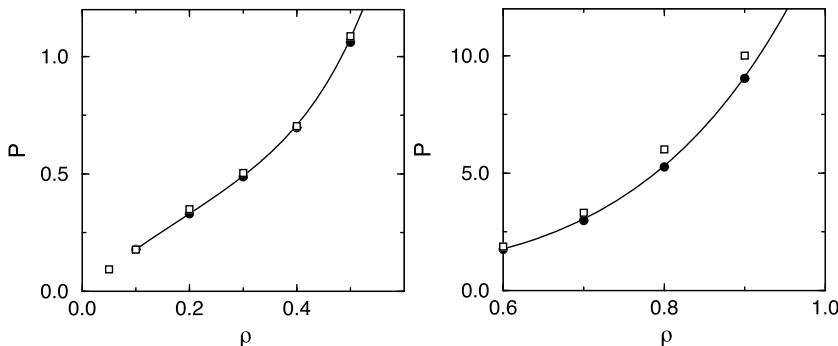


FIGURE N.2 Equation of state of the Lennard-Jones fluid ($T = 2.0$); comparison of a displacement scheme that obeys detailed balance (circles) and one that does not (squares). Both simulations have been performed with 500 particles. The solid curve is the equation of state of Johnson et al. [83]. The figure at the left corresponds to the low-pressure regime. The high-pressure regime is shown in the right-hand figure.

i.e., we give the particles only a *positive* displacement. With such a move, detailed balance is violated, since the reverse move — putting the particle back at x_o — is not possible.¹

For the Lennard-Jones fluid, we can use the program of Case Study 1 to compare the two sampling schemes. The results of these simulations are shown in Fig. N.2. At first sight, the results of the incorrect scheme look reasonable; in fact, at low densities, the results of the two schemes do not show significant differences. But at high densities, the incorrect scheme overestimates the pressure. This overestimate of the pressure is a systematic error: it does not disappear when we perform longer simulations.

The above example illustrates something important: one cannot decide if an algorithm is correct on the basis of the fact that the results *look* reasonable. Simulation results must always be tested on cases where we *know* the correct answer: either a known (and trusted) numerical result, but preferably an exact result that may be known in some limiting case (dilute vapor, dense solid, etc.).

Usually, one does not know *a priori* the size of the optimal maximum displacement in a Monte Carlo simulation. By doing short test runs, we can explore what maximum displacement achieves the best sampling for a given amount of simulation time. However, the maximum step-size should not be changed during a production run, because then one would violate detailed balance [90]: if from one move to the next, the maximum displacement is decreased, then the *a priori* probability for a particle to return to its previous position could be zero, which violates microscopic reversibility.

¹ In Event-Chain MC (section 13.4.4) we will see examples of forward-only algorithms that have been designed to satisfy balance.

N.3 Why count the old configuration again?

Case Study 3 (Why count the old configuration again?). A somewhat counter-intuitive feature of the Metropolis sampling scheme is that, if a trial move is rejected, we should once again count the contributions of the old configuration to the average that we are computing (see acceptance rule (3.2.12)). The aim of this Case Study is to show that this recounting is really essential. In the Metropolis scheme, the acceptance rule for a move from o to n is

$$\text{acc}(o \rightarrow n) = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} \quad \begin{aligned} & \mathcal{U}(n) \geq \mathcal{U}(o) \\ & = 1 \end{aligned} \quad \begin{aligned} & \mathcal{U}(n) < \mathcal{U}(o). \end{aligned}$$

These acceptance rules lead to a transition probability

$$\pi(o \rightarrow n) = \exp\{-\beta[\mathcal{U}(n) - \mathcal{U}(o)]\} \quad \begin{aligned} & \mathcal{U}(n) \geq \mathcal{U}(o) \\ & = 1 \end{aligned} \quad \begin{aligned} & \mathcal{U}(n) < \mathcal{U}(o). \end{aligned}$$

Note that this transition probability must be normalized:

$$\sum_n \pi(o \rightarrow n) = 1.$$

From this normalization, it follows that the probability that we accept the old configuration again is by definition

$$\pi(o \rightarrow o) = 1 - \sum_{n \neq o} \pi(o \rightarrow n).$$

This last equation implies that we should count the contribution of the old configuration again.

It is instructive to use the Lennard-Jones program from Case Study 1 to investigate numerically the error that is made when we only include accepted configurations in our averaging. In essence, this means that in Algorithm 2 we continue attempting to displace the selected particle until a trial move has been accepted.² In Fig. N.3 we compare the results of the correct scheme with those obtained by the scheme in which we continue to displace a particle until a move is accepted. Again the results look reasonable, but the figure shows that large, systematic errors are being made.

N.4 Static properties of the Lennard-Jones fluid

Case Study 4 (Static properties of the Lennard-Jones fluid). Let us illustrate the results of the previous sections with an example. Like in the section on Monte Carlo simulations we choose the Lennard-Jones fluid as our model system. We use

² It is easy to see that this approach leads to the wrong answer if we try to compute the average energy of a two-level system with energy levels E_0 and E_1 . If we include only accepted trial moves in our averaging, we would find that $\langle E \rangle = (E_0 + E_1)/2$, independent of temperature.

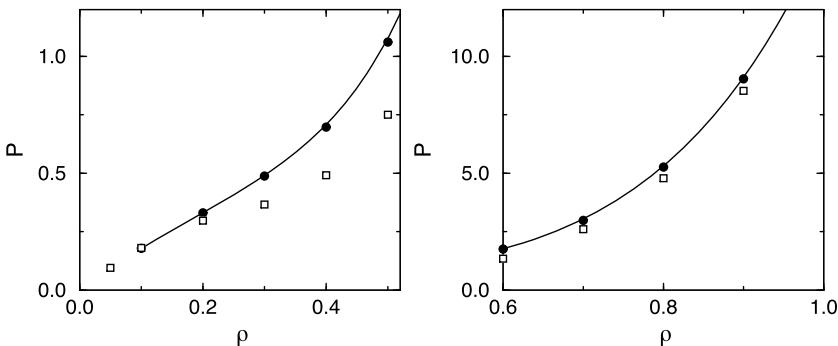


FIGURE N.3 Equation of state of the Lennard-Jones fluid ($T = 2.0$); comparison of a scheme in which particles are displaced until a move is accepted (squares) with the conventional scheme (circles). Both simulations have been performed with 108 particles. The solid curve is the equation of state of Johnson et al. [83]. The left figure is at low pressure, and the right one at high pressure.

a truncated and shifted potential (see also section 3.3.2.2):

$$u^{\text{tr-sh}}(r) = \begin{cases} u^{\text{lj}}(r) - u^{\text{lj}}(r_c) & r \leq r_c \\ 0 & r > r_c \end{cases}, \quad (\text{N.4.1})$$

where $u^{\text{lj}}(r)$ is the Lennard-Jones potential and for these simulations $r_c = 2.5\sigma$ is used.

During a simulation, we must check if the system has equilibrated or at least has reached a state that is stable on the timescale of the simulations. We then collect data about observable quantities, and at the end of the simulation, we compute averages and estimate the statistical errors. This Example illustrated how such a simulation might proceed.

At the beginning of the simulation, we prepare the system in a state that, of course, is not yet equilibrated. Here, we will assume that the particles were originally positioned on the lattice sites of a face-centered cubic crystal.

We initialize the velocities of the particles such that the initial kinetic energy corresponds to a temperature $T = 0.728$. The density is fixed at $\rho = 0.8442$, which is a typical liquid density, close to the triple (gas-liquid-solid) point of the Lennard-Jones fluid. When we start the MD simulation from this initial configuration, the potential energy will decrease, and because energy is conserved, the kinetic energy will go up —see Fig. N.4.

In Fig. N.4 shows the time evolution from the start of the simulation of the total energy, kinetic energy, and potential energy. Note that the total energy, whilst fluctuating slightly, does not drift. The kinetic and potential energies change strongly during the equilibration period, but after that, they oscillate around their equilibrium value. The figure suggests that for this (very small) system, equilibration is complete in less than 1000 time steps. However, longer equilibrations will be needed for larger systems, and for glassy systems, MD may not achieve equilibration at all.

Next, we consider error estimates. We used the method of Flyvbjerg and Petersen [148] to estimate the statistical error in the potential energy. We denote the

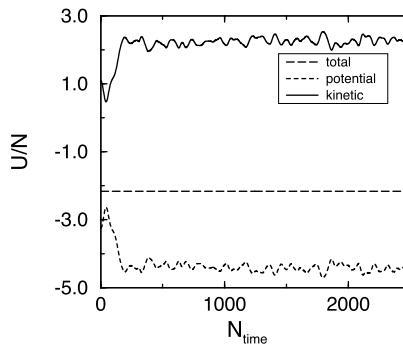


FIGURE N.4 Total, potential, and kinetic energy per particle U/N as a function of the number of time steps N_{time} .

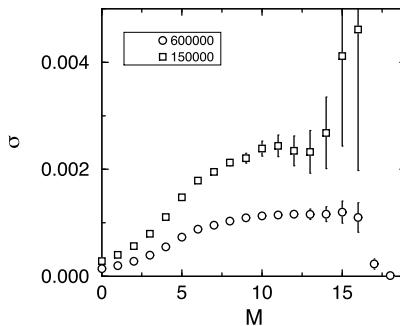


FIGURE N.5 The standard deviation σ in the potential energy as a function of the number of block operations M for a simulation of 150,000 and 600,000 time steps. This variance is calculated using Eq. (5.3.28).

number of blocking operations by M (see Fig. N.5). In this figure, the number of blocking operations by M , and from the plateau, we obtain an estimate of the standard deviation in our results.

The figure also shows the effect of increasing the total length of the simulation by a factor of 4; as expected, the statistical error in the potential energy decreases by a factor of 2.

We obtain the following results: for the potential energy $U = -4.4190 \pm 0.0012$ and for the kinetic energy $K = 2.2564 \pm 0.0012$, the latter corresponding to an average temperature of $T = 1.5043 \pm 0.0008$. For the pressure, we find 5.16 ± 0.02 .

In Fig. N.6, the radial distribution function is shown. To determine $g(r)$, we used Algorithm 8. This $g(r)$ shows the characteristics of a dense liquid. We can use the radial distribution function to calculate the energy and pressure. The potential energy per particle can be calculated from

$$\begin{aligned} U/N &= \frac{1}{2} \rho \int_0^\infty dr u(r) g(r) \\ &= 2\pi\rho \int_0^\infty dr r^2 u(r) g(r) \end{aligned} \quad (\text{N.4.2})$$

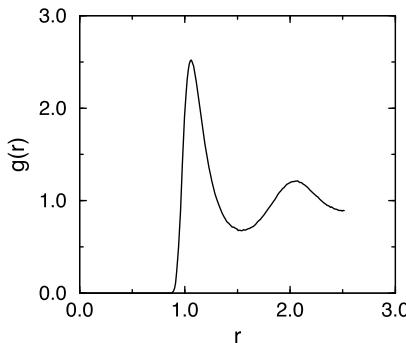


FIGURE N.6 Radial distribution function of a Lennard-Jones fluid close to the triple point: $T = 1.5043 \pm 0.0008$ and $\rho = 0.8442$.

and for the pressure from

$$\begin{aligned} P &= \rho k_B T - \frac{1}{3} \frac{1}{2} \rho^2 \int_0^\infty dr \frac{du(r)}{dr} r g(r) \\ &= \rho k_B T - \frac{2}{3} \pi \rho^2 \int_0^\infty dr \frac{du(r)}{dr} r^3 g(r), \end{aligned} \quad (\text{N.4.3})$$

where $u(r)$ is the pair potential.

Eqs. (N.4.2) and (N.4.3) can be used to check the consistency of the energy and pressure calculations and the determination of the radial distribution function. In our example, we obtained from the radial distribution function for the potential energy $U/N = -4.419$ and for the pressure $P = 5.181$, which is in good agreement with the direct calculation.

N.5 Dynamic properties of the Lennard-Jones fluid

Case Study 5 (Dynamic properties of the Lennard-Jones fluid). As an example of the numerical determination of a dynamical property, we consider the calculation of the self-diffusion coefficient. As shown in the previous section, the diffusion coefficient can be determined either from the mean-squared displacement, or from the velocity auto-correlation function (VACF). Here, we follow the approach of Algorithm 9 to compute these observables.

In Fig. N.7 the mean-squared displacement is shown as a function of the simulation time. From the mean-squared displacement, we can determine the diffusion using Eq. (5.2.13). This equation is however only valid in the limit $t \rightarrow \infty$. In practice, this means that we have to verify that we compute the mean-squared displacement in the long-time regime where all transient effects have died down, and only the linear dependence on t remains.

The Green-Kubo (GK) integral of the Velocity autocorrelation function (VACF) provides an alternative route to calculate the diffusion coefficient (Eq. (5.2.16)). Strictly speaking, the upper limit of the Green-Kubo integral corresponds to $t \rightarrow \infty$. In practice, we must stop at an earlier time t_c , because once the VACF has decayed into the noise, further integration increases the noise but not the signal. A rough

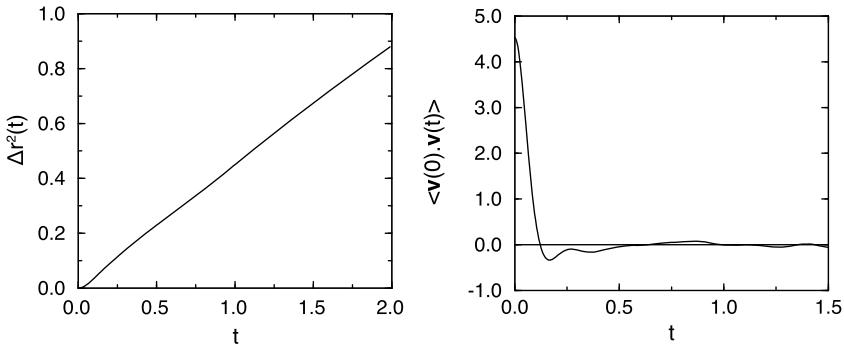


FIGURE N.7 (Left) Mean-squared displacement $\Delta r(t)^2$ as a function of the simulation time t . Note that for long times, $\Delta r(t)^2$ varies linearly with t . The slope is then given by $2dD$, where d is the dimensionality of the system and D is the self-diffusion coefficient. (Right) Velocity auto-correlation function $\langle \mathbf{v}(0) \cdot \mathbf{v}(t) \rangle$ as a function of simulation time t .

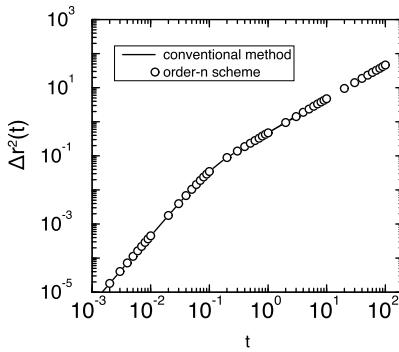


FIGURE N.8 Mean-squared displacement as a function of time for the Lennard-Jones fluid ($\rho = 0.844$, $N = 108$, and $T = 1.50$); comparison of the conventional method with the order- n scheme.

estimate of t_c can be obtained by determining the time when the integral seems to have reached a plateau value. Note that beyond t_c , we expect all transients in the mean-squared displacement to have died down.

N.6 Algorithms to calculate the mean-squared displacement

Case Study 6 (Algorithms to calculate the mean-squared displacement). In this case study, we compare the conventional (Algorithm 9), and the order- n methods (Algorithm 10) to determine the time-dependence of mean-squared displacement. The example considers a Lennard-Jones fluid.

In Fig. N.8 we compare the time-dependence of the mean-squared displacement computed with the conventional method and the one obtained with the order- n scheme. For the order- n scheme, the memory requirement grows logarithmically with time, but for the conventional scheme, it grows linearly. For the present simulation settings, the calculation using the conventional scheme could

not be extended beyond $t > 10$. For the same memory allocation the order- n scheme the calculation could be extended to much longer times with no difficulty: it only depends on the simulation time. In the present example, we stopped at $t = 200$. It is interesting to compare the accuracy of the two schemes. In the conventional scheme, the velocities of the particles at the current time step are used to update the mean-squared displacement of all time intervals. In the order- n scheme, the current time step is only used to update the lowest-order array of \mathbf{v}_{sum} (see Algorithm 10). The block sums of level i are updated only once every n^i time step. Therefore, for a total simulation of M time steps, the number of samples is much less for the order- n scheme; for the conventional scheme, we have M samples for all time steps, whereas the order- n scheme has M/n^i samples for the i th block velocity. Naively, one would think that the conventional scheme, therefore, is more accurate. In the conventional scheme, however, the successive samples will have much more correlation and therefore are not independent. To investigate the effect of these correlations on the accuracy of the results, we used the method of Flyvbjerg and Petersen [148] (see Appendix 5.3.3 and Case Study 4). In this method, the standard deviation is calculated as a function of the number of data blocks. If the data are correlated, the standard deviation will increase as a function of the number of blocks until the number of blocks is sufficient that the data in a data block are uncorrelated. If the data are uncorrelated, the standard deviation will be independent of the number of blocks. This limiting value is the standard deviation of interest.

In these simulations, the time step was $\Delta t = 0.001$, and the block length was set to $n = 10$. For both methods the total number of time steps was equal. To calculate the mean-squared displacement, we have used 100,000 samples for all times in the conventional scheme. For the order- n scheme, we have used 100,000 samples for $t \in [0, 0.01]$, 10,000 for $t \in [0.01, 0.1]$, 1,000 for $t \in [0.1, 1]$, etc. This illustrates that the number of samples in the order- n scheme is considerably less than in the conventional scheme. The accuracy of the results, however, turned out to be the same. This is shown in Fig. N.9 for $t = 0.1$ and $t = 1.0$. Since the total number of data blocking operations that can be performed on the data depends on the total number of samples, the number of blocking operations is less for the order- n method. Fig. N.9 shows that for $t = 0.1$ the order- n scheme yields a standard deviation that is effectively constant after three data blocking operations, indicating the samples are independent, whereas the standard deviation using the conventional method shows an increase for the first six to eight data blocking operations. For $t = 1.0$ the order- n method is independent of the number of data blocks, the conventional method only after 10 data blocks. This implies that one has to average over $2^{10} \approx 1000$ successive samples to have two independent data points. In addition, the figure shows that the plateau value of the standard deviation is essentially the same for the two methods, which implies that for this case, the two methods are equally accurate, but as shown below, the order- n method is computationally cheaper.

In Fig. N.10 we compare the CPU requirements of the two algorithms for simulations with a fixed total number of time steps. This figure shows the increase of the total CPU time of the simulation as a function of the total time for which the mean-squared displacement has been calculated. With the order- n scheme, the CPU time should be (nearly) independent of the total time for which we deter-

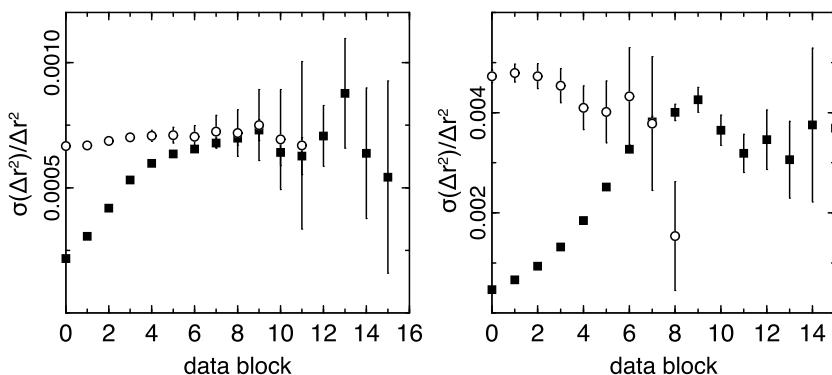


FIGURE N.9 The relative error in the mean-squared displacement as a function of the number of data blocks as defined by Flyvbjerg and Petersen. The figures compare the conventional scheme (solid squares) with the order- n method (open circles) to determine the mean-squared displacement. The right figure is for $t = 0.1$ and the left figure is for $t = 1.0$.

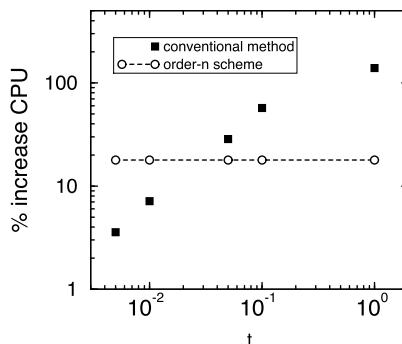


FIGURE N.10 Percentage increase of the total CPU time as a function of the total time for which we determine the mean-squared displacement; comparison of the conventional scheme with the order- n scheme for the same system as is considered in Fig. N.8.

mine the mean-squared displacement, which is indeed what we observe. For the conventional scheme, however, the required CPU time increases significantly for longer times. At $t = 1.0$ the order- n scheme gives an increase of the total CPU time of 17%, whereas the conventional scheme shows an increase of 130%.

This example illustrates that the saving in memory as well as in CPU time of the order- n scheme can be significant, especially if we are interested in the mean-squared displacement at long times.

N.7 Equation of state of the Lennard-Jones fluid

Case Study 7 (Equation of state of the Lennard-Jones fluid). Simulations at constant pressure can also be used to determine the equation of state of a pure component. In such a simulation, the density is determined as a function of the applied pressure and temperature. Fig. N.11 shows that, for the Lennard-Jones fluid, the results of an NPT simulation agree with those obtained in Case Study 1.

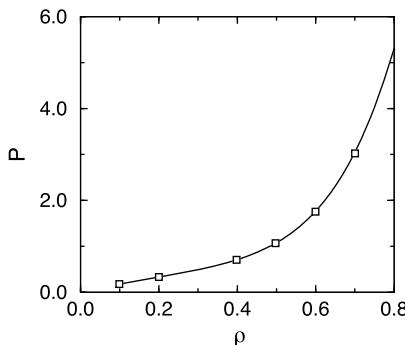


FIGURE N.11 Equation of state of the Lennard-Jones fluid as obtained from *NPT* simulations; isotherms at $T = 2.0$. The solid line is the equation of state of Johnson et al. [83] and the squares are the results from the simulations ($N = 108$).

In simulations of realistic models of molecular liquids, it is important to verify that the model fluid has the same density as the real fluid under atmospheric pressure. Using *NVT* simulations, one would require several simulations to determine the density for which the pressure is approximately 1 atm. In an *NPT* simulation, one would obtain this result in a single simulation. In reduced units, atmospheric pressure is typically much less than one. Hence, a good first estimate of the liquid density at atmospheric pressure follows from an *NPT* simulation at $P = 0$.³

N.8 Phase equilibria from constant-pressure simulations

Case Study 8 (Phase equilibria from constant-pressure simulations). In Case Studies 1 and 7 *NVT* or *NPT* simulations are used to determine the equation of state of a pure substance. In principle, we can construct the liquid-vapor coexistence curve by first fitting the simulation data to an analytical expression, and then determining the liquid and vapor densities for which P , T and μ are the same. Although this route to locate a coexistence curve is quite general, it requires many simulations. More efficient ways to determine the vapor-liquid coexistence curve are discussed in section 6.6. However, in cases where the vapor pressure of the liquid is very small ($\ll 1$ in reduced units), we can determine the density of the coexisting liquid by performing simulations at zero pressure.

To carry out an *NPT*-simulation at zero pressure it is best to start with a liquid density above the estimated coexistence density. During the simulation, the system will quickly reach the (metastable) density for which $P = 0$. However, starting from a lower density is not recommended as the system might then expand without bounds.

Fig. N.12 shows that the densities obtained in a zero-pressure simulation agree well with the true equation of state of the Lennard-Jones fluid, even up to a reduced temperature of 1, but not as the equilibrium density of the vapor becomes

³ Strictly speaking, the liquid is only metastable at $P = 0$. However, as the nucleation barrier for bubble formation is typically much larger than $k_B T$, the lifetime of this metastable state is usually much longer than the simulation time.

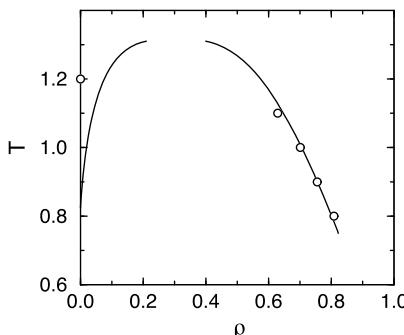


FIGURE N.12 Vapor-liquid coexistence curve of the Lennard-Jones fluid; the solid curve was computed using the Johnson equation of state [83]. The circles indicate the average liquid densities obtained from NPT simulations at $P = 0$. As can be seen from the figure, the zero-pressure method becomes less reliable for $T > 1.2$.

appreciable: for $T > 1$, the zero-pressure simulations predict liquid densities that are too low. Moreover, as the critical temperature T_c is approached, the surface tension tends to zero and, as a consequence, bubble nucleation becomes more likely. Under those conditions, the metastable liquid at $P = 0$ is increasingly likely to evaporate during a simulation. In short: do not use $P = 0$ simulations close to T_c . On a more positive note: not too close to the critical temperature, a reasonable estimate of the liquid density can be obtained by carrying out NPT simulations at $P = 0$.

N.9 Equation of state of the Lennard-Jones fluid - II

Case Study 9 (Equation of state of the Lennard-Jones fluid - II). In Case Studies 1 and 7, we computed the equation of state of a Lennard-Jones fluid, using respectively NVT and NPT simulations. A third way to determine the equation of state is to perform a *Grand-canonical* simulation, imposing the temperature T and the chemical potential μ at constant V , and sample the resulting density and pressure. An example of such a calculation is shown in Fig. N.13.

Grand-canonical simulations are not particularly useful for computing the equation-of-state of a homogeneous fluid, because there will be statistical errors in both the pressure and the density. However, for systems where the pressure itself is ill-defined (e.g., for nano-porous materials), grand-canonical simulations are the method of choice.

N.10 Phase equilibria of the Lennard-Jones fluid

Case Study 10 (Phase equilibria of the Lennard-Jones fluid). Here we use the Gibbs-ensemble technique to determine the vapor-liquid coexistence curve of the Lennard-Jones fluid. In Case Studies 1, 7, and 9, we had already determined parts of the equation of state of this fluid, and in Case Study 8 we had made an estimate of the liquid coexistence-density from a zero-pressure simulation.

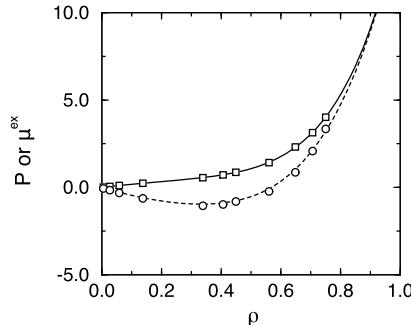


FIGURE N.13 Equation of state of the Lennard-Jones fluid; isotherm at $T = 2.0$. The solid curve represents the equation of state of Johnson et al. [83]; the squares are the results from grand-canonical simulations (with volume $V = 250.047$). The dotted curve is the excess chemical potential as calculated from the equation of state of ref. [83], and the circles are the simulation results. Note that the excess chemical potential is related to the fugacity f through $\beta\mu^{\text{ex}} = \ln(f/\rho)$.

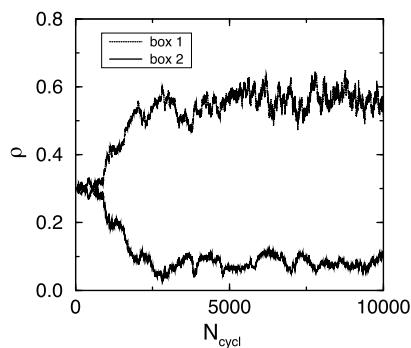


FIGURE N.14 Particle number density in the two boxes of the Gibbs ensemble as a function of the number of Monte Carlo cycles for a system of Lennard-Jones particles; the number of particles was $N = 256$ and temperature $T = 1.2$.

In Fig. N.14, the density of the fluid in the two boxes is plotted as a function of the number of Monte Carlo cycles (as defined in Algorithm 15). The simulation started with equal density in both boxes. During the first 1000 Monte Carlo cycles, the system has not yet “decided” which box would evolve to a liquid density and which to the vapor. After 5000 Monte Carlo cycles, the system seems to have reached equilibrium, and the coexistence properties can be determined.

In Fig. N.15, the phase diagram of the Lennard-Jones as obtained from Gibbs-ensemble simulations is compared with the phase diagram obtained from the equation of state of Johnson et al. [83]. We point out that comparison of the GE phase diagram with the literature data on the critical point (e.g., [212]) is complicated by the fact that different authors use different truncations of the Lennard-Jones potential (see e.g., [80,219]). Such truncations have a large effect on the location of the critical temperature.

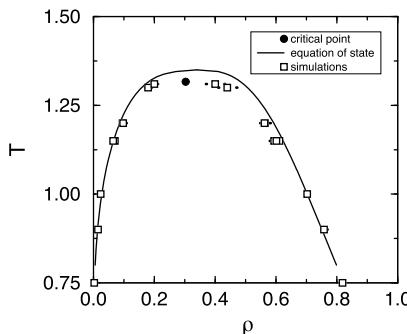


FIGURE N.15 Phase diagram of the Lennard-Jones fluid, using tail correction beyond the cut-off of 2.5σ to mimic the full Lennard-Jones potential, as calculated with the Gibbs-ensemble technique (squares) and equation of state of Johnson et al. (solid curves). The solid circle indicates the estimated critical point.

N.11 Use of Andersen thermostat

Case Study 11 (Use of Andersen thermostat). The Andersen thermostat [180] is arguably the simplest MD thermostat that has been proven to yield a canonical distribution. This implies that both the kinetic energy and the potential energy of the system under consideration are Boltzmann-distributed.

However, the Andersen algorithm conserves neither momentum nor energy, as it attributes new velocities drawn from a Maxwell-Boltzmann distribution to randomly selected particles. We refer to these updates as stochastic “collisions”. The effect of the stochastic collisions is that the dynamical properties of a system with an Andersen thermostat may differ substantially, or even catastrophically, from those of a system without a thermostat, or with a “global” thermostat [256,257].

In the case of diffusion, it is easy to understand intuitively that the Andersen thermostat will decrease the self-diffusion coefficient D^s : a Green-Kubo relation relates D to the integral of the velocity-auto-correlation function (VACF). The longer velocities persist, the larger D . Conversely, any effect that destroys the persistence in the velocity, will decrease D . And destroying the persistence in \mathbf{v} is precisely what the Andersen thermostat does: the higher ν , the frequency of stochastic collisions, the lower D . This effect is illustrated in Fig. N.16.

In practical cases, ν is usually chosen such that the timescale τ_E for the decay of energy fluctuations in a simulation of a system with linear dimension L is comparable to that of thermal fluctuations with a wavelength equal to L , in an unbounded medium: $\tau_E \approx [L^2(N/V)C_P]/\lambda$, where C_P is the heat capacity per molecule (at constant pressure) and λ is the thermal conductivity.

For reasonably large systems, such values of τ_E can only be achieved with rather low collision rates per particle, in which case the effect of collisions on the dynamics may be small [180], but at the same time, the thermostat becomes rather inefficient.

One should not use the Andersen thermostat when computing quantities such as the thermal conductivity or the viscosity.

The reason is that these quantities provide a measure for the rate at which energy or momentum diffuse. But such a description only makes sense if energy and

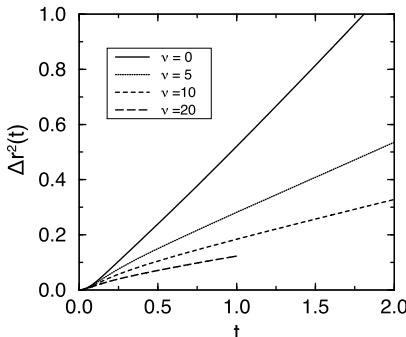


FIGURE N.16 Mean-squared displacement as a function of time for various values of the collision frequency ν of Andersen thermostat. The system studies are a Lennard-Jones fluid ($T = 2.0$ and $N = 108$).

momentum are conserved. With an Andersen thermostat, a local perturbation in the energy or momentum does not diffuse away, and rather, it is screened exponentially. This effect is not described by the (normal) diffusion equation.

Summarizing: do not use the Andersen method when computing transport properties. For static properties, it is fine—and very easy to implement.

Fig. N.16 shows that the frequency of stochastic collisions has a strong effect on the time dependence of the mean-squared displacement. The mean-squared displacement becomes only independent of ν in the limit of very low stochastic collision rates. Yet, all static properties, such as the pressure or potential energy are rigorously independent of the stochastic collision frequency.

N.12 Use of Nosé-Hoover thermostat

Case Study 12 (Use of Nosé-Hoover thermostat). As in Case Study 11, we start by showing that the Nosé-Hoover method reproduces the behavior of a system at constant NVT . In Fig. N.17 we compare the velocity distribution generated by the Nosé-Hoover thermostat with the correct Maxwell-Boltzmann distribution for the same temperature (7.1.1). The figure illustrates that the velocity distribution indeed is independent of the value chosen for the coupling constant Q .

It is instructive to see how the system reacts to a sudden increase in the imposed temperature. Fig. N.18 shows the evolution of the kinetic temperature of the system. After 12,000 time steps, the imposed temperature is suddenly increased from $T = 1$ to $T = 1.5$. The figure illustrates the role of the coupling constant Q . A small value of Q corresponds to a low inertia of the heat bath and leads to rapid temperature fluctuations. A large value of Q leads to a slow, ringing response to the temperature jump.

Next, we consider the effect of the Nosé-Hoover coupling constant Q on the diffusion coefficient. As can be seen in Fig. N.19, the effect is much smaller than in Andersen's method. However, it would be wrong to conclude that the diffusion coefficient is independent of Q . The Nosé-Hoover method simply provides a way to keep the temperature constant more gently than Andersen's method, where

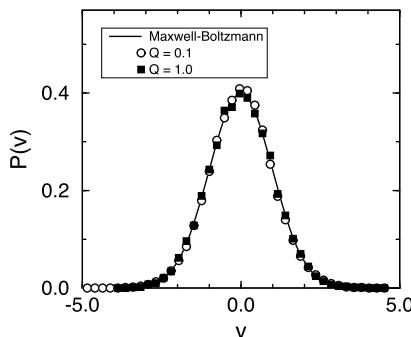


FIGURE N.17 Velocity distribution in a Lennard-Jones fluid ($T = 1.0$, $\rho = 0.75$, and $N = 256$). The solid line is the Maxwell-Boltzmann distribution (7.1.1) the symbols were obtained in a simulation using the Nosé-Hoover thermostat [256].

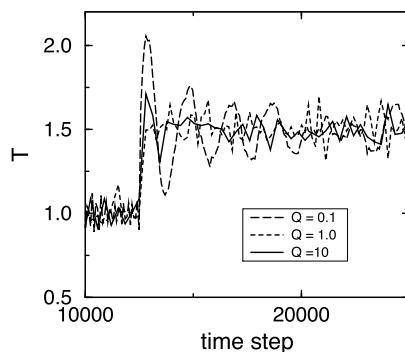


FIGURE N.18 Response of the system to a sudden increase in the imposed temperature. The various lines show the actual temperature of the system (a Lennard-Jones fluid $\rho = 0.75$, and $N = 256$) as a function of the number of time steps for various values of the Nosé-Hoover coupling constant Q .

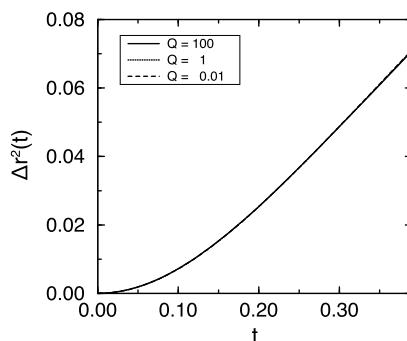


FIGURE N.19 Effect of the coupling constant Q on the mean-squared displacement for a Lennard-Jones fluid ($T = 1.0$, $\rho = 0.75$, and $N = 256$).

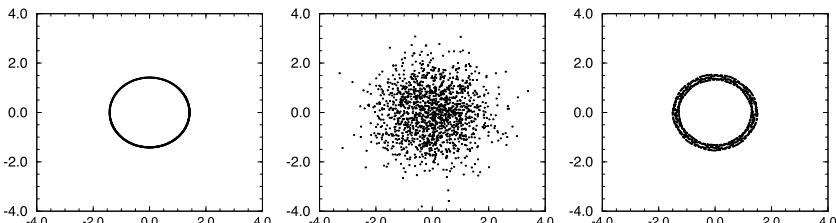


FIGURE N.20 Trajectories of the harmonic oscillator: (from left to right) in the microcanonical ensemble, using the Andersen method and using the Nosé-Hoover method. The y axis is the velocity and the x axis is the position.

particles suddenly get new, random velocities. For the calculations of transport properties, we prefer simple N , V , E simulations.

N.13 Harmonic oscillator (I)

Case Study 13 (Harmonic oscillator (I)). As the equations of motion of the harmonic oscillator can be solved analytically, this model system is often used to test algorithms. However, the harmonic oscillator is also a rather atypical dynamical system, as will become clear when we apply the NH algorithm to this simple model system.

The potential energy function of the harmonic oscillator is

$$u(r) = \frac{1}{2}r^2.$$

The Newtonian equations of motion are

$$\begin{aligned}\dot{r} &= v \\ \dot{v} &= -r.\end{aligned}$$

If we solve the equations of motion of the harmonic oscillator for a given set of initial conditions, we can trace the trajectory of the system in phase space. Fig. N.20 shows a typical phase space trajectory of the harmonic oscillator, in a closed loop, which is characteristic of periodic motion. It is straightforward to simulate a harmonic oscillator at a constant temperature using the Andersen thermostat (see section 7.1.1.1). A trajectory is shown in Fig. N.20. In this case, the trajectories are points that are not connected by lines. This is due to the stochastic collisions with the bath. In this example, we allowed the oscillator to interact with the heat bath at each time step. As a result, the phase space density is a collection of discrete points. The resulting velocity distribution is Gaussian by construction; also, for the positions, we find a Gaussian distribution.

We can also perform a constant-temperature Nosé-Hoover simulation using the algorithm described in SI L.6.2. A typical trajectory of the harmonic oscillator generated with the Nosé-Hoover scheme is shown in Fig. N.20. The most striking feature of Fig. N.20 is that, unlike the Andersen scheme, the Nosé-Hoover method does not yield a canonical distribution in phase space. Even for very long simulations, the entire trajectory would lie in the same ribbon shown in Fig. N.20. Moreover, this band of trajectories depends on the initial configuration.

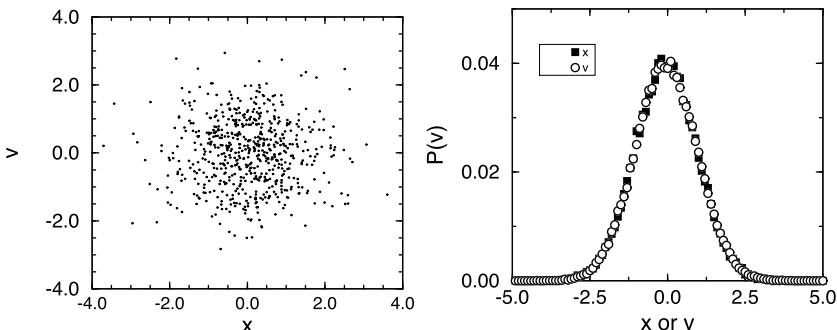


FIGURE N.21 Test of the phase space trajectory of a harmonic oscillator, coupled to a Nosé-Hoover chain thermostat. The left-hand side of the figure shows part of a trajectory: the dots correspond to consecutive points separated by 10,000 time steps. The right-hand side shows the distributions of velocity and position. Due to our choice of units, both distributions should be Gaussians of equal width.

N.14 Nosé-Hoover chain for harmonic oscillator

Case Study 14 (Nosé-Hoover chain for harmonic oscillator). The harmonic oscillator is the obvious model system on which we test the Nosé-Hoover chain thermostat. If we use a chain of two coupling parameters, the equations of motion are

$$\begin{aligned}\dot{r} &= v \\ \dot{v} &= -r - \xi_1 v \\ \dot{\xi}_1 &= \frac{v^2 - T}{Q_1} - \xi_1 \xi_2 \\ \dot{\xi}_2 &= \frac{Q_1 \xi_1^2 - T}{Q_2}.\end{aligned}$$

A typical trajectory generated with the Nosé-Hoover chains is shown in Fig. N.21. The distributions of the velocity and position of the oscillator are also shown in Fig. N.21. Comparison with the results obtained using the Andersen thermostat (see Case Study 13) shows that the Nosé-Hoover chains do generate a canonical distribution, even for the harmonic oscillator.

N.15 Chemical potential: particle-insertion method

Case Study 15 (Chemical potential: particle-insertion method). In this Case Study, we use the Widom test-particle method to determine the excess chemical potential of a Lennard-Jones fluid. The algorithm that we use is a combination of the basic algorithm for performing Monte Carlo simulations at constant N , V , and T (Algorithms 1 and 2) and determining the excess chemical potential (Algorithm 18).

We stress that the tail correction for the chemical potential is similar, but not identical, to that for the potential energy. In the Widom test-particle method, we

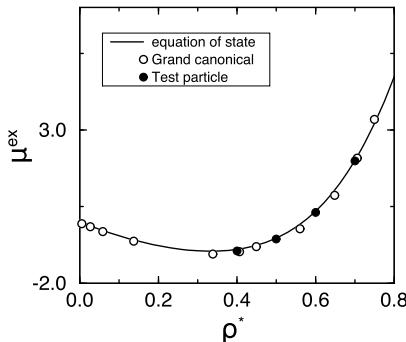


FIGURE N.22 The excess chemical potential of the Lennard-Jones fluid ($T = 2.0$) as calculated from the equation of state, grand-canonical Monte Carlo, and the test particle insertion method.

determine the energy difference:

$$\Delta \mathcal{U} = \mathcal{U}(\mathbf{s}^{N+1}) - \mathcal{U}(\mathbf{s}^N).$$

The tail correction is

$$\begin{aligned}
 \beta \mu^{\text{tail}} &= \mathcal{U}(\mathbf{s}^{N+1})^{\text{tail}} - \mathcal{U}(\mathbf{s}^N)^{\text{tail}} \\
 &= (N+1)u^{\text{tail}}((N+1)/V) - Nu^{\text{tail}}(N/V) \\
 &= \left((N+1)\frac{N+1}{V} - N\frac{N}{V} \right) \frac{1}{2} 4\pi \int_{r_c}^{\infty} dr r^2 u(r) \\
 &\approx \frac{2N}{V} \frac{1}{2} 4\pi \int_{r_c}^{\infty} dr r^2 u(r) \\
 &= 2u^{\text{tail}}(\rho).
 \end{aligned} \tag{N.15.1}$$

In Case Study 9, we performed a grand-canonical Monte Carlo simulation to determine the equation of state of the Lennard-Jones fluid. In the grand-canonical ensemble the volume, chemical potential, and temperature are imposed; the density is determined during the simulation. Of course, we can also calculate the chemical potential during the simulation, using the Widom method. Fig. N.22 shows a comparison of the imposed and measured chemical potentials.

N.16 Chemical potential: overlapping distributions

Case Study 16 (Chemical potential: overlapping distributions). In Case Study 15, we used the Widom test-particle method to determine the chemical potential of the Lennard-Jones fluid. This method fails at high densities, where it becomes extremely unlikely to insert a particle at a position where $\exp(-\beta \Delta \mathcal{U})$ in Eq. (8.5.5) is non-negligible. Yet it is those unlikely insertions that dominate the average $\langle \exp(-\beta \mathcal{U}) \rangle$. Because favorable insertions are so rare, the number of such events is subject to relatively large statistical fluctuations, and hence our estimate for $\langle \exp(-\beta \mathcal{U}) \rangle$ is noisy. The *overlapping-distribution method* does not remove this

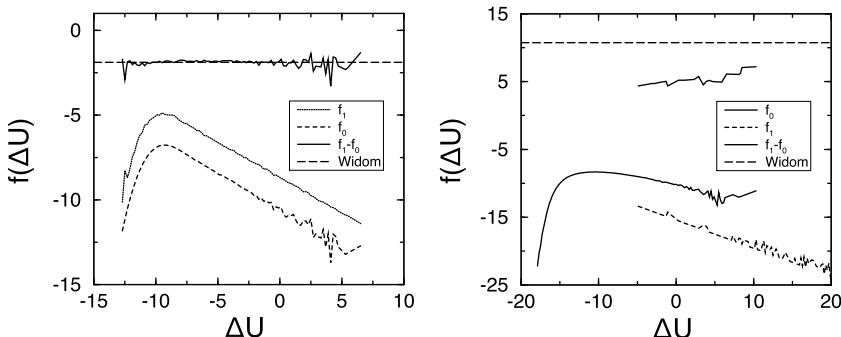


FIGURE N.23 Comparison of the overlapping distribution function method and the Widom particle insertion scheme for measuring the chemical potential of the Lennard-Jones fluid at $T = 1.2$. The solid curve is the particle-insertion result. The dashed curve was obtained using the overlapping distribution method ($\beta\mu_{\text{ex}} = f_1 - f_0$). The units for $\beta\mu_{\text{ex}}$ are the same as for $f(\Delta U)$. The figure on the left corresponds to a moderately dense liquid ($\rho = 0.7$). In this case, the distributions overlap and the two methods yield identical results. The right-hand figure corresponds to a dense liquid ($\rho = 1.00$). In this case, the insertion probability is very low. The distributions f_0 and f_1 hardly overlap, and the two different estimates of $\beta\mu_{\text{ex}}$ do not coincide.

problem, but it provides a good diagnostic tool for detecting such sampling problems.

To implement the overlapping-distribution method, we have to perform two simulations: one simulation using a system of $N + 1$ particles (system 1) and a second system with N particles and one ideal gas particle (system 0). For each of these systems, we determine the distribution of energy differences, Eqs. (8.6.3) and (8.6.4). For system 1, this energy difference ΔU is defined as the change in the total energy of the system that would result if one particle, chosen at random, would be transformed into an ideal gas particle. We now make a histogram of the observed values of ΔU in this system. This calculation can easily be appended to a standard MC move (Algorithm 2), because in a trial move, we randomly select a particle and compute its interaction energy before a trial move. But that interaction energy is precisely the ΔU that we wish to compute. We thus obtain a probability distribution of ΔU : $p_1(\Delta U)$.

For system 0, we have to determine the energy difference ΔU , which is the difference in total energy when the ideal gas particle (which could be anywhere in the system) would be turned into an interacting particle. This energy difference equals the energy of a test particle in the Widom method (section 8.5.1). When we determine $p_0(\Delta U)$, at the same time, we can obtain an estimate of the excess chemical potential from the Widom particle insertion method. As explained in the text, it is convenient not to use $p_0(\Delta U)$ and $p_1(\Delta U)$, but the closely related functions $f_0(\Delta U)$ and $f_1(\Delta U)$, defined in Eqs. (8.6.6) and (8.6.7).

In Fig. N.23 we show how $\mu_{\text{ex}}(\Delta U)$ can be obtained (using Eq. (8.6.8)) from a plot of $f_0(\Delta U)$ and $f_1(\Delta U)$, as a function of ΔU . The results shown in the left part of Fig. N.23 apply to a Lennard-Jones fluid at $\rho = 0.7$, the results on the right are for $\rho = 1.00$.

For the sake of comparison, we have also plotted the results obtained using the Widom particle insertion method. The figure shows that at $\rho = 0.7$ there is a sufficiently large range of energy differences for which the two functions overlap ($-10 < \Delta\mathcal{U} < -5$), in the sense that the noise in both functions is relatively small in this energy range. The result of the overlapping distribution function, therefore, is in good agreement with the results from the Widom method. However, at $\rho = 1.00$, the range of overlap is limited to the wings of the histograms p_0 and p_1 , where the statistical accuracy is poor. As a consequence, our estimate for $\mu_{\text{ex}}(\Delta\mathcal{U})$ is not constant (as it should) but appears to depend on $\Delta\mathcal{U}$. Moreover, the results from the overlapping distribution method are not consistent with the result of the Widom particle insertion method.

Note that two separate simulations are needed to determine the excess chemical potential from the overlapping distribution method. One might think that particle addition and particle removal histograms could be measured in a single simulation of an N -particle system. Such an approach would indeed be correct if there were no difference between the histograms for particle removal from N and $N + 1$ particle systems, but for dense systems containing only a few hundred particles, the system-size dependence of μ_{ex} can be appreciable.

Of course, sometimes simulations on small systems are performed as a preliminary step to larger-scale simulations. Then it is advisable to compute both $p_0(\Delta\mathcal{U})$ and $p_1(\Delta\mathcal{U})$ for the small system in a single simulation, as this allows us to check whether the overlap between the two distributions is likely to be sufficient to justify a larger-scale simulation.

N.17 Solid-liquid equilibrium of hard spheres

Case Study 17 (Solid-liquid equilibrium of hard spheres). In this case study, we locate the solid-liquid coexistence densities of the hard-sphere model. We determine these densities by equating the chemical potential and the pressure of the two phases.

For the liquid phase, we use the equation of state of Speedy [421], which is based on a Padé approximation to simulation data on both the equation of state and the virial coefficients of hard spheres:

$$z_{\text{liquid}} = \frac{P\beta}{\rho} = 1 + \frac{x + 0.076014x^2 + 0.019480x^3}{1 - 0.548986x + 0.075647x^2}.$$

For the solid phase of the hard-sphere model, Speedy proposed the following equation of state [320]:

$$z_{\text{solid}} = \frac{3}{1 - \rho^*} - 0.5921 \frac{\rho^* - 0.7072}{\rho^* - 0.601}, \quad (\text{N.17.1})$$

where $\rho^* = \sigma^3 \rho / \sqrt{2}$. In Fig. N.24, we compare the predictions of this equation of state for the liquid and solid phases with the results from computer simulations of Alder and Wainwright [422] and Adams [171]. As can be seen, the empirical equations of state reproduce the simulation data quite well. To calculate the chemical potential of the liquid phase, we integrate the equation of state (see (9.1.1)) starting

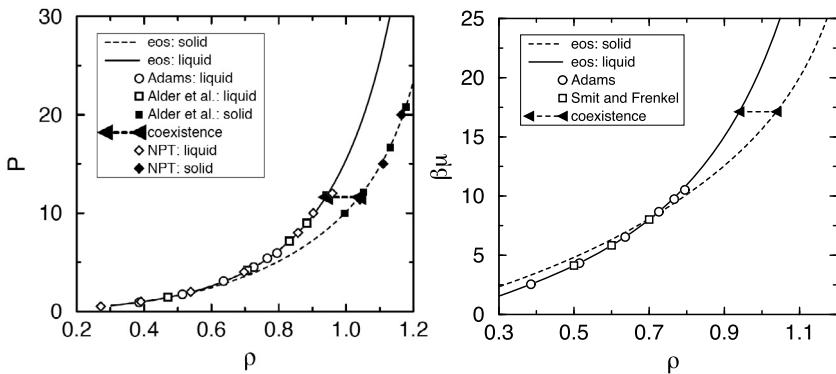


FIGURE N.24 Pressure P (left) and chemical potential μ (right) as a function of the density ρ . The solid curves, showing the pressure and chemical potential of the liquid phase, are obtained from the equation of state of Speedy [421]. The dashed curve gives the pressure of the solid phase as calculated from the equation of state of ref. [320]. The open and filled symbols are the results of computer simulations for the liquid [171,422,423] and solid phases [422], respectively. The coexistence densities are indicated with horizontal lines.

from the dilute gas limit. This yields the Helmholtz free energy as a function of the density. The chemical potential then follows from

$$\beta\mu(\rho) = \frac{\beta G}{N} = \frac{\beta F}{N} + \frac{P}{\rho k_B T}.$$

The free energy per particle of the ideal gas is given by

$$\beta f^{\text{id}}(\rho) = \frac{F^{\text{id}}(\rho)}{Nk_B T} = \ln \rho \Lambda^3 - 1,$$

where Λ is the de Broglie thermal wavelength. In what follows, we shall write

$$\beta f^{\text{id}}(\rho) = \ln \rho - 1.$$

That is, we shall work with the usual reduced densities and ignore the additive constant $3 \ln(\Lambda/\sigma)$, as it plays no role in the location of phase equilibria for classical systems.

Fig. N.24 compares the chemical potential that follows from the Hall equation of state with some of the available simulation data (namely, grand-canonical ensemble simulations of [171] and direct calculations of the chemical potential, using the Widom test-particle method [423] (see Chapter 8).

These results show that we have an accurate equation of state for the liquid phase and the solid phase. Since we know the absolute free energy of the ideal gas phase, we can calculate the free energy and hence the chemical potential of the liquid phase. For the solid phase we can use the equation of state to calculate only free energy differences; to calculate the absolute free energy we have to determine the free energy at a particular density. To perform this calculation, we use the lattice coupling method.

e98 Supporting information for case studies

We must now select the upper limit of the coupling parameter λ (λ_{\max}) and the values of λ for which we perform the simulation. For sufficiently large values of λ we can calculate $\sum_{i=1}^N (\mathbf{r}_i - \mathbf{r}_{0,i})^2$ analytically, using

$$\langle r^2 \rangle_\lambda = \frac{1}{N} \frac{\partial F(\lambda)}{\partial \lambda}.$$

For the noninteracting Einstein crystal, the mean-squared displacement is given by

$$\langle r^2 \rangle_\lambda = \frac{3}{2\beta\lambda}. \quad (\text{N.17.2})$$

For a noninteracting Einstein crystal with fixed center of mass, the free energy is given by Eq. (9.2.23), which gives

$$\langle r^2 \rangle_{\text{Ein},\lambda} = \frac{1}{\beta} \frac{3}{2} \frac{N-1}{N} \frac{1}{\lambda}. \quad (\text{N.17.3})$$

In [314] an analytical expression is derived for the case of an interacting Einstein crystal, which reads

$$\begin{aligned} \langle r^2 \rangle_\lambda &= \langle r^2 \rangle_{\text{Ein},\lambda} - \frac{\beta n}{2} \frac{1}{2a(2\pi\beta\lambda)^{(1/2)} \left(1 - \langle P_{\text{overlap}}^{\text{nn}} \rangle_\lambda\right)} \\ &\times \left\{ [\sigma a - \sigma^2 - 1/(\beta\lambda)] \exp[-\beta\lambda(a-\sigma)^2/2] \right. \\ &\left. + [\sigma a + \sigma^2 - 1/(\beta\lambda)] \exp[-\beta\lambda(a+\sigma)^2/2] \right\}, \end{aligned} \quad (\text{N.17.4})$$

where a is the separation of two nearest neighbors i and j , $a = \mathbf{r}_{0,i} - \mathbf{r}_{0,j}$, σ is the hard-core diameter, and n is the number of nearest neighbors (for example, $n = 12$ for Face Centered Cubic (FCC) and hcp (hexagonal close-packed) solids or $n = 8$ for bcc (body-centered cubic)); $\langle P_{\text{overlap}}^{\text{nn}} \rangle_\lambda$ is the probability that two nearest neighbors overlap. Such probability is given by

$$\begin{aligned} \langle P_{\text{overlap}}^{\text{nn}} \rangle_\lambda &= \frac{\operatorname{erf}[(\beta\lambda/2)^{1/2}(\sigma+a)] + \operatorname{erf}[(\beta\lambda/2)^{1/2}(\sigma-a)]}{2} \\ &- \frac{\exp[-\beta\lambda(\sigma-a)^2/2] - \exp[-\beta\lambda(\sigma+a)^2/2]}{(2\pi\beta\lambda)^{1/2}a} \end{aligned} \quad (\text{N.17.5})$$

This equation can also be used to correct the free energy of a noninteracting Einstein crystal (9.2.23):

$$\frac{\beta F_{\text{Ein}}(\lambda)}{N} = \frac{\beta F_{\text{Ein}}}{N} + \frac{n}{2} \ln \left(1 - \langle P_{\text{overlap}}^{\text{nn}} \rangle_\lambda \right). \quad (\text{N.17.6})$$

We choose λ_{\max} such that, for values of λ larger than this maximum value, $\langle r^2 \rangle_\lambda$ obeys the analytical expression. Typically, this means that the probability of overlap of two harmonically bound particles should be considerably less than 1%. The results of these simulations are presented in Fig. N.25. This figure shows that if we

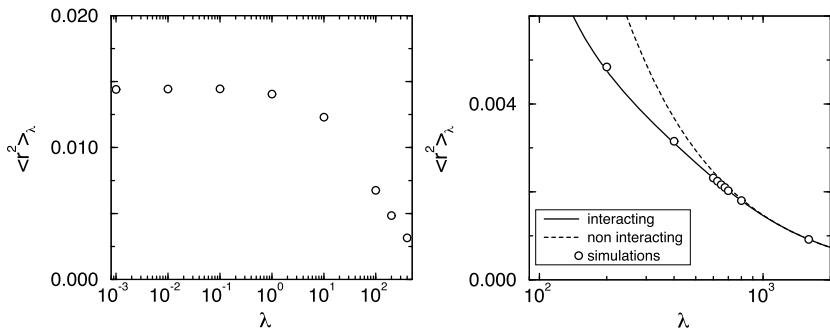


FIGURE N.25 The mean-squared displacement $\langle r^2 \rangle_\lambda$ as a function of the coupling parameter λ for a hard-sphere (FCC) solid of 54 particles (6 layers of 3×3 close-packed atoms at a density $\rho = 1.04$). The figure on the left shows the simulation results for low values of λ , the figure on the right for high values. The solid line takes into account nearest-neighbor interactions (N.17.4); the dashed line assumes a noninteracting Einstein crystal (N.17.3). The open symbols are the simulation results.

rely only on the analytical results of the noninteracting Einstein crystal, we have to take a value for $\lambda_{\max} \approx 1000\text{--}2000$. If we use Eq. (N.17.4) for $\langle r^2 \rangle_\lambda$, $\lambda_{\max} = 500\text{--}1000$ is sufficient.

We should now integrate

$$\frac{\Delta F}{N} = \int_0^{\lambda_{\max}} d\lambda \langle r^2 \rangle_\lambda.$$

In practice, this integration is carried out by numerical quadrature. We therefore must specify the values of λ for which we are going to compute $\langle r^2 \rangle_\lambda$. To improve the accuracy of the numerical quadrature, it is convenient to transform to another integration variable:

$$\frac{\Delta F}{N} = \int_0^{\lambda_{\max}} \frac{d\lambda}{g(\lambda)} g(\lambda) \langle r^2 \rangle_\lambda = \int_{G^{-1}(0)}^{G^{-1}(\lambda_{\max})} d[G^{-1}(\lambda)] g(\lambda) \langle r^2 \rangle_\lambda,$$

where $g(\lambda)$ is an as-yet arbitrary function of λ and $G^{-1}(\lambda)$ is the primitive of the function $1/g(\lambda)$. If we can find a function $g(\lambda)$ such that the integrand, $g(\lambda) \langle r^2 \rangle_\lambda$, is a slowly varying function, we need fewer function evaluations to arrive at an accurate estimate. To do this, we need to have an idea about the behavior of $\langle r^2 \rangle_\lambda$.

For $\lambda \rightarrow 0$, $\langle r^2 \rangle_\lambda \rightarrow \langle r^2 \rangle_0$, which is the mean-squared displacement of an atom around its lattice site in the normal hard-sphere crystal. At high values of λ , where the system behaves like an Einstein crystal, we have $\langle r^2 \rangle_\lambda \rightarrow 3k_B T/(2\lambda)$. This leads to the following guess for the functional form of $g(\lambda)$:

$$g(\lambda) \approx k_B T / \langle r^2 \rangle_\lambda \approx c + \lambda,$$

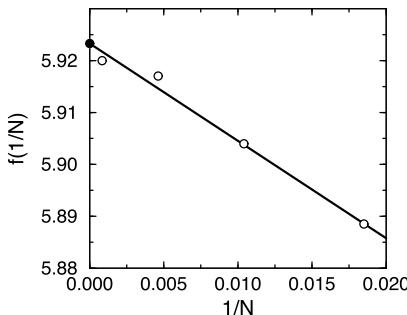


FIGURE N.26 $\beta F^{\text{ex}}/N + \ln(N)/N$ versus $1/N$ for an FCC crystal of hard spheres at a density $\rho\sigma^3 = 1.0409$. The solid line is a linear fit to the data. The coefficient of the $1/N$ term is $-6.0(2)$, and the intercept (i.e., the infinite system limit of $\beta F^{\text{ex}}/N$) is equal to $5.91889(4)$.

where $c = k_B T / \langle r^2 \rangle_0$. Here, $\langle r^2 \rangle_0$ can be estimated from Fig. N.25. The value of c clearly depends on density (and temperature). For $\rho = 1.04$, extrapolation to $\lambda \rightarrow 0$ gives $\langle r^2 \rangle_0 \approx 0.014$, which gives $c = 70$. If we use this function $g(\lambda)$, the free energy difference is calculated from

$$\frac{\Delta F}{N} = \int_{\ln c}^{\ln(\lambda_{\max}+c)} d[\ln(\lambda+c)] (\lambda+c) \langle r^2 \rangle_\lambda .$$

For the numerical integration, we use a n -point Gauss-Legendre quadrature [424]. As the integrand is a smooth function, a 10-point quadrature is usually adequate. As discussed in section 9.2.5, the resulting free energy still depends (slightly) on the system size. An example of the system-size dependence of the excess free energy of a hard-sphere crystal is shown in Fig. N.26 [425]. From this figure, we can estimate the excess free energy of the infinite system to be $\beta f^{\text{ex}} = 5.91889(4)$. This is in good agreement with the estimate of Frenkel and Ladd, $\beta f^{\text{ex}} = 5.9222$ [314].

Once we have one value of the absolute free energy of the solid phase at a given density, we can compute the chemical potential of the solid phase at any other density, using the equation of state of Speedy (see Fig. N.24). The coexistence densities follow from the condition that the chemical potentials and pressures in the coexisting phases should be equal.

Using the value of $5.91889(4)$ from [425] for the solid at $\rho = 1.04086$, we arrive at a freezing density $\rho_l = 0.9391$ and a melting density $\rho_s = 1.0376$. At coexistence, the pressure is $P_{\text{coex}} = 11.567$ and the chemical potential is $\mu_{\text{coex}} = 17.071$. In fact, as we shall argue below, the presence of vacancies in the equilibrium crystal lowers the coexistence pressure slightly: $P_{\text{coex}} = 11.564$. These results are in surprisingly good agreement with the original data of Hoover and Ree [307], who obtained an estimate for the solid-liquid coexistence densities $\rho_s = 1.041 \pm 0.004$ and $\rho_l = 0.943 \pm 0.004$ at a pressure 11.70 ± 0.18 .

The free energy difference between the FCC and HCP for large hard-sphere crystals at melting is very close to 0, but the FCC structure appears to be the more stable phase [303, 412, 426, 427].

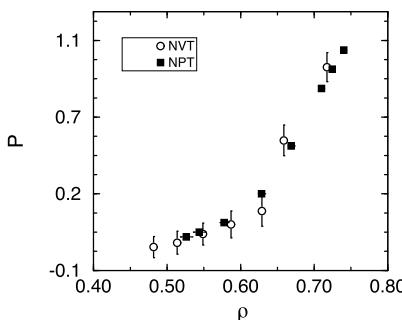


FIGURE N.27 Equation of state of an eight-bead Lennard-Jones chain as obtained from N, V, T and N, P, T simulations using the configurational-bias Monte Carlo scheme. The simulations are performed with 50 chains at a temperature $T = 1.9$.

N.18 Equation of state of Lennard-Jones chains

Case Study 18 (Equation of state of Lennard-Jones chains). To illustrate the configurational-bias Monte Carlo technique described in this section, we determine the equation of state of a system consisting of eight-bead chains of Lennard-Jones particles. The nonbonded interactions are described by a truncated and shifted Lennard-Jones potential. The potential is truncated at $R_c = 2.5\sigma$. The bonded interactions are described with a harmonic spring

$$u^{\text{vib}}(l) = \begin{cases} 0.5k_{\text{vib}}(l-1)^2 & 0.5 \leq l \leq 1.5 \\ \infty & \text{otherwise} \end{cases},$$

where l is the bond length, the equilibrium bond length has been set to 1, and $k_{\text{vib}} = 400$.

The simulations are performed in cycles. In each cycle, we perform on average N_{dis} attempts to displace a particle, N_{cbmc} attempts to (partly) regrow a chain, and N_{vol} attempts to change the volume (only in the case of N, P, T simulations). If we regrow a chain, the configurational-bias Monte Carlo scheme is used. In this move we select at random the monomer from which we start to regrow. If this happens to be the first monomer, the entire molecule is regrown at a random position. For all the simulations, we used eight trial orientations. The lengths of trial bonds are generated with a probability prescribed by the bond-stretching potential (see Case Study 19).

In Fig. N.27 the equation of state as obtained from N, V, T simulations is compared with one obtained from N, P, T simulations. This isotherm is well above the critical temperature of the corresponding monomeric fluid ($T_c = 1.085$, see Fig. 3.3), but the critical temperature of the chain molecules is appreciably higher [512].

N.19 Generation of trial configurations of ideal chains

Case Study 19 (Generation of trial configurations of ideal chains). In section 12.2.3, we emphasized the importance of efficiently generating trial segments for mole-

e102 Supporting information for case studies

cules with strong intramolecular interactions in a [CBMC](#) simulation. In this Case Study, we quantify this. We consider the following bead-spring model of a polymer. The nonbonded interactions are described with a Lennard-Jones potential, and the bonded interactions with a harmonic spring:

$$u^{\text{vib}}(l) = \begin{cases} 0.5k_{\text{vib}}(l - 1)^2 & 0.5 \leq l \leq 1.5 \\ \infty & \text{otherwise} \end{cases},$$

where l is the bond length, the equilibrium bond length has been set to 1, and $k_{\text{vib}} = 400$. The bonded interaction is only the bond stretching. The external (nonbonded) interactions are the Lennard-Jones interactions. We consider the following two schemes of generating a set of trial positions:

1. Generate a random orientation with bond length uniformly distributed in the spherical shell between limits chosen such that they bracket all acceptable bond lengths. For instance, we could consider limits that correspond to a 50% stretching or compression of the bond. In that case, the probability of generating bond length l is given by

$$p_1(l) = \begin{cases} \propto C dl \propto l^2 dl & 0.5 \leq l \leq 1.5 \\ 0 & \text{otherwise} \end{cases}.$$

2. Generate a random orientation and the bond length prescribed by the bond-stretching potential (as described in Algorithm 25). The probability of generating bond length l with this scheme is

$$p_2(l) = \begin{cases} \propto C \exp[-\beta u^{\text{vib}}(l)] dl = C \exp[-\beta u^{\text{vib}}(l)] l^2 dl & 0.5 \leq l \leq 1.5 \\ 0 & \text{otherwise} \end{cases}.$$

Let us consider a case in which the system consists of ideal chains. Ideal chains are defined (see section 12.2.3) as chains having only *bonded* interactions.

Suppose we use method 1 to generate the set of k trial orientations with bond lengths l_1, \dots, l_k , then the Rosenbluth factor, [W](#), for atom i is given by

$$w_i(n) = \sum_{j=1}^k \exp[-\beta u^{\text{vib}}(l_j)].$$

The Rosenbluth factor of the entire chain is

$$W(n) = \prod_{i=1}^{\ell} w_i(n).$$

For the old conformation, a similar procedure is used to calculate its Rosenbluth factor:

$$W(o) = \prod_{i=1}^{\ell} w_i(o).$$

In absence of external interactions, the Rosenbluth factor of the first atom is defined to be $w_1 = k$.

In the second scheme, we generate the set of k trial orientations with a bond length distribution $p_2(l)$. If we use this scheme, we have to consider only the external interaction. Since, for an ideal chain, the external interactions are, by definition, 0, the Rosenbluth factor for each atom is given by

$$w_i^{\text{ext}}(n) = \sum_{j=1}^k \exp[-\beta u^{\text{ext}}(l_j)] = k,$$

and similarly, for the old conformation

$$w_i^{\text{ext}}(o) = k.$$

Hence, the Rosenbluth weight is the same for the new and the old conformations:

$$W^{\text{ext}}(n) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(n) = k^{\ell}$$

and

$$W^{\text{ext}}(o) = \prod_{i=1}^{\ell} w_i^{\text{ext}}(o) = k^{\ell}.$$

The acceptance rule for the first scheme is

$$\text{acc}(o \rightarrow n) = \min[1, W(n)/W(o)]$$

and for the second scheme is

$$\text{acc}(o \rightarrow n) = \min[1, W^{\text{ext}}(n)/W^{\text{ext}}(o)] = 1.$$

Inspection of these acceptance rules shows that, in the second scheme, *all* configurations generated are accepted, whereas in the first scheme, this probability depends on the bond-stretching energy and, therefore, will be less than 1. Hence, it is clearly useful to employ the second scheme.

To show that the results of schemes 1 and 2 are indeed equivalent, we compare the distribution of the bond length of the chain and the distribution of the radius of gyration in Fig. N.28. The figure shows that the results for the two methods are indeed indistinguishable. The efficiency of the two methods, however, is very different. In Table N.1, the difference in acceptance probability is given for some values of the bond-stretching force constant and various chain lengths. The table shows that if we use method 1 and generate a uniformly distributed bond length, we need to use at least 10 trial orientations to have a reasonable acceptance for chains longer than 20 monomers. Note that the corresponding table for the second method has a 100% acceptance for all values of k independent of the chain length.

Most of the simulations, however, do not involve ideal chains but chains with external interactions. For chains with external interactions, the first method performs even worse. First of all, we generate the chains the same way as in the case

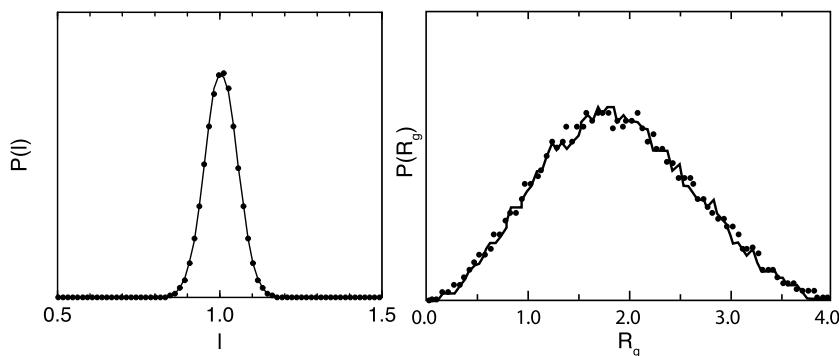


FIGURE N.28 Comparison of methods 1 and 2 for the distribution of bond lengths l (left) and the distribution of the radius of gyration R_g (right). The solid lines represent the results for method 1, and the dots for method 2 ($\ell = 5$ and $k = 5$).

TABLE N.1 Probability of acceptance (%) for ideal chains using uniformly distributed bond lengths (method 1), where ℓ is the chain length and k is the number of trial orientations. The value for the spring constant is $k_{\text{vib}} = 400$ (see [440]). For method 2, the acceptance would have been 100% for all values of k and ℓ .

k	$\ell = 5$	$\ell = 10$	$\ell = 20$	$\ell = 40$	$\ell = 80$	$\ell = 160$
1	0.6	$\ll 0.01$				
5	50	50	10	$\ll 0.01$	$\ll 0.01$	$\ll 0.01$
10	64	58	53	42	$\ll 0.01$	$\ll 0.01$
20	72	66	60	56	44	$\ll 0.01$
40	80	72	67	62	57	40
80	83	78	72	68	62	60

of the ideal chains. The bonded interactions are the same, and we need to generate at least the same number of trial directions to get a reasonable acceptance. In addition, if there are external interactions, we have to calculate the nonbonded interactions for *all* of these trial positions. The calculation of the nonbonded interactions takes most of the CPU time, yet, in the first method, most of the trial orientations are doomed to be rejected solely on the basis of the bonded energy. These two reasons make the second scheme much more attractive than the first.

N.20 Recoil growth simulation of Lennard-Jones chains

Case Study 20 (Recoil growth simulation of Lennard-Jones chains). To illustrate the [Recoil Growth \(RG\)](#) method, we make a comparison between this method and [Configurational-Bias Monte Carlo \(CBMC\)](#). Consider 20 Lennard-Jones chains of length 15. The monomer density is $\rho = 0.3$ at temperature $T = 6.0$. Two bonded monomers have a constant bond length of 1.0, while three successive particles have a constant bond angle of 2.0 radians.

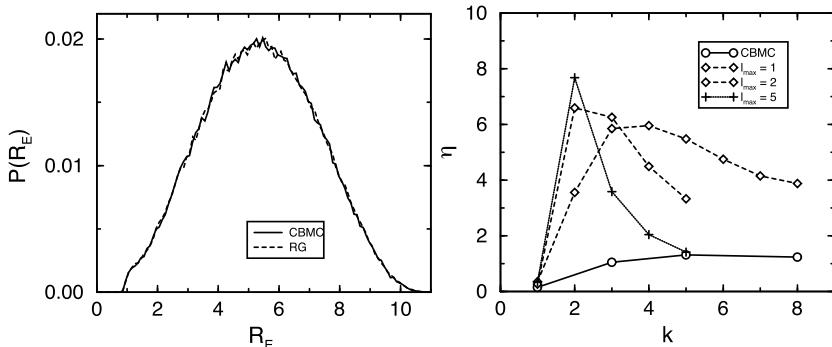


FIGURE N.29 Comparison of configurational-bias Monte Carlo (CBMC) with recoil growth for the simulation of Lennard-Jones chains of length 15. The left figure gives the distribution of the end-to-end distance (R_E). In the right figure the efficiency (η) is a function of the number of trial directions (k) for different recoil lengths (l_{\max}) as well as for CBMC.

In Fig. N.29, the distribution of the end-to-end vector, R_E , of the chain is plotted. In this figure, we compare the results from a CBMC and a RG. Since both methods generate a Boltzmann distribution of conformations, the results are identical (as they should be).

For this specific Case Study, we have compared the efficiency, η , of the two methods. The efficiency is defined as the number of accepted trial moves per amount of CPU time. For CBMC we see that the efficiency increases as we increase k , the number of trial orientations, from 1 to 4. From 4 to 8, the efficiency is more or less constant, and above 8, a decrease in the efficiency is observed.

In the RG scheme, we have two parameters to optimize: the number of trial orientations k and the recoil length l_{\max} . If we use only one trial orientation, recoiling is impossible since there are no other trial orientations. If we use a recoil length of 1, the optimum number of trial orientations is 4, and for larger recoil lengths, the optimum is reached with fewer trial orientations. Interestingly, the global optimum is 2 trial orientations and a recoil length of 3–5. In this regime, the increase in CPU time associated with a larger recoil length is compensated by a higher acceptance. In the present study, optimal RG was a factor 8 more efficient than optimal CBMC.

Case Study 21 (Parallel tempering of a single particle). As a trivial illustration of the power of parallel tempering, we consider a single particle moving an external potential as shown in Fig. N.30(left):

$$U(x) = \begin{cases} \infty & x < -2 \\ 1 \times (1 + \sin(2\pi x)) & -2 \leq x \leq -1.25 \\ 2 \times (1 + \sin(2\pi x)) & -1.25 \leq x \leq -0.25 \\ 3 \times (1 + \sin(2\pi x)) & -0.25 \leq x \leq 0.75 \\ 4 \times (1 + \sin(2\pi x)) & 0.75 \leq x \leq 1.75 \\ 5 \times (1 + \sin(2\pi x)) & 1.75 \leq x \leq 2 \\ \infty & x > 2 \end{cases} \quad (\text{N.20.1})$$

We place the particle initially in the left-most potential energy well.

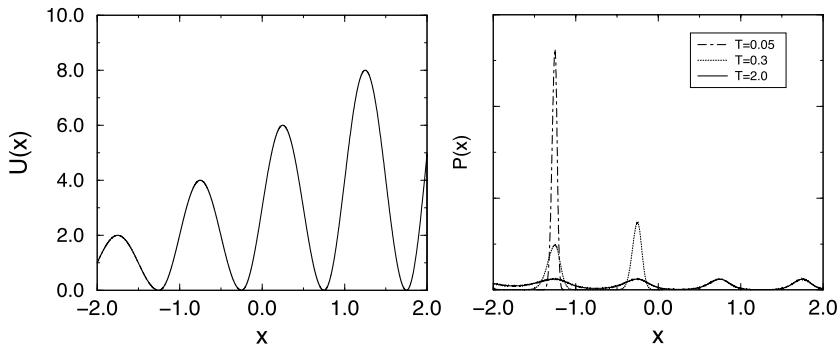


FIGURE N.30 (Left) Potential energy ($U(x)$) as a function of the position x . (Right) Probability ($P(x)$) of finding a particle at position x for various temperatures (T) as obtained from ordinary Monte Carlo simulations. The lower-temperature systems are not (or barely) able to cross the energy barriers separating the wells.

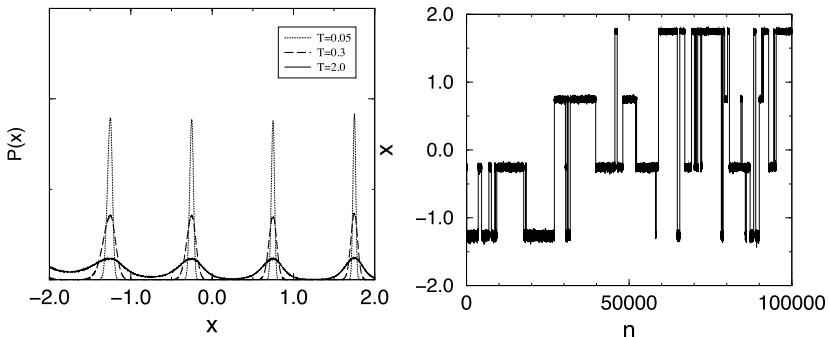


FIGURE N.31 (Left) Probability ($P(x)$) of finding a particle at position x for various temperatures (T) using parallel tempering. (Right) Position (x) as a function of the number of Monte Carlo trial moves (n) for $T = 0.05$. To avoid cluttering this figure with too many jumps, we have only used 0.05% swap moves here.

We first use normal Metropolis MC at three different temperatures ($T = 0.05$, 0.3, and 2.0). At the lowest temperature ($T = 0.05$), the particle is basically trapped in its initial potential energy well during the entire simulation, whereas for the highest temperature ($T = 2.0$), it can explore all wells.

Next, we apply parallel tempering, that is: we allow for temperature swaps between the three systems (see Fig. N.31). Due to the temperature-swap moves, the systems now equilibrate rapidly at all three temperatures. The difference is particularly strong for the probability distribution at the lowest temperature. In the present parallel-tempering simulation, we consider two types of trial moves:

1. Particle displacement: we randomly select one of the three temperatures and carry out a trial displacement Δ of a randomly selected particle at that temperature, choosing the trial displacement Δ for a random distribution, uniform between -0.1 and 0.1 . The acceptance of this trial displacement is determined

by the conventional Metropolis MC rule

$$\text{acc}(o \rightarrow n) = \min\{1, \exp[-\beta(U(n) - U(o))]\}. \quad (\text{N.20.2})$$

2. Temperature swapping. The trial move consists of attempting to swap two randomly selected neighboring temperatures (T_i and T_j). Such a trial move is accepted with a probability given by Eq. (13.1.2)

$$\text{acc}(o \rightarrow n) = \min(1, \exp[(\beta_j - \beta_i) \times (U_j - U_i)]). \quad (\text{N.20.3})$$

We are free to choose the relative rate of displacement and swap moves. In the present example, we used 10% swap moves and 90% particle displacements, as suggested in ref. [550]. Note, however, that other choices are possible.⁴ As can be seen in Fig. N.31, parallel tempering results in a dramatic improvement in the sampling of configuration space at the lowest temperature.

N.21 Multiple time step versus constraints

Case Study 22 (Multiple time step versus constraints). In this Case Study, we consider a system of diatomic Lennard-Jones molecules. We compare two models: the first model uses a fixed bond length l_0 between the two atoms of a molecule. In the second model, we use a bond-stretching potential given by

$$U_{\text{bond}}(l) = \frac{1}{2}k_b(l - l_0)^2,$$

where l is the distance between the two atoms in a molecule. In the simulations, we used $k_b = 50000$ and $l_0 = 1$. In addition to the bond-stretching potential, all non-bonded atoms interact via a Lennard-Jones potential. The total number of diatomics was 125, and the box length 7.0 (in the usual reduced units). The Lennard-Jones potential was truncated at $r_c = 3.0$, while $T = 3.0$. The equations of motion are solved using bond constraints for the first model, while multiple time steps were used for the second model. All simulations were performed in the *NVE* ensemble.

It is interesting to compare the maximum time steps that can be used to solve the equations of motion for these two methods. As a measure of the accuracy with which the equations of motion are solved, we compute the average deviation of the initial energy, which is defined by Martyna et al. [623] as

$$E = \frac{1}{N_{\text{step}}} \sum_{i=1}^{N_{\text{step}}} \left| \frac{E(i \Delta t) - E(0)}{E(0)} \right|,$$

in which $E(i)$ is the total energy at time i .

⁴ There are even algorithms that use an infinite swap rate [551], including all possible permutations of the temperatures. However, such algorithms (see also [552]) do not scale well with the number of distinct temperatures.

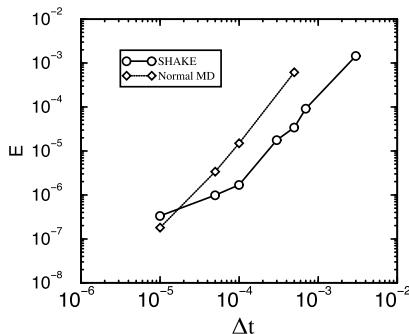


FIGURE N.32 Comparison of the energy fluctuations as a function of the time step for a normal MD simulation with a harmonic bond potential and a constrained MD simulation with the SHAKE algorithm.

For the bond constraints, we use the SHAKE algorithm [606] (see also section 14.1). In the SHAKE algorithm, the bond lengths are exactly fixed at l_0 using an iterative scheme. In Fig. N.32 (left), the energy fluctuations are shown as a function of the time step. Normally one tolerates a noise level in E of $\mathcal{O}(10^{-5})$, which would correspond to a time step of 2×10^{-4} for the first model. This should be compared with a single-time-step Molecular Dynamics simulation using the second model. A similar energy noise level can be obtained with a time step of 9×10^{-5} , which is a factor 2 smaller.

To apply the multiple-time-step algorithm, we have to separate the intermolecular force into a short-range and a long-range part. In the short-range part, we include the bond-stretching potential and the short-range part of the Lennard-Jones potential. To make a split in the Lennard-Jones potential, we use a simple switching function $S(r)$:

$$\begin{aligned} U_{LJ}(r) &= U_{\text{short}}(r) + U_{\text{long}}(r) \\ U^{\text{short}}(r) &= S(r) \times U_{LJ}(r) \\ U^{\text{long}}(r) &= [1 - S(r)] U_{LJ}(r), \end{aligned}$$

where

$$S(r) = \begin{cases} 1 & 0 < r < r_c - \lambda \\ 1 + \gamma^2 (2\gamma - 3) & r_m - \lambda < r < r_m \\ 0 & r_m < r < r_c \end{cases}$$

and

$$\gamma = \frac{r - r_m + \lambda}{\lambda}. \quad (\text{N.21.1})$$

In fact, there are other ways to split the total potential function [624,625]. We have chosen $\lambda = 0.3$ and $r_m = 1.7$. To save CPU time, a list is made of all the atoms that are close to each other (see Appendix I for details); therefore, the calculation of the

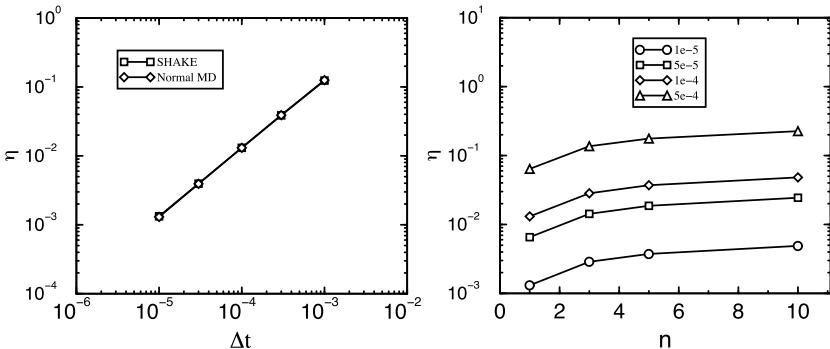


FIGURE N.33 Comparison of the efficiency η for bond constraints (SHAKE) with normal molecular dynamics (left) and multiple times steps (right). The left figure gives the efficiency as a function of the time step and the right figure as a function of the number of small time steps n , $\Delta t = n\delta t$, where the value of δt is given in the symbol legend.

short-range forces can be done very efficiently. For a noise level of 10^{-5} , one is able to use $\delta t = 10^{-4}$ and $n = 10$, giving $\Delta t = 10^{-3}$.

To compare the different algorithms in a consistent way, we compare in Fig. N.33 the efficiency of the various techniques. The efficiency η is defined as the length of the simulation (time step times the number of integration steps) divided by the amount of CPU time that was used. In the figure, we have plotted η for all simulations from Fig. N.32. For an energy noise level of 10^{-5} , the SHAKE algorithm is twice as efficient than normal MD ($n = 1$). This means that hardly any CPU time is spent in the SHAKE routine. However, the MTS algorithm is still two times faster ($n = 10$, $\delta t = 10^{-4}$) at the same efficiency.

N.22 Ideal gas particle over a barrier

Case Study 23 (Ideal gas particle over a barrier). To illustrate the “Bennett-Chandler” approach for calculating crossing rates, we consider an ideal gas particle moving in an external field. This particle is constrained to move on the dimensional potential surface shown in Fig. N.34. This example is rather unphysical because the moving particle cannot dissipate its energy. As a consequence, the motion of the particle is purely ballistic. We assume that, far away on either side of the barrier, the particle can exchange energy with a thermal reservoir. Transition state theory predicts a crossing rate given by Eq. (15.2.2):

$$k_{A \rightarrow B}^{\text{TST}} = \frac{1}{2} |\dot{q}| \frac{\exp[-\beta u(q^*)]}{\int_{-\infty}^{q^*} dq \exp[-\beta u(q)]} = \sqrt{\frac{k_B T}{2\pi m}} \frac{\exp[-\beta u(q^*)]}{\int_{-\infty}^{q^*} dq \exp[-\beta u(q)]}. \quad (\text{a})$$

If we choose the dividing surface q_1 (see Fig. N.34) at the top of the barrier ($q_1 = q^*$) none of the particles that start off with a positive velocity will return to the reactant state. Hence, there is no recrossing of the barrier and transition state theory is exact for this system.

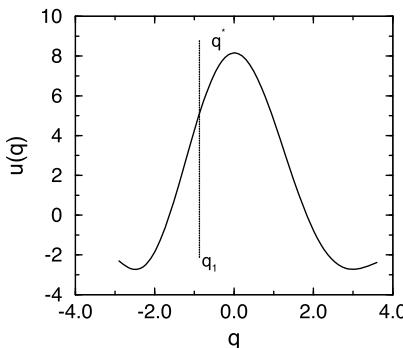


FIGURE N.34 Potential-energy barrier for an ideal gas particle; if the particle has a position to the left of the dividing surface q_1 the particle is in state A (reactant). The region to the right of the barrier is designated as product B . The top of the barrier is denoted by q^* ($q^* = 0$).

Note that transition state theory (Eq. (a)) predicts a rate constant that depends on the location of the dividing surface. In contrast, the Bennett-Chandler expression for the crossing rate is independent of location of the dividing surface (as it should be). To see this, consider the situation that the dividing surface is chosen to be the left of the top of the barrier (i.e., at $q_1 < q^*$). The calculation of the crossing rate according to Eq. (15.2.3) proceeds in two steps. First we calculate the relative probability of finding a particle at the dividing surface. And then we need to compute the probability that a particle that starts with an initial velocity \dot{q} from this dividing surface will, in fact, cross the barrier. The advantage of the present example is that this probability can be computed explicitly. According to Eq. (15.2.4), the relative probability of finding a particle at q_1 is given by

$$\frac{\langle \delta(q - q_1) \rangle}{\langle \theta(q_1 - q) \rangle} = \frac{\exp[-\beta u(q_1)]}{\int_{-\infty}^{q_1} dq \exp[-\beta u(q)]}. \quad (\text{b})$$

If the dividing surface is not at the top of the barrier, then the probability of finding a particle will be higher at q_1 than at q^* , but the fraction of the number of particles that actually cross the barrier will be less than predicted by transition state theory. It is convenient to introduce the time-dependent transmission coefficient, κ . $\kappa = \kappa(t)$, defined as the ratio

$$\kappa(t) \equiv \frac{k_{A \rightarrow B}(t)}{k_{A \rightarrow B}^{\text{TST}}} = \frac{\langle \dot{q}(0) \delta(q(0) - q_1) \theta(q(t) - q_1) \rangle}{0.5 \langle |\dot{q}(0)| \rangle}. \quad (\text{c})$$

The behavior of $\kappa(t)$ is shown in Fig. N.35 for various choices of q_1 . The figure shows that for $t \rightarrow 0$ $\kappa(t) = 1$, and that for different values of q_1 we get different plateau values. The reason $\kappa(t)$ decays from its initial value is that particles that start off with too little kinetic energy cannot cross the barrier and recross the dividing surface (q_1). The plateau value of $\kappa(t)$ provides us with the correction that has to be applied to the crossing rate predicted by transition state theory. Hence, we see that as we change q_1 , the probability of finding a particle at q_1 goes up, and the transmission coefficient goes down. But, as can be seen from Fig. N.35, the actual

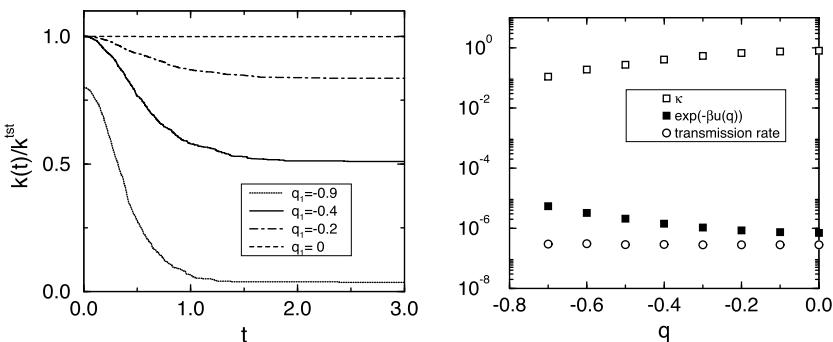


FIGURE N.35 Barrier recrossing: the left figure gives the transmission coefficient as a function of time for different values of q_1 . The right-hand figure shows, in a single plot, the probability density of finding the system at $q = q_1$ (solid squares), the transmission coefficient κ (open squares), and the overall crossing rate (open circles), all plotted as a function of the location of the dividing surface. Note that the overall crossing rate is independent of the choice of the dividing surface.

crossing rate (which is proportional to the product of these two terms) is independent of q_1 , as it should be. Now consider the case that $q_1 > q^*$. In that case, all particles starting with positive \dot{q} will continue to the product side. But now there is also a fraction of the particles with negative \dot{q} that will proceed to the product side. These events will give a negative contribution to κ . And the net result is that the transmission coefficient will again be less than predicted by transition state theory. Hence, the important thing is not if a trajectory ends up on the product side, but if it starts on the reactant side and proceeds to the product side. In a simulation, it is therefore convenient always to compute trajectories in pairs: for every trajectory starting from a given initial configuration with a velocity \dot{q} , we also compute the time-reversed trajectory, i.e., the one starting from the same configuration with a velocity $-\dot{q}$. If both trajectories end up on the same side of the barrier then their total contribution to the transmission coefficient is clearly zero. Only if the forward and time-reversed trajectories end up on different sides of the barrier, do we get a contribution to κ . In the present (ballistic) case, this contribution is always positive. But in general, this contribution can also be negative (namely, if the initial velocity at the top of the barrier is not in the direction where the particle ends up).

We chose this simple ballistic barrier-crossing problem because we can easily show explicitly that the transmission rate is independent⁵ of the location of q_1 . We start with the observation that the sum of the kinetic and potential energies of a particle that crosses the dividing surface q_1 is constant. Only those particles that have sufficient kinetic energy can cross the barrier. We can easily compute the long-time limit of $\langle \dot{q}(0)\theta(q(\infty) - q_1) \rangle$:

$$\langle \dot{q}(0)\theta(q(\infty) - q_1) \rangle = \sqrt{\frac{m\beta}{2\pi}} \int_{v_e}^{\infty} dv v \exp(-\beta mv^2/2)$$

⁵ The general proof that the long-time limit of the crossing rate is independent of the location of the dividing surface was given by Miller [630].

$$= \sqrt{\frac{1}{2\pi m\beta}} \exp(-\frac{1}{2}\beta mv_\epsilon^2),$$

where v_ϵ is the minimum velocity needed to achieve a successful crossing. v_ϵ is given by

$$\frac{1}{2}mv_\epsilon^2 + u(q_1) = u(q^*).$$

It then follows that

$$\langle \dot{q}(0)\theta(q(\infty) - q_1) \rangle = \sqrt{\frac{1}{2\pi m\beta}} \exp\{-\beta[u(q^*) - u(q_1)]\}.$$

This term exactly compensates the Boltzmann factor, $\exp(-\beta u(q_1))$, associated with the probability of finding a particle at q_1 . Hence, we have shown that the overall crossing rate is given by Eq. (a), independent of the choice of q_1 .

The reader may wonder why it is important to have an expression for the rate constant that is independent of the precise location of the dividing surface. The reason is that, although it is straightforward to find the top of the barrier in a one-dimensional system, the precise location of the saddle point in a reaction pathway of a many-dimensional system is usually difficult to determine. With the Bennett-Chandler approach it is not necessary to know the exact location of the saddle point. Still, it is worth trying to get a reasonable estimate, as the statistical accuracy of the results is best if the dividing surface is chosen close to the true saddle point.

The nice feature of the Bennett-Chandler expression for barrier-crossing rates is that it allows us to compute rate constants under conditions where barrier recrossings are important, for instance, if the motion over the top of the barrier is more diffusive than ballistic. Examples of such systems are the cyclohexane interconversion in a solvent [631] and the diffusion of nitrogen in an argon crystal [632].

N.23 Single particle in a two-dimensional potential well

Case Study 24 (Single particle in a two-dimensional potential well). To illustrate the path sampling method, consider a system containing a single particle in the following simple two-dimensional potential [641]:

$$V(x, y) = \left(4(1 - x^2 - y^2)^2 + 2(x^2 - 2)^2 + ((x + y)^2 - 1)^2 + ((x - y)^2 - 1)^2 - 2 \right) / 6. \quad (\text{N.23.1})$$

Note that $V(x, y) = V(-x, y) = V(x, -y)$. Fig. N.36 shows that this potential consists of two stable regions around the points $(-1, 0)$, which we call A, and $(1, 0)$, which we call region B. To be more specific, all points within a distance of 0.7 from $(-1, 0)$ or $(1, 0)$ are defined to be in region A or B, respectively. At a temperature of $T = 0.1$ transitions from A to B are rare events.

To compute the rate of transitions from A to B we used path ensemble simulations. The initial distribution $\mathcal{N}(x_0)$ was chosen to be canonical, i.e.,

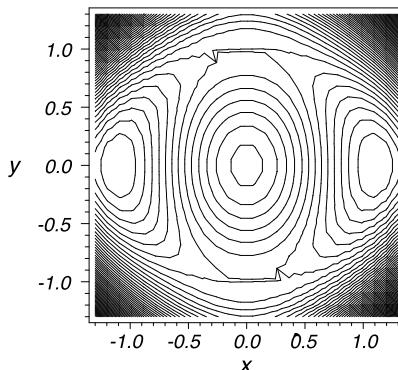


FIGURE N.36 Contour plot of the function $V(x, y)$ defined by Eq. (N.23.1). The two minima are at $(-1, 0)$, A, and $(0, 1)$, B. These minima are separated by a potential energy barrier.

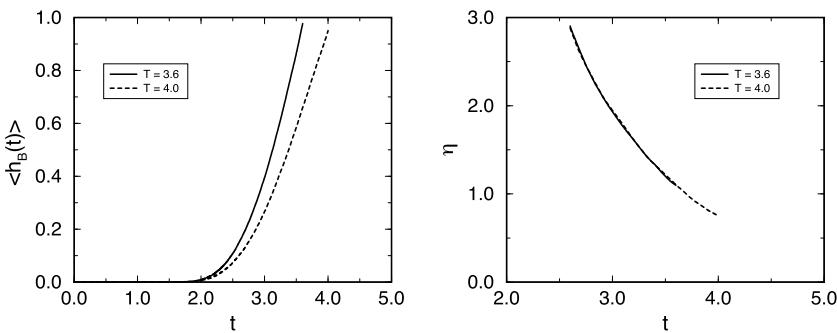


FIGURE N.37 $h_B(t)$ (left) and $\eta(t)$ (right) as a function of time for various values of the total path length T .

$$\mathcal{N}(x_0) \propto \exp[-\beta \mathcal{H}(x_0)].$$

A trajectory was generated using standard Molecular Dynamics simulations (see Chapter 4). The equations of motion were integrated using the velocity-Verlet algorithm with a time step of 0.002.

The first step was the calculation of the coefficient $\eta(t, t')$. This involves the computation of the path ensemble averages $\langle h_B(x_t) \rangle_{A, H_B(\mathcal{T})}$ for various times t . The result of such a simulation is shown in Fig. N.37 for $\mathcal{T} = 4.0$ and $\mathcal{T} = 3.6$. An important question is whether the time \mathcal{T} is long enough. Since we are interested in the plateau of $k(t)$, the function $\langle h_B(x_t) \rangle_{A, H_B(\mathcal{T})}$ must have become a straight line for large values of t . If this function does not show a straight line, the value of \mathcal{T} was probably too short, the process is not a rare event, or the process cannot be described by a single hopping rate. The consistency of the simulations can be tested by comparing the results with a simulation using a shorter (or longer, but this is more expensive) T . Fig. N.37 shows that the results of the two simulations are consistent.

The next step is the calculation of the correlation function $C(t)$. For the calculation of $P(\lambda, t)$, we have defined the order parameter λ as the distance from

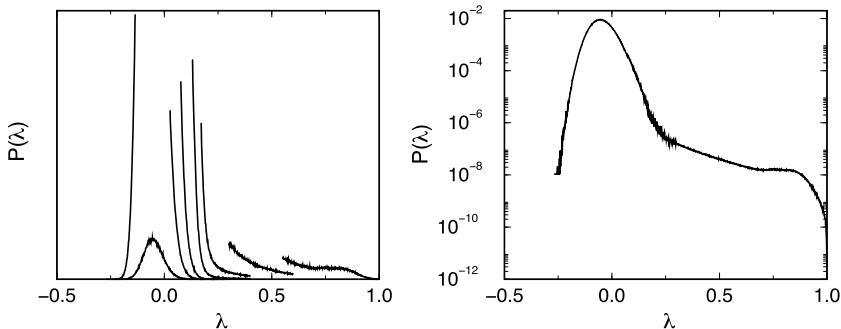


FIGURE N.38 (Left) $P(\lambda, i, t = 3.0)$ for all slices i . (Right) $P(\lambda, t = 3.0)$ when all slices i are combined. The units on the y axis are such that $\int_{-\infty}^1 d\lambda P(\lambda, t) = 1$.

point B:

$$\lambda = 1 - \frac{|\mathbf{r} - \mathbf{r}_B|}{|\mathbf{r}_A - \mathbf{r}_B|}, \quad (\text{N.23.2})$$

in which $\mathbf{r}_B = (1, 0)$. In this way, the region B is defined by $0.65 < \lambda \leq 1$, and the whole phase space is represented by $(-\infty, 1]$. In Fig. N.38 (left), we have plotted $P(\lambda, i, t = 3.0)$ as a function of λ for different slices i . Recombining the slices leads to Fig. N.38 (right). The value of $C(t = 3.0)$ can be obtained by integrating over region B:

$$C(t) = \int_B d\lambda P(\lambda, t). \quad (\text{N.23.3})$$

Combining the results gives for the total crossing rate

$$k = \eta(t) C(t). \quad (\text{N.23.4})$$

Using $t = 3.0$ leads to $\eta(2.0) = 1.94$, $C(3.0) = 4.0 \times 10^{-6}$, and $k = 8.0 \times 10^{-6}$.

N.24 Dissipative particle dynamics

Case Study 25 (Dissipative particle dynamics). To illustrate the **Dissipative Particle Dynamics (DPD)** technique, we have simulated a system of two components (1 and 2). The conservative force is a soft repulsive force, given by

$$\mathbf{f}_{ij}^C = \begin{cases} a_{ij} (1 - r_{ij}) \hat{\mathbf{r}}_{ij} & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases}, \quad (\text{N.24.1})$$

in which $r_{ij} = \|\mathbf{r}_{ij}\|$ and r_c is the cutoff radius of the potential. The random forces are given by Eq. (16.1.3) and the dissipative forces by Eq. (16.1.2). The total force on a particle equals the sum of the individual forces:

$$\mathbf{f}_i = \sum_{i \neq j} \left(\mathbf{f}_{ij}^C + \mathbf{f}_{ij}^S + \mathbf{f}_{ij}^R + \mathbf{f}_{ij}^D \right). \quad (\text{N.24.2})$$

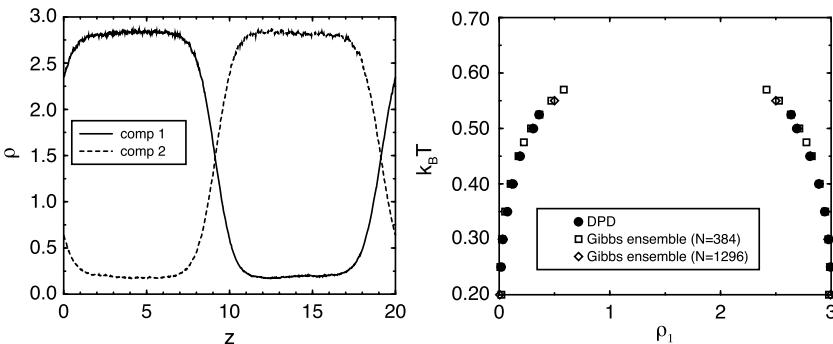


FIGURE N.39 (Left) Density profile for $k_B T = 0.45$. (Right) Phase diagram as calculated using DPD and Gibbs ensemble simulations. Both techniques result in the same phase diagram, but the Gibbs ensemble technique needs fewer particles due to the absence of a surface. In the DPD simulations, we have used a box of $10 \times 10 \times 20$ (in units of r_c^3). The time step of the integration was $\Delta t = 0.03$.

To obtain a canonical distribution, we use

$$\sigma^2 = 2\gamma k_B T$$

$$w^D(r_{ij}) = [w^R(r_{ij})]^2.$$

A simple but useful choice is [682]

$$w^D(r_{ij}) = \begin{cases} (1 - r_{ij}/r_c)^2 & r_{ij} < r_c \\ 0 & r_{ij} \geq r_c \end{cases}$$

with $r_c = 1$. The simulations were performed with $\rho = 3.0$ and $\sigma = 1.5$. We have chosen $a_{ii} = a_{jj} = 25$ and $a_{ij,i \neq j} = 30$. This system will separate into two phases. In the example shown in Fig. N.39, we have chosen the z -direction perpendicular to the interface. The left part of Fig. N.39 shows typical density profiles of the two components. In Fig. N.39 (right), we have plotted the concentration of one of the components in the coexisting phases.

Since we can write down a Hamiltonian for a DPD system, we can also perform standard Monte Carlo simulations [691]. For example, we can also use a Gibbs ensemble simulation (see section 6.6) to compute the phase diagram. As expected, Fig. N.39 shows that both techniques give identical results. Of course, due to the presence of an interface one needs many more particles in such a DPD simulation.

Thermodynamic quantities are calculated using only the conservative force. The pressure of the system is calculated using

$$p = \rho k_B T + \frac{1}{3V} \sum_{i>j} \langle \mathbf{r}_{ij} \cdot \mathbf{f}_{ij}^C \rangle.$$

If the DPD fluid undergoes phase separation into two slabs containing the different fluid phases, and we can compute the associated interfacial tension using the

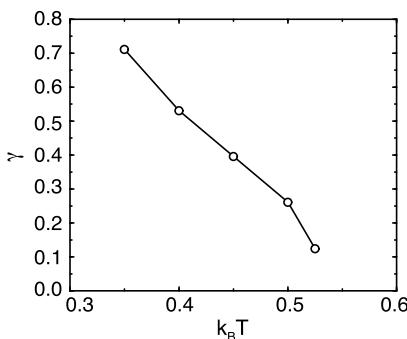


FIGURE N.40 Surface tension as a function of temperature.

techniques described in Section 5.1.6. Fig. N.40 shows this interfacial tension γ as a function of temperature. Clearly, γ decreases with increasing temperature and should vanish at the critical point. As the critical point is approached, the driving force for the formation of a surface (surface tension) becomes very low, and it becomes difficult, therefore, to form a stable interface in a small system.

N.25 Comparison of schemes for the Lennard-Jones fluid

Case Study 26 (Comparison of schemes for the Lennard-Jones fluid). It is instructive to make a detailed comparison of the various schemes to save CPU time for the Lennard-Jones fluid. We compare the following schemes:

1. Verlet list
2. Cell list
3. Combination of Verlet and cell lists
4. Simple N^2 algorithm

We have used the program of Case Study 1 as a starting point. At this point it is important to note that we have not tried to optimize the parameters (such as the Verlet radius) for the various methods; we have simply taken some reasonable values.

For the Verlet list (and for the combination of Verlet and cell lists) it is important that the maximum displacement be smaller than twice the difference between the Verlet radius and cutoff radius. For the cutoff radius we have used $r_c = 2.5\sigma$, and for the Verlet radius $r_v = 3.0\sigma$. This limits the maximum displacement to $\Delta_x = 0.25\sigma$ and implies for the Lennard-Jones fluid that, if we want to use a optimum acceptance of 50%, we can use the Verlet method only for densities larger than $\rho > 0.6\sigma^{-3}$. For smaller densities, the optimum displacement is larger than 0.25. Note that this density dependence does not exist in a Molecular Dynamics simulation. In a Molecular Dynamics simulation, the maximum displacement is determined by the integration scheme and therefore is independent of density. This makes the Verlet method much more appropriate for a Molecular Dynamics simulation than for a Monte Carlo simulation. Only at high densities does it make sense to use the Verlet list.

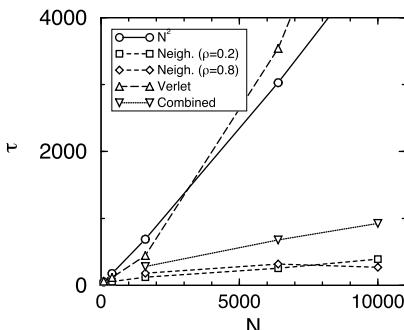


FIGURE N.41 Comparison of various schemes to calculate the energy: τ is in arbitrary units and N is the number of particles. As a test case, the Lennard-Jones fluid is used. The temperature was $T^* = 2$ and per cycle, the number of attempts to displace a particle was set to 100 for all systems. The lines serve to guide the eye.

The cell list method is advantageous only if the number of cells is larger than 3 in at least one direction. For the Lennard-Jones fluid this means that if the number of particles is 400, the density should be lower than $\rho < 0.5\sigma^{-3}$. An important advantage of the cell list over the Verlet list is that this list can also be used for moves in which a particle is given a random position.

From these arguments, it is clear that if the number of particles is smaller than 200–500, the simple N^2 algorithm is the best choice. If the number of particles is significantly larger and the density is low, the cell list method is probably more efficient. At high density, all methods can be efficient, and we have to make a detailed comparison.

To test these conclusions about the N dependence of the CPU time of the various methods, we have performed several simulations with a fixed number of Monte Carlo cycles. For the simple N^2 algorithm, the CPU time per attempt is

$$\tau_{N^2} = cN,$$

where c is the CPU time required to calculate one interaction. This implies that the total amount of CPU time is independent of the density. For a calculation of the total energy, we have to do this calculation N times, which gives the scaling of N^2 . Fig. N.41 shows that, indeed for the Lennard-Jones fluid, the τ_{N^2} increases linearly with the number of particles.

If we use the cell list, the CPU time will be

$$\tau_l = c_l V_l \rho + c_l p_l N,$$

where V_l is the total volume of the cells that contribute to the interaction (in three dimensions, $V_l = 27r_c^3$), c_l is the amount of CPU time required to make a cell list, and p_l is the probability that a new list has to be made. Fig. N.41 shows that the use of a cell list reduces the CPU time for 10,000 particles with a factor 18. Interestingly, the CPU time does not increase with increasing density. We would expect an increase since the number of particles that contribute to the interaction of a particle i increases with density. However, the second contribution to τ_{Neigh}

e118 Supporting information for case studies

(p_l) is the probability that a new list has to be made, depends on the maximum displacement, which decreases when the density increases. Therefore, this last term will contribute less at higher densities.

For the Verlet scheme, the CPU time is

$$\tau_v = c V_v \rho + c_v p_v N^2,$$

where V_v is the volume of the Verlet sphere (in three dimensions, $V_v = 4\pi r_v^3/3$), c_v is the amount of CPU time required to make the Verlet-list, and p_v is the probability that a new list has to be made. Fig. N.41 shows that this scheme is not very efficient. The N^2 operation dominates the calculation. Note that we use a program in which a new list for all particles has to be made as soon as one of the particles has moved more than $(r_v - r_c)/2$; with some more bookkeeping, it is possible to make a much more efficient program, in which a new list is made for only the particle that has moved out of the list.

The combination of the cell and Verlet lists removes the N^2 dependence on the simple Verlet algorithm. The CPU time is given by

$$\tau_c = c V_v \rho + c_v p_v c_l N.$$

Fig. N.41 shows that indeed the N^2 dependence is removed, but the resulting scheme is not more efficient than the cell list alone.

This case study demonstrates that it is not simple to give a general recipe for which method to use. Depending on the conditions and number of particles, different algorithms are optimal. It is important to note that for a Molecular Dynamics simulation, the conclusions may be different.

Appendix O

Small research projects

In this Appendix, we list a few small research projects. These projects involve the development of your own program. A possible strategy is to use the source code of one of the Case Studies as the starting point. It is our experience that the completion of such a research project takes about two weeks, depending on your experience.

O.1 Adsorption in porous media

In this project, we will investigate the adsorption behavior in porous media. As a model we use a slit-like pore. The interactions with the pore are given by

$$U(z) = \begin{cases} 0 & 0 < z < L \\ \infty & \text{otherwise} \end{cases}, \quad (\text{O.1.1})$$

where L is the width of the slit. We will investigate the adsorption of methane, which we model with a Lennard-Jones potential. The starting point is Case Study 9, which uses a program to simulate the Lennard-Jones fluid in the grand-canonical ensemble. The project is to develop a program to simulate an adsorption isotherm of methane in the slit-like pore.

Before starting to program you may want to think about the following points:

1. What is the geometry of the system and how should one apply the periodic boundary conditions?
2. What is the type of Lennard-Jones potential (truncated, truncated and shifted, with or without tail corrections)? And what are the parameters for modeling methane?
3. How is an adsorption isotherm defined thermodynamically?

With your program, try to answer the following questions:

1. Compute the density profile (density as function on the distance z between the plates) for $L = 1, 2, 5$, and 10 for $\rho = 0.7$ and $T = 2.0$. The total number of particles should be of the order of 100 to 500 for the widest slit.
2. Compute the excess chemical potential and chemical potential as a function of the distance between the plates for $L = 1, 2, 5$, and 10 for $\rho = 0.6$ and $T = 2.0$. Try to explain the differences.
3. Compute the adsorption isotherms for $L = 2$ and 5 for $T = 2.0$ and 0.8 . Try to explain the results.

O.2 Transport properties of liquids

Molecular Dynamics simulations can be used to compute the transport properties of liquids. Examples of these transport properties are the (self-) diffusion coefficient and the viscosity. In Case Study 5 the results for the diffusion coefficient are shown. In this project we extend these results to mixtures of Lennard-Jones (LJ) molecules. Before starting to program you may want to think about the following points:

1. The generalization of the diffusion coefficient from a pure component to a mixture is not trivial. Try to find in the literature how one should define a diffusion coefficient of a mixture and how can one compute this in a simulation. See, for example, refs. [59,784].
2. What is the type of LJ potential (truncated, truncated and shifted, with or without tail corrections)? And what are the parameters for modeling argon and krypton?

With your program, try to answer the following questions:

1. Compute the pressure, viscosity, and diffusion coefficient of the LJ fluid at $T = 1.0, 1.5$, and 2.0 for $\rho = 0.7$.
2. Compute the diffusion coefficients D_{11} , D_{12} , and D_{22} for a mixture of 50%-50% LJ particles in which the components 1 and 2 have the same interactions ($\sigma_{12} = \sigma_{11} = \sigma_{22}$ and $\epsilon_{12} = \epsilon_{11} = \epsilon_{12}$) but carry a different color. This means that the particles are labeled. Experimentally this could be done by radioactive labeling.
3. Compute the pressure, viscosity, and diffusion coefficient of the LJ fluid at $T = 1.0, 1.5$, and 2.0 for $\rho = 0.7$. Compute the diffusion coefficients D_{11} , D_{12} , and D_{22} for a mixture of 50-50% LJ particles, but now for a system of 50% argon and 50% krypton (use the parameters of argon to compute the reduced temperatures). Does the Einstein equation for the diffusivity hold for this system?

O.3 Diffusion in a porous medium

The behavior of liquid in confined geometries is different from the behavior in the bulk liquid. In the project we investigate the diffusion coefficient in a cylindrical pore. The starting point of the present study is Case Study 5, which describes the simulations of the dynamic properties of a Lennard-Jones fluid. The interactions with the walls are described with a repulsive potential:

$$U(r) = \begin{cases} \epsilon \left(\frac{\sigma}{L-r} \right)^{10} & 0 \leq r \leq L \\ \infty & r > L \end{cases}, \quad (\text{O.3.1})$$

where L is a radius characterizing the size of the pore and the center of the cylinder is located at $r = 0$ and $\epsilon > 0$, $\sigma > 0$. Some questions that one should answer before one starts programming are the following:

1. Is the potential for the interactions with the walls appropriate for a Molecular Dynamics simulation?
2. What is the volume of the pore as a function of the parameter L ?
3. What is the dimension of our problem? Do we have diffusion in 1, 2, or 3 dimensions?

In the first part of the project we study the diffusion in a smooth pore as defined by the above potential as a function of the pore diameter.

1. Compute the diffusion coefficient of a bulk Lennard-Jones liquid for $\rho = 0.6$ and $T = 2.0$ and 1.5 . Since the program uses an NVE ensemble, it is not possible to simulate at exactly the requested temperature. However, one can ensure to be close to this temperature by an appropriate equilibration of the system (this is also the case for the following two questions) during the first part of the MD simulation.
2. Compute the density as a function of the distance from the center of the pore for $\rho = 0.6$ and $T = 2.0$ and 1.5 and $L = 5$ and 2 . Interpret the results.
3. Compute the diffusion coefficient for $\rho = 0.6$ and $T = 2.0$ and $T = 1.5$ and $L = 5, 2$, and 1 . Interpret the results. The interpretation is not trivial.
4. The above calculations have been performed using the NVE ensemble. This implies that there is no coupling with the atoms of the walls. In a real system the walls are not smooth and can exchange heat with the adsorbed molecules. A possible way of modeling this is to assume that we have an Andersen thermostat in the boundary layer with the wall. Investigate how the results depend on the thickness of the boundary layer and the constant v of the Andersen algorithm.

The next step is to model the corrugation caused by the atoms. This corrugation could be a term:

$$U(z, r) = A \sin^2(\pi z / \sigma_w) \exp\left[-\left(\frac{r - L}{L_0}\right)^2\right], \quad (O.3.2)$$

where z is the distance to the wall, and σ_w is a term characterizing the size of the atoms of the wall and A is the strength of the interaction. The exponential is added to ensure that the potential is localized close to the walls of the cylinder. Investigate the diffusion coefficient as a function of the parameters σ_w and A both in the NVE and in the Andersen thermostat cases.

O.4 Multiple-time-step integrators

The time step in a Molecular Dynamics simulation strongly depends on the steepness of the potential energy surface. However, most potentials like the Lennard-Jones potential are steep at short distances. As short-range interactions can be computed very fast, it would be interesting to use a multiple-time-step integration algorithm, in which short-range (computationally cheap) interactions

are computed every time step and in which long-range (computationally expensive) interactions are evaluated every n time steps ($n > 1$). Recently, there has been a considerable effort to construct time-reversible multiple-time-step algorithms [117,126].

1. Why is it important to use time-reversible integration schemes in MD?
2. Modify Case Study 4 in such a way that pairwise interactions are calculated using a Verlet neighbor list. For every particle, a list is made of neighboring particles within a distance of $r_{\text{cut}} + \Delta$. All lists have to be updated only when the displacement of a single particle is larger than $\Delta/2$. Hint: The algorithm¹ in book of Allen and Tildesley [21] is a good starting point.
3. Investigate how the CPU time per time step depends on the size of Δ for various system sizes. Compare your results with Table 5.1 from ref. [21].
4. Modify the code in such a way that the NVE multiple-time-step algorithm of ref. [126] is used to integrate the equations of motion. You will have to use separate neighbor lists for the short-range and the long-range parts of the potential.
5. Why does one have to use a switching function in this algorithm? Why is it a good idea to use a linear interpolation scheme to compute the switching function from ref. [126]?
6. Make a detailed comparison between this algorithm and the standard Leap-Frog integrator (with the use of a neighbor list) at the same energy drift.

O.5 Thermodynamic integration

The difference in free energy between state A and state B can be calculated by thermodynamic integration:

$$F_A - F_B = \int_{\lambda=0}^{\lambda=1} d\lambda \left\langle \frac{\partial U(\lambda)}{\partial \lambda} \right\rangle_\lambda, \quad (\text{O.5.1})$$

in which $\lambda = 1$ in state A and $\lambda = 0$ in state B . In order to calculate the excess chemical potential of a Lennard-Jones system, we might use the following modified potential² [785]:

$$U(r, \lambda) = 4\epsilon \left(\lambda^5 \left(\frac{\sigma}{r} \right)^{12} - \lambda^3 \left(\frac{\sigma}{r} \right)^6 \right). \quad (\text{O.5.2})$$

Recall that the excess chemical potential is the difference in chemical potential between a real gas ($\lambda = 1$) and an ideal gas ($\lambda = 0$).

1. Make a plot of the modified LJ potential for various values of λ .

¹ https://github.com/Allen-Tildesley/examples/blob/master/md_lj_vl_module.f90.

² As the reader can easily verify, this modified potential is still a normal LJ 12-6 potential, but with different values of ϵ and σ .

2. Show that

$$\left(\frac{\partial^2 F}{\partial \lambda^2} \right) < 0 \quad (\text{O.5.3})$$

when

$$U = U_0 \lambda + U_1 (1 - \lambda). \quad (\text{O.5.4})$$

3. Derive Eq. (O.5.1).
4. Modify the code of Case Study 1 for this modified potential.
5. Perform the thermodynamic integration and compare your results with the conventional particle insertion method.
6. Calculate the chemical potential as a function of the density by scaling σ .

Appendix P

Hints for programming

The official rules for writing a program are that it should involve three persons. One for the design, one for the implementation, and one for the testing of it. Most probably you are doing all steps on your own! The reason why there is a need for three persons is that if you do not realize a certain aspect in the design phase, it is very likely that you do not think about this either while implementing or testing a program. The following tips may be useful in developing a program:

- If you are modifying a program, first try to understand every line of the starting code. If you do not understand the starting point, it is very dangerous to make modifications.
- Try to develop a modification plan consisting of the following steps:
 1. Make a copy of the program in a new directory.
 2. Describe in words why and how a certain part of the code needs to be modified.
 3. Start the programming by writing comment lines describing what the modifications are and make the modifications according to your written instructions. If you find during the implementations that the original ideas were not correct, describe this in your notes. This is to ensure that you think before you do. This may sound obvious and it is, but as you may find out it is a very difficult rule to stick to.
 4. Do not start modifying the entire program. Try to do it in steps. Try to test the modifications as soon as possible. For example, if the program is a Gibbs-ensemble simulation you probably have to modify the parts of the code that perform particle displacements, volume changes, and particle exchanges. However, there is no need to do all this at the same time. First modify the particle displacement and make all the tests you can (for example, energy conservation) without changing the code for the volume displacement and the particle exchange. Only then move to the next part of the code that must be changed.
 5. The first time you run the modified code, you might want to use array-bound checking (this option is not always used by default). Such checks will slow down the code, but are very useful for detecting errors. It is also useful to use code checkers.
 6. Try to find limiting cases for which your program should reproduce known results. For example, if you have written a program for a mixture, the results should be identical to the pure component results if all

interaction parameters are taken to be identical. Sometimes, the results of a calculation can be compared with exact results, e.g., in the limit where the system should behave as an ideal gas. Such a comparison is, of course, not always possible, but if it is possible it provides an important test.

7. For a standard (constant NVE) Molecular Dynamics program, it is a good idea to check the conservation of the total energy. In most systems, the total momentum should also be conserved (usually zero).
 8. In a Monte Carlo program, one has to calculate energy differences for each trial move. It is a good idea to write also some code for computing the total energy of the system by brute force, i.e., without using any sophisticated (time-saving) tricks. Compare the total energy change thus calculated during a simulation, with the change that is computed by adding all the energy changes during individual trial moves: the sum of all the partial energy changes should equal the difference between the initial and final energy computed by brute force to within the error expected on the basis of the precision of multiple additions.
 9. The results of a MC simulation might be dependent on the (quasi) random number generator that is used: there is no such thing as a true random number generator, but some are less bad than others. It is always a good idea to do a MC simulation with two different random number generators. See, for example, ref. [786] for a discussion about this topic.
- If you obtain numbers for the observables that you are computing, try to understand them and ask yourself whether they are reasonable. When using reduced units, most properties are of $\mathcal{O}(1)$, implying that if you find, say, -2.4 the number may be correct, but if you find 2.4×10^{18} , you should be suspicious.
 - If you are writing a parallel code, we strongly recommend that you make a detailed workplan before you start programming. A common mistake is that you forget to pass system variables (for example, temperature or masses of the particles) to all processors.

Bibliography

- [1] W.W. Wood, Early history of computer simulation in statistical mechanics, in: G. Ciccotti, W.G. Hoover (Eds.), Molecular Dynamics Simulations of Statistical Mechanics Systems, North Holland, Amsterdam, 1986, pp. 2–14.
- [2] G. Battimelli, G. Ciccotti, P. Greco, Computer Meets Theoretical Physics, Springer Nature, Cham Switzerland, 2020.
- [3] J.A. Prins, Onze voorstelling omrent de bouw van de stof, *Physica* 8 (1928) 257–268.
- [4] O.K. Rice, On the statistical mechanics of liquids, and the gas of hard elastic spheres, *J. Chem. Phys.* 12 (1) (1944) 1–18, <https://doi.org/10.1063/1.1723874>.
- [5] J.D. Bernal, The Bakerian lecture, 1962: the structure of liquids, *Proc. R. Soc. Lond. A* 280 (1964) 299–322, <https://doi.org/10.1098/rspa.1964.0147>.
- [6] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, *J. Chem. Phys.* 21 (6) (1953) 1087–1092, <https://doi.org/10.1063/1.1699114>.
- [7] N. Metropolis, The beginning of the Monte Carlo method, *Los Alamos Sci.* 12 (1987) 125–130, <http://permalink.lanl.gov/object/tr?what=info:lanl-repo/lareport/LA-UR-88-9067>.
- [8] E. Fermi, J.G. Pasta, S.M. Ulam, Studies of non-linear problems, Technical report, LASL Report LA-1940, 1955.
- [9] T. Dauxois, Fermi, Pasta, Ulam, and a mysterious lady, *Phys. Today* 61 (1) (2008) 55–57, <https://doi.org/10.1063/1.2835154>.
- [10] B.J. Alder, T.E. Wainwright, Molecular dynamics by electronic computers, in: I. Prigogine (Ed.), Proc. of the Int. Symp. on Statistical Mechanical Theory of Transport Processes, Brussels, 1956, Interscience Wiley, New York, 1958, pp. 97–131.
- [11] J.B. Gibson, A.N. Goland, M. Milgram, G.-H. Vineyard, Dynamics of radiation damage, *Phys. Rev.* 120 (4) (1960) 1229–1253, <https://doi.org/10.1103/PhysRev.120.1229>.
- [12] G.-H. Vineyard, Autobiographical remarks of G.-H. Vineyard, in: P.C. Gehlen, J.R. Beeler, R.I. Jaffe (Eds.), Interatomic Potentials and Simulation of Lattice Defects, Plenum, New York, 1972, pp. xiii–xvi.
- [13] A. Rahman, Correlations in motion of atoms in liquid argon, *Phys. Rev.* 136 (2A) (1964) A405, <https://doi.org/10.1103/PhysRev.136.A405>.
- [14] L. Verlet, Computer experiments on classical fluids I. Thermodynamical properties of Lennard-Jones molecules, *Phys. Rev.* 159 (1) (1967) 98, <https://doi.org/10.1103/PhysRev.159.98>.
- [15] J.A. Barker, R.O. Watts, Structure of water; a Monte Carlo calculation, *Chem. Phys. Lett.* 3 (3) (1969) 144–145, [https://doi.org/10.1016/0009-2614\(69\)80119-3](https://doi.org/10.1016/0009-2614(69)80119-3).
- [16] I.R. McDonald, K. Singer, Calculation of thermodynamic properties of liquid argon from Lennard-Jones parameters by a Monte Carlo method, *Discuss. Faraday Soc.* 43 (43) (1967) 40, <https://doi.org/10.1039/df9674300040>.
- [17] P.N. Vorontsov-Vel'yaminov, A.M. El'yashevich, A.K. Kron, Theoretical investigation of the thermodynamics properties of solutions of strong electrolytes by the Monte Carlo method, *Elektrokhimiya* 2 (1966) 708–716.

658 Bibliography

- [18] B.J. Alder, T.E. Wainwright, Phase transition for a hard sphere system, *J. Chem. Phys.* 27 (5) (1957) 1208–1209, <https://doi.org/10.1063/1.1743957>.
- [19] W.W. Wood, J.D. Jacobson, Preliminary results from a recalculation of the Monte Carlo equation of state of hard-spheres, *J. Chem. Phys.* 27 (5) (1957) 1207–1208, <https://doi.org/10.1063/1.1743956>.
- [20] P. Mehta, M. Bukov, C.H. Wang, A.G.R. Day, C. Richardson, C.K. Fisher, D.J. Schwab, A high-bias, low-variance introduction to machine learning for physicists, *Phys. Rep.* 810 (2019) 1–124, <https://doi.org/10.1016/j.physrep.2019.03.001>.
- [21] M.P. Allen, D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press, Oxford, 2017.
- [22] J.M. Haile, *Molecular Dynamics Simulations: Elementary Methods*, John Wiley and Sons, New York, 1992.
- [23] B. Leimkuhler, C. Matthews, *Molecular Dynamics: With Deterministic and Stochastic Numerical Methods*, 1st edition, Interdisciplinary Applied Mathematics, Springer International Publishing: Imprint: Springer, Cham, 2015.
- [24] M.E. Tuckerman, *Statistical Mechanics: Theory and Molecular Simulation*, Oxford University Press, Oxford; New York, 2010.
- [25] D.P. Landau, K. Binder, *A Guide to Monte Carlo Simulation in Statistical Physics*, Cambridge University Press, Cambridge, 2000.
- [26] D.C. Rapaport, *The Art of Molecular Dynamics Simulation*, 2nd edition, Cambridge University Press, Cambridge, UK; New York, NY, 2004.
- [27] M.E.J. Newman, G.T. Barkema, *Monte Carlo Methods in Statistical Physics*, Oxford University Press, Oxford, 1999.
- [28] R.W. Hockney, J.W. Eastwood, *Computer Simulations Using Particles*, McGraw-Hill, New York, 1981.
- [29] W.G. Hoover, *Molecular Dynamics*, Springer, Berlin, 1986.
- [30] W.G. Hoover, *Computational Physics Statistical Mechanics*, Elsevier, Amsterdam, 1991.
- [31] F.J. Vesely, *Computational Physics. An Introduction*, Plenum, New York, 1994.
- [32] D.W. Heermann, *Computer Simulation Methods in Theoretical Physics*, Springer, Berlin, 1990.
- [33] D.J. Evans, G.P. Morriss, *Statistical Mechanics of Non-Equilibrium Liquids*, Academic Press, London, 1990.
- [34] B. Peters, *Reaction Rate Theory and Rare Events*, Elsevier, Amsterdam; Cambridge, MA, 2017.
- [35] S.E. Koonin, *Computational Physics*, Benjamin/Cummings, Menlo Park, Calif., 1986.
- [36] H. Gould, J. Tobochnik, *Computer Simulation Methods*, vols. I and II, Addison-Wesley, Reading, Mass., 1988.
- [37] M.H. Kalos, P.A. Whitlock, *Monte Carlo Methods*, Wiley, New York, 1986.
- [38] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, 3rd edition, Cambridge University Press, Cambridge, 2007.
- [39] J.P. Valleau, S.G. Whittington, A guide to Monte Carlo simulations for statistical mechanics: 1. Highways, in: B.J. Berne (Ed.), *Statistical Mechanics*, Part A, Plenum, New York, 1977, pp. 137–168.
- [40] J.P. Valleau, S.G. Whittington, A guide to Monte Carlo simulations for statistical mechanics: 2. Byways, in: B.J. Berne (Ed.), *Statistical Mechanics*, Part A, Plenum, New York, 1977, pp. 169–194.
- [41] K. Binder, *Applications of the Monte Carlo Method in Statistical Physics*, Springer, Berlin, 1984.
- [42] K. Binder, *The Monte Carlo Method in Condensed Matter Physics*, Springer, Berlin, 1992.
- [43] O.G. Mouritsen, *Computer Studies of Phase Transitions and Critical Phenomena*, Springer, Berlin, 1984.
- [44] G. Ciccotti, W.G. Hoover, Molecular-dynamics simulations of statistical-mechanical systems, in: *Proceedings of the 97th International “Enrico Fermi” School of Physics*, North-Holland, Amsterdam, 1986.

- [45] M. Meyer, V. Pontikis, Proceedings of the NATO ASI on Computer Simulation in Materials Science, Kluwer, Dordrecht, 1991.
- [46] M.P. Allen, D.J. Tildesley, Proceedings of the NATO ASI on Computer Simulation in Chemical Physics, Kluwer, Dordrecht, 1993.
- [47] M. Baus, L.F. Rull, J.P. Ryckaert, Observation, Prediction and Simulation of Phase Transitions, Kluwer, Dordrecht, 1995.
- [48] B.J. Berne, G. Ciccotti, D.F. Coker, Classical and Quantum Dynamics in Condensed Phase Simulations, World Scientific, Singapore, 1998.
- [49] M. Ferrario, G. Ciccotti, K. Binder (Eds.), Computer Simulations in Condensed Matter Systems: from Materials to Chemical Biology, Lecture Notes in Physics, Springer, Berlin; New York, 2006.
- [50] N.G. van Kampen, The Gibbs paradox, in: W.E. Parry (Ed.), Essays in Theoretical Physics, Pergamon, 1984, pp. 303–312.
- [51] R.H. Swendsen, Gibbs' paradox and the definition of entropy, Entropy 10 (1) (2008) 15–18, <https://doi.org/10.3390/entropy-e10010015>.
- [52] D. Frenkel, Why colloidal systems can be described by statistical mechanics: some not very original comments on the Gibbs paradox, Mol. Phys. 112 (17) (2014) 2325–2329, <https://doi.org/10.1080/00268976.2014.904051>.
- [53] Linda E. Reichl, A Modern Course in Statistical Physics, 4th edition, Wiley-VCH Verlag, Weinheim, Germany, 2016.
- [54] Herbert Goldstein, Charles P. Poole, John L. Safko, Classical Mechanics, 3rd edition, Pearson Education Ltd, Harlow, 2014.
- [55] L. Onsager, Reciprocal relations in irreversible processes. I, Phys. Rev. 37 (4) (1931) 405–426, <https://doi.org/10.1103/PhysRev.37.405>.
- [56] L. Onsager, Reciprocal relations in irreversible processes. II, Phys. Rev. 38 (12) (1931) 2265–2279, <https://doi.org/10.1103/PhysRev.38.2265>.
- [57] S.R. de Groot, P. Mazur, Non-equilibrium Thermodynamics, Dover Publications, New York, 1984.
- [58] D. Chandler, An Introduction to Modern Statistical Mechanics, Oxford University Press, New York, 1987.
- [59] J.-P. Hansen, I.R. McDonald, Theory of Simple Liquids, 4th edition, Academic Press, Oxford, 2013.
- [60] J.G. Kirkwood, Statistical mechanics of fluid mixtures, J. Chem. Phys. 3 (5) (1935) 300–313, <https://doi.org/10.1063/1.1749657>.
- [61] J.G. Kirkwood, E.M. Boggs, The radial distribution function in liquids, J. Chem. Phys. 10 (6) (1942) 394–402, <https://doi.org/10.1063/1.1723737>.
- [62] R. Kubo, Statistical-mechanical theory of irreversible processes 1. General theory and simple applications to magnetic and conduction problems, J. Phys. Soc. Jpn. 12 (6) (1957) 570–586, <https://doi.org/10.1143/Jpsj.12.570>.
- [63] M. Thol, G. Rutkai, A. Köster, R. Lustig, R. Span, J. Vrabec, Equation of state for the Lennard-Jones fluid, J. Phys. Chem. Ref. Data 45 (2) (2016) 023101, <https://doi.org/10.1063/1.4945000>.
- [64] M.N. Rosenbluth, Genesis of the Monte Carlo Algorithm for Statistical Mechanics, in: J.E. Gubenatis (Ed.), AIP Conference Proceedings, vol. 690, American Institute of Physics, 2003, pp. 22–30.
- [65] H.L. Anderson, Scientific uses of the maniac, J. Stat. Phys. 43 (5–6) (1986) 731–748, <https://doi.org/10.1007/BF02628301>.
- [66] G.E.P. Box, M.E. Muller, A note on the generation of random normal deviates, Ann. Math. Stat. 29 (2) (1958) 610–611, <https://doi.org/10.1214/aoms/1177706645>.
- [67] N.G. van Kampen, Stochastic Processes in Physics and Chemistry, 3rd edition, North-Holland Personal Library, Elsevier, Amsterdam; Boston, 2007.
- [68] S. Schnabel, W. Janke, Accelerating polymer simulation by means of tree data-structures and a parsimonious Metropolis algorithm, Comput. Phys. Commun. 256 (2020), <https://doi.org/10.1016/j.cpc.2020.107414>.

660 Bibliography

- [69] W.W. Wood, F.R. Parker, Monte Carlo equation of state of molecular interactions with Lennard-Jones potential. I. A supercritical isotherm at about twice the critical temperature, *J. Chem. Phys.* 27 (3) (1957) 720–733, <https://doi.org/10.1063/1.1743822>.
- [70] M.J. Mandel, On the properties of a periodic fluid, *J. Stat. Phys.* 15 (4) (1976) 299–305, <https://doi.org/10.1007/BF01023055>.
- [71] J.E. Jones, On the determinations of molecular fields - 1 from the variation of the viscosity of a gas with temperature, *Proc. R. Soc. Lond. A* 106 (738) (1924) 441–462, <https://doi.org/10.1098/rspa.1924.0081>.
- [72] J.E. Lennard-Jones, Cohesion, *Proc. Phys. Soc.* 43 (1931) 461–482, <https://doi.org/10.1088/0959-5309/43/5/301>.
- [73] M.A. van der Hoef, P.A. Madden, Three-body dispersion contributions to the thermodynamic properties and effective pair interactions in liquid argon, *J. Chem. Phys.* 111 (4) (1999) 1520–1526, <https://doi.org/10.1063/1.479390>.
- [74] C. Valeriani, Z.J. Wang, D. Frenkel, Comparison of simple perturbation-theory estimates for the liquid-solid and the liquid-vapour interfacial free energies of Lennard-Jones systems, *Mol. Simul.* 33 (13) (2007) 1023–1028, <https://doi.org/10.1080/08927020701579352>.
- [75] K.M. Jablonka, D. Ongari, B. Smit, Applicability of tail corrections in the molecular simulations of porous materials, *J. Chem. Theory Comput.* 15 (10) (2019) 5635–5641, <https://doi.org/10.1021/acs.jctc.9b00586>.
- [76] J.G. Powles, The liquid-vapour coexistence line for Lennard-Jones-type fluids, *Physica* 126A (1–2) (1984) 289–299, [https://doi.org/10.1016/0378-4371\(84\)90156-0](https://doi.org/10.1016/0378-4371(84)90156-0).
- [77] B. Smit, D. Frenkel, Vapour-liquid equilibria of the two dimensional Lennard-Jones fluid(s), *J. Chem. Phys.* 94 (8) (1991) 5663–5668, <https://doi.org/10.1063/1.460477>.
- [78] B. Smit, Phase diagrams of Lennard-Jones fluids, *J. Chem. Phys.* 96 (11) (1992) 8639–8640, <https://doi.org/10.1063/1.462271>.
- [79] J.M. Kincaid, B. Hafskjold, Thermal-diffusion factors for the Lennard-Jones spline system, *Mol. Phys.* 82 (6) (1994) 1099–1114, <https://doi.org/10.1080/00268979400100784>.
- [80] B. Hafskjold, K.P. Travis, A.B. Hass, M. Hammer, A. Aasen, O. Wilhelmsen, Thermodynamic properties of the 3d Lennard-Jones/spline model, *Mol. Phys.* 117 (23–24) (2019) 3754–3769, <https://doi.org/10.1080/00268976.2019.1664780>.
- [81] X.P. Wang, S. Ramirez-Hinestrosa, J. Dobnikar, D. Frenkel, The Lennard-Jones potential: when (not) to use it, *Phys. Chem. Chem. Phys.* 22 (19) (2020) 10624–10633, <https://doi.org/10.1039/c9cp05445f>.
- [82] J.J. Nicolas, K.E. Gubbins, W.B. Streett, D.J. Tildesley, Equation of state for the Lennard-Jones fluid, *Mol. Phys.* 37 (5) (1979) 1429–1454, <https://doi.org/10.1080/0026897900101051>.
- [83] J.K. Johnson, J.A. Zollweg, K.E. Gubbins, The Lennard-Jones equation of state revisited, *Mol. Phys.* 78 (3) (1993) 591–618, <https://doi.org/10.1080/00268979300100411>.
- [84] S. Stephan, J. Staubach, H. Hasse, Review and comparison of equations of state for the Lennard-Jones fluid, *Fluid Phase Equilib.* 523 (2020), <https://doi.org/10.1016/j.fluid.2020.112772>.
- [85] Z.W. Salsburg, W.W. Wood, Equation of state of classical hard spheres at high density, *J. Chem. Phys.* 37 (4) (1962) 798–804, <https://doi.org/10.1063/1.1733163>.
- [86] M. Thol, G. Rutkai, R. Span, J. Vrabec, R. Lustig, Equation of state for the Lennard-Jones truncated and shifted model fluid, *Int. J. Thermophys.* 36 (1) (2015) 25–43, <https://doi.org/10.1007/s10765-014-1764-4>.
- [87] V.I. Manousiouthakis, M.W. Deem, Strict detailed balance is unnecessary in Monte Carlo simulation, *J. Chem. Phys.* 110 (6) (1999) 2753–2756, <https://doi.org/10.1063/1.477973>.
- [88] E.P. Bernard, W. Krauth, Event-chain Monte Carlo algorithms for hard-sphere systems (vol 80, 056704, 2009), *Phys. Rev. E* 86 (1) (2012), <https://doi.org/10.1103/PhysRevE.86.017701>.
- [89] M.F. Faulkner, L. Qin, A.C. Maggs, W. Krauth, All-atom computations with irreversible Markov chains, *J. Chem. Phys.* 149 (6) (2018), <https://doi.org/10.1063/1.5036638>.

- [90] M.A. Miller, L.M. Amon, W.P. Reinhardt, Should one adjust the maximum step size in a Metropolis Monte Carlo simulation?, *Chem. Phys. Lett.* 331 (2–4) (2000) 278–284, [https://doi.org/10.1016/S0009-2614\(00\)01217-3](https://doi.org/10.1016/S0009-2614(00)01217-3).
- [91] W.G.T. Kranendonk, D. Frenkel, Simulation of the adhesive-hard-sphere model, *Mol. Phys.* 64 (3) (1988) 403–424, <https://doi.org/10.1080/0026897800100303>.
- [92] A. Gelman, G.O. Roberts, W.R. Gilks, Efficient Metropolis jumping rules, in: J.M. Bernardo, J.O. Berger, A.P. Dawid, A.F.M. Smith (Eds.), *Bayesian Statistics*, vol. 5, Oxford University Press, Oxford, 1996, pp. 599–608.
- [93] G.O. Roberts, A. Gelman, W.R. Gilks, Weak convergence and optimal scaling of random walk Metropolis algorithms, *Ann. Appl. Probab.* 7 (1) (1997) 110–120, <https://doi.org/10.1214/aoap/1034625254>.
- [94] R.D. Mountain, D. Thirumalai, Quantitative measure of efficiency of Monte Carlo simulations, *Physica A* 210 (3–4) (1994) 453–460, [https://doi.org/10.1016/0378-4371\(94\)90092-2](https://doi.org/10.1016/0378-4371(94)90092-2).
- [95] F.J. Vesely, Angular Monte Carlo integration using quaternion parameters: a spherical reference potential for CCl₄, *J. Comput. Phys.* 47 (2) (1982) 291–296, [https://doi.org/10.1016/0021-9991\(82\)90080-8](https://doi.org/10.1016/0021-9991(82)90080-8).
- [96] G. Ciccotti, J.P. Ryckaert, Molecular-dynamics simulation of rigid molecules, *Comput. Phys. Rep.* 4 (6) (1986) 345–392, [https://doi.org/10.1016/0167-7977\(86\)90022-5](https://doi.org/10.1016/0167-7977(86)90022-5).
- [97] M. Fixman, Classical statistical mechanics of constraints: a theorem and application to polymers, *Proc. Natl. Acad. Sci. USA* 71 (8) (1974) 3050–3053, <https://doi.org/10.1073/pnas.71.8.3050>.
- [98] N.G. Van Kampen, Statistical-mechanics of trimers, *Appl. Sci. Res.* 37 (1–2) (1981) 67–75, <https://doi.org/10.1007/BF00382618>.
- [99] L.R. Dodd, T.D. Boone, D.N. Theodorou, A concerted rotation algorithm for atomistic Monte Carlo simulation of polymer melts and glasses, *Mol. Phys.* 78 (4) (1993) 961–996, <https://doi.org/10.1080/00268979300100641>.
- [100] N.H. Tsai, F.F. Abraham, G.M. Pound, Structure and thermodynamics of binary microclusters - Monte-Carlo simulation, *Surf. Sci.* 77 (3) (1978) 465–492, [https://doi.org/10.1016/0039-6028\(78\)90134-6](https://doi.org/10.1016/0039-6028(78)90134-6).
- [101] W.G.T. Kranendonk, D. Frenkel, Computer-simulation of solid liquid coexistence in binary hard-sphere mixtures, *Mol. Phys.* 72 (3) (1991) 679–697, <https://doi.org/10.1080/00268979100100501>.
- [102] L. Berthier, D. Coslovich, A. Ninarello, M. Ozawa, Equilibrium sampling of hard spheres up to the jamming density and beyond, *Phys. Rev. Lett.* 116 (23) (2016), <https://doi.org/10.1103/PhysRevLett.116.238002>.
- [103] D.J. Lee, M.M. Telo Da Gama, K.E. Gubbins, The vapor-liquid interface for a Lennard-Jones model of argon krypton mixtures, *Mol. Phys.* 53 (5) (1984) 1113–1130, <https://doi.org/10.1080/00268978400102891>.
- [104] W. Humphrey, A. Dalke, K. Schulten, VMD: visual molecular dynamics, *J. Mol. Graph. Model.* 14 (1) (1996) 33–38, [https://doi.org/10.1016/0263-7855\(96\)00018-5](https://doi.org/10.1016/0263-7855(96)00018-5).
- [105] H.J.C. Berendsen, *Simulating the Physical World: Hierarchical Modeling from Quantum Mechanics to Fluid Dynamics*, Cambridge University Press, Cambridge; New York, 2007.
- [106] J.L. Lebowitz, J.K. Percus, L. Verlet, Ensemble dependence of fluctuations with application to machine computations, *Phys. Rev.* 153 (1) (1967) 250–254, <https://doi.org/10.1103/PhysRev.153.250>.
- [107] H.L. Xu, R. Cabriolu, B. Smit, Effects of degrees of freedom on calculating diffusion properties in nanoporous materials, *J. Chem. Theory Comput.* 18 (5) (2022) 2826–2835, <https://doi.org/10.1021/acs.jctc.2c00094>.
- [108] R.D. Neidinger, Introduction to automatic differentiation and Matlab object-oriented programming, *SIAM Rev.* 52 (3) (2010) 545–563, <https://doi.org/10.1137/080743627>.
- [109] G.D. Quinlan, S. Tremaine, On the reliability of gravitational n-body integrations, *Mon. Not. R. Astron. Soc.* 259 (3) (1992) 505–518, <https://doi.org/10.1093/mnras/259.3.505>.

662 Bibliography

- [110] R.E. Gillilan, K.R. Wilson, Shadowing, rare events, and rubber bands - a variational Verlet algorithm for molecular-dynamics, *J. Chem. Phys.* 97 (3) (1992) 1757–1772, <https://doi.org/10.1063/1.463163>.
- [111] S. Toxvaerd, Hamiltonians for discrete dynamics, *Phys. Rev. E* 50 (3) (1994) 2271–2274, <https://doi.org/10.1103/PhysRevE.50.2271>.
- [112] R.D. Skeel, What makes molecular dynamics work?, *SIAM J. Sci. Comput.* 31 (2) (2009) 1363–1378, <https://doi.org/10.1137/070683660>.
- [113] W.C. Swope, Hans C. Andersen, P.H. Berens, K.R. Wilson, A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: application to small water clusters, *J. Chem. Phys.* 76 (1) (1982) 637–649, <https://doi.org/10.1063/1.442716>.
- [114] J. Gans, D. Shalloway, Shadow mass and the relationship between velocity and momentum in symplectic numerical integration, *Phys. Rev. E* 61 (4) (2000) 4587–4592, <https://doi.org/10.1103/PhysRevE.61.4587>.
- [115] M.P. Eastwood, K.A. Stafford, R.A. Lippert, M.O. Jensen, P. Maragakis, C. Predescu, R.O. Dror, D.E. Shaw, Equipartition and the calculation of temperature in biomolecular simulations, *J. Chem. Theory Comput.* 6 (7) (2010) 2045–2058, <https://doi.org/10.1021/ct9002916>.
- [116] H.J.C. Berendsen, W.F. Van Gunsteren, Practical algorithms for dynamics simulations, in: G. Ciccotti, W.G. Hoover (Eds.), *Molecular Dynamics Simulations of Statistical Mechanics Systems*, North-Holland, Amsterdam, 1986, pp. 43–65.
- [117] M.E. Tuckerman, B.J. Berne, G.J. Martyna, Reversible multiple time scale molecular-dynamics, *J. Chem. Phys.* 97 (3) (1992) 1990–2001, <https://doi.org/10.1063/1.463137>.
- [118] J.C. Sexton, D.H. Weingarten, Hamiltonian evolution for the hybrid Monte Carlo algorithm, *Nucl. Phys. B* 380 (3) (1992) 665–677, [https://doi.org/10.1016/0550-3213\(92\)90263-B](https://doi.org/10.1016/0550-3213(92)90263-B).
- [119] H. Yoshida, Symplectic integrators for Hamiltonian systems: basic theory, in: S. Ferraz-Mello (Ed.), *Chaos, Resonance and Collective Dynamical Phenomena in the Solar System*, Kluwer Academic Publishers, Dordrecht, 1992, pp. 407–411.
- [120] H. Yoshida, Recent progress in the theory and application of symplectic integrators, *Celest. Mech. Dyn. Astron.* 56 (1993) 27–43, https://doi.org/10.1007/978-94-011-2030-2_3.
- [121] P. Saha, S. Tremaine, Symplectic integrators for solar system dynamics, *Astron. J.* 104 (4) (1992) 1633–1640, <https://doi.org/10.1086/116347>.
- [122] K.D. Hammonds, D.M. Heyes, Shadow Hamiltonian in classical nve molecular dynamics simulations: a path to long time stability, *J. Chem. Phys.* 152 (2) (2020), <https://doi.org/10.1063/1.5139708>.
- [123] R. Olender, R. Elber, Calculation of classical trajectories with a very large time step: formalism and numerical examples, *J. Chem. Phys.* 105 (20) (1996) 9299–9315, <https://doi.org/10.1063/1.472727>.
- [124] R. Elber, J. Meller, R. Olender, Stochastic path approach to compute atomically detailed trajectories: application to the folding of c peptide, *J. Phys. Chem. B* 103 (6) (1999) 899–911, <https://doi.org/10.1021/jp983774z>.
- [125] L. Onsager, S. Machlup, Fluctuations and irreversible processes, *Phys. Rev.* 91 (6) (1953) 1505–1512, <https://doi.org/10.1103/PhysRev.91.1505>.
- [126] G.J. Martyna, M.E. Tuckerman, D.J. Tobias, M.L. Klein, Explicit reversible integrators for extended systems dynamics, *Mol. Phys.* 87 (5) (1996) 1117–1157, <https://doi.org/10.1080/00268979600100761>.
- [127] J. Naghizadeh, S.A. Rice, Kinetic theory of dense fluids. X. measurement and interpretation of self-diffusion in liquid Ar, Kr, Xe and CH₄, *J. Chem. Phys.* 36 (10) (1962) 2710–2720, <https://doi.org/10.1063/1.1732357>.
- [128] H.N.W. Lekkerkerker, R. Tuinier, Depletion interaction, in: *Colloids and the Depletion Interaction*, Springer, Netherlands, Dordrecht, 2011, pp. 57–108.
- [129] T.R. Josephson, R. Singh, M.S. Minkara, E.O. Fetisov, J.I. Siepmann, Partial molar properties from molecular simulation using multiple linear regression, *Mol. Phys.* 117 (23–24) (2019) 3589–3602, <https://doi.org/10.1080/00268976.2019.1648898>.

- [130] M.J. Louwerse, E.J. Baerends, Calculation of pressure in case of periodic boundary conditions, *Chem. Phys. Lett.* 421 (1–3) (2006) 138–141, <https://doi.org/10.1016/j.cplett.2006.01.087>.
- [131] A.P. Thompson, S.J. Plimpton, W. Mattson, General formulation of pressure and stress tensor for arbitrary many-body interaction potentials under periodic boundary conditions, *J. Chem. Phys.* 131 (15) (2009), <https://doi.org/10.1063/1.3245303>.
- [132] B.D. Todd, D.J. Evans, P.J. Daivis, Pressure tensor for inhomogeneous fluids, *Phys. Rev. E* 52 (2) (1995) 1627–1638, <https://doi.org/10.1103/PhysRevE.52.1627>.
- [133] R. Eppinga, D. Frenkel, Monte Carlo study of the isotropic and nematic phases of infinitely thin hard platelets, *Mol. Phys.* 52 (6) (1984) 1303–1334, <https://doi.org/10.1080/00268978400101951>.
- [134] A. Ghoufi, P. Malfreyt, D.J. Tildesley, Computer modelling of the surface tension of the gas-liquid and liquid-liquid interface, *Chem. Soc. Rev.* 45 (5) (2016) 1387–1409, <https://doi.org/10.1039/c5cs00736d>.
- [135] Kaihang Shi, Edward R. Smith, Erik E. Santiso, Keith E. Gubbins, A perspective on the microscopic pressure (stress) tensor: history, current understanding, and future challenges, *J. Chem. Phys.* 158 (4) (2023) 040901, <https://doi.org/10.1063/5.0132487>.
- [136] P. Schofield, J.R. Henderson, Statistical-mechanics of inhomogeneous fluids, *Proc. R. Soc. Lond. A* 379 (1776) (1982) 231–246, <https://doi.org/10.1098/rspa.1982.0015>.
- [137] G.J. Gloor, G. Jackson, F.J. Blas, E. de Miguel, Test-area simulation method for the direct determination of the interfacial tension of systems with continuous or discontinuous potentials, *J. Chem. Phys.* 123 (13) (2005), <https://doi.org/10.1063/1.2038827>.
- [138] J.G. Sampayo, A. Malijevsky, E.A. Muller, E. de Miguel, G. Jackson, Communications: evidence for the role of fluctuations in the thermodynamics of nanoscale drops and the implications in computations of the surface tension, *J. Chem. Phys.* 132 (14) (2010), <https://doi.org/10.1063/1.3376612>.
- [139] J.W. Cahn, Surface stress and the chemical-equilibrium of small crystals. 1. The case of the isotropic surface, *Acta Metall. Mater.* 28 (10) (1980) 1333–1338, [https://doi.org/10.1016/0001-6160\(80\)90002-4](https://doi.org/10.1016/0001-6160(80)90002-4).
- [140] A. Cacciuto, S. Auer, D. Frenkel, Breakdown of classical nucleation theory near isostructural phase transitions, *Phys. Rev. Lett.* 93 (16) (2004), <https://doi.org/10.1103/PhysRevLett.93.166105>.
- [141] F. Leroy, F. Muller-Plathe, Solid-liquid surface free energy of Lennard-Jones liquid on smooth and rough surfaces computed by molecular dynamics using the phantom-wall method, *J. Chem. Phys.* 133 (4) (2010), <https://doi.org/10.1063/1.3458796>.
- [142] F. Leroy, F. Muller-Plathe, Dry-surface simulation method for the determination of the work of adhesion of solid-liquid interfaces, *Langmuir* 31 (30) (2015) 8335–8345, <https://doi.org/10.1021/acs.langmuir.5b01394>.
- [143] J.S. Rowlinson, B. Widom, *Molecular Theory of Capillarity*, Oxford University Press, Oxford, 1982.
- [144] D. Borgis, R. Assaraf, B. Rotenberg, R. Vuilleumier, Computation of pair distribution functions and three-dimensional densities with a reduced variance principle, *Mol. Phys.* 111 (22–23) (2013) 3486–3492, <https://doi.org/10.1080/00268976.2013.838316>.
- [145] B. Rotenberg, Use the force! Reduced variance estimators for densities, radial distribution functions, and local mobilities in molecular simulations, *J. Chem. Phys.* 153 (15) (2020), <https://doi.org/10.1063/5.0029113>.
- [146] D. de las Heras, M. Schmidt, Better than counting: density profiles from force sampling, *Phys. Rev. Lett.* 120 (21) (2018), <https://doi.org/10.1103/PhysRevLett.120.218001>.
- [147] K.A. Colwell, M.N. Jackson, R.M. Torres-Gavosto, S. Jawahery, B. Vlaisavljevich, J.M. Falkowski, B. Smit, S.C. Weston, J.R. Long, Buffered coordination modulation as a means of controlling crystal morphology and molecular diffusion in an anisotropic metal-organic framework, *J. Am. Chem. Soc.* 143 (13) (2021) 5044–5052, <https://doi.org/10.1021/jacs.1c00136>.

664 Bibliography

- [148] H. Flyvbjerg, H.G. Petersen, Error estimates on averages of correlated data, *J. Chem. Phys.* 91 (1) (1989) 461–466, <https://doi.org/10.1063/1.457480>.
- [149] W.S. Price, Pulsed-field gradient nuclear magnetic resonance as a tool for studying translational diffusion 1. Basic theory, *Concepts Magn. Reson.* 9 (5) (1997) 299–336, [https://doi.org/10.1002/\(SICI\)1099-0534\(1997\)9:5<299::Aid-Cmr2>3.3.Co;2-2](https://doi.org/10.1002/(SICI)1099-0534(1997)9:5<299::Aid-Cmr2>3.3.Co;2-2).
- [150] B. Dunweg, K. Kremer, Molecular-dynamics simulation of a polymer-chain in solution, *J. Chem. Phys.* 99 (9) (1993) 6983–6997, <https://doi.org/10.1063/1.465445>.
- [151] I.C. Yeh, G. Hummer, System-size dependence of diffusion coefficients and viscosities from molecular dynamics simulations with periodic boundary conditions, *J. Phys. Chem. B* 108 (40) (2004) 15873–15879, <https://doi.org/10.1021/jp0477147>.
- [152] D.M. Heyes, M.J. Cass, J.G. Powles, W.A.B. Evans, Self-diffusion coefficient of the hard-sphere fluid: system size dependence and empirical correlations, *J. Phys. Chem. B* 111 (6) (2007) 1455–1464, <https://doi.org/10.1021/jp067373s>.
- [153] J. Kärger, D.M. Ruthven, *Diffusion in Zeolites and Other Microporous Solids*, Wiley, New York, 1992.
- [154] H. Jobic, J. Karger, M. Bee, Simultaneous measurement of self- and transport diffusivities in zeolites, *Phys. Rev. Lett.* 82 (21) (1999) 4260–4263, <https://doi.org/10.1103/PhysRevLett.82.4260>.
- [155] A. Marcolongo, P. Umari, S. Baroni, Microscopic theory and quantum simulation of atomic heat transport, *Nat. Phys.* 12 (1) (2016) 80–U111, <https://doi.org/10.1038/Nphys3509>.
- [156] F. Müller Plathe, A simple nonequilibrium molecular dynamics method for calculating the thermal conductivity, *J. Chem. Phys.* 106 (14) (1997) 6082–6085, <https://doi.org/10.1063/1.473271>.
- [157] T. Ikeshoji, B. Hafskjold, Nonequilibrium molecular-dynamics calculation of heat-conduction in liquid and through liquid-gas interface, *Mol. Phys.* 81 (2) (1994) 251–261, <https://doi.org/10.1080/00268979400100171>.
- [158] P. Wirnsberger, D. Frenkel, C. Dellago, An enhanced version of the heat exchange algorithm with excellent energy conservation properties, *J. Chem. Phys.* 143 (12) (2015), <https://doi.org/10.1063/1.4931597>.
- [159] S.T. Cui, P.T. Cummings, H.D. Cochran, The calculation of the viscosity from the auto-correlation function using molecular and atomic stress tensors, *Mol. Phys.* 88 (6) (1996) 1657–1664, <https://doi.org/10.1080/00268979650025696>.
- [160] A. David, A. De Nicola, U. Tartaglino, G. Milano, G. Raos, Viscoelasticity of short polymer liquids from atomistic simulations, *J. Electrochem. Soc.* 166 (9) (2019) B3246–B3256, <https://doi.org/10.1149/2.0371909jes>.
- [161] F. Grasselli, S. Baroni, Invariance principles in the theory and computation of transport coefficients, *Eur. Phys. J. B* 94 (8) (2021), <https://doi.org/10.1140/epjb/s10051-021-00152-5>.
- [162] G. Ciccotti, G. Jacucci, I.R. McDonald, Thought-experiments by molecular-dynamics, *J. Stat. Phys.* 21 (1) (1979) 1–22, <https://doi.org/10.1007/Bf01011477>.
- [163] K. Binder, Finite size scaling analysis of Ising-model block distribution-functions, *Z. Phys. B, Condens. Matter* 43 (2) (1981) 119–140, <https://doi.org/10.1007/Bf01293604>.
- [164] R. Zwanzig, N.K. Ailawadi, Statistical error due to finite time averaging in computer experiments, *Phys. Rev.* 182 (1) (1969) 280–283, <https://doi.org/10.1103/PhysRev.182.280>.
- [165] R.E. Jones, K.K. Mandadapu, Adaptive Green-Kubo estimates of transport coefficients from molecular dynamics based on robust error analysis, *J. Chem. Phys.* 136 (15) (2012), <https://doi.org/10.1063/1.3700344>.
- [166] J.J. Erpenbeck, W.W. Wood, Molecular dynamics techniques for hard-core systems, in: B.J. Berne (Ed.), *Statistical Mechanics*, Part B, Plenum, New York, 1977, pp. 1–40.
- [167] W.W. Wood, Monte Carlo calculations for hard disks in the isothermal-isobaric ensemble, *J. Chem. Phys.* 48 (1968) 415–434, <https://doi.org/10.1063/1.1667938>.
- [168] I.R. McDonald, Npt-ensemble Monte-Carlo calculations for binary liquid mixtures, *Mol. Phys.* 23 (1) (1972) 41–58, <https://doi.org/10.1080/00268977200100031>.

- [169] R. Najafabadi, S. Yip, Observation of finite-temperature strain transformation (f.c.c. \leftrightarrow b.c.c.) in Monte Carlo simulation of iron, *Scr. Metall.* 17 (1983) 1199–1204, [https://doi.org/10.1016/0036-9748\(83\)90283-1](https://doi.org/10.1016/0036-9748(83)90283-1).
- [170] G.E. Norman, V.S. Filinov, Investigation of phase transitions by a Monte-Carlo method, *High Temp. (USSR)* 7 (2) (1969) 216–222.
- [171] D.J. Adams, Chemical potential of hard-sphere fluids by Monte-Carlo methods, *Mol. Phys.* 28 (5) (1974) 1241–1252, <https://doi.org/10.1080/00268977400102551>.
- [172] M. Creutz, Microcanonical Monte Carlo simulation, *Phys. Rev. Lett.* 50 (1983) 1411–1414, <https://doi.org/10.1103/PhysRevLett.50.1411>.
- [173] P. Attard, On the density of volume states in the isobaric ensemble, *J. Chem. Phys.* 103 (22) (1995) 9884–9885, <https://doi.org/10.1063/1.469956>.
- [174] G.J.M. Koper, H. Reiss, Length scale for the constant pressure ensemble: application to small systems and relation to Einstein fluctuation theory, *J. Phys. Chem.* 100 (1) (1996) 422–432, <https://doi.org/10.1021/jp951819f>.
- [175] M. Rovere, D.W. Hermann, K. Binder, Block density distribution function analyses of two-dimensional Lennard-Jones fluids, *Europhys. Lett.* 6 (7) (1988) 585–590, <https://doi.org/10.1209/0295-5075/6/7/003>.
- [176] N.B. Wilding, A.D. Bruce, Density-fluctuations and field mixing in the critical fluid, *J. Phys. Condens. Matter* 4 (12) (1992) 3087–3108, <https://doi.org/10.1088/0953-8984/4/12/008>.
- [177] M. Rovere, P. Nielaba, K. Binder, Simulation studies of gas-liquid transitions in two dimensions via subsystem-block-density distribution analysis, *Z. Phys.* 90 (1993) 215–228, <https://doi.org/10.1007/BF02198158>.
- [178] M. Parrinello, A. Rahman, Crystal-structure and pair potentials - a molecular-dynamics study, *Phys. Rev. Lett.* 45 (14) (1980) 1196–1199, <https://doi.org/10.1103/PhysRevLett.45.1196>.
- [179] M. Parrinello, A. Rahman, Polymorphic transitions in single-crystals - a new molecular-dynamics method, *J. Appl. Phys.* 52 (12) (1981) 7182–7190, <https://doi.org/10.1063/1.328693>.
- [180] Hans C. Andersen, Molecular dynamics simulations at constant pressure and/or temperature, *J. Chem. Phys.* 72 (4) (1980) 2384–2393, <https://doi.org/10.1063/1.439486>.
- [181] J.R. Ray, A. Rahman, Statistical ensembles and molecular dynamics studies of anisotropic solids, *J. Chem. Phys.* 80 (9) (1984) 4423–4428, <https://doi.org/10.1063/1.447221>.
- [182] D.C. Wallace, Thermodynamic theory of stressed crystals and higher-order elastic constants, in: H. Ehrenreich, F. Seitz, D. Turnbull (Eds.), *Solid State Physics: Advances in Research and Applications*, Academic Press, New York, 1970, pp. 301–404.
- [183] D.J. Adams, Grand canonical ensemble Monte-Carlo for a Lennard-Jones fluid, *Mol. Phys.* 29 (1) (1975) 307–311, <https://doi.org/10.1080/00268977500100221>.
- [184] D.J. Adams, Calculating the low temperature vapour line by Monte Carlo, *Mol. Phys.* 32 (3) (1976) 647–657, <https://doi.org/10.1080/00268977600102101>.
- [185] D.J. Adams, Calculating the high-temperature vapor line by Monte-Carlo, *Mol. Phys.* 37 (1) (1979) 211–221, <https://doi.org/10.1080/00268977900100171>.
- [186] L.A. Rowley, D. Nicholson, N.G. Parsonage, Monte Carlo grand canonical ensemble calculation in a gas-liquid transition region for 12-6 argon, *J. Comput. Phys.* 17 (4) (1975) 401–414, [https://doi.org/10.1016/0021-9991\(75\)90042-X](https://doi.org/10.1016/0021-9991(75)90042-X).
- [187] J. Yao, R.A. Greenkorn, K.C. Chao, Monte Carlo simulation of the grand canonical ensemble, *Mol. Phys.* 46 (3) (1982) 587–594, <https://doi.org/10.1080/00268978200101411>.
- [188] M. Mezei, A cavity-biased (T,V,μ) Monte Carlo method for the computer simulation of fluids, *Mol. Phys.* 40 (1980) 901–906, <https://doi.org/10.1080/00268978000101971>.
- [189] J.P. Valleau, L.K. Cohen, Primitive model electrolytes. I. Grand canonical Monte Carlo computations, *J. Chem. Phys.* 72 (11) (1980) 5935–5941, <https://doi.org/10.1063/1.439092>.
- [190] W. van Megen, I.K. Snook, The grand canonical ensemble Monte Carlo method applied to the electrical double layer, *J. Chem. Phys.* 73 (1980) 4656–4662, <https://doi.org/10.1063/1.440657>.

666 Bibliography

- [191] H.J.F. Stroud, E. Richards, P. Limcharoen, N.G. Parsonage, Thermodynamic study of the Linde 5A + methane system, *J. Chem. Soc. Faraday Trans. I* 72 (1976) 942–954, <https://doi.org/10.1039/F19767200942>.
- [192] C.R.A. Catlow, Modelling of Structure and Reactivity in Zeolites, Academic Press, London, 1992.
- [193] B. Smit, T.L.M. Maesen, Molecular simulations of zeolites: adsorption, diffusion, and shape selectivity, *Chem. Rev.* 108 (10) (2008) 4125–4184, <https://doi.org/10.1021/cr8002642>.
- [194] L.D. Gelb, K.E. Gubbins, R. Radhkrishnan, M. Sliwinska-Bartkowiak, Phase separation in confined systems, *Rep. Prog. Phys.* 62 (12) (1999) 1573–1659, <https://doi.org/10.1088/0034-4885/62/12/201>.
- [195] J.L. Soto, A.L. Myers, Monte Carlo studies of adsorption in molecular sieves, *Mol. Phys.* 42 (4) (1981) 971–983, <https://doi.org/10.1080/00268978100100761>.
- [196] G.B. Wood, J.S. Rowlinson, Computer simulations of fluids in zeolite X and Y, *J. Chem. Soc. Faraday Trans. 2* (85) (1989) 765–781, <https://doi.org/10.1039/F29898500765>.
- [197] S.J. Goodbody, K. Watanabe, D. Macgowan, J.P.R.B. Walton, N. Quirke, Molecular simulation of methane and butane in silicalite, *J. Chem. Soc. Faraday Trans. 87* (13) (1991) 1951–1958, <https://doi.org/10.1039/ft9918701951>.
- [198] R.Q. Snurr, R.L. June, A.T. Bell, D.N. Theodorou, Molecular simulations of methane adsorption in silicalite, *Mol. Simul.* 8 (1–2) (1991) 73–92, <https://doi.org/10.1080/08927029108022468>.
- [199] F. Karavias, A.L. Myers, Isosteric heat of multicomponent adsorption: thermodynamics and computer simulations, *Langmuir* 7 (12) (1991) 3118–3126, <https://doi.org/10.1021/la00060a035>.
- [200] P.R. van Tassel, H.T. Davis, A.V. McCormick, Open-system Monte Carlo simulations of Xe in NaA, *J. Chem. Phys.* 98 (1993) 8919–8929, <https://doi.org/10.1063/1.464451>.
- [201] M.W. Maddox, J.S. Rowlinson, Computer simulation of the adsorption of a fluid mixture in zeolite Y, *J. Chem. Soc. Faraday Trans. 89* (19) (1993) 3619–3621, <https://doi.org/10.1039/ft9938903619>.
- [202] B. Smit, Simulating the adsorption isotherms of methane, ethane, and propane in the zeolite silicalite, *J. Phys. Chem.* 99 (15) (1995) 5597–5603, <https://doi.org/10.1021/j100015a050>.
- [203] D.A. Kofke, E.D. Glandt, Monte-Carlo simulation of multicomponent equilibria in a semigrand canonical ensemble, *Mol. Phys.* 64 (6) (1988) 1105–1131, <https://doi.org/10.1080/00268978800100743>.
- [204] W.G.T. Kranendonk, D. Frenkel, Thermodynamic properties of binary hard-sphere mixtures, *Mol. Phys.* 72 (3) (1991) 715–733, <https://doi.org/10.1080/00268979100100521>.
- [205] J.K. Johnson, A.Z. Panagiotopoulos, K.E. Gubbins, Reactive canonical Monte Carlo: a new simulation technique for reacting or associating fluids, *Mol. Phys.* 81 (3) (1994) 717–733, <https://doi.org/10.1080/00268979400100481>.
- [206] W.R. Smith, B. Triska, The reaction ensemble method for the computer simulation of chemical and phase equilibria. I. Theory and basic examples, *J. Chem. Phys.* 100 (4) (1994) 3019–3027, <https://doi.org/10.1063/1.466443>.
- [207] D.W. Yin, M.O. de la Cruz, J.J. de Pablo, Swelling and collapse of polyelectrolyte gels in equilibrium with monovalent and divalent electrolyte solutions, *J. Chem. Phys.* 131 (19) (2009), <https://doi.org/10.1063/1.3264950>.
- [208] E.J. Meijer, D. Frenkel, Computer-simulation of polymer-induced clustering of colloids, *Phys. Rev. Lett.* 67 (9) (1991) 1110–1113, <https://doi.org/10.1103/PhysRevLett.67.1110>.
- [209] P.N. Pusey, Colloidal suspensions, in: J.P. Hansen, D. Levesque, J. Zinn-Justin (Eds.), Liquids, Freezing and Glass Transition: Les Houches Summer Schools of Theoretical Physics Session LI (1989), Les Houches Summer Schools of Theoretical Physics, Elsevier Science Publ BV, Amsterdam, 1991, pp. 763–942.
- [210] P.G. Bolhuis, D.A. Kofke, Monte Carlo study of freezing of polydisperse hard spheres, *Phys. Rev. E* 54 (1) (1996) 634–643, <https://doi.org/10.1103/physreve.54.634>.

- [211] A.Z. Panagiotopoulos, Direct determination of phase coexistence properties of fluids by Monte Carlo simulation in a new ensemble, *Mol. Phys.* 61 (4) (1987) 813–826, <https://doi.org/10.1080/00268978700101491>.
- [212] N.B. Wilding, Critical-point and coexistence-curve properties of the Lennard-Jones fluid - a finite-size-scaling study, *Phys. Rev. E* 52 (1) (1995) 602–611, <https://doi.org/10.1103/PhysRevE.52.602>.
- [213] A.Z. Panagiotopoulos, N. Quirke, M.R. Stapleton, D.J. Tildesley, Phase equilibria by simulations in the Gibbs ensemble: alternative derivation, generalization and application to mixtures and membrane equilibria, *Mol. Phys.* 63 (4) (1988) 527–545, <https://doi.org/10.1080/00268978800100361>.
- [214] A.Z. Panagiotopoulos, Adsorption and capillary condensation of fluids in cylindrical pores by Monte-Carlo simulation in the Gibbs ensemble, *Mol. Phys.* 62 (3) (1987) 701–719, <https://doi.org/10.1080/00268978700102501>.
- [215] A.Z. Panagiotopoulos, Direct determination of fluid phase equilibria by simulation in the Gibbs ensemble: a review, *Mol. Simul.* 9 (1) (1992) 1–23, <https://doi.org/10.1080/08927029208048258>.
- [216] D. Frenkel, Monte Carlo simulations, in: C.R.A. Catlow, S.C. Parker, M.P. Allen (Eds.), *Computer Modelling of Fluids, Polymers and Solids*, in: NATO ASI Series, vol. 293, Kluwer, Dordrecht, 1990, pp. 83–123.
- [217] B. Smit, D. Frenkel, Calculation of the chemical potential in the Gibbs ensemble, *Mol. Phys.* 68 (4) (1989) 951–958, <https://doi.org/10.1080/00268978900102651>.
- [218] L.F. Rull, G. Jackson, B. Smit, The condition of microscopic reversibility in the Gibbs ensemble Monte Carlo simulations of phase equilibria, *Mol. Phys.* 85 (3) (1995) 435–447, <https://doi.org/10.1080/00268979500101231>.
- [219] H. Watanabe, N. Ito, C.K. Hu, Phase diagram and universality of the Lennard-Jones gas-liquid system, *J. Chem. Phys.* 136 (20) (2012), <https://doi.org/10.1063/1.4720089>.
- [220] B. Smit, Ph. de Smedt, D. Frenkel, Computer simulations in the Gibbs ensemble, *Mol. Phys.* 68 (4) (1989) 931–950, <https://doi.org/10.1080/00268978900102641>.
- [221] J.S. Rowlinson, F.L. Swinton, *Liquids and Liquid Mixtures*, Butterworths, London, 1982.
- [222] A.Z. Panagiotopoulos, Molecular simulations of phase coexistence: finite-size effects and the determination of critical parameters for two- and three dimensional Lennard-Jones fluids, *Int. J. Thermophys.* 15 (6) (1994) 1057–1072, <https://doi.org/10.1007/BF01458815>.
- [223] M. Rovere, D.W. Hermann, K. Binder, The gas-liquid transition of the two-dimensional Lennard-Jones fluid, *J. Phys. Condens. Matter* 2 (1990) 7009–7032, <https://doi.org/10.1088/0953-8984/2/33/013>.
- [224] B.M. Mladek, P. Charbonneau, D. Frenkel, Phase coexistence of cluster crystals: beyond the Gibbs phase rule, *Phys. Rev. Lett.* 99 (23) (2007), <https://doi.org/10.1103/PhysRevLett.99.235702>.
- [225] F. Smallenburg, L. Filion, M. Marechal, M. Dijkstra, Vacancy-stabilized crystalline order in hard cubes, *Proc. Natl. Acad. Sci. USA* 109 (44) (2012) 17886–17890, <https://doi.org/10.1073/pnas.1211784109>.
- [226] A.Z. Panagiotopoulos, Molecular simulation of phase equilibria, in: E. Kiran, J.M.H. Leylet Sengers (Eds.), *Supercritical Fluids: Fundamentals for Applications*, Kluwer, Dordrecht, 1994, pp. 411–437.
- [227] A.Z. Panagiotopoulos, Gibbs ensemble techniques, in: Marc Bauss, Luis F. Rull, Jean-Paul Ryckaert (Eds.), *Observation, Prediction, and Simulation of Phase Transitions in Complex Fluids*, Kluwer, Dordrecht, 1995, pp. 463–501.
- [228] A.Z. Panagiotopoulos, Molecular simulation of phase-equilibria - simple, ionic and polymeric fluids, *Fluid Phase Equilib.* 76 (1992) 97–112, [https://doi.org/10.1016/0378-3812\(92\)85080-R](https://doi.org/10.1016/0378-3812(92)85080-R).
- [229] J.P. Valleau, Density-scaling - a new Monte-Carlo technique in statistical-mechanics, *J. Comput. Phys.* 96 (1) (1991) 193–216, [https://doi.org/10.1016/0021-9991\(91\)90271-L](https://doi.org/10.1016/0021-9991(91)90271-L).

668 Bibliography

- [230] J.P. Valleau, The coulombic phase-transition - density-scaling Monte-Carlo, *J. Chem. Phys.* 95 (1) (1991) 584–589, <https://doi.org/10.1063/1.461460>.
- [231] B. Smit, C.P. Williams, E.M. Hendriks, S.W. de Leeuw, Vapour-liquid equilibria for Stockmayer fluids, *Mol. Phys.* 68 (3) (1989) 765–769, <https://doi.org/10.1080/00268978900102531>.
- [232] R.A. van Nordstrand, D.S. Santilli, S.I. Zones, Aluminum- and boron-containing ssz-24: inverse shape selectivity in the afi structure, *Synth. Microporous Mater.* 1 (1992) 373–383.
- [233] P.G. de Gennes, P.G. Pincus, Pair correlations in a ferromagnetic colloid, *Phys. Kondens. Mater.* 11 (1970) 189–198, <https://doi.org/10.1007/BF02422637>.
- [234] V.I. Kalikmanov, Statistical thermodynamics of ferrofluids, *Physica A* 183 (1–2) (1992) 25–50, [https://doi.org/10.1016/0378-4371\(92\)90176-Q](https://doi.org/10.1016/0378-4371(92)90176-Q).
- [235] G.S. Rushbrooke, G. Stell, J.S. Hoye, Theory of polar liquids I. Dipolar hard spheres, *Mol. Phys.* 26 (1973) 1199–1215, <https://doi.org/10.1080/00268977300102411>.
- [236] K.-C. Ng, J.P. Valleau, G.M. Torrie, G.N. Patey, Liquid-vapour co-existence of dipolar hard spheres, *Mol. Phys.* 38 (3) (1979) 781–788, <https://doi.org/10.1080/00268977900102041>.
- [237] J.M. Caillol, Search of the gas-liquid transition of dipolar hard-spheres, *J. Chem. Phys.* 98 (12) (1993) 9835–9849, <https://doi.org/10.1063/1.464362>.
- [238] M.E. van Leeuwen, B. Smit, What makes a polar fluid a liquid?, *Phys. Rev. Lett.* 71 (24) (1993) 3991–3994, <https://doi.org/10.1103/PhysRevLett.71.3991>.
- [239] J.J. Weis, D. Levesque, Chain formation in low density dipolar hard spheres: a Monte Carlo study, *Phys. Rev. Lett.* 71 (17) (1993) 2729–2732, <https://doi.org/10.1103/PhysRevLett.71.2729>.
- [240] L. Rovigatti, J. Russo, F. Sciortino, No evidence of gas-liquid coexistence in dipolar hard spheres, *Phys. Rev. Lett.* 107 (23) (2011), <https://doi.org/10.1103/PhysRevLett.107.237801>.
- [241] M.E. van Leeuwen, C.J. Peters, J. de Swaan Arons, A.Z. Panagiotopoulos, Investigation of the transition to liquid-liquid immiscibility for Lennard-Jones (12, 6) systems, using the Gibbs-ensemble molecular simulations, *Fluid Phase Equilib.* 66 (1991) 57–75, [https://doi.org/10.1016/0378-3812\(91\)85047-X](https://doi.org/10.1016/0378-3812(91)85047-X).
- [242] V.I. Harismiadis, N.K. Koutras, D.P. Tassios, A.Z. Panagiotopoulos, How good is conformal solution theory for phase equilibrium predictions, *Fluid Phase Equilib.* 65 (1991) 1–18, [https://doi.org/10.1016/0378-3812\(91\)87014-Z](https://doi.org/10.1016/0378-3812(91)87014-Z).
- [243] A.Z. Panagiotopoulos, Exact calculations of fluid-phase equilibria by Monte-Carlo simulation in a new statistical ensemble, *Int. J. Thermophys.* 10 (2) (1989) 447–457, <https://doi.org/10.1007/Bf01133541>.
- [244] J.G. Amar, Application of the Gibbs ensemble to the study of fluid-fluid phase-equilibrium in a binary mixture of symmetric non-additive hard-spheres, *Mol. Phys.* 67 (4) (1989) 739–745, <https://doi.org/10.1080/00268978900101411>.
- [245] R.D. Mountain, A.H. Harvey, Computer simulation of fluid-fluid phase equilibria in mixtures of non-additive soft disks, *J. Chem. Phys.* 94 (3) (1991) 2238–2243, <https://doi.org/10.1063/1.459894>.
- [246] E. de Miguel, E. Martin del Rio, M.M. Telo da Gama, Liquid-liquid phase equilibria of symmetrical mixtures by simulation in the semi-grand canonical ensemble, *J. Chem. Phys.* 103 (1995) 6188–6196, <https://doi.org/10.1063/1.470446>.
- [247] M.R. Stapleton, D.J. Tildesley, N. Quirke, Phase equilibria in polydisperse fluids, *J. Chem. Phys.* 92 (7) (1990) 4456–4467, <https://doi.org/10.1063/1.457756>.
- [248] S. Nosé, A unified formulation of the constant temperature molecular dynamics methods, *J. Chem. Phys.* 81 (1) (1984) 511–519, <https://doi.org/10.1063/1.447334>.
- [249] J. Ruiz-Franco, L. Rovigatti, E. Zaccarelli, On the effect of the thermostat in non-equilibrium molecular dynamics simulations, *Eur. Phys. J. E* 41 (7) (2018), <https://doi.org/10.1140/epje/i2018-11689-4>.
- [250] O. Farago, N. Gronbech-Jensen, On the connection between dissipative particle dynamics and the Ito-Stratonovich dilemma, *J. Chem. Phys.* 144 (8) (2016), <https://doi.org/10.1063/1.4942114>.

- [251] H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. Dinola, J.R. Haak, Molecular dynamics with coupling to an external bath, *J. Chem. Phys.* 81 (8) (1984) 3684–3690, <https://doi.org/10.1063/1.448118>.
- [252] S.C. Harvey, R.K.Z. Tan, T.E. Cheatham, The flying ice cube: velocity rescaling in molecular dynamics leads to violation of energy equipartition, *J. Comput. Chem.* 19 (7) (1998) 726–740, [https://doi.org/10.1002/\(SICI\)1096-987x\(199805\)19:7<726::Aid-Jcc4>3.0.co;2-s](https://doi.org/10.1002/(SICI)1096-987x(199805)19:7<726::Aid-Jcc4>3.0.co;2-s).
- [253] E. Braun, S.M. Moosavi, B. Smit, Anomalous effects of velocity rescaling algorithms: the flying ice cube effect revisited, *J. Chem. Theory Comput.* 14 (10) (2018) 5262–5272, <https://doi.org/10.1021/acs.jctc.8b00446>.
- [254] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. I, Wiley, New York, 1957.
- [255] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. II, Wiley, New York, 1966.
- [256] G. Bussi, D. Donadio, M. Parrinello, Canonical sampling through velocity rescaling, *J. Chem. Phys.* 126 (1) (2007) 014101, <https://doi.org/10.1063/1.2408420>.
- [257] W.G. Hoover, Canonical dynamics: equilibrium phase-space distributions, *Phys. Rev. A* 31 (3) (1985) 1695–1697, <https://doi.org/10.1103/PhysRevA.31.1695>.
- [258] C.P. Lowe, An alternative approach to dissipative particle dynamics, *Europhys. Lett.* 47 (2) (1999) 145–151, <https://doi.org/10.1209/epl/i1999-00365-x>.
- [259] E.A. Koopman, C.P. Lowe, Advantages of a Lowe-Andersen thermostat in molecular dynamics simulations, *J. Chem. Phys.* 124 (20) (2006), <https://doi.org/10.1063/1.2198824>.
- [260] P. Langevin, Sur la theorie du mouvement brownien, *C. R. Acad. Sci.* (1908).
- [261] D.A. Sivak, J.D. Chodera, G.E. Crooks, Time step rescaling recovers continuous-time dynamical properties for discrete-time Langevin integration of nonequilibrium systems, *J. Phys. Chem. B* 118 (24) (2014) 6466–6474, <https://doi.org/10.1021/jp411770f>.
- [262] J. Fass, D.A. Sivak, G.E. Crooks, K.A. Beauchamp, B. Leimkuhler, J.D. Chodera, Quantifying configuration-sampling error in Langevin simulations of complex molecular systems, *Entropy* 20 (5) (2018), <https://doi.org/10.3390/e20050318>.
- [263] N.G. van Kampen, Ito versus Stratonovich, *J. Stat. Phys.* 24 (1) (1981) 175–187, <https://doi.org/10.1007/Bf01007642>.
- [264] L.V. Woodcock, Isothermal molecular dynamics calculations for liquid salts, *Chem. Phys. Lett.* 10 (3) (1971) 257, [https://doi.org/10.1016/0009-2614\(71\)80281-6](https://doi.org/10.1016/0009-2614(71)80281-6).
- [265] S. Nosé, A molecular dynamics method for simulation in the canonical ensemble, *Mol. Phys.* 52 (1984) 255–268, <https://doi.org/10.1080/00268978400101201>.
- [266] V.J. Witherspoon, R. Mercado, E. Braun, A. Mace, J. Bachman, J.R. Long, B. Blumich, B. Smit, J.A. Reimer, Combined nuclear magnetic resonance and molecular dynamics study of methane adsorption in m-2(dobdc) metal-organic frameworks, *J. Phys. Chem. C* 123 (19) (2019) 12286–12295, <https://doi.org/10.1021/acs.jpcc.9b01733>.
- [267] M.E. Tuckerman, Y. Liu, G. Ciccotti, G.J. Martyna, Non-Hamiltonian molecular dynamics: generalizing Hamilton phase space principles to non-Hamiltonian systems, *J. Chem. Phys.* 116 (2001) 1678, <https://doi.org/10.1063/1.1378321>.
- [268] S. Toxvaerd, O.H. Olsen, Canonical molecular dynamics of molecules with internal degrees of freedom, *Ber. Bunsenges. Phys. Chem.* 94 (3) (1990) 274–278, <https://doi.org/10.1002/bbpc.19900940316>.
- [269] G.J. Martyna, M.L. Klein, M.E. Tuckerman, Nosé-Hoover chains: the canonical ensemble via continuous dynamics, *J. Chem. Phys.* 97 (4) (1992) 2635–2645, <https://doi.org/10.1063/1.463940>.
- [270] D.M. Heyes, Molecular-dynamics at constant pressure and temperature, *Chem. Phys.* 82 (3) (1983) 285–301, [https://doi.org/10.1016/0301-0104\(83\)85235-5](https://doi.org/10.1016/0301-0104(83)85235-5).
- [271] M. Hecht, J. Harting, T. Ihle, H.J. Herrmann, Simulation of claylike colloids, *Phys. Rev. E* 72 (1) (2005), <https://doi.org/10.1103/PhysRevE.72.011408>.
- [272] M. Ceriotti, G. Bussi, M. Parrinello, Langevin equation with colored noise for constant-temperature molecular dynamics simulations, *Phys. Rev. Lett.* 102 (2) (2009), <https://doi.org/10.1103/PhysRevLett.102.020601>.

670 Bibliography

- [273] G. Bussi, T. Zykova-Timan, M. Parrinello, Isothermal-isobaric molecular dynamics using stochastic velocity rescaling, *J. Chem. Phys.* 130 (7) (2009), <https://doi.org/10.1063/1.3073889>.
- [274] K. Binder, *Monte Carlo Methods in Statistical Physics*, Springer, Berlin, 1979.
- [275] I.R. McDonald, K. Singer, Machine calculation of thermodynamic properties of a simple fluid at supercritical temperatures, *J. Chem. Phys.* 47 (11) (1967) 4766–4772, <https://doi.org/10.1063/1.1701695>.
- [276] W.G. Hoover, F.H. Ree, Use of computer experiments to locate the melting transition and calculate the entropy in the solid phase, *J. Chem. Phys.* 47 (1967) 4873–4878, <https://doi.org/10.1063/1.1701730>.
- [277] J.-P. Hansen, L. Verlet, Phase transitions of the Lennard-Jones system, *Phys. Rev.* 184 (1) (1969) 151–161, <https://doi.org/10.1103/PhysRev.184.151>.
- [278] F.M. Ytreberg, R.H. Swendsen, D.M. Zuckerman, Comparison of free energy methods for molecular systems, *J. Chem. Phys.* 125 (18) (2006) 184114, <https://doi.org/10.1063/1.2378907>.
- [279] C. Chipot, A. Pohorille, *Free Energy Calculations*, Springer, 2007.
- [280] E. Vanden-Eijnden, Some recent techniques for free energy calculations, *J. Comput. Chem.* 30 (11) (2009) 1737–1747, <https://doi.org/10.1002/jcc.21332>.
- [281] S. Singh, M. Chopra, J.J. de Pablo, Density of states-based molecular simulations, *Annu. Rev. Chem. Biomol. Eng.* 3 (2012) 369–394, <https://doi.org/10.1146/annurev-chembioeng-062011-081032>.
- [282] O. Valsson, P. Tiwary, M. Parrinello, Enhancing important fluctuations: rare events and meta-dynamics from a conceptual viewpoint, *Annu. Rev. Phys. Chem.* 67 (2016) 159–184, <https://doi.org/10.1146/annurev-physchem-040215-112229>.
- [283] B.J. Alder, T.E. Wainwright, Phase transition in elastic disks, *Phys. Rev.* 127 (2) (1962) 359, <https://doi.org/10.1103/PhysRev.127.359>.
- [284] S. Toxvaerd, E. Praestgaard, Molecular dynamics calculation of the liquid structure up to a solid surface, *J. Chem. Phys.* 67 (11) (1977) 5291–5295, <https://doi.org/10.1063/1.434707>.
- [285] A.J.C. Ladd, L.V. Woodcock, Triple-point coexistence properties of the Lennard-Jones system, *Chem. Phys. Lett.* 51 (1) (1977) 155–159, [https://doi.org/10.1016/0009-2614\(77\)85375-X](https://doi.org/10.1016/0009-2614(77)85375-X).
- [286] A.J.C. Ladd, L.V. Woodcock, Interfacial and co-existence properties of the Lennard-Jones system at the triple point, *Mol. Phys.* 36 (2) (1978) 611–619, <https://doi.org/10.1080/00268977800101791>.
- [287] J.N. Cape, L.V. Woodcock, Molecular dynamics calculation of phase coexistence properties: the soft-sphere melting transition, *Chem. Phys. Lett.* 59 (2) (1978) 271–274, [https://doi.org/10.1016/0009-2614\(78\)89094-0](https://doi.org/10.1016/0009-2614(78)89094-0).
- [288] Y. Hiwatari, E. Stoll, T. Schneider, Molecular-dynamics investigation of solid-liquid coexistence, *J. Chem. Phys.* 68 (8) (1978) 3401–3404, <https://doi.org/10.1063/1.436246>.
- [289] A. Ueda, J. Takada, Y. Hiwatari, Molecular-dynamics studies of solid-liquid interface of soft-core model, *J. Phys. Soc. Jpn.* 50 (1) (1981) 307–314, <https://doi.org/10.1143/JPSJ.50.307>.
- [290] W.B. Streett, H.J. Raveche, R.D. Mountain, Monte-Carlo studies of fluid-solid phase-transition in Lennard-Jones system, *J. Chem. Phys.* 61 (5) (1974) 1960–1969, <https://doi.org/10.1063/1.1682197>.
- [291] A.B. Belonoshko, N.V. Skorodumova, A. Rosengren, B. Johansson, Melting and critical superheating, *Phys. Rev. B* 73 (1) (2006), <https://doi.org/10.1103/PhysRevB.73.012201>.
- [292] D. Alfe, C. Cazorla, M.J. Gillan, The kinetics of homogeneous melting beyond the limit of superheating, *J. Chem. Phys.* 135 (2) (2011), <https://doi.org/10.1063/1.3605601>.
- [293] D.A. Kofke, Gibbs-Duhem integration - a new method for direct evaluation of phase coexistence by molecular simulation, *Mol. Phys.* 78 (6) (1993) 1331–1336, <https://doi.org/10.1080/00268979300100881>.
- [294] D.A. Kofke, Direct evaluation of phase coexistence by molecular simulation via integration along the saturation line, *J. Chem. Phys.* 98 (5) (1993) 4149–4162, <https://doi.org/10.1063/1.465023>.

- [295] R. Agrawal, D.A. Kofke, Thermodynamic and structural properties of model systems at solid-fluid coexistence, *Mol. Phys.* 85 (1) (1995) 43–59, <https://doi.org/10.1080/00268979500100921>.
- [296] M.H.J. Hagen, E.J. Meijer, G.C.A.M. Mooij, D. Frenkel, H.N.W. Lekkerkerker, Does c-60 have a liquid-phase, *Nature* 365 (6445) (1993) 425–426, <https://doi.org/10.1038/365425a0>.
- [297] E.J. Meijer, D. Frenkel, Colloids dispersed in polymer solution. A computer simulation study, *J. Chem. Phys.* 100 (9) (1994) 6873–6887, <https://doi.org/10.1063/1.467003>.
- [298] W.O. Haag, Catalysis by zeolites—science and technology, *Stud. Surf. Sci. Catal.* 84 (1994) 1375–1394, [https://doi.org/10.1016/S0167-2991\(08\)63680-0](https://doi.org/10.1016/S0167-2991(08)63680-0).
- [299] P.G. Bolhuis, M.H.J. Hagen, D. Frenkel, Isostructural solid-solid transition in crystalline systems with short-ranged interaction, *Phys. Rev. E* 50 (6) (1994) 4880–4890, <https://doi.org/10.1103/physreve.50.4880>.
- [300] M. Dijkstra, The Effect of Entropy on the Stability and Structure of Complex Fluids, Ph.D. thesis, Utrecht University, 1994.
- [301] M. Dijkstra, D. Frenkel, Simulation study of a two-dimensional system of semiflexible polymers, *Phys. Rev. E* 50 (1) (1994) 349–357, <https://doi.org/10.1103/PhysRevE.50.349>.
- [302] P.G. Bolhuis, D.A. Kofke, Numerical study of freezing in polydisperse colloidal suspensions, *J. Phys. Condens. Matter* 8 (47) (1996) 9627–9631, <https://doi.org/10.1088/0953-8984/8/47/074>.
- [303] P.G. Bolhuis, D. Frenkel, Tracing the phase boundaries of hard spherocylinders, *J. Chem. Phys.* 106 (2) (1997) 666–687, <https://doi.org/10.1063/1.473404>.
- [304] E.J. Meijer, F.E. Azhar, Novel procedure to determine coexistence lines by computer simulation. Application to hard-core Yukawa model for charge-stabilized colloids, *J. Chem. Phys.* 106 (11) (1997) 4678–4683, <https://doi.org/10.1063/1.473504>.
- [305] F.A. Escobedo, J.J. de Pablo, Pseudo-ensemble simulations and Gibbs-Duhem integrations for polymers, *J. Chem. Phys.* 106 (7) (1997) 2911–2923, <https://doi.org/10.1063/1.473353>.
- [306] F.A. Escobedo, Tracing coexistence lines in multicomponent fluid mixtures by molecular simulation, *J. Chem. Phys.* 110 (24) (1999) 11999–12010, <https://doi.org/10.1063/1.479137>.
- [307] W.G. Hoover, F.H. Ree, Melting transition and communal entropy for hard spheres, *J. Chem. Phys.* 49 (8) (1968) 3609–3617, <https://doi.org/10.1063/1.1670641>.
- [308] G.S. Stringfellow, H.E. DeWitt, W.L. Slattery, Equation of state of the one-component plasma derived from precision Monte Carlo calculations, *Phys. Rev. A* 41 (2) (1990) 1105–1111, <https://doi.org/10.1103/physreva.41.1105>.
- [309] W.G. Hoover, S.G. Gray, K.W. Johnson, Thermodynamic properties of the fluid and solid phases for the inverse power potentials, *J. Chem. Phys.* 55 (1971) 1128–1136, <https://doi.org/10.1063/1.1676196>.
- [310] B.B. Laird, A.D.J. Haymet, Phase diagram for the inverse sixth power potential system from molecular dynamics simulations, *Mol. Phys.* 75 (1992) 71–80, <https://doi.org/10.1080/00268979200100071>.
- [311] W.G. Hoover, M. Ross, K.W. Johnson, D. Henderson, J.A. Barker, B.C. Brown, Soft-sphere equation of state, *J. Chem. Phys.* 52 (10) (1970) 4931–4941, <https://doi.org/10.1063/1.1672728>.
- [312] J.-P. Hansen, Phase transition of the Lennard-Jones system. II. High-temperature limit, *Phys. Rev. A* 2 (1970) 221–230, <https://doi.org/10.1103/PhysRevA.2.221>.
- [313] H. Ogura, H. Matsuda, T. Ogawa, N. Ogita, A. Ueda, Computer simulations for the melting curve maximum phenomenon, *Prog. Theor. Phys.* 58 (1977) 419–433, <https://doi.org/10.1143/PTP.58.419>.
- [314] D. Frenkel, A.J.C. Ladd, New Monte-Carlo method to compute the free-energy of arbitrary solids - application to the fcc and hcp phases of hard-spheres, *J. Chem. Phys.* 81 (7) (1984) 3188–3193, <https://doi.org/10.1063/1.448024>.
- [315] R. Agrawal, D.A. Kofke, Solid-fluid coexistence for inverse-power potentials, *Phys. Rev. Lett.* 74 (1) (1995) 122–125, <https://doi.org/10.1103/PhysRevLett.74.122>.

672 Bibliography

- [316] L. Onsager, Theories of concentrated electrolytes, *Chem. Rev.* 13 (1) (1933) 73–89, <https://doi.org/10.1021/cr60044a006>.
- [317] S.W. de Leeuw, B. Smit, C.P. Williams, Molecular dynamics studies of polar/nonpolar fluid mixtures: I. Mixtures of Lennard-Jones and Stockmayer fluids, *J. Chem. Phys.* 93 (4) (1990) 2704–2719, <https://doi.org/10.1063/1.458909>.
- [318] N.F. Carnahan, K.E. Starling, Equation of state for nonattracting rigid spheres, *J. Chem. Phys.* 51 (2) (1969) 635, <https://doi.org/10.1063/1.1672048>.
- [319] N.G. Almarza, A cluster algorithm for Monte Carlo simulation at constant pressure, *J. Chem. Phys.* 130 (18) (2009), <https://doi.org/10.1063/1.3133328>.
- [320] R.J. Speedy, Pressure and entropy of hard-sphere crystals, *J. Phys. Condens. Matter* 10 (20) (1998) 4387–4391, <https://doi.org/10.1088/0953-8984/10/20/006>.
- [321] A. Mulero, C.A. Galán, M.I. Parra, F. Cuadros, Equations of state for hard spheres and hard disks, in: A. Mulero (Ed.), *Theory and Simulation of Hard-Sphere Fluids and Related Systems*, vol. 753, Springer, Berlin Heidelberg, 2008, pp. 37–109.
- [322] H. Hansen-Goos, Accurate prediction of hard-sphere virial coefficients b-6 to b-12 from a compressibility-based equation of state, *J. Chem. Phys.* 144 (16) (2016), <https://doi.org/10.1063/1.4947534>.
- [323] A.J. Schultz, D.A. Kofke, Comprehensive high-precision high-accuracy equation of state and coexistence properties for classical Lennard-Jones crystals and low-temperature fluid phases, *J. Chem. Phys.* 149 (20) (2018), <https://doi.org/10.1063/1.5053714>.
- [324] W.F. van Gunsteren, P.K. Weiner, A.J. Wilkinson, *Computer Simulation of Biomolecular Systems: Theoretical and Experimental Applications*, vol. 2, ESCOM, Leiden [Netherlands], 1993.
- [325] M. Bonomi, C. Camilloni, Biomolecular simulations: methods and protocols, in: *Methods in Molecular Biology* 2022, 1st edition, Springer, New York, NY, 2019.
- [326] W.G. Hoover, J.C. Poirier, Determination of virial coefficients from potential of mean force, *J. Chem. Phys.* 37 (5) (1962) 1041, <https://doi.org/10.1063/1.1733209>.
- [327] S. Romano, K. Singer, Calculation of the entropy of liquid chlorine and bromine by computer-simulation, *Mol. Phys.* 37 (6) (1979) 1765–1772, <https://doi.org/10.1080/0026897900101311>.
- [328] N. Metropolis, S.M. Ulam, The Monte Carlo method, *J. Am. Stat. Assoc.* 44 (247) (1949) 335–341, <https://doi.org/10.2307/2280232>.
- [329] A. Bakhshandeh, Y. Levin, Widom insertion method in simulations with Ewald summation, *J. Chem. Phys.* 156 (13) (2022), <https://doi.org/10.1063/5.0085527>.
- [330] B. Widom, Structure of interfaces from uniformity of the chemical-potential, *J. Stat. Phys.* 19 (6) (1978) 563–574, <https://doi.org/10.1007/BF01011768>.
- [331] D. Frenkel, Free energy computations and first order phase transitions, in: G. Ciccotti, W.G. Hoover (Eds.), *Molecular Dynamics Simulations of Statistical Mechanics Systems*, North Holland, Amsterdam, 1986, pp. 151–188, <https://www.researchgate.net/publication/27692139>.
- [332] R. Lustig, Statistical mechanics in the classical molecular dynamics ensemble. I. Fundamentals, *J. Chem. Phys.* 101 (4) (1994) 3048–3059, <https://doi.org/10.1063/1.466446>.
- [333] R. Lustig, Statistical thermodynamics in the classical molecular dynamics ensemble. II. Application to computer simulation, *J. Chem. Phys.* 101 (4) (1994) 3060–3067, <https://doi.org/10.1063/1.466447>.
- [334] K.S. Shing, Infinite-dilution activity-coefficients from computer-simulation, *Chem. Phys. Lett.* 119 (2–3) (1985) 149–151, [https://doi.org/10.1016/0009-2614\(85\)80050-6](https://doi.org/10.1016/0009-2614(85)80050-6).
- [335] P. Sindzingre, G. Ciccotti, C. Massobrio, D. Frenkel, Partial enthalpies and related quantities in mixtures from computer-simulation, *Chem. Phys. Lett.* 136 (1) (1987) 35–41, [https://doi.org/10.1016/0009-2614\(87\)87294-9](https://doi.org/10.1016/0009-2614(87)87294-9).
- [336] U. Heinbruch, J. Fischer, On the application of Widom's test particle method to homogeneous and inhomogeneous fluids, *Mol. Simul.* 1 (1–2) (1987) 109–120, <https://doi.org/10.1080/08927028708080935>.

- [337] J.I. Siepmann, I.R. McDonald, D. Frenkel, Finite-size corrections to the chemical-potential, *J. Phys. Condens. Matter* 4 (3) (1992) 679–691, <https://doi.org/10.1088/0953-8984/4/3/009>.
- [338] K.S. Shing, S.T. Chung, Computer-simulation methods for the calculation of solubility in supercritical extraction systems, *J. Phys. Chem.* 91 (6) (1987) 1674–1681, <https://doi.org/10.1021/j100290a077>.
- [339] K.S. Shing, Infinite-dilution activity-coefficients of quadrupolar Lennard-Jones mixtures from computer-simulation, *J. Chem. Phys.* 85 (8) (1986) 4633–4637, <https://doi.org/10.1063/1.451759>.
- [340] P. Sindzingre, C. Massobrio, G. Ciccotti, D. Frenkel, Calculation of partial enthalpies of an argon-krypton mixture by npt molecular-dynamics, *Chem. Phys.* 129 (2) (1989) 213–224, [https://doi.org/10.1016/0301-0104\(89\)80007-2](https://doi.org/10.1016/0301-0104(89)80007-2).
- [341] K.S. Shing, K.E. Gubbins, Free-energy and vapor liquid equilibria for a quadrupolar Lennard-Jones fluid, *Mol. Phys.* 45 (1) (1982) 129–139, <https://doi.org/10.1080/00268978200100101>.
- [342] K.S. Shing, K.E. Gubbins, The chemical-potential in non-ideal liquid-mixtures computer-simulation and theory, *Mol. Phys.* 49 (5) (1983) 1121–1138, <https://doi.org/10.1080/00268978300101811>.
- [343] C.H. Bennett, Efficient estimation of free-energy differences from Monte-Carlo data, *J. Comput. Phys.* 22 (2) (1976) 245–268, [https://doi.org/10.1016/0021-9991\(76\)90078-4](https://doi.org/10.1016/0021-9991(76)90078-4).
- [344] A.M. Ferrenberg, R.H. Swendsen, Optimized Monte Carlo data analysis, *Phys. Rev. Lett.* 63 (12) (1989) 1195–1198, <https://doi.org/10.1103/PhysRevLett.63.1195>.
- [345] L. Landau, The theory of phase transitions, *Nature* 138 (1936) 840–841, <https://doi.org/10.1038/138840a0>.
- [346] I.R. McDonald, K. Singer, Examination of the adequacy of the 12-6 potential for liquid argon by means of Monte Carlo calculations, *J. Chem. Phys.* 50 (6) (1969) 2308–2315, <https://doi.org/10.1063/1.1671381>.
- [347] G.M. Torrie, J.P. Valleau, Nonphysical sampling distributions in Monte Carlo free-energy estimation: umbrella sampling, *J. Comput. Phys.* 23 (1977) 187–199, [https://doi.org/10.1016/0021-9991\(77\)90121-8](https://doi.org/10.1016/0021-9991(77)90121-8).
- [348] J.P. Valleau, Monte Carlo: choosing which game to play, in: M. Meyer, V. Pontikis (Eds.), *Proceedings of the NATO ASI on Computer Simulation in Materials Science*, vol. 205, Kluwer, Dordrecht, 1991, pp. 67–84.
- [349] J.P. Valleau, Density-scaling Monte Carlo study of subcritical Lennard-Jonesium, *J. Chem. Phys.* 99 (6) (1977) 4718–4728, <https://doi.org/10.1063/1.466069>.
- [350] B.A. Berg, T. Neuhaus, Multicanonical ensemble: a new approach to simulate first-order phase transitions, *Phys. Rev. Lett.* 68 (1) (1992) 9–12, <https://doi.org/10.1103/PhysRevLett.68.9>.
- [351] F.G. Wang, D.P. Landau, Efficient, multiple-range random walk algorithm to calculate the density of states, *Phys. Rev. Lett.* 86 (10) (2001) 2050–2053, <https://doi.org/10.1103/PhysRevLett.86.2050>.
- [352] F.G. Wang, D.P. Landau, Determining the density of states for classical statistical models: a random walk algorithm to produce a flat histogram, *Phys. Rev. E* 64 (5) (2001) 056101, <https://doi.org/10.1103/PhysRevE.64.056101>.
- [353] D.P. Landau, S.H. Tsai, M. Exler, A new approach to Monte Carlo simulations in statistical physics: Wang-Landau sampling, *Am. J. Phys.* 72 (10) (2004) 1294–1302, <https://doi.org/10.1119/1.1707017>.
- [354] F. Calvo, Sampling along reaction coordinates with the Wang-Landau method, *Mol. Phys.* 100 (21) (2002) 3421–3427, <https://doi.org/10.1080/00268970210158632>.
- [355] A. Laio, M. Parrinello, Escaping free-energy minima, *Proc. Natl. Acad. Sci. USA* 99 (20) (2002) 12562–12566, <https://doi.org/10.1073/pnas.202427399>.
- [356] M.S. Shell, P.G. Debenedetti, A.Z. Panagiotopoulos, Generalization of the Wang-Landau method for off-lattice simulations, *Phys. Rev. E* 66 (5) (2002) 056703, <https://doi.org/10.1103/PhysRevE.66.056703>.

674 Bibliography

- [357] Q.L. Yan, J.J. de Pablo, Fast calculation of the density of states of a fluid by Monte Carlo simulations, Phys. Rev. Lett. 90 (3) (2003) 035701, <https://doi.org/10.1103/PhysRevLett.90.035701>.
- [358] A. Troster, C. Dellago, Wang-Landau sampling with self-adaptive range, Phys. Rev. E 71 (6) (2005) 066705, <https://doi.org/10.1103/PhysRevE.71.066705>.
- [359] C.G. Zhou, R.N. Bhatt, Understanding and improving the Wang-Landau algorithm, Phys. Rev. E 72 (2) (2005) 025701, <https://doi.org/10.1103/PhysRevE.72.025701>.
- [360] R.E. Belardinelli, V.D. Pereyra, Wang-Landau algorithm: a theoretical analysis of the saturation of the error, J. Chem. Phys. 127 (18) (2007) 184105, <https://doi.org/10.1063/1.2803061>.
- [361] G. Brown, T.C. Schulthess, Wang-Landau estimation of magnetic properties for the Heisenberg model, J. Appl. Phys. 97 (10) (2005), <https://doi.org/10.1063/1.1847311>.
- [362] P. Poulaing, F. Calvo, R. Antoine, M. Broyer, P. Dugourd, Performances of Wang-Landau algorithms for continuous systems, Phys. Rev. E 73 (5) (2006) 056704, <https://doi.org/10.1103/PhysRevE.73.056704>.
- [363] C.G. Zhou, T.C. Schulthess, S. Torbrugge, D.P. Landau, Wang-Landau algorithm for continuous models and joint density of states, Phys. Rev. Lett. 96 (12) (2006) 120201, <https://doi.org/10.1103/PhysRevLett.96.120201>.
- [364] Q.L. Yan, R. Faller, J.J. de Pablo, Density-of-states Monte Carlo method for simulation of fluids, J. Chem. Phys. 116 (20) (2002) 8745–8749, <https://doi.org/10.1063/1.1463055>.
- [365] E.A. Mastny, J.J. de Pablo, Direct calculation of solid-liquid equilibria from density-of-states Monte Carlo simulations, J. Chem. Phys. 122 (12) (2005) 124109, <https://doi.org/10.1063/1.1874792>.
- [366] N. Rathore, T.A. Knotts, J.J. de Pablo, Density of states simulations of proteins, J. Chem. Phys. 118 (9) (2003) 4285–4290, <https://doi.org/10.1063/1.1542598>.
- [367] T. Huber, A.E. Torda, W.F. van Gunsteren, Local elevation: a method for improving the searching properties of molecular dynamics simulation, J. Comput.-Aided Mol. Des. 8 (6) (1994) 695–708, <https://doi.org/10.1007/Bf00124016>.
- [368] C. Theodoropoulos, Y.H. Qian, I.G. Kevrekidis, “Coarse” stability and bifurcation analysis using time-steppers: a reaction-diffusion example, Proc. Natl. Acad. Sci. USA 97 (18) (2000) 9840–9843, <https://doi.org/10.1073/pnas.97.18.9840>.
- [369] L. Maragliano, E. Vanden-Eijnden, A temperature accelerated method for sampling free energy and determining reaction pathways in rare events simulations, Chem. Phys. Lett. 426 (1–3) (2006) 168–175, <https://doi.org/10.1016/j.cplett.2006.05.062>.
- [370] A. Laio, F.L. Gervasio, Metadynamics: a method to simulate rare events and reconstruct the free energy in biophysics, chemistry and material science, Rep. Prog. Phys. 71 (12) (2008), <https://doi.org/10.1088/0034-4885/71/12/126601>.
- [371] G. Bussi, D. Branduardi, Free-energy calculations with metadynamics: theory and practice, in: Abby L. Parrill, Kenny B. Lipkowitz (Eds.), Reviews in Computational Chemistry, vol. 28, John Wiley and Sons, 2015, pp. 1–49.
- [372] A. Barducci, G. Bussi, M. Parrinello, Well-tempered metadynamics: a smoothly converging and tunable free-energy method, Phys. Rev. Lett. 100 (2) (2008) 020603, <https://doi.org/10.1103/PhysRevLett.100.020603>.
- [373] J.F. Dama, M. Parrinello, G.A. Voth, Well-tempered metadynamics converges asymptotically, Phys. Rev. Lett. 112 (24) (2014) 240602, <https://doi.org/10.1103/PhysRevLett.112.240602>.
- [374] P. Tiwary, J.F. Dama, M. Parrinello, A perturbative solution to metadynamics ordinary differential equation, J. Chem. Phys. 143 (23) (2015) 234112, <https://doi.org/10.1063/1.4937945>.
- [375] M. Bonomi, M. Parrinello, Enhanced sampling in the well-tempered ensemble, Phys. Rev. Lett. 104 (19) (2010) 190601, <https://doi.org/10.1103/PhysRevLett.104.190601>.
- [376] M. Deighan, M. Bonomi, J. Pfaendtner, Efficient simulation of explicitly solvated proteins in the well-tempered ensemble, J. Chem. Theory Comput. 8 (7) (2012) 2189–2192, <https://doi.org/10.1021/ct300297t>.
- [377] O. Valsson, M. Parrinello, Thermodynamical description of a quasi-first-order phase transition from the well-tempered ensemble, J. Chem. Theory Comput. 9 (12) (2013) 5267–5276, <https://doi.org/10.1021/ct400859f>.

- [378] S. Trebst, D.A. Huse, M. Troyer, Optimizing the ensemble for equilibration in broad-histogram Monte Carlo simulations, Phys. Rev. E 70 (4) (2004) 046701, <https://doi.org/10.1103/PhysRevE.70.046701>.
- [379] E.B. Kim, R. Faller, Q.L. Yan, N.L. Abbott, J.J. de Pablo, Potential of mean force between a spherical particle suspended in a nematic liquid crystal and a substrate, J. Chem. Phys. 117 (16) (2002) 7781–7787, <https://doi.org/10.1063/1.1508365>.
- [380] M.S. Shell, P.G. Debenedetti, A.Z. Panagiotopoulos, Flat-histogram dynamics and optimization in density of states simulations of fluids, J. Phys. Chem. B 108 (51) (2004) 19748–19755, <https://doi.org/10.1021/jp047677j>.
- [381] L. Janosi, M. Doxastakis, Accelerating flat-histogram methods for potential of mean force calculations, J. Chem. Phys. 131 (5) (2009) 054105, <https://doi.org/10.1063/1.3183165>.
- [382] M.R. Shirts, J.D. Chodera, Statistically optimal analysis of samples from multiple equilibrium states, J. Chem. Phys. 129 (12) (2008), <https://doi.org/10.1063/1.2978177>.
- [383] J.S. Wang, R.H. Swendsen, Transition matrix Monte Carlo method, J. Stat. Phys. 106 (1–2) (2002) 245–285, <https://doi.org/10.1023/A:1013180330892>.
- [384] M.S. Shell, P.G. Debenedetti, A.Z. Panagiotopoulos, An improved Monte Carlo method for direct calculation of the density of states, J. Chem. Phys. 119 (18) (2003) 9406–9411, <https://doi.org/10.1063/1.1615966>.
- [385] R.G. Ghulghazaryan, S. Hayryan, C.K. Hu, Efficient combination of Wang-Landau and transition matrix Monte Carlo methods for protein simulations, J. Comput. Chem. 28 (3) (2007) 715–726, <https://doi.org/10.1002/jcc.20597>.
- [386] G.R. Smith, A.D. Bruce, A study of the multi-canonical Monte Carlo method, J. Phys. A, Math. Gen. 28 (23) (1995) 6623–6643, <https://doi.org/10.1088/0305-4470/28/23/015>.
- [387] C. Jarzynski, Equilibrium free-energy differences from nonequilibrium measurements: a master-equation approach, Phys. Rev. E 56 (5) (1997) 5018–5035, <https://doi.org/10.1103/PhysRevE.56.5018>.
- [388] C. Jarzynski, Nonequilibrium equality for free energy differences, Phys. Rev. Lett. 78 (14) (1997) 2690–2693, <https://doi.org/10.1103/PhysRevLett.78.2690>.
- [389] G.E. Crooks, Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems, J. Stat. Phys. 90 (5–6) (1998) 1481–1487, <https://doi.org/10.1023/A:1023208217925>.
- [390] G.E. Crooks, Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences, Phys. Rev. E 60 (3) (1999) 2721–2726, <https://doi.org/10.1103/physreve.60.2721>.
- [391] G.E. Crooks, Path-ensemble averages in systems driven far from equilibrium, Phys. Rev. E 61 (3) (2000) 2361–2366, <https://doi.org/10.1103/PhysRevE.61.2361>.
- [392] W. Lechner, C. Dellago, On the efficiency of path sampling methods for the calculation of free energies from non-equilibrium simulations, J. Stat. Mech. Theory Exp. (2007), <https://doi.org/10.1088/1742-5468/2007/04/P04001>.
- [393] H.A. Vinutha, D. Frenkel, Estimation of the equilibrium free energy for glasses using the Jarzynski equality, J. Chem. Phys. 154 (23) (2021), <https://doi.org/10.1063/5.0053531>.
- [394] A.K. Hartmann, High-precision work distributions for extreme nonequilibrium processes in large systems, Phys. Rev. E 89 (5) (2014), <https://doi.org/10.1103/PhysRevE.89.052103>.
- [395] J.P. Nilmeier, G.E. Crooks, D.D.L. Minh, J.D. Chodera, Nonequilibrium candidate Monte Carlo is an efficient tool for equilibrium simulation, Proc. Natl. Acad. Sci. USA 108 (45) (2011) E1009–E1018, <https://doi.org/10.1073/pnas.1106094108>.
- [396] J.P. Nilmeier, G.E. Crooks, D.D.L. Minh, J.D. Chodera, Nonequilibrium candidate Monte Carlo is an efficient tool for equilibrium simulation (vol 108, pg e1009, 2011), Proc. Natl. Acad. Sci. USA 109 (24) (2012) 9665, <https://doi.org/10.1073/pnas.1207617109>.
- [397] P.W. Bridgman, Change of phase under pressure. I. The phase diagram of eleven substances with especial reference to the melting curve, Phys. Rev. 3 (2) (1914) 126–141, <https://doi.org/10.1103/PhysRev.3.126>.

676 Bibliography

- [398] P.W. Bridgman, Change of phase under pressure. I. The phase diagram of eleven substances with especial reference to the melting curve, *Phys. Rev.* 3 (3) (1914) 153–203, <https://doi.org/10.1103/PhysRev.3.153>.
- [399] J.C. Shelley, G.N. Patey, A configurational bias Monte Carlo method for water, *J. Chem. Phys.* 102 (1995) 7656–7663, <https://doi.org/10.1063/1.469017>.
- [400] P. Tilwani, D. Wu, Direct Simulation of Phase Coexistence in Solids Using the Gibbs Ensemble: Configuration Annealing Monte Carlo, Ph.D. thesis, University of Colorado, Golden, 1999.
- [401] E. Antillon, M. Ghazisaeidi, Efficient determination of solid-state phase equilibrium with the multicell Monte Carlo method, *Phys. Rev. E* 101 (6) (2020), <https://doi.org/10.1103/PhysRevE.101.063306>.
- [402] C. Vega, E. Sanz, J.L.F. Abascal, E.G. Noya, Determination of phase diagrams via computer simulation: methodology and applications to water, electrolytes and proteins, *J. Phys. Condens. Matter* 20 (15) (2008), <https://doi.org/10.1088/0953-8984/20/15/153101>.
- [403] P.A. Monson, D.A. Kofke, Solid-fluid equilibrium: insights from simple molecular models, in: I. Prigogine, S.A. Rice (Eds.), *Advances in Chemical Physics*, vol. 115, John Wiley and Sons Inc, New York, 2000, pp. 113–179.
- [404] E.J. Meijer, Computer Simulations of Molecular Solids and Colloidal Dispersions, Ph.D. thesis, Utrecht University, 1993.
- [405] P. Li, N.A. Vermeulen, C.D. Mallikas, D.A. Gomez-Gualdrón, A.J. Howarth, B.L. Mehdi, A. Dohnalkova, N.D. Browning, M. O’Keeffe, O.K. Farha, Bottom-up construction of a superstructure in a porous uranium-organic crystal, *Science* 356 (6338) (2017) 624–627, <https://doi.org/10.1126/science.aam7851>.
- [406] D. Frenkel, Stability of the high-pressure body-centered-cubic phase of helium, *Phys. Rev. Lett.* 56 (8) (1986) 858–860, <https://doi.org/10.1103/PhysRevLett.56.858>.
- [407] E.J. Meijer, D. Frenkel, R.A. Lesar, A.J.C. Ladd, Location of melting-point at 300-k of nitrogen by Monte-Carlo simulation, *J. Chem. Phys.* 92 (12) (1990) 7570–7575, <https://doi.org/10.1063/1.458193>.
- [408] J.Q. Broughton, G.H. Gilmer, Molecular dynamics investigation of the crystal-fluid interface. I. Bulk properties, *J. Chem. Phys.* 79 (1983) 5095–5104, <https://doi.org/10.1063/1.445633>.
- [409] B.Q. Cheng, M. Ceriotti, Computing the absolute Gibbs free energy in atomistic simulations: applications to defects in solids, *Phys. Rev. B* 97 (5) (2018), <https://doi.org/10.1103/PhysRevB.97.054102>.
- [410] B.M. Mladek, P. Charbonneau, C.N. Likos, D. Frenkel, G. Kahl, Multiple occupancy crystals formed by purely repulsive soft particles, *J. Phys. Condens. Matter* 20 (49) (2008), <https://doi.org/10.1088/0953-8984/20/49/494245>.
- [411] T. Schilling, F. Schmid, Computing absolute free energies of disordered structures by molecular simulation, *J. Chem. Phys.* 131 (23) (2009), <https://doi.org/10.1063/1.3274951>.
- [412] A.D. Bruce, N.B. Wilding, G.J. Ackland, Free energy of crystalline solids: a lattice-switch Monte Carlo method, *Phys. Rev. Lett.* 79 (16) (1997) 3002–3005, <https://doi.org/10.1103/PhysRevLett.79.3002>.
- [413] J.A. Schouten, Phase equilibria in binary systems at very high pressures, *Phys. Rep.* 172 (2) (1989) 33–92, [https://doi.org/10.1016/0370-1573\(89\)90154-3](https://doi.org/10.1016/0370-1573(89)90154-3).
- [414] A.F. Goncharov, J.C. Crowhurst, V.V. Struzhkin, R.J. Hemley, Triple point on the melting curve and polymorphism of nitrogen at high pressure, *Phys. Rev. Lett.* 101 (9) (2008), <https://doi.org/10.1103/PhysRevLett.101.095502>.
- [415] W. Ostwald, Studien über die bildung und umwandlung fester korper: 1. Abhandlung: Übersättigung und überkaltung, *Z. Phys. Chem.* 22U (1) (1897) 289–330, <https://doi.org/10.1515/zpch-1897-2233>.
- [416] D. Frenkel, B.M. Mulder, The hard ellipsoid-of-revolution fluid I. Monte Carlo simulations, *Mol. Phys.* 55 (5) (1985) 1171–1192, <https://doi.org/10.1080/00268978500101971>.
- [417] M.D. Eldridge, P.A. Madden, D. Frenkel, Entropy-driven formation of a superlattice in a hard-sphere binary mixture, *Nature* 365 (6441) (1993) 35–37, <https://doi.org/10.1038/365035a0>.

- [418] M.E. Leunissen, C.G. Christova, A.P. Hyyninen, C.P. Royall, A.I. Campbell, A. Imhof, M. Dijkstra, R. van Roij, A. van Blaaderen, Ionic colloidal crystals of oppositely charged particles, *Nature* 437 (7056) (2005) 235–240, <https://doi.org/10.1038/nature03946>.
- [419] A.P. Hyyninen, C.G. Christova, R. van Roij, A. van Blaaderen, M. Dijkstra, Prediction and observation of crystal structures of oppositely charged colloids, *Phys. Rev. Lett.* 96 (13) (2006), <https://doi.org/10.1103/PhysRevLett.96.138308>.
- [420] C. Vega, E.G. Noya, Revisiting the Frenkel-Ladd method to compute the free energy of solids: the Einstein molecule approach, *J. Chem. Phys.* 127 (15) (2007), <https://doi.org/10.1063/1.2790426>.
- [421] R.J. Speedy, Pressure of the metastable hard-sphere fluid, *J. Phys. Condens. Matter* 9 (41) (1997) 8591–8599, <https://doi.org/10.1088/0953-8984/9/41/006>.
- [422] B.J. Alder, T.E. Wainwright, Studies in molecular dynamics. II. Behavior of a small number of elastic spheres, *J. Chem. Phys.* 33 (5) (1960) 1439–1451, <https://doi.org/10.1063/1.1731425>.
- [423] B. Smit, D. Frenkel, Explicit expression for finite size corrections to the chemical potential, *J. Phys. Condens. Matter* 1 (44) (1989) 8659–8665, <https://doi.org/10.1088/0953-8984/1/44/035>.
- [424] M. Abramowitz, I. Stegun, *Handbook of Mathematical Functions*, Dover Publications, New York, 1970.
- [425] J.M. Polson, E. Trizac, S. Pronk, D. Frenkel, Finite-size corrections to the free energies of crystalline solids, *J. Chem. Phys.* 112 (12) (2000) 5339–5342, <https://doi.org/10.1063/1.481102>.
- [426] L.V. Woodcock, Entropy difference between the face-centred cubic and hexagonal close-packed crystal structures, *Nature* 385 (6612) (1997) 141–143, <https://doi.org/10.1038/385141a0>.
- [427] P.G. Bolhuis, D. Frenkel, S.C. Mau, D.A. Huse, Entropy difference between crystal phases, *Nature* 388 (6639) (1997) 235–236, <https://doi.org/10.1038/40779>.
- [428] J.P. Ryckaert, G. Ciccotti, Introduction of Andersen demon in the molecular-dynamics of systems with constraints, *J. Chem. Phys.* 78 (12) (1983) 7368–7374, <https://doi.org/10.1063/1.444728>.
- [429] W.G. Hoover, Entropy for small classical crystals, *J. Chem. Phys.* 49 (4) (1968) 1981, <https://doi.org/10.1063/1.1670348>.
- [430] S.-C. Mau, D.A. Huse, Stacking entropy of hard-sphere crystals, *Phys. Rev. E* 59 (4) (1999) 4396–4401, <https://doi.org/10.1103/PhysRevE.59.4396>.
- [431] C.H. Bennett, B.J. Alder, Studies in molecular dynamics. IX. Vacancies in hard sphere crystals, *J. Chem. Phys.* 54 (11) (1971) 4796–4808, <https://doi.org/10.1063/1.1674755>.
- [432] R.K. Bowles, R.J. Speedy, Cavities in the hard-sphere crystal and fluid, *Mol. Phys.* 83 (1) (1994) 113–125, <https://doi.org/10.1080/00268979400101111>.
- [433] W.C. Swope, Hans C. Andersen, Thermodynamics, statistical thermodynamics, and computer-simulation of crystals with vacancies and interstitials, *Phys. Rev. A* 46 (8) (1992) 4539–4548, <https://doi.org/10.1103/PhysRevA.46.4539>.
- [434] R.J. Speedy, H. Reiss, Cavities in the hard-sphere fluid and crystal and the equation of state, *Mol. Phys.* 72 (5) (1991) 999–1014, <https://doi.org/10.1080/00268979100100741>.
- [435] R.J. Speedy, H. Reiss, A computer-simulation study of cavities in the hard disk fluid and crystal, *Mol. Phys.* 72 (5) (1991) 1015–1033, <https://doi.org/10.1080/00268979100100751>.
- [436] M.A. Bates, D. Frenkel, Influence of vacancies on the melting transition of hard disks in two dimensions, *Phys. Rev. E* 61 (5A) (2000) 5223–5227, <https://doi.org/10.1103/physreve.61.5223>.
- [437] S. Pronk, D. Frenkel, Point defects in hard-sphere crystals, *J. Phys. Chem. B* 105 (28) (2001) 6722–6727, <https://doi.org/10.1021/jp010779e>.
- [438] M.N. Rosenbluth, A.W. Rosenbluth, Monte Carlo simulations of the average extension of molecular chains, *J. Chem. Phys.* 23 (1955) 356–359, <https://doi.org/10.1063/1.1741967>.

678 Bibliography

- [439] M. Müller, W. Paul, Measuring the chemical potential of polymer solutions and melts in computer simulations, *J. Chem. Phys.* 100 (1994) 719–724, <https://doi.org/10.1063/1.466937>.
- [440] S.K. Kumar, I. Szleifer, A.Z. Panagiotopoulos, Determination of the chemical potentials of polymeric systems from Monte Carlo simulations, *Phys. Rev. Lett.* 66 (22) (1991) 2935–2938, <https://doi.org/10.1103/PhysRevLett.66.2935>.
- [441] S.K. Kumar, I. Szleifer, A.Z. Panagiotopoulos, Determination of the chemical potentials of polymeric systems from Monte Carlo simulations, *Phys. Rev. Lett.* 68 (24) (1992) 3658, <https://doi.org/10.1103/PhysRevLett.68.3658>.
- [442] K.K. Mon, R.B. Griffiths, Chemical potential by gradual insertion of a particle in Monte Carlo simulation, *Phys. Rev. A* 31 (2) (1985) 956–959, <https://doi.org/10.1103/physreva.31.956>.
- [443] B. Smit, G.C.A.M. Mooij, D. Frenkel, Comment on “determination of the chemical potential of polymeric systems from Monte Carlo simulations”, *Phys. Rev. Lett.* 68 (24) (1992) 3657, <https://doi.org/10.1103/PhysRevLett.68.3657>.
- [444] J. Harris, S.A. Rice, A lattice model of a supported monolayer of amphiphilic molecules: Monte Carlo simulations, *J. Chem. Phys.* 88 (1988) 1298–1306, <https://doi.org/10.1063/1.454200>.
- [445] J.I. Siepmann, A method for the direct calculation of chemical potentials for dense chain systems, *Mol. Phys.* 70 (6) (1990) 1145–1158, <https://doi.org/10.1080/00268979000101591>.
- [446] D. Frenkel, B. Smit, Unexpected length dependence of the solubility of chain molecules, *Mol. Phys.* 75 (5) (1992) 983–988, <https://doi.org/10.1080/00268979200100761>.
- [447] D. Frenkel, G.C.A.M. Mooij, B. Smit, Novel scheme to study structural and thermal-properties of continuously deformable molecules, *J. Phys. Condens. Matter* 4 (12) (1992) 3053–3076, <https://doi.org/10.1088/0953-8984/4/12/006>.
- [448] J.J. de Pablo, M. Laso, U.W. Suter, Estimation of the chemical potential of chain molecules by simulation, *J. Chem. Phys.* 96 (1992) 6157–6162, <https://doi.org/10.1063/1.462658>.
- [449] K. Kremer, K. Binder, Monte Carlo simulations of lattice models for macromolecules, *Comput. Phys. Rep.* 7 (6) (1988) 259–310, [https://doi.org/10.1016/0167-7977\(88\)90015-9](https://doi.org/10.1016/0167-7977(88)90015-9).
- [450] J. Batoulis, K. Kremer, Statistical properties of biased sampling methods for long polymer-chains, *J. Phys. A, Math. Gen.* 21 (1) (1988) 127–146, <https://doi.org/10.1088/0305-4470/21/1/020>.
- [451] E.J. Maginn, A.T. Bell, D.N. Theodorou, Sorption thermodynamics, siting and conformation of long n-alkanes in silicalite as predicted by configurational-bias Monte Carlo integration, *J. Phys. Chem.* 99 (7) (1995) 2057–2079, <https://doi.org/10.1021/j100007a042>.
- [452] B. Smit, J.I. Siepmann, Simulating the adsorption of alkanes in zeolites, *Science* 264 (5162) (1994) 1118–1120, <https://doi.org/10.1126/science.264.5162.1118>.
- [453] B. Smit, J.I. Siepmann, Computer simulations of the energetics and siting of n-alkanes in zeolites, *J. Phys. Chem.* 98 (34) (1994) 8442–8452, <https://doi.org/10.1021/j100085a027>.
- [454] B. Smit, Grand-canonical Monte Carlo simulations of chain molecules: adsorption isotherms of alkanes in zeolites, *Mol. Phys.* 85 (1) (1995) 153–172, <https://doi.org/10.1080/0026897950010101>.
- [455] G.C.A.M. Mooij, D. Frenkel, The overlapping distribution method to compute chemical-potentials of chain molecules, *J. Phys. Condens. Matter* 6 (21) (1994) 3879–3888, <https://doi.org/10.1088/0953-8984/6/21/012>.
- [456] P. Grassberger, Monte-Carlo simulations of 3d self-avoiding walks, *J. Phys. A, Math. Gen.* 26 (12) (1993) 2769–2776, <https://doi.org/10.1088/0305-4470/26/12/017>.
- [457] P. Grassberger, R. Hegger, Monte-Carlo simulations of off-lattice polymers, *J. Phys. Condens. Matter* 7 (16) (1995) 3089–3097, <https://doi.org/10.1088/0953-8984/7/16/005>.
- [458] T. Garel, H. Orland, Guided replication of random chains - a new Monte-Carlo method, *J. Phys. A, Math. Gen.* 23 (12) (1990) L621–L626, <https://doi.org/10.1088/0305-4470/23/12/007>.
- [459] D. Frenkel, Numerical techniques to study complex liquids, in: Marc Baus, Luis F. Rull, Jean-Paul Ryckaert (Eds.), *Observation, Prediction and Simulation of Phase Transitions in Complex Fluids*, in: NATO ASI Series, vol. 460, Kluwer, Dordrecht, 1995, pp. 357–419.

- [460] P. Grassberger, Pruned-enriched Rosenbluth method: simulations of theta polymers of chain length up to 1,000,000, Phys. Rev. E 56 (3) (1997) 3682–3693, <https://doi.org/10.1103/PhysRevE.56.3682>.
- [461] H. Frauenkron, U. Bastolla, E. Gerstner, P. Grassberger, W. Nadler, New Monte Carlo algorithm for protein folding, Phys. Rev. Lett. 80 (14) (1998) 3149–3152, <https://doi.org/10.1103/PhysRevLett.80.3149>.
- [462] H. Meirovitch, Scanning method as an unbiased simulation technique and its application to the study of self-attracting random walks, Phys. Rev. A 32 (6) (1985) 3699–3708, <https://doi.org/10.1103/PhysRevA.32.3699>.
- [463] P.J. Steinbach, B.R. Brooks, New spherical-cutoff methods for long-range forces in macromolecular simulation, J. Comput. Chem. 15 (7) (1994) 667–683, <https://doi.org/10.1002/jcc.540150702>.
- [464] A. Arnold, F. Fahrenberger, C. Holm, O. Lenz, M. Bolten, H. Dachsel, R. Halver, I. Kabadshow, F. Gahler, F. Heber, J. Isringhausen, M. Hofmann, M. Pippig, D. Potts, G. Sutmann, Comparison of scalable fast methods for long-range interactions, Phys. Rev. E 88 (6) (2013), <https://doi.org/10.1103/PhysRevE.88.063308>.
- [465] V. Ballenegger, Communication: on the origin of the surface term in the Ewald formula, J. Chem. Phys. 140 (16) (2014), <https://doi.org/10.1063/1.4872019>.
- [466] P.P. Ewald, Die berechnung optischer und elektrostatischer gitterpotentiale, Ann. Phys. 64 (1921) 253–287, <https://doi.org/10.1002/andp.19213690304>.
- [467] M.P. Tosi, Cohesion of ionic solids in the Born model, in: F. Seitz, D. Turnbull (Eds.), Solid State Physics: Advances in Research and Applications, Plenum, New York, 1964, pp. 1–120.
- [468] S.G. Brush, H.L. Sahlin, E. Teller, Monte Carlo study of a 1-component plasma I, J. Chem. Phys. 45 (6) (1966) 2102–2118, <https://doi.org/10.1063/1.1727895>.
- [469] S.W. de Leeuw, J.W. Perram, E.R. Smith, Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants, Proc. R. Soc. Lond. A 373 (1980) 27–56, <https://doi.org/10.1098/rspa.1980.0135>.
- [470] S.W. de Leeuw, J.W. Perram, E.R. Smith, Simulation of electrostatic systems in periodic boundary conditions. II. Equivalence of boundary conditions, Proc. R. Soc. Lond. A 373 (1980) 56–66, <https://doi.org/10.1098/rspa.1980.0136>.
- [471] S.W. de Leeuw, J.W. Perram, E.R. Smith, Simulation of electrostatic systems in periodic boundary conditions. III. Further theory and applications, Proc. R. Soc. Lond. A 388 (1983) 177–193, <https://doi.org/10.1098/rspa.1983.0077>.
- [472] J.-P. Hansen, Molecular-dynamics simulations of Coulomb systems in two and three dimensions, in: G. Ciccotti, W.G. Hoover (Eds.), Molecular Dynamics Simulations of Statistical Mechanics Systems, North-Holland, Amsterdam, 1986, pp. 89–129.
- [473] D.M. Heyes, F. van Swol, The electrostatic potential and field in the surface region of lamina and semi-infinite point-charge lattices, J. Chem. Phys. 75 (10) (1981) 5051–5058, <https://doi.org/10.1063/1.441896>.
- [474] M. Neumann, Dipole moment fluctuation formulas in computer simulations of polar systems, Mol. Phys. 50 (4) (1983) 841–858, <https://doi.org/10.1080/00268978300102721>.
- [475] M. Sprik, Finite Maxwell field and electric displacement Hamiltonians derived from a current dependent Lagrangian, Mol. Phys. 116 (21–22) (2018) 3114–3120, <https://doi.org/10.1080/00268976.2018.1431406>.
- [476] J.A. Kolafa, J.W. Perram, Cutoff errors in the Ewald summation formulae for point charge systems, Mol. Simul. 9 (5) (1992) 351–368, <https://doi.org/10.1080/0892702908049126>.
- [477] H.G. Petersen, Accuracy and efficiency of the particle mesh Ewald method, J. Chem. Phys. 103 (9) (1995) 3668–3679, <https://doi.org/10.1063/1.470043>.
- [478] K. Esselink, A comparison of algorithms for long-range interactions, Comput. Phys. Commun. 87 (3) (1995) 375–395, [https://doi.org/10.1016/0010-4655\(95\)00003-X](https://doi.org/10.1016/0010-4655(95)00003-X).
- [479] M. Deserno, C. Holm, How to mesh up Ewald sums. I. A theoretical and numerical comparison of various particle mesh routines, J. Chem. Phys. 109 (18) (1998) 7678–7693, <https://doi.org/10.1063/1.477414>.

680 Bibliography

- [480] T.A. Darden, D. York, L. Pedersen, Particle mesh Ewald: an $N \log(N)$ method for Ewald sums in large systems, *J. Chem. Phys.* 98 (12) (1993) 10089–10092, <https://doi.org/10.1063/1.464397>.
- [481] U. Essmann, L. Perera, M.L. Berkowitz, T.A. Darden, H. Lee, L. Pedersen, A smooth particle mesh Ewald method, *J. Chem. Phys.* 103 (19) (1995) 8577–8593, <https://doi.org/10.1063/1.470117>.
- [482] E.L. Pollock, J. Glosli, Comments on P^3M , FMM, and the Ewald method for large periodic Coulombic systems, *Comput. Phys. Commun.* 95 (1996) 93–110, [https://doi.org/10.1016/0010-4655\(96\)00043-4](https://doi.org/10.1016/0010-4655(96)00043-4).
- [483] C. Sagui, T.A. Darden, Molecular dynamics simulations of biomolecules: long-range electrostatic effects, *Annu. Rev. Biophys. Biomol. Struct.* 28 (1999) 155–179, <https://doi.org/10.1146/annurev.biophys.28.1.155>.
- [484] J.V.L. Beckers, C.P. Lowe, S.W. de Leeuw, An iterative pppm method for simulating coulombic systems on distributed memory parallel computers, *Mol. Simul.* 20 (6) (1998) 369–383, <https://doi.org/10.1080/08927029808022044>.
- [485] M. Deserno, C. Holm, How to mesh up Ewald sums. II. An accurate error estimate for the particle-particle-particle-mesh algorithm, *J. Chem. Phys.* 109 (18) (1998) 7694–7701, <https://doi.org/10.1063/1.477415>.
- [486] D. Wolf, P. Kebinski, S.R. Phillpot, J. Eggebrecht, Exact method for the simulation of coulombic systems by spherically truncated, pairwise $r(-1)$ summation, *J. Chem. Phys.* 110 (17) (1999) 8254–8282, <https://doi.org/10.1063/1.478738>.
- [487] J.H.R. Clarke, D. Brown, Molecular dynamics computer simulation of chain molecule liquids I. The coupling of torsional motions to translational diffusion, *Mol. Phys.* 58 (1986) 815–825, <https://doi.org/10.1080/00268978600101591>.
- [488] I. Fukuda, N. Kamiya, Y. Yonezawa, H. Nakamura, Simple and accurate scheme to compute electrostatic interaction: zero-dipole summation technique for molecular system and application to bulk water, *J. Chem. Phys.* 137 (5) (2012), <https://doi.org/10.1063/1.4739789>.
- [489] F.N. Mendoza, J. Lopez-Lemus, G.A. Chapela, J. Alejandre, The Wolf method applied to the liquid-vapor interface of water, *J. Chem. Phys.* 129 (2) (2008), <https://doi.org/10.1063/1.2948951>.
- [490] A.W. Appel, An efficient program for many-body simulation, *SIAM J. Sci. Stat. Comput.* 6 (1) (1985) 85–103, <https://doi.org/10.1137/0906008>.
- [491] J. Barnes, P. Hut, A hierarchical $\mathcal{O}(N \log N)$ force-calculation algorithm, *Nature* 324 (1986) 446–449, <https://doi.org/10.1038/324446a0>.
- [492] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* 73 (2) (1987) 325–348, [https://doi.org/10.1016/0021-9991\(87\)90140-9](https://doi.org/10.1016/0021-9991(87)90140-9).
- [493] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, *J. Comput. Phys.* 155 (2) (1999) 468–498, <https://doi.org/10.1006/jcph.1999.6355>.
- [494] K.E. Schmidt, Michael A. Lee, Implementing the fast multipole method in three dimensions, *J. Stat. Phys.* 63 (5–6) (1991) 1223–1235, <https://doi.org/10.1007/bf01030008>.
- [495] N. Yoshii, Y. Andoh, S. Okazaki, Fast multipole method for three-dimensional systems with periodic boundary condition in two directions, *J. Comput. Chem.* 41 (9) (2020) 940–948, <https://doi.org/10.1002/jcc.26141>.
- [496] J. Kurzak, B.M. Pettitt, Fast multipole methods for particle dynamics, *Mol. Simul.* 32 (10–11) (2006) 775–790, <https://doi.org/10.1080/08927020600991161>.
- [497] W.D. Elliott, J.A. Board, Fast Fourier transform accelerated fast multipole algorithm, *SIAM J. Sci. Comput.* 17 (2) (1996) 398–415, <https://doi.org/10.1137/S1064827594264259>.
- [498] J.D. Jackson, *Classical Electrodynamics*, 3rd edition, Wiley, New York, 1999.
- [499] X. Zhao, S. Subrahmanyam, K.B. Eisenthal, Orientational fluctuations and phase transitions of long chain molecules at the air/water interface, *Phys. Rev. Lett.* 67 (1991) 2025–2028, <https://doi.org/10.1103/PhysRevLett.67.2025>.
- [500] P. Chartier, E. Darrigrand, E. Faou, A regular fast multipole method for geometric numerical integrations of Hamiltonian systems, *BIT Numer. Math.* 50 (1) (2010) 23–40, <https://doi.org/10.1007/s10543-010-0248-6>.

- [501] D.S. Shamshirgar, R. Yokota, A.K. Tornberg, B. Hess, Regularizing the fast multipole method for use in molecular simulation, *J. Chem. Phys.* 151 (23) (2019), <https://doi.org/10.1063/1.5122859>.
- [502] A.C. Maggs, Auxiliary field Monte Carlo for charged particles, *J. Chem. Phys.* 120 (7) (2004) 3108–3118, <https://doi.org/10.1063/1.1642587>.
- [503] J. Rottler, A.C. Maggs, Long-ranged electrostatics from local algorithms, *Soft Matter* 7 (7) (2011) 3260–3267, <https://doi.org/10.1039/c0sm01057>.
- [504] L. Levrel, A.C. Maggs, Monte Carlo algorithms for charged lattice gases, *Phys. Rev. E* 72 (1) (2005), <https://doi.org/10.1103/PhysRevE.72.016715>.
- [505] I. Pasichnyk, B. Dunweg, Coulomb interactions via local dynamics: a molecular-dynamics algorithm, *J. Phys. Condens. Matter* 16 (38) (2004) S3999–S4020, <https://doi.org/10.1088/0953-8984/16/38/017>.
- [506] J.M. Caillol, D. Levesque, Numerical simulations of homogeneous and inhomogeneous ionic systems - an efficient alternative to the Ewald method, *J. Chem. Phys.* 94 (1) (1991) 597–607, <https://doi.org/10.1063/1.460326>.
- [507] B.W.H. van Beest, G.J. Kramer, R.A. van Santen, Force fields for silicas and aluminophosphates based on ab initio calculations, *Phys. Rev. Lett.* 64 (16) (1990) 1955–1958, <https://doi.org/10.1103/PhysRevLett.64.1955>.
- [508] A.K. Kron, The Monte Carlo method in statistical calculations of macromolecules, *Polym. Sci. USSR* 7 (7) (1965) 1361–1367, [https://doi.org/10.1016/0032-3950\(65\)90209-1](https://doi.org/10.1016/0032-3950(65)90209-1).
- [509] J. Houdayer, The wormhole move: a new algorithm for polymer simulations, *J. Chem. Phys.* 116 (5) (2002) 1783–1787, <https://doi.org/10.1063/1.1433500>.
- [510] J.I. Siepmann, D. Frenkel, Configurational bias Monte-Carlo - a new sampling scheme for flexible chains, *Mol. Phys.* 75 (1) (1992) 59–70, <https://doi.org/10.1080/00268979200100061>.
- [511] J.J. de Pablo, M. Laso, U.W. Suter, Simulation of polyethylene above and below the melting point, *J. Chem. Phys.* 96 (1992) 2395–2403, <https://doi.org/10.1063/1.462037>.
- [512] G.C.A.M. Mooij, D. Frenkel, B. Smit, Direct simulation of phase equilibria of chain molecules, *J. Phys. Condens. Matter* 4 (16) (1992) L255–L259, <https://doi.org/10.1088/0953-8984/4/16/001>.
- [513] G.C.A.M. Mooij, D. Frenkel, A systematic optimization scheme for configurational bias Monte Carlo, *Mol. Simul.* 17 (1) (1996) 41–55, <https://doi.org/10.1080/08927029608024093>.
- [514] B. Chen, J.I. Siepmann, Transferable potentials for phase equilibria. 3. Explicit-hydrogen description of normal alkanes, *J. Phys. Chem. B* 103 (25) (1999) 5370–5379, <https://doi.org/10.1021/jp990822m>.
- [515] M.D. Macedonia, E.J. Maginn, A biased grand canonical Monte Carlo method for simulating adsorption using all-atom and branched united atom models, *Mol. Phys.* 96 (9) (1999) 1375–1390, <https://doi.org/10.1080/00268979909483082>.
- [516] D. Chandler, P.G. Wolynes, Exploiting the isomorphism between quantum theory and classical statistical mechanics of polyatomic fluids, *J. Chem. Phys.* 74 (1981) 4078–4095, <https://doi.org/10.1063/1.441588>.
- [517] V.G. Mavrantzas, T.D. Boone, E. Zervopoulou, D.N. Theodorou, End-bridging Monte Carlo: a fast algorithm for atomistic simulation of condensed phases of long polymer chains, *Macromolecules* 32 (15) (1999) 5072–5096, <https://doi.org/10.1021/ma981745g>.
- [518] M. Dijkstra, D. Frenkel, Evidence for entropy-driven demixing in hard-core fluids, *Phys. Rev. Lett.* 72 (2) (1994) 298–300, <https://doi.org/10.1103/PhysRevLett.72.298>.
- [519] M. Dijkstra, D. Frenkel, J.P. Hansen, Phase-separation in binary hard-core mixtures, *J. Chem. Phys.* 101 (4) (1994) 3179–3189, <https://doi.org/10.1063/1.468468>.
- [520] Hiromi Yamakawa, *Modern Theory of Polymer Solutions*, Harper & Row, New York, 1971.
- [521] M. Vendruscolo, Modified configurational bias Monte Carlo method for simulation of polymer systems, *J. Chem. Phys.* 106 (1996) 2970–2976, <https://doi.org/10.1063/1.473356>.

682 Bibliography

- [522] C.D. Wick, J.I. Siepmann, Self-adapting fixed-end-point configurational-bias Monte Carlo method for the regrowth of interior segments of chain molecules with strong intramolecular interactions, *Macromolecules* 33 (19) (2000) 7207–7218, <https://doi.org/10.1021/ma000172g>.
- [523] Z. Chen, F.A. Escobedo, A configurational-bias approach for the simulation of inner sections of linear and cyclic molecules, *J. Chem. Phys.* 113 (24) (2000) 11382–11392, <https://doi.org/10.1063/1.1328069>.
- [524] T. Biben, Structure et stabilité des fluides à deux composants: des fluides atomiques aux suspensions colloïdales, Ph.D. thesis, Ecole Normal Supérieure de Lyon, 1993.
- [525] T. Biben, P. Bladon, D. Frenkel, Depletion effects in binary hard-sphere fluids, *J. Phys. Condens. Matter* 8 (50) (1996) 10799–10821, <https://doi.org/10.1088/0953-8984/8/50/008>.
- [526] P.G. Bolhuis, D. Frenkel, Numerical study of the phase diagram of a mixture of spherical and rodlike colloids, *J. Chem. Phys.* 101 (11) (1994) 9869–9875, <https://doi.org/10.1063/1.467953>.
- [527] J.C. Shelley, G.N. Patey, A configurational bias Monte Carlo method for ionic solutions, *J. Chem. Phys.* 100 (1994) 8265–8270, <https://doi.org/10.1063/1.466770>.
- [528] K. Esselink, L.D.J.C. Loyens, B. Smit, Parallel Monte Carlo simulations, *Phys. Rev. E* 51 (2) (1995) 1560–1568, <https://doi.org/10.1103/PhysRevE.51.1560>.
- [529] L.D.J.C. Loyens, B. Smit, K. Esselink, Parallel Gibbs-ensemble simulations, *Mol. Phys.* 86 (2) (1995) 171–183, <https://doi.org/10.1080/00268979500101921>.
- [530] K. Esselink, P.A.J. Hilbers, S. Karaborni, J.I. Siepmann, B. Smit, Simulating complex fluids, *Mol. Simul.* 14 (4–5) (1995) 259–274, <https://doi.org/10.1080/08927029508022022>.
- [531] T.J.H. Vlugt, M.G. Martin, B. Smit, J.I. Siepmann, R. Krishna, Improving the efficiency of the cbmc algorithm, *Mol. Phys.* 94 (4) (1998) 727–733, <https://doi.org/10.1080/002689798167881>.
- [532] B. Smit, T.L.M. Maesen, Commensurate “freezing” of alkanes in the channels of a zeolite, *Nature* 374 (6517) (1995) 42–44, <https://doi.org/10.1038/374039a0>.
- [533] R. Evans, Microscopic theories of simple fluids and their interfaces, in: J. Charvolin, J.F. Joanny, J. Zinn-Justin (Eds.), *Liquides aux Interfaces (Liquids at Interfaces)*, North Holland, Amsterdam, 1990, pp. 1–98.
- [534] S.J. Gregg, K.S.W. Sing, *Adsorption, Surface Area and Porosity*, Academic Press, London, 1982.
- [535] J.I. Siepmann, S. Karaborni, B. Smit, Simulating the critical properties of complex fluids, *Nature* 365 (6444) (1993) 330–332, <https://doi.org/10.1038/365330a0>.
- [536] W.J.M. van Well, J.P. Wolthuizen, B. Smit, J.H.C. van Hooff, R.A. van Santen, Commensurate freezing of n-alkanes in silicalite, *Angew. Chem., Int. Ed. Engl.* 34 (22) (1995) 2543–2544, <https://doi.org/10.1002/anie.199525431>.
- [537] R. Krishna, B. Smit, T.J.H. Vlugt, Sorption-induced diffusion-selective separation of hydrocarbon isomers using silicalite, *J. Phys. Chem. A* 102 (40) (1998) 7727–7730, <https://doi.org/10.1021/jp982438f>.
- [538] T.J.H. Vlugt, R. Krishna, B. Smit, Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite, *J. Phys. Chem. B* 103 (7) (1999) 1102–1118, <https://doi.org/10.1021/jp982736c>.
- [539] S. Consta, N.B. Wilding, D. Frenkel, Z. Alexandrowicz, Recoil growth: an efficient simulation method for multi-polymer systems, *J. Chem. Phys.* 110 (6) (1999) 3220–3228, <https://doi.org/10.1063/1.477844>.
- [540] S. Consta, T.J.H. Vlugt, J.W. Hoeth, B. Smit, D. Frenkel, Recoil growth algorithm for chain molecules with continuous interactions, *Mol. Phys.* 97 (12) (1999) 1243–1254, <https://doi.org/10.1080/00268979909482926>.
- [541] H. Meirovitch, Statistical properties of the scanning simulation method for polymer-chains, *J. Chem. Phys.* 89 (4) (1988) 2514–2522, <https://doi.org/10.1063/1.455045>.
- [542] Z. Alexandrowicz, Simulation of polymers with rebound selection, *J. Chem. Phys.* 109 (13) (1998) 5622–5626, <https://doi.org/10.1063/1.477180>.

- [543] Y. Iba, Extended ensemble Monte Carlo, *Int. J. Mod. Phys. C* 12 (5) (2001) 623–656, <https://doi.org/10.1142/S021983101001912>.
- [544] R.H. Swendsen, J.S. Wang, Replica Monte Carlo simulation of spin glasses, *Phys. Rev. Lett.* 57 (21) (1986) 2607–2609, <https://doi.org/10.1103/PhysRevLett.57.2607>.
- [545] D.D. Frantz, D.L. Freeman, J.D. Doll, Reducing quasi-ergodic behaviour in Monte Carlo simulations by J-walking: application to atomic clusters, *J. Chem. Phys.* 93 (4) (1990) 2769–2784, <https://doi.org/10.1063/1.458863>.
- [546] A.P. Lyubartsev, A.A. Martsinovski, S.V. Shevkunov, P.N. Vorontsov-Vel'yaminov, New approach to Monte-Carlo calculation of the free-energy - method of expanded ensembles, *J. Chem. Phys.* 96 (3) (1992) 1776–1783, <https://doi.org/10.1063/1.462133>.
- [547] E. Marinari, G. Parisi, Simulated tempering - a new Monte-Carlo scheme, *Europhys. Lett.* 19 (6) (1992) 451–458, <https://doi.org/10.1209/0295-5075/19/6/002>.
- [548] C.J. Geyer, E.A. Thompson, Annealing Markov chain Monte Carlo with applications to the ancestral inference, *J. Am. Stat. Assoc.* 90 (431) (1995) 909–920, <https://doi.org/10.2307/2291325>.
- [549] Yuji Sugita, Yuko Okamoto, Replica-exchange molecular dynamics method for protein folding, *Chem. Phys. Lett.* 314 (1–2) (1999) 141–151, [https://doi.org/10.1016/S0009-2614\(99\)01123-9](https://doi.org/10.1016/S0009-2614(99)01123-9).
- [550] M. Falcioni, M.W. Deem, A biased Monte Carlo scheme for zeolite structure solution, *J. Chem. Phys.* 110 (3) (1999) 1754–1766, <https://doi.org/10.1063/1.477812>.
- [551] N. Plattner, J.D. Doll, P. Dupuis, H. Wang, Y.F. Liu, J.E. Gubernatis, An infinite swapping approach to the rare-event sampling problem, *J. Chem. Phys.* 135 (13) (2011), <https://doi.org/10.1063/1.3643325>.
- [552] T.Q. Yu, J.F. Lu, C.F. Abrams, E. Vanden-Eijnden, Multiscale implementation of infinite-swap replica exchange molecular dynamics, *Proc. Natl. Acad. Sci. USA* 113 (42) (2016) 11744–11749, <https://doi.org/10.1073/pnas.1605089113>.
- [553] Q.L. Yan, J.J. de Pablo, Hyper-parallel tempering Monte Carlo: application to the Lennard-Jones fluid and the restricted primitive model, *J. Chem. Phys.* 111 (21) (1999) 9509–9516, <https://doi.org/10.1063/1.480282>.
- [554] Q.L. Yan, J.J. de Pablo, Hyperparallel tempering Monte Carlo simulation of polymeric systems, *J. Chem. Phys.* 113 (3) (2000) 1276–1282, <https://doi.org/10.1063/1.481905>.
- [555] A. Bunker, B. Dunweg, Parallel excluded volume tempering for polymer melts, *Phys. Rev. E* 63 (1 Pt 2) (2001) 016701, <https://doi.org/10.1103/PhysRevE.63.016701>.
- [556] S. Auer, D. Frenkel, Prediction of absolute crystal-nucleation rate in hard-sphere colloids, *Nature* 409 (6823) (2001) 1020–1023, <https://doi.org/10.1038/35059035>.
- [557] S. Trebst, E. Gull, M. Troyer, Optimized ensemble Monte Carlo simulations of dense Lennard-Jones fluids, *J. Chem. Phys.* 123 (20) (2005) 204501, <https://doi.org/10.1063/1.2121709>.
- [558] P. Dayal, S. Trebst, S. Wessel, D. Wurtz, M. Troyer, S. Sabhapandit, S.N. Coppersmith, Performance limitations of flat-histogram methods, *Phys. Rev. Lett.* 92 (9) (2004) 097201, <https://doi.org/10.1103/PhysRevLett.92.097201>.
- [559] F.A. Escobedo, Simulation of the density of states in isothermal and adiabatic ensembles, *Phys. Rev. E* 73 (5 Pt 2) (2006) 056701, <https://doi.org/10.1103/PhysRevE.73.056701>.
- [560] F.A. Escobedo, F.J. Martinez-Veracoechea, Optimized expanded ensembles for simulations involving molecular insertions and deletions. I. Closed systems, *J. Chem. Phys.* 127 (17) (2007) 174103, <https://doi.org/10.1063/1.2800320>.
- [561] F.A. Escobedo, F.J. Martinez-Veracoechea, Optimization of expanded ensemble methods, *J. Chem. Phys.* 129 (15) (2008) 154107, <https://doi.org/10.1063/1.2994717>.
- [562] D.M. Ceperley, M. Dewing, The penalty method for random walks with uncertain energies, *J. Chem. Phys.* 110 (20) (1999) 9812–9820, <https://doi.org/10.1063/1.478034>.
- [563] D. Frenkel, K.J. Schrenk, S. Martiniani, Monte Carlo sampling for stochastic weight functions, *Proc. Natl. Acad. Sci. USA* 114 (27) (2017) 6924–6929, <https://doi.org/10.1073/pnas.1620497114>.

684 Bibliography

- [564] S. Duane, A. Kennedy, B.J. Pendleton, D. Roweth, Hybrid Monte Carlo, Phys. Lett. B 195 (2) (1987) 216–222, [https://doi.org/10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X).
- [565] B. Mehlig, D.W. Heermann, B.M. Forrest, Exact Langevin algorithms, Mol. Phys. 76 (6) (1992) 1347–1357, <https://doi.org/10.1080/00268979200102131>.
- [566] B. Mehlig, D.W. Heermann, B.M. Forrest, Hybrid Monte Carlo method for condensed-matter systems, Phys. Rev. B 45 (2) (1992) 679–685, <https://doi.org/10.1103/PhysRevB.45.679>.
- [567] B.M. Forrest, U.W. Suter, Generalized coordinate hybrid Monte Carlo, Mol. Phys. 82 (2) (1994) 393–410, <https://doi.org/10.1080/00268979400100304>.
- [568] C. Jacoboni, L. Reggiani, The Monte-Carlo method for the solution of charge transport in semiconductors with applications to covalent materials, Rev. Mod. Phys. 55 (3) (1983) 645–705, <https://doi.org/10.1103/RevModPhys.55.645>.
- [569] D.T. Gillespie, Exact stochastic simulation of coupled chemical-reactions, J. Phys. Chem. 81 (25) (1977) 2340–2361, <https://doi.org/10.1021/j100540a008>.
- [570] A.B. Bortz, M.H. Kalos, J.L. Lebowitz, New algorithm for Monte-Carlo simulation of Ising spin systems, J. Comput. Phys. 17 (1) (1975) 10–18, [https://doi.org/10.1016/0021-9991\(75\)90060-1](https://doi.org/10.1016/0021-9991(75)90060-1).
- [571] Alastair J. Walker, An efficient method for generating discrete random variables with general distributions, ACM Trans. Math. Softw. 3 (3) (1977) 253–256, <https://doi.org/10.1145/355744.355749>.
- [572] Keith Schwarz, <https://www.keithschwarz.com/darts-dice-coins/>.
- [573] M. Merolle, J.P. Garrahan, D. Chandler, Space-time thermodynamics of the glass transition, Proc. Natl. Acad. Sci. USA 102 (31) (2005) 10837–10840, <https://doi.org/10.1073/pnas.0504820102>.
- [574] D.M. Ceperley, G.V. Chester, M.H. Kalos, Monte-Carlo simulation of a many-fermion study, Phys. Rev. B 16 (7) (1977) 3081–3099, <https://doi.org/10.1103/PhysRevB.16.3081>.
- [575] D. Frenkel, Speed-up of Monte Carlo simulations by sampling of rejected states, Proc. Natl. Acad. Sci. USA 101 (51) (2004) 17571–17575, <https://doi.org/10.1073/pnas.0407950101>.
- [576] G.C. Boulougouris, D. Frenkel, Monte Carlo sampling of a Markov web, J. Chem. Theory Comput. 1 (3) (2005) 389–393, <https://doi.org/10.1021/ct049900m>.
- [577] A.A. Barker, Monte Carlo calculations of radial distribution functions for a proton-electron plasma, Aust. J. Phys. 18 (2) (1965) 119, <https://doi.org/10.1071/Ph650119>.
- [578] D. Frenkel, Waste-recycling Monte Carlo, in: M. Ferrario, G. Ciccotti, K. Binder (Eds.), Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology, vol. 1, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 127–137.
- [579] G. Adjanor, M. Athenes, J.M. Rodgers, Waste-recycling Monte Carlo with optimal estimates: application to free energy calculations in alloys, J. Chem. Phys. 135 (4) (2011), <https://doi.org/10.1063/1.3610423>.
- [580] I. Coluzza, D. Frenkel, Virtual-move parallel tempering, ChemPhysChem 6 (9) (2005) 1779–1783, <https://doi.org/10.1002/cphc.200400629>.
- [581] C. Jarzynski, Targeted free energy perturbation, Phys. Rev. E 65 (4) (2002), <https://doi.org/10.1103/PhysRevE.65.046122>.
- [582] A.J. Schultz, S.G. Moustafa, W.S. Lin, S.J. Weinstein, D.A. Kofke, Reformulation of ensemble averages via coordinate mapping, J. Chem. Theory Comput. 12 (4) (2016) 1491–1498, <https://doi.org/10.1021/acs.jctc.6b00018>.
- [583] F. Noe, S. Olsson, J. Kohler, H. Wu, Boltzmann generators: sampling equilibrium states of many-body systems with deep learning, Science 365 (6457) (2019) 1001, <https://doi.org/10.1126/science.aaw1147>.
- [584] P. Wirnsberger, A.J. Ballard, G. Papamakarios, S. Abercrombie, S. Racaniere, A. Pritzel, D.J. Rezende, C. Blundell, Targeted free energy estimation via learned mappings, J. Chem. Phys. 153 (14) (2020), <https://doi.org/10.1063/5.0018903>.
- [585] M. Gabrie, G.M. Rotskoff, E. Vanden-Eijnden, Adaptive Monte Carlo augmented with normalizing flows, Proc. Natl. Acad. Sci. USA 119 (10) (2022), <https://doi.org/10.1073/pnas.2109420119>.

- [586] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed, B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, *J. Mach. Learn. Res.* 22 (2021).
- [587] S. Kullback, R.A. Leibler, On information and sufficiency, *Ann. Math. Stat.* 22 (1) (1951) 79–86, <https://doi.org/10.1214/aoms/1177729694>.
- [588] K. Muller, L.D. Brown, Location of saddle points and minimum energy paths by a constrained simplex optimization procedure, *Theor. Chim. Acta* 53 (1) (1979) 75–93, <https://doi.org/10.1007/BF00547608>.
- [589] P. Wirnsberger, G. Papamakarios, B. Ibarz, S. Racaniere, A.J. Ballard, A. Pritzel, C. Blundell, Normalizing flows for atomic solids, *Mach. Learn.: Sci. Technol.* 3 (2) (2022), <https://doi.org/10.1088/2632-2153/ac6b16>.
- [590] A. Coretti, S. Falkner, P.L. Geissler, C. Dellago, Learning mappings between equilibrium states of liquid systems using normalizing flows, arXiv, arXiv:2208.10420, 2022, <http://dx.doi.org/10.48550/arXiv.2208.10420>.
- [591] R.H. Swendsen, J.S. Wang, Nonuniversal critical dynamics in Monte Carlo simulations, *Phys. Rev. Lett.* 58 (2) (1987) 86–88, <https://doi.org/10.1103/PhysRevLett.58.86>.
- [592] U. Wolff, Collective Monte-Carlo updating for spin systems, *Phys. Rev. Lett.* 62 (4) (1989) 361–364, <https://doi.org/10.1103/PhysRevLett.62.361>.
- [593] C. Dress, W. Krauth, Cluster algorithm for hard spheres and related systems, *J. Phys. A, Math. Gen.* 28 (23) (1995) L597–L601, <https://doi.org/10.1088/0305-4470/28/23/001>.
- [594] J.W. Liu, E. Luijten, Rejection-free geometric cluster algorithm for complex fluids, *Phys. Rev. Lett.* 92 (3) (2004), <https://doi.org/10.1103/PhysRevLett.92.035504>.
- [595] D. Frenkel, Advanced Monte Carlo techniques, in: M.P. Allen, D.J. Tildesley (Eds.), *Computer Simulation in Chemical Physics*, in: NATO ASI Series, vol. 393, Kluwer Academic Publishers, Dordrecht, 1993, pp. 93–152.
- [596] S. Whitelam, P.L. Geissler, Avoiding unphysical kinetic traps in Monte Carlo simulations of strongly attractive particles, *J. Chem. Phys.* 127 (15) (2007), <https://doi.org/10.1063/1.2790421>.
- [597] S. Whitelam, P.L. Geissler, Avoiding unphysical kinetic traps in Monte Carlo simulations of strongly attractive particles (vol 127, art no 154101, 2007), *J. Chem. Phys.* 128 (21) (2008), <https://doi.org/10.1063/1.2931944>.
- [598] E.P. Bernard, W. Krauth, D.B. Wilson, Event-chain Monte Carlo algorithms for hard-sphere systems, *Phys. Rev. E* 80 (5) (2009), <https://doi.org/10.1103/PhysRevE.80.056704>.
- [599] M. Michel, S.C. Kapfer, W. Krauth, Generalized event-chain Monte Carlo: constructing rejection-free global-balance algorithms from infinitesimal steps, *J. Chem. Phys.* 140 (5) (2014), <https://doi.org/10.1063/1.4863991>.
- [600] E.P. Bernard, W. Krauth, Event-chain Monte Carlo algorithms for hard-sphere systems (vol 80, 056704, 2009), *Phys. Rev. E* 86 (1) (2012), <https://doi.org/10.1103/PhysRevE.86.017701>.
- [601] S.C. Kapfer, W. Krauth, Cell-veto Monte Carlo algorithm for long-range systems, *Phys. Rev. E* 94 (3) (2016), <https://doi.org/10.1103/PhysRevE.94.031302>.
- [602] E.A.J.F. Peters, G. de With, Rejection-free Monte Carlo sampling for general potentials, *Phys. Rev. E* 85 (2) (2012), <https://doi.org/10.1103/PhysRevE.85.026703>.
- [603] S.C. Kapfer, W. Krauth, Irreversible local Markov chains with rapid convergence towards equilibrium, *Phys. Rev. Lett.* 119 (24) (2017), <https://doi.org/10.1103/PhysRevLett.119.240603>.
- [604] Z. Lei, W. Krauth, Mixing and perfect sampling in one-dimensional particle systems, *Europhys. Lett.* 124 (2) (2018), <https://doi.org/10.1209/0295-5075/124/20003>.
- [605] R. Car, M. Parrinello, Unified approach for molecular dynamics and density-functional theory, *Phys. Rev. Lett.* 55 (22) (1985) 2471–2474, <https://doi.org/10.1103/PhysRevLett.55.2471>.
- [606] J.P. Ryckaert, G. Ciccotti, H.J.C. Berendsen, Numerical integration of the Cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes, *J. Comput. Phys.* 23 (3) (1977) 237–341, [https://doi.org/10.1016/0021-9991\(77\)90098-5](https://doi.org/10.1016/0021-9991(77)90098-5).

686 Bibliography

- [607] Hans C. Andersen, Rattle - a velocity version of the shake algorithm for molecular-dynamics calculations, *J. Comput. Phys.* 52 (1) (1983) 24–34, [https://doi.org/10.1016/0021-9991\(83\)90014-1](https://doi.org/10.1016/0021-9991(83)90014-1).
- [608] S.W. de Leeuw, J.W. Perram, H.G. Petersen, Hamilton's equations for constrained dynamical systems, *J. Stat. Phys.* 61 (1990) 1203–1222, <https://doi.org/10.1007/BF01014372>.
- [609] G. Ciccotti, Molecular dynamics simulations of nonequilibrium phenomena and rare dynamical events, in: M. Meyer, V. Pontikis (Eds.), *Proceedings of the NATO ASI on Computer Simulation in Materials Science*, Kluwer, Dordrecht, 1991, pp. 119–137.
- [610] S. Bonella, A. Coretti, R. Vuilleumier, G. Ciccotti, Adiabatic motion and statistical mechanics via mass-zero constrained dynamics, *Phys. Chem. Chem. Phys.* 22 (19) (2020) 10775–10785, <https://doi.org/10.1039/d0cp00163e>.
- [611] Dominik Marx, Jurg Hutter, *Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods*, first paperback edition, Cambridge University Press, Cambridge, 2012.
- [612] G. Galli, A. Pasquarello, First-principle molecular dynamics, in: M.P. Allen, D.J. Tildesley (Eds.), *Computer Simulation in Chemical Physics*, Kluwer Academic Publishers, Dordrecht, 1993, pp. 261–313.
- [613] D.K. Remler, P.A. Madden, Molecular-dynamics without effective potentials via the Car-Parrinello approach, *Mol. Phys.* 70 (6) (1990) 921–966, <https://doi.org/10.1080/00268979000101451>.
- [614] M.E. Tuckerman, Ab initio molecular dynamics: basic concepts, current trends and novel applications, *J. Phys. Condens. Matter* 14 (50) (2002) R1297–R1355, <https://doi.org/10.1088/0953-8984/14/50/202>.
- [615] H. Löwen, P.A. Madden, J.-P. Hansen, Ab initio description of counterion screening in colloidal suspensions, *Phys. Rev. Lett.* 68 (1992) 1081–1084, <https://doi.org/10.1103/PhysRevLett.68.1081>.
- [616] H. Löwen, P.A. Madden, J.-P. Hansen, Nonlinear counterion screening in colloidal suspensions, *J. Chem. Phys.* 98 (1993) 3275–3289, <https://doi.org/10.1063/1.464099>.
- [617] C.G. Gray, K.E. Gubbins, *Theory of Molecular Fluids*, vol. 1, Fundamentals, The International Series of Monographs on Chemistry, vol. 9, Clarendon Press, Oxford, 1984.
- [618] M.-L. Saboungi, A. Rahman, J.W. Halley, M. Blander, Molecular dynamics studies of complexing in binary Molten salts with polarizable anions: MAX₄, *J. Chem. Phys.* 88 (1988) 5818–5823, <https://doi.org/10.1063/1.454541>.
- [619] M. Sprik, M.L. Klein, A polarizable model for water using distributed charge sites, *J. Chem. Phys.* 89 (12) (1988) 7556–7560, <https://doi.org/10.1063/1.455722>.
- [620] M. Wilson, P.A. Madden, Polarization effects in ionic systems from first principles, *J. Phys. Condens. Matter* 5 (1993) 2687–2706, <https://doi.org/10.1088/0953-8984/5/17/004>.
- [621] M. Sprik, Computer simulation of the dynamics of induced polarization fluctuations in water, *J. Chem. Phys.* 95 (6) (1991) 2283–2291, <https://doi.org/10.1021/j100159a034>.
- [622] A. Coretti, S. Bonella, G. Ciccotti, Communication: constrained molecular dynamics for polarizable models, *J. Chem. Phys.* 149 (19) (2018), <https://doi.org/10.1063/1.5055704>.
- [623] G.J. Martyna, D.J. Tobias, M.L. Klein, Constant-pressure molecular-dynamics algorithms, *J. Chem. Phys.* 101 (5) (1994) 4177–4189, <https://doi.org/10.1063/1.467468>.
- [624] P. Procacci, M. Marchi, Taming the Ewald sum in molecular dynamics simulations of solvated proteins via a multiple time step algorithm, *J. Chem. Phys.* 104 (8) (1996) 3003–3012, <https://doi.org/10.1063/1.471067>.
- [625] P. Procacci, M. Marchi, G.J. Martyna, Electrostatic calculations and multiple time scales in molecular dynamics simulation of flexible molecular systems, *J. Chem. Phys.* 108 (21) (1998) 8799–8803, <https://doi.org/10.1063/1.476326>.
- [626] C.H. Bennett, Exact defect calculations in model substances, in: A.S. Nowick, J.J. Burton (Eds.), *Diffusion in Solids*, Academic Press, New York, 1975, pp. 73–113.
- [627] D. Chandler, Statistical-mechanics of isomerization dynamics in liquids and transition-state approximation, *J. Chem. Phys.* 68 (6) (1978) 2959–2970, <https://doi.org/10.1063/1.436049>.

- [628] R. Kapral, Progress in the theory of mixed quantum-classical dynamics, *Annu. Rev. Phys. Chem.* 57 (2006) 129–157, <https://doi.org/10.1146/annurev.physchem.57.032905.104702>.
- [629] M.J. Ruiz-Montero, D. Frenkel, J.J. Brey, Efficient schemes to compute diffusive barrier crossing rates, *Mol. Phys.* 90 (6) (1997) 925–941, <https://doi.org/10.1080/002689797171922>.
- [630] W.H. Miller, Importance of nonseparability in quantum mechanical transition-state theory, *Acc. Chem. Res.* 9 (8) (1976) 306–312, <https://doi.org/10.1021/ar50104a005>.
- [631] M.A. Wilson, D. Chandler, Molecular dynamics study of the cyclohexane interconversion, *Chem. Phys.* 149 (1990) 11–20, [https://doi.org/10.1016/0301-0104\(90\)80127-J](https://doi.org/10.1016/0301-0104(90)80127-J).
- [632] M.H. Müser, G. Ciccotti, Two-dimensional orientational motion as a multichannel reaction, *J. Chem. Phys.* 103 (1995) 4273–4278, <https://doi.org/10.1063/1.470666>.
- [633] M. von Smoluchowski, Über brownsche molekularbewegung unter einwirkung ausserer krafte und deren zusammenhang mit der verallgemeinerten diffusionsgleichung, *Ann. Phys.* 48 (24) (1916) 1103–1112, <https://doi.org/10.1002/andp.19163532408>.
- [634] T.S. van Erp, Dynamical rare event simulation techniques for equilibrium and nonequilibrium systems, *Adv. Chem. Phys.* 151 (2012) 27–60, <https://doi.org/10.1002/978-1-118-30951-3>.
- [635] W.N. E, E. Vanden-Eijnden, Transition-path theory and path-finding algorithms for the study of rare events, *Annu. Rev. Phys. Chem.* 61 (2010) 391–420, <https://doi.org/10.1146/annurev.physchem.040808.090412>.
- [636] G. Hummer, From transition paths to transition states and rate coefficients, *J. Chem. Phys.* 120 (2) (2004) 516–523, <https://doi.org/10.1063/1.1630572>.
- [637] E. Vanden-Eijnden, Transition path theory, in: Mauro Ferrario, Giovanni Ciccotti, Kurt Binder (Eds.), *Computer Simulations in Condensed Matter Systems: From Materials to Chemical Biology*, vol. 1, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 453–493.
- [638] E. Vanden-Eijnden, M. Venturoli, Exact rate calculations by trajectory parallelization and tilting, *J. Chem. Phys.* 131 (4) (2009), <https://doi.org/10.1063/1.3180821>.
- [639] P.G. Bolhuis, C. Dellago, D. Chandler, Sampling ensembles of deterministic transition pathways, *Faraday Discuss.* 110 (1998) 421–436, <https://doi.org/10.1039/a801266k>.
- [640] D. Chandler, Finding transition pathways: throwing ropes of rough mountain passes, in the dark, in: B.J. Berne, G. Ciccotti, D.F. Coker (Eds.), *Classical and Quantum Dynamics in Condensed Phase Simulations*, International School Enrico Fermi, World Scientific, Singapore, 1998, pp. 51–66.
- [641] C. Dellago, P.G. Bolhuis, F.S. Csajka, D. Chandler, Transition path sampling and the calculation of rate constants, *J. Chem. Phys.* 108 (5) (1998) 1964–1977, <https://doi.org/10.1063/1.475562>.
- [642] C. Dellago, P.G. Bolhuis, D. Chandler, On the calculation of reaction rate constants in the transition path ensemble, *J. Chem. Phys.* 110 (1999) 6617–6625, <https://doi.org/10.1063/1.478569>.
- [643] P.G. Bolhuis, D. Chandler, C. Dellago, P.L. Geissler, Transition path sampling: throwing ropes over rough mountain passes, in the dark, *Annu. Rev. Phys. Chem.* 53 (2002) 291–318, <https://doi.org/10.1146/annurev.physchem.53.082301.113146>.
- [644] L.R. Pratt, A statistical method for identifying transition states in high dimensional problems, *J. Chem. Phys.* 9 (9) (1986) 5045–5048, <https://doi.org/10.1063/1.451695>.
- [645] C. Dellago, P.G. Bolhuis, P.L. Geissler, Transition path sampling, *Adv. Chem. Phys.* 123 (2002) 1–78, <https://doi.org/10.1002/0471231509.ch1>.
- [646] P.G. Bolhuis, D.W.H. Swenson, Transition path sampling as Markov chain Monte Carlo of trajectories: recent algorithms, software, applications, and future outlook, *Adv. Theory Simul.* 4 (4) (2021), <https://doi.org/10.1002/adts.202000237>.
- [647] Richard P. Feynman, Albert R. Hibbs, Daniel F. Styer, *Quantum Mechanics and Path Integrals*, Dover Publications, Mineola, N.Y., 2010.
- [648] T.S. van Erp, D. Moroni, P.G. Bolhuis, A novel path sampling method for the calculation of rate constants, *J. Chem. Phys.* 118 (17) (2003) 7762–7774, <https://doi.org/10.1063/1.1562614>.

688 Bibliography

- [649] A.K. Faradjian, R. Elber, Computing time scales from reaction coordinates by milestoning, *J. Chem. Phys.* 120 (23) (2004) 10880–10889, <https://doi.org/10.1063/1.1738640>.
- [650] A. Haji-Akbari, Forward-flux sampling with jumpy order parameters, *J. Chem. Phys.* 149 (7) (2018), <https://doi.org/10.1063/1.5018303>.
- [651] R.J. Allen, P.B. Warren, P.R. ten Wolde, Sampling rare switching events in biochemical networks, *Phys. Rev. Lett.* 94 (1) (2005), <https://doi.org/10.1103/PhysRevLett.94.018104>.
- [652] R.J. Allen, C. Valeriani, P.R. ten Wolde, Forward flux sampling for rare event simulations, *J. Phys. Condens. Matter* 21 (46) (2009), <https://doi.org/10.1088/0953-8984/21/46/463102>.
- [653] R.J. Allen, D. Frenkel, P.R. ten Wolde, Simulating rare events in equilibrium or nonequilibrium stochastic systems, *J. Chem. Phys.* 124 (2) (2006) 024102, <https://doi.org/10.1063/1.2140273>.
- [654] F.A. Escobedo, E.E. Borrero, J.C. Araque, Transition path sampling and forward flux sampling. Applications to biological systems, *J. Phys. Condens. Matter* 21 (33) (2009), <https://doi.org/10.1088/0953-8984/21/33/333101>.
- [655] D. Moroni, P.G. Bolhuis, T.S. van Erp, Rate constants for diffusive processes by partial path sampling, *J. Chem. Phys.* 120 (9) (2004) 4055–4065, <https://doi.org/10.1063/1.1644537>.
- [656] S. Hussain, A. Haji-Akbari, Studying rare events using forward-flux sampling: recent breakthroughs and future outlook, *J. Chem. Phys.* 152 (6) (2020), <https://doi.org/10.1063/1.5127780>.
- [657] I. Sanchez-Burgos, A. Garaizar, C. Vega, E. Sanz, J.R. Espinosa, Parasitic crystallization of colloidal electrolytes: growing a metastable crystal from the nucleus of a stable phase, *Soft Matter* 17 (3) (2021), <https://doi.org/10.1039/d0sm01680b>.
- [658] P.L. Geissler, C. Dellago, D. Chandler, Kinetic pathways of ion pair dissociation in water, *J. Phys. Chem. B* 103 (18) (1999) 3706–3710, <https://doi.org/10.1021/jp984837g>.
- [659] P. Hanggi, P. Talkner, M. Borkovec, Reaction-rate theory - 50 years after Kramers, *Rev. Mod. Phys.* 62 (2) (1990) 251–341, <https://doi.org/10.1103/revmodphys.62.251>.
- [660] J. Wedekind, D. Reguera, Kinetic reconstruction of the free-energy landscape, *J. Phys. Chem. B* 112 (35) (2008) 11060–11063, <https://doi.org/10.1021/jp804014h>.
- [661] L.S. Bartell, D.T. Wu, A new procedure for analyzing the nucleation kinetics of freezing in computer simulation, *J. Chem. Phys.* 125 (19) (2006), <https://doi.org/10.1063/1.2363382>.
- [662] J. Wedekind, R. Strey, D. Reguera, New method to analyze simulations of activated processes, *J. Chem. Phys.* 126 (13) (2007), <https://doi.org/10.1063/1.2713401>.
- [663] H.A. Kramers, Brownian motion in a field of force and the diffusion model of chemical reactions, *Physica* 7 (1940) 284–304, [https://doi.org/10.1016/S0031-8914\(40\)90098-2](https://doi.org/10.1016/S0031-8914(40)90098-2).
- [664] T.J.H. Vlugt, B. Smit, On the efficient sampling of pathways in the transition path ensemble, *Comput. Phys. Commun.* 2 (2) (2001) 3–7, <https://doi.org/10.1039/b009865p>.
- [665] G. Mills, H. Jonsson, G.K. Schenter, Reversible work transition-state theory - application to dissociative adsorption of hydrogen, *Surf. Sci.* 324 (2–3) (1995) 305–337, [https://doi.org/10.1016/0039-6028\(94\)00731-4](https://doi.org/10.1016/0039-6028(94)00731-4).
- [666] G. Henkelman, H. Jonsson, A dimer method for finding saddle points on high dimensional potential surfaces using only first derivatives, *J. Chem. Phys.* 111 (15) (1999) 7010–7022, <https://doi.org/10.1063/1.480097>.
- [667] G. Henkelman, B.P. Uberuaga, H. Jonsson, A climbing image nudged elastic band method for finding saddle points and minimum energy paths, *J. Chem. Phys.* 113 (22) (2000) 9901–9904, <https://doi.org/10.1063/1.1329672>.
- [668] G. Henkelman, H. Jonsson, Improved tangent estimate in the nudged elastic band method for finding minimum energy paths and saddle points, *J. Chem. Phys.* 113 (22) (2000) 9978–9985, <https://doi.org/10.1063/1.1323224>.
- [669] G.T. Barkema, N. Mousseau, Event-based relaxation of continuous disordered systems, *Phys. Rev. Lett.* 77 (21) (1996) 4358–4361, <https://doi.org/10.1103/PhysRevLett.77.4358>.
- [670] A.F. Voter, Hyperdynamics: accelerated molecular dynamics of infrequent events, *Phys. Rev. Lett.* 78 (20) (1997) 3908–3911, <https://doi.org/10.1103/PhysRevLett.78.3908>.

- [671] M.R. Sorensen, A.F. Voter, Temperature-accelerated dynamics for simulation of infrequent events, *J. Chem. Phys.* 112 (21) (2000) 9599–9606, <https://doi.org/10.1063/1.481576>.
- [672] A.F. Voter, Parallel replica method for dynamics of infrequent events, *Phys. Rev. B* 57 (22) (1998) 13985–13988, <https://doi.org/10.1103/PhysRevB.57.R13985>.
- [673] H. Sidky, W. Chen, A.L. Ferguson, Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation, *Mol. Phys.* 118 (5) (2020), <https://doi.org/10.1080/00268976.2020.1737742>.
- [674] D.D. Wang, P. Tiwary, State predictive information bottleneck, *J. Chem. Phys.* 154 (13) (2021), <https://doi.org/10.1063/5.0038198>.
- [675] A. Malevanets, R. Kapral, Mesoscopic model for solvent dynamics, *J. Chem. Phys.* 110 (17) (1999) 8605–8613, <https://doi.org/10.1063/1.478857>.
- [676] A. Malevanets, R. Kapral, Solute molecular dynamics in a mesoscale solvent, *J. Chem. Phys.* 112 (16) (2000) 7260–7269, <https://doi.org/10.1063/1.481289>.
- [677] G. Gompper, T. Ihle, D.M. Kroll, R.G. Winkler, Multi-particle collision dynamics: a particle-based mesoscale simulation approach to the hydrodynamics of complex fluids, *Adv. Polym. Sci.* 221 (2009) 1–87, https://doi.org/10.1007/12_2008_5.
- [678] S. Succi, *The Lattice Boltzmann Equation: for Complex States of Flowing Matter*, Oxford University Press, Oxford UK, 2018.
- [679] P.J. Hoogerbrugge, J.M.V.A. Koelman, Simulating microscopic hydrodynamics phenomena with dissipative particle dynamics, *Europhys. Lett.* 19 (1992) 155–160, <https://doi.org/10.1209/0295-5075/19/3/001>.
- [680] J.M.V.A. Koelman, P.J. Hoogerbrugge, Dynamic simulation of hard-sphere suspensions under steady shear, *Europhys. Lett.* 21 (3) (1993) 363–368, <https://doi.org/10.1209/0295-5075/21/3/018>.
- [681] I. Pagonabarraga, D. Frenkel, Dissipative particle dynamics for interacting systems, *J. Chem. Phys.* 115 (11) (2001) 5015–5026, <https://doi.org/10.1063/1.1396848>.
- [682] R.D. Groot, P.B. Warren, Dissipative particle dynamics: bridging the gap between atomistic and mesoscopic simulation, *J. Chem. Phys.* 107 (11) (1997) 4423–4435, <https://doi.org/10.1063/1.474784>.
- [683] P. Espanol, P.B. Warren, Statistical mechanics of dissipative particle dynamics, *Europhys. Lett.* 30 (4) (1995) 191–196, <https://doi.org/10.1209/0295-5075/30/4/001>.
- [684] P. Espanol, Hydrodynamics from dissipative particle dynamics, *Phys. Rev. E* 52 (2) (1995) 1734–1742, <https://doi.org/10.1103/physreve.52.1734>.
- [685] C.A. Marsh, G. Backx, M.H. Ernst, Fokker-Planck-Boltzmann equation for dissipative particle dynamics, *Europhys. Lett.* 38 (6) (1997) 411–415, <https://doi.org/10.1209/epl/i1997-00260-6>.
- [686] C.A. Marsh, G. Backx, M.H. Ernst, Static and dynamic properties of dissipative particle dynamics, *Phys. Rev. E* 56 (2) (1997) 1676–1691, <https://doi.org/10.1103/PhysRevE.56.1676>.
- [687] A.J. Masters, P.B. Warren, Kinetic theory for dissipative particle dynamics: the importance of collisions, *Europhys. Lett.* 48 (1) (1999) 1–7, <https://doi.org/10.1209/epl/i1999-00105-4>.
- [688] C.A. Marsh, J.M. Yeomans, Dissipative particle dynamics: the equilibrium for finite time steps, *Europhys. Lett.* 37 (8) (1997) 511–516, <https://doi.org/10.1209/epl/i1997-00183-2>.
- [689] I. Pagonabarraga, M.H.J. Hagen, D. Frenkel, Self-consistent dissipative particle dynamics algorithm, *Europhys. Lett.* 42 (4) (1998) 377–382, <https://doi.org/10.1209/epl/i1998-00258-6>.
- [690] P. Espanol, P.B. Warren, Perspective: dissipative particle dynamics, *J. Chem. Phys.* 146 (15) (2017), <https://doi.org/10.1063/1.4979514>.
- [691] S.M. Willemse, T.J.H. Vlugt, H.C.J. Hoefsloot, B. Smit, Combining dissipative particle dynamics and Monte Carlo techniques, *J. Comput. Phys.* 147 (2) (1998) 507–517, <https://doi.org/10.1006/jcph.1998.6099>.
- [692] J.B. Avalos, A.D. Mackie, Dissipative particle dynamics with energy conservation, *Europhys. Lett.* 40 (2) (1997) 141–146, <https://doi.org/10.1209/epl/i1997-00436-6>.

690 Bibliography

- [693] P. Espanol, Dissipative particle dynamics with energy conservation, *Europhys. Lett.* 40 (6) (1997) 631–636, <https://doi.org/10.1209/epl/i1997-00515-8>.
- [694] P. Espanol, M. Revenga, Smoothed dissipative particle dynamics, *Phys. Rev. E* 67 (2) (2003) 026705, <https://doi.org/10.1103/PhysRevE.67.026705>.
- [695] L.B. Lucy, Numerical approach to testing of fission hypothesis, *Astron. J.* 82 (12) (1977) 1013–1024, <https://doi.org/10.1086/112164>.
- [696] R.A. Gingold, J.J. Monaghan, Smoothed particle hydrodynamics - theory and application to non-spherical stars, *Mon. Not. R. Astron. Soc.* 181 (2) (1977) 375–389, <https://doi.org/10.1093/mnras/181.3.375>.
- [697] J.J. Monaghan, Smoothed particle hydrodynamics, *Rep. Prog. Phys.* 68 (8) (2005) 1703–1759, <https://doi.org/10.1088/0034-4885/68/8/r01>.
- [698] G.A. Bird, Approach to translational equilibrium in a rigid sphere gas, *Phys. Fluids* 6 (10) (1963) 1518–1519, <https://doi.org/10.1063/1.1710976>.
- [699] G.A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Engineering Science Series, Clarendon Press, Oxford, UK, 1995.
- [700] G.A. Bird, Recent advances and current challenges for dsmc, *Comput. Math. Appl.* 35 (1–2) (1998) 1–14, [https://doi.org/10.1016/S0898-1221\(97\)00254-X](https://doi.org/10.1016/S0898-1221(97)00254-X).
- [701] F.J. Alexander, A.L. Garcia, The direct simulation Monte Carlo method, *Comput. Phys.* 11 (6) (1997) 588–593, <https://doi.org/10.1063/1.168619>.
- [702] T. Ihle, D.M. Kroll, Stochastic rotation dynamics. I. Formalism, Galilean invariance, and Green-Kubo relations, *Phys. Rev. E* 67 (6) (2003), <https://doi.org/10.1103/PhysRevE.67.066705>.
- [703] A.W. Zantop, H. Stark, Multi-particle collision dynamics with a non-ideal equation of state. I, *J. Chem. Phys.* 154 (2) (2021), <https://doi.org/10.1063/5.0037934>.
- [704] A. Lamura, G. Gompper, Numerical study of the flow around a cylinder using multi-particle collision dynamics, *Eur. Phys. J. E* 9 (5) (2002) 477–485, <https://doi.org/10.1140/epje/i2002-10107-0>.
- [705] G.R. McNamara, G. Zanetti, Use of the Boltzmann-equation to simulate lattice-gas automata, *Phys. Rev. Lett.* 61 (20) (1988) 2332–2335, <https://doi.org/10.1103/PhysRevLett.61.2332>.
- [706] R. Benzi, S. Succi, M. Vergassola, The lattice Boltzmann equation: theory and applications, *Phys. Rep.* 222 (3) (1992) 145–197, [https://doi.org/10.1016/0370-1573\(92\)90090-m](https://doi.org/10.1016/0370-1573(92)90090-m).
- [707] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, *Annu. Rev. Fluid Mech.* 30 (1998) 329–364, <https://doi.org/10.1146/annurev.fluid.30.1.329>.
- [708] U. Frisch, B. Hasslacher, Y. Pomeau, Lattice-gas automata for the Navier-Stokes equation, *Phys. Rev. Lett.* 56 (14) (1986) 1505–1508, <https://doi.org/10.1103/PhysRevLett.56.1505>.
- [709] R. Adhikari, K. Stratford, M.E. Cates, A.J. Wagner, Fluctuating lattice Boltzmann, *Europhys. Lett.* 71 (3) (2005) 473–479, <https://doi.org/10.1209/epl/i2004-10542-5>.
- [710] B. Dunweg, U.D. Schiller, A.J.C. Ladd, Statistical mechanics of the fluctuating lattice Boltzmann equation, *Phys. Rev. E* 76 (3) (2007), <https://doi.org/10.1103/PhysRevE.76.036704>.
- [711] A.J.C. Ladd, Lattice-Boltzmann methods for suspensions of solid particles, *Mol. Phys.* 113 (17–18) (2015) 2531–2537, <https://doi.org/10.1080/00268976.2015.1023755>.
- [712] Richard P. Feynman, Robert Benjamin Leighton, Matthew Sands, *The Feynman Lectures on Physics*, Addison-Wesley, Reading, Mass., 1966.
- [713] J.D. Meiss, Symplectic maps, variational-principles, and transport, *Rev. Mod. Phys.* 64 (3) (1992) 795–848, <https://doi.org/10.1103/RevModPhys.64.795>.
- [714] S.K. Gray, D.W. Noid, B.G. Sumpter, Symplectic integrators for large-scale molecular-dynamics simulations - a comparison of several explicit methods, *J. Chem. Phys.* 101 (5) (1994) 4062–4072, <https://doi.org/10.1063/1.467523>.
- [715] M.E. Tuckerman, C.J. Mundy, G.J. Martyna, On the classical statistical mechanics of non-Hamiltonian systems, *Europhys. Lett.* 45 (2000) 149–155, <https://doi.org/10.1209/epl/i1999-00139-0>.
- [716] M.E. Tuckerman, G.J. Martyna, Understanding modern molecular dynamics: techniques and applications, *J. Phys. Chem. B* 104 (2) (2000) 159–178, <https://doi.org/10.1021/jp992433y>.

- [717] J.G. Kirkwood, F.P. Buff, The statistical mechanical theory of solutions 1, *J. Chem. Phys.* 19 (6) (1951) 774–777, <https://doi.org/10.1063/1.1748352>.
- [718] B.Q. Cheng, Computing chemical potentials of solutions from structure factors, *J. Chem. Phys.* 157 (12) (2022), <https://doi.org/10.1063/5.0107059>.
- [719] D. Bedeaux, S. Kjelstrup, S.K. Schnell, *Nano-Thermodynamics: Theory and Application*, World Scientific, Singapore, 2023.
- [720] D. Frenkel, A.J.C. Ladd, Elastic-constants of hard-sphere crystals, *Phys. Rev. Lett.* 59 (10) (1987) 1169, <https://doi.org/10.1103/PhysRevLett.59.1169>.
- [721] D.R. Squire, A.C. Holt, W.G. Hoover, Isothermal elastic constants for argon. theory and Monte Carlo calculations, *Physica* 42 (3) (1969) 388, [https://doi.org/10.1016/0031-8914\(69\)90031-7](https://doi.org/10.1016/0031-8914(69)90031-7).
- [722] M. Sprik, R.W. Impey, M.L. Klein, 2nd-order elastic-constants for the Lennard-Jones solid, *Phys. Rev. B* 29 (8) (1984) 4368–4374, <https://doi.org/10.1103/PhysRevB.29.4368>.
- [723] O. Farago, Y. Kantor, Fluctuation formalism for elastic constants in hard-spheres-and-tethers systems, *Phys. Rev. E* 61 (3) (2000) 2478–2489, <https://doi.org/10.1103/PhysRevE.61.2478>.
- [724] S. Sengupta, P. Nielaba, K. Binder, Elastic moduli, dislocation core energy, and melting of hard disks in two dimensions, *Phys. Rev. E* 61 (6) (2000) 6294–6301, <https://doi.org/10.1103/PhysRevE.61.6294>.
- [725] H. Bekker, E.J. Dijkstra, M.K.R. Renardus, H.J.C. Berendsen, An efficient, box shape independent non-bonded force and virial algorithm for molecular dynamics, *Mol. Simul.* 14 (3) (1995) 137–151, <https://doi.org/10.1080/0897029508022012>.
- [726] D.J. Auerbach, W. Paul, A.F. Bakker, C. Lutz, W.E. Rudge, F.F. Abraham, A special purpose parallel computer for molecular-dynamics - motivation, design, implementation, and application, *J. Phys. Chem.* 91 (19) (1987) 4881–4890, <https://doi.org/10.1021/j100303a004>.
- [727] J.G. Kirkwood, F.P. Buff, The statistical mechanical theory of surface tension, *J. Chem. Phys.* 17 (3) (1949) 338–343, <https://doi.org/10.1063/1.1747248>.
- [728] J.H. Irving, J.G. Kirkwood, The statistical mechanical theory of transport processes. IV. The equations of hydrodynamics, *J. Chem. Phys.* 18 (6) (1950) 817, <https://doi.org/10.1063/1.1747782>.
- [729] J.P.R.B. Walton, D.J. Tildesley, J.S. Rowlinson, The pressure tensor at the planar surface of a liquid, *Mol. Phys.* 48 (6) (1983) 1357–1368, <https://doi.org/10.1080/00268978300100971>.
- [730] M.J.P. Nijmeijer, A.F. Bakker, C. Bruin, J.H. Sikkenk, A molecular dynamics simulation of the Lennard-Jones liquid-vapour interface, *J. Chem. Phys.* 89 (6) (1988) 3789–3792, <https://doi.org/10.1063/1.454902>.
- [731] David Ruelle, *Statistical Mechanics: Rigorous Results*, The Mathematical Physics Monograph Series, W.A. Benjamin, New York, 1969.
- [732] Robert B. Dingle, *Asymptotic Expansions: Their Derivation and Interpretation*, Academic Press, London; New York, 1973.
- [733] B. Smit, Computer Simulation of Phase Coexistence: from Atoms to Surfactants, Ph.D. thesis, Utrecht University, 1990.
- [734] K.K. Mon, K. Binder, Finite size effects for the simulation of phase coexistence in the Gibbs ensemble near the critical point, *J. Chem. Phys.* 96 (9) (1992) 6989–6995, <https://doi.org/10.1063/1.462557>.
- [735] J.R. Recht, A.Z. Panagiotopoulos, Finite-size effects and approach to criticality in Gibbs ensemble simulations, *Mol. Phys.* 80 (4) (1993) 843–852, <https://doi.org/10.1080/00268979300102701>.
- [736] B. Widom, Some topics in the theory of fluids, *J. Chem. Phys.* 39 (11) (1963) 2802–2812, <https://doi.org/10.1063/1.1734110>.
- [737] R. Faller, Q.L. Yan, J.J. de Pablo, Multicanonical parallel tempering, *J. Chem. Phys.* 116 (13) (2002) 5419–5423, <https://doi.org/10.1063/1.1456504>.
- [738] W.G. Hoover, Constant pressure equations of motion, *Phys. Rev. A* 34 (3) (1986) 2499–2500, <https://doi.org/10.1103/physreva.34.2499>.

692 Bibliography

- [739] K. Cho, J.D. Joannopoulos, L. Kleinman, Constant-temperature molecular dynamics with momentum conservation, *Phys. Rev. E* 47 (5) (1993) 3145–3151, <https://doi.org/10.1103/physreve.47.3145>.
- [740] D.M. Heyes, M. Barber, J.H.R. Clarke, Molecular dynamics computer simulation of surface properties of crystalline potassium chloride, *J. Chem. Soc. Faraday Trans. II* 73 (1977) 1485–1496, <https://doi.org/10.1039/f29777301485>.
- [741] A. Grzybowski, E. Gwozdz, A. Brodka, Ewald summation of electrostatic interactions in molecular dynamics of a three-dimensional system with periodicity in two directions, *Phys. Rev. B* 61 (10) (2000) 6706–6712, <https://doi.org/10.1103/PhysRevB.61.6706>.
- [742] A.H. Widmann, D.B. Adolf, A comparison of Ewald summation techniques for planar surfaces, *Comput. Phys. Commun.* 107 (1–3) (1997) 167–186, [https://doi.org/10.1016/S0010-4655\(97\)00090-8](https://doi.org/10.1016/S0010-4655(97)00090-8).
- [743] S.W. de Leeuw, J.W. Perram, Electrostatic lattice sums for semi-infinite lattices, *Mol. Phys.* 37 (1979) 1313–1322, <https://doi.org/10.1080/00268977900100951>.
- [744] E.R. Smith, Electrostatic potentials for simulations of thin layers, *Mol. Phys.* 65 (5) (1988) 1089–1104, <https://doi.org/10.1080/00268978800101621>.
- [745] E. Spohr, Effect of boundary conditions and system size on the interfacial properties of water and aqueous solutions, *J. Chem. Phys.* 107 (1994) 6342–6348, <https://doi.org/10.1063/1.474295>.
- [746] J. Hautman, M.L. Klein, An Ewald summation method for planar surfaces and interfaces, *Mol. Phys.* 75 (2) (1992) 379–395, <https://doi.org/10.1080/00268979200100301>.
- [747] I.C. Yeh, M.L. Berkowitz, Ewald summation for systems with slab geometry, *J. Chem. Phys.* 111 (7) (1999) 3155–3162, <https://doi.org/10.1063/1.479595>.
- [748] E.R. Smith, Electrostatic energy in ionic crystals, *Proc. R. Soc. Lond. A* 375 (1763) (1981) 475–505, <https://doi.org/10.1098/rspa.1981.0064>.
- [749] P.S. Crozier, R.L. Rowley, E. Spohr, D. Henderson, Comparison of charged sheets and corrected 3d Ewald calculations of long-range forces in slab geometry electrolyte systems with solvent molecules, *J. Chem. Phys.* 112 (21) (2000) 9253–9257, <https://doi.org/10.1063/1.481546>.
- [750] T.J.H. Vlugt, Adsorption and Diffusion in Zeolites: a Computational Study, Ph.D. thesis, University of Amsterdam, 2000.
- [751] M. Dijkstra, Confined thin films of linear and branched alkanes, *J. Chem. Phys.* 107 (8) (1997) 3277–3288, <https://doi.org/10.1063/1.474678>.
- [752] M.G. Martin, J.I. Siepmann, Novel configurational-bias Monte Carlo method for branched molecules. Transferable potentials for phase equilibria. 2. United-atom description of branched alkanes, *J. Phys. Chem. B* 103 (21) (1999) 4508–4517, <https://doi.org/10.1021/jp984742e>.
- [753] P.V.K. Pant, D.N. Theodorou, Variable connectivity method for the atomistic Monte Carlo simulation of polydisperse polymer melts, *Macromolecules* 28 (21) (1995) 7224–7234, <https://doi.org/10.1021/ma00125a027>.
- [754] M.G. Wu, M.W. Deem, Analytical rebridging Monte Carlo: application to cis/trans isomerization in proline-containing, cyclic peptides methods for cyclic peptides, *J. Chem. Phys.* 111 (1999) 6625–6632, <https://doi.org/10.1063/1.479952>.
- [755] M.G. Wu, M.W. Deem, Efficient Monte Carlo methods for cyclic peptides, *Mol. Phys.* 97 (4) (1999) 559–580, <https://doi.org/10.1080/0026897990482855>.
- [756] G.C.A.M. Mooij, Novel Simulation Techniques for the Study of Polymer Phase Equilibria, Ph.D. thesis, University of Amsterdam, 1993.
- [757] B. Smit, S. Karaborni, J.I. Siepmann, Computer simulations of vapour-liquid phase equilibria of n-alkanes, *J. Chem. Phys.* 102 (5) (1995) 2126–2140, <https://doi.org/10.1063/1.469563>.
- [758] M. Laso, J.J. de Pablo, U.W. Suter, Simulation of phase equilibria for chain molecules, *J. Chem. Phys.* 97 (4) (1992) 2817–2819, <https://doi.org/10.1063/1.463022>.
- [759] J.I. Siepmann, S. Karaborni, B. Smit, Vapor-liquid equilibria of model alkanes, *J. Am. Chem. Soc.* 115 (14) (1993) 6454–6455, <https://doi.org/10.1021/ja00067a088>.

- [760] C. Tsonopoulos, Critical constants of normal alkanes from methane to polyethylene, *AIChE J.* 33 (12) (1987) 2080–2083, <https://doi.org/10.1002/aic.690331219>.
- [761] W.L. Jorgensen, J.D. Madura, C.J. Swenson, Optimized intermolecular potential function for liquid hydrocarbons, *J. Am. Chem. Soc.* 106 (22) (1984) 6638–6646, <https://doi.org/10.1021/ja00334a030>.
- [762] S. Toxvaerd, Molecular dynamics calculation of the equation of state of alkanes, *J. Chem. Phys.* 93 (6) (1990) 4290–4295, <https://doi.org/10.1063/1.458709>.
- [763] C. Tsonopoulos, Z. Tan, Critical constant of normal alkanes from methane to polyethylene II. Application of the Flory theory, *Fluid Phase Equilib.* 83 (1993) 127–138, [https://doi.org/10.1016/0378-3812\(93\)87015-S](https://doi.org/10.1016/0378-3812(93)87015-S).
- [764] M.J. Anselme, M. Gude, A.S. Teja, The critical temperatures and densities of the n-alkanes from pentane to octadecane, *Fluid Phase Equilib.* 57 (1990) 317–326, [https://doi.org/10.1016/0378-3812\(90\)85130-3](https://doi.org/10.1016/0378-3812(90)85130-3).
- [765] Y.-J. Sheng, A.Z. Panagiotopoulos, S.K. Kumar, I. Szleifer, Monte Carlo calculation of phase equilibria for a bead-spring polymeric model, *Macromolecules* 27 (2) (1994) 400–406, <https://doi.org/10.1021/ma00080a012>.
- [766] F.A. Escobedo, J.J. de Pablo, Simulation and prediction of vapour-liquid equilibria for chain molecules, *Mol. Phys.* 87 (2) (1996) 347–366, <https://doi.org/10.1080/00268979650027496>.
- [767] M. Mondello, G.S. Grest, Molecular dynamics of linear and branched alkanes, *J. Chem. Phys.* 103 (16) (1995) 7156–7165, <https://doi.org/10.1063/1.470344>.
- [768] M. Mondello, G.S. Grest, E.B. Webb, P. Peczak, Dynamics of n-alkanes: comparison to Rouse model, *J. Chem. Phys.* 109 (2) (1998) 798–805, <https://doi.org/10.1063/1.476619>.
- [769] J.D. Moore, S.T. Cui, H.D. Cochran, P.T. Cummings, Rheology of lubricant basestocks: a molecular dynamics study of c-30 isomers, *J. Chem. Phys.* 113 (19) (2000) 8833–8840, <https://doi.org/10.1063/1.1318768>.
- [770] C. McCabe, S.T. Cui, P.T. Cummings, P.A. Gordon, R.B. Saeger, Examining the rheology of 9-octylheptadecane to giga-pascal pressures, *J. Chem. Phys.* 114 (4) (2001) 1887–1891, <https://doi.org/10.1063/1.1334676>.
- [771] M.G. Martin, J.I. Siepmann, Transferable potentials for phase equilibria (trappe): I. United-atom description of n-alkanes, *J. Phys. Chem. B* 102 (14) (1998) 2569–2577, <https://doi.org/10.1021/jp972543>.
- [772] S.K. Nath, F.A. Escobedo, J.J. de Pablo, On the simulation of vapor-liquid equilibria for alkanes, *J. Chem. Phys.* 108 (23) (1998) 9905–9911, <https://doi.org/10.1063/1.476429>.
- [773] C.D. Wick, M.G. Martin, J.I. Siepmann, Transferable potentials for phase equilibria. 4. United-atom description of linear and branched alkenes and alkylbenzenes, *J. Phys. Chem. B* 104 (33) (2000) 8008–8016, <https://doi.org/10.1021/jp001044x>.
- [774] S.K. Nath, B.J. Banaszak, J.J. de Pablo, A new united atom force field for α -olefins, *J. Chem. Phys.* 1114 (2001) 3612, <https://doi.org/10.1063/1.1343487>.
- [775] M.E. van Leeuwen, B. Smit, Molecular simulations of the vapour-liquid coexistence curve of methanol, *J. Phys. Chem.* 99 (7) (1995) 1831–1833, <https://doi.org/10.1021/j100007a006>.
- [776] B. Chen, J.J. Potoff, J.I. Siepmann, Monte Carlo calculations for alcohols and their mixtures with alkanes. transferable potentials for phase equilibria. 5. United-atom description of primary, secondary, and tertiary alcohols, *J. Phys. Chem. B* 105 (15) (2001) 3093–3104, <https://doi.org/10.1021/jp003882x>.
- [777] D. Wu, D. Chandler, B. Smit, Electrostatic analogy for surfactant assemblies, *J. Phys. Chem.* 96 (10) (1992) 4077–4083, <https://doi.org/10.1021/j100189a030>.
- [778] B. Smit, K. Esselink, P.A.J. Hilbers, N.M. van Os, I. Szleifer, Computer simulations of surfactant self-assembly, *Langmuir* 9 (1) (1993) 9–11, <https://doi.org/10.1021/la00025a003>.
- [779] F.H. Stillinger, Variational model for micelle structure, *J. Chem. Phys.* 78 (7) (1983) 4654–4661, <https://doi.org/10.1063/1.445310>.
- [780] G. Orkoulas, A.Z. Panagiotopoulos, Chemical-potentials in ionic systems from Monte-Carlo simulations with distance-biased test particle insertions, *Fluid Phase Equilib.* 83 (1993) 223–231, [https://doi.org/10.1016/0378-3812\(93\)87025-V](https://doi.org/10.1016/0378-3812(93)87025-V).

694 Bibliography

- [781] G. Orkoulas, A.Z. Panagiotopoulos, Free-energy and phase-equilibria for the restricted primitive model of ionic fluids from Monte-Carlo simulations, *J. Chem. Phys.* 101 (2) (1994) 1452–1459, <https://doi.org/10.1063/1.467770>.
- [782] G.L. Deitrick, L.E. Scriven, H.T. Davis, Efficient molecular simulation of chemical potentials, *J. Chem. Phys.* 90 (4) (1989) 2370, <https://doi.org/10.1063/1.455979>.
- [783] M.R. Stapleton, A.Z. Panagiotopoulos, Application of the excluded volume map sampling to phase equilibrium calculations in the Gibbs ensemble, *J. Chem. Phys.* 92 (2) (1990) 1285–1293, <https://doi.org/10.1063/1.458138>.
- [784] I.M.J.J. van de Ven-Lucassen, T.J.H. Vlugt, A.J.J. van der Zanden, P.J.A.M. Kerkhof, Using molecular dynamics to obtain Maxwell-Stefan diffusion coefficients in liquid systems, *Mol. Phys.* 94 (1998) 495–503, <https://doi.org/10.1080/002689798168006>.
- [785] G. Ilario, I.G. Tironi, W.F. van Gunsteren, A molecular dynamics study of chloroform, *Mol. Phys.* 83 (2) (1994) 381–403, <https://doi.org/10.1080/00268979400101331>.
- [786] P. Hellekalek, Good random number generators are (not so) easy to find, *Math. Comput. Simul.* 46 (1998) 485–505, [https://doi.org/10.1016/S0378-4754\(98\)00078-0](https://doi.org/10.1016/S0378-4754(98)00078-0).

Acronyms

ACF	Auto-Correlation Function. 154 , 164
BCC	Body Centered Cubic. 99 , 272
BD	Brownian Dynamics. 244
CA	Cellular-Automaton. 569 , 570
CBMC	Configurational-Bias Monte Carlo. 368 , 413 , 436 , 438 , 447 , 448 , 485 , e46–e48 , e53 , e55–e57 , e102 , e104 , e105
DFT	Density Functional Theory. 506
DPD	Dissipative Particle Dynamics. 561–563 , 566 , 621 , e65 , e66 , e114
DSMC	Direct Simulation Monte Carlo Method. 567 , 568
ECMC	Event Chain Monte Carlo. 402 , 487–489
FCC	Face Centered Cubic. 99 , 272 , 338 , 340 , 344 , 364 , 569 , e98–e100
FFS	Forward-Flux Sampling. 547–549 , 552 , 553
FFT	Fast Fourier Transform. 389 , 390 , 404
FMM	Fast-Multipole Method. 373 , 394 , 397 , 403
GCMC	Grand canonical Monte Carlo. 205 , 208 , 214
GE	Gibbs Ensemble. 216 , 218 , 226
GK	Green-Kubo. 43 , 154 , 155 , 160 , 163 , 164 , 177 , e82
GR	Greengard and Rokhlin. 394 , 396 , 397
HCP	Hexagonal Close Packed. 272 , 340 , 344 , e100
KMC	Kinetic Monte Carlo. 469–471
LB	Lattice-Boltzmann. 569 , 570
LE	Langevin Equation. 242
LJ	Lennard-Jones. 68 , 70 , 73 , 75 , 77 , 103
MBAR	Multistate Bennett Acceptance Ratio. 293 , 312 , 313 , 315 , 316 , 521

MC	Monte Carlo. 24, 38, 54, 102, 118, 125, 148, 167, 168, 404, 405, 455, 626
MCMC	Markov-Chain Monte Carlo. 54, 78, 475
MD	Molecular Dynamics. 11, 36–38, 53, 62, 97, 125, 128, 147, 148, 155, 167, 168, 174, 391, 398, 403, 405, 455, 626
MEM	Maxwell Equations Method. 399, 401, 403, 404
MFPT	Mean First-Passage Times. 552, 553, 555
ML	Machine Learning. 5, 475, 476, 558
MPC	Multi-particle Collision Dynamics. 560, 567, 569
MTS	Multiple-Time-Step. 512, 514, e109
NEB	Nudged Elastic Band. 557
NH	Nosé-Hoover. 251, e24
PMA	Parsimonious Metropolis Algorithm. 62
PME	Particle Mesh Ewald. 389, 392, 398
PPPM	Particle-Particle/Particle-Mesh. 389, 390, 392, 403
PT	Parallel Tempering. 456, 457, 459, 461, 463, 556
RG	Recoil Growth. 446–448, e57, e104, e105
SDPD	Smoothed Dissipative Particle Dynamics. 566, 567, 621
SGMC	Semigrand Monte Carlo. 207, 209
SI	Supplementary Information. xvi
SPME	Smooth Particle Mesh Ewald. 389, 392
SRD	Stochastic Rotation Dynamics. 560, 568, 569
SW	Swendsen-Wang. 480, 482–484
TI	Thermodynamic Integration. 278, 328
TIS	Transition Interface Sampling. 546–548, 552
TPS	Transition-Path Sampling. 535–538, 543, 546–548, 552, 555
TPT	Transition Path Theory. 535, 548–550
TST	Transition State Theory. 521, 528
VACF	Velocity autocorrelation function. 157, 158, 176, 177, e82
VMD	Visual Molecular Dynamics. 94
VMMC	Virtual-Move Monte Carlo. 486
WF	Wang-Ramirez-Dobnikar-Frenkel. 72, 93, 103, 108, 177
WK	Wiener-Khinchin. 44
WL	Wang-Landau. 304–309
WTM	Well-Tempered Metadynamics. 310

Glossary

A	Surface area. 135
A	Observable. 29 , 55
C_V	Heat capacity, constant volume. 129
C_p	Heat capacity, constant pressure. 130
D	Diffusion coefficient. 43 , 148 , 155 , 176 , 238 , 529 , 533 , e83 , e89
D^c	Corrected, collective, or Maxwell-Stefan diffusion coefficient. 152 , 153
D^s	Self-diffusion coefficient. 151 , 153 , e89
D^t	Fick or transport diffusion coefficient. 151 , 153
E	Energy. 12 , 29 , 128
\mathbf{E}	Electric field. 383 , 507
F	Helmholtz free energy. 18 , 22 , 24 , 28 , 33 , 275
G	Gibbs free energy. 18 , 22 , 34 , 187 , 268 , 273
H	Enthalpy. 18 , 128
\mathcal{H}	Hamiltonian. 25 , 29 , 31 , 39 , 54 , 91 , 501 , 578 , 607 , e26
\mathbf{J}_h	Enthalpy flux. 597
\mathcal{K}	Kinetic energy. 29 , 90 , 130 , 500 , 573
L	Box length. 130 , 131 , 374
L_{ij}	Onsager's transport coefficients. 600
\mathcal{L}	Lagrangian. 90 , 494 , 575 , 576
M	Jacobian Matrix. 582 , 584 , 587
N	Number of particles. 11 , 15 , 29
\mathcal{N}	Probability density. 33 , 56 , 189
$\mathcal{N}(o)$	Probability to find the system in configuration o . 59
\mathcal{O}	Order symbol. 29 , 55
P	Probability. 27
P	Pressure. 16 , 23 , 130
\mathbf{P}	Polarization. 384
Q	Partition function. 28 , 29 , 32 , 34 , 54
\mathbf{Q}	Quaternion of unit norm. 88
R	The gas constant. 46
\mathcal{R}	Random number. 8 , 61
S	Action. 573–575

S	Entropy. 14–16, 19, 26, 122, 595, 598
\dot{S}	Entropy production. 596, 600
$S(q)$	Structure factor. 139, 591
T	Temperature. 14, 15
U	Potential energy. 29, 90, 500, 573
V	Volume. 15, 29, 374
W	Rosenbluth factor. 360, 408, 413, 415, e47, e51, e102
W	Virial. 76, 132, e76
Z	Configurational integral. 33
$\text{acc}(o \rightarrow n)$	The probability of accepting a trial move from o to n . 60
d	Dimensionality. 31, 47, 55, 67, 131, 149, 275
f	Fugacity (note we define f as the <i>density</i> of the ideal gas). 46, 152, 198, 231, 283, 462
g	Gravitational acceleration. 49, 579
$g(r)$	Radial distribution function. 140, 143, 144, e81
g_k	The Kirkwood factor. 385
h	Planck's constant. 30, 126, 583
h	Partial molar enthalpy. 128
iL	Liouville operator. 118, 509, e33
k_B	Boltzmann's constant. 27
m	Mass. 74
p	Momentum. 30
p_n	Normal component of the pressure tensor. e5
p_t	Tangential component of the pressure tensor. e5
q	Charge. 374
q	Heat. 12, 15
r_c	Cut-off radius. 66, 67
s	Molar entropy. 24
s	Scaled coordinates. 130, 187
v	Molar volume. 24
w	Work. 12, 13, 15, 16
Λ	Thermal Broglie wavelength. 32, 126, 199, 266, 336
Ξ	Grand-canonical partition function. 36, 204, 206, 591
Φ	Grand potential. 19, 21, 22, 36, 591
Ω	Partition function. 583
Ω	Number of eigenstates. 25, 33
α	Transition matrix of the Markov chain. 60, 61, 81, 82
α	Isothermal compressibility. 507
$\alpha(o \rightarrow n)$	The transition matrix that determines the probability of performing a trial move from o to n . 60
β	Inverse temperature. 27

β_T	Polarizability. 135
γ	Surface tension. 47, 135, 138, 269, e4, e5
ϵ	Unit of energy. 74
ϵ	Dielectric constant. 383
ϵ_0	Dielectric permittivity of vacuum. 372
ζ	Bulk viscosity. 622
η	Efficiency. 13
η	Shear viscosity. 153, 610, 612, 622
κ	Thermal conductivity. 622
κ	Transmission coefficient. 528, 530, e110
λ	Coupling parameter. 39, 40, 42
λ	Lyapunov exponent. 107
λ_B	Bjerrum length. 372
λ_T	Thermal conductivity. 154
μ	Dipole moment (value). 507
μ	Chemical potential. 16, 24, 198, e97
μ	Dipole moment (vector). 39, 382, 413, 609
ξ	Fugacity fraction. 208
$\pi(o \rightarrow n)$	Transition probability to go from state o to state n . 59, 61
σ	Unit of length. 74
σ	Entropy production per unit volume. 596
σ_e	Electrical conductivity. 154, 610
σ_{xy}	Shear stress. 612
ϕ	Fugacity coefficient. 46, e69

Index

Note

Entries in the SI, start with **e**, see, [xvi](#)

A

- Absolute Temperature, [14](#)
- Acceptance rule
 - biased sampling, [407](#)
 - canonical ensemble, [60](#), [63](#), [183](#)
 - CBMC fixed endpoints, [434](#)
 - configurational-bias Monte Carlo, [414](#),
[416](#), [421](#)
 - Gibbs ensemble, [219](#)
 - Gibbs-ensemble technique, [e52](#)
 - grand-canonical ensemble, [200](#), [442](#),
[443](#)
 - isobaric-isothermal ensemble, [187](#)
 - Metropolis scheme, [60](#)
 - noise, [467](#)
 - NPT ensemble, [187](#)
 - orientational bias, [408](#)
 - parallel tempering, [458](#)
 - path ensemble, [544](#)
- Acceptance-rejection technique, [424](#)
- Accepting a trial move, [61](#)
- Action
 - discretized, [123](#)
 - Onsager-Machlup, [123](#)
- Activation-relaxation technique, [557](#)
- Adsorption
 - Example, [202](#)
 - methane in zeolite, [203](#)
- Algorithm, [xxii](#), [7](#)
- cell lists, [630](#), [631](#)
- cell lists and Verlet lists, [632](#), [633](#)
- combined lists, [632](#), [633](#)
- configurational-bias Monte Carlo,
[426](#)–[428](#)
- configurational-bias Monte Carlo
(lattice), [417](#), [418](#)
- diffusion, [156](#), [160](#)
- equations of motion: Andersen
thermostat, [241](#)
- equations of motion: Nosé-Hoover
thermostat, [e38](#)
- equations of motion: Velocity-Verlet,
[115](#)
- equations of motion: Verlet algorithm,
[105](#)
- exchange of particle, [201](#)
- force, calculation of the, [103](#)
- Gaussian distribution, [637](#)
- generate an Einstein crystal, [335](#)
- generate bond and torsion angle, [641](#)
- generate bond angle, [640](#)
- generate bond length with harmonic
springs, [639](#)
- Gibbs-ensemble technique, [222](#), [e19](#)
- growing a chain on a lattice, [418](#)
- growing an alkane, [426](#)
- growing ethane, [427](#)
- growing propane, [428](#)
- initialization, [101](#)
- linked lists, [630](#), [631](#)
- mean-squared displacement, [156](#), [160](#)
- Molecular Dynamics: Andersen
thermostat, [240](#)
- Molecular Dynamics: Nosé-Hoover
thermostat, [e39](#)
- Molecular Dynamics: NVE ensemble,
[100](#)

- Molecular Dynamics: Velocity-Verlet, 115
 Monte Carlo technique (NVT), 334
 Monte Carlo technique: μ VT ensemble, 200, 201
 Monte Carlo technique: (fixed center of mass), 334
 Monte Carlo technique: NPT ensemble, 189, 190
 Monte Carlo technique: NVT ensemble, 63, 64
 multiple time step, 513
 orientational bias, 409
 particle displacement, 64
 particle displacement (fixed center of mass), 334
 particle exchange (Gibbs), e21
 particle insertion method, 282
 radial distribution function, 147
 random vector on a unit sphere, 638
 selection of trial orientations, 638
 trial position of n-alkane, 428
 velocity autocorrelation function, 156, 160
 Verlet, 122
 Verlet lists, 627–629
 volume change (Gibbs), e19
 volume change (NPT), 190
 Widom method, 282
- Alkanes
 critical properties, e52
 Example, 361, 443
 generation of trial positions, 424
- Andersen thermostat
 Algorithm, 240, 241
 Case Study, e89
 Example, 238
 Exercise, 259
 harmonic oscillator, 254, e92
 Lennard-Jones, 238, e89
- Antisymmetric matrix, 581
- Auxiliary function, 18
- B**
- Barrier crossing
 Case Study, e109
 Example, 524
 Basic assumption, 25
- Bennet-Chandler approach, 520
 Block averages
 statistical error, 171
 Blue-Moon ensemble, 522
 Blue-moon ensemble
 Automatic Differentiation, 524
 Bond formation scheme, 484
 acceptance rule, 484
 Bonded potential energy, 419
 Boundary conditions, 65
 Branched alkanes
 configurational-bias Monte Carlo, e45
 Broglie wavelength, 32
- C**
- Canonical ensemble, 32
 Monte Carlo technique, 182
 Monte Carlo technique, justification of, 183
- Canonical transformation
 symplectic condition, 583
- Car-Parrinello method, 506
- Carbon footprint, xvi
- Case Study, xxii
 μ VT ensemble, e87
 Andersen thermostat, e89
 barrier crossing, e109
 cell lists, e116
 chemical potential: Lennard-Jones, e94
 comparison CPU saving schemes, e116
 configurational-bias Monte Carlo, e101
 constraints, e107
 detailed balance, e77
 diffusion, e83
 dissipative particle dynamics, e114
 dynamic properties of the Lennard-Jones fluid, e82
 Einstein crystal, e96
 equation of state: Lennard-Jones, e85
 equation of state: Lennard-Jones chains, e101
 Gibbs-ensemble technique, e87
 hard spheres, e96
 harmonic oscillator, e92
 keep old configuration, e79
 Lennard-Jones, e87
 Molecular Dynamics, e79
 Monte Carlo technique, e87
 multiple time step, e107

- Nosé-Hoover thermostat, [e90](#)
- NPT ensemble, [e85](#)
- NVT ensemble, [e75](#)
- overlapping distribution, [e94](#)
- parallel tempering, [e105](#)
- particle insertion method, [e93](#)
- path ensemble, [e112](#)
- phase equilibria: Lennard-Jones, [e86](#)
- rare events, [e109](#)
- recoil growth, [e104](#)
- SHAKE, [e107](#)
- solid-liquid phase equilibrium of hard spheres, [e96](#)
- static properties of the Lennard-Jones fluid, [e79](#)
- trial configurations of ideal chains, [e101](#)
- Verlet lists, [e116](#)
- Widom method, [e93](#)
- Cell lists, [629](#)
 - Algorithm, [630, 631](#)
 - Case Study, [e116](#)
 - Example, [634](#)
- Cell-veto method, [490](#)
- Chain molecules
 - chemical potential, [352](#)
 - concerted rotation, [91](#)
 - Example, [463](#)
- Chemical potential, [16, 21, 24](#)
 - acceptance-ratio method, [293](#)
 - Case Study, [e94](#)
 - chain molecules, [352](#)
 - Example, [283, 290](#)
 - excess chemical potential, [281](#)
 - expanded-ensemble method, [464](#)
 - finite-size corrections, [286](#)
 - ideal gas, [e68](#)
 - incremental, [352](#)
 - Lennard-Jones, [283, 290, e93](#)
 - mixtures, [287](#)
 - modified Widom method, [352](#)
 - multi canonical ensemble method, [e18](#)
 - NPT ensemble, [285](#)
 - NVE ensemble, [286](#)
 - NVT ensemble, [281](#)
 - overlapping distribution, [284, 289, 363](#)
 - particle insertion method, [280, 281](#)
 - recursive sampling, [365](#)
- Rosenbluth sampling, [361](#)
- tail correction, [284, e94](#)
- thermodynamic integration, [351](#)
- Widom method, [280, 281](#)
- Classical thermodynamics, [12](#)
 - pressure, [23](#)
- Clausius-Clapeyron equation, [271](#)
- Cluster moves, [479](#)
 - Example, [e63](#)
 - lattice, [480](#)
 - Wolf algorithm, [482](#)
- Coarse-grained model, [560](#)
- Colloids, [560](#)
 - Example, [438](#)
- Committor, [550, 617](#)
- Compressibility, [135](#)
 - phase space, [587](#)
- Computer experiments, [125](#)
 - compressibility, [135](#)
 - diffusion, [148](#)
 - dynamical properties, [147](#)
 - heat capacity, [129](#)
 - internal energy, [128](#)
 - order- n algorithm to measure correlations, [155](#)
 - partial molar quantities, [128](#)
 - pressure, [130](#)
 - pressure by thermodynamic integration, [133](#)
 - pressure by virtual volume changes, [134](#)
 - radial distribution function, [140](#)
 - static properties, [127](#)
 - structure factor, [139](#)
 - surface free-energy density, [138](#)
 - surface stress, [138](#)
 - surface tension, [135](#)
 - surface tension from virtual moves, [137](#)
 - temperature, [127](#)
- Concerted rotation, [91, e48](#)
- Configurational Integral, [33](#)
- Configurational-bias Monte Carlo, [405](#)
 - acceptance rule, [414, 416, 421](#)
 - Algorithm (alkane), [426, 428](#)
 - Algorithm (ethane), [427](#)
 - Algorithm (lattice), [417, 418](#)
 - Algorithm (propane), [428](#)
 - branched alkanes, [e45](#)

- Case Study, e101
 Example, 423, 429
 Exercise, 451
 explicit-hydrogen model, 427
 fixed endpoints (continuum), 434
 fixed endpoints (lattice), 432
 Gibbs-ensemble technique, e50
 justification (lattice), 416
 justification (off-lattice), 421
 lattice, 414
 off-lattice, 417
 Recoil growth, versus, 445
 super-detailed balance, 422
 trial orientations, 424
- Constrained dynamics
 averages, 499
 Case Study, e107
 Example, 510
 probability density, 502
 SHAKE, 510, e107
- Constraints, 494
 Automatic Differentiation, 499
- Coordinate transformation
 canonical, 580
- Coulomb potential, 374
- Critical exponents, 225
- D**
- Damped truncation, 393
 Density of states, 301, 303
 Detailed balance, 182, 486
 biased configurations, 406
 canonical ensemble, 184
 Case Study, e77
 Example, 79
 grand-canonical ensemble, 201
 Metropolis scheme, 60
 orientational bias, 410
 super-detailed balance, 412, 422
- Dielectric constant, 385
 Differences
 chemical potential, 287
- Diffusion, 148, 176
 Algorithm, 156, 160
 Andersen thermostat, 239, e90
 Case Study, e83
 Example, 163, 164
 Lennard-Jones, 163, 164, 239, 254, e90
- Nosé-Hoover thermostat, 254, e91
 NVE simulations, 163, e83
- Diffusion coefficient, 150
 Diffusive barrier crossing, 527
 Dissipative particle dynamics, 561
 Case Study, e114
 Example, 564
- Dynamic Monte Carlo, 469
 Dynamic properties
 diffusion, 148
 order- n algorithm to measure correlations, 155
- E**
- Early rejection scheme, 484
 Einstein crystal
 Algorithm, 335
 Case Study, e96
 Example, 335
 free energy, 327
 free energy (constrained), 340
 partition function, 327
- Einstein relation, 150
 Elastic constants, 612
 Electric susceptibility, 39
 Electrical conductivity, 154, 610
 End-bridging Monte Carlo, e48
- Ensemble, 31
 μ VT, 34
 average, 31
 canonical, 32
 grand-canonical, 34
 isobaric-isothermal, 33
 micro-canonical, 31
 NPT, 33
 NVE, 31
 NVT, 32
 path ensemble, 537
- Ensemble average, 31
 canonical ensemble, 55
 constrained dynamics, 499
 generalized coordinates, 91
 path ensemble, 537
 Rosenbluth sampling, 356
- Enthalpy, 18
 Enthalpy fluxes, 597
 Entropy, 14
 definition, 27
- Entropy production, 595

- Equation of state
 - Case Study, e75
 - Example, 75, 191, 206, 423
 - Lennard-Jones, 75, 191, 206, e85
 - Lennard-Jones chains, 423, e101
- Equations of motion
 - accuracy, 106, 109, 110
 - Algorithm, 105, 115, e38
 - energy conservation, 110
 - Euler algorithm, 112
 - Example, 576, 579
 - Hamiltonian, 573, 579
 - Lagrangian, 573, 576
 - Leap Frog algorithm, 113
 - Liouville formulation, 118
 - Lyapunov instability, 106
 - multiple time step, 509, 514
 - Nosé-Hoover, e32
 - predictor-corrector algorithm, e3
 - speed, 110
 - velocity Verlet algorithm, 113, 514
 - velocity-corrected Verlet algorithm, 114
 - Verlet algorithm, 105
- Ergodicity, 36, 37
 - trial moves, 86
- Error estimate, e80
- Euler algorithm, 112
- Eulerian angles, 88
- Event-chain Monte Carlo, 487
- Event-Driven Monte Carlo, 402
- Ewald summation, 372, 373
 - accuracy, 386
 - boundary conditions, 385
 - Coulombic interactions, 374
 - dielectric constant, 383
 - dipolar interactions, 382
 - Example, 402, e43
 - Fourier part, 379
 - real-space sum, 382
 - self interaction, 380
 - slab geometry, e43
 - two dimensions, e41
- Example, xxiii
 - μ VT ensemble, 206
 - adsorption in zeolites, 202
 - adsorption of alkanes in zeolites, 443
 - alkanes, 361
- Andersen thermostat, 238
- barrier crossing, 524
- cell lists, 634
- chain molecules, 463
- chemical potential chain molecules, 361
- chemical potential: Lennard-Jones, 283, 290
- cluster moves, e63
- colloids and polymers, 438
- comparison CPU saving schemes, 634
- configurational-bias Monte Carlo, 423, 429
- constraints, 510
- critical properties of alkanes, e52
- detailed balance, 79
- diffusion, 163, 164
- diffusion coefficients, 151
- diffusion coefficients of confined gases, 249
- dipolar spheres, 413
- dissipative particle dynamics, 564
- dynamic properties of the Lennard-Jones fluid, 163
- Einstein crystal, 335
- equation of state: Lennard-Jones, 75, 191, 206
- equation of state: Lennard-Jones chains, 423
- Ewald summation, 402, e43
- finite-size effects (Gibbs), e14
- free energy of a solid, e71
- Gibbs ensemble (dense liquids), e71
- Gibbs ensemble (ionic fluid), 226
- Gibbs ensemble (mixtures), 227
- Gibbs ensemble (polar fluid), 226
- Gibbs-Duhem integration, 213, 272
- Gibbs-ensemble technique, 223
- grand-canonical ensemble, 443
- Greengard and Rokhlin, 402
- Hamiltonian, 579
- hard spheres, 213, 335, 344
- harmonic oscillator, 254, e25
- Henry coefficients in porous media, 361
- histogram reweighting technique, 462
- keep old configuration, 86
- Lagrangian, 576

- Lennard-Jones, 75, 79, 86, 144, 163, 164, 191, 192, 206, 223, 238, 253, 283, 290
 methane, 202
 mixture of hard disks, 92
 Molecular Dynamics, 144, 163, 164, 264
 Monte Carlo, 264
 Monte Carlo technique, 75, 79, 86, 191, 192, 206, 223, 335
 multiple time step, 510
 n-butane, 302
 normalizing flow method, 477
 Nosé-Hoover thermostat, 253
 NPT ensemble, 191, 192
 NVT ensemble, 75
 orientational bias, 413
 overlapping distribution, 290
 parallel tempering, 459, 462, 463, 555, e74
 particle insertion method, 283
 path ensemble, 541
 phase equilibria, 462
 phase equilibria: Lennard-Jones, 192, 223
 polydispersity, 213
 polymers, 463
 radial distribution function, 143
 rare events, 524, 541
 recoil growth, 449
 Rosenbluth sampling, 364
 self-consistent histogram method, 462
 semigrand ensemble, 212, 213
 SHAKE, 510
 solid, e71
 solid-liquid phase equilibrium of hard spheres, 335
 static Monte Carlo sampling, 477
 static properties of the Lennard-Jones fluid, 144
 transition path sampling, 550
 transition-path sampling, 555
 trial configurations of ideal chains, 429
 umbrella sampling, 302
 vapor-liquid equilibria, 212
 variance reduction, 143
 velocity Verlet, 115
 Verlet lists, 634
 Widom method, 283
 zeolite, 202, 361
 Exchange of particle Algorithm, 201
 Excluded volume map sampling, e71
 Exercise
 Andersen thermostat, 259
 configurational-bias Monte Carlo, 451
 free energy, 231
 Gibbs-ensemble technique, 231
 hard spheres, 230
 Ising model, 230
 Lennard-Jones, 94, 175
 Molecular Dynamics, 175, 259
 Monte Carlo, 94, 95, 230
 Monte Carlo integration, 93
 NPT ensemble, 230
 NVT ensemble, 259
 phase equilibrium, 231
 photon gas, 94
 statistical mechanics, 49–51
 vapor-liquid equilibrium, 231
 Widom method, 231
 Expanded-ensemble method, 464
 External potential energy, 358, 419
- F**
 Fast multipole methods, 394
 Fcc
 free energy, 344
 Fick's law, 148
 Finite-size corrections
 chemical potential, 286
 Finite-size effects
 free energy, 340, e100
 Ising model, e15
 Lennard-Jones (2d), e16
 surface, 135
 First law of thermodynamics, 12, 17
 Fluctuations
 non-equilibrium thermodynamics, 597
 Force calculation
 Algorithm, 103
 Automatic Differentiation, 102
 Forward-flux sampling, 547
 jumpy, 548
 Fourier Part of Ewald Sum, 379
 Fourier transformation, 376
 Free energy

- constrained Einstein crystal, 340
 density of states, 303
 Einstein crystal, 327
 Exercise, 231
 expanded-ensemble method, 464
 finite-size effects, 340, e100
 fixed center of mass, 340
 Gibbs, 18, 21
 Helmholtz, 18, 20
 lattice-coupling-expansion method, e71
 lattice-switch Monte Carlo, 344
 metadynamics, 309
 Molecular Einstein Crystal, 331
 multi-canonical ensemble method, e18
 non-equilibrium work, 317
 self-consistent histogram method, 315
 solid, 326, 344
 thermodynamic integration, 324
 Wang-Landau method, 304
 Free energy of a solid
 Example, e71
 Free-energy profiles
 MBAR, 312
 Fugacity, 440
 Fugacity coefficient, e69
- G**
- Gauss-Legendre quadrature, 339, e100
 Generalized coordinates, 90
 ensemble average, 91
 Gibbs free energy, 18, 21, 187
 statistical thermodynamics, 34
 Gibbs-Bogoliubov inequality, 278
 Gibbs-Duhem, 593
 Gibbs-Duhem integration
 Example, 213, 272
 Gibbs-ensemble technique
 acceptance rule, 219, e52
 Algorithm, 222
 analyzing the results, 224
 Case Study, e87
 chain molecules, e50
 configurational-bias Monte Carlo, e50
 critical exponents, 225
 critical point, 224
 dense liquids, e71
 density evolution, 223, e88
 dipolar hard-sphere fluid, 227
 Example, 223, 226, 227, e71
 Exercise, 231
 excluded volume map sampling, e71
 finite-size effects, 225, e14
 free energy, e9
 free energy density, e7
 ionic fluid, 226
 law of rectilinear diameter, 224
 Lennard-Jones fluid, 224, e89
 mixtures, 227
 partition function, 218
 polar fluid, 226
 probability density, 218
 restricted primitive model, 227
 saddle point theorem, e7
 scaling law, 225
 schematic sketch, 218
 Stockmayer fluid, 227
 thermodynamic limit, e7
 Grand Potential, 36
 Grand-canonical ensemble, 34
 Case Study, e87
 chain molecules, 441
 Example, 206
 justification of the algorithm, 201
 Monte Carlo technique, 195
 schematic sketch, 197
 Green-Kubo integrals
 optimal truncation, 162, 177
 Green-Kubo relation, 43, 153
 Greengard and Rokhlin
 Example, 402
- H**
- Hamilton formalism
 statistical mechanics, 580
 Hamiltonian, 25, 573
 Example, 579
 Hamiltonian (non-) system
 Liouville theorem, 588
 Hamiltonian thermodynamic integration, 40
 Hard spheres
 Case Study, e96
 chemical potential, 336, e97
 equation of state, 336, e97
 Example, 213, 335, 344
 Exercise, 230

- free energy (finite-size effects), 340, e100
 free energy solid, 344
 freezing, 213
 solid-liquid phase equilibrium, 335, e96
- Harmonic oscillator
 Andersen thermostat, 254, e92
 Case Study, e92
 Example, 254, e25
 Nosé-Hoover chains, e93
 Nosé-Hoover thermostat, 254, e92
 trajectories, 255, e25
- Hcp
 free energy, 344
- Heat, 12
 Heat capacity, 129, 174
- Helmholtz free energy, 18, 20, 185
 definition, 28
 elastic constants, 613
 excess, 185
 ideal gas, 185
 statistical thermodynamics, 33
- Henry coefficient, 362
- Histogram reweighting technique
 Example, 462
- Holonomic constrain, 522
- Hybrid Monte Carlo, 468
- Hyper-parallel tempering, 462
- Hyper-sphere approach, 402
- Hyperdynamics, 558
- I**
- Ideal chain, 358, 419, 441
 chemical potential, 443
- Importance sampling, 57
- Importance-sampling scheme, 56
- Initialization
 Algorithm, 101
- Interfacial tension, e5
- Internal energy, 128
- Internal potential energy, 358
- Interstitial
 concentration, 346
- Ion
 Example, 550
- Irreducibility
 Monte Carlo technique, 61
- Ising model
- Exercise, 230
 finite-size effects, e15
- Isobaric-isothermal ensemble, 33
 Case Study, e86
 Example, 191, 192
 Monte Carlo technique, 184
 schematic sketch, 185
- Isotension-isothermal ensemble
 Monte Carlo technique, 193
- J**
- Jacobian, 90, 582
 elastic constant, 614
- Jarzynski's identity, 317
- Jumpy forward-flux sampling, 548
- K**
- Kinetic Monte Carlo, 469
- Kirkwood g factor, 385
- Kirkwood-Buff, e5
- Kirkwood's coupling-parameter method, 277
- L**
- Lagrange interpolation method, 392
- Lagrange multiplier, 497
- Lagrangian, 493, 573
 Example, 576
- Lagrangian strain tensor, 612
- Landau Free Energies, 296
- Landau free energy, 297
- Langevin Dynamics
 molecular dynamics, 242
- Lattice-Boltzmann method, 569
- Lattice-coupling-expansion method, e71
- Lattice-switch Monte Carlo, 344
- Law of rectilinear diameter, 224
- Leap Frog algorithm, 113
- Lennard-Jones
 Andersen thermostat, 238, e89
 Case Study, e89
 chemical potential, 206, 283, 284, 290, 292, e93
 diffusion, 163, 164, e83
 energies, 145, e81
 equation of state, 75, 77, 80, 87, 191, 206, e85

- Example, 75, 79, 86, 144, 163, 164, 191, 192, 206, 223, 238, 253, 283, 290, 462
- Exercise, 94, 175
- finite-size effects, e16
- force, 104
- Gibbs ensemble, 224, e89
- mean-squared displacement, 163, 239, 254, e90
- Molecular Dynamics, 144, 163, e79
- Nosé-Hoover thermostat, 253, e90
- phase diagram, 70, 224, e89
- phase equilibria, 192, e86
- radial distribution function, 146, e82
- statistical error, calculation of, 145, e81
- truncated and shifted potential, 144, e80
- truncation of the potential, effect of, 70
- vapor-liquid coexistence, 192, e87
- velocity autocorrelation function, 163, e83
- velocity distribution, 253, e91
- Lennard-Jones chains
- Case Study, e104
 - equation of state, 423, e101
 - Example, 423, 449
 - recoil growth, 449, e104
- Linear response theory, 38
- dissipation, 607
 - dynamic, 41
 - static, 39
- Linked lists, 629
- Algorithm, 630, 631
- Liouville, e34
- Liouville formulation, 118
- multiple time step, 509
 - Nosé-Hoover algorithm, e33
- Liouville operator, 118
- Liouville theorem
- non-Hamiltonian system, 588
- Long-range interactions, 68
- damped truncation, 393
 - Event-Driven Monte Carlo, 402
 - Ewald summation, 373
 - Example, 402
 - fast multipole methods, 394
 - for Monte Carlo simulations, 398
 - hyper-sphere approach, 402
- Maxwell equations method, 399
- particle-mesh approaches, 388
- Lyapunov instability, 106
- ## M
- Machine learning, 5, 475
- finding collective variables, 558
 - invertible mappings, 475
 - Monte Carlo sampling, 475
- Markov process, 59
- Markov-Chain Monte Carlo Method, 54
- Matrix
- antisymmetric, 581
 - Jacobian, 582
 - symplectic, 583
 - transposed, 582
- Maxwell construction, 77, e76
- Maxwell Equations Method, 399
- Maxwell-Boltzmann distribution, 101
- Mean first-passage time, 552
- Mean-squared displacement
- Algorithm, 156, 160
- Measurements
- numerical, 125
 - physical, 125
- Mechanical quantities, 275
- Mesoscale dynamics, 569
- Mesoscopic model, 560
- Metadynamics
- free energy, 309
- Methane
- adsorption isotherm, 203
 - Example, 202
- Metropolis method
- Monte Carlo, 56
 - schematic sketch, 57
- Micro-canonical ensemble, 31
- Monte Carlo technique, e5
 - partition function, 583
- Microscopic reversibility, e78
- Mobility, 43
- Model fluid
- alkanes, e52
 - $\text{Br}_2\text{-Cl}_2\text{-BrCl}$, 212
 - dipolar hard-sphere fluid, 227
 - hard spheres, 213, 335, 344, e96
 - ideal chains, 429, e101
 - ions in water, 550

- Lennard-Jones, 75, 79, 86, 144, 163, 164, 191, 192, 206, 223, 238, 253, 283, 290, 462, e82
- Lennard-Jones chains, 423, e101
- Lennard-Jones dumbbell, 510, e107
- methane, 202
- point dipoles, 413
- polymers, 463
- restricted primitive model, 227
- soft spheres, 272
- Stockmayer fluid, 227, 277
- Molecular Dynamics
- Algorithm (NVE), 100, 101, 105
 - Algorithm (NVT), 240, 241
 - boundary conditions, 65
 - Case Study, e79
 - equilibration, 73
 - Example, 144, 163, 164
 - Exercise, 175, 259
 - initialization, 72
 - Langevin dynamics, 242
 - Lennard-Jones, 144, 163, 164, e79
 - Liouville formulation, 118
 - NPT ensemble, 258
 - NVE ensemble, 98
 - NVT ensemble, 234, 254
 - potential, truncation of, 67
- Monte Carlo, 54
- beyond detailed balance, 486
 - cell-veto method, 490
 - cluster moves, 479
 - dynamic, 469
 - end-bridging, e48
 - enhanced sampling, 473
 - event chain, 487
 - Exercise, 94, 95, 230, 259
 - hybrid, 468
 - kinetic, 469
 - machine learning, 475
 - mapping, 473
 - Metropolis method, 56
 - noise, 467
 - rebridging, e48
 - rejection free, 468
 - sampling rejected moves, 471
 - waste recycling, 471
- Monte Carlo integration
- Exercise, 93
- Monte Carlo technique
- Algorithm (μ VT), 201
 - Algorithm (fixed center of mass), 334
 - Algorithm (f VT), 200
 - Algorithm (Gibbs), 222, e21
 - Algorithm (NPT), 189, 190
 - Algorithm (NVT), 63, 64
 - boundary conditions, 65
 - canonical ensemble, 182
 - canonical ensemble: justification of algorithm, 183
 - Case Study, e86
 - configurational-bias Monte Carlo (lattice), 417, 418
 - efficiency, 188
 - equilibration, 73
 - Example, 75, 79, 86, 191, 192, 206, 223, 335
 - grand-canonical ensemble, 195
 - initialization, 72
 - isobaric-isothermal ensemble, 184
 - isotension-isothermal ensemble, 193
 - justification, 182
 - Metropolis scheme, 57
 - micro-canonical ensemble, e5
 - orientational bias, 409
 - path ensemble, 543
 - potential, truncation of, 67
 - Multi-Canonical Ensemble method, e18
 - Multi-canonical method, 345
 - Multi-particle collision dynamics, 567
 - Multiple time step
 - Algorithm, 513
 - Case Study, e107
 - Example, 510
 - Liouville formulation, 509
- N**
- Navier-Stokes equation, 621
 - Neighbor list, 625
 - Non-equilibrium thermodynamics, 595
 - fluctuations, 597
 - Onsager reciprocal relations, 600
 - Non-equilibrium work, 603
 - Normalizing flow method
 - example, 477 - Nosé-Hoover algorithms, e32
 - Nosé-Hoover chains
 - equations of motion, e33

- harmonic oscillator, e93
- Liouville formulation, e33
- Trotter expansion, e33
- Nosé-Hoover thermostat
 - Algorithm, e39
 - Case Study, e90
 - diffusion, 254, e91
 - Example, 253
 - harmonic oscillator, 254, e92
 - Lennard-Jones, 253, e90
- NPT ensemble
 - Exercise, 230
- Nudged elastic band, 557
- NVE simulations
 - diffusion, 163, e83
- NVT ensemble
 - Exercise, 259
- O**
 - Onsager coefficient
 - transport, 152
 - Onsager Reciprocal relations, 600
 - Onsager regression hypothesis, 600
 - Onsager-Machlup action, 123
 - Order parameter, 266, 279
 - Order- n algorithm to measure
 - correlations, 155
 - Orientational bias, 407
 - Overlapping distribution
 - Case Study, e94
 - chain molecules, 363
 - chemical potential, 284, 289
 - Example, 290
 - polymers, e57
 - Rosenbluth sampling, 363
- P**
 - Parallel Monte Carlo, 436
 - Parallel tempering, 457
 - acceptance rule, 458
 - Case Study, e105
 - Example, 459, 462, 463, 555, e74
 - Partial molar quantities, 128
 - Particle displacement
 - Algorithm, 64, 334
 - fixed center of mass, 333
 - Particle exchange
 - Algorithm, e21
 - Particle insertion method
 - Algorithm, 282
 - Case Study, e93
 - Example, 283
 - Particle-insertion method, 280
 - Particle-mesh approaches, 388
 - Particle-Mesh method, 373
 - Partition function, 28
 - μVT ensemble, 197
 - canonical ensemble, 54
 - configurational part, 56
 - Einstein crystal, 327
 - elastic constant, 613
 - Gibbs ensemble, 218
 - grand-canonical ensemble, 197
 - isobaric-isothermal ensemble, 186
 - microcanonical ensemble, 583
 - Nosé-Hoover, 246
 - NPT ensemble, 186
 - Path ensemble, 536, 537
 - acceptance rule, 544
 - Case Study, e112
 - Example, 541
 - Monte Carlo technique, 543
 - shifting moves, 546
 - shooting moves, 545
 - Path-sampling techniques, 534
 - Periodic boundary conditions, 65
 - minimum image convention, 67
 - nearest image convention, 67
 - schematic representation, 65
 - Phase equilibria
 - alkanes, e52
 - $\text{Br}_2\text{-Cl}_2\text{-BrCl}$, 212
 - Case Study, e96
 - Example, 192, 223, 335, 462
 - Exercise, 231
 - freezing soft spheres, 272
 - Gibbs-Duhem integration, 271
 - Gibbs-ensemble technique, 217
 - hard spheres, 335, e96
 - Lennard-Jones, 192, 223, e86
 - semigrand ensemble, 212
 - zero pressure, 192, e86
 - Phase space, 31
 - coordinates, 39
 - Phase space compressibility
 - dynamical system, 587

- Phase transitions
 first order, 267
 direct coexistence, 268
- Phenomenological rate equations, 516
- Photon gas
 Exercise, 94
- Poisson equation, 390
- Poisson's equation, 379
- Polarization, 507
- Polydispersity
 Example, 213
- Polymers
 Example, 438, 463
- Potential, truncation of, 67
 simple truncation, 69
 truncated and force-shifted, 71
 truncated and shifted, 71
- Potential energy
 bonded, 419
 external, 358
 from radial distribution function, 146, e81
 internal, 358
 nonbonded, 419
- Power spectrum, 44
- Predictor-corrector algorithm, 111, e3
- Pressure, 23, 130
 from radial distribution function, 146, e82
 thermodynamic integration, 133
 trial volume method, 321
 virtual volume changes, 134
- Probability density
 canonical ensemble, 183
 Gibbs ensemble, 218
 grand-canonical ensemble, 199
 isobaric-isothermal ensemble, 187
 NPT ensemble, 187
- Pruned-enriched Rosenbluth method, 366
- Q**
 Quaternion, 88
- R**
 Radial distribution function, 140
 Algorithm, 147
 force method, 143
 potential energy, 146, e81
- pressure, 146, e82
- Random-number generators, 61
- Rare events, 515
 Bennet-Chandler approach, 520
 Blue-Moon ensemble, 522
 Case Study, e109
 committor, 550
 diffusive barrier crossing, 527
 Example, 524, 541
 forward-flux sampling, 547
 jumpy forward-flux sampling, 548
 mean first-passage time, 552
 path-sampling techniques, 534
 saddle point, 556
 transition interface sampling, 546
 transition path theory, 549
 transition state theory, 521
 transition-path sampling, 535
- Reaction coordinate, 266
- Rebridging Monte Carlo, e48
- Recoil growth
 acceptance rule, 449
 algorithm, 446
 Case Study, e104
 Configurational-bias Monte Carlo, versus, 445
 Example, 449
 justification of the method, e54
 Lennard-Jones chains, 449, e104
 super-detailed balance, e55
- Recursive sampling
 chemical potential, 365
- Reduced units, 73, 92
- Rejection-free Monte Carlo, 468
- Reptation moves, 413
- Restricted primitive model, 227
- Rosenbluth factor, 408, 414
- Rosenbluth sampling, 352
 chemical potential, 356
 continuum model, 357
 ensemble average, 356
 lattice model, 353
 overlapping distribution, 363
 pruned-enriched, 366
- S**
 Saddle point, 556
 activation-relaxation technique, 557
 hyperdynamics, 558

- nudged elastic band, 557
- temperature-accelerated dynamics, 558
- Scaling law, 225
- Scanning method, 368
- Second law of thermodynamics, 12, 15
- Self interaction
 - Ewald summation, 380
- Self-consistent histogram method
 - Example, 315, 462
- Self-diffusion coefficient, 176
- Self-interaction, 376
- Semigrand ensemble, 206, e50
 - Example, 212, 213
- Shadow orbit, 109, 123
- SHAKE, 498
 - constrained dynamics, 510, e107
- Shear stress, 612
- Shear viscosity, 153, 612
- Shifting moves
 - path ensemble, 546
- Shooting moves
 - path ensemble, 545
- Single-occupancy cell method, 325
- Smoothed dissipative particle dynamics, 566, 621
- Solid
 - Example, e71
 - free energy, 326
- Static Monte Carlo sampling
 - example, 477
- Static properties
 - compressibility, 135
 - heat capacity, 129
 - internal energy, 128
 - partial molar quantities, 128
 - pressure, 130
 - pressure by thermodynamic integration, 133
 - pressure by virtual volume changes, 134
 - radial distribution function, 140
 - structure factor, 139
 - surface free-energy density, 138
 - surface stress, 138
 - surface tension, 135
 - surface tension from virtual moves, 137
 - Tension of curved surfaces, 138
 - Statistical error, 167
 - block averages, 171
 - correlation functions, 169
 - system size, 167
 - Statistical mechanics, 11
 - classical, 28
 - Exercise, 49–51
 - Hamilton formalism, 580
 - Statistical thermodynamics, 11, 25
 - Gibbs free energy, 34
 - grand potential, 36
 - Helmholtz free energy, 33
 - Steady-state velocity, 43
 - Stirling approximation, 344
 - Stockmayer fluid, 277
 - Gibbs ensemble, 227
 - Structure factor, 139, 591
 - Super-detailed balance, 412, 422, e55
 - Surface free-energy density, 138
 - Surface stress, 138
 - Surface tension, 135, e4
 - sketch, 136
 - virtual moves, 137
 - Swendsen-Wang
 - Ising model, 480
 - Symplectic, 121
 - canonical transformation, 583
 - condition, 583
 - equations of motion, 581
 - matrix, 583
 - System size
 - statistical error, 167
 - T**
 - Tail corrections, 371
 - chemical potential, 284, e94
 - energy, 68
 - Lennard-Jones, 69
 - pressure, 70
 - Temperature, 127
 - classical thermodynamics, 14
 - instantaneous, 98
 - kinetic energy, 127
 - microcanonical ensemble, 98
 - scaling, 99
 - thermodynamic definition, 27
 - Temperature-accelerated dynamics, 558
 - Thermal conductivity, 154
 - Thermal quantities, 275
 - Thermodynamic integration, 274

- chain molecules, 351
- coupling parameter, 277
- free energy, 324
- Hamilton, 40
- Thermodynamics, 11, 12
 - first law, 12
 - second law, 12
 - third law, 18
- Third law of thermodynamics, 18
- Transition interface sampling, 546
- Transition matrix, 59
- Transition path sampling
 - Example, 550
- Transition path theory, 549
- Transition state ensemble, 552
- Transition state theory, 521
- Transition-path sampling, 535
 - Example, 555
- Transposed matrix, 582
- Trial moves
 - linear rigid molecules, 88
 - non-rigid molecules, 89
 - nonlinear rigid molecules, 88
 - orientational moves, 87
 - translations, 81
- Trotter expansion, e34
 - Nosé-Hoover algorithm, e33
- Trotter identity, 119
- U**
 - Umbrella sampling, 521
 - Example, 302
- United-atom model
 - alcohols, e54
 - alkenes, e54
 - alkylbenzenes, e54
 - branched alkanes, e54
 - linear alkanes, e54
- Units, reduced, 73
- V**
 - Vacancies
 - concentration, 346
 - free energy, 346
- Van der Waals equation of state, 46
- Vapor-liquid equilibrium
 - Exercise, 231
- Velocity autocorrelation function
 - Algorithm, 156, 160
- Velocity Verlet, 121
 - Algorithm, 115
 - Velocity Verlet algorithm, 113
 - in Andersen thermostat, 240, 241
 - in Nosé-Hoover thermostat, e32
 - Liouville formulation, 510
 - multiple time step, 513, 514
 - Velocity-corrected Verlet algorithm, 114
 - Verlet algorithm, 105, 112, 122, 498
 - Algorithm, 105
 - in NVE simulation, 105
 - Verlet lists, 625
 - Algorithm, 627–629
 - Case Study, e116
 - Example, 634
 - Viscosity, 611
 - Volume change
 - Algorithm, 190, e19
 - energy difference, cheap way of calculating, 188
 - Gibbs-ensemble technique, 220
 - molecules, 191
 - Voronoi, 86
- W**
 - Wang-Landau method, 304
 - Waste recycling, 471
 - Water
 - Example, 550
 - WF potential
 - Algorithm force calculation, 103
 - Widom method, 280
 - Algorithm, 282
 - Case Study, e93
 - Example, 283
 - Exercise, 231
 - mixtures, 287
 - Wiener-Khinchin theorem, 44
 - Wigner-Seitz cell, 343, 346, 349
 - definition, 343
 - Wolff algorithm, 482
 - Work, 12
 - Wormhole moves, 413
- Z**
 - Zeolite
 - adsorption isotherm, 203
 - Example, 202, 443
 - structure, example of a, 203
 - structure solution, e74

Author index

A

Aasen, A., 71, 223, 269, e88
Abascal, J.L.F., 324
Abbott, N.L., 312
Abercrombie, S., 475, 479
Abraham, F.F., 92, 630
Abramowitz, M., 339, e100
Abrams, C.F., 460, e107
Ackland, G.J., 329, 340, 344, 346, e100
Adams, D.J., 181, 197, 286, 336, 337, e96, e97
Adhikari, R., 570
Adjanor, G., 473
Adolf, D.B., e41, e42
Agrawal, R., 271–273
Ailawadi, N.K., 169
Alder, B.J., 4, 5, 97, 213, 268, 272, 336, 346, 349, 560, e96, e97
Alejandre, J., 394
Alexander, F.J., 567
Alexandrowicz, Z., 446
Alfe, D., 269
Allen, M.P., 6, 7, 40, 60, 88, 97, 130, 218, 358, 374, 393, 401, 469, 484, 506, e14, e61, e122
Allen, R.J., 547
Almarza, N.G., 277
Amar, J.G., 228
Amon, L.M., 81, e78
Andersen, Hans C., 113, 194, 234, 237–239, 241, 245, 258, 310, 350, 499, 506, e89
Anderson, H.L., 54
Andoh, Y., 394
Anselme, M.J., e53
Antillon, E., 324

Antoine, R., 308
Appel, A.W., 394
Araque, J.C., 547
Arnold, A., 371, 389, 393, 398, 401, 403, 404
Assaraf, R., 141, 143, 144
Athenes, M., 473
Attard, P., 186
Auer, S., 138, 463, 556
Auerbach, D.J., 630
Avalos, J.B., 566
Azhar, F.E., 271

B

Bachman, J., 250
Backx, G., 562, 566
Baerends, E.J., 131
Bakhshandeh, A., 282
Bakker, A.F., 630, e5
Ballard, A.J., 475, 479
Ballenegger, V., 373, 386
Banaszak, B.J., e54
Barber, M., e41, e42
Barducci, A., 310, 311
Barkema, G.T., 7, 557
Barker, A.A., 473
Barker, J.A., 4, 272, 325
Barnes, J., 394
Baroni, S., 154, 162, 463
Bartell, L.S., 554
Bastolla, U., 367
Bates, M.A., 350, 616
Batoulis, J., 361, 364, 414
Battimelli, G., 3, 5, 54
Baus, M., 7
Baus, Marc, 366

- Bauss, Marc, 226
 Beauchamp, K.A., 243
 Beckers, J.V.L., 390
 Bedeaux, D., 592
 Bee, M., 153
 Beeler, J.R., 4, 5
 Bekker, H., 626, 627
 Belardinelli, R.E., 306
 Bell, A.T., 202, 362, 363
 Belonoshko, A.B., 269
 Bennett, C.H., 289, 293, 346, 349, 515,
 e60
 Benzi, R., 569, 570
 Berendsen, H.J.C., 97, 118, 236, 244, 252,
 256, 498, 511, 626, 627, e108
 Berens, P.H., 113, 241
 Berg, B.A., 302, 311, 345, e18
 Berger, J.O., 84
 Berkowitz, M.L., 389, 391–393, e42, e43
 Bernal, J.D., 3
 Bernard, E.P., 79, 487, 489
 Bernardo, J.M., 84
 Berne, B.J., 7, 118, 181, 468, 493, 516,
 535, 556, e61, e122
 Berthier, L., 92
 Bhatt, R.N., 305, 306
 Biben, T., 436, 438
 Binder, K., 7, 97, 167, 193, 225, 266, 290,
 361, 413, 473, 616, e15
 Binder, Kurt, 534, 535, 617
 Bird, G.A., 567
 Bladon, P., 436
 Blander, M., 507
 Blas, F.J., 138
 Blumich, B., 250
 Blundell, C., 475, 479
 Board, J.A., 394
 Boggs, E.M., 40
 Bolhuis, P.G., 214, 215, 271, 323, 329,
 332, 340, 344, 436, 438–440,
 535, 537, 539, 541, 544,
 546–548, e100, e112
 Bolten, M., 371, 389, 393, 398, 401, 403,
 404
 Bonella, S., 506, 509
 Bonomi, M., 279, 311
 Boone, T.D., 91, 432, e48–e50
 Borgis, D., 141, 143, 144
 Borkovec, M., 552
 Borrero, E.E., 547
 Bortz, A.B., 470, 471
 Boulogouris, G.C., 471
 Bowles, R.K., 349
 Box, G.E.P., 58, 99, 328, 474, 639
 Branduardi, D., 310
 Braun, E., 236, 237, 244, 250, 256
 Brey, J.J., 523, 531, 534
 Bridgman, P.W., 323
 Brodka, A., e41, e42
 Brooks, B.R., 371
 Broughton, J.Q., 326
 Brown, B.C., 272, 325
 Brown, D., 393
 Brown, G., 308
 Brown, L.D., 477
 Browning, N.D., 325, 326, 331
 Broyer, M., 308
 Bruce, A.D., 193, 216, 225, 317, 329,
 340, 344, 346, 462, 463, e100
 Bruin, C., e5
 Brush, S.G., 373
 Buff, F.P., 591, 592, e5
 Bukov, M., 6
 Bunker, A., 463, 464
 Burton, J.J., 515
 Bussi, G., 238, 245, 253, 256, 257, 259,
 310, 311, e89, e91
- C**
 Cabriolu, R., 98, 249, 250
 Cacciuto, A., 138
 Cahn, J.W., 138, 283
 Caillol, J.M., 227, 402, 413
 Calvo, F., 304, 306, 308, 309
 Camilloni, C., 279
 Campbell, A.I., 331
 Cape, J.N., 268, 272
 Car, R., 493, 506
 Carnahan, N.F., 277
 Cass, M.J., 151
 Cates, M.E., 570
 Catlow, C.R.A., 202, 218
 Cazorla, C., 269
 Ceperley, D.M., 467, 468, 471
 Ceriotti, M., 257, 328

- Chandler, D., 39, 312, 432, 471, 515, 516, 527, 535, 537, 539, 541, 552, e63, e64, e112
- Chao, K.C., 197
- Chapela, G.A., 394
- Charbonneau, P., 226, 270, 324, 329, 332
- Chartier, P., 398
- Charvolin, J., 443
- Cheatham, T.E., 236, 237, 244
- Chen, B., 427, e54
- Chen, S., 569, 570
- Chen, W., 558
- Chen, Z., 436
- Cheng, B.Q., 328, 592
- Cheng, H., 394, 403
- Chester, G.V., 471
- Chipot, C., 268
- Cho, K., e29
- Chodera, J.D., 243, 312, 320
- Chopra, M., 268, 303
- Christova, C.G., 331
- Chung, S.T., 287
- Ciccotti, G., 1, 3, 5, 7, 40, 54, 90, 91, 97, 118, 163, 251, 252, 285–288, 328, 341, 374, 473, 498, 505, 506, 509, 511, 516, 522, 524, 527, 535, 556, 587, 610, e22, e24, e32, e108, e112
- Ciccotti, Giovanni, 534, 535, 617
- Clarke, J.H.R., 393, e41, e42
- Cochran, H.D., 154, e54
- Cohen, L.K., 197
- Coker, D.F., 7, 516, 535, 556
- Coluzza, I., 473
- Colwell, K.A., 143, 144
- Consta, S., 446, e57
- Coppersmith, S.N., 467
- Coretti, A., 479, 506, 509
- Coslovich, D., 92
- Creutz, M., 181, e5
- Crooks, G.E., 243, 317, 318, 320, 603
- Crowhurst, J.C., 330
- Crozier, P.S., e43
- Csajka, F.S., 535, 541, e112
- Cuadros, F., 277
- Cui, S.T., 154, e54
- Cummings, P.T., 154, e54
- D**
- Dachsel, H., 371, 389, 393, 398, 401, 403, 404
- Daivis, P.J., 134
- Dalke, A., 94
- Dama, J.F., 310, 311
- Darden, T.A., 389–393
- Darrigrand, E., 398
- Dauxois, T., 4
- David, A., 154
- Davis, H.T., 202, e71
- Dawid, A.P., 84
- Day, A.G.R., 6
- Dayal, P., 467
- de Gennes, P.G., 227
- de Groot, S.R., 38, 148, 567, 595, 599, 621
- de la Cruz, M.O., 211
- de las Heras, D., 141
- de Leeuw, S.W., 227, 277, 373, 374, 390, 499, e42
- de Miguel, E., 138, 228, e17
- De Nicola, A., 154
- de Pablo, J.J., 211, 268, 272, 303, 304, 308, 309, 312, 352, 413, 461–463, e20, e52, e54
- de Smedt, Ph., 224, 270, e10, e14, e15
- de Swaan Arons, J., 227
- de With, G., 487, 489
- Debenedetti, P.G., 304, 308, 309, 312, 317
- Deem, M.W., 79, 460, 471, e48–e50, e74, e107
- Deighan, M., 311
- Deitrick, G.L., e71
- Dellago, C., 154, 304, 306, 319, 479, 535, 537, 539, 541, 544, 546, 552, e112
- Deserno, M., 389–393
- Dewing, M., 467, 468
- DeWitt, H.E., 272
- Dijkstra, E.J., 626, 627
- Dijkstra, M., 226, 271, 324, 331, 432, 436, 438, e47
- Dingle, Robert B., e7
- Dinola, A., 236, 244, 252, 256
- Dobnikar, J., 72, 74, 93, 103
- Dodd, L.R., 91, e48, e49
- Dohnalkova, A., 325, 326, 331

- Doll, J.D., 457, 460, e107
 Donadio, D., 238, 245, 253, 256, e89, e91
 Doolen, G.D., 569, 570
 Doxastakis, M., 312
 Dress, C., 483, 485
 Dror, R.O., 115–117, 127, 128, 237
 Duane, S., 468, e61
 Dugourd, P., 308
 Dunweg, B., 151, 400, 401, 403, 463,
 464, 570
 Dupuis, P., 460, e107
- E**
 E, W.N., 534, 535, 544, 549
 Eastwood, J.W., 7, 113, 389, 392, 393,
 403, 629
 Eastwood, M.P., 115–117, 127, 128, 237
 Eggebrecht, J., 393
 Ehrenreich, H., 194, 612, 616
 Eisenthal, K.B., 396
 Elber, R., 123, 547, 548
 Eldridge, M.D., 331
 Elliott, W.D., 394
 El'yashevich, A.M., 4
 Eppinga, R., 135, 187
 Ernst, M.H., 562, 566
 Erpenbeck, J.J., 181
 Escobedo, F.A., 272, 436, 467, 547, e54
 Espanol, P., 561, 562, 564, 566, 567,
 621–624, e65, e66
 Espinosa, J.R., 549
 Esselink, K., 388, 436, 437, e63
 Essmann, U., 389, 391–393
 Evans, D.J., 7, 134, 161, 236
 Evans, R., 443
 Evans, W.A.B., 151
 Ewald, P.P., 373, 374
 Exler, M., 303
- F**
 Fahrenberger, F., 371, 389, 393, 398, 401,
 403, 404
 Falcioni, M., 460, e74, e107
 Falkner, S., 479
 Falkowski, J.M., 143, 144
 Faller, R., 308, 309, 312, e20
 Faou, E., 398
 Faradjian, A.K., 547, 548
 Farago, O., 235, 564, 616
 Farha, O.K., 325, 326, 331
 Fass, J., 243
 Faulkner, M.F., 79, 402, 487, 490, 491
 Feller, W., 238
 Ferguson, A.L., 558
 Fermi, E., 4
 Ferrario, M., 7, 97, 473
 Ferrario, Mauro, 534, 535, 617
 Ferraz-Mello, S., 121
 Ferrenberg, A.M., 290
 Fetisov, E.O., 129
 Feynman, Richard P., 537, 573
 Filinov, V.S., 181, 196
 Filion, L., 226, 324
 Fischer, J., 286
 Fisher, C.K., 6
 Fixman, M., 90, 91, 499
 Flannery, B.P., 7, 44, 61, 140, 173, 389,
 424, 425, 638
 Flyvbjerg, H., 145, 165, 172–174, e80,
 e84
 Forrest, B.M., 469, e61, e62
 Frantz, D.D., 457
 Frauenkron, H., 367
 Freeman, D.L., 457
 Frenkel, D., 30, 69, 72, 74, 82, 92, 93,
 103, 135, 138, 154, 187, 207,
 212, 218, 224, 226, 232,
 270–272, 285–288, 319, 323,
 324, 326, 328–332, 336, 337,
 339, 340, 344, 350, 352, 364,
 366, 413, 423, 424, 432, 436,
 438–440, 446, 463, 468, 471,
 473, 484, 523, 531, 534, 547,
 556, 561, 563, 616, e10, e14,
 e15, e17, e18, e52, e53, e57,
 e58, e71, e97, e98, e100, e101
 Frisch, U., 569
 Fukuda, I., 393
- G**
 Gabrie, M., 475, 476, 478, 479
 Gahler, F., 371, 389, 393, 398, 401, 403,
 404
 Galán, C.A., 277
 Galli, G., 506
 Gans, J., 115, 116, 127
 Garaizar, A., 549

- Garcia, A.L., 567
 Garel, T., 365
 Garrahan, J.P., 471
 Gehlen, P.C., 4, 5
 Geissler, P.L., 479, 486, 535, 544, 546, 552
 Gelb, L.D., 202
 Gelman, A., 84
 Gerstner, E., 367
 Gervasio, F.L., 310
 Geyer, C.J., 457
 Ghazisaeidi, M., 324
 Ghoufi, A., 137, 138, 403
 Ghulghazaryan, R.G., 317
 Gibson, J.B., 4
 Gilks, W.R., 84
 Gillan, M.J., 269
 Gillespie, D.T., 470, 471
 Gillilan, R.E., 110, 122, 123
 Gilmer, G.H., 326
 Gingold, R.A., 566, 621
 Glandt, E.D., 207, 210, 212, 213, 228
 Gloor, G.J., 138
 Glosli, J., 390
 Goland, A.N., 4
 Goldstein, Herbert, 37, 245, 494, 573, 575
 Gomez-Gualdrón, D.A., 325, 326, 331
 Gompper, G., 560, 568, 569
 Goncharov, A.F., 330
 Goodbody, S.J., 202, 203
 Gordon, P.A., e54
 Gould, H., 7
 Grassberger, P., 365–367
 Grasselli, F., 162
 Gray, C.G., 507
 Gray, S.G., 272, 325
 Gray, S.K., 586
 Greco, P., 3, 5, 54
 Greengard, L., 394, 403
 Greenkorn, R.A., 197
 Gregg, S.J., 443
 Grest, G.S., e54
 Griffiths, R.B., 352
 Gronbech-Jensen, N., 235, 564
 Groot, R.D., 561, 563, 564, e115
 Grzybowski, A., e41, e42
 Gubbins, K.E., 76–78, 80, 87, 92, 191, 192, 202, 206, 211, 223, 277, 289, 507, e60, e76–e78, e80, e86–e88
 Gubbins, Keith E., 137
 Gubenatis, J.E., 54
 Gubernatis, J.E., 460, e107
 Gude, M., e53
 Gull, E., 467
 Gwozdz, E., e41, e42
- H**
- Haag, W.O., 271
 Haak, J.R., 236, 244, 252, 256
 Hafskjold, B., 71, 154, 223, 269, e88
 Hagen, M.H.J., 271, 563
 Haile, J.M., 6, 97
 Haji-Akbari, A., 547, 549
 Halley, J.W., 507
 Halver, R., 371, 389, 393, 398, 401, 403, 404
 Hammer, M., 71, 223, 269, e88
 Hammonds, K.D., 121
 Hanggi, P., 552
 Hansen, J.-P., 39, 154, 267, 272, 372, 374, 506, e120
 Hansen, J.P., 213, 432, 436, 438, 560
 Hansen-Goos, H., 277
 Harismiadis, V.I., 227
 Harris, J., 352, 354, 413
 Harting, J., 256
 Hartmann, A.K., 320
 Harvey, A.H., 228
 Harvey, S.C., 236, 237, 244
 Hass, A.B., 71, 223, 269, e88
 Hasse, H., 76
 Hasslacher, B., 569
 Hautman, J., e42
 Haymet, A.D.J., 272
 Hayryan, S., 317
 Heber, F., 371, 389, 393, 398, 401, 403, 404
 Hecht, M., 256
 Heermann, D.W., 7, 469, e61, e62
 Hegger, R., 365
 Heinbruch, U., 286
 Hellekalek, P., e126
 Hemley, R.J., 330
 Henderson, D., 272, 325, e43
 Henderson, J.R., 137
 Hendriks, E.M., 227

- Henkelman, G., 557
 Hermann, D.W., 193, 225
 Herrmann, H.J., 256
 Hess, B., 398
 Heyes, D.M., 121, 151, 256, 375, e41, e42
 Hibbs, Albert R., 537
 Hilbers, P.A.J., 437, e63
 Hiwatari, Y., 268
 Hockney, R.W., 7, 113, 389, 392, 393, 403, 629
 Hoefsloot, H.C.J., 565, e115
 Hoeth, J.W., 446, e57
 Hofmann, M., 371, 389, 393, 398, 401, 403, 404
 Holm, C., 371, 389–393, 398, 401, 403, 404
 Holt, A.C., 616
 Hoogerbrugge, P.J., 561, 563
 Hoover, W.G., 1, 7, 40, 54, 97, 118, 238, 245, 252, 267, 272, 280, 285, 286, 325, 339, 344, 374, 610, 616, e23, e65, e89, e100
 Houdayer, J., 413, 432
 Howarth, A.J., 325, 326, 331
 Hoye, J.S., 227
 Hu, C.K., 223, 317, e88
 Huber, T., 309
 Hummer, G., 151, 534
 Humphrey, W., 94
 Huse, D.A., 311, 340, 346, 467, e100
 Hussain, S., 549
 Hut, P., 394
 Hutter, Jurg, 506
 Hynninen, A.P., 331
- I**
 Iba, Y., 456, 457, 464
 Ibarz, B., 479
 Ihle, T., 256, 560, 568, 569
 Ikeshoji, T., 154
 Ilario, G., e122
 Imhof, A., 331
 Impey, R.W., 616
 Irving, J.H., e5
 Isenberghausen, J., 371, 389, 393, 398, 401, 403, 404
 Ito, N., 223, e88
- J**
 Jablonka, K.M., 69
 Jackson, G., 138, 222, 229
 Jackson, J.D., 395
 Jackson, M.N., 143, 144
 Jacoboni, C., 470
 Jacobson, J.D., 5, 213, 272
 Jacucci, G., 163
 Jaffe, R.I., 4, 5
 Janke, W., 62
 Janosi, L., 312
 Jarzynski, C., 317, 473, 474, 478
 Jawahery, S., 143, 144
 Jensen, M.O., 115–117, 127, 128, 237
 Joannopoulos, J.D., e29
 Joanny, J.F., 443
 Jobic, H., 153
 Johansson, B., 269
 Johnson, J.K., 76–78, 80, 87, 191, 192, 206, 211, 223, 277, e76–e78, e80, e86–e88
 Johnson, K.W., 272, 325
 Jones, J.E., 68
 Jones, R.E., 171
 Jonsson, H., 557
 Jorgensen, W.L., e53
 Josephson, T.R., 129
 June, R.L., 202
- K**
 Kabadshow, I., 371, 389, 393, 398, 401, 403, 404
 Kahl, G., 329, 332
 Kalikmanov, V.I., 227
 Kalos, M.H., 7, 61, 470, 471
 Kamiya, N., 393
 Kantor, Y., 616
 Kapfer, S.C., 487–491
 Kapral, R., 516, 560, 567–569
 Karaborni, S., 437, 444, e52–e54
 Karavias, F., 202
 Kärger, J., 151
 Karger, J., 153
 Keblinski, P., 393
 Kennedy, A., 468, e61
 Kerkhof, P.J.A.M., e120
 Kevrekidis, I.G., 310
 Kim, E.B., 312
 Kincaid, J.M., 71

- Kiran, E., 226
 Kirkwood, J.G., 40, 277, 591, 592, e5
 Kjelstrup, S., 592
 Klein, M.L., 175, 252, 507–509, 511, 616, e24, e33–e35, e40–e42, e107, e122
 Kleinman, L., e29
 Knotts, T.A., 309
 Koelman, J.M.V.A., 561, 563
 Kofke, D.A., 207, 210, 212–215, 228, 270–273, 277, 324, 332, 475
 Kohler, J., 475–479
 Kolafa, J.A., 387
 Koonin, S.E., 7
 Koopman, E.A., 242
 Koper, G.J.M., 186
 Köster, A., 46
 Koutras, N.K., 227
 Kramer, G.J., 403
 Kramers, H.A., 554
 Kranendonk, W.G.T., 82, 92, 207, 332
 Krauth, W., 79, 402, 483, 485, 487–491
 Kremer, K., 151, 361, 364, 413, 414
 Krishna, R., 437, 438, 445, 450, e46, e47
 Kroll, D.M., 560, 568, 569
 Kron, A.K., 4, 413
 Kubo, R., 43
 Kullback, S., 476
 Kumar, S.K., 352, 431, e53, e104
 Kurzak, J., 394, 396, 397
- L**
- Ladd, A.J.C., 268, 272, 326, 330, 337, 339, 344, 570, 616, e71, e98, e100
 Laio, A., 304, 309, 310
 Laird, B.B., 272
 Lakshminarayanan, B., 475
 Lamura, A., 569
 Landau, D.P., 7, 266, 303, 304, 308
 Landau, L., 297, 323
 Langevin, P., 242
 Laso, M., 352, 413, e52
 Lebowitz, J.L., 98, 129, 181, 286, 470, 471
 Lechner, W., 319
 Lee, D.J., 92
 Lee, H., 389, 391–393
- Lee, Michael A., 394
 Lei, Z., 489
 Leibler, R.A., 476
 Leighton, Robert Benjamin, 573
 Leimkuhler, B., 6, 97, 106, 243
 Lekkerkerker, H.N.W., 128, 271
 Lennard-Jones, J.E., 68
 Lenz, O., 371, 389, 393, 398, 401, 403, 404
 Leroy, F., 138
 Lesar, R.A., 326, 330, e71
 Leunissen, M.E., 331
 Levelt Sengers, J.M.H., 226
 Levesque, D., 213, 227, 402, 560
 Levin, Y., 282
 Levrel, L., 399, 401
 Li, P., 325, 326, 331
 Likos, C.N., 329, 332
 Limcharoen, P., 202
 Lin, W.S., 475
 Lipkowitz, Kenny B., 310
 Lippert, R.A., 115–117, 127, 128, 237
 Liu, J.W., 483, 485, 486
 Liu, Y.F., 460, e107
 Liu, Yi, 251, 252, 587, e22, e24, e32
 Long, J.R., 143, 144, 250
 Lopez-Lemus, J., 394
 Louwerse, M.J., 131
 Lowe, C.P., 240, 242, 390, 562
 Löwen, H., 506
 Loyens, L.D.J.C., 436, 437
 Lu, J.F., 460, e107
 Lucy, L.B., 566, 621
 Luijten, E., 483, 485, 486
 Lustig, R., 46, 78, 285, 286
 Lutz, C., 630
 Lyubartsev, A.P., 457, 464
- M**
- Mace, A., 250
 Macedonia, M.D., 427
 Macgowan, D., 202, 203
 Machlup, S., 123
 Mackie, A.D., 566
 Madden, P.A., 68, 277, 331, 506, 507
 Maddox, M.W., 202
 Madura, J.D., e53
 Maesen, T.L.M., 202, 249, 443, 444

- Maggs, A.C., 79, 399, 401, 402, 487, 490, 491
 Maginn, E.J., 362, 363, 427
 Malevanets, A., 560, 567–569
 Malfreyt, P., 137, 138, 403
 Malijevsky, A., 138
 Malliakas, C.D., 325, 326, 331
 Mandadapu, K.K., 171
 Mandel, M.J., 66
 Manousiouthakis, V.I., 79, 471
 Maragakis, P., 115–117, 127, 128, 237
 Maragliano, L., 310
 Marchi, M., 512, 514, e108
 Marcolongo, A., 154, 162, 463
 Marechal, M., 226, 324
 Marinari, E., 457
 Marsh, C.A., 562, 563, 566
 Martin, M.G., 437, 438, 450, e47, e48, e54
 Martin del Rio, E., 228, e17
 Martinez-Veracoechea, F.J., 467
 Martiniani, S., 468
 Martsinovski, A.A., 457, 464
 Martyna, G.J., 118, 175, 251, 252, 468, 493, 509, 511, 512, 514, 587, e22, e24, e32–e35, e40, e41, e61, e107, e108, e122
 Marx, Dominik, 506
 Massobrio, C., 285, 287, 288, 328
 Masters, A.J., 562
 Mastny, E.A., 309
 Matsuda, H., 272, 325
 Matthews, C., 6, 97, 106, 243
 Mattson, W., 132, 133
 Mau, S.-C., 346
 Mau, S.C., 340, e100
 Mavrantzas, V.G., 432, e48, e50
 Mazur, P., 38, 148, 567, 595, 599, 621
 McCabe, C., e54
 McCormick, A.V., 202
 McDonald, I.R., 4, 39, 154, 163, 181, 184, 187, 267, 286, 301, 372, e120
 McNamara, G.R., 569, 570
 Mehdi, B.L., 325, 326, 331
 Mehlig, B., 469, e61, e62
 Mehta, P., 6
 Meijer, E.J., 212, 271, 325, 326, 330, e71
 Meirovitch, H., 368, 446
 Meiss, J.D., 586
 Meller, J., 123
 Mendoza, F.N., 394
 Mercado, R., 250
 Merolle, M., 471
 Metropolis, N., 4, 56, 57, 63, 281
 Meyer, M., 7, 97, 302, 505, 522, 524
 Mezei, M., 197, 205, e71
 Michel, M., 487, 489
 Milano, G., 154
 Milgram, M., 4
 Miller, M.A., 81, e78
 Miller, W.H., 527, e111
 Mills, G., 557
 Minh, D.D.L., 320
 Minkara, M.S., 129
 Mladek, B.M., 226, 270, 324, 329, 332
 Mohamed, S., 475
 Mon, K.K., 352, e15
 Monaghan, J.J., 566, 621
 Mondello, M., e54
 Monson, P.A., 324
 Mooij, G.C.A.M., 271, 352, 364, 413, 423, 424, e52, e53, e58, e101
 Moore, J.D., e54
 Moosavi, S.M., 236, 237, 244, 256
 Moroni, D., 546–548
 Morriss, G.P., 7, 161, 236
 Mountain, R.D., 85, 228, 268
 Mouritsen, O.G., 7, 266
 Mousseau, N., 557
 Moustafa, S.G., 475
 Mulder, B.M., 331
 Mulero, A., 277
 Muller, E.A., 138
 Muller, K., 477
 Müller, M., 351, 352
 Muller, M.E., 58, 99, 328, 474, 639
 Muller Plathe, F., 154
 Muller-Plathe, F., 138
 Mundy, C.J., 587
 Müser, M.H., 527, e112
 Myers, A.L., 202
- N**
- Nadler, W., 367
 Naghizadeh, J., 176
 Najafabadi, R., 181, 194
 Nakamura, H., 393

- Nalisnick, E., 475
 Nath, S.K., e54
 Neidinger, R.D., 102, 477, 499, 524
 Neuhaus, T., 302, 311, 345, e18
 Neumann, M., 386
 Newman, M.E.J., 7
 Ng, K.-C., 227
 Nicholson, D., 197
 Nicolas, J.J., 76, 78, e76, e77
 Nielaba, P., 193, 225, 616
 Nijmeijer, M.J.P., e5
 Nilmeier, J.P., 320
 Ninarello, A., 92
 Noe, F., 475–479
 Noid, D.W., 586
 Norman, G.E., 181, 196
 Nosé, S., 234, 245, 506
 Nowick, A.S., 515
 Noya, E.G., 324, 332
- O**
 Ogawa, T., 272, 325
 Ogita, N., 272, 325
 Ogura, H., 272, 325
 Okamoto, Yuko, 457
 Okazaki, S., 394
 O’Keeffe, M., 325, 326, 331
 Olender, R., 123
 Olsen, O.H., 252
 Olsson, S., 475–479
 Ongari, D., 69
 Onsager, L., 38, 123, 162, 277, 595, 600
 Orkoulas, G., e65
 Orland, H., 365
 Ostwald, W., 331
 Ozawa, M., 92
- P**
 Pagonabarraga, I., 561, 563
 Panagiotopoulos, A.Z., 211, 216–218,
 220, 223, 225–228, 270, 304,
 308, 309, 312, 317, 352, 431,
 e15–e17, e53, e65, e71, e104
 Pant, P.V.K., e48–e50
 Papamakarios, G., 475, 479
 Parisi, G., 457
 Parker, F.R., 63, 66, 76, e76
 Parker, S.C., 218
- Parra, M.I., 277
 Parrill, Abby L., 310
 Parrinello, M., 194, 234, 238, 245, 253,
 256, 257, 259, 268, 270, 304,
 309–311, 493, 506, e89, e91
 Parry, W.E., 30
 Parsonage, N.G., 197, 202
 Pasichnyk, I., 400, 401, 403
 Pasquarello, A., 506
 Pasta, J.G., 4
 Patey, G.N., 227, 323, 329, 436
 Paul, W., 351, 352, 630
 Peczak, P., e54
 Pedersen, L., 389–393
 Pendleton, B.J., 468, e61
 Percus, J.K., 98, 129, 181, 286
 Perera, L., 389, 391–393
 Pereyra, V.D., 306
 Perram, J.W., 373, 374, 387, 499, e42
 Peters, B., 7, 516, 534, 548, 554
 Peters, C.J., 227
 Peters, E.A.J.F., 487, 489
 Petersen, H.G., 145, 165, 172–174, 387,
 392, 499, e80, e84
 Pettitt, B.M., 394, 396, 397
 Pfaendtner, J., 311
 Phillipot, S.R., 393
 Pincus, P.G., 227
 Pippig, M., 371, 389, 393, 398, 401, 403,
 404
 Plattner, N., 460, e107
 Plimpton, S.J., 132, 133
 Pohorille, A., 268
 Poirier, J.C., 280
 Pollock, E.L., 390
 Polson, J.M., 339, 344, e100
 Pomeau, Y., 569
 Pontikis, V., 7, 97, 302, 505, 522, 524
 Poole, Charles P., 37, 245, 494, 573, 575
 Postma, J.P.M., 236, 244, 252, 256
 Potoff, J.J., e54
 Potts, D., 371, 389, 393, 398, 401, 403,
 404
 Poulain, P., 308
 Pound, G.M., 92
 Powles, J.G., 69, 151
 Praestgaard, E., 268
 Pratt, L.R., 535

- Predescu, C., 115–117, 127, 128, 237
 Press, W.H., 7, 44, 61, 140, 173, 389, 424,
 425, 638
 Price, W.S., 148
 Prigogine, I., 324
 Prins, J.A., 3
 Pritzel, A., 475, 479
 Procacci, P., 512, 514, e108
 Progigine, I., 4
 Pronk, S., 339, 344, 350, e100
 Pusey, P.N., 213, 560
- Q**
 Qian, Y.H., 310
 Qin, L., 79, 402, 487, 490, 491
 Quinlan, G.D., 110
 Quirke, N., 202, 203, 217, 220, 227, 228,
 270, e15, e17
- R**
 Racaniere, S., 475, 479
 Radhkrishnan, R., 202
 Rahman, A., 4, 194, 234, 259, 270, 310,
 507
 Ramirez-Hinestrosa, S., 72, 74, 93, 103
 Raos, G., 154
 Rapaport, D.C., 7, 97
 Rathore, N., 309
 Raveche, H.J., 268
 Ray, J.R., 194
 Recht, J.R., e17
 Ree, F.H., 267, 272, 325, 339, e100
 Reggiani, L., 470
 Reguera, D., 553–555
 Reichl, Linda E., 37, 38, 44, 609
 Reimer, J.A., 250
 Reinhardt, W.P., 81, e78
 Reiss, H., 186, 350
 Remler, D.K., 506
 Renardus, M.K.R., 626, 627
 Revenga, M., 566, 567, 621–624
 Rezende, D.J., 475, 479
 Rice, O.K., 3
 Rice, S.A., 176, 324, 352, 354, 413
 Richards, E., 202
 Richardson, C., 6
 Roberts, G.O., 84
 Rodgers, J.M., 473
- Rokhlin, V., 394, 403
 Romano, S., 280
 Rosenbluth, A.W., 4, 56, 57, 63, 351, 352,
 354, 408, 414
 Rosenbluth, M.N., 4, 54, 56, 57, 63, 351,
 352, 354, 408, 414
 Rosengren, A., 269
 Ross, M., 272, 325
 Rotenberg, B., 141, 143, 144
 Rotskoff, G.M., 475, 476, 478, 479
 Rottler, J., 399
 Rovere, M., 193, 225
 Rovigatti, L., 227, 234, 235
 Roweth, D., 468, e61
 Rowley, L.A., 197
 Rowley, R.L., e43
 Rowlinson, J.S., 139, 202, 224, 225, e5
 Royall, C.P., 331
 Rudge, W.E., 630
 Ruelle, David, e6, e7
 Ruiz-Franco, J., 234, 235
 Ruiz-Montero, M.J., 523, 531, 534
 Rull, L.F., 7, 222, 229
 Rull, Luis F., 226, 366
 Rushbrooke, G.S., 227
 Russo, J., 227
 Ruthven, D.M., 151
 Rutkai, G., 46, 78
 Ryckaert, J.P., 7, 90, 91, 341, 498, 511,
 e108
 Ryckaert, Jean-Paul, 226, 366
- S**
 Sabhapandit, S., 467
 Saboungi, M-L., 507
 Saeger, R.B., e54
 Safko, John L., 37, 245, 494, 573, 575
 Sagui, C., 390
 Saha, P., 121
 Sahlin, H.L., 373
 Salsburg, Z.W., 77, e77
 Sampayo, J.G., 138
 Sanchez-Burgos, I., 549
 Sands, Matthew, 573
 Santilli, D.S., 227
 Santiso, Erik E., 137
 Sanz, E., 324, 549
 Schenter, G.K., 557
 Schiller, U.D., 570

- Schilling, T., 329
 Schmid, F., 329
 Schmidt, K.E., 394
 Schmidt, M., 141
 Schnabel, S., 62
 Schneider, T., 268
 Schnell, S.K., 592
 Schofield, P., 137
 Schouten, J.A., 330
 Schrenk, K.J., 468
 Schulten, K., 94
 Schulthess, T.C., 308
 Schultz, A.J., 277, 475
 Schwab, D.J., 6
 Schwarz, Keith, 470
 Sciortino, F., 227
 Scriven, L.E., e71
 Seitz, F., 194, 373, 612, 616
 Sengupta, S., 616
 Sexton, J.C., 118
 Shalloway, D., 115, 116, 127
 Shamshirgar, D.S., 398
 Shaw, D.E., 115–117, 127, 128, 237
 Shell, M.S., 304, 308, 309, 312, 317
 Shelley, J.C., 323, 329, 436
 Sheng, Y.-J., e53
 Shevkunov, S.V., 457, 464
 Shi, Kaihang, 137
 Shing, K.S., 285, 287, 289, e60
 Shirts, M.R., 312
 Sidky, H., 558
 Siepmann, J.I., 129, 286, 352, 354, 362,
 363, 413, 427, 436–438, 444,
 450, e47, e48, e52–e54
 Sikkenk, J.H., e55
 Sindzingre, P., 285, 287, 288, 328
 Sing, K.S.W., 443
 Singer, K., 4, 267, 280, 301
 Singh, R., 129
 Singh, S., 268, 303
 Sivak, D.A., 243
 Skeel, R.D., 110
 Skorodumova, N.V., 269
 Slattery, W.L., 272
 Sliwinska-Bartkowiak, M., 202
 Smallenburg, F., 226, 324
 Smit, B., 69, 70, 98, 143, 144, 202, 203,
 218, 222–224, 227, 229, 232,
 236, 237, 244, 249, 250, 256,
 270, 277, 336, 337, 352, 362,
 363, 413, 423, 436–438,
 443–446, 450, 555, 556, 565,
 e10, e14, e15, e17, e18, e46,
 e47, e52–e54, e57, e63, e64,
 e97, e101, e115
 Smith, A.F.M., 84
 Smith, E.R., 373, 374, e42, e43
 Smith, Edward R., 137
 Smith, G.R., 317
 Smith, W.R., 211
 Snook, I.K., 197
 Snurr, R.Q., 202
 Sorensen, M.R., 557, 558
 Soto, J.L., 202
 Span, R., 46, 78
 Speedy, R.J., 277, 335, 336, 349, 350,
 e96, e97
 Spohr, E., e42, e43
 Sprik, M., 386, 507, 508, 616
 Squire, D.R., 616
 Stafford, K.A., 115–117, 127, 128, 237
 Stapleton, M.R., 217, 220, 227, 228, 270,
 e15, e17, e71
 Stark, H., 569
 Starling, K.E., 277
 Staubach, J., 76
 Stegun, I., 339, e100
 Steinbach, P.J., 371
 Stell, G., 227
 Stephan, S., 76
 Stillinger, F.H., e63
 Stoll, E., 268
 Stratford, K., 570
 Streett, W.B., 76, 78, 268, e76, e77
 Strey, R., 554
 Stringfellow, G.S., 272
 Stroud, H.J.F., 202
 Struzhkin, V.V., 330
 Styer, Daniel F., 537
 Subrahmanyam, S., 396
 Succi, S., 560, 569, 570
 Sugita, Yuji, 457
 Sumpter, B.G., 586
 Suter, U.W., 352, 413, 469, e52, e62
 Sutmann, G., 371, 389, 393, 398, 401,
 403, 404

- Swendsen, R.H., 30, 268, 290, 317, 457, 480
 Swenson, C.J., e53
 Swenson, D.W.H., 535
 Swinton, F.L., 224
 Swope, W.C., 113, 241, 350
 Szleifer, I., 352, 431, e53, e63, e104
- T**
 Takada, J., 268
 Talkner, P., 552
 Tan, R.K.Z., 236, 237, 244
 Tan, Z., e53
 Tartaglino, U., 154
 Tassios, D.P., 227
 Teja, A.S., e53
 Teller, A.H., 4, 56, 57, 63
 Teller, E., 4, 56, 57, 63, 373
 Telo Da Gama, M.M., 92
 Telo da Gama, M.M., 228, e17
 ten Wolde, P.R., 547
 Teukolsky, S.A., 7, 44, 61, 140, 173, 389, 424, 425, 638
 Theodoropoulos, C., 310
 Theodorou, D.N., 91, 202, 362, 363, 432, e48–e50
 Thirumalai, D., 85
 Thol, M., 46, 78
 Thompson, A.P., 132, 133
 Thompson, E.A., 457
 Tildesley, D.J., 6, 7, 40, 60, 76, 78, 88, 97, 130, 137, 138, 217, 220, 227, 228, 270, 358, 374, 393, 401, 403, 469, 484, 506, e5, e14, e15, e17, e61, e76, e77, e122
 Tilwani, P., 324
 Tironi, I.G., e122
 Tiwary, P., 268, 310, 311, 558
 Tobias, D.J., 175, 509, 511, e33–e35, e40, e41, e107, e122
 Tobochnik, J., 7
 Todd, B.D., 134
 Torbrugge, S., 308
 Torda, A.E., 309
 Tornberg, A.K., 398
 Torres-Gavosto, R.M., 143, 144
 Torrie, G.M., 227, 301
 Tosi, M.P., 373
 Toxvaerd, S., 110, 122, 252, 268, e53
 Travis, K.P., 71, 223, 269, e88
 Trebst, S., 311, 467
 Tremaine, S., 110, 121
 Triska, B., 211
 Trizac, E., 339, 344, e100
 Troster, A., 304, 306
 Troyer, M., 311, 467
 Tsai, N.H., 92
 Tsai, S.H., 303
 Tsonopoulos, C., e52, e53
 Tuckerman, M.E., 6, 97, 118, 175, 251, 252, 468, 493, 506, 509, 587, e22, e24, e32–e35, e40, e41, e61, e122
 Tuinier, R., 128
 Turnbull, D., 194, 373, 612, 616
- U**
 Uberuaga, B.P., 557
 Ueda, A., 268, 272, 325
 Ulam, S.M., 4, 281
 Umari, P., 154, 162, 463
- V**
 Valeriani, C., 69, 547
 Valleau, J.P., 7, 197, 227, 301, 302
 Valsson, O., 268, 310, 311
 van Beest, B.W.H., 403
 van Blaaderen, A., 331
 van de Ven-Lucassen, I.M.J.J., e120
 van der Hoef, M.A., 68, 277
 van der Zanden, A.J.J., e120
 van Erp, T.S., 534, 546–548
 Van Gunsteren, W.F., 118
 Van Gunsteren, W.F., 236, 244, 252, 256, 279, 309, e122
 van Hooff, J.H.C., 445
 van Kampen, N.G., 30, 60
 Van Kampen, N.G., 90
 van Kampen, N.G., 238, 244
 Van Kampen, N.G., 499
 van Kampen, N.G., 552, e65
 van Leeuwen, M.E., 227, e54
 van Megen, W., 197
 van Nordstrand, R.A., 227
 van Os, N.M., e63
 van Roij, R., 331
 van Santen, R.A., 403, 445

- van Swol, F., 375
 van Tassel, P.R., 202
 van Well, W.J.M., 445
 Vanden-Eijnden, E., 268, 310, 460, 475, 476, 478, 479, 534, 535, 544, 549, 617, e107
 Vega, C., 324, 332, 549
 Vendruscolo, M., 436
 Venturoli, M., 535
 Vergassola, M., 569, 570
 Verlet, L., 4, 76, 98, 129, 181, 267, 286, 625, e76
 Vermeulen, N.A., 325, 326, 331
 Vesely, F.J., 7, 89, 110
 Vetterling, W.T., 7, 44, 61, 140, 173, 389, 424, 425, 638
 Vineyard, G.-H., 4, 5
 Vinutha, H.A., 319
 Vlaisavljevich, B., 143, 144
 Vlugt, T.J.H., 437, 438, 445, 446, 450, 555, 556, 565, e45–e47, e57, e115, e120
 von Smoluchowski, M., 528, 554
 Vorontsov-Vel'yaminov, P.N., 4, 457, 464
 Voter, A.F., 557, 558
 Voth, G.A., 310, 311
 Vrabec, J., 46, 78
 Vuilleumier, R., 141, 143, 144, 506, 509
- W**
 Wagner, A.J., 570
 Wainwright, T.E., 4, 5, 97, 213, 268, 272, 336, 560, e96, e97
 Walker, Alastair J., 470, 491
 Wallace, D.C., 194, 612, 616
 Walton, J.P.R.B., 202, 203, e5
 Wang, C.H., 6
 Wang, D.D., 558
 Wang, F.G., 303, 304
 Wang, H., 460, e107
 Wang, J.S., 317, 457, 480
 Wang, X.P., 72, 74, 93, 103
 Wang, Z.J., 69
 Warren, P.B., 547, 561–564, 567, e65, e66, e115
 Watanabe, H., 223, e88
 Watanabe, K., 202, 203
 Watts, R.O., 4
- Webb, E.B., e54
 Wedekind, J., 553–555
 Weiner, P.K., 279
 Weingarten, D.H., 118
 Weinstein, S.J., 475
 Weis, J.J., 227
 Wessel, S., 467
 Weston, S.C., 143, 144
 Whitelam, S., 486
 Whitlock, P.A., 7, 61
 Whittington, S.G., 7
 Wick, C.D., 436, e54
 Widmann, A.H., e41, e42
 Widom, B., 139, 225, 283, e5, e17
 Wilding, N.B., 193, 216, 223, 225, 329, 340, 344, 346, 446, 462, 463, e88, e100
 Wilhelmsen, O., 71, 223, 269, e88
 Wilkinson, A.J., 279
 Willemsen, S.M., 565, e115
 Williams, C.P., 227, 277
 Wilson, D.B., 487, 489
 Wilson, K.R., 110, 113, 122, 123, 241
 Wilson, M., 507
 Wilson, M.A., 527, e112
 Winkler, R.G., 560, 568, 569
 Wirnsberger, P., 154, 475, 479
 Witherspoon, V.J., 250
 Wolf, D., 393
 Wolff, U., 482
 Wolthuizen, J.P., 445
 Wolynes, P.G., 432
 Wood, G.B., 202
 Wood, W.W., 1, 5, 54, 63, 66, 76, 77, 181, 184, 213, 272, e76, e77
 Woodcock, L.V., 244, 252, 256, 268, 272, 340, e100
 Wu, D., 324, e63, e64
 Wu, D.T., 554
 Wu, H., 475–479
 Wu, M.G., e48–e50
 Wurtz, D., 467
- X**
 Xu, H.L., 98, 249, 250
- Y**
 Yamakawa, Hiromi, 434

- Yan, Q.L., 304, 308, 309, 312, 461–463,
e20
- Yao, J., 197
- Yeh, I.C., 151, e42, e43
- Yeomans, J.M., 563
- Yin, D.W., 211
- Yip, S., 181, 194
- Yokota, R., 398
- Yonezawa, Y., 393
- York, D., 389, 390, 392
- Yoshida, H., 121
- Yoshii, N., 394
- Ytreberg, F.M., 268
- Yu, T.Q., 460, e107
- Z**
- Zaccarelli, E., 234, 235
- Zanetti, G., 569, 570
- Zantop, A.W., 569
- Zervopoulou, E., 432, e48, e50
- Zhao, X., 396
- Zhou, C.G., 305, 306, 308
- Zinn-Justin, J., 213, 443, 560
- Zollweg, J.A., 76–78, 80, 87, 191, 192,
206, 223, 277, e76–e78, e80,
e86–e88
- Zones, S.I., 227
- Zuckerman, D.M., 268
- Zwanzig, R., 169
- Zykova-Timan, T., 259