

Martin O. Steinhauser

Computational Multiscale Modeling of Fluids and Solids

Theory and Applications

Second Edition



Springer

Computational Multiscale Modeling of Fluids and Solids

Martin O. Steinhauser

Computational Multiscale Modeling of Fluids and Solids

Theory and Applications

Second Edition



Springer

Martin O. Steinhauser
Fraunhofer Institute for High-Speed
Dynamics, Ernst-Mach-Institut EMI
Freiburg
Germany

ISBN 978-3-662-53222-5
DOI 10.1007/978-3-662-53224-9

ISBN 978-3-662-53224-9 (eBook)

Library of Congress Control Number: 2016952512

© Springer-Verlag Berlin Heidelberg 2008, 2017

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, express or implied, with respect to the material contained herein or for any errors or omissions that may have been made.

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer-Verlag GmbH Germany
The registered company address is: Heidelberger Platz 3, 14197 Berlin, Germany

*To Katrin,
Pia
and Sven*

Preface to the 2nd Edition

The first edition of *Computational Multiscale Modeling of Fluids and Solids—Theory and Applications* has been very successful and was well received by the computational science community worldwide, so a 2nd edition has become necessary eight years after the first edition was published. This monograph actually is the first of its kind, summarizing and presenting both the physical and mathematical foundations of the field of computational multiscale modeling, along with key algorithms and source code in a succinct way using a consistent notation. Since publication of the first edition, I have received numerous e-mail messages with positive comments on the book from researchers around the world and personal feedback from readers at many international conferences.

For the second edition, the text and some of the figures have been revised and typographical errors have been corrected. Vectors are now written in boldface notation rather than using arrows for indicating that a quantity has several components in some vector space. The reason for this change is that the arrow symbol can be misleading, as a vector is very generally defined in mathematics and usually refers to a certain algebraic structure of a class of objects that may or may not have anything to do with the notion of a scalar magnitude with direction. Also, at the beginning of the chapters, I have now included short summaries that state briefly the content of the material covered. I hope that this further improves the clarity of presentation of the material. One great advantage of this monograph is that it really focuses on the *fundamental* methods and techniques in multiscale computational science. Hence, this work is still up-to-date and will remain state-of-the-art for many years to come, as it presents the basic methods and algorithms in the field.

I have been teaching computational science courses based on this book for several years at the University of Freiburg in Germany and at the University of Basel in Switzerland, and I hope that this 2nd edition will also find its place on the bookshelves of researchers around the world as an advanced graduate text in the areas of multiscale modeling and computational materials science.

Notifications of typos in this book are highly appreciated and you can do so by reaching me via ResearchGate (<https://www.researchgate.net>) or simply sending an e-mail to martin.steinhauser@unibas.ch.

Freiburg and Basel
July 2016

Martin O. Steinhauser

Preface to the 1st Edition

This volume provides an overview of some of the basics of the underlying physical theories, the mathematical concepts, the numerical tools, and the methods with which the dynamics of condensed matter is treated on several lengths and associated timescales, with a focus on particle-based methods. These methods form the basis of a research area which is generally termed “computational multiscale physics” (CMP) or “computational materials science” (CMS), meaning the modeling and simulation of condensed matter in the fluid or solid state with the aid of computer-implemented efficient algorithms.

The presented methods and theories in this book range from the atomic quantum scale (nanometers), the microscopic scale (from nano- to micrometers), and the intermediate mesoscopic scale (micrometers to centimeters), to the scale of macroscopic structures (centimeters and meters). The order of chapters roughly follows this classification of scales, and each chapter starts with an introductory overview of the physical basics of condensed matter modeling on the respective scale.

Presented topics include quantum mechanical ab initio methods in Hilbert space which are frequently used to predict the electronic and/or atomic structures of different materials, described as many particle systems in statistical thermodynamics. These methods are very restricted in the number of treated particles. Semiempirical techniques such as the tight-binding method and yet more empirical methods such as classical molecular dynamics and Monte Carlo simulations are discussed as well. We present the applications of these methods in equilibrium and non-equilibrium solid-state physics (shock wave physics being one particularly interesting application) as well as in soft matter physics, e.g., polymers. We also discuss classical field theory which is the physical basis of methods such as the finite element method for solving partial differential equations in Sobolev spaces (a special class of Hilbert spaces), usually used in engineering applications of fluid dynamics and with macroscopic structural problems.

With respect to the above said, this book offers primarily orientation in the large field of computational multiscale physics by putting together several numerical methods in a unified fashion with a particular emphasis on explaining the underlying physical theories. These theories may be considered as statements of established “truths” about nature and are used as a general basis for model building,

independent of the application of any particular numerical technique. A good understanding of the physics behind each numerical technique is essential for making useful choices of algorithms. In this respect, this book is necessarily limited as one can neither hope to discuss exhaustively *all* aspects of physical theories that are important for a thorough understanding of certain numerical methods, nor can one hope to discuss *all* aspects of computer simulation in a volume of finite size. However, by focusing on the *fundamental principles* which form the common theoretical basis of many different computational methods, I have attempted to provide a systematic overview which may serve as a sound introduction into several research areas, so that researchers and students can quickly step on from this basic level to more detailed, advanced, and complex problems, but with a thorough understanding of the guiding principles. The selection of examples, illustrating some numerical or theoretical methods, is inevitably biased by the author's own research interests.

Many references to the basic *primary* literature are provided where possible in order to provide the interested reader with several starting points for his or her own research. For those readers, who do not have ready access to primary literature, e.g., through university library loan services, recent references are provided at the end of several chapters, including easily accessible, less technical volumes. Exercises are occasionally included in the main text and problems are provided at the end of some chapters in order to show some tricks of the trade with certain methods or to illustrate further some theoretical considerations. Several example applications for numerical techniques and references to research topics of current interest are provided. Boxes summarize certain topics in succinct form in the main text and can be used as a convenient source of reference.

To make this book yet more valuable for those readers interested in learning how to write their own simulation codes, I have also included several guidelines for designing and implementing programs in Chap. 2 from my own professional experience in software industry. Additionally, more explicit instructions for writing a basic functioning molecular dynamics program are provided in Chap. 6. Thus, not only the theoretical basics and the essential algorithms of several research areas in computational multiscale physics but also some explicit code samples which may be used as a template for writing your own simulation programs are provided.

There is no doubt that computer simulation as an interdisciplinary tool for solving complex scientific problems has become indispensable in modern research both for scientists at universities and for engineers in industrial laboratories. Numerical predictions combine the elements of theories and models of physics, mathematics, chemistry, computer science, professional code development, materials and engineering science, and even biology. Thus, multiscale physics has a strong interdisciplinary character which is why a thorough understanding of the essentials of various scientific fields is a prerequisite for understanding the recent progress achieved in this area of research. The diversity of basic theories and numerical methods is not easy to overview and to understand for students entering the field of computer simulation. Even for scientists active in this research area, it is

not an easy task to survey all recent developments even in their own narrow field of expertise.

The number of scientists and engineers using computer simulation as a general tool is steadily increasing. Non-experts entering this field will probably merely want to use commercial software for solving complex problems, and they will not be primarily interested in techniques or theoretical “dead freight.” Yet, I hope to convince those readers who consider themselves just users of ready-made software packages that it is very interesting to understand some of the common physical and mathematical principles on which these programs are based and, yet more important, that a thorough understanding of the functioning of a simulation code may enhance considerably the efficiency with which such programs are used.

The presented material has emerged from my research experience as a principal investigator and managing project director in scientific research projects both in industrial and in institutional research in the fields of computer science, materials science, polymer physics, and genome research.¹ Some of the material covered in this book was originally presented at several national and international conferences and workshops and is presented here in an extended version. Because of the rapid progress in computational methods and modeling strategies in different areas of condensed matter research today and because of the lively interest in these topics that I experienced among students and colleagues over the past years, it became evident that there has been a demand for a book which provides a unified overview of many important concepts and which focuses on the underlying basic theories in the field of CMS.

The fundamental basics in most areas of computational modeling will not change in the near future. Thus, working in the field of computational physics, it is important to realize that mostly those things are important to be learned and understood which are invariant against general proclivities, software package versions, or computer languages. In this sense, this book also offers a basic introduction into physical and numerical model building in a context, in which these topics are probably not conveyed in standard university courses.

This book addresses to advanced students, primarily on the graduate or advanced undergraduate level who want to enter the field of computer simulations. It may also be helpful for researchers and engineers in physics, chemistry, polymer science, and materials science who are active in computer simulation in industrial laboratories and in universities, and who are interested in a presentation of the fundamental mathematical and physical principles underlying the different numerical methods that are used in materials science on different lengths and timescales.

The level of expertise that is presupposed of the reader in order to be able to follow the text is first of all a certain amount of scientific maturity, with the precise direction not being overly important. Some physical background knowledge at least in classical mechanics and some exposure to a minimum amount of the modern

¹See, e.g., <http://www.chemie.unibas.ch/~steinhauser/>.

abstract mathematical notation of set theory and functions on sets are helpful but not essential. This notation is introduced and explained in Part I of this book, particularly in Chap. 3. The same holds for the theory and notation of tensors and manifolds which are important mathematical tools for a succinct and precise formulation of physical theories as found in advanced treatises on theoretical physics.

A word about the organization of this book is in order. In Part I, many important mathematical and physical basics that are needed in the formulation of theories are summarized. Also, some important facts about computer science and a general overview of length scales in the natural sciences are included.

Chapters 1 and 2 are introductory in style. The current status of computer applications in the natural sciences as well as modeling strategies and the development of a notion of “algorithm” and “computability” are discussed from a historical perspective.

In Chap. 3, the most important mathematical and physical basics for applying different theories and models in computational materials science are introduced. The choice of topics covered here is biased by the author’s own interests. It is hoped that this chapter provides a sound mathematical background of many abstract concepts prevalently used in physical theories.

Finally, Chap. 4 provides a basic introduction into numerical simulation techniques, from the key algorithms for integrating partial or ordinary differential equations on a computer, to some software engineering techniques that are helpful when writing your own simulation programs.

The chapters in Part II follow the length scales from bottom-up starting with a presentation of methods on the atomic scale in Chap. 5 taking into account the electronic degrees of freedom. The most basic methods are ab initio self-consistent Hartree–Fock methods, which are very common in quantum chemistry for obtaining the ground-state energy and geometry of small molecules. Starting from an initial guess of the basis wave function, the Schrödinger equation is reduced to a one-particle wave function for one electron in the averaged (mean) potential of all other electrons for which the initial wave functions are used. Therefore, this method is also called mean-field approach. Inserting the results repeatedly into the eigenvalue problem of the Schrödinger equation, it is solved self-consistently until convergence of the solution occurs. A good guess for the initial form of the basis wave functions is provided by using the Slater determinant which correctly includes the symmetry property of the wave function.

It is important to understand that even the so-called ab initio methods rely on some fundamental approximations. The most important one is the adiabatic Born–Oppenheimer approximation which assumes that the heavy atomic nuclei are in essence stationary compared to the fast-moving light electrons, and thus, the wave function can be separated into a product of a core wave function and an electron wave function which can be treated separately. Density Functional Theory (DFT) is discussed as well as half-empirical methods (tight-binding and Car–Parrinello) and empirical methods which use coarse-grained potentials.

Chapter 6 provides an overview of coarse-grained methods on the microscopic scale, mostly covering the molecular dynamics and Monte Carlo methods.

Instructions of how to write and organize a basic molecular dynamics program are included. Examples from polymer physics and solid-state physics illustrate these two methods which are abundant on the microscopic length scale.

Chapter 7 treats some methods abundant on the meso- and macroscale and provides some examples. The meso- and macroscale is the scale where fundamental research—usually based on the particle concept—is not dominating the field of materials research any more. Instead, on this scale, one enters the area of engineering applications, making ingenious use of fluid dynamics concepts and relying fundamentally on the continuum approximation, which works excellent for many practical purposes. The basics of dislocation dynamics, elasticity theory, and continuum fluid theory, as well as Monte Carlo methods and geometric models with applications in solid-state physics are presented. The finite element method, along with smoothed particle hydrodynamics, is well established for applications on this scale, and both methods have been incorporated in many commercial program packages, which allow for simulating the materials response to external loading, making use of a large variety of material constitutive models. The meso-/macroscale is a length scale with probably the largest variety of structural properties of solid and liquid systems, due to an interplay between energetic effects (which dominate on small scales) and entropic effects, which come into play by the many different conformations that materials can acquire on intermediate and large length scales.

Finally, Chap. 8 provides a short overview of the most pressing problems that might be worthy being addressed with multiscale methods in the near future.

It is my pleasant duty to acknowledge the encouragement, interest, and help that I had from colleagues and students when writing this book. In this respect, a special vote of thanks is due to my former colleague Dr. Martin Kühn with whom I had endless and very fruitful discussions on the foundations of some research areas in physics and mathematics while working on the manuscript during the year 2006. He also helped improving parts of the manuscript by very useful suggestions.

I am particularly indebted to Prof. Dr. Alexander Blumen of the University of Freiburg, who has always provided support and encouragement in a very friendly and helpful manner for all aspects of our joint and my own research in Freiburg.

I also owe thanks to Prof. Dr. Klaus Thoma, director of the Fraunhofer Ernst-Mach-Institute for High-Speed Dynamics (EMI) in Freiburg, for his very kind continuous support of my independent research on the application of multiscale methods in shock wave, solid-state, and polymer physics. Along with him, I had the honor and pleasure to lead and manage a large nationwide project (“MMM-Tools: Methods for Multiscale Materials Modeling”) during the years 2003–2005 (with a 3.5 Mio Euro budget), devoted explicitly to the development of multiscale methods and efficient software applications in materials research.

Julian Schneider helped me with preparing some data and figures on polymer simulations in Chap. 6 and also read parts of the manuscript, while Kai Grass helped developing a mesoscopic particle model for the description of brittle failure in materials upon shock loading which is discussed in Chap. 7.

Last but not least, I wish to thank my family for their love, enduring patience, and great understanding while I was working on the manuscript. It was our precious time that I had to steal in order to finish this book.

Freiburg, Germany
July 2007

Martin O. Steinhauser

Contents

Part I Fundamentals

1	Introduction	3
1.1	Physics on Different Length- and Timescales	5
1.1.1	Electronic/Atomic Scale	6
1.1.2	Atomic/Microscopic Scale	7
1.1.3	Microscopic/Mesoscopic Scale	7
1.1.4	Mesoscopic/Macroscopic Scale	10
1.2	What are Fluids and Solids?	11
1.3	The Objective of Experimental and Theoretical Physics	13
1.4	Computer Simulations – A Review	14
1.4.1	A Brief History of Computer Simulation	17
1.4.2	Computational Materials Science	25
1.5	Suggested Reading	27
2	Multiscale Computational Materials Science	29
2.1	Some Terminology	32
2.2	What Is Computational Material Science on Multiscales?	33
2.2.1	Experimental Investigations on Different Length Scales	34
2.3	What Is a Model?	37
2.3.1	The Scientific Method	38
2.4	Hierarchical Modeling Concepts Above the Atomic Scale	45
2.4.1	Example: Principle Model Hierarchies in Classical Mechanics	47
2.4.2	Structure-Property Paradigm	48
2.4.3	Physical and Mathematical Modeling	48
2.4.4	Numerical Modeling and Simulation	56
2.5	Unifications and Reductionism in Physical Theories	57
2.5.1	The Four Fundamental Interactions	59
2.5.2	The Standard Model	61
2.5.3	Symmetries, Fields, Particles and the Vacuum	63
2.5.4	Relativistic Wave Equations	70
2.5.5	Suggested Reading	77

2.6	Computer Science, Algorithms, Computability and Turing Machines	78
2.6.1	Recursion.	81
2.6.2	Divide-and-Conquer.	83
2.6.3	Local Search	86
2.6.4	Simulated Annealing and Stochastic Algorithms	88
2.6.5	Computability, Decidability and Turing Machines	89
2.6.6	Efficiency of Algorithms	99
2.6.7	Suggested Reading	106
	Problems	107
3	Mathematical and Physical Prerequisites	109
3.1	Introduction	110
3.2	Sets and Set Operations	113
3.2.1	Cartesian Product, Product Set.	117
3.2.2	Functions and Linear Spaces	118
3.3	Topological Spaces	126
3.3.1	Charts	133
3.3.2	Atlas	134
3.3.3	Manifolds	136
3.3.4	Tangent Vectors and Tangent Space	138
3.3.5	Covectors, Cotangent Space and One-Forms	141
3.3.6	Dual Spaces.	146
3.3.7	Tensors and Tensor Spaces	148
3.3.8	Affine Connections and Covariant Derivative	153
3.4	Metric Spaces and Metric Connection.	156
3.5	Riemannian Manifolds	159
3.5.1	Riemannian Curvature	160
3.6	The Problem of Inertia and Motion: Coordinate Systems in Physics	162
3.6.1	The Special and General Principle of Relativity	163
3.6.2	The Structure of Spacetime	167
3.7	Relativistic Field Equations.	168
3.7.1	Relativistic Hydrodynamics	169
3.8	Suggested Reading	171
	Problems	171
4	Fundamentals of Numerical Simulation	175
4.1	Basics of Ordinary and Partial Differential Equations in Physics	176
4.1.1	Elliptic Type	180
4.1.2	Parabolic Type.	182
4.1.3	Hyperbolic Type	183
4.2	Numerical Solution of Differential Equations	185

4.2.1	Mesh-Based and Mesh-Free Methods	186
4.2.2	Finite Difference Methods	191
4.2.3	Finite Volume Method.	194
4.2.4	Finite Element Methods.	197
4.3	Elements of Software Design	199
4.3.1	Software Design	202
4.3.2	Writing a Routine	205
4.3.3	Code-Tuning Strategies	208
4.3.4	Suggested Reading	211
	Problems	212

Part II Computational Methods on Multiscales

5	Computational Methods on Electronic/Atomistic Scale	215
5.1	Introduction	215
5.2	Ab-Initio Methods.	217
5.3	Physical Foundations of Quantum Theory	220
5.3.1	A Short Historical Account of Quantum Theory	221
5.3.2	A Hamiltonian for a Condensed Matter System.	224
5.3.3	The Born–Oppenheimer Approximation	224
5.4	Density Functional Theory	227
5.5	Car–Parinello Molecular Dynamics.	228
5.5.1	Force Calculations: The Hellmann–Feynman Theorem	230
5.5.2	Calculating the Ground State	231
5.6	Solving Schrödinger’s Equation for Many-Particle Systems: Quantum Mechanics of Identical Particles	233
5.6.1	The Hartree–Fock Approximation	234
5.7	What Holds a Solid Together?	245
5.7.1	Homonuclear Diatomic Molecules.	245
5.8	Semi-empirical Methods	247
5.8.1	Tight-Binding Method	249
5.9	Bridging Scales: Quantum Mechanics (QM) - Molecular Mechanics (MM)	253
5.10	Concluding Remarks.	254
6	Computational Methods on Atomistic/Microscopic Scale	257
6.1	Introduction	257
6.1.1	Thermodynamics and Statistical Ensembles	261
6.2	Fundamentals of Statistical Physics and Thermodynamics	263
6.2.1	Probabilities	263
6.2.2	Measurements and the Ergodic Hypotheses	264

6.2.3	Statistics in Phase Space and Statistical Ensembles	266
6.2.4	Virtual Ensembles	270
6.2.5	Entropy and Temperature	270
6.3	Classical Interatomic and Intermolecular Potentials.	272
6.3.1	Charged Systems	272
6.3.2	Ewald Summation	274
6.3.3	The P^3M Algorithm.	275
6.3.4	Van der Waals Potential	276
6.3.5	Covalent Bonds	276
6.3.6	Embedded Atom Potentials	277
6.3.7	Pair Potentials	279
6.4	Classical Molecular Dynamics Simulations.	281
6.4.1	Numerical Ingredients of MD Simulations.	281
6.4.2	Integrating the Equations of Motion	287
6.4.3	Periodic Boundary Conditions	291
6.4.4	The Minimum Image Convention	292
6.4.5	Efficient Search Strategies for Interacting Particles	292
6.4.6	Making Measurements.	295
6.5	Liquids, Soft Matter and Polymers	300
6.5.1	Scaling and Universality of Polymers	303
6.6	Monte Carlo Simulations	310
	Problems	312
7	Computational Methods on Mesoscopic/Macroscopic Scale	313
7.1	Example: Meso- and Macroscale Shock-Wave Experiments with ceramics	317
7.2	Statistical Methods: Voronoi Tesselations and Power Diagrams for Modeling Microstructures of Ceramics	319
7.2.1	Reverse Monte Carlo Optimization	321
7.3	Dissipative Particle Dynamics.	325
7.4	Ginzburg–Landau/Cahn–Hilliard Field Theoretic Mesoscale Simulation Method	327
7.5	Bridging Scales: Soft Particle Discrete Elements for Shock Wave Applications	328
7.6	Bridging Scales: Energetic Links between MD and FEM.	338
7.6.1	Bridging Scales: Work-Hardening	340
7.7	Physical Theories for Macroscopic Phenomena: The Continuum Approach.	341
7.7.1	The Description of Fluid Motion	342
7.8	Continuum Theory	343
7.8.1	The Continuum Hypothesis	344
7.9	Theory of Elasticity	346
7.9.1	Kinematic Equations	349

7.9.2	The Stress Tensor	354
7.9.3	Equations of Motion of the Theory of Elasticity	356
7.9.4	Constitutive Equations	356
7.10	Bridging Scale Application: Crack Propagation in a Brittle Specimen	359
8	Perspectives in Multiscale Materials Modeling	361
Appendix A: Further Reading	365	
Appendix B: Mathematical Definitions	367	
Appendix C: Sample Code for the Main Routine of a MD Simulation	369	
Appendix D: A Sample Makefile	371	
Appendix E: Tables of Physical Constants	375	
References	383	
Index	403	

List of Algorithms

Algorithm 2.1	The sieve of Eratosthenes	80
Algorithm 2.2	The largest common divisor (Euclid's algorithm)	80
Algorithm 2.3	Rest-recursive function	82
Algorithm 2.4	Recursive calculation of a power	84
Algorithm 2.5	Recursive calculation of the Fibonacci numbers	85
Algorithm 2.6	Simulated annealing, Metropolis Algorithm (Version I)	89
Algorithm 2.7	A sort algorithm	100
Algorithm 4.1	The velocity Verlet algorithm	191
Algorithm 4.2	A code example of badly written PDL	205
Algorithm 4.3	A code example of improved PDL	206
Algorithm 4.4	Example of a straightforward piece of code for the summation of matrix elements	210
Algorithm 4.5	Example of an optimization strategy for the summation of matrix elements using pointers	210
Algorithm 5.1	Float chart of the conjugate gradient algorithm	227
Algorithm 5.2	The Gram–Schmidt Algorithm	241
Algorithm 5.3	Flow chart of the Hartree–Fock algorithm	242
Algorithm 5.4	Hartree–Fock algorithm	243
Algorithm 6.1	P^3M -Algorithm to calculate the reciprocal sum	275
Algorithm 6.2	Principal flow diagram of a MD simulation	282
Algorithm 6.3	A general MD simulation scheme	283
Algorithm 6.4	Code fragment showing a convenient organization of data within C-structures	285
Algorithm 6.5	Rescaling the velocity of particles to a predefined temperature T in a microcanonical ensemble	286
Algorithm 6.6	Code fragment of the all-particle search	288
Algorithm 6.7	Code fragment of a brute force $\mathcal{O}(N^2)$ LJ force calculation	290
Algorithm 6.8	Metropolis Algorithm (Version II)	311

List of Boxes

Box 1.1	The von Neumann computer model	18
Box 2.1	Definition of a group, Abelian group	66
Box 2.2	$SU(2)$ Symmetry.	68
Box 2.3	$SU(3)$ Symmetry.	69
Box 2.4	Lagrangian formulation of classical mechanics	71
Box 2.5	Sequence alignment to a fraction of human α - and β -globin.	86
Box 3.1	The definition of a ring and a field.	121
Box 3.2	The definition of a linear vector space \mathfrak{L}	121
Box 3.3	The definitions of a normed vector space and a Banach space	124
Box 3.4	Some frequently used function spaces	125
Box 3.5	The scalar product in a vector space	125
Box 3.6	Definition of an inner product	146
Box 3.7	Dual vector spaces	147
Box 4.1	Lipschitz condition	177

Part I

Fundamentals

Subtle is the Lord but he is not malicious

A. Einstein

Abstract

In this introduction we are first going to discuss the different length and time scales on which we seek a description of nature. Then, in Sect. 1.2 we learn about the major difference between fluids and solids. In Sect. 1.3 we talk about the relation between theory, experiment and simulation in physics. Section 1.4 provides a review of the history of computer simulations, super computing and the development of computer hardware and software. Finally, Sect. 1.5 closes this first chapter with many suggestions for useful further reading.

This book deals with two aspects of modern physics and engineering science: the theoretical description of the behavior of matter on different length and time scales and the implementation of efficient algorithms on computers to solve the various differential equations arising from these physical theories. In doing so we focus on the fluid and solid states of matter, albeit the numerical methods are usually neither restricted to any particular phase of matter nor to any particular intrinsic length scale.

With his main opus “Philosophiae Principia Naturalis Mathematica”, Sir Isaac Newton created the foundation of what today is called “classical physics” (meaning physics before the advent of quantum theory in 1925/1926 by Heisenberg and Schrödinger). Newton formulated universal laws of motion for mechanical systems, i.e. for solid bodies, and he also formulated a theory of gravitation which explains the universal attraction of bodies. Newton published his writings in Latin as was customary at the time – hence, the original of the *Principia* is very difficult to read, as it is not only written in latin, but in latin that was common at the time. However, one can find translations of the original in almost all languages, see e.g. [1]. In order to

describe the motion of bodies efficiently, Newton invented his method of “fluxions”, which allowed for a graphical representation of moving bodies in terms of a curved line, composed of tangents placed next to each other.

The German mathematician Willhelm Gottfried Leibniz – independently from Newton – also invented calculus as a method to tackle the description of the motion of bodies, based on the notion of a curve (representing e.g. velocity) as being composed of an infinite number of points. Both mathematicians obtained the same results, but because the writing style of Leibnitz was much clearer than Newton’s and because he published his results before Newton, his symbolic notation of calculus has prevailed.¹ Both of them fought a fierce controversy over the invention of calculus in the later parts of their lives.

Since the discoveries of Newton and Leibnitz, the modeling of the world is done with the methods of calculus, i.e. differential equations that include time and space as independent variables. The invention of calculus as a means to mathematically tackle the infinitesimal change of the movement of a body along a path has been overwhelmingly successful during the course of several centuries and it still is *the* standard method of model building in (classical) physics and engineering. Calculus was instrumental in the discovery of the basic laws of classical particle, continuum and statistical mechanics, electromagnetism, quantum theory, quantum field theory and the general theory of relativity (GTR).

Although classical physics since Newton’s *Principia* – i.e. physics before the advent of quantum theory in the year 1900² – has vastly changed and has been adopted to new experimental evidence, not as much has changed in the mathematical framework in which physical laws and ideas are formulated, namely in the form of differential equations in continuous four-dimensional space-time. Solving differential equations can be very elaborate and almost always, except for the simplest of circumstances (usually triggered by symmetries), exact analytical solutions are rare. Much of theoretical science today results from a *linear* analysis of the respective differential equations for various disciplines. Despite the fact that many natural phenomena are essentially *non-linear*, the advances in solving non-linear equations have been very modest. In analytic theory one is compelled to introduce heuristic approximations. In order to evaluate these approximations in theory, one makes comparisons with experiments. The problem that arises here is that the analytic models are usually no “ab initio” descriptions, based solely on fundamental equations, but they contain unknown parameters which have to be “fitted” to the experiment. Therefore, a comparison between theory and experiment is often questionable, as errors of approximations in theories could be canceled by inadequately determined parameters. In this case, computer simulations can be used for providing the missing links between theory and experiment. The prime example for this are the models of

¹For example the stylized letter “s” used as the integral sign representing a summation.

²The very first mentioning of the quantization of energy took place in a lecture by Max Planck on 14 December 1900 in Berlin [2]. Max Planck originally used the term “Energieelement” (energy element) and with this assumption he could explain his law of black body radiation, published in a previous lecture [3] on 19 October 1900. A detailed description of the events that led Planck to the introduction of his quantum of action h can be found in [4].

statistical thermodynamics where many particle systems are described as classical systems with interactions between the atoms or molecules. Like a Maxwellian demon, one can intervene in computer simulations of many particle systems and change single properties of individual molecules or atoms in a model. In this way one can evaluate the quality of an analytic approximation based on the same model that is implemented in the simulation.

1.1 Physics on Different Length- and Timescales

Today, fast parallelized computer systems allow for solving complex, non-linear many body problems directly, not involving any preceding mathematical approximations, except for the unavoidable numerical errors occurring when re-formulating differential equations in discretized form for the treatment on a computer. Computer simulations are not only a connecting link between analytic theory and experiment, allowing to test theories, but they can be also be used as an exploratory tool in “computer experiments” under conditions which would not be feasible, too expensive, or too dangerous in real experiments in the laboratory.

This is true for physical problems on very different length- and time scales, be it the simulation of super novae explosions in astrophysics, high-speed impact experiments under extreme pressure conditions or quantum fluctuations in hadron³ physics. Unfortunately, there is no simple “hierarchy” that is connected with a length scale according to which the great diversity of simulation methods could be sorted out (see discussion in Chap. 2). For example, Monte Carlo lattice methods can be applied at the femtoscale of Quantum chromodynamics⁴ [5] (10^{-15} m), at the Ångstrømscale [6] (10^{-10} m) of solid state crystal lattices, or at the micrometer scale [7] (10^{-6} m) when simulating grain growth processes of polycrystal solid states. Thus, with computer simulations, it is important to establish first, which phenomena and properties one is primarily interested in. Many different properties are governed by structural hierarchies and processes on very different length- and associated timescales. An efficient modeling of the system under investigation therefore requires special simulation techniques which are adopted to the respective relevant length- and timescales, cf. Fig. 1.1. In the following, a short overview of the hierarchy of length scales common in materials science is provided and summarized in Table 1.1. This division is primarily based on a spatial, rather than a strict physical classification.

³Hadrons are elementary particles which are subject to the strong interaction, cf. Sect. 2.5.

⁴The theory of the strong force between quarks, the constituents of hadrons.

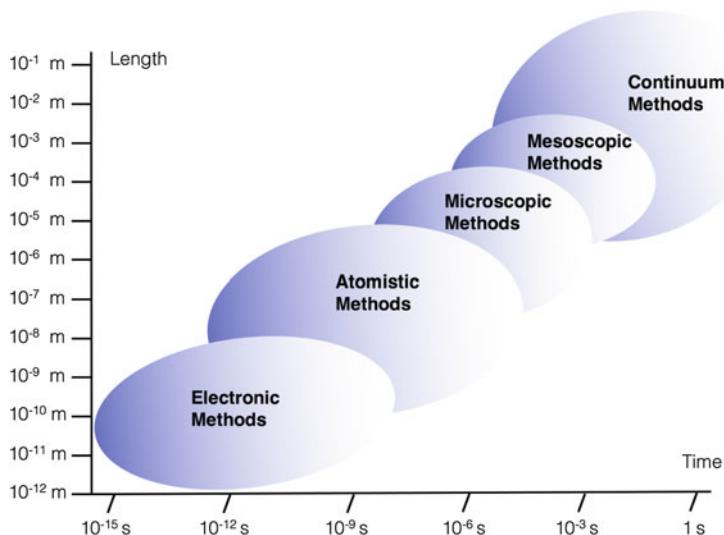


Fig. 1.1 Computational methods and their relevant length and time scales in computational materials science

1.1.1 Electronic/Atomic Scale

The subatomic electronic structure of a material yields information on molecular geometry, magnetic properties, nuclear magnetic resonance (NMR), infrared (IR) or ultraviolet (UV) spectroscopic data, quantum mechanical ground and excited states and on the chemistry of materials.⁵ A modeling on this scale needs to take into account the degrees of freedom of the electrons explicitly. Some basic simulation methods, so called *ab initio* methods (discussed in Chap. 5) have been developed which solve the many particle Schrödinger equation approximately, usually based on the Born–Oppenheimer approximation (BOA). With *ab initio* methods, the only information that has to be provided are the number of atoms and the positions of the atoms within the system. In contrast, *semi-empirical* or *empirical* approaches require a model of the interactions between the atoms to be supplied. There are many well-known software packages used in materials science and quantum chemistry that are available to academic and industrial users directly and free of cost (e.g. ACESII, AMPAC, CPMD, GAMESS, QUANTUM ESPRESSO, SIESTA, VASP) or through commercial vendors (e.g. CASTEP, GAUSSIAN, Molpro). Most of these codes are very old (in parts from the 1970s) and thus can be considered to be ancient legacy FORTRAN code. Some of them are based on density functional theory DFT (discussed in Chap. 5) and were developed by different scientific teams during the past 40–50 years. The results of quantum mechanical calculations are often used

⁵Chemical reactions only involve the (valence) electrons and not the atomic nuclei.

in the design of *classical* molecular force fields (discussed in Chap. 6), providing a connection to the next scale.

1.1.2 Atomic/Microscopic Scale

Simulations performed at the atomic or microscopic scale of molecules are much more diverse than those typical of quantum chemistry and a wide range of properties from thermodynamic to bulk transport properties of solids and fluids can be calculated. As a result of this diversity, researchers in a broad array of disciplines (e.g. physics, chemistry, chemical engineering, molecular biology, biochemistry, nanoscience or even geochemistry) contribute to the development and enhancement of methods on this length scale with typical associated time scales ranging roughly from 10^{-12} to 10^{-6} s for the longest runs on the largest supercomputers. For computer simulations using semi-empirical or classical force fields, there are several academic software packages freely available (e.g. Chemistry at Harvard Macromolecular Mechanics (CHARMM), DL_POLY, GROMACS, NAMD, IMD, XMD) or by prior registration and free licenses (e.g. Groningen Molecular Simulation (GROMOS)). Systems considered on the microscopic scale are still mainly determined in their behavior by their energy, albeit the motion of the electrons can be neglected. Thus, individual atoms or clusters of atoms can be described with methods based on *classical* interaction potentials. The two oldest used methods are Molecular Dynamics (MD) and Monte Carlo (MC) methods based on classical force fields. Additional interaction potentials for modeling covalent bonds, Coulomb interactions, torsion and bending in molecules are only *effective* interactions on this scale, as the quantum mechanical nature of the particle interactions and the electronic contributions are neglected. Due to its simplicity and numerical efficiency, the Lennard-Jones (LJ) potential even today is an often used generic model potential.

1.1.3 Microscopic/Mesoscopic Scale

Many real material systems have structures much larger than can be studied based on the atomic/microscopic scale. For example, the properties of block-copolymer materials are strongly influenced by the molecular segregation into mesoscale domains with typical time scales raging from 10^{-8} s to 10^{-4} s. This is the typical domain of *soft matter systems* or *complex fluids*, e.g. polymers, amphiphiles or colloidal systems. It is the scale on which self-organization of matter in biological systems, e.g. cells or membranes, occurs. These systems are driven by an interplay between their energy and entropy, as they can attain many configurations and packing densities. In solid states, dislocation dynamics, crystallizations and phase transformations typically occur on this scale and in polycrystalline systems nucleation and grain growth play a fundamental role. *Particle-based methods* on this length and time scale include many variations of MD and MC methods using *effective* interaction potentials (discussed in Chap. 6), *dissipative particle dynamics* (DPD) or the *discrete element*

Table 1.1 Common classification of length scales in materials science. Listed are also typical scopes of different simulation methods and some typical applications

Scale (m)	Typical simulation methods	Typical applications
Electronic/Atomic		
$\sim 10^{-12} - 10^{-9}$	Self-consistent Hartree-Fock (SC-HF)	Ground states
$\sim 10^{-12} - 10^{-9}$	Self-consistent DFT (SC-DFT)	NMR, IR and UV spectra
$\sim 10^{-12} - 10^{-9}$	Car-Parinello (ab initio) molecular dynamics (CPMD)	Molecular geometry, entropy, enthalpy
$\sim 10^{-12} - 10^{-9}$	Tight-binding (TB)	Electronic properties
$\sim 10^{-12} - 10^{-9}$	Quantum Monte Carlo (QMC)	Chemical reactions, tribology, dipole moments
Atomic/Microscopic		
$\sim 10^{-9} - 10^{-6}$	Molecular dynamics (MD) and Monte Carlo (MC) using <i>classical</i> force fields	Equations of state Ising model, Bulk transport properties
$\sim 10^{-9} - 10^{-6}$	Hybrid MD/MC	Viscosity
$\sim 10^{-9} - 10^{-6}$	Embedded atom method (EAM)	Rheology of fluids
$\sim 10^{-9} - 10^{-6}$	Particle in cell (PIC)	Phase equilibrium, polymers, DNA models, biomaterials, charged materials

(continued)

Table 1.1 (continued)

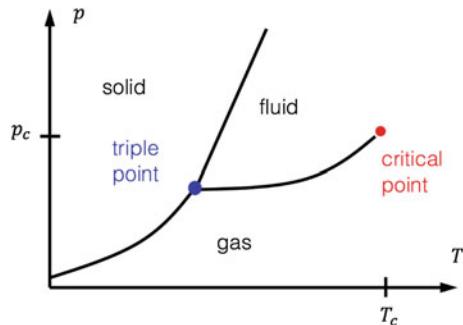
Scale (m)	Typical simulation methods	Typical applications
Microscopic/Mesoscopic		
$\sim 10^{-8} - 10^{-1}$	MD and MC using <i>effective</i> force fields	Complex fluids soft matter
$\sim 10^{-9} - 10^{-3}$	Dissipative particle dynamics (DPD)	Granular matter
$\sim 10^{-9} - 10^{-3}$	Phase field models	Fracture mechanics
$\sim 10^{-9} - 10^{-3}$	Cellular automata (CA)	Grain growth
$\sim 10^{-9} - 10^{-4}$	Mean Field Theory (MFT)	Phase transformations
$\sim 10^{-6} - 10^2$	Finite element methods FEMs including microstructure	Polycrystal elasticity, polycrystal plasticity
$\sim 10^{-6} - 10^2$	Smoothed particle hydrodynamics SPH	Diffusion, Fracture and Failure
$\sim 10^{-9} - 10^{-4}$	Lattice-Boltzmann method (LBM)	Interface motion
$\sim 10^{-9} - 10^{-4}$	Dislocation dynamics	Dislocations
$\sim 10^{-6} - 10^0$	Discrete element method (DEM)	Grain boundaries
Mesoscopic/Macroscopic		
$\sim 10^{-3} - 10^2$	Hydrodynamics	Macroscopic flow
$\sim 10^{-3} - 10^2$	Computational fluid dynamics CFD	Macroscopic elasticity
$\sim 10^{-6} - 10^2$	Finite element methods (FEM)	Macroscopic plasticity
$\sim 10^{-6} - 10^2$	Smoothed particle hydrodynamics (SPH)	Fracture mechanics
$\sim 10^{-6} - 10^2$	Finite difference methods (FDM)	Aging of materials
$\sim 10^{-6} - 10^0$	Cluster & Percolation models	

method (DEM) (discussed in Chap. 7). With these methods the individual particles do not represent “elementary” particles, i.e. atoms, but complete *clusters* of atoms or molecules that are treated as classical particles. The “bead-spring model” (discussed in Chap. 6), connecting the particles by flexible, entropic springs is widely used in polymer physics on this scale. The lattice Boltzmann method (LBM) is a simulation technique which solves the Navier–Stokes equation of fluid flow on a lattice, and which considers a typical volume element of fluid to be composed of a collection of particles that are represented by a particle velocity distribution function for each fluid component at each grid point, i.e. it is a hybrid particle/mesh method. *Cellular automata* are discrete – that is lattice-based – dynamical systems that are typically used as sampling schemes for nucleation and crystallization simulations in engineering applications. *Phase field models*, e.g. of Ginzburg–Landau type, are used to calculate diffusion and phase transitions on this scale. The mesoscopic scale is also the regime where methods, based on continuum theory are used. For engineering applications, mesh-based methods, e.g. FEM, (discussed in Chap. 7) are used almost exclusively for the calculation of fluid flows, solid mechanics and coupled fluid/solid systems. These methods are also used for classical fracture mechanics, albeit particle-based methods in recent years have been proved to be as viable as mesh-based methods in this respect, see e.g. [8, 9, 10, 11]. Another method often utilized on the mesoscale is mean field theory (MFT) which is a continuum method based on the approximation of an effective one-particle Hamiltonian describing the interactions of many particles in a system. A continuum method that is based on the conservation equations of continuum theory but which avoids many mesh distortion problems of FEM approaches, is the method of *smoothed particle hydrodynamics* (SPH) (cf. the discussion in Chap. 7).

1.1.4 Mesoscopic/Macroscopic Scale

When averaging over many degrees of freedom one finally arrives at the phenomenological macroscopic scale. Here, continuum models are used which describe the phenomenological (visco-)elastic behavior of solids and the properties of fluids based on the Navier–Stokes equation. Mostly, mesh-based methods such as FEM, and other procedures of CFD for the solution of partial differential equations are used, which are typically closely connected to applications in engineering sciences. In technical applications, one usually aims at a direct connection to macroscopically measured parameters without introducing any structural microscopic quantities. Many commercial codes that are based on methods of continuum theory are available for structural and coupled fluid-structural simulations in engineering applications. However, one has to bear in mind that the general applicability, validity and transferability of the underlying material models implemented in such codes, is usually rather limited.

Fig. 1.2 Typical phase diagram of condensed matter. In addition to these “traditional” three phases of matter there are also the plasma state (where the electrons are separated from the atoms, i.e. when the atoms are ionized) and the Bose–Einstein condensate



1.2 What are Fluids and Solids?

From thermodynamics, the phase diagram for the description of condensed matter is well known, cf. Fig. 1.2. At low temperatures and high pressures, the atomic constituents form a solid state which is usually crystalline, i.e. the atoms or molecules are arranged in an orderly fashion in a three-dimensional lattice. Thus, a fundamental **characteristic of solid states** is the presence of **long-range (periodic) order**. The periodic arrangement of atoms or ions in a perfect crystal allows for several simplifying assumptions in a theoretical description. The potential that describes the atomic interactions

$$U(\mathbf{r}) = U(\mathbf{r} + \mathbf{R}) \quad (1.1)$$

in this case has the periodicity of the underlying *Bravais lattice* defined by all points with position vectors \mathbf{R} of the form $\mathbf{R} = n_1\mathbf{a}_1 + n_2\mathbf{a}_2 + n_3\mathbf{a}_3$, where the three *primitive vectors* $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ are any three vectors not all in the same plane, and the $n_i, i = 1, 2, 3$ are integers. The problem of describing the dynamics of electrons in a solid is in principle a many-electron problem, as the full Hamiltonian of the solid contains not only the one-electron potentials describing the interactions of the electrons with the massive atomic nuclei, but also pair potentials describing the electron–electron interaction.

As a consequence of the periodicity of the potential U , one can make use of *Bloch's theorem*, see e.g. [12, 13] which states that the eigenstates ψ_n of the one-electron Hamiltonian $H = -\hbar^2\nabla^2/2m + U(\mathbf{r})$ for all \mathbf{R} in a Bravais lattice can be chosen as plane waves such that with each ψ there is an associated wave vector \mathbf{k} for which the equation

$$\psi_{n\mathbf{k}}(\mathbf{r} + \mathbf{R}) = \exp(i\mathbf{k}\cdot\mathbf{R}) \psi_{n\mathbf{k}}(\mathbf{r}) \quad (1.2)$$

is valid for every position vector \mathbf{R} in the Bravais lattice. Of course, perfect crystal periodicity is an idealization. Real solids are never absolutely pure, and in the neighborhood of impurities, the solid is not the same as elsewhere in the crystal. Imperfections in crystals are actually of great importance for many properties of crystals, e.g. electrical conductivity of metals, and in a theoretical description one usually treats the effects of imperfections as small perturbations of the properties of a hypothetical, perfectly periodic crystal structure.

At high temperatures and low pressures, ordinary substances are in the gas phase. Gases are unordered in contrast to solid states and can thus be treated with statistical methods. The properties of fluids are somewhat in between those of gases and solids.⁶ From typical experiments such as neutron or X-ray scattering it is known that **fluids** exhibit an ordered structure in small spatial domains. This is a property similar to crystals. However, the single molecules are not bound to any fixed point and thus, there is **no long-range order**. These properties render the study of fluids particularly difficult and interesting.

Beyond the critical point (T_c , p_c) any difference between fluids and gases vanishes because there is no difference in their densities any more. Below (T_c , p_c) one can distinguish gases and fluids e.g. by their different densities or compressibilities. The two phases are separated by *phase transitions of first order*. The local order of fluids is studied by means of the *pair correlation function* $g(r)$:

Definition 1.2.1 (Pair correlation function)

$$g(r) = \frac{V}{4\pi r^2 N^2} \left\langle \sum_i \sum_{j \neq i} \delta(r - r_{ij}) \right\rangle, \quad (1.3)$$

where V is the considered volume that contains N particles, $\delta(r - r_{ij})$ is Dirac's delta function and $r_{ij} = |\mathbf{r}_j - \mathbf{r}_i|$. The pair correlation function $g(r)$ expresses is the conditional probability density of finding a particle (or molecule) at position \mathbf{r} , provided that there is a particle at the coordinate origin $\mathbf{r} = 0$, cf. our discussion of statistical methods in Chap. 6. In Fig. 1.3 a typical pair correlation function of a fluid is displayed schematically.

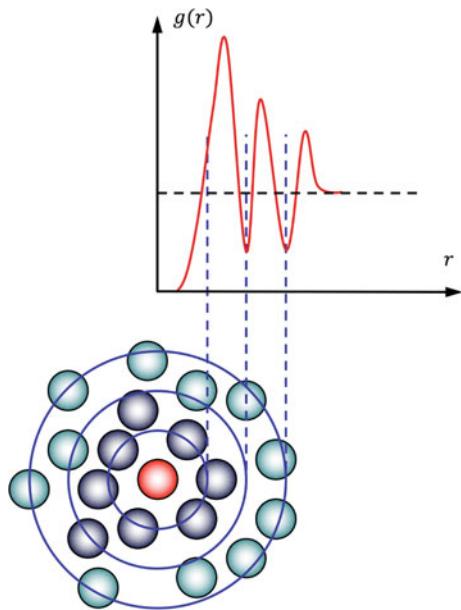
Typical are the peaks of $g(r)$ for small values of $r = |\mathbf{r}|$, resulting from the local order. This is similar to the behavior of an idealized solid where one obtains delta-peaks at multiples of the lattice constant. For large correlations, $g(r)$ becomes structureless like in a gas; $g(r)$ is then determined in essence by the density of the system. The advantage of studying the pair correlation function is, that it is directly accessible to scattering experiments with light, electrons or neutrons. The size of the scatterer determines which wavelength is appropriate. The intensity $I(\mathbf{k})$ of the scattered particles is given by

$$I(\mathbf{k}) \sim I_0 |a|^2 N S(\mathbf{k}), \quad (1.4)$$

where \mathbf{k} is the scattering vector, N is the number of particles, I_0 is the intensity of the incoming beam and a is the scattering amplitude. The intensity is proportional to

⁶There are, of course, more than only three phases of condensed matter as has been mentioned in the caption of Fig. 1.2. For example, a *plasma* is a state of matter where the electrons have been separated from their nuclei. *Quantum fluids* – that is Bose–Einstein condensates – e.g. ${}^3\text{He}$, show interesting properties that are very different from the ones of ordinary fluids.

Fig. 1.3 Pair correlation function $g(r)$ of a fluid



the *structure function* $S(\mathbf{k})$ which can be measured in experiments, and the structure function is related with the pair correlation via a Fourier-transformation:

$$S(\mathbf{k}) = 1 + \frac{N}{V} \int_V [g(r) - 1] e^{i\mathbf{k}\mathbf{r}} d^3r. \quad (1.5)$$

Thus, the pair correlation can be determined by Fourier transforming the measured structure function. Examples of structure functions obtained from simulations of polymeric fluids are presented and discussed in Chap. 6.

1.3 The Objective of Experimental and Theoretical Physics

Many laws of nature, so-called “basic” or “fundamental equations”, have been found in physics by conducting numerous experiments, collecting data of the phenomena under observation, and a gradual generalization from the specific experimental conditions. This method of extracting basic laws from experimental evidence is called the *inductive method* and is in fact the “traditional” way of making scientific discoveries. For example, Johannes Kepler (1571–1630) discovered the laws that bear his name by tediously trying different possible equations (using algebraic and geometric methods) for the description of the planets’ orbits around the sun and the time needed for a revolution; a numerical task that took many years of his life and which could nowadays probably be solved within seconds after having typed all the

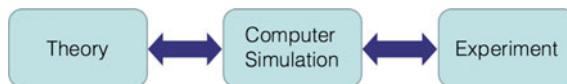


Fig. 1.4 Computer Simulation as a tool combining theory and experiment

available data into a modern personal computer (PC). Maxwell's equations or Newton's second law also has been formulated by generalization and deduction from the experimental observations.⁷ It is left to the reader whether he wants to believe that laws of nature are "free inventions of human mind" or whether these laws are literally written "in the book of nature" and are waiting to be discovered by some human ingenious deductions from experiments. In any case, some creativity is involved to achieve a formulation of generally accepted laws of nature in mathematical form from a mere data collection.

In physics, there is a second method of making true statements about nature which is called the *deductive method* and which is the objective of *theoretical physics*. On the one hand, starting with generally accepted principles, so-called axioms, theoreticians try to affirm their general validity by way of mathematical deduction such that no contradiction to known experimental results occurs. On the other hand, the objective is to deduce new physical laws from the given axioms and thereby to advance experimental research. As one of many examples we quote the prediction of the existence of the electron-neutrino as a new electric neutral particle in a famous letter of 4 December 1930⁸ by Wolfgang Pauli (1900–1958) [16], based on the observation that the conservation laws of momentum and energy in the β -decay are only fulfilled if a new by then unknown particle exists. The neutrino was empirically discovered not until 1956 [17].

With the advent of fast computer workstations and PCs since the mid 1980s, a new branch of scientific research has evolved during the past 30 years which is intermediate between theory and experiment, cf. Fig. 1.4. This is the area of *computer simulation*.

1.4 Computer Simulations – A Review

Computer simulation is adding a new dimension to scientific investigation and has been established as an investigative research tool which is as important as the "traditional" approaches of experiment and theory. The experimentalist is concerned with obtaining factual information concerning physical states and processes. The

⁷The validity of Newton's second law has been tested to the level of $5 \times 10^{-14} \text{ m/s}^2$ [14], which is a thousandfold improvement in precision over the best previous test [15].

⁸In this open letter to Lise Meitner on the occasion of a meeting of the "radioactive group" on 4 Dec 1930 in Tübingen, Germany, he started with the opening: "Dear radioactive ladies and gentlemen".

theorist, challenged by the need to explain measured physical phenomena, invents idealized models which are subsequently translated into a mathematical formulation. As is common in theory, most mathematical analysis of the basic laws of nature as we know them, is too complex to be done in full generality and thus one is forced to make certain model simplifications in order to make predictions. Hence, a comparison between a theoretical prediction and an experimental interpretation is frequently doubtful because of the simplifying approximations with which the theoretical solution was obtained or because of the uncertainty of the experimental interpretation. For example, even the relatively “simple” laws of Newtonian mechanics become unsolvable, as soon as there are more than two interacting bodies involved, cf. Example 37 on p. 258. Most of materials science however deals with many ($N \sim 10^{23}$) particles, atoms, molecules or abstract constituents of a system.

Computer simulations, or *computer experiments*, are much less impaired by non-linearity, many degrees of freedom or lack of symmetries than analytical approaches. As a result, computer simulations establish their greatest value for those systems where the gap between theoretical prediction and laboratory measurements is large. Instead of constructing layers of different approximations of a basic law (formulated as equations), the numerical approach simulates directly the original problem with its full complexity without making many assumptions. The principal design of virtually all computer simulation programs is displayed in Fig. 1.5 on p. 16.

Usually, during a *pre-processing phase* some administrative tasks are done (system setup, defining structure and simulation input variables, etc.) before the actual simulation run is started. Analyzing data “on-the-fly” during the *simulation phase* is usually too expensive; therefore, data snapshots of the system are stored during certain preset time intervals which can later be analyzed and visualized during the *post-processing phase*. Very often, the pre- and post-processing code is separated from the main simulation code and since the mid 1990s, GUIs⁹ (Graphical User Interfaces) are commonly used for these tasks. The starting point for a computer simulation is the invention of an idealized adequate model of the considered physical process. This model is written in the language of mathematics and determined by physical laws, initial and boundary conditions. The question, as to when a model is “adequate” to a physical problem is not easy to answer. There are sometimes many concurrent modeling strategies and it is a difficult question which aspects are essential and which ones are actually unimportant or peripheral. For example, modeling impact phenomena between high-energy particles requires relativistic quantum mechanics, whereas in a classical model, energy dissipation caused by friction probably has to be taken into account on a phenomenological level.

In many cases, the basic question as to which model is to be used is the main problem. The next step to prepare the model for an algorithmic description, is the discretization of the time variable (for dynamic problems) and of the spatial domain

⁹In UNIX environments, TCL/TK is the classical script language used to program GUIs. Since the mid 1990s, a C++ based graphical library – Qt – is available for Open-Source developments under the GNU General Public Licence.

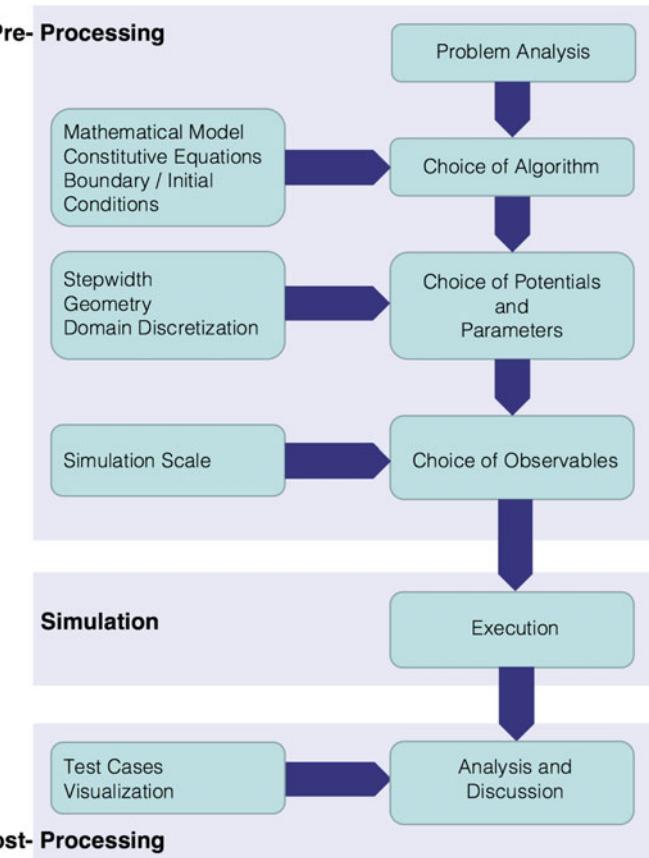


Fig. 1.5 Principal design of a computer simulation. Usually, some pre-processing as preparation for the main simulation is done with a *pre-processor*. This piece of computer code might be integrated in the main source code or – in particular in commercial codes – is written as an extra piece of code, compiled and run separately from the main simulation code. During pre-processing, many administrative tasks can be done which are not related to the actual simulation run. In large-scale simulations, data are stored during execution for later analysis. This analysis and visualization is done during post-processing. The advantage of separating pre- and post-processing from the actual simulation code is that the code design remains clearly arranged and the important issue of optimization for speed only has to be done for the relevant pieces of the main simulation code

in which the constitutive equations of the problem are to be solved. Then appropriate algorithms for solving the equations of the mathematical model have to be chosen and implemented on a computer. Before trusting the output of a newly written computer program and before applying it to new problems, the code should always be tested to that effect, whether it reproduces known analytic or experimental results. This is a necessity as the correctness and plausibility of the outcome of an algorithm (usually dimensionless numbers) cannot be predicted by looking at the source code.

The success of a computer experiment in essence depends on the creation of a model which is sufficiently detailed such that the crucial physical effects are reproduced and yet sufficiently simple for the simulation to still be feasible.

1.4.1 A Brief History of Computer Simulation

The origins of modern computer systems, i.e. systems that are not based on purely mechanical devices, lie in the development of the first programmable computer Z3 by Konrad Zuse in the 1930s in Germany. More well known however is the American Electronic Numerical Integrator And Computer ENIAC which was developed during war time at the Los Alamos Laboratories (founded in 1943) and which started to operate in 1946. It was used for calculating ballistic curves and for hydrodynamics calculations of shock waves in the context of thermonuclear reactions. The ENIAC (displayed in Fig. 1.6) weighed 30 tons, used more than 18,000 vacuum tubes (thus counting to the “first generation” of electronic computers) and computed 1000 times as fast as its electro-mechanical competitors [18]. It was programmed by plugging cables and wires and setting switches, using a huge plugboard that was distributed

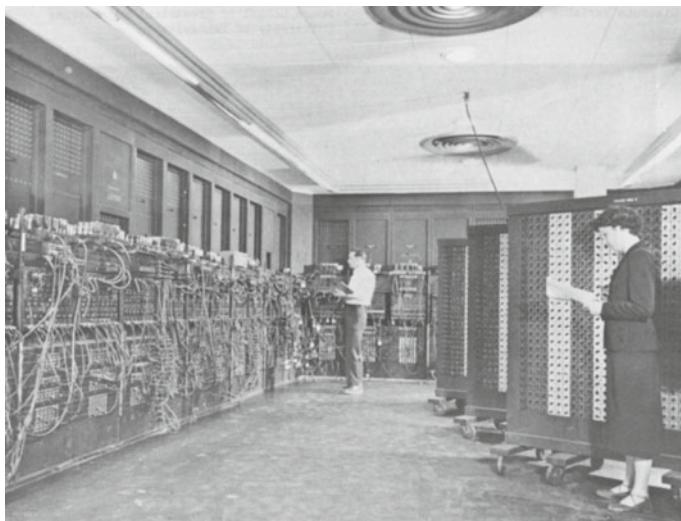


Fig. 1.6 Programming panels and cables of the ENIAC on the left. To reprogram the ENIAC one had to rearrange the patch cords that one can observe on the left and the settings of 3000 switches at the function tables that one can observe on the right. The very first problem run on ENIAC was on the feasibility of the hydrogen bomb. It required only 20 seconds after processing half a million punch cards for six weeks. This very first ENIAC program remains classified even today. U.S. Army Photo

Box 1.1 The von Neumann computer model

The von Neumann principles state that any computer should consist of a CPU unit, a controller unit, a memory unit, an input/output unit and a data bus unit. The CPU carries out the arithmetic and logic calculations, the memory unit allows a program to be loaded from an external source and can store intermediate results. The controller unit allows for the communication of the CPU with peripheral hardware, the I/O unit makes it possible to communicate data from/to external sources and the system bus represents a communication network among the above mentioned parts.

over the entire machine. ENIAC's basic clock speed¹⁰ was 100,000 cycles per second. For comparison, a typical PC today with an Intel i7 CPU employs clock speeds of at least 4,000,000,000 cycles per second.

In 1946, A.W. Burks, H.H. Goldstine and John von Neumann published a paper [19] that set forth the principles according to which a computer has to be built, which were later called the “von Neumann principles”, see Box 1.1. In this paper, it was realized that a program could be represented electronically just as the data, thus doing away with the cumbersome way of programming by a physical modification of all the switches and patch cords. The MANIAC (Mathematical Analyzer, Integrator And Computer) was built from 1948–1952 at Los Alamos Laboratories according to the von Neumann principles (Fig. 1.7). The MANIAC was programmed at the lowest level, in machine language, directly addressing and shifting contents of data cells in memory.

In the beginning 1950s, electronic computers became ever more available for non-military use. This marks the beginning of the field of scientific computer simulation. Among the first non-military problems carried out with the MANIAC were phase shift analysis problems of the pion-proton scattering by Enrico Fermi, Hans Bethe, Nicolas Metropolis et al. [20, 21], non-linear oscillator studies¹¹ [22] and some first early studies of the genetic code by G. Gamov [23]. One of the most famous calculations on the MANIAC was done in the year 1953 by N. Metropolis et al. [24]: the equation of state in two dimensions for a system of rigid spheres was calculated using the Monte Carlo (MC) method.¹² These calculations on the classical mechanical N -body problem introduced what is today known as *importance sampling*, also referred to as the *Metropolis algorithm* (discussed in Sect. 2.6.4 in Chap. 2 and in Sect. 6.6 in Chap. 6). This algorithm, which was ranked as one of the “10 algorithms with the greatest influence on the development and practice of science and engineering in the 20th century” [26], is still in widespread use today.

The first ever molecular dynamics (MD) simulation was reported by Alder and Wainwright in 1956 at a Symposium on Transport Processes in Brussels. The proceedings were not published until 1958 [27]. In their paper they calculated the velocity autocorrelation function and the self-diffusion coefficient for a system of $N = 256$

¹⁰The number of pulses per second of the CPU.

¹¹These studies later gave rise to the field of soliton research.

¹²The term “Monte Carlo” goes back to a 1949 paper by Metropolis and Ulam, see [25].

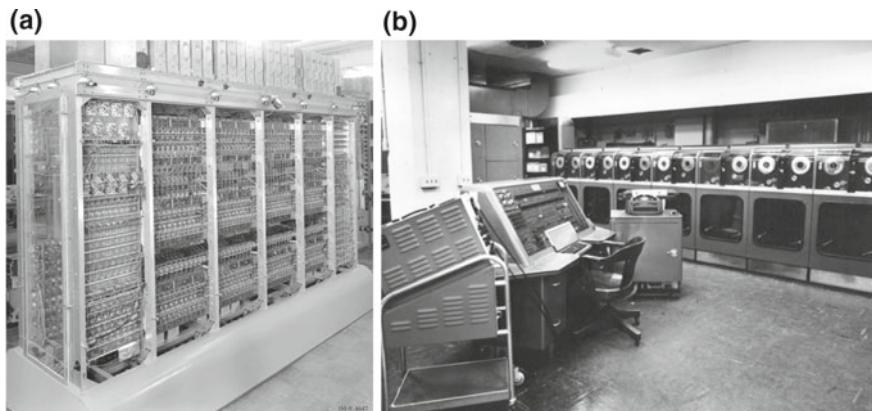


Fig. 1.7 First generation computers from the early 1950s. **a** The MANIAC computer, developed by John von Neumann and completed in 1952, was responsible for the calculations of “Mike”, the first hydrogen-bomb. Von Neumann also helped Ulam and Metropolis develop new means of computing on such machines, including the Monte Carlo method, which has found widespread application and is considered today among the top 10 algorithms of the 20th century [26]. **b** The first commercial (non-military) computer – UNIVAC – built in the United States. UNIVAC was designed by J. Presper Eckert and John Mauchly, the designers of the ENIAC. The machine was 7.62 m by 15.24 m in length, contained 5,600 tubes, 18,000 crystal diodes, and 300 relays. It utilized serial circuitry, 2.25 MHz bit rate, and had an internal storage capacity of 1,000 words or 12,000 characters. The UNIVAC was used for general purpose computing with large amounts of input and output and it was also the first computer to come equipped with a magnetic tape unit and to use buffer memory. Its reported processing speed was 0.525 milliseconds for arithmetic functions, 2.15 milliseconds for multiplication and 3.9 milliseconds for division

hard spheres¹³ with periodic boundary conditions. The calculations were done on a “UNIVAC” calculator (*UNI*versal *A*utomatic *C*omputer), the first commercial mass produced computer (Fig. 1.7) and later on an IBM-704. In 1957, Wood and Jacobsen [28] and Alder and Wainwright [29] showed with a molecular dynamics simulation that a molecular fluid of spherical particles which have *no* attractive interaction, but only a short-ranged repulsion, show a first-order phase transition, i.e. they showed that such a system can in fact crystallize. The first simulation of a “real” fluid using MD was published in 1964 by Rahman who simulated 864 Lennard-Jones particles of fluid argon [30].

During the 1950 and 1960s only slow progress in computational methods was achieved, which was probably mainly due to the limited computing resources available during that time. The so-called *second generation* of computers saw several important developments at all levels of computer system design, from the technology

¹³Today, as a rule of thumb, $N \sim 10^5$ particles is about the system size limit for a simple hard sphere MD simulation that can still be usefully run on a single PC, however producing much better statistics than was possible in 1957.



Fig. 1.8 The IBM 7094, a typical mainframe computer of the 1970s. Photo courtesy of IBM

used to build the basic circuits to the programming languages used to write scientific applications. Electronic switches in this era were based on discrete diode and transistor technology with a switching time of approximately $0.3\ \mu\text{s}$. Memory technology was based on magnetic cores which could be accessed in random order. Important innovations in computer architecture included index registers for controlling loops and floating point units for calculations based on real numbers. Prior to this, accessing successive elements in an array was quite tedious and often involved writing self-modifying code (programs which modified themselves as they ran) which is impossible in most high level languages. At the time viewed as a powerful application of the principle that programs and data were fundamentally the same, today, this practice is now frowned upon as spaghetti-coding which leads to programs that are extremely hard to debug and read. During this era many high level programming languages were introduced, including FORTRAN (1956), ALGOL (1958), and COBOL (1959). Important commercial machines of this era include the IBM 704 and its successors, the 709 and 7094, see Fig. 1.8. None of these first high-level computer languages are used today anymore for scientific computing, except maybe for FORTRAN because of some very old legacy codes which are still in use. Any new software for scientific use in high-performance computing today is written almost exclusively in C or C++.

The second generation of hardware also saw the first two supercomputers designed specifically for numeric processing in scientific applications. The term “supercomputer” is generally reserved for a machine that is at least an order of magnitude more powerful than most other machines of its era. Since 1983, twice a year, a list of the

sites operating the 500 most powerful computer systems is assembled and released.¹⁴ The best performance on the Linpack benchmark¹⁵ is used as performance measure for ranking the computer systems. The list contains a variety of information including the system specifications and its major application areas.

Any PC of today is much more powerful than the mainframe computers of the 1960 and 70s, cf. Fig. 1.8. Mainframe computers also were very expensive as most of the assembly, such as the wiring had to be done by hand and they were typically installed at large computing centers at universities, research institutes and in the aeronautics, automotive, defense, and nuclear industries. This era of the *third generation* of computers brought innovations in memory technology along with the miniaturization of transistors in integrated circuits (semiconductor devices with several transistors built into one physical component), and the use of integrated circuits semiconductor memories which were started to be used instead of magnetic cores. Also microprogramming as a technique for efficiently designing complex processors, the coming of age of pipelining and other forms of parallel processing and the introduction of operating systems that included time-sharing culminated in the invention of the microprocessor at the beginning of the 1970s by Intel. Intel produced the 8088 processor¹⁶ which was to become the first personal computer. In 1981 IBM decided to standardize on the Intel line of 8088 processors for their line of PCs. The Intel i7 processor of today's PCs is still compatible with the 8088 processor of IBM's first PC.

The alternative to time sharing in the 1970s was batch mode processing, where the computer runs only one program at a time. In exchange for getting the computer's full attention at run-time, one had to agree to prepare the program off-line on a key punch machine which generated punch cards, cf Fig. 1.9. Each card could hold only 1 program statement. To submit one's program to a mainframe, one placed the stack of cards in the hopper of a card reader. The program was submitted and would be run whenever the computer made it that far. When one came back later, one could see at the printout showing the results whether the run was successful. Obviously, a program run in batch mode could not be interactive. Two important events marked the early part of the third generation: the development of the C programming language and the UNIX operating system, both at Bell Labs. In 1972, Dennis Ritchie and Brian Kernighan, seeking to generalize Thompson's language "B", developed the C language [31]. Thompson and Ritchie then used C to write a version of UNIX for the DEC PDP-11. This C-based UNIX was soon ported to many different computers, relieving users from having to learn a new operating system each time when computer hardware was changed. UNIX or one of the many derivatives of UNIX (such as Linux, developed in 1994) is now a de facto standard on virtually every computer system which is used for scientific high-performance computing.

¹⁴<http://www.top500.org>.

¹⁵<http://www.netlib.org/linpack/index.html>.

¹⁶Intel priced the 8088 microprocessor at \$360 as an insult to IBM's famous 360 mainframe which cost millions of dollars.



Fig. 1.9 An IBM key punch machine of the 1970s which operates like a typewriter except it produces punched cards (lying on the table) rather than a printed sheet of paper. Photo courtesy of IBM

The *fourth generation* of computer systems in the 1980s saw the use of large scale integration (more than 100, 000 devices per chip) in the construction of computing elements. At this scale entire processors may fit onto a single chip. Gate delays dropped to about 1 ns per gate. Semiconductor memories replaced core memories as the main memory in most systems; until this time the use of semiconductor memory in most systems was limited to registers and cache. During this period, high speed vector processors, such as the CRAY 1, CRAY X-MP and CYBER 205 dominated high performance computing, cf. Fig. 1.10. Computers with large main memory, such as the CRAY 2, began to emerge. A variety of parallel architectures began to appear. However, during this period, the parallel computing efforts were of a mostly experimental nature and most computational science was carried out on vector processors. Microcomputers and workstations were introduced and were widely used as an alternative to time-shared mainframe computers.

The development of the next generation of computer systems was characterized mainly by the acceptance of parallel processing. Until this time parallelism was limited to pipelining and vector processing, or at most to a few processors sharing jobs. This *fifth generation* of computers saw the introduction of machines with hundreds of processors that could all be working on different parts of a single program. Other new developments were the widespread use of computer networks and the increasing use of single-user workstations. Scientific computing in this period was still dominated by vector processing. Most manufacturers of vector processors introduced parallel models, but there were very few (two to eight) processors in this parallel machines. In the area of computer networking, both wide area network (WAN) and



Fig. 1.10 The Cray X-MP, the world's fastest supercomputer from 1983–1985 with a total operating performance of 800 Megaflops (Floating point operations per second). Note the Cray-typical circular arrangement of processor stacks. Photo courtesy of Michael John Muuss

local area network (LAN) technology developed at a rapid pace, stimulating a transition from the traditional mainframe computing environment toward a distributed computing environment in which each user has his own workstation for relatively simple tasks (editing and compiling programs, reading mail) but sharing large, expensive resources such as file servers and supercomputers. RISC¹⁷ technology (a reduced and hard-coded CPU instruction set which allows much faster loading and parallel processing of commands) and decreasing costs for RAM (Random Access Memory) brought tremendous gains in computational power of relatively low cost workstations and servers. This period also saw a marked increase in both the quality and quantity of scientific visualization which today is a real problem with multi-billion particle simulations.

Important Algorithmic Developments

It is probably correct to say that most of the basic numerical algorithms and the general methodology, cf. Fig. 1.5 on p. 16, were developed during the first 30 years of computer simulation from the 1940s to 60s. In the following we only mention a few of many important developments.

One of the first problems treated numerically with a computer was the numerical simulation of the properties of dense liquids (phase diagrams, equations of state)

¹⁷Reduced Instruction Set Computing. Virtually every CPU today is a RISC processor, or at least contains RISC elements.

which is a N -body problem from statistical mechanics that cannot be solved analytically. Developments of numerical hydrodynamics in so-called *hydrocodes*, e.g. the introduction of the concept of artificial viscosity, goes back to military high-velocity and shock physics research conducted by E. Teller at Los Alamos Laboratories in the 1940 and 50s (see e.g. [32]). For a more detailed discussion, see [33]. Later, in the beginning 1950s, when computers became increasingly available for non-military use, these methods were applied for shock wave research in astrophysics and cosmology. The numerical developments finally resulted in the publication of a classic paper by Lucy in 1979 [34], introducing a new method, called SPH which is based on the idea of replacing the equations of fluid dynamics by equations for particles. It has the advantage of avoiding many mesh distortion problems of ordinary Lagrangian mesh-based numerical methods. This new idea was integrated in many commercial and research hydrocodes available for engineering applications, discussed in Chap. 7. The name “hydrocode” is derived from the fact that these codes in their early implementations in the 1950s only were applied to fluid dynamics problems. Hydrocodes – sometimes also called *wavecodes* due to their emphasis on shock wave propagation – have provided hydrodynamics simulations of materials for a long time. Later however, when material strength was included, the misnomer was kept.

Inspired by the success of early numerical simulations, the various methods were gradually applied to many other problems in physics, mathematics, chemistry and biology.

In 1965, the *Fast Fourier Transform* (FFT) algorithm, a sophisticated method based on recursion for the calculation of the discrete Fourier transform of numbers was developed by Cooley and Tukey [35] which reduces the effort of calculation to $\mathcal{O}(N \log N)$.

In 1987 the *Fast Multipole Algorithm* (FMA) was developed by Greengard and Rokhlin [36] for the calculation of N -body problems governed by Coulombic interactions (discussed in Chap. 6), which reduced the amount of work needed from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. The last two algorithms were ranked among the ten most significant algorithms of the 20th century [26].

Loup Verlet published his basic algorithm which is usually the best for the integration of classical equations of motion in 1967 [37]. This algorithm is discussed in detail in Chap. 4. In the field of genome research, in 1970, Needleman and Wunsch developed the first algorithm for the direct, global comparison of protein sequences [38]. In genetic epidemiology, the fundamental *Elston-Steward algorithm* which calculates the likelihood of a pedigree of nuclear families, was developed in 1971 [39] and is still used today in genetic association and coupling studies.

Today, at the high end of supercomputing are new technologies, such as *cell-based chips*, which include several CPU cores on one single chip that can work in parallel with peak performances in the order of $\sim 10^7$ gigaflops, gradually approaching the exaflops (10^{18} flops) barrier. During the last decade *Grid Computing* evolved as a new key technology enabling scientists and engineers in research to use supercomputing resources of many research centers in heterogeneous environments through high-speed networks and may thus change the way how supercomputing is done in the future, see e.g. [40].

1.4.2 Computational Materials Science

With the increasing general availability of fast computers since the 1980s, also numerical simulations of practical interest in the engineering sciences (traditionally based almost solely on finite element analysis) for product design and testing became applicable. Materials systems of industrial interest are highly heterogeneous and are characterized by a variety of defects, interfaces, and microstructural features. Predicting the properties and performance of such materials is central for modern product design in industry. It was only over the past five to ten years that *multiscale* materials modeling on computers evolved to a cost-efficient working tool of industrial interest. Unfortunately, due to the complexity of structural hierarchies in condensed matter on different scales, there is no *single* all-embracing computational model or physical theory which can explain and predict all material behavior in *one* unified approach, cf. our discussion of the role of models in the natural sciences in Chap. 2. Therefore, the explicit architecture of different important classes of materials such as metals, ceramics, glasses or polymers on different length scales, is incorporated in different models with limited validity.

As mentioned above, the first simulation methods ever developed and implemented on full electronic computers were MC and MD methods, fully rooted in *classical physics*. Many problems of classical MD techniques lie in the restriction to small (atomic and microscopic) length and time scales. In atomistic MD simulations of hard matter, i.e. crystalline systems which are mostly governed by their available energy states, the upper limit is typically a cube with an edge length of a few hundred nanometers simulated for a few nanoseconds. With *coarse-grained models*, (where the individual MD particles represent complete clusters of atoms or molecules) this limit can be extended to microseconds or even seconds. In this respect, so-called *complex fluids* – in modern terminology called *soft matter systems* – such as polymers (very long chain molecules,¹⁸ discussed in Chap. 6) are an exceptionally interesting class of materials. *Quantum mechanical* computer simulation methods were also developed, e.g. *density functional theory* (DFT) which calculates the ground state energy of many particle systems or ab initio methods that combine DFT with classical MD in a way that the degrees of freedom of the electrons can be treated explicitly, in contrast to using “effective potentials” between atoms. These methods are even more restricted with respect to time and length scales as displayed in Fig. 1.1 and Table 1.1.

Microscopic MD methods for typical engineering applications on dislocation dynamics, ductility and plasticity, failure, cracks and fracture under shock loading in solids were extended to large-scale simulations of more than 10^8 particles during the late 1990s by Abraham and coworkers [41, 42], Holian and Lomdahl [43] and others [44]. These large scale simulations present a serious analysis problem due to the astronomical size of data sets produced during a typical MD simulation. For

¹⁸In fact, human DNA is the longest polymer occurring naturally with a number of repeating units on the order of $N \sim 10^9$.

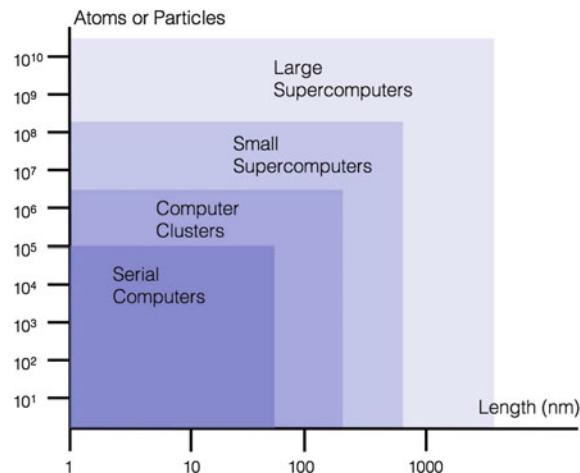
example, the smallest useful data set that needs to be stored at each MD timestep contains at least the three particle coordinates (X , Y , Z). In double precision (8 bytes per coordinate), which is generally needed to be able to restart a system based on the coordinates of a previous run, this amounts to 24 gigabytes of storage needed just for *one* snapshot of the system when performing a billion particle simulation. Usually – besides the particle coordinates – the velocities, potential and kinetic energies and a few more quantities are saved which results in roughly 10 doubles needed to be stored at each time step. Thus, the minimum of needed storage space per time step is roughly 80 GB for a billion particle simulation. This amount of data requires sophisticated analysis tools and also on-the-fly analysis capabilities of the simulation software, even on modern platforms with several hundreds of Tbytes of hard-disk capacity.

Atomic oscillations (e.g. C-C-bond oscillations in polyethylene molecules) are typically of the order 10^{14} Hz. These atomic oscillations have to be resolved for evolving the system correctly. Thus, one time step in an MD simulation is typically of the order of femtoseconds ($\sim 10^{-15}$ s); with several million timesteps that can be simulated in an MD run, the largest available length- and timescales for atomic systems are typically limited to the order of a few hundred nanometers simulated for a few hundred nanoseconds. With larger computer systems, one will be able to simulate even larger systems; however, the available time scale does not necessarily grow with the number of available processors, as the time domain cannot be decomposed and be distributed over many CPUs as it is done with the spacial domain.

Today, more than 10^9 particles have been simulated in atomistic shock wave and ductile failure simulations [45] and the largest hitherto reported atomistic simulation run was performed with more than 1.9×10^{10} particles [46], opening the route to investigations of physical structure-property phenomena on the micrometer scale, which overlaps with typical computational methods based on continuum theory.

The improvement of simulation techniques which allow for a coupling of different length and time scales and for simulating larger systems on mesoscopic and macroscopic scales has been a very active field of research in computer simulation during the last ten years and many research programs devoted to intensive methodological development have been initiated in many research areas. Thus, developing new efficient algorithms and new methods for modeling and simulating physical systems has by no means come to an end, cf. [47]. Even in MD and MC methods, neither their efficient implementation nor identifying their areas of application have been completely perfected. Coupling strategies and extending the available physical length and time scales in simulations up to at least micrometers and -seconds and beyond in atomistic particle simulations remains one of the top motivations in all different fields of basic and applied research using supercomputers, cf. Fig. 1.11.

Fig. 1.11 Illustration of the available system size (edge length of a simulated cube of classical particles or atoms) and the necessary computer hardware in modern computational science



1.5 Suggested Reading

As stated in the preface to the first edition of this monograph, probably with no book one can hope to cover *all* aspects and details of the huge field of computer simulations in the natural sciences. Ever since the method of computer simulation has evolved into a basic, common tool in research and industry during the 1990s, several excellent (however often outdated) reference books have been published which are devoted to different single aspects of computer simulations of fluids and solids on multi-scales, e.g. Hockney [48], Allen and Tildesley [49], Ciccotti et al. [50], Hoover [51], Heermann [52], Haile [53], Rappaport [54], Raabe [55], Landau and Binder [56], Frenkel and Smit [57], Ray [58] or Engquist et al. [59]. Other books dealing with multiscale modeling are often mere collections of articles by many different authors that offer no real guidance in obtaining an overview of the field and no unified nomenclature - hence, such books are often of very limited practical value such as the voluminous *Handbook of Materials Modeling* edited by S. Yip [60] which is a collection of 150 articles by different authors on the topic of computational approaches in materials research.

Recent reviews on computational techniques are e.g. Cox et al. [61] who discuss the field of atomistic dynamic fracture simulations, Steinhauser [62] who discusses coarse-grained simulation approaches with recent applications in the field of polymer physics, Li and Liu [63] discussing meshless particle methods with applications, Gates et al. [64] who discuss multi-scale modeling and simulation techniques for advanced materials such as high-performance nanotube-reinforced composites and polymers, or Demenget et al. [65] who discuss multiscale experimental and computational challenges in biology and medicine.

Books that are focused on certain modeling aspects and applications on multiscales are e.g. Lépineoux [66] (plasticity), Chuang et al. [67] (deformation and fracture), del Piero [68] (structure deformations), or Alt et al. (Eds.) [69] (Polymer and Cell Dynamics).

I think there is a world market for maybe five computers.

Thomas Watson, IBM chairman, 1943

Abstract

In this chapter we start with a discussion of a fundamental assumption on the nature of matter, namely the idea of particles as basic constituents and the consequences thereof. In Sects. 2.1–2.3 we focus on clarifying some terminology in computational multiscale modeling and learn about the concepts of “models” in science. Section 2.4 focuses on hierarchical modeling concepts in classical and quantum physics which is followed by discussing the standard model of elementary particle physics and the attempts of reducing all understanding of material behavior to the interactions of some basic constituents (particles). In Sect. 2.6 we introduce the basics of computer science such as recursive programming, the concepts of formal (computer) languages, finite automata, complexity theory and the Turing machine. The chapter ends with some reading suggestions and several problems as an offer for the ambitious reader.

One might wonder why one does not derive all physical behavior of matter from an as small as possible set of fundamental equations, e.g. the Dirac equation of relativistic quantum theory. However, the quest for the fundamental principles of physics is not yet finished; thus, the appropriate starting point for such a strategy still remains unclear. But even if we knew all fundamental laws of nature, there is another reason, why this strategy does not work for ultimately predicting the behavior of matter on any length scale, and this reason is the growing complexity of fundamental theories – based on the dynamics of particles – when they are applied to systems of macroscopic (or even microscopic) dimensions.

The idea that matter is made of small particles, called atoms,¹ goes back to the ideas of Leucippus and Democritus (460–370 B.C.) in classical Greek philosophy of the fifth century B.C. see e.g. [70, 71], and has been very successful in the development of modern concepts in physics. Introduced as a working hypothesis in chemistry by John Dalton² (1766–1847) at the beginning 19th century for explaining the stoichiometry in chemical reactions, the reality of atoms was finally generally accepted among the scientific community roughly 100 years later due to overwhelming experimental evidence and theoretical achievements, e.g. Boltzmann's kinetic gas theory which is based on the pre-condition that atoms exist, or Einstein's famous 1905 paper [74] in which he developed a statistical theory of the Brownian motion which allowed to calculate the size of molecules and atoms.

Despite these theoretical achievements, there were famous opponents such as Ernst Mach (1838–1916) who was captured in his philosophy of positivism, only accepting empirical data as basic elements of a physical theory. As atoms at that time could not be directly observed, he attributed them to the realm of metaphysical nonsense³. Albert Einstein (1879–1955) later wrote about his 1905 paper:

My principal aim was to find facts that would guarantee as much as possible the existence of atoms of definite size. [...] The agreement of these considerations with Planck's determination of the true molecular size from the laws of radiation (for high temperatures) convinced the skeptics, who were quite numerous at that time (Ostwald, Mach), of the reality of atoms.

(Albert Einstein [76], 1946, p. 45)

In fact, modern experiments with the largest microscopes available, that is particle accelerators of high-energy physics, revealed that even the constituents of atoms, neutrons and protons themselves show an inner structure and are composed of yet even smaller particles, so-called *quarks*.⁴ The idea of the existence of ever smaller particles of matter seems to have come to an end with quarks, due to *quark confinement*, a property of quarks which renders it impossible to observe them as isolated particles.⁵

¹From Greek “ἄτομο” (indivisible).

²The original publication is [72]. For a review, originally published in 1856 and available today as unabridged facsimile, see [73].

³In contrast to Ernst Mach – roughly half a century later – Richard Feynman starts the first chapter of his famous lecture series on physics [75] with the remark that the atomic hypothesis, i.e. the idea that matter is made of single small particles, contains the most information on the world with the least number of words.

⁴This peculiar naming of the currently smallest known constituents of matter after the sentence “Three quarks for Master Mark” that appears in James Joyce's novel “Finnegan's Wake”, goes back to Murray Gell-Mann (Nobel prize 1969).

⁵This property of the strong interaction was discovered by D. Gross, D. Politzer, and F. Wilczek (Nobel prize 2004) and is due to an increase of the strong coupling constant (and along with it an increase of the strong force) with increasing distance of the quarks. That is, if one tries to separate quarks, energy is “pumped” into the force field until – according to $E = mc^2$ – quark-anti-quark systems come into being. The original publications are [77–79].

At the beginning 20th century it was realized that the classical (pre-quantum) laws of physics could not be valid for the description of systems at the atomic scale and below. Rutherford's scattering experiments with H_2 -particles hitting a gold foil in 1911 [80] had shown that atoms could not be elementary. The eventual development of the formulation of a quantum theory, during the years 1925–1927, also changed the “scientific paradigm” of what is to be considered “understanding” and also of what a physical theory is expected to accomplish. In a passage of his book “Der Teil und das Ganze” [81], Werner Heisenberg (1901–1976) writes in Chap. 5 about a discussion with Albert Einstein in which Einstein claims that the idea to base a theory only on *observable* elements, i.e. on elements or objects which are measurable and perceptible in experiments, is nonsense. It is interesting to note that Einstein used this “philosophy” himself as a heuristic concept for deriving the special theory of relativity, eliminating such unobservable, metaphysical concepts like “absolute space”, “absolute time”, and the idea of an “ether”, an ominous substance which – in 19th century physics – was supposed to define an inertial system (IS) and thus, an absolute frame of reference in space. According to Einstein – as Heisenberg writes – it is *theory* that determines what is measurable in experiments and not vice versa.

In this context one may ask questions such as: “What actually *is* a “theory” and what are the characteristics of a theory?” “What does “modeling” actually mean?” “What *is* a model?” “Is a model “reality”?” “What is “reality” in the natural sciences?” “What is the difference between a model and a fundamental law of nature?” And dealing with computational physics one could consequently ask the question as to what degree the result of a computer program can be considered “reality”. Albert Einstein writes about the relevance of such epistemological questions in the natural sciences in a 1916 memorial lecture for Ernst Mach:

How does it happen that a properly endowed natural scientist comes to concern himself with epistemology? Is there no more valuable work in his specialty? I hear many of my colleagues saying, and I sense it from many more, that they feel this way. I cannot share this sentiment. When I think about the ablest students whom I have encountered in my teaching, that is, those who distinguish themselves by their independence of judgment and not merely their quick-wittedness, I can affirm that they had a vigorous interest in epistemology. [...] Concepts that have proven useful in ordering things easily achieve such an authority over us that we forget their earthly origins and accept them as unalterable givens. Thus they come to be stamped as “necessities of thought”, “a priori givens”, etc. The path of scientific advance is often made impassable for a long time through such errors. For that reason, it is by no means an idle game if we become practiced in analyzing the long commonplace concepts and exhibiting those circumstances upon which their justification and usefulness depend, how they have grown up, individually, out of the givens of experience. By this means, their all-too-great authority will be broken.

(Albert Einstein [82], 1916, pp. 101–102)

Obviously, when applying or deriving physical theories and models of reality, there are certain implicit and usually unspoken assumptions being made. In the following sections it is tried to provide a few suggestions as answers to the above questions and to discuss materials science within this context of model building. Then, in

Sect. 2.5, the degree to which physical laws have been unified in modern physics is shortly discussed. Finally, in Sect. 2.6, some fundamentals of computer science and the notions of “algorithm” and “computability” and their eventual formalization in the concept of a *universal Turing machine* are discussed.

2.1 Some Terminology

In science, one seeks after *systems* in which all different notions, concepts, and theoretical constructs are combined into one consistent *theory* which explains the diversity of physical phenomena with very few, general *principles*. Usually nothing can be changed in this system, or else it fails. This fact is the hallmark of a physical theory. A prominent example is the (*strong*) *principle of equivalence*, which states that the inertial mass m_I (a measure of a body’s resistance against acceleration), the passive gravitational mass m_P (a measure of the reaction of a body to a gravitational field) and the active gravitational mass m_A (a measure of an object’s source strength for producing a gravitational field) are the same. Thus, one can simply refer to the *mass* of a body, where $m = m_I = m_P = m_A$. If this principle was found to be wrong in any experiment in the future, the whole theory of general relativity – which is a field theory of gravity – would completely break down, because it is based fundamentally on this principle. However, in Newton’s theory of gravitation, this principle just appears as a theorem which states an empirical observation – another coincidence. Nothing follows from it, and nothing would change in Newton’s theory if it was not valid.

In mathematics, such principles, which form the theoretical basis of all developments, are called *axioms*. Together with a set of rules of inference and theorems, derived according to these rules, they build a theory. In fact, the degree to which a theory or model has been “axiomatized” in the natural sciences can be used as a criterion, as to whether a theory is considered to be as “closed”. Examples of *closed theories*, i.e. model systems in which all hitherto known experimental facts can at least in principle be explained and derived from a handful of axioms, are all *classical theories* in physics, based on Newtonian mechanics, such as mechanics, thermodynamics, electrodynamics and also the special theory of relativity. These theories are thought to be on excellent ground in both evidence and reasoning, but each of them is still “just a theory”. Theories can never be proved and are subject to tests. They can only be falsified. They are subject to change when new evidence comes in.

In the previous considerations we have repeatedly used the words “system”, “theory”, and “model”. These terms are generally more or less mixed up and used interchangeably in the context of model building and there exists no strict, commonly accepted definition of these terms; however, “theory” usually is used in science as a more general term than “model” in the sense that a theory can combine different models (e.g. particles and waves) within one theory (e.g. quantum field theory). Hence, the term “model” generally refers to a lower level of abstraction, whereas a complete theory may combine several models.

In this volume no particular distinction between “model” and “theory” is made and the two terms are used interchangeably. Finally, a “system” is a theoretical construct which includes all different hierarchical basic axioms, notions, models, and theories which pertain to some specific phenomenon in a way, which renders it impossible to change anything within the system without making it fail. If, despite numerous possibilities to be falsified, a certain system or theory does not lead to any contradiction to known experimental facts, then it is called a *law of nature*.

2.2 What Is Computational Material Science on Multiscales?

Strictly speaking, materials science of condensed matter today is focused on the properties of condensed matter systems on a length scale comprising roughly 10–12 orders of magnitude, ranging from roughly 10 Å to a few hundred meters for the largest buildings or constructions. The two extremes, physics at the smallest scales below the atomic dimensions ($\leq 1 \text{ \AA}$) or under extreme conditions, e.g. Bose–Einstein condensates, or the properties of neutrons and protons, as well as large-scale structures such as stars or galaxies and galaxy clusters, which are millions of light years in extend, are not an object of study in materials science or engineering. Nevertheless, it is physics, that provides the basic theories and modeling strategies for the description of matter on all scales.

The reason why there is no single, perfect and all-comprising model for calculating material properties on all scales relevant for materials science, is, that nature exhibits complex structural hierarchies which occur in chemistry and engineering devices as well as in biological systems (self-organization of matter), investigated in the life sciences. Remarkably, on the Ångstrømscale there are only atoms but then on larger length scales these basic constituents build complex hierarchical structures in biology and chemistry [83], which are treated with different theories that have a certain range of applicability, see Fig. 2.1. These different theories have to prove successful in comparison with experiments.

Materials of industrial importance such as glasses, ceramics, metals or (bio) polymers, today are increasingly regarded as hierarchical systems, cf. Fig. 2.2 and there has been a focus of research on the investigation of the different structures of components, of classes of materials on various structural hierarchies, and their combination within “process chains”. Hence, the typical structural or architectural features of materials on different scales have to be taken into account. Usually one of two possible strategies is pursued. In a *bottom-up* approach the many degrees of freedom on smaller scales are averaged out to obtain input parameters relevant on larger scales. In contrast, a *top-down* approach tries to establish the structure on smaller scales starting from a coarse-grained level of resolution.

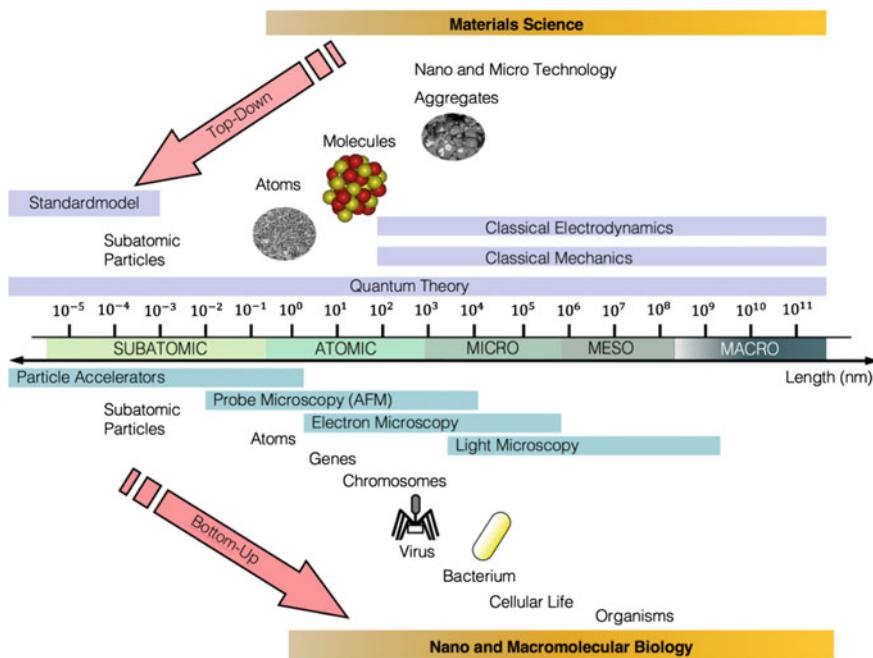


Fig. 2.1 Scope of application of different fundamental physical theories in life sciences (bottom) and in the areas of materials science and technology (top). In the latter area one uses a *top-down* approach by miniaturizing technical devices, whereas nature itself uses a *bottom-up* by directly manipulating atoms and molecules in biological systems. For subatomic particles, the *standard model* (cf. Sect. 2.5) is the accepted theory that explains all hitherto observed elementary particles in accelerator experiments. Today, it is generally believed, that *quantum theory* is the most fundamental theory which is in principle valid for the description of material behavior on all length scales, cf. the discussion in the introduction of Chap. 5. However, due to the numerical complexity of many particle systems treated on the basis of the Dirac or Schrödinger equation, *classical mechanics* (instead of quantum mechanics) and *classical electrodynamics* (instead of quantum electrodynamics – the simplest prototype of a quantum field theory – where the electromagnetic field is quantized itself) are useful approximative theories on length scales larger than $\sim 10 \text{ \AA}$. The classical theories however are not valid for quantum systems of atomic or subatomic dimensions, for phenomena occurring at speeds comparable to that of light (special relativistic mechanics) and they also fail for the description of large scale structures in the universe in strong gravitational fields (here the GTR is needed). The typical scopes of important experimental research methods using microscopes are also displayed to scale

2.2.1 Experimental Investigations on Different Length Scales

For large scale structures in the universe, experimental data are collected with telescopes, scanning a broad range of the electromagnetic spectrum. For small structures of fluids and solids in materials science, scattering techniques, such as Brillouin, neutron, Raman or electron scattering as well as different microscopy techniques – as

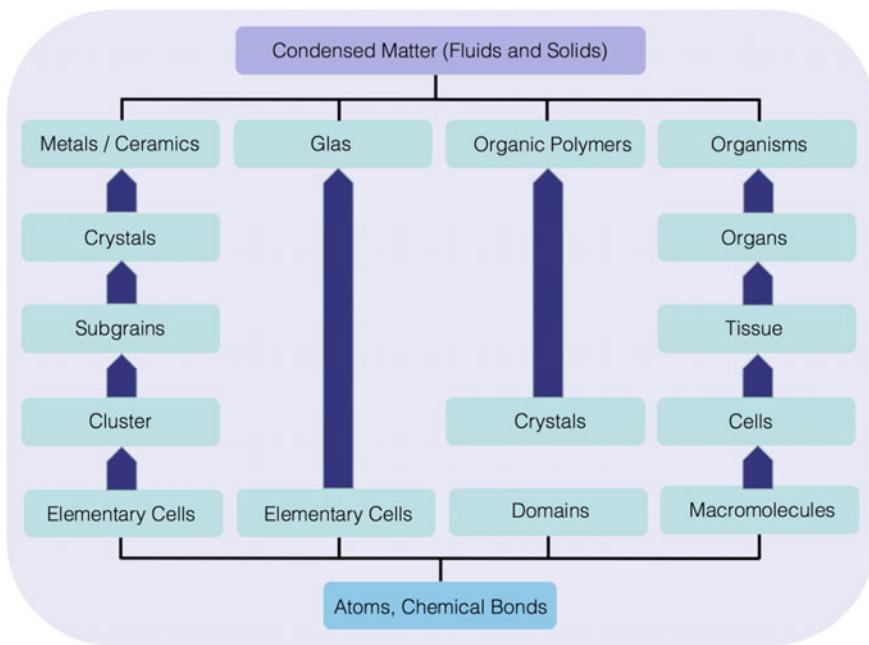


Fig. 2.2 Hierarchical view of structural properties of important classes of materials

depicted in Fig. 2.1 – are used on different length scales.⁶ The minimum separation d that can be resolved by any kind of a microscope is given by the following formula:

$$d = \lambda/(2n \sin\lambda), \quad (2.1)$$

where n is the refractive index⁷ and λ is the wavelength. The *resolution* of a microscope is the finest detail that can be distinguished in an image and is quite different from its *magnification*. For example, a photograph can be enlarged indefinitely using more powerful lenses, but the image will blur together and be unreadable. Therefore, increasing the magnification will not improve resolution. Since resolution and d are inversely proportional, and (2.1) suggests that the way to improve the resolution of a microscope is to use shorter wavelengths and media with larger indices of refraction.

The electron microscope exploits these principles by using the short *de Broglie wavelength* of accelerated electrons to form high-resolution images. The de Broglie wavelength of electrons (e^-) is given by

$$\lambda = \frac{2\pi}{k} = \frac{2\pi\hbar}{p} = \frac{h}{\sqrt{2m_e E}} \approx \sqrt{\frac{150}{V[Volt]}} [\text{\AA} = 10^{-10} \text{ m}], \quad (2.2)$$

⁶Also compare Fig. 7.23 on p. 349.

⁷The refractive index $n = 1$ in the vacuum of an electron microscope.

where k is the wave vector, p is the momentum and energy E is given in electronvolts, that is the acceleration voltage of the electrons.

An electron microscope is an instrument that uses electrons instead of light for the imaging of objects. In 1926, Hans Busch in Germany discovered that magnetic fields could act as lenses by causing electron beams to converge to a focus. A few years later, Max Knoll and Ernst Ruska made the first modern prototype of an electron microscope [84].

There are two types of electron microscopes: the Transmission (TEM) and the Scanning (SEM) (or Scanning Tunneling (STM)) Electron Microscope. In a TEM, a monochromatic beam of electrons is accelerated through a potential of 40–100 kilovolts (kV) and passed through a strong magnetic field that acts as a lens. With a TEM one can look at structures in solids and replicas of dead cells after fixation and sputtering with heavy metal, e.g. gold atoms. With this technique, electrons are reflected off the surface of the specimen. The resolution of a modern TEM is about 0.2 nm. This is the typical separation between two atoms in a solid. This resolution is 1,000 times greater than a light microscope and about 500,000 times greater than that of a human eye. The SEM is similar to the TEM except for the fact that it causes an electron beam to scan rapidly over the surface of the sample and yields an image of the topography of the surface. The resolution of a SEM is about 10 nm. The resolution is limited by the width of the exciting electron beam and by the interaction volume of electrons in a solid. As an example, in Fig. 2.3 an SEM picture of the granular surface structure and the fracture surface of Al_2O_3 are shown in different resolutions as indicated in Fig. 2.3.

Atomic Force Microscopy (AFM) is a form of scanning probe microscopy where a small probe is scanned across the sample to obtain information about the sample's surface. The information gathered from the probe's interaction with the surface can be as simple as physical topography or as diverse as the material's physical, magnetic,

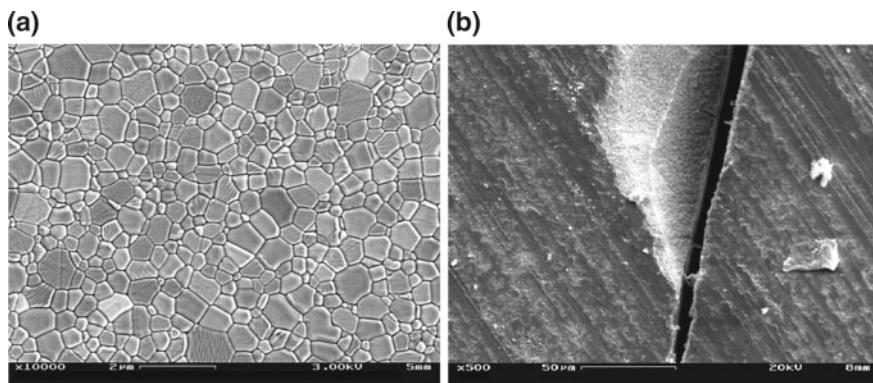


Fig. 2.3 **a** SEM micrograph section of an etched Al_2O_3 ceramic surface exhibiting the granular structure on the microscale. **b** Fracture surface of Al_2O_3 after an edge-on impact experiment (discussed in Chap. 7) at a speed of $\approx 400 \text{ m/s}$. Photos by Martin O. Steinhauser

or chemical properties. These data are collected as the probe is raster-scanned across the sample to form a map of the measured property relative to the X-Y position. Thus, a microscopic image showing the variation in the measured property, e.g. height or magnetic domains, is obtained for the area imaged.

Today, electron microscopy is widely used in physics, chemistry, biology, material science, metallurgy and many other technological fields. It has been an integral part in the understanding of the complexities of cellular structure, the fine structure of metals and crystalline materials as well as numerous other areas of the “microscopic world”.

2.3 What Is a Model?

The complexity of the world is obviously too large in order to be comprehended as a whole by limited human intellect. Thus, in science, a “trick” is used in order to still be able to derive some simple and basic laws and to develop a mental “picture” of the world. This trick consists in the *isolation* of a system from its surroundings in the first place, i.e. one restricts the investigated system to well-defined, controllable and reproducible conditions. These conditions are called *initial conditions*. After this preparation of a system, one performs experiments and investigates which states the system is able to attain in the course of time. The underlying assumption in this procedure is, that there are certain laws which determine in principle the temporal development of a system, once it has been prepared in an initial state and left to itself.

Usually, model building is lead by the conviction that there exists an “objective reality” around us, i.e. a reality which is independent of the individual observer who performs experiments and makes observations.⁸

The idea that there are fundamental laws of nature goes back to Greek philosophy, but until Galileo Galilei (1564–1642), there were only very few experimentally verifiable consequences of this idea. To identify fundamental laws, the considered system has to be isolated from its particular surrounding. Take as an example the ballistic, parabolic curve of a kicked football on Earth. The fact that an object which is thrown away on Earth follows a parabolic path *is* a law of nature; however, it is not a *fundamental* law. One realizes the fundamental law when abstracting from the Earth’s atmosphere and then, in a next step, completely abstracting from the special viewpoint on Earth. When throwing away a ball in empty space, far enough away from any gravitating sources, there is no gravity which will force it on a ballistic curve,⁹ that is, it will simple follow a straight line (a *geodesic*, to be exact). This finding is due to Galilei and is consequently called Galilei’s law of inertia. Hence, the fact, that objects thrown on Earth describe ballistic curves (i.e. not straight lines)

⁸Solipsism in this context is not a scientific category, as it renders all rational discussions useless and impossible.

⁹Of course, this is only true, if the object has a velocity component parallel to the surface of Earth.

is only due to the particular circumstances, the special point of view of the observer on Earth, or physically spoken, his frame of reference, which is not the simplest possible one. In the simplest possible frame of reference – a freely falling reference frame – which is “isolated” from both, air resistance and gravity, the ball will follow a simpler path, that is a straight line. Systems in which moving objects that are not subject of any external forces – e.g. due to gravity or electromagnetic fields – follow a straight line, are called *inertial systems*. Mathematically spoken, the inertial systems build an *equivalence class* of an infinite number of systems. The inertial systems are also those frames of reference in which Newton’s mechanics and the special theory of relativity are valid.

One remarkable thing about fundamental laws of physics is, that they are *deterministic* and *time-reversible*,¹⁰ or time-symmetric (symplectic), i.e. in the example of the flying ball above, each point on the ballistic curve together with its velocity can be viewed as initial state of the system which then – due to the laws of nature – completely determines the future behavior in a *classical sense*. From a fundamental point of view, it is thus amazing that processes in nature seem to occur only in a certain direction such that they can be distinguished in “past” and “future” events.¹¹ One could say, *because* the fundamental laws of nature do not distinguish any direction of time, in closed systems, eventually all processes die out which are not time reversible, i.e. the system approaches its thermal *equilibrium state*. In equilibrium, no process occurs any more, except remaining time-reversible fluctuations about the equilibrium state which have equal probability. This tendency of the laws of nature is exploited for example in the computational methods of Molecular Dynamics and Monte Carlo simulations, discussed in Chap. 6.

2.3.1 The Scientific Method

The roots of the scientific method as an interplay between theory (models or systems) and experiment, practiced in the natural sciences today, lie in the philosophy of Plato (427–347) and Aristotle (384–322). An important fundamental idea of Plato is to consider all observable things only as incomplete pictures or reflections of an ideal mathematical world of ideal forms or ideas which he illustrated in his famous *Allegory of the Cave* at the beginning of book 7 of *Republic*, see e.g. [85]. Aristotle however radically discarded Plato’s dual concept of on the one hand, ideal forms, and on the other hand, perceivable phenomena. According to him only the phenomena themselves could be considered as true sources of knowledge about nature. In a process of *generalizing induction*, see Fig. 2.4, explanatory principles are attained. Using general propositions which include those principles, statements on the

¹⁰Time-reversibility (or time-symmetry) is actually broken in certain rare elementary particle physics processes which was shown in 1964 by J.L. Cronin and V.L. Fitch (Nobel Prize 1980) at CERN.

¹¹For example, it has never been observed, that the pieces of a broken cup cool off and repair themselves, although this process is not forbidden by a fundamental law of nature.

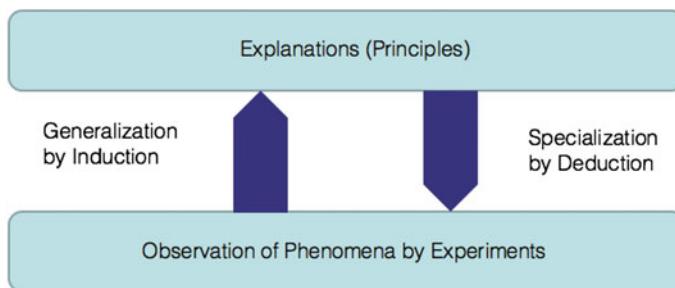


Fig. 2.4 Aristotle's inductive-deductive principle

phenomena are formulated by pure reasoning (deduction). An important point in the Aristotelian method is the fact that his way of modeling does not try to explain things by reduction to higher-ranking mathematical structures (as Plato did). From a modern point of view however, it is astonishing, that Aristotle did not systematically introduce or “invent” the *experiment* as a way to decide about the usefulness and applicability of the explanations of phenomena. This crucial step of artificially *isolating and idealizing* a system from its complex environment was achieved by Galilei [86], which he describes in his Dialogue of 1638, cf. Fig. 2.5a.

He realized the epistemological importance of the experiment which confronts the assumed explanatory principles with the phenomena and thus provides a means of testing and falsifying mathematically formulated hypotheses about nature, cf. Fig. 2.6.

Isaac Newton finally introduced in his Principia, cf. Fig. 2.5b, the *axiomatic method* into the natural sciences, where the formulation of mathematical principles in terms of axioms is achieved in a process of intuitive generalization. He was the first one who had the idea that the mechanical behavior of the world might work like a clock that is set; with this idea Newton introduced the crucial partition of the world into *initial conditions* on the one hand, and *laws of nature* on the other hand. In this context Einstein writes in a letter to Maurice Solovine:

I see the matter schematically in this way:

- (1) The E's (immediate experiences) are our data.
- (2) The axioms from which we draw our conclusions are indicated by A. Psychologically the A's depend on the E's. But there is no logical route leading from the E's to the A's, but only an intuitive connection (psychological), which is always “re-turning”.
- (3) Logically, specific statements S, S', S'' are deduced from A; these statements can lay claim to exactness.
- (4) The A's are connected to the E's (verification through experience). [...] But the relation between S's and E's is (pragmatically) much less certain than the relation between the A's and the E's. If such a relationship could not be set up with a high degree of certainty (though it may be beyond the reach of logic), logical machinery would have no value in the “comprehension of reality”. [accentuations by Einstein]

(Albert Einstein [87], 1952, pp. 137)

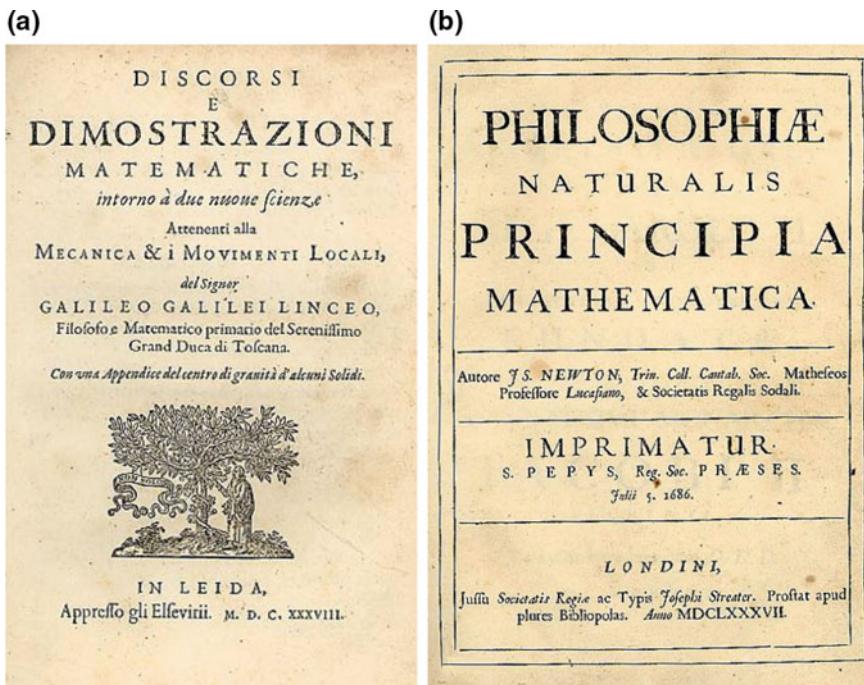


Fig. 2.5 Title pages of the first editions of Galilei's Discorsi **a** and Newton's Principia **b**. Photos republished with permission. From the original in the Rare Books and special Collections, University of Sydney Library

Thus, according to Einstein, it is impossible to find a strict logical connection between the observed phenomena and the system of axioms by pure induction. Rather, one has to find general principles by *intuition*, using heuristic principles, e.g. symmetry, logical and mathematical simplicity, or keeping the number of independent logical assumptions in a theory as small as possible, cf. [88, 89].

With the axiomatic method one tries to formulate theories on two levels. The first level is the one that states fundamental theorems, principles and the axioms themselves, which summarize experimental facts, e.g. Newton's axioms or Maxwell's equations which are “true” or “real” in the sense that they are statements on the behavior of nature. The second level is the level of the theory itself and its interpretation. On this level one may introduce objects and terms such as “potential”, “point particle”, “wave function”, “atom”, “quark”, “field”, etc. which are used to derive and predict properties and states of systems based on the axioms. One might wonder why it is necessary to *interpret* a physical theory when it already has been formulated in axiomatic form. The reason for this need is, that a theory, in order to explain phenomena, needs to contain abstract concepts as elements which go beyond a mere phenomenological description. However, it is not a priori clear to what extend these concepts represent anything in “reality” and can thus be given an explicit meaning.

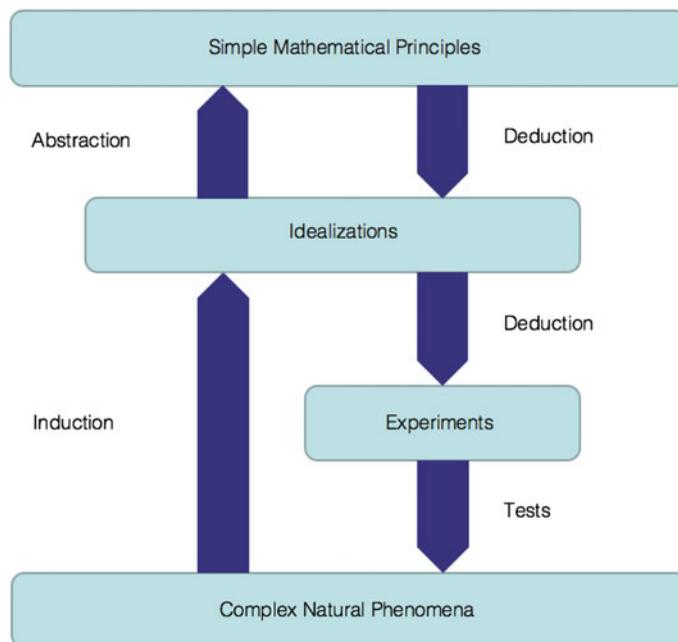


Fig. 2.6 Galilei's method of using experiments to test idealizations of theories which in turn are based on abstract mathematical principles

For example, Newton was able to add the notions of an absolute space and time to his theory, which are not defined within his system and which are not used in the formulation of Newton's axioms, i.e. the basic theorems on the first level. Thus, such terms do not really change any consequences or predictions of the theory which could be tested in experiments. Ernst Mach was one of the most prominent critics of such redundancies in Newton's theory which he expresses in his classic "Die Mechanik in ihrer Entwicklung – Historisch kritisch dargestellt" [90]. The idea to remove and to avoid redundant elements in theories (sometimes called *Occam's razor*¹²) can be a very useful heuristic method in the attempts to falsify or improve a theory.

A naïve and primitive notion of "existence" and "reality" is connected with something that can be seen and "perceived" with one's own senses or at least "detected" with some instruments. In modern science, *every* element of a theory which leads to consequences that can be tested and falsified is considered to be "as real as a chair".¹³ In this sense, it is theory that determines what "reality" is. An example is

¹² After the 14th century monk William of Occam who often used the *principle of unnecessary plurality* of medieval philosophy in his writings, such that his name eventually became connected to it.

¹³ See e.g. the comments on p. 54 of Steven Weinberg's book "Dreams of a final theory" [91].

the assumption (or hypothesis) of the existence of quarks, the constituents of neutrons and protons in the atomic nuclei. Teller writes about “this philosopher problem” [92]:

I take it that the “philosopher problem” refers to the attitude crudely summarized by saying that “if we can’t see it, can’t see it under any circumstances, then it isn’t real”. [...] When we see chairs, tables, and any sort of middle-sized macroscopic objects, we do so only via the good offices of a flood of photons, massaged by a lot of optics, interaction with the neurons in the retina, data manipulation in the optic nerve and further cortical processing, until whatever neural processes that ensue count as perception. Now, given that our perception of ordinary chairs and tables is this indirect, what grounds could we have for denying reality (in whatever sense chairs and tables are real) to something we see only slightly more indirectly? [...] In whatever sense you think chairs and tables are real, and once you appreciate the indirectness of our perception of these things, the greater indirectness of seeing smaller things is not going to be an in-principle reason for thinking the smaller things are not real in the same sense. Of course, as the chain involved in indirect perception gets longer, the chance of error may increase; the chances that we have been fooled about quarks may well be larger than the [...] chances that we have been fooled about chairs and tables. But the *kind* of reason we have for thinking that quarks are real differs in degree, not in kind, from the *kind* of reason we have for thinking that chairs and tables are real, always with the same sense of the word “real”.
 [accentuations by Teller]

(Edward Teller [92], 1997, p. 635)

Despite the fact that it is impossible to detect and virtually “see” quarks separately as individual particles, due to the properties of the interactions between them, these particles are considered to be “real”, because their existence helps to establish a consistent theory – the standard model (cf. Sect. 2.5) – which explains all hitherto observed interactions, gravitation excepted.

Pais writes in “*Einstein lived here*” [93] in Chap. 10 that Einstein often distinguished between two kinds of theories in physics: *theories based on principles* and *constructive theories*. This idea was first formulated in a brief article by Einstein in the *Times of London* in 1919 [94]. With a constructive theory, one tries to describe complex observations in relatively simple formalisms, usually based on hypothetical axioms; an example would be kinetic gas theory. With a principle theory, the starting point is not hypothetical axioms, but a set of well-confirmed, empirically found generalized properties – so-called *principles* – of nature; examples include the first and second law of thermodynamics. Ultimate understanding requires a constructive theory, but often, according to Einstein, progress in theory is impeded by premature attempts at developing constructive theories in the absence of sufficient constraints by means of which to narrow the range of possibilities. It is the function of principle theories to provide such constraint, and progress is often best achieved by focusing first on the establishment of such principles. In the following, some important general principles are listed that can be found in textbooks on theoretical physics, and which are often used as heuristic guidelines in the development of models in materials science, and for their numerical counterparts.

- Hamilton's principle (Principle of minimization of the integral of action).
- Principle of minimal potential energy (Dirichlet's variational principle).
- Fermat's Principle (Principle of the shortest path of light).
- Principle of virtual work by d'Alembert.
- Ritz' variational principle.
- Galilei's principle of relativity.
- Principle of special relativity.
- Principle of general relativity.
- Principle of general covariance.
- Symmetries of groups and group operations.
- Energy-momentum conservation.
- Angular-momentum conservation.

In this list we haven't mentioned some principles which are applied only on the level of elementary particle physics, such as the principle of charge and parity (CP)-invariance (which is only violated in the decay of the neutral K-meson) or the principle of charge, parity, and time (CPT)-invariance which is assumed to be valid for *all* known fundamental interactions.

As discussed above, there is no general learnable way of guessing or finding physical laws; rather, for want of a logical path often intuition and (sometimes even unconscious) ad-hoc hypotheses finally lead to success.¹⁴ A law of nature that has been found or a theory that has been formulated is just a guess which is then put to the (experimental) test. Some common key features with the development of any model are:

- Simplifying assumptions must be made.
- The number of logically independent elements and heuristic theorems which are not derived from basic notions (axioms) should be as small as possible.
- Boundary conditions or initial conditions must be identified.
- The range of applicability of the model should be understood.

It is important to realize, that a theory or a model can only convincingly explain phenomena if it contains abstract concepts as elements which go beyond mere observation. An example for this is Maxwell's displacement current $\partial\mathbf{E}/\partial t$ (in appropriate units), which he added to Ampère's original law based purely on theoretical considerations, which cannot be obtained from observation.

Example 1 (Maxwell's and Einstein's Field Equations) There is an interesting analogy between Maxwell's development of the field equations of electromagnetism in 1865 [95] and Einstein's development of the field equations of general relativity half

¹⁴ Albert Einstein's quest for a formulation of general relativity during the years 1907–1915 is the classic example, cf. Chap. 3 on p. 109.

a century later. Beginning in the 1850s, Maxwell elaborated ideas of Faraday to give a complete account of electrodynamics based on the concept of continuous fields of force (in contrast to forces acting at a distance). In modern terminology he arrived at the first gauge theory of physics, using the following four partial differential equations which encode observational facts for the electric and magnetic fields $\mathbf{E}(\mathbf{x}, t)$, $\mathbf{B}(\mathbf{x}, t)$ and their corresponding sources, charge density and current density $\rho(\mathbf{x}, t)$, $\mathbf{j}(\mathbf{x}, t)$, directly derived from experiment:

$$\nabla \cdot \mathbf{E} = 4\pi\rho, \quad (2.3a)$$

$$\nabla \times \mathbf{B} = \frac{4\pi}{c}\mathbf{j} + \frac{1}{c}\frac{\partial \mathbf{E}}{\partial t}, \quad (2.3b)$$

$$\nabla \times \mathbf{E} = -\frac{1}{c}\frac{\partial \mathbf{B}}{\partial t}, \quad (2.3c)$$

$$\nabla \cdot \mathbf{B} = 0. \quad (2.3d)$$

Maxwell added the extra term $\frac{1}{c}\frac{\partial \mathbf{E}}{\partial t}$, which was *not* obtained from experiment, to Ampère's law in (2.3b), such that the continuity equation $\nabla \cdot \mathbf{j} = -\partial \rho / \partial t$ is also fulfilled in the case of time dependent fields. The continuity equation then follows from (2.3a) and (2.3b). Additionally, the inclusion of the displacement current leads to transverse electromagnetic waves propagating in a vacuum at the speed of light. Thus, the combination of charge conservation and Coulomb's law implies that the divergence of equation (2.3b) vanishes and renders the set of equations mathematically consistent.

With a similar consideration Einstein arrived at the final field equations of general relativity; the simplest hypothesis involving only the metric coefficients g_{mn} and their first derivatives, is that the Ricci tensor R_{mn} equals the stress energy tensor T_{mn} , (see Chap. 3 for a proper introduction to tensors). It turns out however, that the divergence of R_{mn} does not vanish as it should in order to satisfy local conservation of mass-energy. However, the tensor $R_{mn} - 1/2 g_{mn}R$ does have vanishing divergence due to Bianchi's identity.¹⁵ Thus, when including the additional trace term $-1/2 g_{mn}R$ one yields the complete and mathematically consistent field equations of general relativity:

$$R_{mn} - \frac{1}{2}g_{mn}R = -\frac{8\pi G_N}{c^4}T_{mn}. \quad (2.4)$$

Einstein commented on this in a letter to Michele Besso in 1918 in which he was chiding Besso for having suggested (in a previous letter) that, in view of Einstein's theory of relativity, "speculation has proved itself superior to empiricism". Einstein disavowed this suggestion, pointing out the empirical base for all the important developments in theoretical physics, including the special and general theory of relativity. He concludes:

¹⁵ $R_{ijkl|m} + R_{ijlm|k} + R_{ijmk|l} = 0$, see e.g. [96].

No genuinely useful and profound theory has ever really been found purely speculatively. The closest case would be Maxwell's hypothesis for displacement current. But there it involved accounting for the fact of the propagation of light (& open circuits).

(Albert Einstein [87], 1918, p. 524)

The question to what extend a physical theory maps a small part of reality has been answered differently at different times, but one can distinguish at least three different convictions or “paradigms”:

1. *Phenomenological*: one seeks an economic description of sensory perceptions and defines them as “reality”.
2. *Operational*: one seeks instructions, according to which the descriptive elements of the theory can be measured, and defines them as “reality”.
3. *Realistic*: one seeks abstract principles and theorems that go beyond a mere description of observations and defines “reality” by all those elements of the theory the consequences of which can be falsified by experimental tests.

The phenomenological point of view interprets theories as an instrument of describing observations in an economic way; this attitude could be described as positivism of Machian character. The Copenhagen interpretation of quantum theory, cf. Chap. 5, is based on an operational interpretation of a physical theory. Here, everything that can be measured is considered to be “real”. A realistic interpretation of theories assumes that the used notions in a theory go beyond a mere description of observations and all elements or objects that are introduced in the theory are considered to be “real” if they lead to any consequences that can be tested in experiment. For example, in the current theory of elementary particles – the standard model – symmetry principles play a fundamental role; they build the basic mathematical ontology of physics. In this context Weinberg writes in “*The rise of the standard model*” [92]:

The history of science is usually told in terms of experiments and theories and their interaction. But there is a deeper level to the story – a slow change in the attitudes that define what we take as plausible and implausible in scientific theories. Just as our theories are the product of experience with many experiments, our attitudes are the product of experience with many theories. [...] The rise of the standard model was accompanied by profound changes in our attitudes toward symmetries and field theory.

(Steven Weinberg [92], 1997, p. 36)

2.4 Hierarchical Modeling Concepts Above the Atomic Scale

The equations of fundamental theories such as quantum theory become too complex when applied to macroscopic systems, which involve an astronomical number of constituents. Thus, various approximative theories have been devised, which lead to equations that can be solved at least numerically. Each theory has its range of

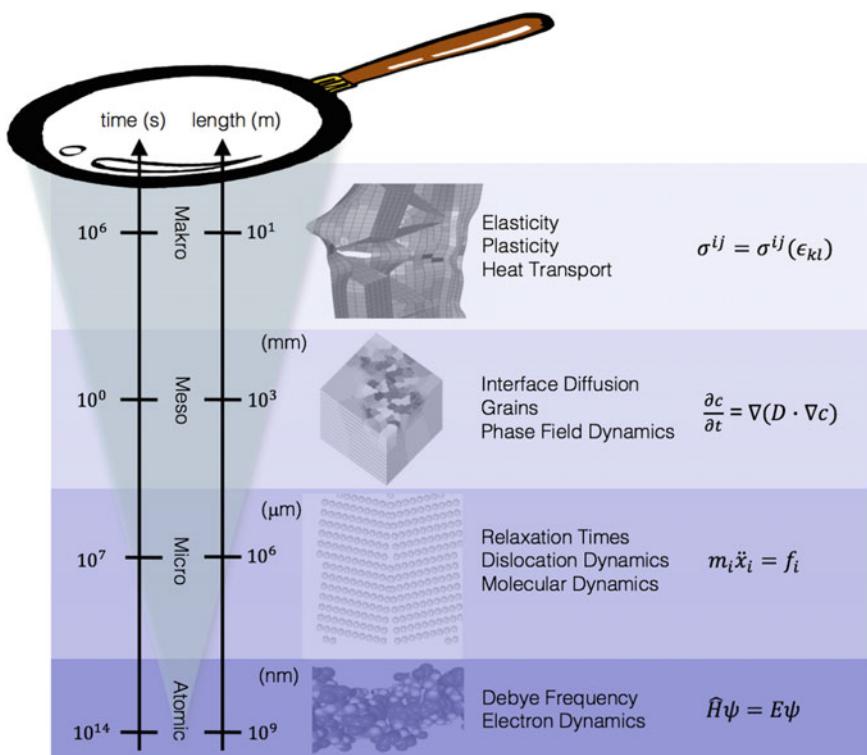


Fig. 2.7 The structural hierarchy of length scales in nature is also reflected in fundamental physical theories based on fields and particles, which have different scopes of validity. Depicted are some typical physical phenomena on different length scales and some basic equations of physical theories used on the respective scale, e.g. the Schrödinger equation at the atomic scale, classical Newtonian particles dynamics, the diffusion equation based on the concept of fields, or the constitutive equation for large scale elasticity and plasticity, connecting stress (force per unit area) and strain in a body, modeled as a continuum with an infinite number of degrees of freedom

applicability and this is the main reason why there are so many different computational methods that are used in engineering and materials science, cf. Fig. 2.7.

Classical Newtonian mechanics is a scientific system which is only approximately valid for the description of the dynamics of condensed matter at small velocities and for lengths larger than $\sim 10^{-10}$ m. For smaller distances, classical mechanics has to be replaced by a different system – quantum theory, or sometimes called quantum mechanics, when referring specifically to the description of the motion of particles with mass. For large velocities, special-relativistic mechanics, where space and time are united in spacetime, has to be used, see Sect. 3.6.1 in Chap. 3 on p. 109.

2.4.1 Example: Principle Model Hierarchies in Classical Mechanics

Classical theoretical mechanics is an appropriate theory for the description of phenomena occurring between objects above the spacial dimensions of atoms ($\sim 1\text{\AA} = 10^{-10}\text{ m}$) and for small velocities. For the description of the dynamics of objects of atomic dimensions, classical physics breaks down and instead quantum theory is a more appropriate model. Today, it is believed that quantum theory is *the* fundamental theory, underlying all subatomic, atomic, microscopic, mesoscopic and macroscopic objects and natural phenomena. For a discussion and references, see Chap. 5.

There are different layers or hierarchies of models in theoretical physics. These could be grouped according to the number of general natural phenomena which can be explained by the respective model system. In *classical mechanics* for example there are different abstraction layers, or hierarchies of the mechanical systems, which we shortly discuss in the following.

Mechanics of Mass Points

The possibly simplest of all model systems is the model of a mass point which is based on the notion of a dimensionless object, i.e. an imaginary object which has no extension in space but contains all mass m of the considered system. This is equivalent with the notion of a “point” in mathematics which can be identified by assigning some numbers to it in an appropriately chosen coordinate system. The motion of mass points is then described within the classical Newtonian system. Whether this approximation is useful depends on the considered system (e.g. the motion of Earth around the Sun).

Mechanics of Point Systems

If there are many mass points one speaks of a *point system*, or a many N –particle system. By deriving equations of motion for point systems, some general principles, so called *integral principles* such as d'Alembert's or Hamilton's principle can be derived. The advantage of general principles in the derivation of equations of motion (EOM) is that there is no preferred coordinate system. Thus, using variational calculus one can obtain EOM in general coordinates. Due to the large number of mass points in a macroscopic body of order $\mathcal{O}(10^{23})$, and the just as many ordinary differential equations, a general solution is only possible when symmetries are present.¹⁶

Mechanics of Rigid Bodies

A rigid body is a special system of mass points in which one can neglect the relative motion of the points with respect to each other. Real systems fulfill this condition only approximately. However, many bodies – under normal conditions – change their shape and volume only marginally. A rigid body has $6N$ degrees of freedom (3 translations and 3 rotations of N particles); thus, for such systems one obtains $6N$ differential equations of 2nd order.

¹⁶In fact, the N -body problem is analytically unsolvable for $N \geq 3$, cf. Example 37 on p. 258.

Mechanics of Continua

In continuum mechanics one only considers motions of bodies, in which neighboring mass points (e.g. atoms) approximately move into the same direction. In this case one can approximate the whole body as a continuum with an infinite number of degrees of freedom, cf. the discussion in Sect. 7.7 on p. 341. One advantage of this approximation is that the number of equations is reduced drastically; instead of having to solve many ordinary differential equations, one has to solve only a few but *partial* differential equations. Usually one distinguishes *elastomechanics* which treats deformable solid media (solid states), where the constituents are still bound strongly together such that small deviations lead to strong forces (inner tensions), and *hydrodynamics* which is applicable to gases and fluids where the constituents are bound only weakly together (see Chap. 7).

It can be useful to employ different models for the same system. For example, when considering the revolution of Earth around the Sun, it is useful to model the Earth (and the Sun) as a mass point which contains all mass. However, when considering its rotary motion it is useful to use the model of an extended stiff body. However, both of these models will fail when trying to describe tidal effects due to movements of matter within the earth's crust. Here one has to use continuum theoretical concepts. Which model is the best in which situation is not an easy task to decide and no algorithm can help a scientist here in making a useful decision. Sometimes there are some empirical rules (so called *heuristics*, i.e. concepts which have proved successful in the past even if these concepts are rather empirical) which can be used as a guide. In the end, it is the experience and intuition of the scientist which lead to the use of a certain model in some situation.

2.4.2 Structure-Property Paradigm

In experimental materials science today one usually breaks a system (e.g. a material such as a metal specimen or a ceramic plate) into smaller pieces and investigates the properties of the obtained smaller structures with the aid of microscopes. The idea, or conviction behind this procedure is based on our hierarchical view of the structure of matter and the believe that if one can understand the mechanics of a small subsystem, one can also understand the whole system. One assumes as a working hypothesis that the observed macroscopic phenomenological properties of a material (e.g. its plastic or elastic reaction to an external load) are determined *in principle* by the properties of its micro-, meso-, and nanostructure. This assumption is called the *structure-property paradigm* of materials science, cf. Table 2.1.

2.4.3 Physical and Mathematical Modeling

As presented in Fig. 2.1 on p. 34, materials science covers roughly 12 orders of magnitude in size, along with the associated time scales of physical processes. While

Table 2.1 Illustration of the structure-property paradigm of materials science. Molecular structural properties of systems determine their mesoscopic structures and ultimately the observed macroscopic properties of solids and fluids

Structure			Property
Molecular			Macroscopic
Nano	Micro	Meso	Macro
Electronic structure	Molecule size	Volume ratio	Viscosity
Inter-atomic interaction	Molecular weight	Packing density	Strength
Bond angles	Fiber/matrix interaction	Fiber orientation	Toughness
Bond strength	Grain size distribution	Flexibility	Modulus
Bond failure	Cross-link density	Dispersion	Stress/strain
Chemical sequence	Defects	Heat transport	Plasticity
Unit cell	Crystallinity	Grain orientation	Durability

atomistic methods based on quantum mechanics prevail in detailed microstructure simulations at the nanoscale, as well as coarse-grained atomistic simulations neglecting the electrons at the microscale, such a detailed numerical treatment of systems cannot be done at the meso- and macroscale. Here, one has to average out the many atomic degrees of freedom and use “superatoms” which represent large clusters of atoms. With this approach, systems on the meso- and macroscale can still be treated with a classical particle approach solving Newton’s equations of motion, see e.g. [62, 97]. The physical and mathematical part of model building often go hand in hand, as physical ideas on the materials behavior are usually formulated using mathematical concepts and (differential) equations.

State Variables and Equations of State for Micro- and Mesostructural Evolution

When investigating fluids or solids using continuum based microstructural simulations, one is usually not interested in the dynamics, i.e. positions and momenta of single fluid particles or superatoms in the solid; rather, one is interested in the average behavior of the system’s macroscopic state variables such as temperature T , pressure p , density ρ , displacement u^i , stress σ^{ij} and strain ϵ^{ij} , or free energy F in Ginzburg-Landau type models, etc., which are described as continuum variables. In thermodynamics, *extensive* state variables such as entropy S , volume V , or energy E of a heterogeneous system are additive with respect to the different phases of a system, i.e. they are proportional to the amount of substance, e.g. mass m or number of particles N . In contrast, *intensive* state variables are independent of the amount of substance and may assume different values in different phases of a system. Examples are refraction index, p , ρ , or T , which may be defined *locally*, that is they are parameterized fields with an infinite number of degrees of freedom within the framework of classical field theories. Usually time and position are used for parameterizing field properties, i.e. the state variables are functions of position and time. To determine

the spacial dependency of intensive variables, one needs additional conditional equations, for example from hydrodynamics or in the form of other phenomenological equations of state. Examples for equations of state are Hooke's law in dislocation dynamics, nonlinear elasticity laws in polymer dynamics, or the free energy functional in Ginzburg-Landau type microstructural phase field models. The question as to how many state variables are needed to completely characterize a closed system at equilibrium is answered by Gibb's phase rule (see Problem 6.2 on p. 312):

Definition 2.1 (*Gibb's phase rule*) The number of degrees of freedom f needed to fully characterize a closed system at equilibrium is given by

$$f = C + 2 - P, \quad (2.5)$$

where C is the number of chemical components, P is the number of phases and f labels the degrees of freedom.

Phenomenological descriptions based on thermodynamic equations of state and continuum mechanics are prevailing in typical engineering applications on the meso- and macroscale, for example in the prediction of material behavior of composite materials such as concrete, metal-matrix composites, polymer fiber-reinforced composites or multi-phase ceramics under various load or processing conditions. A great disadvantage of detailed phenomenological descriptions taking into account structural features observed on the meso- and microscale, is the often large number of state variables that is required. Such an approach, involving many variables, may quickly degrade a transparent physical model based on few assumptions to a mere empirical polynomial fit, where the state variables just serve as fitting parameters. Too great a number of parameters often reduces the value and the physical significance of a model considerably. In many practical engineering approaches in industry, a many-variable fitting approach may be helpful to gradually optimize materials in certain manufacturing processes, too complicated for an explicit description, however, it is less desirable in *physically* oriented simulations taking into account micro- and mesostructures.

State variables in mesoscale simulations are usually weighted with certain additional, empirical parameters. Often, these mesoscopic parameters are nonlinearly coupled with other equations of state which renders such models numerically rather complex. By definition, state variables are defined for systems at thermal equilibrium. A system like a polycrystalline microstructure however, is generally not at equilibrium and the systems evolution may occur irreversible; hence, the system's evolution equations are generally path dependent, i.e. they are no total differentials which can be readily integrated. This difficulty gave rise to the increased application of statistical models in recent years, often based on the Monte Carlo Method [98] (cf. Sect. 6.6). MC methods have been used e.g. for the simulation of diffusion behavior or short range ordering [99, 100], recrystallization [101], grain growth [102, 103], or boundary misorientations [104, 105]. Among the most successful mesoscale models for microstructural evolution simulations are vertex models, phase field models [106], cellular automata [107, 108] and Potts models [109]. In the beginning

1980s it was realized that Potts domain structures are similar to granular microstructures. As both systems are characterized by a space-filling array of domains which evolve to minimize the boundary area, the Potts model was used for a variety of simulations such as late-stage sintering [110], or grain growth in polycrystals in 2 dimensions (2D) [111–114], and in 3D [115]. As a result of the above-mentioned modeling approaches for microstructure dynamics, one obtains a set of governing equations, which model the microstructural elements of the considered solid by means of state variables, that are functions of position, time or other parameters, such as the dislocation density or grain curvature.

Kinematic Equations

The mere description of the motion (changes of position) of objects in classical mechanics without reference to their origin, that is forces, is called *kinematics*. Kinematic equations allow for the calculation of certain mechanical properties which are based on coordinates and its derivatives with respect to time, velocity and acceleration, e.g. strains, strain rates, crystal orientations, or rigid body spin¹⁷ to name but a few. It is very important to understand that positions are assigned with reference to arbitrarily chosen coordinate systems. Therefore, positions, that is, coordinates have no intrinsic physical (i.e. metrical) meaning. They are simply numbers that are used to label events in spacetime which change when a different coordinate system is used which may be linearly shifted or rotated with respect to the original one.

Spacetime is a unification of space and time into one single (flat) manifold, called *Minkowski space*¹⁸ which simplifies a large amount of physical theory; in particular, it is the underlying structure to be used for the description of events introduced in the special and general theory of relativity (for a formal introduction see Sects. 3.6.1 and 3.6.2 in Chap. 3).

Example 2 (Special Relativistic Kinematics and Lorentz Transformations) An event in spacetime is a point x^μ , ($\mu = 1, 2, 3, 4$) specified by its time and three spacial coordinates, i.e. $x^\mu = (ct, \mathbf{x})$ which are called *Minkowski coordinates*

$$x^0 = ct, \quad x^1 = x, \quad x^2 = y, \quad x^3 = z. \quad (2.6)$$

The quantity denoted by x^μ is often called *four-vector* (instead of “components of a four-vector” $\mathbf{x} = x^\alpha \mathbf{e}_\alpha$ with orthonormal basis $\mathbf{e}_\alpha \mathbf{e}^\beta = \delta_\alpha^\beta = \delta_{\alpha\beta}$). The transformation between contra- (upper) and covariant (lower) components is achieved by the metric tensor of Minkowski space, that is, the Minkowski metric $\eta_{\mu\nu}$, cf. (3.16) on p. 138:

¹⁷In continuum theory this is the antisymmetric part of the tensor of the displacement derivatives.

¹⁸After the mathematician Hermann Minkowski, mathematics professor at the Polytechnikum ETH in Zurich and teacher of Einstein, who introduced this unifying concept in a famous lecture at Cologne in 1908, see pp. 54–71 in [116].

$$x_\mu = \sum_{\nu=1}^4 \eta_{\mu\nu} x^\nu = (ct, -\mathbf{x}). \quad (2.7)$$

Using a Minkowski metric, the infinitesimal distance ds^2 between two events (two points in Minkowski space) is given by $ds^2 = c^2 dt^2 - d\mathbf{x}^2$. The *worldline* of an object, e.g. a particle, is the path that this particle takes in spacetime and represents its total history. In a sense, physical reality is given by all the events described in spacetime. Applying the special principle of relativity (see Sect. 3.6.1 on p. 163), one obtains for two different inertial frames of reference Σ and Σ' , cf. Fig. 2.8, the proper transformation laws of space time coordinates, which are the linear *Lorentz transformations*:

$$x'^\alpha = \Lambda_\beta^\alpha x^\beta + a^\alpha, \quad (2.8)$$

where a^α represents a time and space translation, and $\Lambda_\beta^\alpha \in SO(3)$. The group of Lorentz-transformations is called Poincaré group and contains (just like the Galilei group) 10 parameters. The translations and rotations build a subgroup of both, the Galilei group and the Lorentz group. In this subgroup one normally excludes a

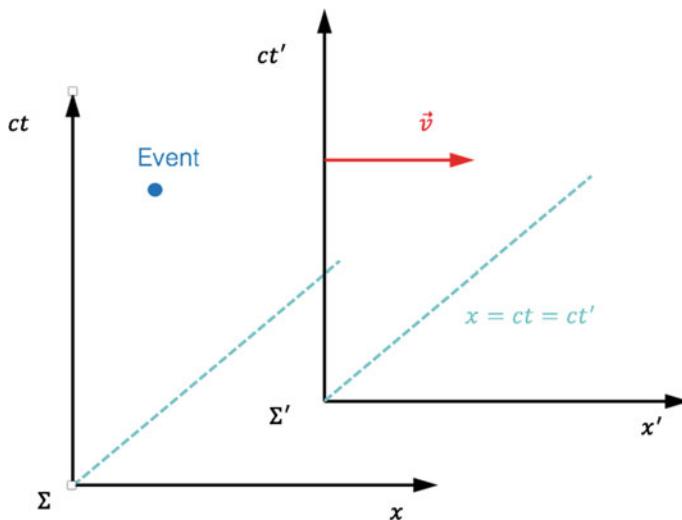


Fig. 2.8 A spacetime diagram in Minkowski space. Two coordinate systems Σ and Σ' move with velocity v relative to each other. This configuration is called *standard configuration* in the special theory of relativity. For simplicity, only two coordinates, x , and ct (time expressed as distance traveled by light in time interval t , with the conversion factor c (velocity of light) are shown. A light signal (*dotted line*), e.g. a flash travels at speed $v = c$, thus, its worldline is represented as a bisecting line in both coordinate systems. Let the coordinates of a certain event in Σ be (t, x) . In Σ' , which moves relative to Σ with velocity v in positive x -direction, the very same event is described by coordinates (t', x')

change of handedness, i.e. $\text{Det}(\Lambda) = +1$. In Minkowski space, the scalar products of four-vectors are invariant under Lorentz-transformations, i.e. $\Lambda_\alpha^\beta \Lambda_\beta^\gamma = \delta_\alpha^\gamma$. For example, for two vectors $A^\mu = (A^0, \mathbf{A})$ and $B^\mu = (B^0, \mathbf{B})$ the tensor product¹⁹ $A \cdot B = A_\mu B^\mu = A^\mu B_\mu = A^0 B^0 - \mathbf{A} \cdot \mathbf{B}$. The Lorentz-transformations are distinguished in that they leave the *proper time interval*

$$d\tau^2 = c^2 dt^2 - (dx)^2 = \sum_{\mu=0}^3 dx^\mu dx_\mu . \quad (2.9)$$

invariant. We show this property in the following example and in Problem 1. The proper time $d\tau$ is a Lorentz invariant and in the rest system of an observer it coincides with the coordinate time. Taking the derivative of the position four-vector with respect to τ is also a four-vector, the four-velocity u^μ .

$$u^\mu = \frac{dx^\mu}{d\tau} = \left(\frac{cdt}{d\tau}, \frac{d\mathbf{x}}{d\tau} \right) = \left(\frac{cdt}{d\tau}, \frac{d\mathbf{x}}{dt} \frac{dt}{d\tau} \right) = \frac{dt}{d\tau}(c, \mathbf{v}) . \quad (2.10)$$

To calculate $\frac{dt}{d\tau}$ we go back to (2.9) which can be written as:

$$(d\tau)^2 = (dt)^2 \left(1 - \frac{1}{c^2} \left(\frac{d\mathbf{x}}{dt} \right)^2 \right) = (dt)^2 \left(1 - \left(\frac{\mathbf{v}}{c} \right)^2 \right) . \quad (2.11)$$

Hence,

$$\frac{dt}{d\tau} = \frac{1}{\sqrt{1 - (\frac{\mathbf{v}}{c})^2}} =: \gamma , \quad (2.12)$$

and

$$\frac{dx^\mu}{d\tau} = \gamma(c, \mathbf{v}) . \quad (2.13)$$

Multiplying this equation with the invariant mass m yields the four-momentum

$$p^\mu = mc^2 = m \frac{dx^\mu}{d\tau} = m\gamma(c, \mathbf{v}) = \left(\frac{E}{c}, \mathbf{p} \right) . \quad (2.14)$$

The invariant of the four-momentum is obtained by calculating the inner product of p^μ :

$$\sum_{\mu=0}^3 p^\mu p_\mu = \frac{E^2}{c^2} - \mathbf{p}^2 = m^2 c^2 . \quad (2.15)$$

This is the relativistic energy-momentum relation:

$$E = c \sqrt{(mc)^2 + \mathbf{p}^2} . \quad (2.16)$$

In the rest-system of an observer ($\mathbf{p} = 0$) the energy is

$$E_0 = mc^2 , \quad (2.17)$$

¹⁹A vector is just a first order tensor.

an equation, that has been first derived by Albert Einstein in 1905 [117] and in several later publications [118, 119]. For a photon ($m = 0$), for which no rest system exists, the energy is $E = c\mathbf{p}$. In some textbooks on relativity, even in the famous 1921 article by W. Pauli [120], a distinction between a *rest mass* m_0 and a *relativistic, velocity-dependent mass* $m_0\gamma$ is made; this however is deceptive, as the mass of a system is a fundamental property of matter and as such an invariant, which does not change with the frame of reference. Thus, it is important to understand that not mass, but rather the *energy* and the *momentum* of a system depend on the state of motion of the observer. In the reference frame in which the system is at rest (in its *rest system*), its energy is not zero, but $E_0 = mc^2$, i.e. proportional to its frame-independent mass m . Energy E and momentum \mathbf{p} are both components of a four-vector (2.14), and transform together when changing the coordinate system.

A light signal in the standard configuration of Fig. 2.8 travels at a speed of $c = dx/dt$. Therefore, $ds^2 = 0$ for light signals. Thus, it follows

$$\begin{aligned} ds'^2 &= c^2 dt'^2 - dx'^2 = \eta_{\alpha\beta} dx'^{\alpha} dx'^{\beta} \\ &= \eta_{\alpha\beta} \Lambda_{\gamma}^{\alpha} \Lambda_{\delta}^{\beta} dx^{\gamma} dx^{\delta} = ds = \eta_{\gamma\delta} dx^{\gamma} dx^{\delta} = 0. \end{aligned} \quad (2.18)$$

Hence, by comparison, one obtains

$$\Lambda_{\gamma}^{\alpha} \Lambda_{\delta}^{\beta} \eta_{\alpha\beta} = \eta_{\gamma\delta}. \quad (2.19)$$

The space-time translations in (2.8) drop out when taking the differential

$$dx'^{\alpha} = \Lambda_{\beta}^{\alpha} dx^{\beta}. \quad (2.20)$$

Because of $x^2 = x'^2$ and $x^3 = x'^3$ in the standard configuration of Fig. 2.8 one can write the transformation as a matrix in the form (the motion is only in x -direction):

$$\Lambda = (\Lambda_{\beta}^{\alpha}) = \begin{pmatrix} \Lambda_0^0 & \Lambda_1^0 & 0 & 0 \\ \Lambda_0^1 & \Lambda_1^1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.21)$$

From (2.19) it follows:

$$(\Lambda_0^0)^2 - (\Lambda_0^1)^2 = 1, \quad (2.22a)$$

$$(\Lambda_1^1)^2 + (\Lambda_1^0)^2 = -1, \quad (2.22b)$$

$$\Lambda_0^0 \Lambda_1^0 - \Lambda_0^1 \Lambda_1^1 = 0. \quad (2.22c)$$

As a solution of the set of equations in (2.22) one may set $\Lambda_1^0 = -\sinh \Theta$ and $\Lambda_0^1 = -\sinh \Theta$. This yields

$$\Lambda = (\Lambda_{\beta}^{\alpha}) = \begin{pmatrix} \Lambda_0^0 & \Lambda_1^0 \\ \Lambda_0^1 & \Lambda_1^1 \end{pmatrix} = \begin{pmatrix} \cosh \Theta & -\sinh \Theta \\ -\sinh \Theta & \cosh \Theta \end{pmatrix}. \quad (2.23)$$

For the origin of Σ' the following equation applies, cf. Fig. 2.8:

$$x'^1 = 0 = \Lambda_0^1 ct + \Lambda_1^1 vt, \quad (2.24)$$

and it follows:

$$\tanh \Theta = -\frac{\Lambda_0^1}{\Lambda_1^1} = \frac{v}{c} = \beta. \quad (2.25)$$

With (2.21), (2.23) and (2.25) all matrix elements are fixed. Expressed as a function of velocity v , the matrix elements are:

$$\Lambda_0^0 = \Lambda_1^1 = \gamma = (1 - \beta^2)^{-1/2}, \quad (2.26a)$$

$$\Lambda_1^0 = \Lambda_0^1 = \frac{-v/c}{\sqrt{1 - \frac{v^2}{c^2}}}. \quad (2.26b)$$

As a result, the *Lorentz-transformations* are given by:

Definition 2.2 (The Lorentz-Transformations)

$$x' = \gamma(x - vt), \quad y' = y, \quad z' = z, \quad (2.27a)$$

$$ct' = \gamma(ct - \beta x). \quad (2.27b)$$

These transformations allow for transforming spacetime coordinates from one *locally* defined inertial system (IS) to another IS with its own *local* spacetime coordinates. Thus, special relativity provides a fundamental insight into the structure of spacetime, used in physical theories in that each single observer carries his own set of coordinates, with which spacetime intervals between events are calculated. Hence, there is no *global* coordinate system that can be provided for all observers in inertial systems and due to this there is no common notion of simultaneity.

A clock, that is at rest in system Σ in Fig. 2.8 displays the so-called proper time $\tau = t$. Thus, the proper time between two events, measured by a clock that is at rest in a frame of reference Σ coincides with the coordinate time and is a directly measurable quantity. In a new (primed) coordinate system Σ' with coordinates x^α , the coordinate differentials are given by

$$dx^{\alpha'} = \Lambda_\beta^\alpha dx^\beta. \quad (2.28)$$

Thus, the new coordinate time $d\tau'^2$ will be

$$d\tau'^2 = \eta_{\alpha\beta} dx'^\alpha dx'^\beta = \eta_{\alpha\beta} \Lambda_\lambda^\alpha \Lambda_\delta^\beta dx^\lambda dx^\delta = \eta_{\lambda\delta} dx^\lambda dx^\delta, \quad (2.29)$$

and therefore

$$d\tau'^2 = d\tau. \quad (2.30)$$

It is easy to see that for $\beta = \frac{v}{c} \ll 1$, prefactor $\gamma = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!} (\beta)^n \approx 1$ in zeroth approximation, thus yielding the approximatively valid *Galilei transformation* (cf. (3.10) on p. 124), where time parameter t does not depend kinematically on the spacial coordinates, that is, on the arbitrarily chosen inertial frame of reference. In this case, the underlying spacetime manifold has the same *global* properties for all observers.

All so-called special relativistic “paradoxa”, such as length contraction ($L' = \gamma L$), time dilatation ($t' = \gamma t$),²⁰ or the relativity of simultaneity are simple consequences of special relativistic kinematics. That is, they arise simply because the numerical values of spacetime coordinates change when switching between inertial systems.

In engineering contexts, similar kinematic properties arise with objects defined in continuum theory such as the deformation or velocity gradients, when they are transformed to a different coordinate system, see e.g. [123]. Usually, the transformations considered in this context, are very special *Euclidean transformations* which include a translation of the system and an orthogonal rotation, and the so-called “non-objectivity”²¹ of several quantities does not come as a surprise as they are not defined as covariant²² tensor equations.

It is important to realize that the changing coordinates (and their transformation rules) are actually not the point of special (and general) relativity theory,²³ but rather those properties of spacetime, which are *invariant* upon coordinate changes, such as events as such, or the infinitesimal *spacetime interval* $ds^2 = c^2 dt^2 - d\mathbf{x}^2$, which connects physical events in spacetime through *timelike* ($ds^2 > 0$), *spacelike* ($ds^2 < 0$), and null ($ds^2 = 0$) worldlines. The implications of this theoretical structure underlying classical physics are further considered in Sect. 3.6.1 of Chap. 3 on p. 163.

2.4.4 Numerical Modeling and Simulation

Once a decision is made, a physical model is expressed in mathematical equations which (usually) can be solved in a systematic way, that is, in a way that can be formulated as a finite stochastic or deterministic algorithm²⁴ and implemented as a computer program. The numerical solutions of the governing equations associated with the physical and mathematical model are then interpreted and provide answers to the specific real system which was transformed into the model system. In Table 2.2, several key classification aspects in physical and mathematical modeling are collocated. A comparison of the answers for a specific problem obtained by mathematically exploiting a specific model, finally provides some ideas about the general validity and the quality of a model system and the derivations and

²⁰The most prominent application of this kinematic effect is the *twin paradox*, see e.g. [121, 122], which is based on the asymmetry between the twin that stays in the same IS at home, and the one who travels, and has to accelerate, i.e. to switch inertial systems, in order to return. For a modern treatment of the clock paradox, see [122].

²¹Non-invariance of equations upon coordinate transformations.

²²See Sect. 3.3.8 on p. 153.

²³Thus in a sense, the term “theory of relativity” is a misnomer and it had probably better be called “theory of invariants”.

²⁴For a definition of “algorithm”, see Sect. 2.6.

Table 2.2 General key aspects to be considered in the development of physical and mathematical models for simulation applications beyond the atomic scale

Classification	Example
Spatial dimension	1D, 2D, 3D
Kind of discretization	Particles, super particles, continuum (fields)
Spacial scale	Macroscopic, mesoscopic, microscopic, nanoscopic
State variables	Strain, displacement, dislocation density, temperature
Material properties	Hooke's law, Taylor equations, multiparameter plasticity
Degree of predictability	Ab initio, coarse-grained, phenomenological, empirical

theoretical concepts associated with it. This principal procedure in physical and numerical modeling is illustrated in the flowchart of Fig. 2.9.

2.5 Unifications and Reductionism in Physical Theories

In Sect. 2.3 it was stated that on the one hand one has to isolate a system in order to extract the fundamental, universal laws; on the other hand, the natural sciences – and in particular physics – are very much focused on *unifying* different approaches, models and theories, i.e. to reduce existing systems to ever more consistent and fundamental systems that include an ever larger part of observed reality. This paradigm of “reductionism” expresses the idea that it is possible to understand the functioning of arbitrary complex (physical, chemical, biological) systems by reducing them to simpler and smaller systems and by applying the laws of nature to these subsystems. A pessimist could say, that this “fool-proof philosophy” has not led to anything that goes beyond old Platonist concepts of “ideal forms” that are hidden behind the empirical observations (see e.g. Chap. 4 in “The Presence of the Past” by Rupert Sheldrake [124]), but its simplicity is very attractive and has led to the successful unification of different theories of physical phenomena during the last 200 years, see Fig. 2.10.

The paradigm of reductionism is carried to extremes in modern high energy particle physics. Here, it led to the successful development of the *standard model* in the 1960s and 70s, which explains the structure of subatomic particles and all fundamental interactions, gravitation excepted, by the existence of structureless, pointlike elementary spin 1/2 particles, called “quarks” and “leptons” along with their associated symmetries. The model is referred to as “standard”, because it provides a theory of all fundamental constituents of matter as an ontological basis.

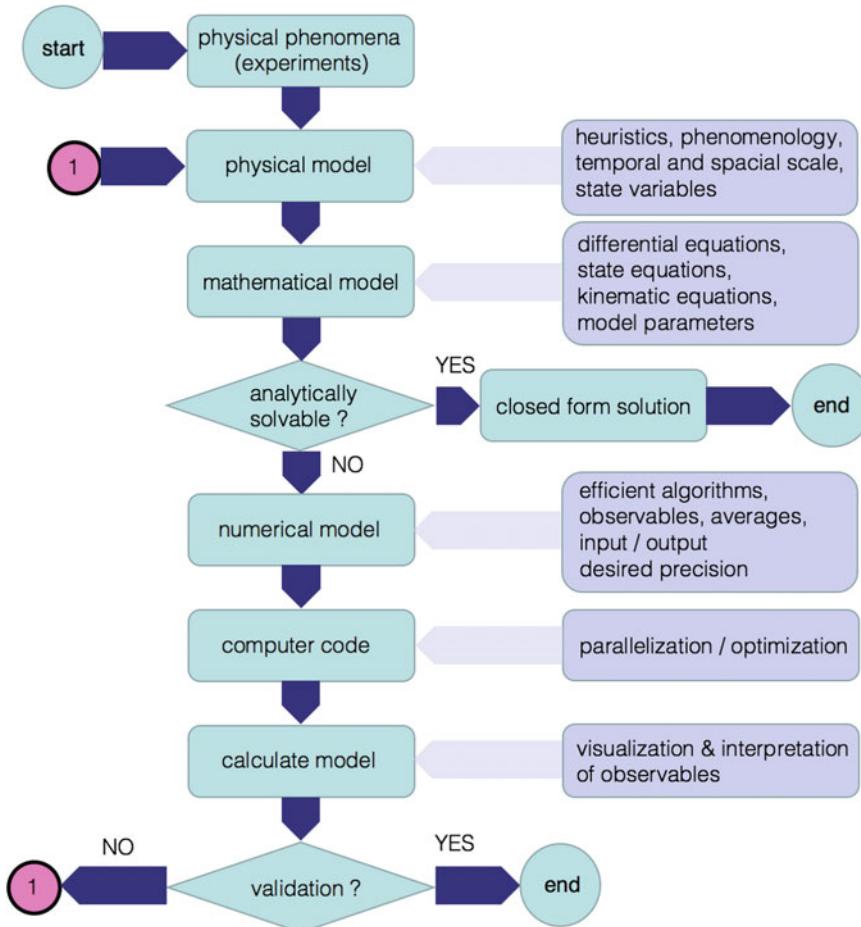


Fig. 2.9 Principal physical, mathematical and numerical modeling scheme illustrated as flow chart. Starting from the experimental evidence one constructs physical theories which determine in principle what can be measured. A mathematical formulation usually leads to differential equations, integral equations, or master (rate) equations for the dynamic (i.e. time dependent) development of certain state variables within the system's abstract state space. Analytic solutions of these equations are usually rarely possible, except for simplified approximation usually due to symmetry. Thus, efficient algorithms for the treated problem have to be found and implemented as a computer program. Execution of the code yields approximate numerical solutions to the mathematical model which describes the dynamics of the physical "real" system. Comparison of the obtained numerical results with experimental data allows for a validation of the used model and subsequent iterative improvement of theory

In this section, for the sake of completeness, we end our discussion of model building with a short (and very superficial) presentation of the ideas pertaining to this fundamental theory and its range of application. The interested reader is referred to specialized literature, see Sect. 2.5.5 on p. 77.

2.5.1 The Four Fundamental Interactions

According to our present knowledge, there are four known fundamental interactions, see Table 2.3 and compare Fig. 2.10:

1. weak interaction (the force related to β -decay);
2. strong interaction (nuclear force);
3. electromagnetic interaction (Coulomb force);
4. gravitational interaction.

The two nuclear forces (1) and (2) do not pertain to any human everyday experience and are very short ranged. They are treated within the framework of non-Abelian²⁵ quantum field theories which couple the special principle of relativity with quantum theory. The strong interaction, which keeps protons and neutrons in the nucleus together, has a range of 10^{-15} m. The weak interaction finally even has such a short range ($\leq 10^{-17}$ m) that it only manifests itself in certain particle collisions or decay processes within the nuclei of atoms. Weak interactions form the first

Table 2.3 In modern physics, it is assumed that all interactions between particles – and thus, the totality of physical reality – can be described by four fundamental interactions. Each fundamental force has a certain interaction range. In two cases, the range is infinity. The relative strength of the forces can be compared when using *natural units* such that the forces can be assigned dimensionless numbers. The gravitational interaction in these natural units is about 39 orders of magnitude weaker than the strong interaction. Table compiled from [125, 126]

	Weak	Strong	Electromagnetism	Gravitation
Range	$\ll 10^{-15}$ m	10^{-15} m	∞	∞
Example	β -decay of atomic nuclei	Atomic nuclei	Forces between charges	Forces between astronomic objects
Strength	$G_{Fermi} = 1.02 \times 10^{-5}$	$g^2 \approx 1$	$e^2 = 1/137$	$G_{Newton} = 5.9 \times 10^{-39}$
Affected particles	Quarks/leptons	Quarks	Charged particles	All
Exchange particles	Vector bosons W^\pm, Z^0	Gluons g_i ($i = 1, \dots, 8$)	Photon γ	Higgs H (graviton)

²⁵For the definition of an Abelian group, see Box 2.1 on p. 68.

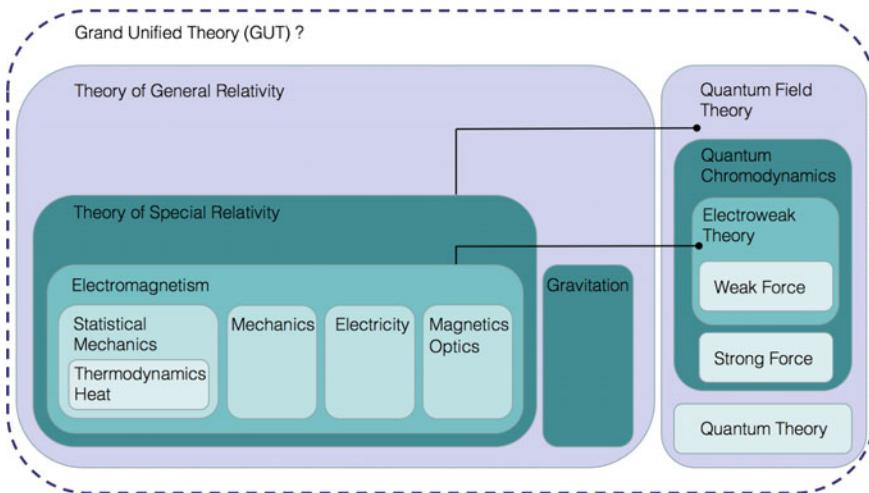


Fig. 2.10 Reductionism in physics. Important unifications of *classical* physical theories based on the concepts of particles and fields are summarized on the *left*. Unifications based on quantum theory By extending the Galilean principle of relativity to *all* processes in physics including electrodynamics, the STR (1905) could reconcile the apparent discord between Newton's system of classical mechanics (1687) and Maxwell's equations (1865). Combining STR with quantum theory (1925/1926) yields quantum field theory (1931) in which fields are treated as operators in Hilbert space, which obey commutator relations and which gives rise to the concepts of antimatter and particle annihilation and creation out of the vacuum (Dirac's hole theory). Within this framework, the fundamental weak force and electromagnetism could be united in the electroweak theory and the strong interactions of hadrons could be explained in a dynamic theory of quarks, called QCD (1970). Thus, three of the four known fundamental interactions have been unified in more general quantum field theories; gravitation, the weakest of the four interactions, which was transformed from a theory with action at-a-distance (1687) into a proper non-instantaneous field theory by the general theory of relativity (1915), might be unified with the other forces in an all-embracing "theory of everything", or GUT, a "Grand Unified Theory"

step in the nuclear chain reaction in the interior of the sun, where two protons fuse and a deuterium nucleus, a positron²⁶ and a neutrino come into being. These two interactions can be neglected in the atom except within the nucleus and thus they can be completely neglected in conventional engineering applications and in most applications in physics. The so-called *first-principles*, or *ab initio calculations* (discussed in Chap. 5) only take into account electromagnetism (3) and gravitational forces (4) in the framework of *non-relativistic quantum mechanics*. Technically speaking, ab initio methods solve the Schrödinger equation to determine the electron density distribution, and the atomic structures of various materials.

Gravitation is the weakest of all fundamental interactions, see Table 2.3, but plays a dominant role on a cosmic scale, because the planets and stars in galaxies are large

²⁶The positron e^+ is the anti-particle of the electron e^- .

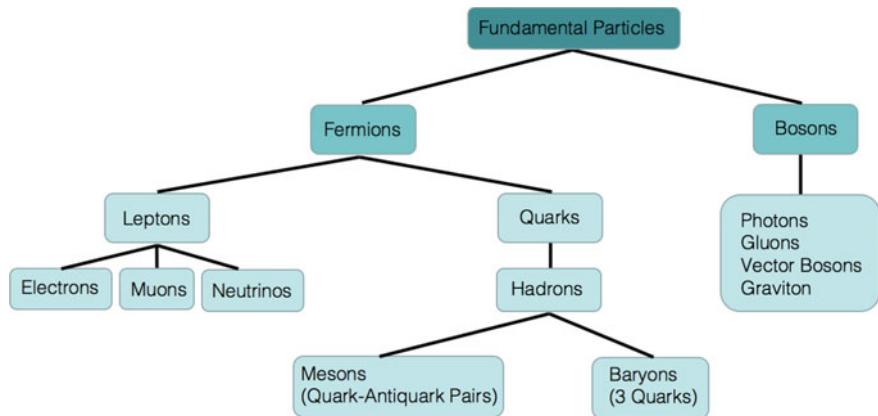


Fig. 2.11 The fundamental particles according to the standard model are the *fermions* (particles with spin 1/2) and the *bosons* (particles with integral spin or spin 0). The fermions split into the quarks, of which the hadrons – particles of the strong interaction – are made, and the leptons, which do not participate in the strong interaction. Baryons are triplets of quarks, such as the proton (uud) or the neutron (udd) and mesons consist of quark-anti-quark pairs, e.g. the pion ($\bar{d}u$)

agglomerations of electrically neutral masses. Hence, the electromagnetic interaction between electrons and atom cores is in principle responsible for *all* chemical and physical properties of ordinary solids, fluids and gases. Compared to continuum mechanics methods, atomic scale simulations are truly ab initio. However, even in ab initio methods, there are several approximations involved in simulations of the quantum state of many-electron systems, e.g. the *Born-Oppenheimer approximation*, discussed in Chap. 5.

2.5.2 The Standard Model

According to the current standard model of elementary particle physics based on quantum field theory, the fundamental ontology of the world is a set of interacting, quantized fields, which arise as two types of fields in the standard model: matter fields and interaction fields, cf. Fig. 2.11.

The quanta of matter fields, called fermions, have half-integral spins. They obey Pauli's exclusion principle which is the basis of structured matter: only a single fermion can occupy a particular quantum state. The quanta of the interaction fields, or *bosons*, have integral spins and thus, many bosons can occupy one quantum state.²⁷ There are 12 matter fields, organized in three generations, or families, and each has its anti-field, cf. Table 2.4.

²⁷For example, a coherent laser beam comprises billions of photons oscillating in a single state.

Table 2.4 There are 12 known elementary particles called fermions (quarks and leptons) and 4 exchange particles according to the standard model of elementary particle physics. The particles, which are sources of fields, are listed along with their quantum numbers B , L , their charge Q , and their masses m . Each particle also has an anti-particle (not listed); ordinary stable matter is made of particles of the first generation only. The other particles show up only in high-energy experiments for very short time intervals. All exchange particles except the vector bosons are stable. Table compiled from [125, 126]

Generation			Spin	Baryon	Lepton	Charge
1	2	3	S	B	L	Q
Quarks						
u (up) $m = 5 \text{ MeV}$	c (charm) $m = 1.5 \text{ GeV}$	t (top) $m = 174 \text{ GeV}$	1/2	1/3	0	+2/3
d (down) $m = 10 \text{ MeV}$	s (strange) $m = 200 \text{ MeV}$	b (bottom) $m = 4.7 \text{ GeV}$	1/2	1/3	0	-1/3
Leptons						
ν_e $m \sim 0$	ν_μ $m \sim 0$	ν_τ $m \sim 0$	1/2	0	1	0
e^- $m = 0.511 \text{ MeV}$	μ^- $m = 105 \text{ MeV}$	τ^- $m = 1.7 \text{ GeV}$	1/2	0	1	-1
Exchange particles						
Photon	(Stable)	γ	1	0	0	0
Gluons	(Stable)	$g_i, i = 1, \dots, 8$	1	0	0	0
Vector bosons	($\sim 10^{-25} \text{ s}$)	Z, W^\pm	1	0	0	0, ± 1
Higgs	(Stable)	H	2	0	0	0

The higher generations are just replicas of the first generation with short lifetimes and show up only in high energy cosmic rays or in certain particle reactions in accelerators. All stable matter in the universe according to this model is made up of only three matter fields of the first generation: electron (e^-), up quark (u), and down quark (d) fields. Protons (uud) and neutrons (udd) are made of quarks and are the constituents of atoms. The neutrinos interact weakly with everything and are not part of stable matter. All particles listed in Table 2.4 are elementary, i.e. they have no known substructure; this, however, does not exclude their decay into other particles, e.g. the myon μ^- may decay into an electron and two neutrinos (the electron anti-neutrino $\bar{\nu}_e$ and the muon neutrino) according to $\mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu$. The possible number of decay processes is restricted by empirical conservation laws of quantum mechanical quantities; some examples are listed on p. 76, in Table 2.6 and the examples on p. 71.

The interaction fields are permanently coupled to the matter fields, whose charges are their sources. For example, the electric charge is the source of the electromagnetic field, and the color charges of the quarks are the source of the strong interaction. Fundamental interactions occur only between matter and interaction fields, and they occur at a point. The mathematical form of the point coupling for the

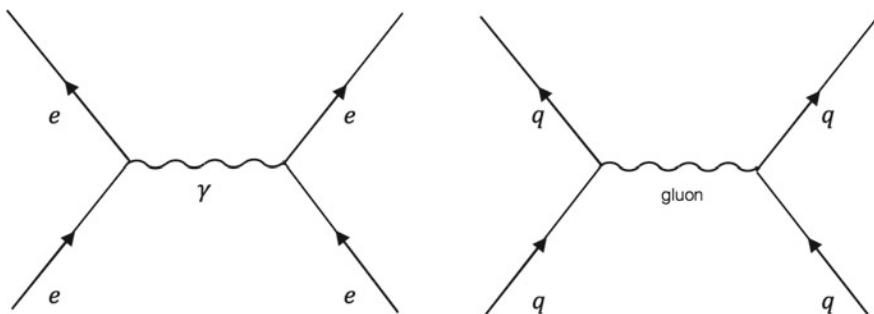


Fig. 2.12 Matter fields are represented in Feynman diagrams as straight lines and the interaction fields by wavy lines. Displayed are examples of the electromagnetic interaction between electrons (e) (left) and the strong interaction between quarks (q) (right). Feynman diagrams are a graphical way of representing the contributions to the scattering S -matrix elements in perturbation theory, which can be applied, if the theory is renormalizable

nuclear forces is the same as in the case of the electromagnetic field and can be graphically represented as Feynman diagrams, cf. Fig. 2.12. For this classical field, R.P. Feynman (1918–1988) [75], and independently J.S. Schwinger (1918–1994) [127] and S. Tomonaga²⁸ developed a theory, quantum electrodynamics (QED), which is consistent with quantum theory and which allowed to calculate probability amplitudes, cross-sections and decay rates for the electromagnetic interaction. It was the only consistent, that is, renormalizable quantum field theory until the t'Hooft publication in the 1970s [128]. This theory became the prototype of all subsequent quantum field theories. The standard model has been tested to 10^{-18} m and at present, there is no known contradiction to any experiment. Table 2.5 gives a selected overview of important discoveries in the field of elementary particle physics; several of these were later awarded the Nobel prize, e.g. the discovery of the Ω^- -particle, which had been theoretically predicted with correct properties due to the $SU(3)$ symmetries of quark theory, cf. Fig. 2.13.

2.5.3 Symmetries, Fields, Particles and the Vacuum

The four fundamental forces, or interactions emerge from the exchange of particles, so-called *bosons*, which is depicted schematically in Fig. 2.12. The basic objects of this fundamental physical picture of the world are the concepts of *field*, *particle*, *vacuum* and their underlying *symmetries*. These symmetries used in the quantum field theories of the standard model build the mathematical ontology of physics.

At the beginning 20th century it was realized that classical physics is inadequate for the description of quantum structures. In 1925, non-relativistic quantum theory was established and quickly became the basis of a large part of physics, including

²⁸The three shared the Nobel prize in 1965.

Table 2.5 A selection of important discoveries in the history of elementary particle physics

Year	Discovery	Reference
5th century BC	4 basic elements in Greek philosophy:earth, air, fire and water	The presocratics [71]
1789	List of 30 elements	Lavoisier [129]
1868	Periodic table of elements	Mendeleev [130]
1896	Electron	Thomson [131]
1905	Special theory of relativity	Einstein [117]
1911	Atomic nucleus	Rutherford [80]
1911	Nucleus and shell model of atoms	Bohr [132]
1915	General theory of relativity	Einstein [133]
1925	Quantum theory	Heisenberg [81]
1930	Quantum field theory	Dirac [134, 135]
1930	Prediction of neutrino	Pauli [16]
1932	Neutron	Chadwick [136]
1932	Positron	Anderson [137, 138]
1948/1949	Quantum electrodynamics (QED)	Schwinger [127], Feynman [75], Tomonaga
1956	CP-Violation	Landa et al. [139]
1961	Eightfold way	Gell-Mann [140]
1964	Quark model	Gell-Mann [141], Zweig [142, 143]
1964	Ω^- -particle	Bernes et al. [144], Glashow [145],
1961–1969	Electroweak theory	Weinberg [146], Salam [147]
1972	QCD	Fritsch, Gell-Mann [148]
1973	Asymptotic freedom of quarks	Politzer [79], Gross, Wilzek [77, 78]
1974	J/Ψ -particle	Aubert et al. [149]
1974	Renormalizability	t’Hooft [128]
1983	Intermediate vector bosons W, Z^\pm	Rubbia et al. [150]
1995	Top quark	Abe et al. [151]
2014	Higgs boson	The CMS Collaboration [152]

atomic, molecular and solid-state phenomena. However, in nuclear and high-energy physics, this theory is unsatisfactory because it is incompatible with the principle of special relativity advanced by Einstein in 1905 [117]. Dirac formulated *quantum field theory* (QFT) by uniting quantum mechanics and special relativity in 1930 [134,135], predicting the existence of a new particle, called “anti-electron”, then unknown to experimental physics, having the same mass and opposite charge of an electron. He further predicted that this new stable particle could be produced in a high vacuum by an “encounter between two hard γ -rays (of energy at least half a million volts)”, leading “simultaneously to the creation of an electron and an anti-electron” [135].

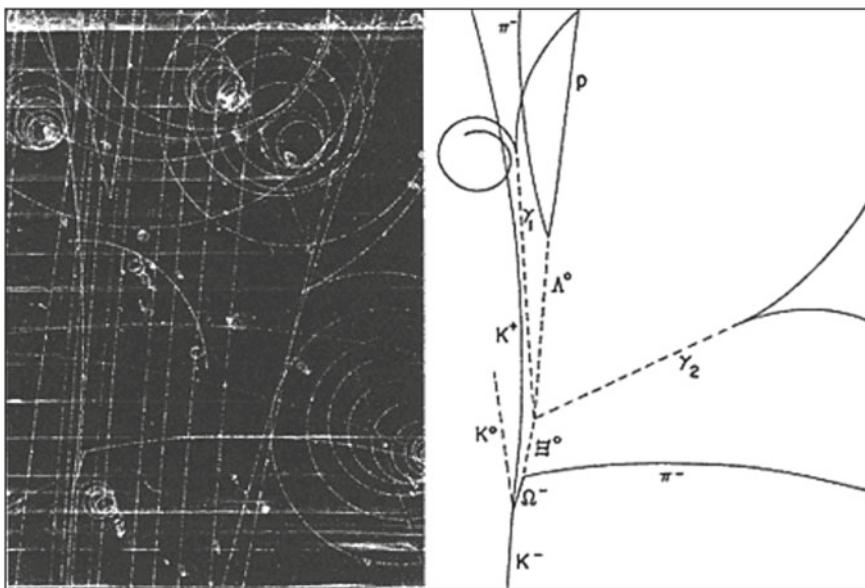


Fig. 2.13 Bubble chamber photograph (*left*) and *line diagram* (*right*) of event showing the first Ω^- -particle. An incoming K^- -meson interacts with a proton in the liquid hydrogen of the bubble chamber and produces an Ω^- , a K^0 and a K^+ meson which all decay into other particles. Neutral particles which produce no tracks in the chamber are shown by *dashed lines*. The presence and properties of the neutral particles are established by analysis of the tracks of their charged decay products and application of the laws of conservation of mass and energy. Photo courtesy of Brookhaven National Laboratory

Quantum field theory introduced into physics the concept of anti-particles and the concept of creation (and annihilation) of new particles from pure energy, thus changing also the scientific notion of the “vacuum”.²⁹ The anti-particle of the electron (e^+) was experimentally discovered in 1933 by C.D. Anderson [137].

The extension of Dirac’s relativistic quantum theory to include *nuclear* interactions took another 25 years. During this process, *gauge fields*, i.e. fields with local symmetries, the idea of which first appeared in the general theory of relativity (in the form of varying orientations of local inertial frames), became dominant. Weyl tried to generalize this idea and suggested that the “scale” of local frames should also be allowed to vary, so the frames would be enlarged or reduced as one goes about in the manifold.³⁰ The variation of the frame’s scale would be reconciled by

²⁹In Dirac’s *hole theory*, a “vacuum” is interpreted as the negative energy spectrum of the solutions of his equation. The holes in the “Dirac sea” of negative energies were first interpreted by Dirac as protons [134], but this idea was quickly abandoned under the impression of several arguments put forward by W. Pauli and others.

³⁰For a discussion of the importance of manifolds in modern physical theory, see Chap. 3.

Box 2.1 Definition of a group, Abelian group

A group G is a set of elements $\{g_i\}$ with a single rule of composition \circ such that:

- (a) G is closed, i.e. for any two elements $g_1, g_2 \in G$, $g_1 \circ g_2 \in G$,
- (b) The composition is associative; i.e. $g_1 \circ (g_2 \circ g_3) = (g_1 \circ g_2) \circ g_3$,
- (c) G contains an identity element e such that for all $g \in G$, $g \circ e = e \circ g = g$,
- (d) For every $g \in G$ there exists an inverse element $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = e$.

A group G is commutative or Abelian, if $g_1 \circ g_2 = g_2 \circ g_1$ for all $g_1, g_2 \in G$.

the electromagnetic field, just as the variation of their orientation is reconciled by the gravitational field. Weyl called this “Eichinvarianz”, which was translated into English in the 1920s as “gauge invariance”. Weyl’s idea did not work; Einstein pointed out that the proposed scale change renders the rate of a clock dependent on its history, which is not acceptable. With the development of a quantum theory in mid 1920s, it was realized that what varies from point to point is not the scale but the phase of the electron wave function. However, the old names “gauge invariance”, “gauge fields” and “gauge theories” are still prevalent.³¹

The electromagnetic field is not self-interacting, that is, its field quanta, the photons, do not carry electric charge; thus, the photons do not stick together to form a “light ball”. Mathematically, this feature can be seen in the fact that the local symmetry group of electromagnetism is commutative, or Abelian, cf. Box 2.1. In contrast, the symmetry groups of the nuclear interactions are non-commutative, or non-Abelian. The reason for this is that the field quanta of the weak and strong interactions carry coupling charges and thus interact with themselves, whereas the photon is massless and does not carry charge. This is what makes non-Abelian theories more complicated than electromagnetism.

In physical literature, the term “field” has at least two different connotations. First, a field is a continuous dynamical system, i.e. a system with an infinite number of degrees of freedom. Second, a field is also a dynamical variable characterizing such a system or at least some aspect of a system. The description of field properties is *local*, concentrating on a point entity and its infinitesimal displacement. Literally speaking, the world of fields is “full”, whereas in the mechanistic world, particles are separated by empty space across which forces act instantaneously at a distance.

According to the principle of special relativity, no signal can travel at a velocity faster than the velocity c of light; thus, c determines an upper bound at which forces between particles can act. For the conservation laws of energy-momentum to be valid at every moment in time, one assumes that a particle gives rise to a field in its surrounding space which transports energy and momentum. Taking into account that energy and momentum is quantized one is led to the identification of these field quanta as particles. Thus, combining special relativity and quantum mechanics

³¹If one were to rename gauge fields today, they would probably be called “phase fields” as the symmetry with matter fields is with respect to their phase, not to some length scale.

naturally leads to the concept of a field theory in which the fields are quantized themselves and are made up of “exchange particles”, the field quanta.

Symmetries

STR rests on two simple postulates: the principle of special relativity and the constancy of the speed of light. The general theory of relativity GTR also has two fundamental postulates: the principle of general relativity and the equivalence principle. Here, we are interested in one aspect of these principles, namely the idea of symmetry. Each principle specifies an equivalence class of coordinate systems which constrains the content of a physical theory. Whatever these constraints may be, the idea of the principles is that certain physical quantities are invariant under certain groups of coordinate transformations; stated in this way the principles are symmetry principles as they state invariance of a system against certain symmetry operations which transform the object back into itself or which leave the object unchanged or invariant. The set of symmetry transformations form a group. A group is an algebraic structure with a single rule of composition, cf. Box 2.1.

The *invariant* features of symmetries are usually the focus of interest. For example, the invariants under the Galilean group of transformations are the time interval as well as the spatial distance and one invariant under the Lorentz group is the spacetime interval $d\tau^2 = c^2dt^2 - dx^2$. In mathematics, in general, the profundity of a concept is associated with its generality; thus, the largest transformation group defines the most fundamental concept. A system, characterized by a large symmetry group retains only the important features. In elementary particle physics, one seeks ever larger symmetry groups in the attempt to ever larger unification. For example, the electromagnetic interaction is characterized by a unitary group of order 1, $U(1)$, the weak interaction by the special unitary group $SU(2)$, see Box 2.5.3, and their unification in the electroweak interaction is achieved by the larger group $SU(2) \times U(1)$; thus, the standard model is based on the $U(1) \times SU(2) \times SU(3)$ symmetry. In QED, demanding $U(1)$ gauge invariance means that the theory is supposed to be invariant under the transformation $\Psi(x) \rightarrow \exp(i\alpha(x))\Psi(x)$, where $\exp(i\alpha(x))$ is a *local* phase transformation, whereas the transformation is *global* if $\alpha(x) = \text{const}$. For more details on groups and their properties, the interested reader is referred to more specialized literature, cf. Sect. 2.5.5.

The symmetry $SU(3)$ underlies the strong interaction which treats u , d , and s quarks as equivalent, see Box 2.5.3. If the symmetry group of a physical system is reduced to one of its subgroups (for example, the $SU(2)$ isospin symmetry is a subgroup of $SU(3)$), one says that the *symmetry is broken*. As a result of a broken symmetry, one obtains more invariants and more features.

Conservation Laws

Another important group of general principles are *conservation laws* which are a consequence of underlying symmetries in a system, usually expressed in the Hamilton operator \mathcal{H} (in quantum systems) or the Hamilton function H for classical systems. For example, if $\frac{\partial H}{\partial t} = 0$, that is if H is time-independent, energy is conserved and $H = E = \text{const}$. Homogeneity of space (H is invariant against translations) leads to conservation of momentum, and isotropy of space leads to conservation of angular momentum. In the following, some important conservation laws are listed:

Box 2.2 $SU(2)$ Symmetry

A set of operators $J_i \in \mathbf{C}^{2,2}$ which obey the relation

$$[J_i, J_k] = i\epsilon_{ikl}J_l \quad (i, k, l \in \{1, 2, 3\}) \quad (2.31)$$

is called an $SU(2)$ -algebra. A possible representation of this algebra is given, e.g. by the Pauli matrices. Setting

$$J_1 = \frac{1}{2} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, J_2 = \frac{1}{2} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, J_3 = \frac{1}{2} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (2.32)$$

one easily shows that the commutator relations of (2.5.3) are fulfilled. The quantities J_j are called generators of the group. If a Hamilton operator is invariant under the transformations of a group, then it commutes with the generators, i.e. in the case of $SU(2)$ symmetry it commutes with the J_j .

$$U = \exp \left(-i \sum_{j=1}^3 \phi_j J_j \right) \quad (2.33)$$

are the special unitary transformations in 2D and the ϕ_j are field variables.

- Conservation of energy-momentum
- Conservation of angular momentum
- Conservation of baryon number B (nucleons and hyperons³² are assigned +1 and their anti-particles are assigned -1)
- Conservation of lepton number L (ν_e^- and e^- are assigned +1 and their anti-particles get -1)
- Conservation of isospin I_z (only valid for the strong interaction)
- Conservation of strangeness S (valid for the strong and electromagnetic interaction but not for the weak interaction)
- Conservation of parity P (valid for the strong and electromagnetic interaction but not for the weak interaction)

The general connection between symmetries and conservation laws is provided by two theorems published by Emmy Noether in the article “Invariante Variationsprobleme” in 1918 [153]. Table 2.6 shows a selection of important conserved quantities and the – according to Noether’s theorem – associated symmetries.

Example 3 (Some Particle Decays)

- Myon decay:

The process $\mu^- \rightarrow e^- + \bar{\nu}_e + \nu_\mu$ is allowed. It obeys conservation of myon lepton number $1 \rightarrow 0 + 0 + 1$ and electron lepton number $0 \rightarrow 1 + (-1) + 0$.

The process $\mu^- \rightarrow e^- + \bar{\nu}_e$ violates conservation of L .

³²Hyperons are baryons with a strangeness quantum number.

Box 2.3 SU(3) Symmetry

Let U be an unitary $n \times n$ matrix, i.e. $U \in U(n)$, $U^\dagger U = 1$ and H an hermitean $n \times n$ matrix. Then U can be written as

$$U = \exp(iH). \quad (2.34)$$

As H is hermitean, there are n^2 independent, real parameters for H and U . With $\det(U) = 1$, the matrices U build the special unitary group $SU(n)$, which, due to (2.5.3), depends on $n^2 - 1$ parameters λ_ν , which are called *generators*. It can be shown that

$$\det(U) = \det(\exp(iH)) = \exp(i\text{Tr}H). \quad (2.35)$$

It is

$$U = \exp(-i\alpha_\nu \lambda_\nu), \quad (2.36)$$

$$H = -i\alpha_\nu \lambda_\nu. \quad (2.37)$$

and

$$\text{Tr}(\lambda_\nu). \quad (2.38)$$

Hence, the $n^2 - 1 = 3^2 - 1 = 8$ traceless generators of $SU(n)$ have to be traceless matrices. It is common to introduce the following convention for the generators: $J_j = \frac{1}{2}\lambda_j$

$$\lambda_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \lambda_2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \lambda_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (2.39a)$$

$$\lambda_4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \lambda_5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix}, \lambda_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad (2.39b)$$

$$\lambda_7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix}, \lambda_8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{pmatrix}, \quad (2.39c)$$

The corresponding representation of the group is $\exp(-\frac{i}{2} \sum_j \phi_i \lambda_j)$. One obtains different representations by finding eight different generators, which obey the same commutator relations as the J_i , i.e. $[J_i, J_k] = i\epsilon_{ikl}J_l$ with the anti-symmetric quantities ϵ_{ikl} . Instead of J_3 and J_8 , the corresponding quantum numbers in physics are called *isospin* and *hypercharge*.

- Proton decay:

The process $p \rightarrow e^+ \gamma$ violates conservation of B .

- Electron decay:

The process $e^- \rightarrow \mu^- + \nu_e + \bar{\nu}_\mu$ is forbidden, as $m_e < m_\mu + m_{\nu_e} + m_{\mu_\nu}$.

- Neutron decay: The process $n \rightarrow p + e^- + \bar{\nu}_e$ obeys the conservation of electron lepton number: $0 \rightarrow 0 + 1 + (-1)$.

Table 2.6 Some conserved quantities in elementary particle physics and their associated symmetries due to Noether's theorem

Conserved quantity	Symmetry
Four-momentum $p^\mu = (E_{tot}/c^2, \mathbf{p}_{tot})$	Spacetime translation
Electric charge Q	Gauge invariance
Baryon B and lepton L numbers	$SU(1)_B, SU(1)_L$
Only electromagnetic interaction	
Parity P	Reversion of spacial configuration
Time reversal T	Direction of time
Charge conjugation C	Matter \leftrightarrow anti-matter

Exercise 1 Which one of the four neutrinos ($\nu_e, \bar{\nu}_e, \nu_\mu, \bar{\nu}_\mu$) – cf. Table 2.4 on p. 62 – is involved in the following reactions?

- (a) (?) + $p \rightarrow n + e^+$,
- (b) (?) + $n \rightarrow p + \mu^-$,
- (c) (?) + $n \rightarrow p + e^-$.

Solution 1 Considering different conservation laws as stated above, the answers are:

- (a) $\bar{\nu}_e$ (b) ν_μ (c) ν_e .

2.5.4 Relativistic Wave Equations

The canonical starting point for quantum field theories is the variation of the classical integral of action

$$S \equiv \int_{t_1}^{t_2} dt \int d^3x \mathcal{L}(\phi, \partial_\mu \phi), \quad (2.40)$$

with *Lagrangian density* $\mathcal{L}(\phi(x), \partial_\mu \phi(x))$, which is a function of the field $\phi(x)$ with an infinite number of degrees of freedom, and its four-gradient $\partial_\mu \phi(x) \equiv \partial/\partial x_\mu = (\partial/\partial t, -\nabla)$. Performing a variation δ of the integral in (2.40) and applying Hamilton's principle ($\delta S = 0$), as well as the boundary conditions ($\delta\phi(t_1, \mathbf{x}) = 0 = \delta\phi(t_2, \mathbf{x})$) of (2.40) leads to the equation of motion for the fields, i.e. the *Euler-Lagrange equations*:

$$\partial_\mu \frac{\partial \mathcal{L}}{\partial (\partial_\mu \phi(x))} - \frac{\partial \mathcal{L}}{\partial \phi(x)} = 0. \quad (2.41)$$

The field Eq. (2.41) are covariant, if the Lagrangian density is Lorentz invariant, i.e. a Lorentz-scalar. Due to the boundary conditions of the variation, they are not changed, when a total divergence $\partial_\mu A$ with some field A is added to the Lagrangian density. The symmetries of the field equations and the quantum numbers of the elementary particles are obtained from the symmetries of the Lagrangian density.

Box 2.4 Lagrangian formulation of classical mechanics

The classical equations of motion are obtained by applying Hamilton's principle of least action. A variation δ of the integral of least action

$$S \equiv \int_{t_1}^{t_2} dt L(q^i, \dot{q}^i), \quad (2.42)$$

where $L(q^i, \dot{q}^i) = T - V$, the difference between kinetic and potential energy, depends on the generalized coordinates and velocities. Using

$$\delta S = \delta \int_{t_1}^{t_2} dt L(q^i, \dot{q}^i) = 0, \quad (2.43)$$

with fixed boundaries t_1 and t_2 of the integral in (2.5.4) yields the particle trajectories $q^i(t)$ for which the action S is minimized, i.e. stationary. These equations are called *Euler-Lagrange Equations* (see. Problem 3 on p. 107).

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q^i} = 0. \quad (2.44)$$

The same formalism is used in classical mechanics for deriving equations of motion from a classical Lagrangian function $L(q^i, \dot{q}^i)$, which is a function of generalized coordinates and velocities, see Box 2.4.

Example 4 (Lagrangian Density of Maxwell's Equations) Introducing the four-potential

$$A^\mu = (\Phi, \mathbf{A}) \quad (2.45)$$

in Minkowski space and the antisymmetric field tensor.

$$F^{\mu\nu} \equiv \partial^\nu A^\mu - \partial^\mu A^\nu = -F^{\nu\mu}, \quad (2.46)$$

with

$$F^{\mu\nu} = \begin{pmatrix} 0 & E_1 & E_2 & E_3 \\ -E_1 & 0 & B_3 & -B_2 \\ -E_2 & -B_3 & 0 & B_1 \\ -E_3 & B_2 & -B_1 & 0 \end{pmatrix}, \quad (2.47)$$

where a short-form ∂^ν of the four-gradient was used. The co- and contravariant four-gradients and the operator of the wave equation \square are defined as:

$$\partial_\nu \equiv \frac{\partial}{\partial x^\nu} = \left(\frac{1}{c} \frac{\partial}{\partial t}, \nabla \right), \quad (2.48a)$$

$$\partial^\nu \equiv \frac{\partial}{\partial x_\nu} = \left(\frac{1}{c} \frac{\partial}{\partial t}, -\nabla \right), \quad (2.48b)$$

$$\square = \partial_\nu \partial^\nu = \frac{1}{c^2} \frac{\partial^2}{\partial t^2} - \Delta. \quad (2.48c)$$

Maxwell's equations (2.3)a–d can be derived from the Lagrangian density

$$\mathcal{L} = -\frac{1}{4}F^{\mu\nu}F_{\mu\nu} - j_\mu A^\mu \quad (2.49)$$

in covariant form as

$$\partial^\lambda F^{\mu\nu} + \partial^\mu F^{\nu\lambda} + \partial^\nu F^{\lambda\mu} = \partial^{[\lambda} F^{\mu\nu]} = 0, \quad (2.50a)$$

$$\partial_\mu F^{\mu\nu} = 0, \quad (2.50b)$$

$$\partial_\mu F^{\mu\nu} = -j^\nu. \quad (2.50c)$$

In (2.50a) we have used the notation of antisymmetry brackets “[]” which are used to denote antisymmetric components of tensors.³³ Using (2.50a) and (2.50c), one obtains the covariant form of the continuity equation

$$\partial_\mu j^\nu = -\partial_\nu \partial_\mu F^{\mu\nu} = 0, \quad (2.51)$$

with the source four-vector

$$j^{\mu(x)} = (\rho(x), \mathbf{j}(x)). \quad (2.52)$$

Maxwell's equations do not change under the transformation

$$A_\mu(x) \rightarrow A_{\mu'}(x) = A_\mu(x) + \partial_\mu \Phi(x), \quad (2.53)$$

where $\Phi(x)$ is some scalar field. Due to this gauge invariance, one can impose the Lorentz gauge condition

$$\partial_\mu A^\mu = 0, \quad (2.54)$$

which does not change the fields, i.e. the physics.

In Quantum field theories, both, the mass fields (fermions) and the interaction fields (bosons) are described as operators in Hilbert space which obey certain commutation relations and manipulate the fields. The Schrödinger equation reads

$$\mathcal{H}\psi = i\hbar \frac{\partial \psi}{\partial t}, \quad (2.55)$$

and it describes non-relativistic spineless point particles, with the energy operator $\mathcal{H} = \mathcal{H}(\mathcal{P}, x) = \mathcal{H}(\frac{\hbar}{i}\nabla, x)$. In non-relativistic quantum mechanics $\mathcal{H} = \mathcal{T} + V$ where $\mathcal{T} = \frac{\mathcal{P}^2}{2m}$ is the operator corresponding to non-relativistic kinetic energy T and momentum p , respectively, i.e. $v \ll c$, and V is the potential energy. In (2.55) ψ is the wave function describing the single particle amplitude. Usually, in quantum field theory, the symbol ψ is reserved for spin 1/2 fermions and the symbol ϕ is used for spin 0 bosons. For relativistic particles ($v \sim c$), the total energy E is given by the Einstein relation $E^2 = \mathcal{P}^2 + m^2$. Thus, the square of the relativistic Hamiltonian H^2 is simply given by promoting the momentum to operator status, i.e.:

$$H^2 \rightarrow \mathcal{H}^2 = \mathcal{P}^2 c^2 + m^2 c^4. \quad (2.56)$$

³³For example, $T^{ab}_{[cd]} = \frac{1}{2}(T^{ab}_{cd} - T^{ab}_{dc})$.

Working with this operator in (2.55) and inserting the momentum operator in position space $\mathcal{P} \rightarrow -\frac{i}{\hbar}\nabla$, one yields the *Klein–Gordon equation*

$$\left(\square + \left(\frac{mc}{\hbar}\right)^2\right)\phi(x) = 0, \quad (2.57)$$

In (2.57) the box notation was introduced, cf. (2.48) on p. 71:

$$\square = \partial_\mu \partial^\mu = \partial^2 / \partial t^2 - \nabla^2, \quad (2.58)$$

Equation (2.57) is the classical homogeneous wave equation for the field $\phi(x)$. The operator \square is Lorentz invariant, so the Klein–Gordon equation is relativistically covariant, that is, it transforms into an equation of the same form, provided that ϕ is a scalar function. Thus, under a Lorentz transformation $(ct, \mathbf{x}) \rightarrow (ct', \mathbf{x}')$,

$$\phi(t, \mathbf{x}) \rightarrow \phi'(t', \mathbf{x}'). \quad (2.59)$$

The Klein–Gordon equation has plane wave solutions of the form:

$$\phi(x) = N e^{-i(Et - \mathbf{p}\mathbf{x})}, \quad (2.60)$$

where N is a normalization constant and $E = \pm\sqrt{c^2\mathbf{p}^2 + m^2c^4}$ with positive and negative energy solutions. The negative solutions for E render it impossible to interpret the Lorentz invariant ϕ as a wave function of a particle (as in non-relativistic quantum theory), because $|\phi|^2$ does not transform like a density. The spectrum of the energy operator is not bounded and one could extract arbitrarily large amounts of energy from the system by driving it into ever more negative energy states. Also, one cannot simply throw away these solutions because they are needed to define a *complete* set of states. These interpretive problems disappear if one introduces the idea of a quantized field and considers ϕ as a quantum field in the sense of a usual dynamic variable. In this case, the positive and negative energy modes are simply associated with operators that create or destroy particles.

Historically, due to the above mentioned problems in interpreting ϕ as a wave function and to define a probability density (see Problem 4), Dirac tried to find a different equation of first order with respect to time derivatives, hoping that this similarity to the non-relativistic Schrödinger equation would allow such an interpretation. It turned out that Dirac's hopes were in vain, but he did find another covariant equation which allowed for negative solutions, too. His Ansatz was a Hamiltonian of the form

$$\mathcal{H} = \sum_i^3 \alpha_i \mathcal{P}_i + \beta mc^2, \quad (2.61)$$

where \mathcal{P}_i are the three components of the momentum operator $\mathcal{P} = \frac{i}{\hbar}\nabla$. It can be shown that from the requirement $H^2 = \mathcal{H}^2 + m^2c^4$ it follows that α_i and β must be interpreted as 4×4 matrices, and the considered field ψ as a multi-component *spinor* ψ_σ on which these matrices act, thus yielding the *position space Dirac equation*

$$i\hbar \frac{\partial \psi_\sigma}{\partial t} = -i\hbar c \sum_{\tau=1}^N \left(\sum_{i=1}^3 \alpha_i \partial_{x^i} + \sum_{\tau=1}^N \beta_{\sigma\tau} mc^2 \right) \psi_\tau \quad (2.62)$$

$$= \sum_{\tau=1}^N \mathcal{H}_{\sigma\tau} \psi_\tau. \quad (2.63)$$

The following combination of matrices is useful for a symmetric formulation of (2.62) in spacetime:

$$\gamma^0 = \beta, \gamma^i = \beta\alpha_i, (i = 1, 2, 3). \quad (2.64)$$

The quantities γ^μ may be combined to define a four-vector in Minkowski space

$$\gamma^\mu = (\gamma^0, \gamma^1, \gamma^2, \gamma^3), \text{ and } \gamma_\mu = g_{\mu\nu}\gamma^\nu = (\gamma_0, \gamma_1, \gamma_2, \gamma_3), \quad (2.65)$$

thus yielding the *covariant Dirac equation*:

$$\left(i\hbar \sum_{i=0}^4 \gamma^i \partial_i - mc \right) \psi = (i\hbar \not{\partial} - mc)\psi = 0, \quad (2.66)$$

where in the second term Feynman's "dagger-notation" was used, i.e.:

$$\not{\partial} = \nabla = \sum_{\mu=0}^4 \gamma^\mu \frac{\partial}{\partial x^\mu} = \frac{\gamma^0}{c} \frac{\partial}{\partial t} + \boldsymbol{\gamma} \cdot \boldsymbol{\nabla}. \quad (2.67)$$

The Dirac equation is the equation of motion for the field operator ψ describing spin 1/2 fermions. From (2.67) one can derive a Lorentz-invariant Lagrangian density for free Dirac particles:

$$\mathcal{L} = \bar{\psi} (i\hbar \sum_{i=0}^4 \gamma^i - mc) \psi \quad (2.68)$$

where $\bar{\psi}$ is the adjoint field. Variation $\delta\bar{\psi}$ in (2.68) yields (2.62) and variation $\delta\psi$ yields the adjoint Dirac equation.

Local Gauge Symmetries

Quantum mechanical expectation values of observables

$$\langle O \rangle = \int \psi^* O \psi \quad (2.69)$$

and the corresponding Lagrange functions are invariant against phase rotations of the field function $\psi(x)$:

$$\psi(x) \rightarrow \psi'(x) = \exp(i\alpha) \psi(x) d^3x. \quad (2.70)$$

If α is a constant angle, then the $U(1)$ gauge invariance (2.70) is *global* and, according to Noether's theorem, leads to conservation of electric charge Q .

Exercise 2 (*Show that $SU(1)$ symmetry leads to conservation of charge Q*) The global $SU(1)$ symmetry rotates the phase of a field ϕ according to:

$$\phi(x) \rightarrow \phi'(x) = \exp(iQ\alpha) \phi(x), \quad (2.71a)$$

$$\phi^*(x) \rightarrow \phi'^*(x) = \exp(-iQ\alpha) \phi^*(x). \quad (2.71b)$$

The infinitesimal rotation of the fields is thus:

$$\phi(x) \rightarrow \phi'(x) = \phi(x) + \delta\phi(x) = \phi(x) + iQ(\delta\alpha)\phi(x), \quad (2.72a)$$

$$\phi^*(x) \rightarrow \phi^{*\prime}(x) = \phi^*(x) + \delta\phi^*(x) = \phi^*(x) - iQ(\delta\alpha)\phi^*(x). \quad (2.72b)$$

As $\delta\alpha$ is *not* a function of position $\delta(\partial_\mu\phi) = iQ(\delta\alpha)\partial_\mu\phi$. Demanding gauge invariance of the Lagrangian density $\mathcal{L}(\phi, \phi^*, \partial_\mu\phi, \partial_\mu\phi^*)$ and using (2.41) one obtains for arbitrary $\delta\alpha$:

$$\delta\mathcal{L} = \frac{\partial\mathcal{L}}{\partial\phi}\delta\phi + \frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi)}\delta(\partial_\mu\phi) + \frac{\partial\mathcal{L}}{\partial\phi^*}\delta\phi^* + \frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi^*)}\delta(\partial_\mu\phi^*) \quad (2.73a)$$

$$= \left[\partial_\mu \frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi)} \right] iQ(\delta\alpha)\phi + \frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi)} iQ(\delta\alpha)\partial_\mu\phi + c.c. \quad (2.73b)$$

$$= iQ(\delta\alpha)\partial_\mu \left[\frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi)} \phi \right] - iQ(\delta\alpha)\partial_\mu \left[\frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi^*)} \phi^* \right] \equiv 0. \quad (2.73c)$$

Hence, there is a continuity equation for the four-current $j^\mu = (\rho, \mathbf{j})$ of charge $Q = \int d^3x\rho$:

$$\partial_\mu j^\mu = \frac{\partial}{\partial t}\rho + \nabla \cdot \mathbf{j} = 0, \quad (2.74)$$

with

$$j^\mu \equiv -iQ \left(\frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi)} \phi - \frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi^*)} \phi^* \right). \quad (2.75)$$

Hence, in a closed system, charge Q is conserved ($\frac{dQ}{dt} = 0$). Global $U(1)$ gauge symmetries lead to conserved quantum numbers (in this case, charge Q).

If α is a function of space, i.e. $\alpha = \alpha(x)$, (2.70) is a *local* gauge transformation, i.e. one demands that the expectation values (2.69) are invariant against a *local* choice of phase factors, cf. Fig. 2.14. A local, position dependent phase transformation

$$\phi(x) \rightarrow \phi'(x) = \exp(iQ\alpha(x))\phi(x) \quad (2.76)$$

yields

$$\partial_\mu\phi(x) \rightarrow \partial_\mu\phi'(x) = \exp(iQ\alpha(x)) [\partial_\mu\phi(x) + iQ(\partial_\mu\alpha(x))\phi(x)], \quad (2.77)$$

which is not form-invariant. The Lagrange density can be made covariant by substituting the partial derivative by a gauge invariant derivative:

$$\partial_\mu \rightarrow D_\mu \equiv \partial_\mu + ieQA_\mu(x), \quad (2.78)$$

where $q = eQ$ is the electric charge of the field $\psi(x)$ (electric charge, Q: quantum number), and A_μ is the four-potential (2.45) which transforms under phase rotations as

$$A_\mu(x) \rightarrow A'_\mu(x) = A_\mu(x) - \frac{1}{e}\partial_\mu\alpha(x). \quad (2.79)$$

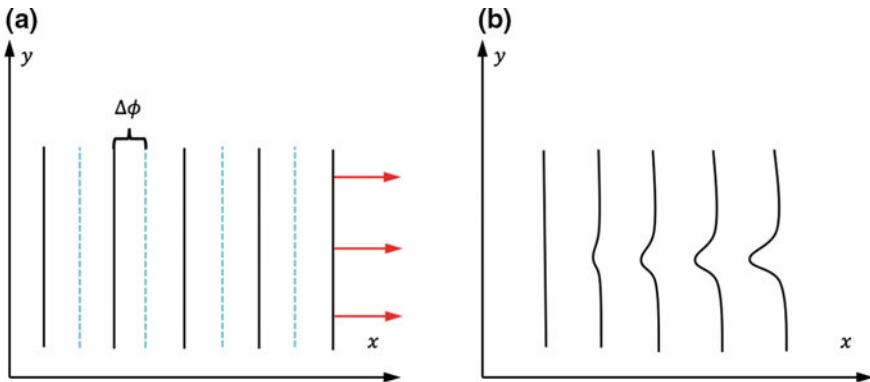


Fig. 2.14 Illustration of a global gauge symmetry versus local gauge symmetry with plane waves propagating in x direction. In **a** a global gauge transformation changes the phase of the plane wave at every position (x, y) by the same amount $\Delta\phi$. Thus, the plane wave fronts move to the *dotted lines* and there is on average no effect on the shape of the wave propagation. In **b** a local phase transformation changes the wave fronts differently at different locations (x, y) , i.e. $\Delta\phi = \Delta\phi(x, y)$. The transformed wave is no plane wave any more. This change of shape is explained by the introduction of external interactions

With this gauge transformation one obtains:

$$D_\mu \phi(x) = \partial_\mu + ieQA_\mu \rightarrow D'_\mu \phi'(x) \quad (2.80a)$$

$$= (\partial_\mu + ieQA'_\mu(x)) \exp(iQ\alpha(x))\phi \quad (2.80b)$$

$$= \exp(iQ\alpha(x))[\partial_\mu + iQ\partial_\mu\alpha(x) + ieQA_\mu(x)] \quad (2.80c)$$

$$- iQ\partial_\mu\alpha(x)\phi \quad (2.80d)$$

$$= \exp(iQ\alpha(x)) [\partial_\mu + iQe\partial_\mu A_\mu(x)]\phi(x) \quad (2.80e)$$

$$\equiv \exp(iQ\alpha(x)) D_\mu \phi(x). \quad (2.80f)$$

Hence, the invariance of $\phi^* D_\mu \phi$ under $U(1)$ phase transformations is achieved by introducing an interaction for the field ϕ , cf. Fig. 2.14.

The coupling of the matter field ϕ and the gauge interaction field $A_\mu(x)$ is uniquely determined by demanding *local gauge invariance*, i.e. by introducing the gauge invariant derivation

$$D_\mu \phi(x) = \partial_\mu \phi(x) + ieQA_\mu x\phi(x), \quad (2.81)$$

also called *minimal gauge invariant coupling*. With the above derivations using $U(1)$ gauge invariance, one obtains for the Lagrangian density of QED, which describes the interaction of photons (the electromagnetic interaction field) with fermions:

$$\mathcal{L}_{QED} = \mathcal{L}_{fermion}^{free} + \mathcal{L}_{photon}^{free} + \mathcal{L}_{interaction} \quad (2.82a)$$

$$= -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{\psi}(i\hbar \sum_{i=0}^4 \gamma^i - mc)\psi - ej^\mu A_\mu \quad (2.82b)$$

$$= -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + \bar{\psi}i\hbar\sum_{i=0}^4\gamma^i\partial_\mu\psi - m\bar{\psi}\psi - ej^\mu A_\mu \quad (2.82c)$$

$$= \mathcal{L}_{fermion}^{kin} + \mathcal{L}_{photon}^{kin} + \mathcal{L}_{fermion}^{mass} + \mathcal{L}_{interaction} \quad (2.82d)$$

$$= T - V. \quad (2.82e)$$

For a detailed discussion of non-Abelian $SU(2)$ and $SU(3)$ gauge symmetries within the $U(1) \times SU(2) \times SU(3)$ gauge symmetry of the standard model – usually discussed in the language of differential geometry (local sections of principal fiber bundles) – the reader is referred to specialized literature, see Sect. 2.5.5.

The standard model is one of the best tested theories in physics. However, it too, can only be an approximation of a more general theory, due to several problems such as:

- There are more than 30 free parameters (e.g. particle masses and constants of nature) that have to be determined by measurements.
- Why does dark matter exist and why is there more normal matter than antimatter in the known universe?
- The gravitational force is completely excluded.

We end this section with an appropriate quote by Sheldon L. Glashow (Nobel prize 1979), who writes in “The Charm of Physics” about the motivation to do fundamental physics [145]:

“Do we do fundamental physics to explain the world about us?” is a question that is often asked. The answer is NO! The world about us was explained 50 years ago or so. Since then, we have understood why the sky is blue and why copper is red. That’s elementary quantum mechanics. It’s too late to explain how the work-a-day world works. It’s been done. The leftovers are things like neutrinos, muons, and K -mesons – things that have been known for half a century, still have no practical application, and probably never will [...] So it is that we are not trying to invent a new toothpaste. What we are trying to do is to understand the birth, evolution, and fate of our universe. We are trying to know why things must be exactly the way they are. We are trying to expose the ultimate simplicity of nature. For it is in the nature of elementary-particle physicists (and some others) to have faith in simplicity, to believe against all reason that the fundamental laws of physics, of nature or of reality are in fact quite simple and comprehensible. So far, this faith has been extraordinarily productive: Those who have it often succeed; those without it, always fail.

(Sheldon L. Glashow [145], 1991, p. 109)

2.5.5 Suggested Reading

For a recent description of latest accelerator experiments at DESY, PETRA and HERA, see e.g. [154]. Elementary introductions into the standard model can be found in Close [155], Halzen and Martin [156], or Nachtmann et al. [157]. Standard references to relativistic quantum theory and gauge theories are Aitchison and Hey [158], Sakurai [159], or Mandl and Shaw [160]. A classic is Bjorken and Drell [161, 162]

which is very succinct and quickly advances from chapter to chapter. An excellent book covering the history of elementary particle physics from the 1960s to 1970s is edited by Hoddeson et al. [92]. A very paedagogical introduction to ideas of elementary particle physics is achieved in the classic by Dodd [163]. Some good popular books on elementary particle physics and the principles of field theory are Okun [164] and Fritsch [165]. Lattice gauge theory, which was developed in a ground breaking work by M. Creutz [166], in order to be able to perform Monte Carlo simulations of the basic equations of quantum chromodynamics on a lattice, is discussed in Montvay and Münster [167].

2.6 Computer Science, Algorithms, Computability and Turing Machines

Computer science provides a multitude of concepts, methods of description, models, algorithms, or simply ideas which serve to the general purpose of visualizing, organizing and analyzing complex phenomena of reality. In principle, the modeling strategies are basically the same as in the natural sciences. From identifying the most important contents of a real system one derives an abstract model (abstraction). The model might be a formula, an equation, an algorithm, an automata, a graph, etc., cf. Fig. 2.15.

Creating a model of the real complex system allows for saving the model in binary form on a computer with subsequent simulation and analysis based on some algorithm. From the input/output properties of the model system one can make predictions for the behavior of the real system (interpretation). The most important point in this process is the identification of the essential features that characterize the real system in a unique way. Oversimplification is a common weakness of this modeling process.

Computer science is dominated by algorithms. What is an algorithm?

An intuitive notion of an algorithm is a step-by-step procedure (a *finite* set of well-defined instructions) for accomplishing some task which, given an initial state, will terminate in a defined end-state, cf. Fig. 2.16.

An algorithm allows for “mechanically” accomplishing some task by following the step-by-step instructions, even without any intellectual insight into the procedure or the problem to be solved. The computational complexity and efficient implementation of algorithms are very important in scientific computing, and this usually depends on suitable data structures. Algorithms have to be written in a way, such that each step is comprehensible. Hence, the single basic steps have to be a subset of an agreed upon set of *elementary operations*. Algorithms that are implemented on computers are formulated in suitable computer languages and are subsequently translated (compiled) into a machine code which actually consists only of elementary single steps. An algorithm can be written down in several ways, e.g. as *written text* with *step-by-step directions* as in Algorithm 2.1, as *pseudo-code* revealing the elements

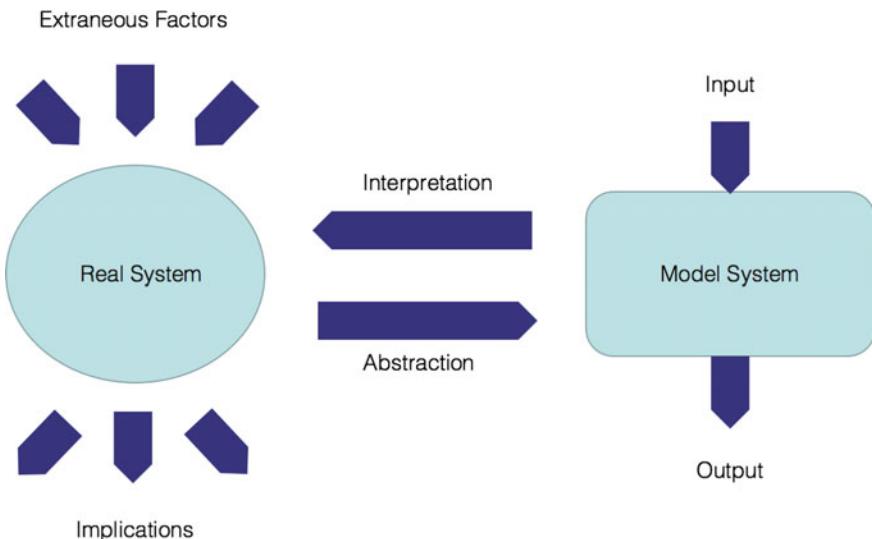


Fig. 2.15 Modeling of a real complex structure in computer science. In computer simulations, one utilizes numerical models. These models can come in many shapes, sizes, and styles. It is important to emphasize that such a model is not the real world but merely a human construct to help one better understand real world systems. In general, all models have an information input, an information processor, and an output of expected results. In an abstraction process the respective model system is extracted from the essential parts of the real system. From the interpretation of the input/output behavior of the model system one draws conclusions on the behavior of the real system



Fig. 2.16 An algorithm is a set of instructions which – applied to an initial state – develops the system into a defined end-state

of a computer language that have to be used when implementing the algorithm, cf. Algorithm 2.2, or as *explicit* code sample in a computer language, cf. Algorithm 2.3.

We start out with some examples of algorithms, some of which have been known for a long time. The *sieve of Eratosthenes* (see e.g. [168]) is one of the oldest known algorithms. It used for determining all prime numbers up to some specified integer.

Example 5 (Application of Algorithm 2.1) We want to apply this algorithm to the first 17 integers:

- Steps 1 and 2:
list $A = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17\}$, list $B = \{2\}$.
- Step 3:
2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17.

Algorithm 2.1: The sieve of Eratosthenes

// This is one of the oldest known algorithms which is used for finding all prime numbers up to a specified integer.

Input: Finite list $A \in \mathbb{N}$ of integers, starting with 2.

Output: All prime numbers contained in the list A .

begin

1. Write a list (called A) of numbers from 2 to the largest number you want to test for primality.
2. Write the number 2, the first prime number, in another list for primes found. Call this list B.
3. Strike off 2 and all multiples of 2 from list A.
4. The first remaining number in the list is a prime number.
Write this number into list B.
5. Strike off this number and all multiples of this number from list A.
The crossing-off of multiples can be started at the square of the number, as lower multiples have already been crossed out in previous steps.
6. Repeat steps 4 through 6 until no more numbers are left in list A.

end

Thus $A = \{3, 5, 7, 9, 11, 13, 15, 17\}$.

- Step 4:
 $B = \{2, 3\}$.
- Step 5:
 $A = \{3, 5, 7, \underline{9}, 11, 13, \underline{15}, 17\} = \{5, 7, 11, 13, 17\}$.
- Step 6:
Repeating Steps 4 to 6 finally yields: $B = \{2, 3, 5, 7, 11, 13, 17\}$.

A different example of a well-known algorithm is Euclid's algorithm which determines the largest common divisor of two natural numbers a and b , see the pseudo-code in Algorithm 2.2.

Algorithm 2.2: The largest common divisor (Euclid's algorithm)

Input: A finite set $A = \{a_1, a_2, \dots, a_n\}$ of integers

Output: The largest element in the set

```
max ←  $a_1$ 
for  $i \leftarrow 2$  to  $n$  do
    if  $a_i > max$  then
        max ←  $a_i$ 
    end
end
return max
```

After input of the numbers a and b it is tested whether $b = 0$. If this is not true then a is set to the value of b and the new value of b is the rest of the division of (the previous) a value by b . This is repeated until the rest of the division is 0. Then the searched for result is a .

Sometimes an algorithm which is derived directly from a mathematical definition or a formula is not necessarily the most efficient one. An example of this fact is provided on p. 84 in Example 6.

2.6.1 Recursion

An important principle in computer science for the description of algorithms, functions, or data structures is *recursion*. The basic principle of recursion is that a function calls itself. Recursion is also one of the most important tools for the implementation of efficient search- and sort-algorithms which are also abundant in computational materials science, usually in the form of look-up tables for the determination of interacting particles or finite elements of a system. Almost all search algorithms need a direct access to all elements that are to be sorted. When using a list, there is no such immediate access; thus one uses an index table which contains references to consecutive elements. Generally speaking, search algorithms are based on the strategy that they do not compare *all* elements of a data set S but only certain elements $s_i \in S$ which distinguish this data set from other data sets in a unique way. These specific elements are called “keys”. Thus, in a search, only the keys are important.

Exercise 3 (Write a recursive version of Euclid’s algorithm. The function may not contain any FOR, GOTO or WHILE construct)

Solution 2 We use the key word PROCEDURE for calling a subroutine named “Euclid” which in turn calls itself in the following pseudo-code:

```

1 PROCEDURE Euclid(int a, b)
2   if b=0 then return a;
3   else return Euclid (b, a mod b)
4 END

```

The concept of recursive functions was introduced by John McCarthy, the inventor of the programming language LISP [169]. The first step in writing a recursive function is a specification of its input/output behavior. When writing the function in the next step one usually uses an if-statement to catch the beginning of the recursion.

Exercise 4 (Write a recursive function that calculates the faculty $n!$)

Solution 3 We use the fact that $n! = n \times (n-1)!$, i.e. the main problem is successively reduced to ever smaller problems until the trivial case ($0! = 1$) occurs. In the language C++ or C this reads

```

1 long Faculty(long n)
2 {
3     if (n < 0) return (0);
4     if (n == 0) return (1);
5     return (n * Faculty(n-1));
6 }
```

In Exercise 3, the recursive mathematical formula is identical with the implementation (besides syntactic details). The first *if*-statement makes sure, that recursion is only called with positive numbers. One great disadvantage of recursive functions is the additional overhead in terms of computation time as with each recursive call the function's return address has to be saved and memory for all local parameters has to be provided for. This disadvantage can be avoided by using *rest-recursive functions*, in particular if the used compiler supports “last-call optimization”. A rest-recursive function call is implemented in C or C++ as displayed in Algorithm 2.3.

This function is illustrated in Fig. 2.17. In Fig. 2.17a the usual recursive function calls are displayed. This function provides at each call the case $(n - 1)$ and multiplies it with n . After recursion to the lowest point the function goes upwards again and provides each respective result to the calling function. Thus, in each recursive step both, the arguments and the respective result are important. In Fig. 2.17b the rest-recursive function call is shown. Here, at each call, the value of n is passed but also the current state of the faculty calculation. Thus, in each recursive step the calculation proceeds by one step until it is finished at the lowest step. Then the final result is simply returned successively to the calling functions. The passed parameters are not needed any more and there is no need to save a return address of the calling function, as the final result can be passed directly to the very first calling function.

Algorithm 2.3: Rest-recursive function

```

1 /* Calculates the faculty n! rest-recursively */
2 long Faculty(long n, long y)
3 {
4     if (n < 0) return (0);
5     if (n == 0) return (1);
6     if (x == 1) return (y);
7     return (Faculty(n-1, y * (n-1)));
8 }
```

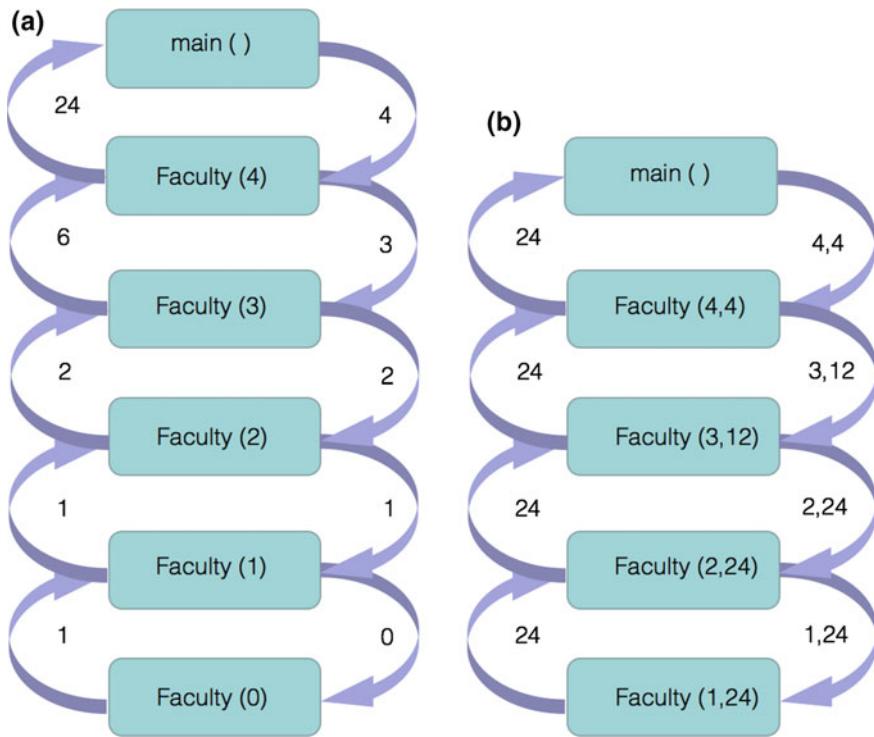


Fig. 2.17 Illustration of a recursive function call **a** according to Example 2 versus a rest-recursive function call **b** according to Algorithm 2.3

2.6.2 Divide-and-Conquer

Divide-and-Conquer is a recursive programming technique which can be applied to some problems, provided that they have a suitable structure. If a problem of size n is to be solved, then this problem is split into two (or more) sub-problems of size $n/2$ which can then be solved by using the same recursive algorithm. The two solutions can then be merged to a solution for the original problem. Examples for applications of this strategy for problem solving can be found in some common search algorithms, such as *merge-sort*, *bubble-sort*, *quick-sort* or *binary search*, see Sect. 2.6.6 on search algorithms. It can be shown that these algorithms require an effort which is no better than $\mathcal{O}(n \log n)$ in the worst case. With quick-sort in particular, this is only true *on average*. In single cases the effort can be $\mathcal{O}(n^2)$. For an introduction of the \mathcal{O} -notation see Sect. 2.6.6.

Algorithm 2.4: Recursive calculation of a power

```

1 PROCEDURE Power(a,n)
2 If n = 0 then return 1
3   else if even(n) then return squared(a,n DIV 2)
4     else return a * squared(Power(a,(n-1) DIV 2)) END
5   END

```

Example 6 (An algorithm for the power of a number a)

If a power such as a^n is to be calculated for a larger number n , it is advisable, *not* to multiply a n times with a . It is way better to use the following recursion:

$$a^n = \begin{cases} 1 & : \text{if } n = 0, \\ (a^{n/2})^2 & : \text{if } n > 0 \text{ and } n \text{ even,} \\ a \times (a^{(n-1)/2})^2 & : \text{if } n < 0 \text{ and } n \text{ uneven.} \end{cases} \quad (2.83)$$

Equation (2.83) is provided as pseudo-code in Algorithm 2.4.

The construct `even(n)` in Algorithm 2.4 tests whether the value of n is even and `squared(n)` calculates the square of n . With each recursive function call the number n is halved; thus when calculating the n -th power one needs at the most $\log_2 n$ function calls.

Dynamic Programming

When using recursive function calls, one usually applies a “top-down” principle. A system of size n is split into tasks of smaller size, e.g. $(n - 1)$, $(n - 2)$ or $n/2$ and the corresponding function is called recursively. These subsequent recursive calls are usually independent of each other; thus, it might happen, that the *same* recursive call is done several times within the recursive structure.

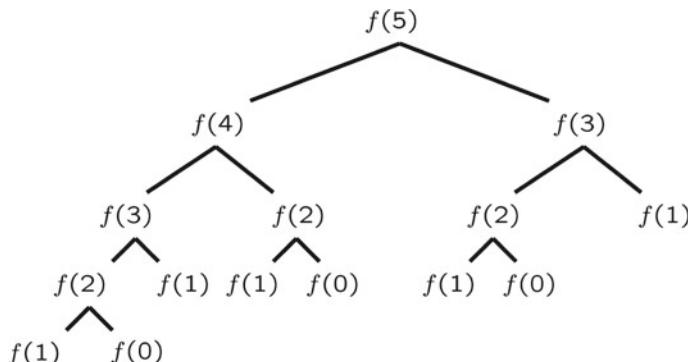


Fig. 2.18 Exponential effort for a recursive calculation of $f_B(5)$

Algorithm 2.5: Recursive calculation of the Fibonacci numbers

```

1 PROCEDURE f_B(n)
2   if n = 0 then return 0
3   else if n = 1 then return 1
4   else return f(n-1) + f(n-2) END
5 END

```

In recursion of $f_B(5)$ the value of $f(5)$ is calculated once, $f(4)$ two times, $f(3)$ three times, $f(2)$ five times and $f(0)$ three times as depicted in Fig. 2.18.

This means that the recursion of $f_B(n)$ leads to an exponential effort. In this case it is more efficient to store calculated values and reuse them if needed again. This principle of working with a table in which calculated values are stored for later reuse is called *dynamic programming*. The table is filled “bottom-up”, i.e. in the order $i = 0, 1, 2, \dots, n$.

Example 7 (The Fibonacci sequence) A standard example for recursion is the calculation of the Fibonacci numbers

$$f_B(n) = f(n - 1) + f(n - 2), f(0) = 0. \quad (2.84)$$

A pseudo-code for a corresponding recursive algorithm is shown in Algorithm 2.5.

Example 8 (Pairwise alignment in biological sequence analysis) An important example for the use of dynamic programming techniques is the *editing distance* between two words $(a_1 a_2 \dots a_i)$ and $(b_1 b_2 \dots b_j)$ which is defined as the minimum number of elementary editing actions (“delete”, “insert”, “substitute”) necessary to transform the first word into the second one. For example, the words “ANANAS” and “BANANA”. Here, the editing distance is two, because:

ANANAS → BANANAS (insert),
BANANAS → BANANA (delete).

The editing distance between words is of great importance in genome research when comparing alignments of sequences (or parts of them) and then deciding whether the alignment is more likely to have occurred because the sequences are related, or just by chance. Box 2.5 shows the sequence alignments to a fragment of human α -globin and β -globin protein sequence (taken from the SWISS-PROT database identifier HBA_HUMAN.³⁴) The central line in the alignment indicates identical positions with letters, and “similar” positions with a plus sign.

In comparing DNA or protein sequences one is looking for evidence that they have diverged from a common ancestor by a process of mutation and selection. The basic

³⁴www.expasy.ch/sprot.

Box 2.5 Sequence alignment to a fraction of human α - and β -globin

HBA_HUMAN	GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAKL
G+	+VK+HGKKV A++++AH+D++ +++++LS+LH KL
HBB_HUMAN	GNPKVKAHGKKVLGAFSDGLAHLNDNLKGTFATLSELHCDKL

mutational processes that are considered are *substitutions*, which change residues in a sequence, and *insertions* and *deletions*, which add or remove residues. Insertions and deletions are together referred to as *gaps*. Natural selection has an effect on this process by screening the mutations, so that some sorts of change may be seen more than others. The editing distance of sequences can be used to evaluate a scoring for the pairwise sequence alignment. The total score that is assigned to an alignment will be a sum of terms for each aligned pair of residues, plus terms for each gap. In a probabilistic interpretation, this will correspond to the logarithm of the relative likelihood that the sequences are related, compared to being unrelated, cf. [8]. We will discuss pairwise alignment in a later chapter as an important application in computational genome research.

Greedy Algorithms

With dynamic programming, a table containing all hitherto found optimal solutions is used and updated in each step. *Greedy algorithms* do not make use of such a table; rather, the decision about the next solution component is based solely upon the *locally* available information. The possible next steps for the solution of a problem are compared with an evaluation function. This is to say that a decision for the next solution step among the available alternatives is based on maximizing (or minimizing) some figure of merit which in turn is based on comparing the locally available next steps. There is no *global* criterion as there is no list containing the history of solution steps. This explains the naming of the algorithm: At each step one simply continues into the direction which – from a local perspective – is the most promising one. In many cases, the greedy strategy results in an acceptable, albeit not optimal solution. Thus, this kind of algorithm (also called *Greedy-heuristics*) is mostly used for problems, where no other comparably efficient algorithms are known. A typical application would be the *traveling salesman problem*, (see Sect. 2.6.3) which can be solved with a Greedy-heuristic.

2.6.3 Local Search

In the previously discussed algorithms the problem solving strategy has always been a partition of the total problem of size $n = (a_1, a_2, \dots, a_n)$ into sub-problems $i = (a_1, a_2, \dots, a_i)$ ($i < n$) of smaller size. A different strategy is adopted with algorithms that use a *local search* strategy. Here, one starts with solutions of the complete problem of size n and modifies these by small manipulations (in genomics

they are called “mutations”), e.g. by modifying only *one* component of the solution:

$$(a_1, a_2, \dots, a_{j-1}, a_j, a_{j+1}, \dots, a_n) \longrightarrow (a_1, a_2, \dots, a_{j-1}, \bar{a}_j, a_{j+1}, \dots, a_n). \quad (2.85)$$

This strategy is based on searching for a new solution within the neighborhood of the previous one. The new solution should have improved properties compared with the old one. What “neighborhood” means depends on the context of the problem; e.g. in physics this might simply be some neighborhood of a system at a point in Γ phase space, cf. Chap. 6. Mutations are usually chosen randomly.

Example 9 (Traveling Salesman Problem) Consider a set of n points in R^2 . Which one is the graph which yields the *shortest* connection of all points?

As a first attempt one might guess one complete solution, i.e. an arbitrary connection of all points. This temporary solution is then subsequently improved by identifying crossing connection lines (edges) and removing them by local mutations, cf. Fig. 2.19.

The initial configuration can be generated either at random or by a simple Greedy-heuristic.

A principle problem with this strategy is, that there is no guarantee to find the optimal solution by lining up local improvements, demanding that each single mutation improves some figure of merit (in this case: minimizes the traveled distance).

The improvements are done as long as there are no local moves left which could further improve the current solution. This problem is typical for molecular dynamics simulations, e.g. in a micro-canonical ensemble (i.e. one, that keeps the total energy of the system constant). Under some unfavorable circumstances it might occur that the system is “trapped” in a local energy minimum; thus the “true” equilibrium state, i.e. the global minimum energy state is not attained by the system. This is illustrated in Fig. 2.20.

Sometimes it is better to allow for temporary local deterioration on the path to the optimal global solution. Such a strategy is described in the next section and is one of the most widely used principles in computational physics. Yet another strategy is to start the local search process many times with random initial configurations as discussed in Sect. 2.6.4.

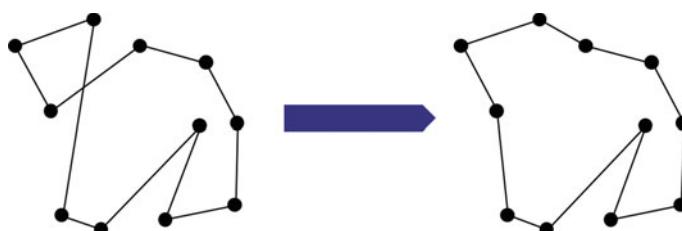


Fig. 2.19 The traveling salesman problem solved by a local search strategy for $n = 11$

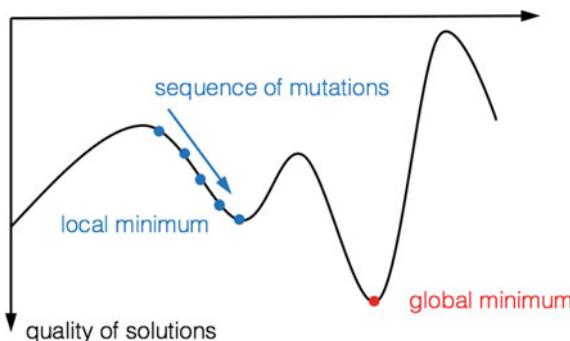


Fig. 2.20 Illustration of the *local search* principle. The sequence of mutations *always* improves some figure of merit in each single step. In physical applications, the quality of successive solutions might be determined by minimizing the total energy of the system. Thus, the system can be “trapped” in a *local* minimum state of energy which however is not the searched (optimal) *global* minimum

2.6.4 Simulated Annealing and Stochastic Algorithms

A method to overcome the discussed fundamental problem of local search in Sect. 2.6.3 is the use of random numbers. The use of random numbers in algorithms is a sometimes surprisingly efficient method for solving problems. Algorithms that make use of random numbers are generally termed *stochastic algorithms*. In contrast to this, algorithms which do not use random numbers and which – at all times – have only *one* possible next step, are called *deterministic algorithms*. Many stochastic algorithms are used in the field of algebra and number theory. In the natural sciences, so-called *Monte-Carlo algorithms* which contain a random element for the acceptance of new system states are very common and are often used in materials science, solid state physics and thermodynamics, cf. our discussion in Chap. 6. A very simple form of a stochastic algorithm would be to randomly shuffle some numbers in a list which are subsequently used as input for the quick-sort algorithm. By statistically permuting the list of numbers which is to be sorted by quick-sort results in an efficiency of $\mathcal{O}(n \log n)$ for every function call (instead of an *average* efficiency of $\mathcal{O}(n \log n)$).

A mutation of a system (in physics also called “trial move” or simply “trial”) is accepted with a certain probability despite its worse quality in comparison with the solution in the original state. The probability of accepting a worse solution depends on two parameters Δ and T where Δ is the difference of the quality of the old versus the new solution, that is, if the new solution is much worse than the previous one, then the acceptance probability $P(\Delta, t)$ is accordingly small. The second parameter T is called *temperature*. Initially, one starts at a high temperature T which then gradually decreases and approaches zero. That is, at the beginning, very large deteriorations of the solutions are still accepted with a certain probability P which decreases during the solution process. In the end, the procedure acts as a local search which accepts

only quality improvements as a new system state. For the probability of acceptance the function in (2.86) is used.

$$P(\Delta, T) = \begin{cases} 1 & : \text{new solution is better than the previous one ,} \\ \exp(-\Delta/T) & : \text{else ,} \end{cases} \quad (2.86)$$

Algorithm 2.6 provides a generic pseudo-code scheme of the Metropolis algorithm, respectively simulated annealing.

Algorithm 2.6: Simulated annealing, Metropolis Algorithm (Version I)

```

1 Function K is some cost function which has to be
2   minimized.
3 tFinal and k are some constants.
4 rnd is a uniformly distributed random number
5   in the interval [0,1].
6
7 Set t := initial temperature
8 Set a := initial solution
9 while ( t > tFinal ) DO
10   for i:= 1 TO k DO
11     Choose a mutation b from a at random
12     if K(b) < K(a) then a:= b END
13   else if rnd <= exp[-(K(b)-K(a))] THEN a:= b END
14   END
15 t := t * 0.9
16 END
17 OUTPUT a

```

The strategy of using a time-dependent (i.e. dependent of the number of mutations) acceptance probability is similar to many physical processes. One starts at a high temperature and then slowly cools down the system which gradually approaches an equilibrium – e.g. crystalline – state. This crystalline state corresponds to an optimal solution of the system and explains the terminology “simulated annealing” for this simulation strategy.

2.6.5 Computability, Decidability and Turing Machines

Computability is a branch of computer science that deals with principal questions such as:

- How can one formalize the concept of “algorithm”?
- What is an a computable function (an algorithmically solvable problem)?
- Are there any not-computable functions?

- Are there any problems that are well-posed but which are not computable by any algorithm?

An *intuitive notion* of an algorithm, respectively “computability” contains the following elements [170]:

- A *finite* description,
- Unique rules for solving a problem, i.e. in every situation the next step is uniquely defined,
- A clear definition of input/output behavior of the algorithm,
- A solution after a *finite* number of steps.

Any description of an algorithm – generally done with some computer language – uses a finite alphabet; thus, there is only a *finite* number of descriptions of algorithms. On the other hand, there is an uncountable number of functions $f : \mathbb{N} \rightarrow \mathbb{N}$. The following question arises: For how many of this uncountable number does no algorithm exist, i.e. which of these functions are not computable? Based on the above intuitive notion of an algorithm this question cannot be answered.

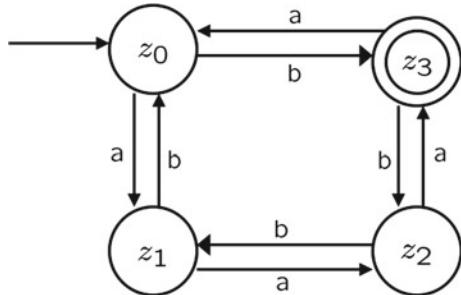
Automata

In order to find answers to these questions, the notion of an algorithm as a dynamic process that leads from state to state, was formalized in terms of *deterministic finite automata* (DFAs) which can be represented as directed graphs. Graph theory itself is a powerful modeling tool – abundant in computer science – for the visualization of relations between objects. Each knot of a graph represents a certain state which can be assumed by the considered automaton. The knots are connected by edges which are labeled with a symbol from some working alphabet, i.e. the set of symbols that can be read by the automaton. By reading a symbol the automaton goes from one state to a different one. Any finite, non-empty set can be considered to be a working alphabet. The elements of an alphabet are then called characters, letters or symbols. The sets $\Sigma = \{a, b, c, d, e, f, g, h\}$, $\Sigma = \{0, 2\}$ or $\Sigma = \{\text{condition, procedure, if, then, else, begin, end, while, ...}\}$ are examples of alphabets. A “word” is a combination of elements of an alphabet. For example $bbiaaaaacf$ is a word on the alphabet Σ . The length $|l|$ of a word, e.g. $|bbiaaaaacf| = 9$ is the number of symbols which are included in a word. The set of all words on an alphabet Σ is denoted as Σ^* .

Example 10 Let the alphabet $\Sigma = \{a, b\}$. An example for a DFA would be the graph in Fig. 2.21.

This automaton has the four states z_0, z_1, z_2, z_3 . The start state is indicated by the arrow to the first state z_0 . The end state(s) are denoted by a double circle (in this case only z_3). An automaton such as in the above example processes a “word” w by consecutively reading the symbols one at a time and performing the corresponding changes of state. If the state after processing the whole word is the end state, then

Fig. 2.21 Illustration of a deterministic finite automaton as a directed graph



the word is said to be “accepted”. For example, the above automaton accepts the word $bbbaa$ going through states $z_0, z_3, z_2, z_1, z_2, z_3$ but not e.g. aab . The set of all accepted words is called the accepted language and is denoted as $T(M)$, i.e.:

$$T(M) = \{w \in A^* \mid \text{the automaton } M \text{ accepts the word } w\} . \quad (2.87)$$

For an introduction into the notation of sets such as in (2.87) the reader is referred to Chap. 3.

DFAs are models for simple “computers”. They are equipped with a memory in the sense that each respective state of the automaton is a piece of finite information. A consequence of this is that some languages such as $\{a^n b^n \mid n \geq 0\}$ cannot be processed. One step to reduce this restriction of automata is the use of a memory which is not restricted by the number of states. Such automata are called *cellar machines*.

Cellar machines have no principal limitation in the number of states they can assume. They are equipped with a stack structure (also called “LIFO” structure – “last in first out”) as memory access. This means that the piece of information that has been stored *last* will be accessed *first* in reading mode, cf. Fig. 2.22. A LIFO structure is an example of an abstract data type (see Problem 5) abundant in computer science which is also often used for book-keeping of interactions between particles or finite elements in implementations of simulations programs for fluids or solids.

If the cellar memory is empty, there may be no memory access in read mode. This is indicated by the sign “#” at the bottom of the memory. This is the lowest sign in the cellar memory and when this is read it indicates that the memory (apart from “#” itself) is empty. One step of a calculation with a cellar machine can be described as follows: The automaton reads a sign from the input tape. Depending on the current state of the automaton and of the sign read, the uppermost sign in the cellar memory is substituted by a sequence of cellar symbols which also may be empty. Similar to DFAs, a graphical representation of the change of states is used for cellar automata, cf. Fig. 2.23. The edge in this case reads: If the cellar automaton is in state z and the current input character is a with A being the top character, then the automaton switches to state z' in the next step and substitutes A with $B_1 B_2 \dots B_k$.

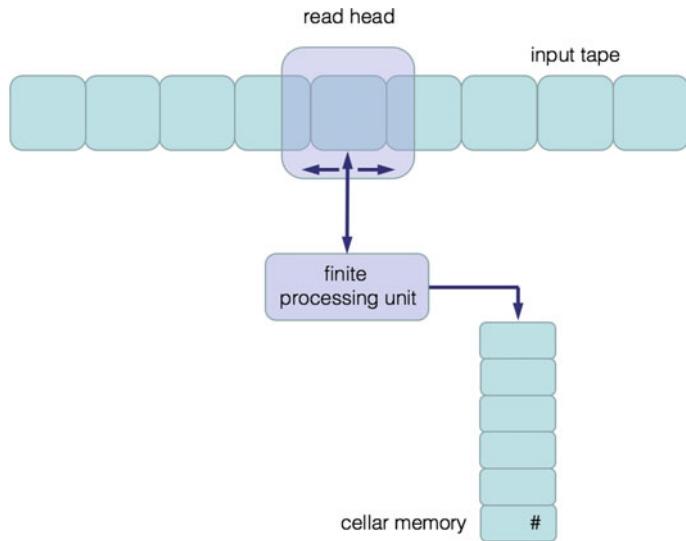


Fig. 2.22 Illustration of a cellular automaton

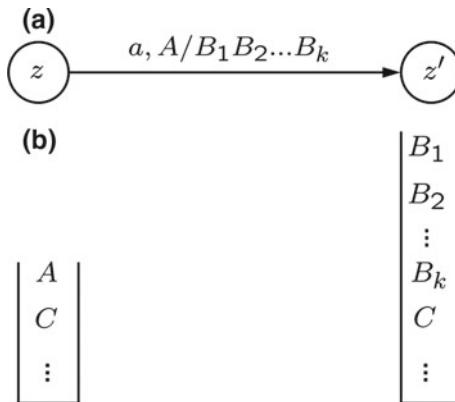
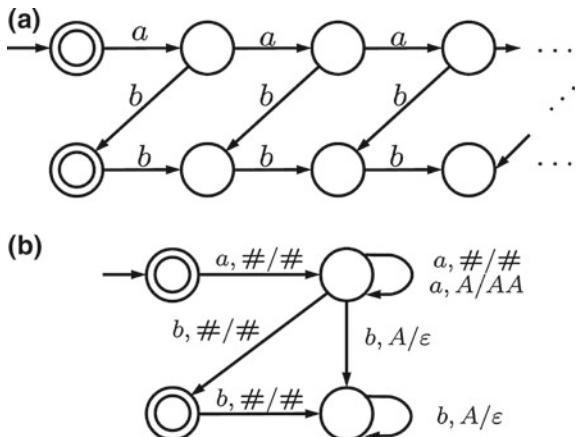


Fig. 2.23 Graphical representation of the change of states in a cellular automaton. In **a** the notation for a transition from state z to state z' is depicted with A being the top character in the cellar memory and the sequence B_1, B_2, \dots, B_k being the next characters to be read by the automaton. In **b** the states of the cellar memory before and after reading the sequence of characters are depicted

In a mathematical notation one can describe the transition of a cellular machine from state z to state z' with a function δ :

$$\delta(z, a, A) = (z', B_1 B_2 \dots B_k) . \quad (2.88)$$

Fig. 2.24 The language $S = \{a^n b^n\}$ which is accepted by a cellar automaton. In **a** the notation for an infinite DFA is shown and in **b** the graphical representation of a cellar automaton accepting S is displayed



Cellar automata can express a larger set of languages than simple DFAs, e.g. the language $S = \{a^n b^n \mid n \geq 0\}$ which is not accepted by any finite automaton, can be accepted by a cellar automaton.

Example 11 (Draw the language $a^n b^n$ accepted by a cellar automaton in graphical representation) The solution is depicted in Fig. 2.24.

The cellar automaton in Example 11b works deterministic. Each read a (except the first one) is written as A into the cellar memory. At any one time when b 's are read, an A is removed from memory. As soon as no input characters are left on the tape, the cellar memory is empty, i.e. the lowest character “#” in the cellar memory is recognized.

A cellar automaton can only write or read at the end of the cellar memory. The next logical step in the formalization of the notion of a computer removes the restriction of the LIFO structure of cellar automata, i.e. one introduces automata with a *random access memory* on all memory cells. Hence, the cellar memory is substituted by a sequential memory in the form of a memory tape which can be read or labeled by a read/write head. There is no additional “computing power” when making a distinction between a working-tape and an input-tape. Thus, input is read from *one* tape and results are written on the same tape by use of a finite read/write head which can move freely and stepwise in both directions along the tape, cf. Fig. 2.25. The tape can only be altered at the current position of the read/write head. Such a generalized model of an automaton is called a *Turing machine*.

Turing Machines

The Turing machine is a fundamental concept for a simplest, memory-based computability model. Computation in this model means stepwise modification of the memory content. This can be represented by a transition function δ , e.g.

$$\delta(z, a) = \delta(z', b, x) . \quad (2.89)$$

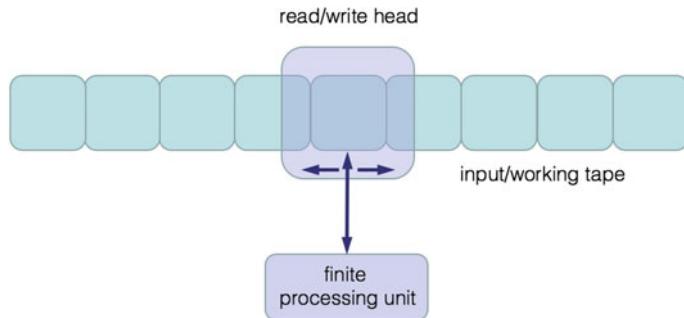


Fig. 2.25 A Turing machine with read/write head that moves freely and stepwise along a working/input tape

In this case, the Turing machine in state z reads a character a and then assumes state z' substituting a by b where a and b are characters of the working alphabet. After reading of a character, the Turing machines moves the read/write head by one step which is denoted by $x \in M = \{L, R, N\}$. The letters mean “left” (L), “right” (R) and “neutral” (N) (no step at all).

A mathematically precise definition of a Turning machine is the following

Definition 1 (Turing machine) A Turing machine (TM) is a 7-tuple

$$M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E), \text{ where}$$

Z is the finite set of machine states,

Σ is the input alphabet,

$\Gamma \supset \Sigma$ is the working alphabet,

$\delta : Z \times \Gamma \rightarrow Z \times \Gamma \times \{L, R, N\}$ is the transition function in the deterministic case,

$\delta : Z \times \Gamma \rightarrow P(Z \times \Gamma \times \{L, R, N\})$ is the transition function in the non-deterministic case,

$z_0 \in Z$ is the initial state,

$\square \in \Gamma - \Sigma$ is the blank,

$E \supseteq Z$ is the set of final states.

Remark 1 The Turing machine introduces a fundamental notion of a simplest memory-oriented computing model. Computation in this model means stepwise altering the contents of memory cells. The memory of a Turing machine consists of single cells which may each contain one letter of a finite alphabet. The memory cells may be addressed by moving the write/read head stepwise from cell to cell.

In the year 1936 Turing proposed the following formal definition of “computability” by means of a Turing machine [171].

Definition 2 (Computability) A function $f : \Sigma^* \rightarrow \Sigma^*$ (e.g. a function on N) is called *computable* or *Turing computable*, if there exists a Turing machine TM which

can calculate this function f from any input x , i.e. if a Turing machine reads the binary representation of a natural number x and stops with $f(x)$ (again, in binary representation) as result on the input/working tape after a finite number of steps.

In other words, a function f is called “computable” if there exists an *algorithm* which is able to compute the value of f for any input as argument within a finite number of steps.

Remark 2 Turing’s definition marks the end of a development of notions of computability in an attempt to derive a general algorithm (or method) which allows for proving or disproving *all* mathematical theorems by using solely the underlying axioms of the system. This attempt of deriving a complete axiomatic basis of all mathematics along with a proof of its consistency (*Hilbert’s Program*) was started by Hilbert in 1900 in a famous lecture [172]. In 1923 Skolem [173] considered as a basis for computability the so-called *primitive recursive* functions which are defined inductively. First, one declares a set of functions axiomatically as primitive recursive, i.e. they are computable. For the rest of the definition one provides rules as to how to obtain new – per definition – computable functions from already known computable functions. One rule is *insertion*, i.e. if f and g are computable then also $f(g(x))$ is computable. Another rule is *iteration*, also called called primitive recursion, i.e. if $f(x)$ is a computable function then also $h(n, x) = \underbrace{f(f(\dots f(x) \dots))}_{n\text{-times}}$ is computable.

In 1931 Kurt Gödel (1906–1978) introduced the primitive recursive function in his work [174] and proved that any sufficiently complex algebraic system is either incomplete or contradictory, i.e. he could prove that Hilbert’s idea of a complete axiomatic basis of all mathematics was doomed. Several other proposals for the definition of computability were published, e.g. S.C. Kleene [175] and A. Church [176] in 1936 proposed a definition of computability based on the so-called λ -definable functions. They also showed in the same publications that this definition of computability is equivalent with the one based on primitive recursive functions [175]. In 1937 it was shown that the λ -definition of computability is also equivalent with Turing’s definition [171, 177]. Gödel and Turing in essence showed that not all questions that can be asked within an axiomatic system in mathematics, in computer science or in general in some logical system are decidable within the bounds of the system. There are problems in mathematics, e.g. the diophantine equations³⁵ [178], which can neither be proved nor disproved with the underlying, available axioms.

³⁵Diophantine equations are polynomial equations with integer coefficients for which an integer solution is sought. This problem is also known as Hilbert’s 10th problem which was raised by him in the year 1900 [172]. It was not before 1970, when Hilbert’s 10th problem could be proved to be unsolvable by Yuri Matiyasevič.

It turned out, that all proposed definitions of “computability” are equivalent to each other. Particularly, there has never been found a computational model which could not – in principle – be represented by a Turing machine. This general observation is also true for the so-called Quantum computers or DNA-computers. Based on this observation A. Church in 1936 made the following proposition [177]:

Proposition 1 (*Church’s thesis*) *The notion of computability is adequately defined by the model of a Turing machine.*

Remark 3 Note that this is a proposition, that has been generally accepted, i.e. it cannot be proved. The term “Church’s” or “Church-Turing” thesis seems to have been first introduced by Kleene who provides a good survey in Chaps. 12 and 13 of [179].

In summary, every effectively calculable function that has been investigated has turned out to be computable by a Turing machine. All known methods or operations for obtaining new effectively calculable functions from given effectively calculable functions are paralleled by methods for constructing new Turing machines from given Turing machines. All attempts to provide an exact analysis of the intuitive notion of an effectively calculable function, i.e. of the notion of computability, have turned out to be equivalent in the sense that each analysis offered has been proved to pick out the same class of functions, namely those that are computable by a Turing machine. Because of the diversity of the various analyses with this respect, see e.g., Church’s thesis is generally accepted.

Definition 2.3 (*Decidability*) A set $A \subseteq \Sigma^*$ of a Turing machine (i.e., an accepted language $T(M) \subseteq \Sigma^*$), is called *decidable* if the characteristic function $\chi_T : \Sigma^* \rightarrow \{0, 1\}$ of T can be computed. For all $w \in \Sigma^*$

$$\chi_T(w) = \begin{cases} 1 & : w \in T , \\ 0 & : w \notin T . \end{cases} \quad (2.90)$$

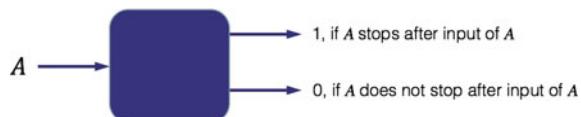
Remark 4 For a definition of $T(M)$ see (2.87) on p. 91.

When it comes to the question of decidability of formal languages, these languages are also called “Entscheidungsprobleme”. With such an algorithm, which is depicted in Fig. 2.26 as a black box, the input is some word over an alphabet. On the other hand, when computing a function, the input is a subset of the natural numbers. Numbers, however, can be represented as words of the alphabet $\{0, 1\}$ in binary form and vice versa, words can be represented as numbers. For this, one simply numbers all symbols of the alphabet A , starting with zero; A word is then interpreted as a number over the number system with basis $|A|$. For a Turing machine, a “natural” form of input/output is given by a binary representation of words over the alphabet $\{0, 1\}$.



Fig. 2.26 Decidability of a formal language. The corresponding algorithm is depicted as *black box*. Such problems are also called “Entscheidungsprobleme”

Fig. 2.27 A *block box* algorithm for the proof of the undecidability of the Halting problem



Example 12 (The Halting Problem) Algorithms are written down in the form of programs, e.g. for a Turing machine. Programs can be written down as words of a certain alphabet. In this representation, programs can be used as input for another algorithm³⁶. In the following, a particular language will be considered which has the following alphabet:

$$H = \{A \mid A \text{ is a program, which, when used as input of } A \text{ stops after a finite number of steps.} \quad (2.91)$$

This particular problem is called *Halting problem* and it is undecidable.

Proof Assuming that the Halting problem was decidable, there is an algorithm which can be depicted as black box, cf. Fig. 2.27.

Using this algorithm, one constructs a new algorithm, which stops exactly, if the black box outputs 0. (The actual output of the algorithm is irrelevant). In the case of 1 as output, the algorithm never stops, i.e. it enters an infinite loop.

Let z the new code of this algorithm. The question is whether z stops or not, i.e. whether $z \in H$, or $z \notin H$. If z stops after input of z , then by construction of the algorithm, the black box outputs 0 upon input of z . However, the black box is the assumed decidability algorithm for the Halting problem. If the black box outputs 0 after input of z , then this means that the algorithm does not stop upon input of z . Likewise, assuming that z does not stop after input of z , then it follows analogously that z stops after input of z . Thus, there is a logical contradiction and the Halting problem is not decidable. ■

As a result of the Halting problem, one realizes, that there are functions, which are not computable. The characteristic function of the Halting problem is one example (Fig. 2.28).

³⁶A simple example for an algorithm that has as input a different algorithm is a compiler.

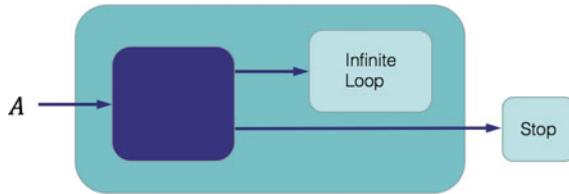


Fig. 2.28 Result of the Halting problem. Either the algorithm enters an infinite loop or it stops, depending on the output

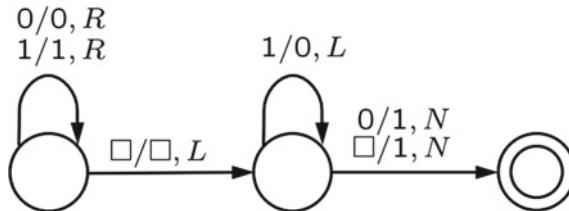


Fig. 2.29 Graphical representation of a Turing machine that calculates the function $f(x) = x + 1$. This machine – without changing the tape content – first moves the read/write head all the way to the *right* to the position of the lowest bit. If this bit is 0 then it is set to 1 and the machine is done. If the bit is a 1 then the bit is set to 0 and the head moves one step to the *left*. This procedure is repeated until the leftmost bit is read and changed accordingly

Example 13 (A Turing machine for the function $f(x) = x + 1$) The Turing machine depicted in Fig. 2.29 calculates the function $f(x) = x + 1$. If one starts this machine with a number x (in binary representation), then it stops after a finite number of steps in a defined end state and the number in the current state of the working tape is $x + 1$ (again, in binary representation). Thus, the function $f(x) = x + 1$ is a “Turing computable” function.

Example 14 (A Turing machine that substitutes all characters) The transition function δ in Fig. 2.29 defines the transition of a Turing machine from one state to the next. This can also be depicted in a *transition table*. Let

$$\begin{aligned} Z &= \{z_0, z_e\} , \\ \Sigma &= \{a, b\} , \\ \Gamma &= \{a, b, \square\} . \end{aligned}$$

The transition table is a 5-tuple $(z_i, \gamma_i, \gamma_j, M)$ given in Table 2.7.

A nice Java-program which provides a GUI and which simulates the (binary) input and output of a Turing machine can be obtained in the world wide web.³⁷ The

³⁷ <http://ais.informatik.uni-freiburg.de/turing-applet/>.

Table 2.7 A transition table for a Turing Machine that substitutes each a with b and vice versa. The machine stops when a blank is reached

z_i	γ_i	γ_j	M	z_j
z_0	a	b	R	z_0
z_0	b	a	R	z_0
z_0	\square	\square	R	z_e

machine accepts as input words in binary format and a transition table such as in Table 2.7 and checks whether the word is accepted.

We have seen previously that when the “computational power” of a computational model such as the Turing machine is restricted too much, then eventually one reaches a point at which the computational model cannot calculate anymore all possible functions. This is the case, e.g. when a Turing machine is “degraded” to a DFA.

Something similar happens when one restricts the programming language of the underlying model too much, e.g. when only allowing FOR-computable functions but no WHILE-loops. The main difference between a WHILE- and a FOR- loop is that with the latter the number of loops is set at the beginning, whereas a WHILE-loop decides dynamically after each loop whether a stop-criterion has been reached. Thus, a program which *only* uses FOR-loops definitely stops after a finite number of steps. The *Ackermann function* $A(x, y) : \mathbb{N}^2 \rightarrow \mathbb{N}$ which is defined for all $(x, y) \in \mathbb{N}$ by the following recursive scheme (see [180, 181]):

$$A(0, y) = y + 1 , \quad (2.92a)$$

$$A(x + 1, 0) = A(x, 1) , \quad (2.92b)$$

$$A(x + 1, y + 1) = A(x, A(x + 1, y)) , \quad (2.92c)$$

is an example for a function which is only WHILE-computable, but not FOR-computable, see e.g. [82]. Furthermore, it can be shown that FOR-computability and *primitive recursive* computability are equivalent [182]. Thus, $A(x, y)$ is an example of a computable function which is *not* primitive recursive.

2.6.6 Efficiency of Algorithms

While there are some problems which fall into the category of “computable” it turns out that the associated algorithms are useless for all practical purposes because they require astronomical computation times. The *Ackermann function* (2.92)a–c is just one of many examples which are computable *in principle* but which need astronomical computing times, even at small input values. This function grows larger than is possible by substitution or recursion and only for small values of the arguments ($x < 4$ and $y < 4$) an explicit expression of $A(x, y)$ can be given. For the original publication of this function by Ackermann in 1928, which was slightly different from the modern textbook version given above, see [186]; also compare Problem 6.

The computing time is measured by the number of *elementary steps* (ES) needed by some algorithm until it stops, i.e. until the problem is solved. Examples for elementary steps are:

- Executing one of the elementary operations ($+$, $-$, \times , DIV , MOD),
- Assigning a value, i.e. changing the contents of a memory,
- Initializing a loop variable,
- Testing an *if*-condition.

Example 15 (Number of elementary steps of a very simple sort algorithm) Consider the following piece of pseudo-code (2.7) which gets as input an array $a[1, \dots, n]$ which is to be sorted.

Algorithm 2.7: A sort algorithm

```

1 for i := 1 TO n - 1 DO
2   for j := 1 TO n DO
3     if a[i] > a[j] then
4       h := a[i]; a[i] := a[j]; a[j] := h;
5     END
6   END
7 END
```

The kernel of two loops in Algorithm 2.7 consists of one *if*-condition (1 ES) and – in the positive test case – three assignments (3 ES) which switch $a[i]$ with $a[j]$. Each i - and j -loop counts 1 ES. Thus, one can directly write down the number of ES:

$$\sum_{k=1}^{n-1} \left(1 + \sum_{l=k+1}^n (1 + 4) \right) = (n-1) + \sum_{k=1}^{n-1} \sum_{l=k+1}^n 5 \quad (2.93)$$

$$= (n-1) + 5 \times \frac{n \times (n-1)}{2} \quad (2.94)$$

$$= 2.5 \times n^2 - 1.5 \times (n-1). \quad (2.95)$$

Assuming that *one* ES on an average computer takes 10^{-9} s, one can sort arrays containing 20000 elements within one second. Often, one is only interested in how the run time of an algorithm depends on the number of input elements n , i.e. one only considers the leading term in the computation time. In the example above one would speak of a “quadratic”, or “order n^2 runtime” and writes symbolically $\mathcal{O}(n^2)$. The meaning of this notation is the following:

Table 2.8 Overview of typical runtimes of algorithms occurring in materials science applications. Depicted are the number of elementary steps and the corresponding realtimes for the different algorithms under the assumption that one ES takes 10^{-9} s

Algorithm	Runtime	$n = 10$	$n = 20$	$n = 50$	$n = 100$
A_1	n	10 ES	10 ES	10 ES	10^{-7} s
		10^{-8} s	2×10^{-8} s	5×10^{-8} s	10^{-7} s
A_2	n^2	100 ES	400 ES	2500 ES	10000 ES
		10^{-7} s	4×10^{-7} s	2.5×10^{-6} s	10^{-5} s
A_3	n^3	1000 ES	8000 ES	10^5 ES	10^6 ES
		10^{-6} s	8×10^{-6} s	10^{-4} s	0.001 s
A_4	2^n	1024 ES	10^5 ES	10^{15} ES	10^{30} ES
		10^{-6} s	0.001 s	13 days	$\sim 10^{13}$ years
A_5	$n!$	$\sim 10^6$ ES	$\sim 10^{18}$ ES	$\sim 10^{64}$ ES	10^{158} ES
		0.003 s	77 years	10^{48} years	$\sim 10^{141}$ years

Definition 2.4 (\mathcal{O} -notation) A function $g(n)$ is of order $f(n)$, i.e. $g(n) = \mathcal{O}(f(n))$ if there are constants c and n_0 such that $\forall n \geq n_0: g(n) \leq c \times f(n)$.

The symbol “ \forall ” is short for “for all” in mathematical notation.

Example 16 The function $2n^2 + 5n$ is of order n^2 , or in symbolic notation: $2n^2 + 5n = \mathcal{O}(n^2)$, as one can choose $c = 3$. Then $2n^2 + 5n \leq 3n^2 \quad \forall n > 5$. Thus, the previous relation is true for e.g. $n_0 = 5$.

To classify the efficiency of algorithms we consider in Table 2.8 five different algorithms A_1, A_2, A_3, A_4, A_5 with corresponding runtimes $n, n^2, n^3, 2^n, n!$, where n is the considered system size, e.g. the number of atoms, particles or finite elements in some simulation program. These runtimes are typical for different applications in materials science. We again assume that one elementary step takes 10^{-9} s on a real computer.

It is obvious from Table 2.8 that *exponential* runtimes (algorithms A_4 and A_5) are generally not acceptable for all practical purposes. For these algorithms, even with very small system sizes n one reaches runtimes which are larger than the estimated age of the universe (10^{10} years). Algorithm A_5 could be a solution of the traveling salesman problem (see Sect. 2.6.3). If the first point out of n has been visited, there are $(n - 1)$ choices for the second one. This finally results in an exponential runtime of at the least $n!$ steps. A runtime 2^n as in A_4 is typical for problems where the solution space of the problem consists of a subset of a given set of n objects; There are 2^n possible subsets of this basis set. The “efficient” algorithms A_1, A_2, A_3 with runtimes of at the most n^3 are the most commonly used ones in computational materials science.

Usually, in atomistic simulations one assumes the interactions between particles to be pairwise additive. Hence, the interaction of particles (or atoms) in a system depends only on the current position of *two* particles. Sometimes however, three-body interactions have to be included, e.g. when considering bending and torsion potentials in chain molecules, cf. Sect. 6.3.7. These potentials depend on the position of at least three different particles. Solving the Schrödinger equation in ab initio simulations also leads to a n^3 -dependency of the runtime. This is the main reason why ab initio methods are restricted to very small system sizes (usually not more than 1000 atoms can be considered). Solving the classical Newtonian equations of motion with a “brute-force” strategy leads to a n^2 -efficiency $\left(\frac{n \times (n-1)}{2}\right)$ of the algorithm that calculates the interactions of particles. This is also generally true in finite element codes where special care has to be taken when elements start to penetrate each other. Usually one uses so-called *contact-algorithms* which use a simple spring model between penetrating elements. The spring forces try to separate the penetrating elements again and the core of the contact algorithm is a lookup-table of element knots which is used to decide whether two elements penetrate each other or not. This algorithm in its plain form has an efficiency of n^2 . As an n^2 efficiency of an algorithm still restricts the system size to very small systems of a few thousand particles one uses several methods to speed-up the efficiency of algorithms in computer simulations. Usually, this is done by using sorted search tables which can then be processed linearly (and thus reaching an efficiency of $\sim n \log n$). Hence, when it comes to the efficiency of algorithms in materials science, one will always try to minimize the effort to $\mathcal{O}(n)$ (with a remaining prefactor that might still be very large). A discussion of several of the most important speed-up techniques commonly used in MD simulation codes is provided in Sect. 6.4.1.

Another consideration in Table 2.9 shows why algorithms A_1 , A_2 and A_3 may be considered to be *efficient*. Assuming that the available computer systems – due to a technology jump – will be 10 or 100 times faster than today, then the efficiency of algorithms A_1 , A_2 and A_3 will be shifted by a factor, whereas for the exponential algorithms A_4 , A_5 the efficiency will be shifted only by an additive constant, cf. Table 2.9.

Table 2.9 Speedup of the runtime of different algorithms assuming a hardware speedup factor of 10 and 100. The efficiency of polynomial algorithms will be shifted by a factor while exponential algorithms are only improved by an additive constant

Algorithm	Runtime	Efficiency	Speedup factor 10	Speedup factor 100
A_1	n	n_1	$10 \times n_1$	$100 \times n_1$
A_2	n^2	n_2	$\sqrt{10} \times n_2 = 3.16 \times n_2$	$\sqrt{100} \times n_2 = 10 \times n_2$
A_3	n^3	n_3	$\sqrt[3]{10} \times n_3 = 2.15 \times n_3$	$\sqrt[3]{100} \times n_3 = 4.64 \times n_3$
A_4	2^n	n_4	$\log_2(10 \times n_4) = n_4 + 3.3$	$\log_2(100 \times n_4) = n_4 + 6.6$
A_5	$n!$	n_5	$\approx n_5 + 1$	$\approx n_5 + 2$

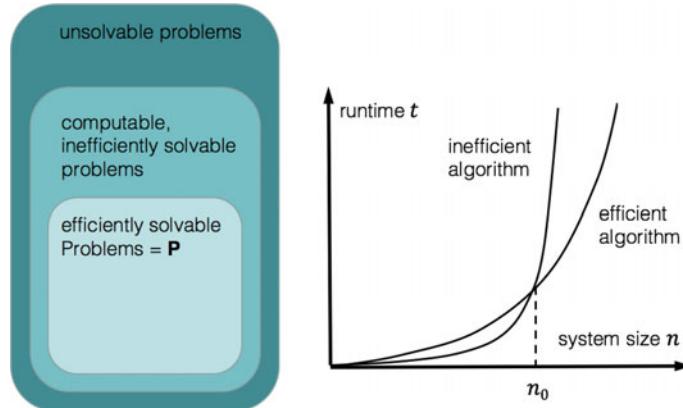


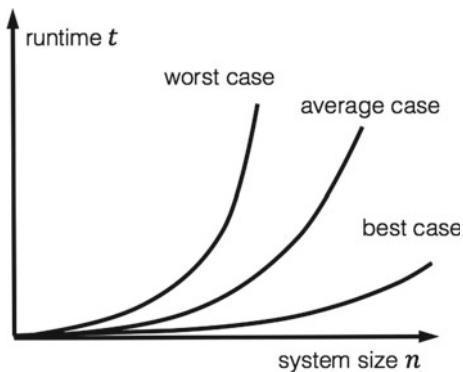
Fig. 2.30 Illustration of the class **P** of efficiently solvable problems, inefficiently solvable and unsolvable problems along with their runtime behavior

Algorithms A_1, A_2 and A_3 have *polynomial* runtimes. An algorithm is said to be *efficient* if its runtime – which depends on some input n – has a polynomial upper bound. For example, the runtime function $2n^4(\log_2 n)^4 + 3\sqrt{n}$ has a polynomial upper bound (for large n), e.g. n^5 . In \mathcal{O} -notation this is expressed as $\mathcal{O}(n^k)$ with k being the degree of the polynomial. Algorithms A_4 and A_5 on the other hand have no polynomial upper limit. Thus, they are called *inefficient*. The class of Problems that can be solved with efficient algorithms – i.e. algorithms that are polynomially bounded – are denoted with the letter **P**, cf. Fig. 2.30. The set of polynomials is closed under addition, multiplication and composition. Thus, **P** is a very robust class of problems. Combining several polynomial algorithms results into an algorithm which again exhibits a polynomial runtime.

Remark 5 Due to the robustness of the definition of the class **P** of efficient algorithms, an inefficient algorithm can have a shorter runtime than its efficient counterpart, up to a certain system size n_0 . For example, an algorithm with a runtime $1000 \times n^{1000}$ falls into the class **P** whereas an algorithm with a runtime 1.1^n is exponential and thus inefficient. However, the exponential algorithm only exhibits longer runtimes than the efficient one for system sizes $n \sim 123000$, cf. Fig. 2.30.

In Example 15 on p. 100 the “worst-case” runtime for a simple sort-algorithm was considered assuming that the *if*-condition within the loop of Algorithm 2.7 is true and thus, three elementary steps are *always* executed. In the “best-case” – e.g. if the array has been sorted – this *if*-condition is not true and there are only $(n - 1) + n(n - 1) = n^2 - 1$ elementary steps. For a randomly shuffled array one can show that the expectation value for the number of elementary steps is $\sum_{k=2}^n (1/k) \approx \ln n$ [184]. Thus, with a randomly sorted array the total number of ES in this example

Fig. 2.31 Illustration of the worst-case, average-case and best-case behavior of algorithms. The average-case behavior of an algorithm is usually the most difficult to determine



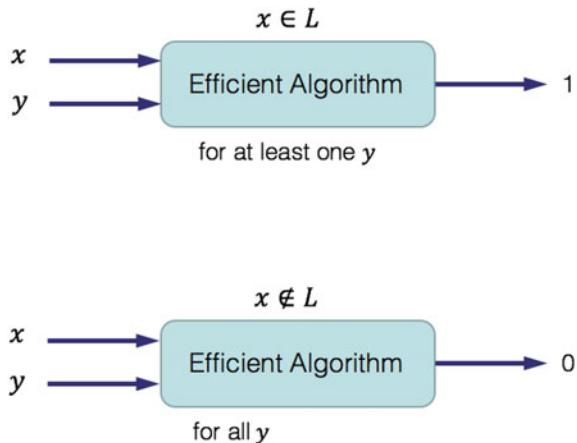
is roughly $n^2 + 3n \ln n$. Hence, the *actual* runtime of an algorithm lies somewhere between the worst-case and the average-case runtime behavior, cf. Fig. 2.31.

Remark 6 (Cryptology) The fact, that for certain problems no efficient algorithm is known, is the basis of almost all practical cryptographical concepts, e.g. password files $f(x)$ on a computer system are generated from the input x as a new password. However, nobody who can read the password file will be able to calculate the inverse function $f^{-1}(x)$, i.e. to find an x' for which $f(x') = f(x)$. Although the password is not known to the system, it can be checked simply by calculating the function $f(y)$ again with the provided password y at login and comparing it with $f(x)$. For the function f , one preferably used so-called *one-way functions*, i.e. functions for which f can be calculated efficiently, but for which no efficient algorithm is known to calculate f^{-1} . For example, a common choice is $f(x) = a^x \text{MOD} n$, where $x = 1, 2, \dots, n - 1$, $a \in \mathbb{N}$ and n is a prime number. The number of necessary steps to calculate this function is given by $\mathcal{O}(\log x)$, i.e. *linear* in the binary representation of x and there is no efficient algorithm to calculate $f(x)^{-1}$ [185].

A special class of problems for which no efficient algorithms are known are the so-called **NP-complete** problems. The letters **NP** are short for *non-deterministic polynomial runtime*. **NP** problems have the same computability model as **P** problems, but with a non-deterministic algorithm. The NP-complete problems are the most difficult problems in NP in the sense that they are the smallest subclass of NP that could conceivably remain outside of **P**, the class of deterministic polynomial-runtime problems. The reason is that a deterministic, polynomial-runtime solution to any NP-complete problem would also be a solution to every other problem in NP.

We provide in the following a formal definition of a **NP**-problem, following Steven Cook's original article in which the first **NP**-problem was published [186], although in this article the term **NP** was not yet used. A Language L , i.e. a "Entscheidungsproblem" lies within the class **NP**, if there is an efficient algorithm, which works with two input variables, such that the following two conditions are fulfilled:

Fig. 2.32 Illustration of a language that is an element of class NP

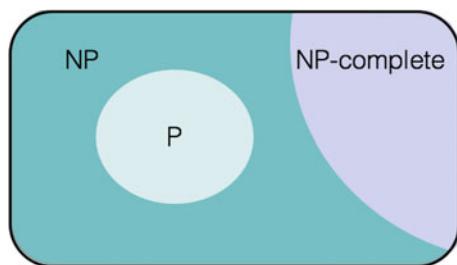


1. If $x \in L$, then there is a second word y as input (where the length $|y|$ of y has to be polynomial bounded within the length $|x|$) such that the above mentioned efficient algorithm provides 1 as output.
2. If $x \notin L$, then, after input of x , the output of the efficient algorithm is always 0, independent of the arbitrary second input y .

This definition is depicted in Fig. 2.32.

The importance of this definition lies in the fact that no more efficient algorithm for the decision whether $x \in L$ is known except of trying out the complete set of potential input elements $y \in \Sigma$ until the efficient algorithm outputs a 1. When such a y is found then $x \in L$. If the complete search of all elements y was done and the output was always 0 then $x \notin L$. The length of y is bounded by a polynomial p of the length $|x|$; Hence, the search effort is of order $|\Sigma|^{p(|x|)}$ which is exponential, i.e. non-polynomial. The great importance of NP-class problems lies in the fact that more than 3000 NP-problems are known from such different research areas as number theory, computational geometry, graph theory, sets and partitions, program optimization, automata and language theory and many more, see e.g. [187]. This also has consequences for code optimization in materials science applications. For example, when writing a massive parallel computer program then there is no known optimization strategy for a domain decomposition (a splitting of the considered domain into several parts which are then assigned to different processors) of the considered system which leads a guaranteed optimal solution. In graph theory there exists no efficient algorithm that minimizes the distance between vertices of a graph which is of importance, e.g. when generating a high-quality mesh for finite element applications.

Fig. 2.33 The NP-complete problems are the smallest subclass of NP for which no efficient algorithms are known



Today, it is generally assumed that all problems in **P** are contained in the class **NP**, cf. Fig. 2.33.

So far, no proof that decides whether $\mathbf{P} = \mathbf{NP}$ or $\mathbf{P} \neq \mathbf{NP}$ is known, i.e. it is unknown whether NP-complete problems are in fact solvable in polynomial time.³⁸

With these remarks we end our discussion of the attempts to formalize the notions of “computability” and “algorithm”.

2.6.7 Suggested Reading

A good starting point to appreciate the developments in the modeling of real systems by means of algorithms would be Chabert [188] and Lee [189]. There is a multitude of excellent textbooks on graph theory, formal languages and automata. Cormen et al. [190] provides a good introduction to algorithms in graph theory. Diestel [191] and Gibbons provide a sophisticated treatment of graph theory. Prömel and Steger [192] treat the Steiner tree problem in depth and provide many useful algorithms for practical problems. Gruska [193] and Hopcroft [170] provide a solid introduction into the foundations of automata theory. A good treatment of Markov chains can be found in Motwani [185]. The original article by Markov is [194] and for a discussion in English, see e.g. [195]. The Monte Carlo Method was introduced into physics in 1953 by Nicolas Metropolis et al. [24]. A nice historical account of the Turing machine is provided in Copeland [196]. A standard introduction into complexity and computability is provided by Ausellio et al. [197] or Cooper [198]. Classic textbooks on computability and **NP**-problems are e.g. Garey and Johnson [199] or Homer and Selman [200].

³⁸This is one of the great unsolved problems in mathematics. The Clay Mathematics Institute in Cambridge, MA, U.S.A. is offering a 1 million \$ reward to anyone who has a formal proof that $\mathbf{P} = \mathbf{NP}$ or that $\mathbf{P} \neq \mathbf{NP}$.

Problems

Problem 1 Proper Time Interval $d\tau$

Show that the Lorentz-transformations Λ_{β}^{α} leave the proper time $d\tau$ (see p. 52) invariant.

Problem 2 Conservation Laws

State for each of the following particle reactions whether it is forbidden or not. If applicable, state the conservation law that is violated.

- (a) $\bar{p} + p \rightarrow \mu^+ + e^-$,
- (b) $n \rightarrow p + e^- + \nu_e$,
- (c) $p \rightarrow n + e^+ + \nu_e$.

Problem 3 Euler-Lagrange Equations

Perform the variation of (2.40) and show that (2.41) are the corresponding equations of motion.

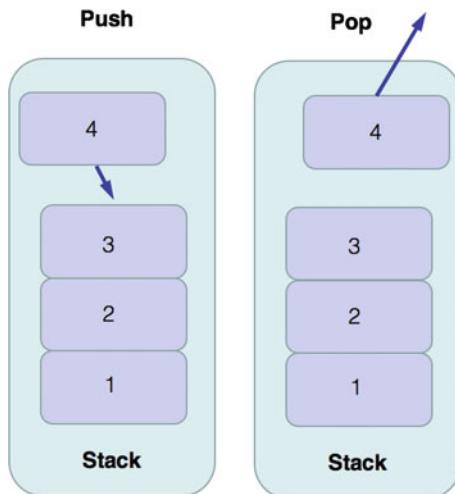
Problem 4 Klein–Gordon and Dirac Equation

Show that the field ψ of (2.62) on p. 73 satisfies the correct energy-momentum relation, i.e. it satisfies the Klein–Gordon equation (2.57). Derive from this a set of equations for the α_i and β .

Problem 5 Abstract Data Types: LIFO Structure

The two basic operations of stacks (LIFO structures) are *push* (putting one data element on the stack) and *pop*, cf. Fig. 2.34.

Fig. 2.34 Push (a) and pop (b) operation with stacks



Write an implementation of the stack with a push and pop functionality in C++ using a modular design, i.e. use a header file “Stack.h” for declarations, a file “Stack.cpp” and a main procedure which tests this implementation.

Problem 6 An implementation of the Ackermann function

Write a recursive implementation of the Ackermann function (2.92a), (2.92b), (2.92c). How long does it take to compute $A(5, 0)$? (You can go and drink a coffee in the mean time). What about $A(5, 1)$?

Abstract

We begin in this chapter with an introductory Sect. 3.1 in which we consider different notions of the concept of space that play a fundamental role in physical and mathematical theories. All theory is usually formulated by using objects with certain algebraic properties that are embedded in a particular concept of “space”. Then, in Sect. 3.2, starting from a fundamental discussion of the properties of sets we define the concept of a function, a commutative ring and a field. We define a linear vector space and discuss the important concept of basis vectors and linear forms (functionals) as mappings between vector spaces. In Sect. 3.3 we study topological spaces which are in essence open sets endowed with some additional properties such as “neighborhood” and “connectivity”. From this we define a metric and the concept of charts, coordinate systems and manifolds. The concept of manifolds as a general notion of “space”, endowed with certain properties necessary to do physics then leads us to tangent and cotangent spaces, one-forms and tensors which are important geometric quantities used abundantly in modern physical theories. From a discussion of metric spaces and a metric connection in Sect. 3.4 we come to a discussion of Riemannian manifolds in Sect. 3.5, which are the fundamental mathematical structures in which the general theory of relativity is formulated. The classical problems of inertia, motion and coordinate systems in Sect. 3.6 lead us to taking a short glimpse on the formulation of relativistic field equations for the description of condensed matter (that is, of fluids and solids) in Sect. 3.7. This chapter ends with Sect. 3.8 in which further literature is listed, followed by many exercises.

In almost all problems treated in theoretical physics there is some sort of continuous space involved. This might be the phase space of generalized coordinates in classical mechanics, Hilbert space for problems in quantum theory, Sobolev space (a special class of Hilbert space) when solving partial differential equations in fluid or solid

mechanics, Γ -space in thermodynamics, non-Euclidean curved space–time in the theory of general relativity, Minkowski space in the special theory of relativity, and so on.

All of the above mentioned spaces have them geometrical properties, but they also share something in common, which has to do with them being continuous spaces rather than discrete lattice points. As different notions of space also play a role in the formulation of differential equations, it will be very helpful for a deeper understanding of numerical techniques in Part II of this book, to have reviewed some of the mathematical formalizations, notions and properties of spaces which are commonly used in physical theories.

3.1 Introduction

The purpose of this chapter is to introduce our notation of functions and equations and to review some of the formal concepts of spaces in mathematics and physics, including a topological space, tangent space, spaces with an affine and metric connection, spaces with a differential structure, linear vector spaces in which tensors are defined as geometric, coordinate independent objects, and (Riemannian) manifolds. The style of presentation in this chapter will be of formal nature, that is to say we will introduce several definitions and deductions first in a way which is common in mathematical literature. This is not done to defer the less prepared reader who suffers from the typical succinct “definition-theorem-proof”-style of modern mathematical literature but it is simply a matter of economy. Many examples and illustrations are given of the discussed topics to allow for a “pictorial way of thinking” which is probably the way most physicists think when developing and working with theories.

In this respect, this chapter provides orientation and a review of mathematical structures of space which are important in physics as well. In fact, there is a complete hierarchy of spaces, or different levels of structure in the theoretical development, cf. Fig. 3.1.

We start out with some mathematical facts on set theory, mostly for the sake of fixing our notation. We provide some definitions and deduce some facts that follow and which will be needed in the formulation of the physical theories discussed later on. This part is also provided to give a sound footing and a handy reference to the inadequately prepared reader. Occasionally, some examples are provided as an offer to those readers who want to get some working knowledge on the issues that are discussed.

We will shortly consider some basic facts on topology which is based on the properties of open sets alone and allows for a very general idea of “space” – the *topological manifold*. Topological mathematical structures include the concepts of limit, neighborhood and continuity, which give rise to the idea of abstract spaces. The fundamental relations of topology, such as the one of “inclusion”, or “part of”, are qualitative; algebraic concepts of algebraic topology and algebraic geometry make a topological structure quantifiable by assigning it to an algebraic structure, e.g. to the real number system. We will then see how we approach the concept of space which is

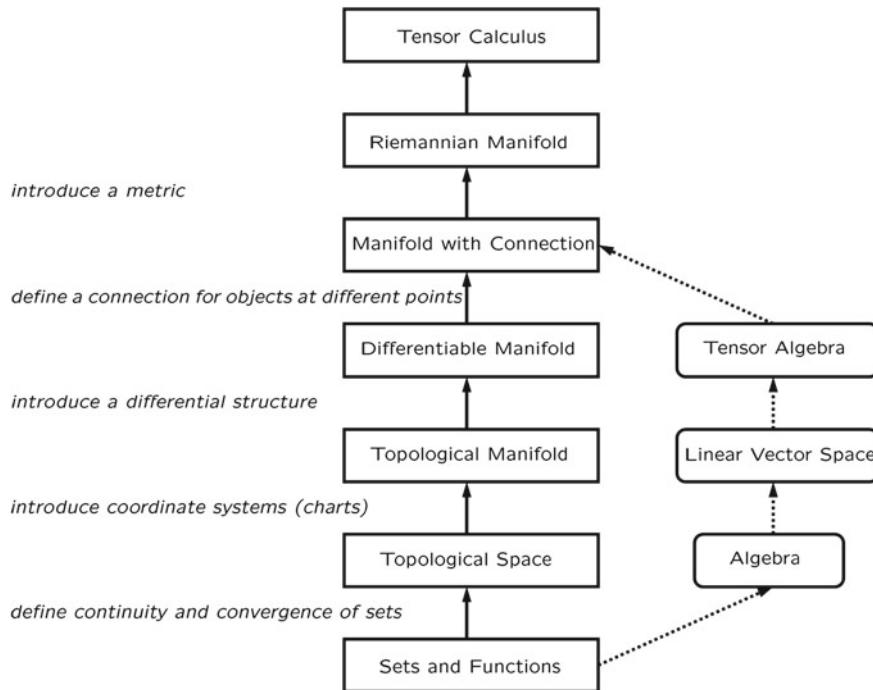


Fig. 3.1 Hierarchical structure of the formalizations of “space” starting at the lowest level with sets using a minimum of algebraic structure

Table 3.1 The main concepts of abstract space definitions in mathematics and their respective levels of application

Structural level	Main concept	Structure
Sets	Points	Mere point sets with no structure
Topology	Locality, neighborhood	Topological space
Differentiability	Vectors at curves	Differentiable manifold with atlas
Connection	Parallelism (of vectors)	Manifolds with affine connections
Metric	Distance (between points)	Manifold with metric connections

used in most areas of physics by adding some structure (a connection, a differential structure and finally a metric) to the topological manifold. This finally leads to the concept of space in terms of *Riemannian manifolds*, named after the mathematician Bernhard Riemann (1826–1866), who introduced differential geometric concepts. The levels of structure of a Riemannian manifold include point sets, topology, differentiability and affine and metric connections. In most textbooks, these different levels are presented in a mixed-up fashion. Table 3.1 exhibits these different structural layers. We will further see that the concept of a *linear space* is fundamental for the definition of tensors. The importance of a *linear vector space* in physics is based on the superposition principle which is essential for solving differential equations.

The theory of tensors is introduced in a linear vector space by considering them as geometric objects which are independent of the frame of reference (or mathematically speaking, of the coordinate system) which is chosen for a description of events in spacetime. The very name indicates its origin in the theory of elasticity. Several recent books on continuum mechanics – in particular the ones on an engineering level – only use tensors to the extend that they introduce tensor notation in Cartesian coordinates as a convenient shorthand for writing equations. This is a rather harmless use of tensors. Tensor theory in such textbooks is often either completely ignored or some “index tensor notation” is introduced without a solid theoretical basis except for maybe some remarks in the appendix. Actually, tensor theory is closely connected to differential geometry and to concepts in topology; it is an important tool to write down generally covariant equations, i.e. equations that are valid in any coordinate system without changing their form. It is hoped that the approach to tensor theory which is attempted in this chapter will not only be of use to natural scientists but will also provide a solid understanding of this theory to the reader with an engineering background.

Some subtle topological aspects and many formal proofs of theorems in the discussion of tensors and manifolds – which many mathematicians may consider essential – will be ignored in the discussion in this book, as many of the topological assumptions which are needed to understand what tensors are and how they are used in manifolds, are either extremely technical (separability, paracompactness), or very intuitive (continuity, the Hausdorff property). The former are needed mostly for proofs of existence problems – e.g. the existence of Riemannian metrics is proved using paracompactness – with which we will not concern ourselves any further. It is important to understand that not all concepts which are used in the context of Riemannian manifolds are meaningful (or even defined) on all structural levels, e.g. on the topological

Table 3.2 Applicability of different concepts used in the context of Riemannian manifolds

Concept	Topology	Differentiable structure	Affine connection	Riemannian metric
Continuity of functions	×	×	×	×
Tangent vectors		×	×	×
Tensors		×	×	×
Lie derivatives		×	×	×
Differential forms		×	×	×
Geodesics			×	×
Curvature			×	×
Covariant derivatives			×	×
Parallel transport			×	×
Distance between points				×
Angles between vectors				×

level there is no notion of differentiability. Table 3.2 summarizes some mathematical concepts and their level of application.

After having introduced tensors as the appropriate tool to describe physical properties of systems which change from point to point we will consider tensor calculus and shortly discuss the special and general theory of relativity. Here, we will consider the four-dimensional spacetime (Minkowski space) of the special theory of relativity, and its curved equivalents (Pseudo-Riemannian spaces) in the general theory of relativity. While this is not a book on relativity theory, we still want to discuss some elements of both theories, as they are the prime applications for the theory of tensors. The last section provides a very short discussion of relativistic hydrodynamics.

Tensors are also abundant in typical engineering applications, such as fluid dynamics or structural mechanics where the stress tensor is an example for a very important tensorial quantity. Our general focus is not mathematical rigor – albeit sporadically we do include some mathematical proofs for the sake of completeness – but a basic understanding of the mathematical framework of the discussed tools in order to be able to appreciate their proper use in physics and in numerical applications. Many references to more specialized and to *primary* literature (and its English translations where possible) are provided throughout this chapter.

3.2 Sets and Set Operations

In order to procure a fine grasp of a mathematical notion of space one first has to take a look at some basics of set theory. We discuss the theory very informally. Our primary aim here is to get acquainted with the rudiments of set theory and to establish our notation.

We do not define what a set is but accept it as a primitive notion. Therefore we imagine a *set* to be a collection of objects, be these arbitrary symbols, numbers, or any other entities. The objects that have been summarized in a set are called the *elements* or *points* of this set. Note that there is no association at all between the points of a set. The points are independent of each other. For example, there is no relation between points which are near to each other, because there is no definition of “nearness” nor its opposite, “distance”. Any subset of a number of points of a set is equivalent to any other subset of the same number of points. One can merely count the points of a set and list them in various orders. Finally, the idea of an element s being a member of a given set S is denoted by $s \in S$, read “is an element of”. When s does not belong to S we write $s \notin S$.

Remark 7 Only differing objects are summarized in a set S , i.e. no element of S may be contained more than once in S . An entity of objects which are not necessarily different from each other is not called a set but a *system*.¹

¹Note the different use of the term “system” in physics and mathematics, of Sect. 3.1.

A set can be identified in one of two ways: by listing all distinguishing features which members have and non-members do not have or (if this is inconvenient or impossible) by simply listing its members. According to the latter scheme, the set consisting of the elements a and b is completely defined by placing all its elements within braces $\{a, b\}$. A set is determined by the membership of its elements alone. Thus, two sets A and B are *equal*, written $A = B$, iff they contain the same members, where “iff” stands for the phrase “if and only if”. Otherwise we write $A \neq B$. To specify sets according to the other scheme we introduce the following convenient notation: If p is a property and x an element, then the expression $p(x)$ states “the element x has property p ”; as a symbolic statement we can write $\{x \mid p(x)\}$. This reads “the set of all elements for which $p(x)$ is a true statement”. For example, if p is the property of being an even number between 1 and 11, we can write this as $\{x \mid x \text{ is an even number between 1 and 11}\} = \{2, 4, 6, 8, 10\}$.

When every element of A is also an element of B , i.e. $x \in A$ implies $x \in B$, or symbolic $x \in A \Rightarrow x \in B$, then A is a subset of B . This relation of sets is written $A \subset B$ or $B \supset A$, which is read “ A is contained in B ” and “ B contains A ”, respectively. The relation $A \not\subset B$ signifies that A is not a subset of B . As there are no restrictions imposed upon the sets A and B above, we have $A \subset A$ which is clearly in accord with the concept of a subset. This leads to a first theorem.

Theorem 1 (The following statements are equivalent)

- (1) $A = B$,
- (2) $A \subset B$ and $B \subset A$.

The two statements in the previous theorem are either both true or both false. This fact is used in virtually all proofs in which the equality of sets is in question. When the relations $A \subset B$ and $B \not\subset A$ are true, then A is a *proper subset* of B .

Remark 8 Although the word “contain” is used for both symbols “ \in ” and “ \subset ”, the meaning is different in each case, and which one is meant can be determined from the context. Frequently, an element x and the single-element set $\{x\}$ are *not* distinguished, which destroys the distinction between $x \subset x$ which is always true and $x \in x$ which is generally false as the set x does not necessarily contain the element x .

Finally, we mention the important concept of a set containing no elements. Consider the formula $E = \{x \mid x \neq x\}$. The set E specifies a well-defined set, but as there are no elements x such that $x \neq x$, the set E is said to be *empty*. The empty set is subset of every set. While the empty set was obtained through a specific formula, its uniqueness is readily established.

Theorem 2 (There is only one empty set \emptyset)

Proof If X is not a subset of A , then there must be an element $x \in X$ such that $x \notin A$. When A is an arbitrary set and X is an empty set, then $X \subset A$. Similarly, if B is a given set and Y another empty set, then $Y \subset B$. Since there is no restriction implied in the choice of sets A and B , we may put $A = Y$ and $B = X$. With this choice we find that the relations $X \subset Y$ and $Y \subset X$ are both true and the proof follows from Theorem 1. ■

Remark 9 Set theory was introduced by Georg Cantor (1845–1918) in the late 19th century to cope with infinities in mathematics. This theory eventually became the foundation of most mathematical disciplines. However, simple notions of sets may lead to surprising contradictions, so-called antinomies in set theory which led to a deep crisis in mathematics at the end of the 19th century. As an example we mention Russel's antinomy [201] (after Bertrand Russel (1872–1970)): Our notion of sets allows a set to be contained in itself as an element. A set which does not contain itself as an element be called “normal set”. Thus, normality of a set means $S \notin S$. Now consider all sets \mathfrak{S} of normal sets and ask, whether this set is normal. If \mathfrak{S} was normal ($\mathfrak{S} \notin \mathfrak{S}$) then \mathfrak{S} would be contained in the set of all normal sets \mathfrak{S} , i.e. $\mathfrak{S} \in \mathfrak{S}$, thus we have $\mathfrak{S} \notin \mathfrak{S} \Rightarrow \mathfrak{S} \in \mathfrak{S}$. With the same reasoning we obtain $\mathfrak{S} \in \mathfrak{S} \Rightarrow \mathfrak{S} \notin \mathfrak{S}$ because if \mathfrak{S} was not normal ($\mathfrak{S} \in \mathfrak{S}$), it would not belong to the set \mathfrak{S} of all normal sets which implies $\mathfrak{S} \notin \mathfrak{S}$. In summary we have the contradictory result $\mathfrak{S} \in \mathfrak{S} \Leftrightarrow \mathfrak{S} \notin \mathfrak{S}$.

In order to avoid this set-theoretical antinomy, we may not build sets unscrupulous, but we agree once and for all that all sets A in a given discussion belong to the *power set* $P(X) = \{A \mid A \subset X\}$ of some some a priori known set X . The power set $P(X)$ of a set X consists of all subsets of X .

Example 17 Let $X = \emptyset$. Then $P(\emptyset) = \{\emptyset\}$. If $X = \{1, 2\}$ then the power set has four members: $P(X) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

For two sets A and B the *union* ($A \cup B$) is the collection of all elements belonging to A or B or both.

Note that the word “or” is used here in the inclusive sense, i.e. the statement “ x or y ” implies “ x or y or both x and y ”. The *intersection* of A and B , ($A \cap B$), read “ A intersect B ” is the set consisting of those elements which belong to both A and B . The sets are *disjoint* when their intersection is empty. Finally, we write the *difference set* or *relative complement* of those members of A which are excluded from B as ($A \setminus B$). A formal summary of the preceding follows:

Definition 3.1 (*Union, Intersection, Difference*) $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$,
 $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$,
 $A \setminus B = A - B = \{x \mid x \in A \text{ and } x \notin B\}$.

Sometimes it is convenient to use the union and intersection of more than two sets. To include this case, one starts with a collection of sets (a family of sets) A_i which is labeled with a subscript which varies over some index set \mathfrak{I} . This means that to each element $i \in \mathfrak{I}$ there corresponds a set A_i , or in formal notation:

Definition 3.2 (General Union and Intersection) Let \mathfrak{I} be a given set. With each $i \in \mathfrak{I}$ let there be associated a set A_i . Then $\bigcup_{i \in \mathfrak{I}} = \{x \mid x \in A_i \text{ for at least one } i \in \mathfrak{I}\}$,

$$\bigcap_{i \in \mathfrak{I}} = \{x \mid x \in A_i \text{ for all } i \in \mathfrak{I}\}.$$

A very important special case of families of a finite or infinite number of subsets A_i ($i \in \mathfrak{I}$) occurs when $A = \bigcup_{i \in \mathfrak{I}} A_i$ and $A_i \cap A_j = \emptyset$ for $i \neq j$. If x and y are in A , we write $x \sim y$ when both elements belong to the *same* subset A_i , $x \not\sim y$ otherwise. The set \mathfrak{P} of all subsets of A is called a *partition* of A , if each element of A belongs to one and only one set of \mathfrak{P} . The notion of a partition is visualized in Fig. 3.2 and forms the basis of the definition of an *equivalence relation*, which is an important tool in mathematics for comparing arbitrary sets with each other.

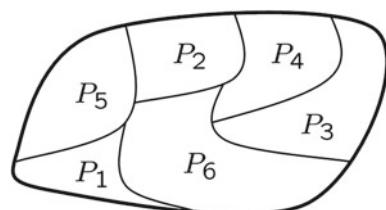
Definition 3.3 (Equivalence Relation) An *equivalence relation* on a set A with arbitrary elements x, y, z, \dots is a relation \sim for which the following holds:

- (1) $x \sim x$ (reflexivity);
- (2) if $x \sim y$, then $y \sim x$ (symmetry);
- (3) if $x \sim y$ and $y \sim z$, then $x \sim z$ (transitivity);

Corollary 1 *The partition of a set gives rise to an equivalence relation.*

Proof Let A be a given set, and \sim an equivalence relation. We associate with each element $a \in A$ the subset $E(a) = \{x \in A \mid x \sim a\}$. The set E is called an *equivalence class*. Clearly $A = \bigcup_{a \in A} E(a)$. We assert that if $E(a)$ and $E(b)$ are arbitrary equivalence classes of A , then either $E(a) = E(b)$, or else $E(a) \cap E(b) = \emptyset$, i.e. two equivalence classes are either identical or disjoint. Indeed, if $x \in E(a) \cup E(b) \neq \emptyset$, then $x \sim a$ and $x \sim b$, and because of Definition 3.3, $a \sim b$. But this implies that each x in $E(a)$ is equivalent to b , and so $E(a) = E(b)$. ■

Fig. 3.2 A partition $\mathfrak{P} = \{P_1, P_2, \dots, P_6\}$ of a set A which is given by disjoint subsets P_i



As a summary of our consideration of equivalence relations we may say that every partition gives rise to an equivalence relation and vice versa.

3.2.1 Cartesian Product, Product Set

The ordered pair is an entity which consists of a pair of elements distinguished as a *first element* and a *second element* of the ordered pair. The ordered pair with a first element $a \in A$ and second element $b \in B$ is denoted (a, b) . Thus, two elements of a set, e.g. $\{a, b\} = \{b, a\}$ are not ordered, whereas for the ordered pair $(a, b) \neq (b, a)$.

Definition 3.4 (Cartesian Product) The set of ordered pairs of elements from some sets A and B , denoted $A \times B$, $A \times B = \{(a, b) \mid a \in A, b \in B\}$, is called the *Cartesian product* or *product set* of A and B .

One can iterate the operation of taking Cartesian products, e.g. $A \times (B \times C)$. In this case, no distinction is made between $A \times (B \times C)$ and $(A \times B) \times C$. Both are considered to be the same set of triple Cartesian products (3-tuples) $A \times B \times C = \{(a, b, c) \mid a \in A, b \in B, c \in C\}$. More generally, we only use the n-fold Cartesian product $A_1 \times A_2 \times \dots \times A_n$ rather than the many different ones which could be obtained by distributing parentheses so as to take the products two at a time. A repeated use of the same set is generally denoted using exponential notation, e.g. $\mathbf{R} \times \mathbf{R} \times \mathbf{R} = \mathbf{R}^3$ which is the common 3-dimensional Euclidean vector space.

Remark 10 Relations between sets can be depicted visually in a *Venn diagram* as figures in the plane. Some examples are given in Fig. 3.3.

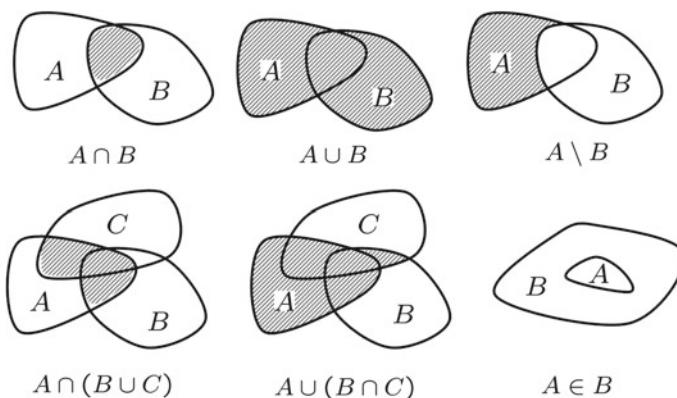


Fig. 3.3 Some set relations depicted as Venn diagrams

3.2.2 Functions and Linear Spaces

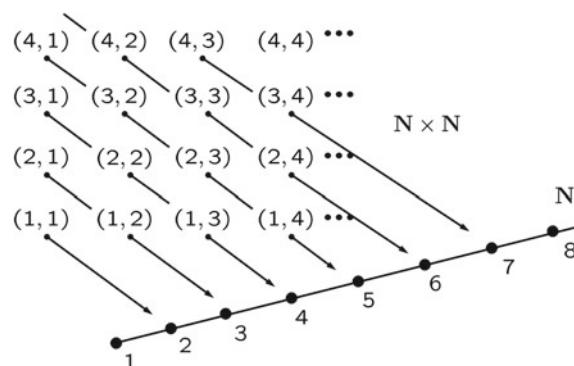
One of the central ideas in mathematics (and consequently, also in physics) is that of a function. A *function* f from A into B is a rule which assigns to each member a of a set A *exactly one* element $b \in B = fa$. The set A is the *domain* of $f : A \rightarrow B$. The *image* (range, target) of f is $fA = \{fa \mid a \in A\} \subset B$. The set B is called the *range set* of f . An element of the range, $b = fa$, is called a *value* of f , or the *image* of a under f . It is convenient at times to refer to $f : A \rightarrow B$ as the *graph* of a function, which is defined as the subset $\{(x, f(x)) \mid x \in A\}$ of $A \times B$.

Remark 11 The distinction between the graph of a function and the function itself is not a conceptual one, but it is mere linguistic convenience; the graph *is* the function. Synonyms for “function” are “transformation”, “map”, “mapping” or “operator”, the latter usually reserved for functions in “linear spaces”. It is customary to write $f(a)$ instead of fa , which we used above. In the notation of physical equations, we shall avoid the customary parentheses when we consider operators, such as ∇f or $Df = f'$ unless they are required to resolve ambiguity. Parentheses have to be used where the element a is itself composite, e.g. $f(a + b)$ is not the same as $fa + b = (fa) + b$.

Example 18 (Addition of Positive Integers) The addition of positive integers is a function $f : \mathbf{N} \times \mathbf{N} \rightarrow \mathbf{N}$. It can be specified by the formula $f(x, y) = x + y$, or as the set $f = \{((x, y), x + y) \mid x, y \in \mathbf{N}\}$. A graphical representation of this function is given in Fig. 3.4.

Example 19 (Metric Space) In physics one needs to be able to calculate the distance between certain objects, e.g. atoms, particles, molecules, events, centers of masses, etc. Thus, one needs a proper definition of “distance” and a recipe to calculate it. If, for example, one thinks of objects of a set X in n -dimensional space \mathbf{R}^n , a function d that yields the distance between two points $x, y \in X$ in \mathbf{R}^n can be defined as, e.g.: $d(x, y) = (\sum_{i=1}^n |x_i - y_i|^2)^{1/2}$. This formula specifies a function $d : X \times X \rightarrow \mathbf{R}^n$, which is endowed with three special properties: For arbitrary numbers

Fig. 3.4 Graphical representation of the function $f = \{((x, y), x + y) \mid x, y \in \mathbf{N}\}$



x , y , and z :

- (a) $d(x, y) \geq 0$, $d(x, y) = 0$ iff $x = y$,
 - (b) $d(x, y) = d(y, x)$,
 - (c) $d(x, y) \leq d(x, z) + d(y, z)$.
- (3.1)

This function d is said to be a *metric*; X together with d constitute a *metric space* (X, d) . The metrics defined here are subtly different from the “physicist’s metrics” which are not positive definite (property (a)) and which are used e.g. in the theories of special and relativity. These metrics are introduced in Sect. 3.7. For the SRT/GRT metrics the axiom (a) of (3.1) is not valid, i.e. the metrics in physics are not positive definite, therefore, these metrics are correctly called “pseudo-metrics”.

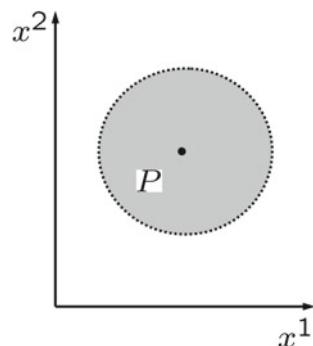
In any metric one can define the “open ball” with center x and radius $r > 0$ with respect to the metric d as:

$$B(x, r) = \{y \mid d(x, y) < r\}. \quad (3.2)$$

This set is illustrated for \mathbf{R}^2 and its natural Euclidean metric in Fig. 3.5. The open ball of a point X in a manifold is one possibility to extend the notion of an ϵ -neighborhood in \mathbf{R} into \mathbf{R}^n . We will return to this definition when considering open sets as basis neighborhoods of topological manifolds in Sect. 3.3.

If $f : A \rightarrow B$ (or, in short notation $fA = B$), i.e. each element of B has at least one element of A mapped into it, then one says that f is *onto* (or “surjective”), or that f maps A onto B (in contrast to “into” above). If finally for every $b \in fA$ there is just *one* $a \in A$ such that $b = fa$, i.e. $f(a_1) = f(a_2) \Rightarrow a_1 = a_2$ for all members of A , then f is said to be *one-to-one* (or “injective”), or short 1-1. In order to inverse a function $f : A \rightarrow B$, it is necessary that the equation $y = f(x)$ has for each $y \in B$ *exactly one* solution $x \in A$. For the existence of a solution of the equation $y = f(x)$ in the first place, the function f has to be surjective and for f having not more than one solution in A , it has to be injective. Thus, for a function f which bijective, i.e. both, one-to-one and onto, the *inverse* of f , $f^{-1} : fA \rightarrow A$ exists, and is defined by setting $f^{-1}fa = a$.

Fig. 3.5 Graphical representation of the neighborhood (open ball) of a point P in \mathbf{R}^2 . The interior of the disk is bounded by the circle of radius r . The circle itself - denoted by a dotted line - is not part of this neighborhood



Example 20 Consider a function $f : \mathbf{R} \rightarrow \mathbf{R}$. Then $f(x) = x^3 - x$ is onto, but not 1-1, $f(x) = \exp(x)$ is one-to-one, but not onto, $f(x) = x^3$ is both and $f(x) = x^2$ is neither, c.f. Fig. 3.6.

If $f : A \rightarrow B$ and $g : B \rightarrow C$, then the *composite* of g and f , denoted $g \circ f$, is the function obtained by following f by g , applied to every $a \in A$ for which this makes sense: $(g \circ f)a = g(fa)$, c.f. Fig. 3.7.

Note that in physics literature, the symbol “ \circ ” is rarely used to denote composite functions; it is common to use several brackets, e.g. $g(f(x))$.

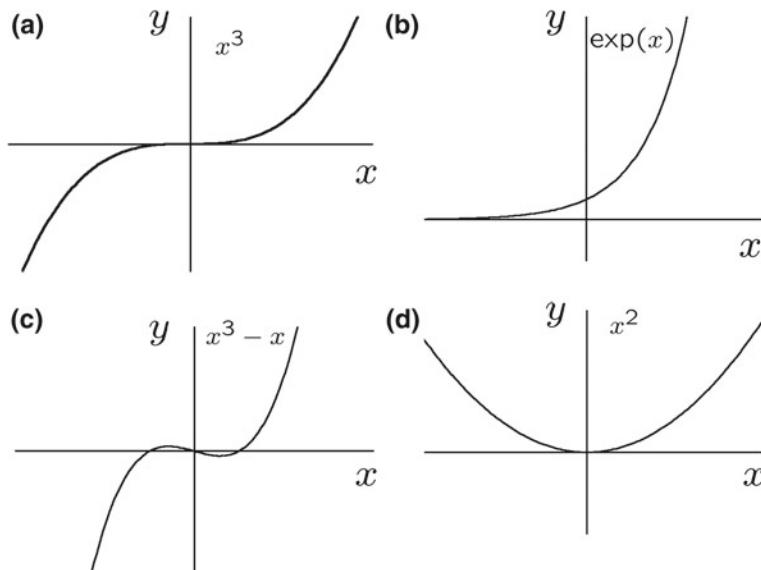


Fig. 3.6 Graph of the functions **a** x^3 (onto, and 1-1), **b** $\exp(x)$ (1-1, not onto), **c** $x^3 - x$ (onto, not 1-1), **d** x^2 (neither)

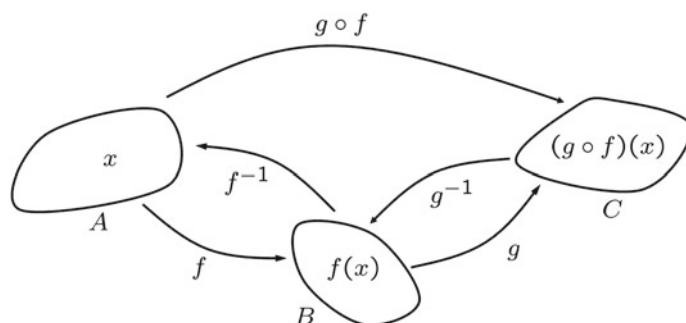


Fig. 3.7 Illustration of functions f , g and their composite $g \circ f$ as well as the inverse f^{-1}

Next, we use set theory to consider linear structures and we will define a *linear vector space* \mathcal{L} which is the foundation of the definition of the Euclidean space \mathbf{R} , respectively its generalization to n dimensions \mathbf{R}^n .

The important concept of a *linear vector space* which is defined over a field \mathbf{K} is introduced with the next definition. For the definition of a field \mathbf{K} we refer the reader to Box 3.1.

Linear Vector Space \mathcal{L}

It is most straightforward to start with a definition of a linear vector space which is provided in Box 3.2.

Box 3.1 The definition of a ring and a field

A set M of objects a, b, c, \dots that obey the following axioms of addition and multiplication is called a **commutative ring**:

1. $a + (b + c) = (a + b) + c$ (associativity)
2. $a + b = b + a$ (commutativity (Abelian group property) with respect to addition)
3. $a + b = c$ (existence of a neutral and negative element of addition)
4. $ab = ba$ (commutativity (Abelian group property) with respect to multiplication)
5. $a(bc) = (ab)c$ (associativity with respect to multiplication)
6. $a(b + c) = ab + ac$
7. $(b + c)a = ba + ca$ (distributivity)

If additionally the following axioms are fulfilled by the objects of M then it is called a **field**.

8. $a1 = a = 1a$ (existence of a neutral element of multiplication)
9. $ab = 1 = ba$ (existence of an inverse element of multiplication)

Example: The real numbers \mathbf{R} and the complex numbers \mathbf{C} are fields.

Box 3.2 The definition of a linear vector space \mathcal{L}

A linear vector space \mathcal{L} over a field \mathbf{K} is a set \mathbf{X} of elements – called vectors – for which commutative addition and a neutral element of addition “0”, as well as a multiplication of a vector a of \mathbf{X} with a number λ of the field \mathbf{K} are defined with the following properties:

$$\text{if } a, b \in \mathbf{X} \implies a + b \in \mathbf{X}, \quad a + b = b + a, \quad (3.3a)$$

$$a + 0 = a, \quad a, 0 \in \mathbf{X}; \quad (3.3b)$$

$$\lambda(a + b) = \lambda a + \lambda b; \quad (3.3c)$$

$$(\lambda + \beta)a = \lambda a + \beta a; \quad (3.3d)$$

$$\lambda(\beta a) = (\lambda\beta)a. \quad (3.3e)$$

Here, $a, b \in \mathbf{X}$ and $\lambda, \beta \in \mathbf{K}$. The set \mathbf{X} is an Abelian group due to the commutativity of the basic operations. An Abelian group with the properties (3.3a–3.3e) is called **linear vector space** or simply **linear space**. If $\mathbf{K} = \mathbf{R}$ (the field of real numbers) then \mathcal{L} is called real vector space, if $\mathbf{K} = \mathbf{C}$ then \mathcal{L} is called a complex vector space. Example: The set of $(m \times n)$ matrices (for fixed m and n) constitute a linear vector space, or polynomials $P_n(x)$ with coefficients in \mathbf{R} or \mathbf{C} .

Remark 12 The addition in Box 3.2 is a binary operation $+ : \mathcal{L} \times \mathcal{L} \rightarrow \mathcal{L}$, the multiplication with an element of \mathbf{K} is a map $\cdot : \mathbf{K} \times \mathcal{L} \rightarrow \mathcal{L}$.

Example 21 (The Euclidean linear vector space \mathbf{R}^n) We have already made use of the a generalization of the space of the real numbers \mathbf{R} which is obtained by defining the n -tupel

$$x = (x^1, x^2, \dots, x^n), \quad x^i \in \mathbf{R}. \quad (3.4)$$

The space \mathbf{X} of Box 3.2 is then called \mathbf{R}^n , because an n -tupel of real numbers are needed for its definition. The addition of two n -tupels is defined component-wise and a scalar multiplication with an element $\lambda \in \mathbf{K} = \mathbf{R}$ is defined by $\lambda x = (\lambda x^1, \lambda x^2, \dots, \lambda x^n)$. Thus, \mathbf{R}^n is a linear vector space. Generally, the space \mathbf{R}^n is called *Euclidean* vector space.

Remark 13 Euclidean spaces in mathematical literature are defined in many different ways, depending on the author and context. As a result there is a whole host of slightly different meanings of “Euclidean space” and it is not always clear, as to how much algebraic, analytic or geometric structure is involved when this term is used. Among many standards to choose from are:

- The set \mathbf{R}^n together with its usual Lebesgue measure.
- The set \mathbf{R}^n with its usual topology.
- The linear space \mathbf{R}^n with its usual differentiable structure.
- The linear space \mathbf{R}^n with a metric but no inner product.
- Flat affine space of n dimensions with inner product.

In this work we will use the term “Euclidean space” for the linear vector space \mathbf{R}^n with a positive definite inner product.²

The importance of the concept of a linear vector space lies in its generality; this concept can be applied to many different and general objects such as matrices, polynomials, functions or tensors which – by definition – are linear vector spaces. Linearity is an important concept as it implies the usual rules for addition and multiplication which are used as axioms for the real number system. The terms “linear space” and “vector space” are often used interchangeably in mathematical and physical literature (and also in this text), but the term “vector space” refers to the historical context of physics and geometry whereas the term “linear space” refers to the modern, abstract definition of spaces which is based on set theory. This term emphasizes the fact that the elements of a linear space are not necessarily vectors with end-points embedded in a larger linear space. In physics it is common to denote vectors, i.e. objects of a linear vector space with the arrow symbol, i.e. \vec{a} instead of the common mathematical bold-face notation **a**.

²For a definition of the inner product, see Box 3.6.

Basis and Basis Vectors

The general concept of a *basis \mathbf{B} of a vector space \mathbf{X}* allows for a different representation of the objects of a vector space. The elements of \mathbf{B} are called basis vectors. Let $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ be arbitrary vectors in \mathbf{X} and $\lambda_1, \lambda_2, \dots, \lambda_n$ be numbers of the field \mathbf{K} then the sum

$$\sum_{i=1}^n \lambda_i \mathbf{a}_i \quad (3.5)$$

is called a *linear combination* of the vectors $\mathbf{a}_i \in \mathbf{X}$. The numbers $\lambda_i \in \mathbf{K}$ are called coefficients. If a finite system of vectors $S = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\} \subset \mathbf{X}$ is *linear independent* and if every vector $\mathbf{a} \in \mathbf{X}$ can be written as linear combination of vectors of S then S is called a Basis \mathbf{B} .

With a basis, all objects of \mathbf{X} can be expressed as a linear combination of basis vectors, e.g. if $\mathbf{X} = \mathbf{R}^n$, a vector \mathbf{a} can be written by means of linear independent basis vectors \mathbf{b}_i as:

$$\mathbf{a} = \sum_{i=1}^n a^i \mathbf{b}_i . \quad (3.6)$$

The numbers a^i are the components of the vector \mathbf{a} in the basis representation and thus can be replaced by its components ($\mathbf{a} = \{a^i\}$, $i = 1, 2, 3$).

Remark 14 In (3.6) we have used the common index notation for coordinates with a superscript index i , indicating the concepts of a *contravariant* vector. The reason for this notation will soon become clear when tensors are introduced as geometric objects in Sect. 3.3.4, which can be represented with their components in a co- and contravariant basis.

Particularly useful for the component description of vectors are basis vectors which are pairwise orthogonal to each other because such an orthogonal set of vectors is linear independent. It is common to normalize all basis vectors. Thus, one obtains an *orthonormal* basis system, i.e.

$$\mathbf{b}_i \mathbf{b}^j = \delta_i^j = \delta_{ij} , \quad (3.7)$$

with the usual definition of the Kronecker-Delta symbol $\delta_i^j = \delta_{ij} = \delta_j^i = \delta_i^j$:

$$\delta_i^j = \begin{cases} 1 & : i = j, \\ 0 & : i \neq j. \end{cases} \quad (3.8)$$

Norm of a Vector Space

In ordinary \mathbf{R}^n the *norm* of a vector (see Box 3.3) has a simple interpretation, namely its length. However, the general definition of a norm is such that it can be used for functions in function spaces (with an infinite number of degrees of freedom) as well.

Box 3.3 The definitions of a normed vector space and a Banach space

A normed space is a vector space over a field \mathbf{K} in which a norm $\|x\|$ with the following properties is defined:

1. $\|x\| \geq 0$
2. $\|x\| = 0 \implies x = 0$
3. $\|\lambda x\| = |\lambda| \|x\|, \lambda \in \mathbf{K}$
4. $\|x + y\| \leq \|x\| + \|y\|$

Convergence of functions is defined by Cauchy series of elements (x_n) . If every Cauchy series converges to the same element of a space, then this space is called **complete**. A **complete normed vector space** is called **Banach space**.

Example 22 (The space of $L^p(X)$ integrable functions) The function space $L^p(X)$ of all functions which are p integrable on $X \subset \mathbf{R}$ consists of all functions f for which the following relation is valid:

$$\|f\|_p = \left(\int_X dx \|f(x)\|^p \right)^{1/p} < \infty. \quad (3.9)$$

A special case of this function space is the space of the square integrable functions $L^2(X)$ which is a *Hilbert space*. This is the fundamental space which is used in quantum theory. It is also a Banach space, i.e. it is complete. The main reason why this space plays a fundamental role in quantum theory, is the fact that the quantum mechanical motion of particles, expressed in the Schrödinger equation, is based on a wave function ψ which can be decomposed into an infinite series of plain waves, mathematically described by Fourier integrals. These, in turn belong just to the class $L^2(X)$.

A normed vector space is automatically a metric space as the norm $\|x - y\|$ of two points can be used to deduce a metric, i.e. a distance function $d(x, y) = \|x - y\|$. The opposite is not generally true. A vector space with scalar product is called “pre-Hilbert space”, cf. Fig. 3.8. Based on the general concept of a linear vector space we can proceed to define a *linear function* and the important notion of a *linear form*.

Definition 3.5 (Linear Function, Linear Form) A function $f : A \rightarrow B$ is said to be *linear* if A and B are linear vector spaces over a common field \mathbf{K} and if f has the following properties of *additivity*

$$f(x + y) = f(x) + f(y), \quad x, y \in A, \quad (3.10)$$

and *homogeneity*

$$f(\lambda x) = \lambda f(x), \quad \lambda \in \mathbf{K}, x \in A. \quad (3.11)$$

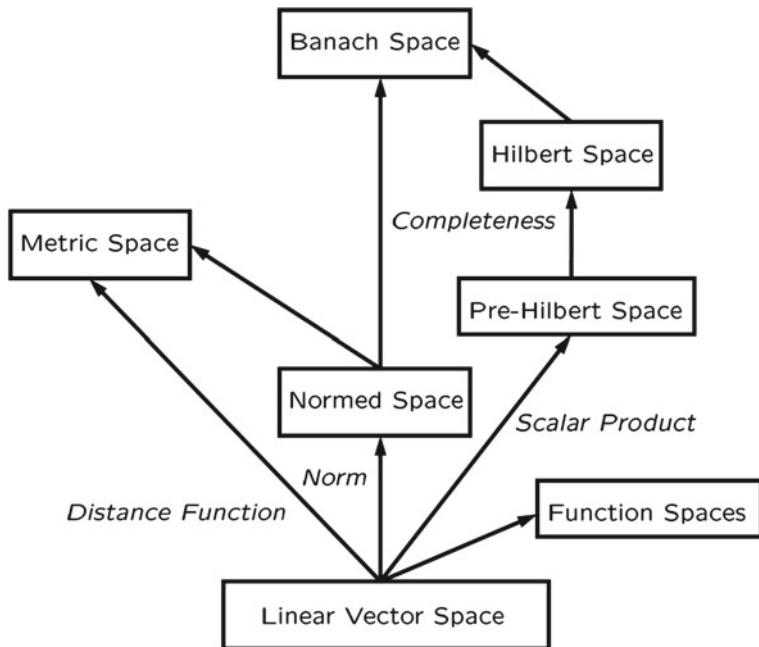


Fig. 3.8 Illustration of the hierarchy of some important linear vector spaces in computational physics

Box 3.4 Some frequently used function spaces

Let X be a subset of \mathbf{R}^n . Then:

$C^m(X)$: m -times smoothly differentiable function on X .

$C^\infty(X)$: arbitrary times smoothly differentiable function on X .

$C_0^\infty(X)$: arbitrary times smoothly differentiable function on X which is only different from 0 on a closed subset of X .

Box 3.5 The scalar product in a vector space

Let M be a subset of \mathbf{R}^n . Then:

The *scalar product* in a vector space M is a positive hermitian function: $M \times M \rightarrow \mathbf{C}$ (or \mathbf{R}), denoted as (x, y) with $x, y \in M$ and $\lambda \in \mathbf{K}$, which obeys the following axioms:

1. $(x, y) = (x, y)^* \Rightarrow (x, x) \in \mathbf{R}$
2. $(\lambda x, y) = \lambda^*(x, y)$ and $(x, \lambda y) = \lambda(x, y)$ (anti-linearity)
3. $(x + y, z) = (x, z) + (y, z)$ and $(x, y + z) = (x, y) + (x, z)$
4. $(x, x) \geq 0$
5. $(x, x) = 0 \Rightarrow x = 0$

A linear function $f : A \rightarrow A$ of a vector space onto itself is called *linear transformation*. If the range set of a linear function $f : A \rightarrow \mathbf{K}$ is the field \mathbf{K} of the vector space A , then f is called a *linear functional* or *linear form*. Functions are objects that “exist” in a function space. Frequently used spaces for functions in physics are denoted in Box 3.4. The representation of objects such as functions in terms of their components in some appropriately chosen basis allows for a generalization of the *scalar product* of two vectors, say $f, g \in L^2(X)$, written in components:

$$(f, g) = \int_X dx f^\star(x)g(x), \quad (3.12)$$

where “ \star ” denotes the conjugate complex. The scalar product in (3.12) fulfills all axioms of a vector space (Box 3.5).

A summary of the relations of some linear spaces that are of importance in multiscale computational physics is depicted in Fig. 3.8

3.3 Topological Spaces

Topology is a branch of mathematics designed to let one handle the concept of “continuity” without the additional baggage of “distance” or “differentiability”. It is possible and very useful to keep these concepts distinct. Continuity is defined in terms of neighborhoods or “open sets”. An open set is a set which has no boundary. This means that all points of an open set are “interior points”, which means that they are surrounded by other points of the same set, cf. Theorem 3. The word “topology” was introduced by J.B. Listing in 1847 in “Vorstudien in Topologie” [202]. The word “topology” is derived from Greek and means the study or analysis of place or locality. The fundamental structural unit of topology is the neighborhood which is a subset of the whole space. The neighborhood of each point is the set of nearest neighbors of that point. This concept is introduced in Definition 3.8 on p. 129.

In the case of a metric space, an infinite set of neighborhoods around each point is required to define a notion of locality. These neighborhoods form an infinite sequence of sets of progressively smaller diameter closing in on a given point. This corresponds to the ε - δ concept of continuity, where locality is expressed in terms of the limit of arbitrarily small distance. Metric spaces are discussed in Sect. 3.4.

The most general definition of a topology is expressed in terms of an arbitrarily large, possibly uncountable, set of neighborhoods for each point in space. Then, for example, a point is said to be a limit of a sequence of points if all neighborhoods of the point contain an infinite number of elements of the sequence. In some textbooks topology is defined as the study of those properties of topological spaces which are invariant under homeomorphisms which is in line with Felix Klein’s 1872 “Erlanger Programm” according to which any geometry is characterized by its invariants. However, the mere study of homeomorphisms (a mapping which maps open sets onto open sets) is not all that is to the study of topology.

The purpose of this section is to introduce just enough technical issues of topology in order to be able to understand the mathematical concept of space in the definition of a manifold which is a fundamental entity used in theoretical physics.

We start our considerations on topology with some set X , consisting of arbitrary points and call this set a “space”. This notion of space need not have anything to do with the intuitive, primitive notion of space. Next, we want to be able to discuss relations between the points, i.e. the elements of X , e.g. the concept of continuity or the neighborhood of a point, without having to introduce the concept of “distance” or “differentiability”. These will be introduced later. The important concept of an open set is defined in the following theorem.

Theorem 3 (Open Sets) *A set X is open, iff the interior of the set, denoted X^0 , equals the set, i.e. $X^0 = X$.*

Remark 15 As an illustration the reader could imagine a set X as a line, cf. Fig. 3.9a. These points are no inner points of X . For an arbitrary point to be an inner point of a distance, one should be able to draw a circle around it which lies completely within the set X . Each circle around a point of a line however lies at least partially outside this line. Accordingly, one can appreciate that the boundary points of a circle are no interior points of the boundary, cf. Fig. 3.9b. However, if one looks at a circle without its boundary, then all points of it are inner points, cf. Fig. 3.9c. This is because every point of the circle without boundary has some positive distance from the boundary and thus one can define a circle around this point which lies completely in the interior of the point set of the circle without boundary.

To discuss relations between points of a set one has to define sets which pertain to these relations, in other words, one has to define subsets (power sets) of X . In doing so, we do not allow arbitrary subsets, but the family of sets which belong to the

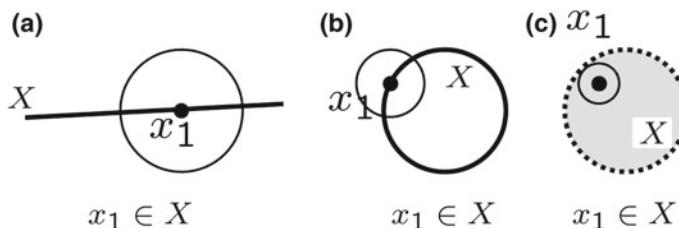


Fig. 3.9 Illustration of open sets X . **a** A point $x_1 \in X$ of a line is not an interior point of the set X ; **b** The boundary points of a circle are no interior points of the boundary; **c** All points of a circle without its boundary (displayed as *dashed line*) are inner points

power set $P(X)$ have to have certain properties which finally leads to a definition of a topological space as follows:

Definition 3.6 (*Topology, Topological Space*) A topology \mathfrak{T} on a set X is a subset \mathfrak{T} of $P(X)$, $\mathfrak{T} \subset P(X)$, such that the following axioms are satisfied:

- (a) $\emptyset \in \mathfrak{T}$,
- (b) $X \in \mathfrak{T}$,
- (c) The union of any finite or infinite collection of open sets is open,
- (d) The intersection of any finite number of open sets is open.

The ordered pair (X, \mathfrak{T}) is called a *topological space*; it is common to simply call X the topological space and \mathfrak{T} the topology. The set X in a tuple always satisfies $X = \cap \mathfrak{T}$. The elements of \mathfrak{T} are called the *open sets* of X .

Remark 16 With the extensive use of sets when developing basic mathematical concepts, it becomes apparent why it was necessary to precede a study of the properties of sets in Sect. 3.2. A topology for a set X is a set of subsets of X which contains the empty set, the set X , and any finite intersection or arbitrary union of sets in X . In (d), only *finite* numbers of open sets are allowed as the intersection of an *infinite* number of open sets may possibly contain points which are no inner points. In formal notation this reads: if $X_1, X_2 \in \mathfrak{T}$, then $X_1 \cap X_2 \in \mathfrak{T}$, and (c) formally reads: if $\{X_i \mid i \in \mathcal{I}\} \subset \mathfrak{T}$, then $\bigcup_{i \in \mathcal{I}} X_i \in \mathfrak{T}$.

The notion of a topological space is the abstract mathematical formalization of the idea of “space”, based only on the properties and relations of open sets without reference to how these sets are built. Note, that the same space can have many different topologies. In particular, there are always the *discrete topology* for which the topology equals the power set ($\mathfrak{T} = P(X)$) and the *concrete topology* for which $\mathfrak{T} = \{\emptyset, X\}$. These are so trivial as to be practically useless, except as a source of interesting counter examples in purely mathematical texts.

Example 23 Suppose that $X = \{1, 2\}$, then the different topologies are:

1. $\mathfrak{T}_1 = \{\emptyset, X\}$ (trivial or concrete topology on the set X),
2. $\mathfrak{T}_2 = \{\emptyset, X, \{1\}\}$,
3. $\mathfrak{T}_3 = \{\emptyset, X, \{2\}\}$,
4. $\mathfrak{T}_4 = \{\emptyset, X, \{1\}, \{2\}\}$.

The discrete topology are all topologies on X .

Note that a set may be only closed or only open, or both, or neither. If $X \subset \mathfrak{T}$, $\{X, \mathfrak{T}\}$ a topological space, then the union of all open sets contained in X is the *interior* of X . Thus $X^0 = \bigcup\{Y \mid Y \subset X \text{ and } Y \in \mathfrak{T}\}$. By (c) of Definition 3.6, the interior of X is an open set itself and is in fact one of the open sets of which we

take the union in its definition. It is the biggest open subset of X . Just as “union” and “intersection” are dual notions, the dual notion to “interior” is “closure”. The closure of $A \in \mathfrak{T}$ is the intersection of all closed sets containing X and is denoted \bar{X} . Thus $\bar{X} = \bigcap\{Y \mid X \in Y \text{ and } X - B \in \mathfrak{T}\}$ is closed and is the smallest closed set containing X . For all subsets A, B of X we have:

Note, that the definition of a topology could equally well be based on closed sets with corresponding axioms to Definition 3.6 which we state as theorems:

Theorem 4 (Closed Sets) A set is closed iff the closure of the set equals the set, i.e.

$\bar{X} = X$. (a) A finite union of closed sets is a closed set.

(b) An arbitrary intersection of closed sets is closed.

(c) \emptyset and X are closed sets.

Complete knowledge of the operation of taking the interior or the closure is adequate to determine the topology of a space. We continue with some important definitions.

Definition 3.7 (Boundary) The *boundary (derived set)* of a set $X \subset \mathfrak{T}$ is the set $\partial X = \bar{X} - X^0$. The elements of ∂X are called *boundary points* of X .

Definition 3.8 (Neighborhood) A *neighborhood* of a point $x \in X$ is any set $A \subset X$ such that $x \in A^0$. In particular, any open set containing x is a neighborhood of x .

Definition 3.9 (Connectivity) A topological space $\{X, \mathfrak{T}\}$ is *connected* if X cannot be partitioned into two non-empty open sets.

Another formulation of the same concept, in terms of its negation, is the following: A topological space $\{X, \mathfrak{T}\}$ is disconnected, iff there are nonempty open subsets $G \subset \mathfrak{T}$ and $H \subset \mathfrak{T}$ such that $G \cap H = \emptyset$ and $G \cup H = X$. The connectivity of sets is illustrated in Fig. 3.10.

One first defines the set of all neighborhoods in the topological space (X, \mathfrak{T}) in which the set of points is defined. A different choice of neighborhoods (which are open sets) gives rise to a different classification of sets into connected and disconnected. The ability of open sets to separate sets is the fundamental task of open sets. Two subsets of a topological space are disconnected by covering them with a pair of disjoint open sets. Note that there is still no concept of “distance” or even “differentiability” introduced.

Example 24 Let a set X contain at least three different points x, y, z . We define

$$\mathfrak{T} = \emptyset \cup \{Y \subset X \mid x \in Y\}. \quad (3.13)$$

\mathfrak{T} is closed with respect to union and intersection, thus it defines a topology $\{X, \mathfrak{T}\}$ on X . The subset $A = \{y, z\}$ contains the open subsets $U = \{y\} = A \cap \{x, y\}$ and $V = \{z\} = A \cap \{x, z\}$ in A . But these two subsets may not be subdivided into disjoint subsets in X , because each such subset contains the point x .

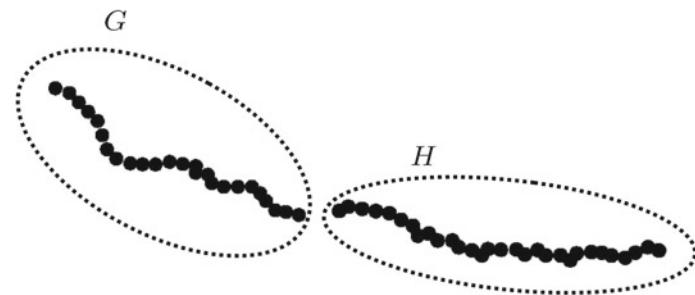


Fig. 3.10 Illustration of two disconnected (open) sets in a topological space. The elements are depicted by *filled dots*. In this example, there are two sets G and H , for which $G \cap H = \emptyset$ and which completely cover X , i.e. $G \cup H = X$. Thus, the neighborhoods of the points in G and H are disjoint. These neighborhoods “disconnect” the set into two non-empty portions. Thus, the topological space is disconnected

Definition 3.6 on p. 128 of a topological space is sufficient to capture the concept of continuity in the following definition.

Definition 3.10 (Continuous Function) Let X, Y be topological spaces. A function $f : X \rightarrow Y$ is *continuous* if for every open set $G \in Y$, $f^{-1}G$ is open in X . In other words, a function f is continuous if f^{-1} maps open sets into open sets.

Remark 17 This is a definition. It is meaningless to try to “prove” it. An image of this definition is provided in Fig. 3.11.

Definition 3.10 is an abstraction of the usual ϵ - δ definition of a continuous function $f : \mathbf{R} \rightarrow \mathbf{R}$, with $y = f(x)$ in real analysis (i.e. in the space \mathbf{R} of the real numbers). Here, the ϵ serves to define a neighborhood of y , given a priori, and the requirement was, that there be a neighborhood of x determined by δ , such that f maps the δ -neighborhood into the ϵ -neighborhood.

Next, we define the convergence of a function f in a topological space X . Convergence of a function f ($\lim_{x_n \rightarrow x} f(x_n) = y$) is generally defined by the convergence of a sequence $\{x_n\}$ in the following definition.

Definition 3.11 (Limit, Convergence) A sequence $\{x_n\}_{n \in \mathbf{N}}$ in a topological space X is said to have a limit $x \in X$ if each neighborhood of x contains almost all members of x_n , that is the set $\{n \in \mathbf{N} \mid x_n \notin U\}$ is finite. The sequence and the function f are then said to be *convergent* in point x . If the sequence is convergent for all points $x \in X$, $\{x_n\}$ and f are said to be convergent in X .

Remark 18 The notion of a *Cauchy series* which underlies the axiom of closedness of the real number system or *uniform closedness* of sequences of functions requires

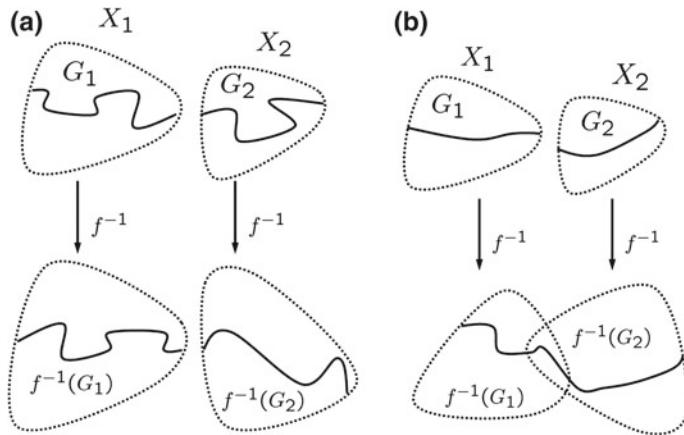


Fig. 3.11 A picture of a continuous function $f : X \rightarrow Y$ which preserves the connectivity of open sets. In **a** the function f is continuous as both, the image (G_1 and G_2) and the pre-image $f^{-1}(G) = f^{-1}(G_1) \cup f^{-1}(G_2)$ are disconnected. In **b** the function f is discontinuous as in the domain of f there is no “gap” ($f^{-1}(G) = f^{-1}(G_1) \cap f^{-1}(G_2) \neq \emptyset$), but in the range of f there is a gap

an additional structure (the order axioms, or a metric based on these axioms), i.e. a notion of “ $<$ ”, “ $=$ ”, “ $>$ ” of the members in an open set.

One problem arises in this definition of a limit of functions which we just state here as a fact: The limit of convergent sequences in topological spaces is in general not unique. In order to achieve uniqueness one has to add an additional property, or structure, which is called the Hausdorff property.

Definition 3.12 (Hausdorff Topologies) A topological space $\{X, \mathfrak{T}\}$ is a Hausdorff space if for every $x, y \in X$, $x \neq y$, there are neighborhoods U, V , or x, y , respectively, such that $U \cap V = \emptyset$. In other words, every two points can be separated by disjoint neighborhoods.

The Hausdorff property is sometimes called *axiom of separation* T_2 . T_2 implies the T_1 axiom of separation which does not require disjoint covering pairs. T_1 states, that for every pair of distinct points x_1 and x_2 , there is an open set X_1 which contains x_1 but does not contain x_2 , cf. Fig. 3.12. Due to axiom T_1 , in a Hausdorff space the singleton sets (sets with only one element) x are closed sets.

It is important to understand that complete general topological spaces are rather uninteresting even to mathematicians, cf. Remark 16 on p. 128. Therefore, one usually studies some special classes of topological spaces which have additional “interesting” properties. One of these properties is the Hausdorff separability property.

This property is also essential for the formulation of laws of nature in physics which is done with differential equations in continuous space time. In physics one

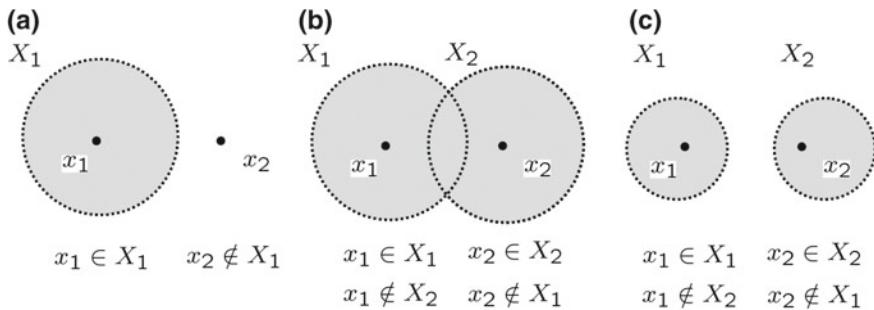


Fig. 3.12 **a** A T_1 space which separates points $x_1, x_2 \in X$ such that $x_1 \in X_1$ and $x_2 \notin X_1$. **b** T_1 separation does not require disjoint covering pairs. **c** A Hausdorff T_2 space which requires disjoint covering pairs

needs to be able to measure distances between the studied objects. This can be done by defining a distance function d , thus yielding a metric space according to (3.1) on p. 119. In metric spaces one can certainly define continuity; but you do not need to be in a metric space to do so. That is the whole point of going through the trouble of defining a general topological space. In \mathbf{R} (or generally in metric spaces) continuity in the topological sense is equivalent to the standard $\varepsilon - \delta$ sense used in real analysis. Here, a function $f : \mathbf{R} \rightarrow \mathbf{R}$ is said to be continuous at a point x iff

$$\forall \varepsilon > 0 \ \exists \delta > 0 : |y - x| < \delta \implies |f(y) - f(x)| < \varepsilon. \quad (3.14)$$

This version of continuity in classical analysis has now largely been superseded by the topological version in Definition 3.10. In (3.14) we have made use of the very useful succinct notation symbols \forall (read: “for all”) and \exists (read: “there exists”).

Out of all the possible topologies which can be attached to a metric space, there is a single canonical topology which is generated by the set of all open balls in the metric space, cf. the definition on p. 119. This is the “topology induced by the metric d ”, denoted as *metric topology*.

Definition 3.13 (*Topology induced by a metric*) The topology on a metric space (M, d) is the topology induced by the metric function d on M and is written explicitly as:

$$(d, \mathfrak{T}) = \{A \subset X \mid \forall x \in A \ \exists r = r(x) > 0 \text{ such that } B(x, r) \subset A\}. \quad (3.15)$$

Remark 19 In any metric space the metric topology is Hausdorff.

Proof Let x and y be distinct, and let $\delta = d(x, y)$ be the distance between x and y which is – according to (3.1) – guaranteed positive. Now consider the open balls $B_1 = B(x, \delta/3)$ and $B_2 = B(y, \delta/3)$. Then $B_1 \cap B_2 = \emptyset$ due to the triangle inequality. ■

Since all spaces which are of relevance to physics are metric spaces, and since all metric spaces are Hausdorff, nearly all of applied mathematics and physics takes place in some sort of metric space. Thus, one does not have to worry too much about non-Hausdorff topologies. The topology in Definition 3.13 is well-defined because an arbitrary set of subsets of any set M will always generate a topology on M . This does not necessarily mean that the topology will have any nice properties. For example, if $d(x, y) = 0$ for all $x, y \in M$, the topology will be trivial.

Finally, we conclude the preliminary work on topologies in this chapter with the definition of a *homeomorphism* which later allows for a terse definition of a Riemannian manifold:

Definition 3.14 (Homeomorphism) A *homeomorphism* between two topological spaces $\{X_1, \mathcal{T}_1\}, \{X_2, \mathcal{T}_2\}$ is a bijection $f : X_1 \rightarrow X_2$ such that both f and f^{-1} are continuous. X_1 and X_2 are said to be topologically equivalent spaces.

A property of a topological space is said to be a *topological property* if every homeomorphic space has the property. A topological *invariant* is a rule which associates to topological spaces an object which is a topological property of the space. The object usually consists of a number or some algebraic system. With the next definitions we approach the concept of manifolds in physics.

3.3.1 Charts

We start with the following definition.

Definition 3.15 (Charts, Coordinate Systems) A *chart* or *coordinate system* $(U, f, f(U))$ on an open subset $U \in \mathcal{T}$ is a set $f(U) \subset \mathbf{R}^n$ together with a homeomorphism $f : U \rightarrow f(U) \subset \mathbf{R}^n$ such that the image $f(U)$ is open in \mathbf{R}^n . The inverse of f is called *local parameterization*.

The fact that f is a homeomorphism means that f is invertible and because U is an open subset in \mathcal{T} we have induced a topology $\{U, \mathcal{T}\}$ on M . That is, in any locally Euclidean space each point x lies in at least one chart. These charts can be thought of as “maps” (in the cartographic sense) that cover part of the space X , cf. Fig. 3.13.

Remark 20 It is better to use the term “chart” in manifold theory as the word “map” already has a technical meaning in mathematical physics. Why is U taken as subset of \mathbf{R}^n ? The answer is, that \mathbf{R}^n is a locally Euclidean flat linear vector space (for a definition see Example 21 on p. 122) in which all properties of functions in real analysis – in particular the concepts of differentiation and integration which one can build in \mathbf{R}^n – are well-defined and can be used to develop theories of ordinary and partial differential equations. The language to describe manifolds clearly shows

the subject's origin in cartography of Earth, which is everywhere locally Euclidean approximatively, but not globally. Hence, it is impossible to create an atlas which covers the whole Earth on a single sheet of paper; one has to use at least two overlapping charts.

Definition 3.16 (Coordinates) The image of the chart, $f(M) = U \subset \mathbf{R}^n$, is a set of n -tupels of real numbers. Each point x in M corresponds to a single n -tuple, which are referred to as the coordinates of the point x in the given chart. It is common in the physical literature (see e.g. [96, 203, 204]) to write the coordinates with upper index as x^a with $a = 1, 2, \dots, n$, or x^μ with $\mu = 0, 1, 2, 3$ running over the four spacetime coordinates. Conventionally, time t is chosen as the first coordinate $t = x^0$.

By introducing a chart on a manifold as a bijective mapping of open subsets of a manifold into open subsets of \mathbf{R}^n , we are able to define a coordinate system in a general open point set of a topology. Note that yet there is no concept of differentiability which one needs in order to define physically meaningful laws in the form of differential equations.

3.3.2 Atlas

Next, we obtain an atlas by patching all coordinate charts together. This idea leads to the following definition.

Definition 3.17 (Atlas) An atlas is an indexed collection (a family) of charts $\{(U_\alpha, f_\alpha)\}$ that covers the entire locally Euclidian space \mathbf{R}^n . Thus, the family of charts satisfies two conditions:

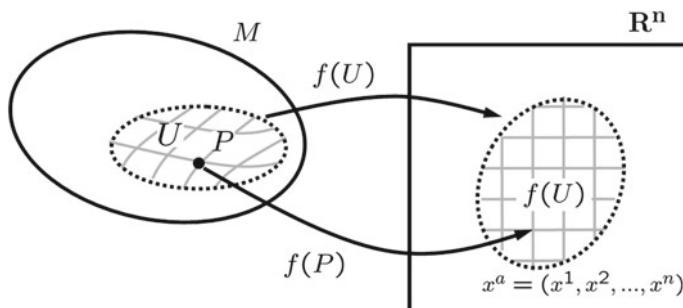


Fig. 3.13 A coordinate chart that maps 1-1 an open subset $U \in M$ onto an open subset $f(U)$ of locally Euclidean points of \mathbf{R}^n . This bijective map associates every point P in M with a unique n -tuple of numbers (x^1, x^2, \dots, x^n) . Thus, by this map M acquires a coordinate system, illustrated by drawing coordinate lines in M which are the images of the Euclidean coordinate lines in \mathbf{R}^n under f^{-1}

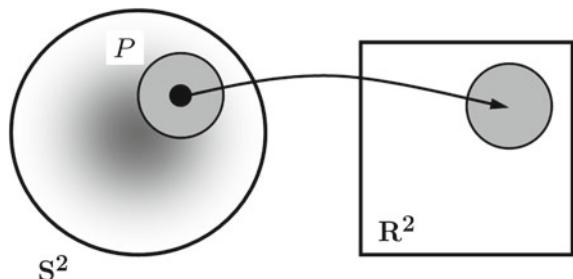
- (a) $\bigcup_{\alpha \in \mathcal{I}} = \mathbf{R}^n$, that is, the U_α cover \mathbf{R}^n ,
- (b) The intersection $U_{\alpha,\beta} = U_\alpha \cap U_\beta \neq \emptyset$ is an open set and the mapping $f_\alpha \circ f_\beta : f_\alpha(U_{\alpha,\beta}) \Leftrightarrow f_\beta(U_{\alpha,\beta})$.

By definition, any locally Euclidean space has at least one atlas. A chart therefore, is simply a coordinate system on some open set, and an atlas is a system of charts which are smoothly related to their overlap.

Example 25 (The sphere S^2 as a manifold) One of the simplest examples of a manifold which illustrates the importance of allowing more than one coordinate chart, is the sphere S^2 . With “sphere”, the two-dimensional surface is meant and not its interior. A two-sphere is characterized by a set of points in \mathbf{R}^3 for which $(x^1)^2 + (x^2)^2 + (x^3)^2 = \text{const}$. Any point on the sphere has a sufficiently small neighborhood which can be mapped 1-1 onto a disc in \mathbf{R}^2 , cf. Fig. 3.14.

As an explicit example of a map, consider the usual spherical coordinates, with $\vartheta = x^1$ and $\varphi = x^2$. The sphere is mapped onto the rectangle $R = \{(x^1, x^2) \mid 0 \leq x^1 \leq \pi, 0 \leq x^2 \leq 2\pi\}$ as shown in Fig. 3.15. It is important to notice that the map breaks down at the poles $\vartheta = 0$ and $\vartheta = \pi$, where one point is mapped onto the whole line $x^1 = 0, 0 \leq x^2 \leq 2\pi$. Thus, at the pole, there is no map. The same happens for points with $\varphi = 0$, which are mapped to two places, $x^2 = 0$ and $x^2 = 2\pi$. Hence, there is no single *non-degenerate* coordinate system, i.e. a map, that covers the whole manifold S^2 and which ascribes a *unique* set of coordinate numbers to each point. To avoid these problems, the map from S^2 onto \mathbf{R}^2 has to be restricted to the open region $0 < x^1 < \pi, 0 < x^2 < 2\pi$. Then the two poles and the curve $\varphi = 0$ joining them are left out on the map. Hence, in the case of a two-sphere, at least two maps are needed to cover it completely. The second one could be a spherical coordinate system where its line $\Phi = 0$ falls into the equator of the first system, e.g. from $\varphi = \pi/2$ to $\varphi = 2\pi/2$. Then every point on S^2 is located at least within one of the two coordinate charts.

Fig. 3.14 1-1 mapping of a small neighborhood of point P on S^2 onto a disc in \mathbf{R}^2



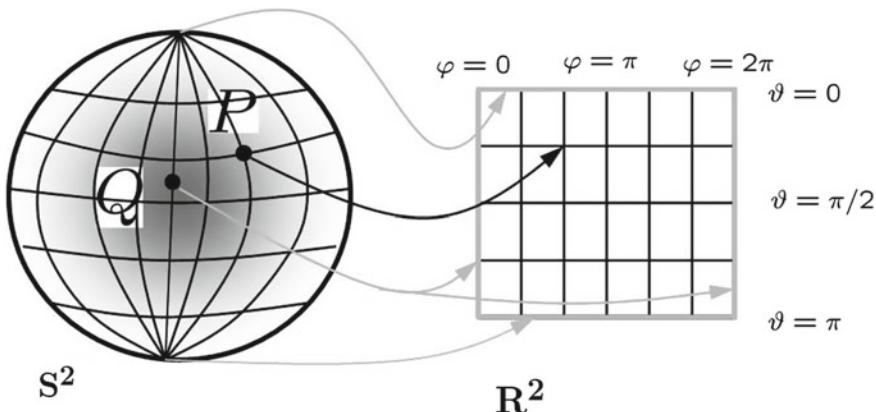


Fig. 3.15 Spherical coordinates are degenerate at the two poles $\vartheta = 0$, $\vartheta = \pi$ and for $\varphi = 0$. Ordinary points like P can be mapped 1-1 onto \mathbf{R}^2 . However, the image of the poles and for all points Q for which $\varphi = 0$ is not uniquely defined

3.3.3 Manifolds

Definition 3.18 (*Topological Manifold \mathfrak{M}*) A n -dimensional topological Manifold \mathfrak{M} is a Hausdorff space with at least one countable basis for the topology on \mathfrak{M} which is locally homeomorph to \mathbf{R}^n .

In other words, topological manifolds are just patched-together pieces of a Euclidean space up to a homeomorphism.

Remark 21 The condition that there be at least one countable basis is sometimes expressed as the property of “paracompactness”, “second-countable” or “metrizable”. All of these terms imply each other and thus are essentially equivalent. What is important is the fact that this property serves to keep the number of charts under control. Second-countable means that there is at least one countable basis for the topology on \mathfrak{M} and implies that there is at least one way of covering the space with a countable number of charts.

If there are two charts $\phi : U_\alpha \rightarrow \mathbf{R}^n$ and $\psi : U_\beta \rightarrow \mathbf{R}^n$ with $U_\alpha \cap U_\beta \neq \emptyset$ then on the open set $U_\alpha \cap U_\beta$ there are two charts defined, cf. Fig. 3.16. The homeomorphism $f_\beta \circ f_\alpha^{-1} : f_\alpha(U_\alpha \cap U_\beta) \rightarrow f_\beta(U_\alpha \cap U_\beta)$ between open subsets of \mathbf{R}^n is called *change of charts*. With the help of an atlas one can transfer any property of \mathbf{R}^n into the manifold \mathfrak{M} . The most prominent example is the notion of the differentiability property of functions. This property however, is only independent of the respective chosen chart if one uses a *differentiable atlas*. Such an atlas has the property that all of its changes of charts (which are mappings of open subsets in \mathbf{R}^n) are C^∞ , i.e. all derivatives of arbitrary order exist and are continuous. Then all changes of

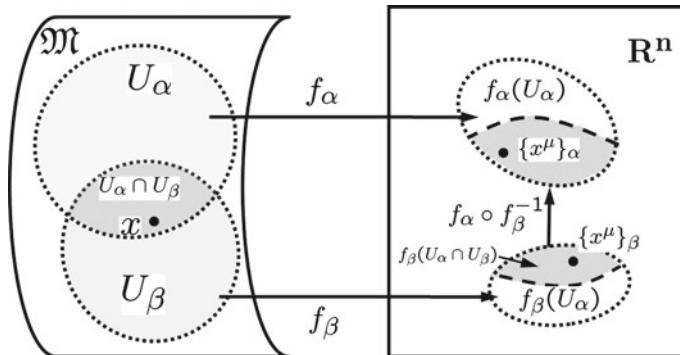


Fig. 3.16 A differentiable manifold \mathfrak{M} with overlapping open neighborhoods U_α and U_β . The coordinate function f_α establishes a coordinate system $f_\alpha(U_\alpha)$ for the coordinate patch U_α in \mathfrak{M} . To each point x in U_α , the function f_α assigns a set of coordinates $\{x^\mu\}_\alpha$. Within the overlap region $U_\alpha \cap U_\beta$ of two different coordinate patches, the compatibility of the local coordinate systems $f_\alpha(U_\alpha)$ and $f_\beta(U_\beta)$ is ensured by the coordinate transformation $f_\beta \circ f_\alpha^{-1}$. This coordinate transformation maps $\{x^\mu\}_\beta$ into $\{x^\mu\}_\alpha$ and identifies them as the coordinates of the same point x

charts are even *diffeomorphisms*, i.e. 1-1 and *differentiable* in both directions. The two charts (U_α, f_α) and (U_β, f_β) are called C^∞ compatible or simply compatible. An atlas $A = \{(U_\alpha, f_\alpha) \mid \alpha \in A\}$ is called differentiable atlas, if $\forall \alpha, \beta \ U_\alpha^{-1} \circ f_\beta$ is compatible. A chart (U, f) is called atlas compatible if it is compatible with every chart $(U_\alpha, f_\alpha) \in A$. Given an atlas on \mathfrak{M} one can define a function $f : \mathfrak{M} \rightarrow \mathbf{R}$ in a neighborhood of a point $x \in \mathfrak{M}$. This function is differentiable if $f_\beta \circ f_\alpha^{-1}$ is differentiable for any chart of the atlas in the neighborhood of $f_\alpha(x)$. As all charts of the atlas are differentiable it does not matter which specific chart is chosen.

Let A be a differentiable atlas. Then the set $S \supset A$ of all atlas compatible charts is called maximum differentiable atlas or, more common, *differentiable structure*.

Remark 22 By placing suitable restrictions on $g \circ f_\alpha^{-1}$ one can also define C^0, C^k, C^ω (analytic) manifolds (cf. Box 3.4 on p. 125). A manifold of class C^0 is a topological manifold with no differentiable structure. Usually, in most applications in materials science, the order k is irrelevant and thus ordinarily C^∞ is meant.

Definition 3.19 (Differentiable Manifold) A differentiable manifold \mathfrak{M} is a topological manifold together with a differentiable structure S .

Example 26 (Euclidean Space \mathbf{R}^3) Euclidean \mathbf{R}^3 is a manifold with a particularly simple globally defined coordinate chart and so it has a one-chart atlas. The same applies to any \mathbf{R}^n .

Example 27 (Minkowski space (\mathbf{R}^4, η)) Minkowski space is the fundamental space in which the theory of special relativity is developed, cf. Example 2 on p. 55. It is

characterized by a (flat) Minkowski metric $\eta_{\alpha\beta}$ ($\alpha, \beta = (1, 2, 3, 4)$) with signature -2 and Minkowski coordinates $x^0 = ct$, $x^1 = x$, $x^2 = y$, $x^3 = z$.

$$(\eta_{\alpha\beta}) = \begin{pmatrix} +1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (3.16)$$

Minkowski space is also a manifold with a *globally* defined coordinate chart.

Remark 23 (Signature of the Minkowski metric) There is no general convention for defining the signature of the Minkowski metric in physical literature. In this book we follow the convention $\text{sign}(\eta_{\alpha\beta}) = (+1, -1, -1, -1)$ that was adopted by *Albert Einstein* (1879–1955) in his original paper in 1916 in which he first published the final form of his field equations [205].³ Many authors also use this signature convention for the spacetime coordinates, e.g. Weinberg [96], Taylor and Wheeler [206], Goenner [207], Landau/Lifschitz [208], d’Inverno [209], Fliessbach [210], Bergmann [211] or H. Weyl [212]. Other authors however, e.g. Misner, Thorne and Wheeler [213] or R. Wald [203] inverse the signature of time and space coordinates, thus using $\text{sign}(\eta_{\alpha\beta}) = (-1, +1, +1, +1)$. Schrödinger [204] uses the signature -2 but with time t as the fourth coordinate ($x^\mu = (x^1, x^2, x^3)$, $\mu = (1, 2, 3)$ and $x^4 = ct$) and Stefani [214] and Schmutzler [215] use the signature $\text{sign}(\eta_{\alpha\beta}) = (+1, +1, +1, -1)$, again with time as the fourth coordinate in Minkowski space.

With the notion of a differentiable manifold \mathfrak{M} one introduces a concept of space where space looks locally like an Euclidean (flat) linear vector space (usually \mathbf{R}^n). However, its *global* properties may be very different. Space is not “absolute” in the sense that there is no global coordinate system which covers all space; The correspondence to the Euclidean “flatness” is only local. In the general theory of relativity, the differentiable Manifold is the fundamental concept of curved spacetime in physics. Thus, in physical applications, the dimensionality of the underlying manifold is usually $n = 4$ and the correspondence of the neighborhoods of certain points in the manifold to the Euclidean flat space has its physical correspondence in the inertial systems of reference (the tangent spaces to \mathfrak{M}) for which the special relativity theory (in a Minkowski space) is valid.

3.3.4 Tangent Vectors and Tangent Space

Next we want to define vectors in a differentiable manifold. For this, we use a curve, defined by a mapping $h : \mathbf{R} \rightarrow \mathfrak{M}$, or, in parameterized notation commonly used

³Actually, Einstein had presented the final field equations in 1915 in a lecture at the Berlin Academy of Science [133].

in physics: $h(\lambda) \in \mathfrak{M}$ for $\lambda \in \mathbf{R}$. Any homeomorphism (1 – 1 mapping) from \mathbf{R} to \mathbf{R} allows to re-parameterize the curve $\lambda \rightarrow \lambda'(\lambda)$ without changing the image $h(\mathbf{R})$ in \mathfrak{M} , and in each chart $U \in \mathfrak{M}$, the curve $h(\lambda)$ passing through point P on \mathfrak{M} , is described by the equations x^i , $i = 1, \dots, n$. Consider another differentiable function $f(\lambda)$ which gives the value of f at the point, whose parameter value is λ . The composite map $f \circ h$ gives a map from \mathbf{R} to $f(U) \subset \mathbf{R}^n$, cf. Fig. 3.17.

$$x^a(\lambda) = f \circ h(\lambda) = f(h(\lambda)) = f(h(x^1(\lambda), \dots, x^n(\lambda))) = f(x^i(\lambda)) \in f(U) \subset \mathbf{R}^n. \quad (3.17)$$

Differentiating and using the chain rule yields:

$$\frac{dh}{d\lambda} = \sum_i \frac{dx^i}{d\lambda} \frac{\partial f}{\partial x^i}. \quad (3.18)$$

As this is true for any function g one can write (3.18) as an operator

$$\frac{d}{d\lambda} = \sum_i \frac{dx^i}{d\lambda} \frac{\partial}{\partial x^i}, \quad (3.19)$$

and interpret this equation such that the set of numbers $\{dx^i/d\lambda\}$ are components of a vector tangent to the curve $x^i(\lambda)$. The use of the term “vector” relies on familiar concepts from Euclidean space, where vectors are defined by analogy with displacements Δx^i . In manifolds however, there is no distance relation between points that have finite distance Δx^i . Thus one needs a definition of vectors that relies only on infinitesimal neighborhoods of points of \mathfrak{M} . Consider two numbers a and b , and $x^i = x^i(\mu)$ which is another curve through P . Then at P

$$\frac{d}{d\mu} = \sum_i \frac{dx^i}{d\mu} \frac{\partial}{\partial x^i}, \quad (3.20)$$

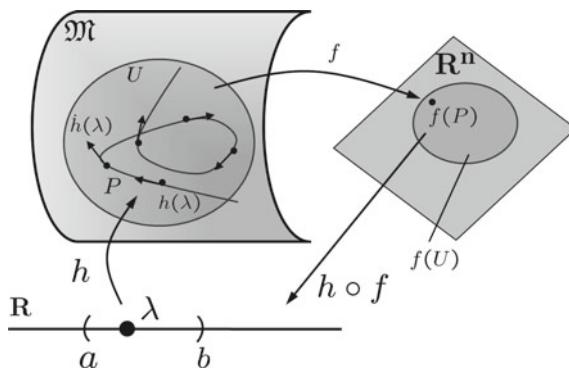


Fig. 3.17 A parameterized curve $h(\lambda) : \mathbf{R} \rightarrow \mathfrak{M}$, $\lambda \mapsto h(\lambda)$ is a function that maps a segment of the real line into a differentiable manifold \mathfrak{M} . The tangent vectors $h(\lambda)$ are the directional derivatives at each point $h(\lambda)$ of the curve. The coordinate map f from a region U of \mathfrak{M} containing point P onto a region $f(U)$ of \mathbf{R}^n gives rise to a composite map $f \circ h$ from \mathbf{R} to \mathbf{R}^n . This composite map is just the expression of $f(P)$ in terms of the coordinates $\{x^\mu\}$

and

$$a \frac{d}{d\lambda} + b \frac{d}{d\mu} = \sum_i \left(a \frac{dx^i}{d\lambda} + b \frac{dx^i}{d\mu} \right) \frac{\partial}{\partial x^i}. \quad (3.21)$$

The numbers on the left hand side of (3.21) are components of a new vector, which is the tangent to some curve through P . Hence, there must exist a curve with parameter ν , such that:

$$\frac{d}{d\nu} = \sum_i \left(a \frac{dx^i}{d\lambda} + b \frac{dx^i}{d\mu} \right) \frac{\partial}{\partial x^i}. \quad (3.22)$$

Comparing this result with (3.21) one obtains:

$$a \frac{dx^i}{d\lambda} + b \frac{dx^i}{d\mu} = \frac{d}{d\nu}. \quad (3.23)$$

Hence, the directional derivatives along curves form a vector space at P . In any coordinate system, there are special curves, the coordinate lines themselves and the derivatives along them are given by $\partial/\partial x^i = \delta^i$. Equation (3.19) shows that $d/d\lambda$ has components $\{dx^i/d\lambda\}$ on this basis. As a result, $d/d\lambda$ is the tangent vector to the curve $x^i(\lambda)$ and thus the space of all tangent vectors at P , and the space of all derivatives along curves at P are in correspondence. This point of view of a tangent vector does not involve displacements over finite differences. We summarize these results in the following three Definitions 3.20–3.22.

Definition 3.20 (Tangent Vector, Components) The *tangent vector* to a parameterized curve $h(\lambda)$ in a point $P = h(\lambda_0)$ is the geometric object, denoted as τ , which has components

$$\tau^a = \frac{dx^a}{d\lambda} \quad (3.24)$$

in any chart U surrounding the point P .

Using the chain rule in \mathbf{R}^n , one can *re-parameterize* ($\lambda \rightarrow \lambda'$) the curve λ as follows:

$$\tau^{a'} = \frac{dx^a}{d\lambda'} = \frac{d\lambda}{d\lambda'} \frac{dx^a}{d\lambda} = \frac{d\lambda}{d\lambda'} \tau^a. \quad (3.25)$$

If we change coordinate patches, U to U' , cf. Fig. 3.16, using

$$f' \circ f^{-1} : f^{-1}(U) \rightarrow f'(U), \quad (3.26)$$

then, again by the chain rule

$$\tau^{a'} = \frac{dx^{a'}}{d\lambda} = \sum_{b=1}^n \frac{\partial x^{a'}}{\partial x^b} \frac{dx^b}{d\lambda} = \sum_{b=1}^n \frac{\partial x^{a'}}{\partial x^b} \tau^b. \quad (3.27)$$

That is: Once the components of the tangent vector τ in a point P have been determined in any one coordinate chart, there are unambiguous rules for changing to any other chart. Also note that it only makes sense to talk about the tangent vector components at a particular point $P = h(\lambda)$ on the curve and in the manifold.

Remark 24 (Notation of Tensors) Tensors can be considered as abstract, geometric objects, the existence of which is – just like in the case of “events” in spacetime – independent of the particular coordinate system that is chosen for a description of these objects. The objects themselves are invariant under coordinate transformations – it is the coordinates that look differently in different charts (coordinate systems). The notation of tensors in this book is such, that the tensors themselves, as geometric objects (without specific coordinates in some chart), are denoted as upright lower case letters such as t , g , b , whereas the components are denoted as $t^a{}_b$, g_{ij} , b_i .

With the next definition we generalize the concept of a tangent vector.

Definition 3.21 (Contravariant Vector) Any geometric object, that transforms in the same way as a tangent vector (3.27) is called a *contravariant vector* or *contravariant tensor of rank 1*.

Remark 25 The denotation “contravariant vector” is actually old-fashioned, because it makes a reference to their explicit transformation behavior under a change of charts, and can mostly be found in older treatments of tensor algebra. In modern formulation, the tangent vectors are simply called “vectors”.

Definition 3.22 (Tangent Space) The set of all possible tangent vectors at a specific point P in a manifold \mathfrak{M} is a vector space, denoted $T_P(\mathfrak{M})$, and is called *tangent space* at P , cf. Fig. 3.18.

Remark 26 (Tangent space of a sphere) It is part of everyday experience that the spherical shape of the Earth can be neglected in small domains; intuitively one approximates the spherical shape by its tangent space and assumes that the axioms of (flat) Euclidean geometry are valid within this space. In a curved non-Euclidean geometry, axioms such as the Pythagorean theorem, or the axiom of parallelism, are not valid anymore, i.e. it has to be defined explicitly what parallelism is supposed to mean.

3.3.5 Covectors, Cotangent Space and One-Forms

There is a second set of distinct vectors which is natural to define in a topological manifold, the *co-vectors* or *contravariant* vectors, a natural extension of the notion of “gradient”. We define the *gradient vector* in a point $P \in \mathfrak{M}$ as follows:

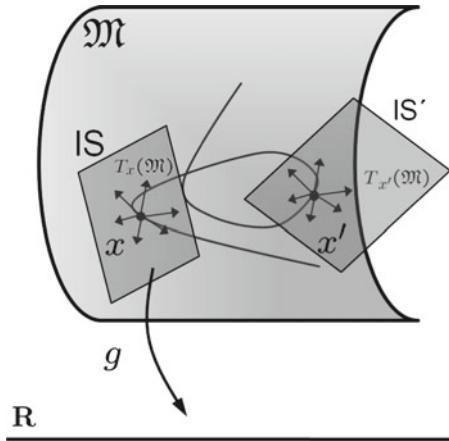


Fig. 3.18 The set of all tangent vectors in a point x in \mathfrak{M} constitutes the *tangent space* T_x over x . A metric tensor (cf. Sect. 3.5) $g : T_x \rightarrow \mathbf{R}$ defines an inner product of two vectors in a single tangent vector space T_x . Note that vectors in different tangent spaces generally cannot be compared; they become comparable with the construction of a curve and a *connection* on the topological manifold (which is not displayed). Tangent spaces $T_P(\mathfrak{M})$ represent the Minkowski spaces of local observers, i.e. the local inertial frames of reference. Each IS is endowed with its own local coordinate time, measured by synchronized clocks and after introducing a metric tensor, distances between points can be measured

Definition 3.23 (Gradient Vector) The gradient to a scalar function ϕ at a point P in a topological manifold \mathfrak{M} is the abstract object, denoted v , for which the components in any chart surrounding P are calculated as:

$$v_a = \frac{\partial \phi}{\partial x^a} . \quad (3.28)$$

Remark 27 (Transformation of a Gradient Vector) After having defined how to calculate the components of the gradient we can look at the transformation behavior of g under a change of coordinate charts $U \rightarrow U'$, using $f' \circ f^{-1} : f^{-1}(U) \rightarrow f'(U)$. By the chain rule we obtain

$$v_{a'} = \frac{\partial \phi}{\partial x^{a'}} = \sum_{b=1}^n \frac{\partial x^b}{\partial x^{a'}} \frac{\partial \phi}{\partial x^b} = \sum_{b=1}^n \frac{\partial x^b}{\partial x^{a'}} v_b . \quad (3.29)$$

That means, once one knows the components of the gradient in any one chart, there are unambiguous rules for changing to any other chart. As in the case of the tangent vector, it only makes sense to talk about the gradient vector components at a particular point P in the manifold. Note that the transformation laws of gradient and tangent vectors are not the same; they are said to be *co-gradient* to each other. To define tangent vectors t one needs a map $\mathbf{R} \rightarrow \mathfrak{M}$, however to define gradient vectors one needs a map $\mathfrak{M} \rightarrow \mathbf{R}$.

With the above said we can now define *covariant vectors* as geometric objects which transform according to the above transformation law:

Definition 3.24 (Covariant Vector) Any geometric object, that transforms in the same way as a gradient vector (3.29) in a manifold, i.e.

$$v_{a'} = \sum_{b=1}^n \frac{\partial x^b}{\partial x^{a'}} v_b \quad (3.30)$$

is called a *covariant vector* or *covector*.

Definition 3.25 (Cotangent Space) The set of all possible covariant vectors at a specific point P in a manifold is a vector space, denoted $T_P^*(\mathcal{M})$, called the *cotangent space* of \mathcal{M} at point P .

One-Forms

The denotation “covector” is old-fashioned as it makes reference to the transformation behavior of the components of the covector when changing the coordinate system. In modern terminology, this object is called a *one-form*, i.e. a linear form of one argument; in other words, a one-form is a real valued linear function of vectors: A one-form $\omega \in T_P^*(\mathcal{M}) \rightarrow \omega_a$ at point P associates with a vector $t \in T_P(\mathcal{M}) \rightarrow t^a$ at the same point P a real number, which is called $v(t)$. With this last notation, the idea of v being a function on vectors t is expressed. In physics literature, one-forms are often denoted with the symbol “ \sim ”, in analogy to the symbol “ \rightarrow ” which denotes a vector of $T_P(\mathcal{M})$, i.e. the real number that is associated by a one-form $\tilde{\omega}$ with a vector t is written as $\tilde{\omega}(t)$. This physicist’s notation is – in practical applications – superior to the rigorous mathematical notation, which states explicitly the domain and range of mappings and the spaces in which objects are defined; while the latter is common practice in mathematics, it makes many statements look cumbersome and is sometimes obstructive for practical calculations, where usually all properties of objects are assumed to be “just right” in their properties for the purpose of the calculation. In this book, both notations, one that is common in mathematics textbooks, and one that is more akin to a sloppy physicist’s style, are used and the reader should be aware from the context, which specific properties the considered objects are endowed with.

The linearity of the above function $\tilde{\omega}$ means

$$\tilde{\omega}(at + br) = a\tilde{\omega}(t) + b\tilde{\omega}(r), \quad (3.31)$$

where a and b are real numbers. Multiplication of a one-form $\tilde{\omega}$ by real numbers and the addition of one-forms is defined straightforwardly:

$$(a\tilde{\omega})(t) = a(\tilde{\omega}(t)), \quad (3.32a)$$

$\forall t$, and $\tilde{\omega} + \tilde{\sigma}$ is a one-form such that $\forall t$

$$(\tilde{\omega} + \tilde{\sigma})(t) = \tilde{\omega}(t) + \tilde{\sigma}(t). \quad (3.32b)$$

Thus, one-forms at a point P satisfy the axioms of a vector space, which is called the dual to T_P . The reason why this space is dual, is that vectors can also be regarded as real-valued, linear functions of one-forms, in the following manner: Given a vector t , its value on any one-form $\tilde{\omega}$ is defined as $\tilde{\omega}(t)$. This expression is linear, since its value on $a\tilde{\omega} + b\tilde{\sigma}$ is, according to (3.32):

$$(a\tilde{\omega} + b\tilde{\sigma})(t) = (a\tilde{\omega})(t) + (b\tilde{\omega})(t). \quad (3.33)$$

Hence, it is the linearity property of the underlying vector spaces which allows one to regard both, t and $\tilde{\omega}$ as a function that takes the other as argument and produces a real number: vectors and one-forms are *dual* to each other which is also expressed in notation:

$$\tilde{\omega}(t) \equiv t(\tilde{\omega}) \equiv \langle \tilde{\omega}, t \rangle. \quad (3.34)$$

The construction of the number $\tilde{\omega}(t)$ is often called *contraction* of $\tilde{\omega}$ with t .

Examples of one-forms are:

1. Column and row vectors in matrix algebra, which are combined (in the right order) to give a real number. For example, in \mathbf{R}^3 the row vector $(-4, 7, 5)$ may be thought of as a function which takes an arbitrary column vector into a real number:

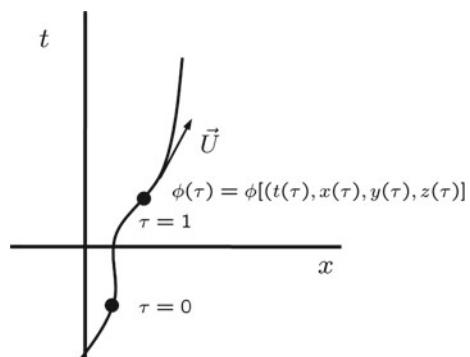
$$(-4, 7, 5) \begin{pmatrix} x \\ y \\ z \end{pmatrix} = -4x + 7y + 5z. \quad (3.35)$$

2. The derivative of a function. Consider a scalar field $\phi(x)$ defined at every event x in spacetime. The worldline of a particle assigns a value of ϕ at each event on the world line and this value changes from event to event, cf. Fig. 3.19. The four-velocity has components $\mathbf{U} = \left(\frac{dt}{d\tau}, \frac{dx}{d\tau}, \frac{dy}{d\tau}, \frac{dz}{d\tau} \right)$. The rate of change of the curve is given by

$$\frac{d\phi}{d\tau} = \frac{\partial\phi}{\partial t} \frac{\partial t}{\partial\tau} + \frac{\partial\phi}{\partial x} \frac{\partial x}{\partial\tau} + \frac{\partial\phi}{\partial y} \frac{\partial y}{\partial\tau} + \frac{\partial\phi}{\partial z} \frac{\partial z}{\partial\tau} \quad (3.36a)$$

$$= \frac{\partial\phi}{\partial\tau} U^t + \frac{\partial\phi}{\partial x} U^x + \frac{\partial\phi}{\partial y} U^y + \frac{\partial\phi}{\partial z} U^z. \quad (3.36b)$$

Fig. 3.19 A worldline parameterized with proper time τ , and the values of a scalar field ϕ along the worldline



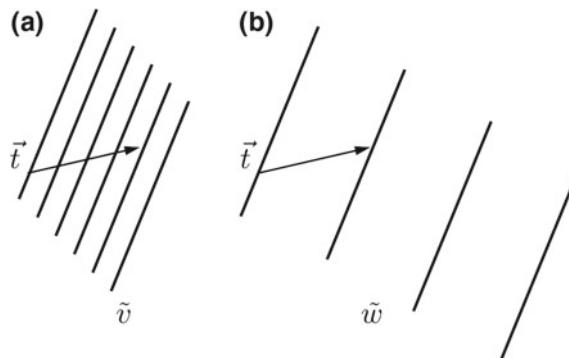


Fig. 3.20 **a** Picture of a one-form $\tilde{v} \in T_p(\mathfrak{M}^*)$ as a set of equidistant surfaces (contour lines) that are pierced by an arrow $t \in T_p(\mathfrak{M})$. The piercing produces a number $\tilde{v}(t)$; its value is the number of surfaces pierced. **b** The value of a smaller one-form \tilde{w} on the same vector t which produces a smaller number $\tilde{w}(t)$. The smaller the distance of surfaces, the larger the one form

As a result, the last equation produces from the vector \mathbf{U} the number $d\phi/d\tau$ which represents the rate of change of ϕ on a curve on which \mathbf{U} is the tangent. As $d\phi/d\tau$ is a function of \mathbf{U} , it is a one-form – denoted $d\phi$ with components $(\frac{\partial\phi}{\partial t}, \frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z})$.

For a visualization of vectors, one usually imagines an arrow. For a one-form, a similar picture can be employed by defining equidistant surfaces that are pierced by an arrow, cf. Fig. 3.20. Note that the one-form does not define a unique direction as it is not a vector. Rather, it defines a way of “slicing” the space. The *vector gradient*, also denoted with symbol $\vec{\nabla}$, or ∇ , is often introduced as a vector in elementary vector calculus. The arrow symbol on ∇ stresses the fact that ∇ is an element of T_P , the tangent space at point P of a manifold. Thus, it is identified with the one-form that pierces the maximum number of contour lines *per unit length*. In order to do this, one has to use a metric, which provides a definition of a scalar product between vectors and of distance, and thus allows to measure a unit length. Thus, if there is a metric defined, a vector can be associated with a one-form, i.e. a gradient that points to the maximum change of a scalar function. Such a vector is a normal vector, i.e. a vector orthogonal to all vectors tangent to a surface.

3. Dirac vectors of Hilbert space \mathfrak{H} which are used in quantum theory, are denoted as bras $\langle \Psi |$, which are vectors, and kets $|\Phi \rangle$ which are one-forms, whose contraction is $\langle \Psi | \Phi \rangle$, a complex number. The vector algebra of virtually all linear vector spaces employed in physics involves an inner product between the elements of these spaces, i.e. vectors. The reason for this is that in order to do physics one has to be able to do measurements. The inner (dot) product is a linear function of two vectors (and thus, can be interpreted as a tensor of rank two), which associates a number with them, see Box 3.6.

Box 3.6 Definition of an inner product

An **inner product** on a vector space X is a map $A \times A \rightarrow \mathbf{R}$, denoted $\langle \square | \square \rangle$, with the properties:

- Bilinearity: $a \mapsto \langle a | b \rangle$ is linear for every b , and $a \mapsto \langle a | b \rangle$ is linear for every b ;
- Symmetry: $\langle a | b \rangle = \langle b | a \rangle$;
- Nondegeneracy: If c satisfies $\langle c | a \rangle = 0$, $\forall a$, then $c = 0$.

This definition is a little more general than the one often provided in linear algebra, since it does not assume that $\langle a | a \rangle \geq 0$.

Remark 28 An engineer might probably wonder – after having dug through this thicket of definitions and abstractions – what it is finally good for. The answer is, that defining tensors based on manifolds and based on the different spaces in which co- and contravariant geometric objects are defined, provides a sound understanding of the use of coordinate systems in many applications of practical importance. However, in (non-curved) Euclidean space with Cartesian coordinates one normally does not distinguish between vectors (which are tangents of curves) and covectors (which are gradients of scalar functions). Thus, in the case of Cartesian coordinates, both indices are the same and no distinction between components in different vector spaces have to be made. This is because Cartesian coordinates have the special property that they are orthonormal, i.e. the product of Cartesian, normalized basis vectors $e_i e_j = \delta_{ij}$. But even in Euclidean flat space, if one uses non-orthogonal general curvilinear coordinates or skew coordinate systems, this vector/covector distinction becomes important.

3.3.6 Dual Spaces

Having defined both tangent space $T_p(\mathfrak{M})$ and cotangent space $T_p^*(\mathfrak{M})$ we can now ask for the relation of these two vector spaces to each other. These spaces are called *dual* to each other in the sense of vector space duality. For a rigorous mathematical definition of a dual space, see e.g. [216]. Specifically, any covariant vector $v \in T_p^*(\mathfrak{M})$ can be viewed as a linear mapping from $T_p(\mathfrak{M})$ to \mathbf{R} which was already used in the discussion of one-forms on p. 143. This is seen the easiest way when using coordinates. Let $v \in T_p^*(\mathfrak{M}) \rightarrow v_a$ and $t \in T_p(\mathfrak{M}) \rightarrow t^a$, then we define an admittedly less strict but more efficient physicist's notation as:

$$\tilde{v}(t) = v(t) = \sum_{a=1}^n v_a t^a \in \mathbf{R}. \quad (3.37)$$

Note, that the combination $v_a t^a$ is independent of the coordinate system. The object v_a transforms “covariantly” and t^a transforms “contravariantly”, and in the combination $v_a t^a$ the two transformation matrices, being inverses, cancel:

$$\sum_{b=1}^n \frac{\partial x^a}{\partial x^{b'}} \frac{\partial x^{b'}}{\partial x^c} = \sum_{b=1}^n \frac{\partial x^{a'}}{\partial x^b} \frac{\partial x^b}{\partial x^{c'}} = \delta_c^a . \quad (3.38)$$

Example 28 (Dual Spaces and linear forms) Consider two linear functionals α and β on a linear vector space A according to the definition in Box 3.7: Then the sum of the two forms

$$(\alpha + \beta)(x) := \alpha(x) + \beta(x) , \quad x \in A , \quad (3.41)$$

also is a linear form on A , as

$$(\alpha + \beta)(\lambda x + \mu y) = \alpha(\lambda x + \mu y) + \beta(\lambda x + \mu y) \quad (3.42a)$$

$$= \lambda \alpha(x) + \mu \alpha(y) + \lambda \beta(x) + \mu \beta(y) \quad (3.42b)$$

$$= \lambda [\alpha(x) + \beta(x)] + \mu [\alpha(y) + \beta(y)] \quad (3.42c)$$

$$= \lambda(\alpha + \beta)(x) + \mu(\alpha + \beta)(y) . \quad (3.42d)$$

Likewise, the product of a linear form with a number of the field \mathbf{K}

$$(\lambda \alpha)(x) := \lambda \alpha(x) , \quad x \in A , \quad (3.43)$$

is a linear form. Hence, $\phi : A^* \times A \rightarrow \mathbf{K}$,

Box 3.7 Dual Vector Spaces

Let A and B be two finite dimensional vector spaces over a common field \mathbf{K} . A function $\phi : A \times B \rightarrow \mathbf{K}$ is called a bilinear function or bilinear form, if ϕ is linear with respect to both arguments, i.e. if

$$\phi(\lambda x + \mu y, z) = \lambda \phi(x, z) + \mu \phi(y, z) , \quad (3.39)$$

for all vectors $z \in B$, respectively if

$$\phi(x, \lambda y + \mu z) = \lambda \phi(x, y) + \mu \phi(x, z) \quad (3.40)$$

for all vectors $x \in A$.

The function ϕ is called scalar product of two vector spaces A and B (i.e. ϕ is a tensor of rank 2) and is commonly denoted with the symbol $\langle \square, \square \rangle : A \times B \rightarrow \mathbf{K}$, i.e. $\langle x, y \rangle := \phi(x, y)$. The vector spaces A and B are said to be dual to each other with respect to the scalar product $\langle \square, \square \rangle$. The dual space of a vector space A is usually denoted as A^* ; it is the (linear) vector space of all maps $f : A \rightarrow \mathbf{K}$. Such maps are called linear functionals or linear forms and it can be shown that their composition with scalar functions and with addition and multiplication with a number obeys the axioms of a linear vector space, cf. Example 28.

$$\phi(\alpha, x) := \alpha(x) , \quad \alpha \in A^* , \quad x \in A , \quad (3.44)$$

is a bilinear form.

Consider the vector space A with respect to a fixed Cartesian basis $\mathbf{B} = \{e_1, e_2, \dots, e_N\}$. For $i = 1, 2, \dots, N$ the functions $\varepsilon^i : A \rightarrow \mathbf{K}$,

$$\varepsilon^i(\mathbf{a}) := a^i. \quad (3.45)$$

are considered. The map ε^i associates with each vector $\mathbf{a} \in A$ its i th component a^i with respect to the basis $\mathbf{B} = \{e_1, e_2, \dots, e_N\}$. Applying the linear form ε^i to the basis vector e_j the coordinates of which are all equal zero except for the j th coordinate, one obtains:

$$\varepsilon^i(e_j) = \langle \varepsilon^i, e_j \rangle = \delta_j^i = \delta_{ij} = \delta^{ij} = \delta_i^j = \begin{cases} 1 & \text{if } i \neq j, \\ 0 & \text{if } i = j. \end{cases} \quad (3.46)$$

Einstein's Summation Convention

We now introduce **Einstein's summation convention**. If nothing else is said, then over repeated pairs of covariant and contravariant indices is summed over which leads to the following succinct notation:

$$\sum_{a=1}^n v_a t^a = v_a t^a. \quad (3.47)$$

Equation (3.47) actually defines a Cartesian product (see Definition 3.4 on p. 117) of linear vector spaces $(v \in T_P^*) \times (\tau \in T_P) \rightarrow \mathbf{R}$. The dual of the dual space is the original space, i.e. $(T_P^*)^* = T_P$.

The product $v_a t^a$ has an important interpretation in terms of directional derivatives. If $v \in T_P^*$ arises from the gradient of a scalar field $\phi(P)$, and $\tau \in T_P$ arises as the tangent vector to a curve $h(\lambda)$ then

$$v(t) = v_a t^a = \frac{\partial \phi}{\partial x^a} \frac{dx^a}{d\lambda} = \frac{d\phi}{d\lambda}. \quad (3.48)$$

That is, $v(t)$ is simply the total derivative of the scalar field $\phi(P)$ along the curve $h(\lambda)$. Since $\phi(\lambda) : \mathbf{R} \rightarrow \mathbf{R}$ has a meaning independent of any coordinates that are introduced to the manifold, $v(t)$ is obviously coordinate invariant.

3.3.7 Tensors and Tensor Spaces

After having defined linear vector spaces, tangent vectors, one-forms, and duality, one can define general tensors as multilinear functions in linear vector spaces which are dual to each other. Just as an event as such in physical spacetime is independent of the specific frame of reference of the observer, tensors are defined as geometric objects (linear functions of several variables) which are independent of any specifically chosen coordinate system.

Definition 3.26 (Tensors) Consider the linear tangent space T_p of a manifold. The scalar-valued multilinear functions with variables all in either T_p or its dual T_p^* are called *tensors over T_p* and the linear vector spaces they form are the *tensor spaces*

over T_P . The number of variables from T_P^* and T_P are called the *type number* or *degrees* of the tensor, with the number of variables from T_P^* called the *contravariant degree*, the number of T_P the *covariant degree*.

Thus, for a multilinear function on $T_P^* \otimes T_P \otimes T_P \otimes T_P$ the tensor type is $(3, 1)$. The symbol \otimes denotes the Cartesian product of two structures, in this case the Cartesian product of a number of vector spaces. The space of multilinear functions is denoted as

$$T_1^2(P) = T_P \otimes T_P^* \otimes T_P^*. \quad (3.49)$$

with $P \in \mathfrak{M}$; Generalizing (3.49), a multilinear function $f : (T_P^*)^r \otimes (T_P)^s \rightarrow \mathbf{R}$ at a point P in \mathfrak{M} is called a tensor $T_s^r(P)$ of type (s, r) .

Let t_1, t_2, \dots, t_s be s covectors in point P and v^1, v^2, \dots, v^r be r tangent vectors at point P , then $T_s^r(t_1, t_2, \dots, t_s; v^1, v^2, \dots, v^r) \in \mathbf{R}$ and this function is defined to be linear in each one of its $(r + s)$ arguments.

The linear space $T = \lambda(T_P^*)^r \otimes (T_P)^s$ is a linear vector space of dimension $(r + s)$. T could equally well be considered an element of the dual of $(T_P^*)^r \otimes (T_P)^s$. In this case, one has

$$[(T_P^*)^r \otimes (T_P)^s]^* = [(T_P^*)^r]^* \otimes [(T_P)^s]^* \quad (3.50)$$

$$= [(T_P^*)^*]^r \otimes [(T_P)^*]^s \quad (3.51)$$

$$= (T_P)^r \otimes (T_P^*)^s. \quad (3.52)$$

That is:

$$T_s^r(P) \in (T_P)^r \otimes (T_P^*)^s = \underbrace{T_P \otimes \cdots \otimes T_P}_{r \text{ times}} \otimes \underbrace{T_P^* \otimes \cdots \otimes T_P^*}_{s \text{ times}}. \quad (3.53)$$

Tensor Components

We state a simple fact about tensor components as a theorem.

Theorem 5 *A tensor is determined by its values on a basis and its dual basis. These values are the components of the tensor with respect to the tensor products of basis and dual basis elements, which form bases of the tensor spaces.*

Proof Let $A \in T_s^r$ and $A_{j_1, \dots, j_s}^{i_1, i_2, \dots, i_r} = A(\varepsilon^{i_1}, \dots, \varepsilon^{i_r}; e_{j_1}, \dots, e_{j_s})$. Then for any $\tau^1, \dots, \tau^r \in T_P^*$ and $\sigma_1, \dots, \sigma_s \in T_P$, one has $\tau^p = a_i^p \varepsilon^i$, $\sigma_q = b_q^i e_i$, with transformation matrices a_i^p and b_i^q , $p = 1, \dots, r$; $q = 1, \dots, s$. It follows:

$$\begin{aligned} A(\tau^1, \dots, \tau^r, \sigma_1, \dots, \sigma_s) &= a_{i_1}^1 \dots a_{i_r}^r b_i^{j_1} \dots b_s^{j_s} A(\varepsilon^{i_1}, \dots, \varepsilon^{i_r}, e_{j_1}, \dots, e_{j_s}) \\ &= a_{i_1}^1 \dots a_{i_r}^r b_i^{j_1} \dots b_s^{j_s} A_{j_1, \dots, j_s}^{i_1, \dots, i_r} \\ &= A_{j_1, \dots, j_s}^{i_1, \dots, i_r} \langle e_{i_1}, \tau^1 \rangle \dots \langle e_{i_r}, \tau^r \rangle \langle \sigma_1, \varepsilon^{j_1} \rangle \dots \langle \sigma_s, \varepsilon^{j_s} \rangle \\ &= A_{j_1, \dots, j_s}^{i_1, \dots, i_r} \langle e_{i_1} \otimes \dots \otimes e_{i_r} \otimes \varepsilon^{j_1} \otimes \dots \otimes \varepsilon^{j_s} \rangle (\tau^1, \dots, \tau^r, \sigma_1, \dots, \sigma_s). \end{aligned} \quad (3.54)$$

Hence,

$$A = A_{j_1, \dots, j_s}^{i_1 \dots i_r} e_{i_1} \otimes \cdots \otimes e_{i_r} \otimes \varepsilon^{j_1} \otimes \cdots \otimes \varepsilon^{j_s}, \quad (3.55)$$

since they have the same values as functions on $(T_P^*)^r \otimes (T_P)^s$. ■

If t_1, t_2, \dots, t_r are r cotangent vectors at point P with components $t_1^a, t_2^a, \dots, t_r^a$, and v^1, v^2, \dots, v^s are s tangent vectors at P with components $v_b^1, v_b^2, \dots, v_b^s$, then the numbers $T^{a_1, a_2, \dots, a_r}_{ b_1, b_2, \dots, b_s}$ are called the *components* of the tensor $T_r^s(t_1, t_2, \dots, t_r; v^1, v^2, \dots, v^r)$ with respect to the specified chart, respectively coordinates.

Remark 29 (Transformation of a T_r^s Tensor) T_r^s transforms under a change of charts such that each one of the upper (contravariant) indices transforms like the components of a tangent vector (3.27) and each one of the lower (covariant) indices transforms like a gradient (3.29), the partial derivative of a scalar function). We provide some examples:

$$T_1^1 : X_{b'}^{a'} = X^a{}_b \frac{\partial x^{a'}}{\partial x^a} \frac{\partial x^b}{\partial x^{b'}}, \quad (3.56)$$

$$T_0^2 : X^{a'b'} = X^{ab} \frac{\partial x^{a'}}{\partial x^a} \frac{\partial x^{b'}}{\partial x^b}, \quad (3.57)$$

$$T_2^0 : X_{a'b'} = X_{ab} \frac{\partial x^a}{\partial x^{a'}} \frac{\partial x^b}{\partial x^{b'}}. \quad (3.58)$$

Tensor Fields

A tensor T is an object defined at *one* point P in a manifold \mathfrak{M} . A *tensor field* is an assignment of a tensor $T_r^s(P) \in [(T_P)^r \otimes (T_P^*)^s]$ to *each* point $P \in \mathfrak{M}$, that is, T is a multi-linear function which changes from point to point and the arguments of which are vectors and one-forms of T_P and T_P^* , respectively.

Tensor Algebra

In this section we consider the most important algebraic manipulations with tensors at the same point P of a manifold, i.e. with manipulation that do not compare tensors at different points in manifolds. We provide examples for the algebraic operations in specific coordinate systems, where the tensors have upper and lower components.

(A) Linear combinations. Tensors with the same number of upper and lower indices can be linearly combined to give a tensor with these indices, i.e.

$$C_a^b = \lambda A_a^b + \beta B_a^b, \quad (3.59)$$

where λ and β are scalars.

Example 29 (Show that C is a tensor) The tensor property of C is shown by transforming it to an different (primed) coordinate system (a different coordinate chart on $T_P(\mathfrak{M})$):

$$C_{a'}^{b'} = \lambda A_{a'}^{b'} + \beta B_{a'}^{b'} \quad (3.60)$$

$$= \lambda \frac{\partial x^{a'}}{\partial x^c} \frac{\partial x^d}{\partial x^{b'}} C_d^c + \beta \frac{\partial x^{a'}}{\partial x^c} \frac{\partial x^d}{\partial x^{b'}} B_d^c \quad (3.61)$$

$$= \frac{\partial x^{b'}}{\partial x^c} \frac{\partial x^d}{\partial x^{a'}} C_d^c . \quad (3.62)$$

■

(B) Direct Products. The product of two tensors yields a tensor whose upper and lower indices consist of all the upper and lower indices of the two original tensors, i.e. the resulting space is simply the product space of the original spaces. For example, if A_b^a and B^c are tensors, and

$$C_b^{ac} = A_b^a B^c , \quad (3.63)$$

then C_b^{ac} is a tensor, i.e.

$$C_b^{a'c'} = A_{b'}^{a'} + B^{c'} = \frac{\partial x^{a'}}{\partial x^d} \frac{\partial x^e}{\partial x^{b'}} A_e^d \frac{\partial x^{c'}}{\partial x^f} B_f^c = \frac{\partial x^{a'}}{\partial x^d} \frac{\partial x^e}{\partial x^{b'}} \frac{\partial x^{c'}}{\partial x^f} C_f^{de} . \quad (3.64)$$

(C) Contraction. Taking the inner product of a one-form $\tilde{A} \in T_p^\star$ and a vector $B \in T_p$ is called contraction, cf. the discussion of one-forms in Sect. 3.3.5. In a specific coordinate system, one-forms have covariant components, and vectors have contravariant components. Thus, a contraction means setting an upper and a lower index equal. For example

$$A^{ab} = B_c^{abc} . \quad (3.65)$$

Then A^{ab} is a tensor. The prove is left to the reader as Problem 3.6.

(D) Tensor Densities. An important example for a tensor density is the determinant g ($g < 0$) of the metric tensor

$$g = \text{Det}(g_{ab}) \quad (3.66)$$

The transformation rule for g_{ab} is

$$g_{a'b'} = \frac{\partial x^c}{\partial x^{a'}} g_{cd} \frac{\partial x^d}{\partial x^{b'}} . \quad (3.67)$$

Taking the determinant of (3.67) yields

$$g' = \left| \frac{\partial x}{\partial x'} \right|^2 g , \quad (3.68)$$

where $\partial x/\partial x'$ is the *Jacobian* of the transformation $x' \rightarrow x$, that is, it is the determinant D of the transformation matrix $\partial x^c/\partial x^{a'}$. A quantity such as g , which transforms like a scalar except for extra factors of the Jacobian, is called a *scalar density*, and, likewise, a quantity that transforms like a tensor except for extra

factors of Jacobian determinant is called a tensor density. The number of factors of the determinant $|\partial x'/\partial x|$ is called the *weight* of the density; Thus, g in (3.68) is a scalar density of weight -2 , since

$$\left| \frac{\partial x}{\partial x'} \right| = \left| \frac{\partial x'}{\partial x} \right|^{-1} \quad (3.69)$$

as can be seen from taking the determinant of the equation

$$\frac{\partial x^a}{\partial x'^b} \frac{\partial x'^b}{\partial x^c} = \delta_b^a. \quad (3.70)$$

Any tensor density of weight W can be expressed as an ordinary tensor times a factor $-g^{-W/2}$. For example, a tensor density A_b^a of rank W transforms according to

$$A_{b'}^{a'} = \left| \frac{\partial x}{\partial x'} \right|^W \frac{\partial x'^a}{\partial x^c} \frac{\partial x^d}{\partial x'^b} A_d^c, \quad (3.71)$$

and employing (3.68) yields:

$$g'^{W/2} A_{b'}^{a'} = \left| \frac{\partial x}{\partial x'} \right|^W \frac{\partial x'^a}{\partial x^c} \frac{\partial x^d}{\partial x'^b} A_d^c. \quad (3.72)$$

The importance of tensor densities comes from integration theory, which states that a volume element d^4x under a coordinate transformation $x \rightarrow x'$ becomes

$$d^4x' = \left| \frac{\partial x}{\partial x'} \right| d^4x. \quad (3.73)$$

Thus, in particular, $\sqrt{-g} d^4x$ is an invariant volume element.

There is one tensor density whose components are the same in all coordinates systems. This is the so-called *Levi-Civita* tensor density ε^{abcd} . In a general coordinate system, this tensor density is defined as:

$$\varepsilon^{abcd} = \begin{cases} +1 & : \text{even permutations of } a, b, c, d, \\ -1 & : \text{odd permutations of } a, b, c, d, \\ 0 & : \text{some indices equal.} \end{cases} \quad (3.74)$$

Exercise 5 (*Totally antisymmetric tensor ε^{abcd}*) Proof that $\varepsilon^{abcd} = -\varepsilon_{abcd}$ in orthonormal coordinate systems.

Solution 4 The metric components in an orthonormal coordinate system are given by:

$$g^{00} = 1, \quad g^{ii} = -1, \quad g^{ik} = 0, \quad \text{for } i \neq k. \quad (3.75)$$

It is

$$\varepsilon^{abcd} = g^{ae} g^{bf} g^{cg} g^{dh} \varepsilon_{efgh}. \quad (3.76)$$

For each non-vanishing element, the expression $g^{ae} g^{bf} g^{cg} g^{dh}$ contributes a negative sign, thus it follows

$$\varepsilon^{abcd} = -\varepsilon_{abcd}. \quad (3.77)$$

■

3.3.8 Affine Connections and Covariant Derivative

We consider a contravariant tensor field $A(x^a)$ with components $A^a(x^b)$ which under a change of chart ($x^a \rightarrow x^{a'}$) are

$$A^{a'} = \frac{\partial x^{a'}}{\partial x^b} A^b. \quad (3.78)$$

Taking the partial derivatives $\partial/\partial x^b = \partial_b$ of this tensor field of rank 1 yields:

$$\partial_{c'} A^{a'} = \partial_{c'} \left(\frac{\partial x^{a'}}{\partial x^b} A^b \right) \quad (3.79)$$

$$= \partial_{c'} \left(\frac{\partial x^{a'}}{\partial x^b} \right) A^b + \left(\frac{\partial x^{a'}}{\partial x^b} \right) \partial_{c'} A^b \quad (3.80)$$

$$= \left[\frac{\partial}{\partial x^{c'}} \left(\frac{\partial x^{a'}}{\partial x^b} \right) \right] A^b + \left(\frac{\partial x^{a'}}{\partial x^b} \right) \left(\frac{\partial x^d}{\partial x^{c'}} \right) \partial_d A^b \quad (3.81)$$

$$= \left[\frac{\partial x^d}{\partial x^{c'}} \left(\frac{\partial^2 x^{a'}}{\partial x^d \partial x^b} \right) \right] A^b + \left(\frac{\partial x^{a'}}{\partial x^b} \right) \left(\frac{\partial x^d}{\partial x^{c'}} \right) \partial_d A^b \quad (3.82)$$

If only the last term in (3.82) were present then this would be the usual transformation law of a T_1^1 tensor. The presence of the first term however destroys the tensorial properties of the partial derivative $\partial_c A^a$. This does not come as a surprise as the partial derivative involves comparing a quantity evaluated at two neighboring points, say P and Q , dividing by some parameter expressing the separation of P and Q and then taking the limit as this parameter goes to zero. In the case of a contravariant vector field the derivative involves calculating

$$\partial_c A^a = \lim_{\delta x \rightarrow 0} \frac{A^a(x + \delta x) - A^a(x)}{\delta x}. \quad (3.83)$$

Equation (3.83) means subtracting two different contravariant vectors A in two different vector spaces, one at Point $P(x)$, $(A(x) \in T_P(\mathfrak{M}))$, and one at point $Q(x + \delta x)$, $(A(x + \delta x) \in T_Q(\mathfrak{M}))$. Thus, one has to find a way to build a covariant derivative of a tensor i.e., one which is independent of the coordinate system.

According to (3.82), the partial derivative of a tensor is not a tensor due to additional terms that appear in the expression for the derivative. Hence, in order to be able to write down general covariant equations including tensor derivatives, one needs to find a way of defining the derivative of a tensor, such that it becomes independent of the coordinate system. There are at least three routes to tensor differentiation which can be found in textbooks on differential geometry:

- **The Lie derivative.** The Lie derivative defines a rule which drags a vector along a curve in the manifold; thus, differentiation in this case is defined along a curve and restricts the directions in which the differentiation is done.

- **The exterior derivative.** The exterior derivative is defined only for a special class of tensors, namely totally antisymmetric tensors, which play a fundamental role in integration theory and in *exterior*, or *Grassmann algebra*, which is a prerequisite for the study of *exterior calculus* or *calculus of differential forms*. Differential forms offer another approach to coordinate-free multivariable calculus which can be thought of as generalization of ordinary vector calculus, including line, surface, and volume integrals. The interested reader is referred to specialized literature listed in Sect. 3.8. Thus, with the exterior derivative, one restricts the *kind* of tensors that is differentiated.
- **The covariant derivative.** The covariant derivative introduces an additional structure on a topological manifold – the affine connection – which connects different points on a curve on a manifold and thereby allows to transport vectors from point to point and to compare them. This is the approach which is introduced in this section.

Parallel Transport

Figure 3.21 shows how two vectors in Euclidean space, which are separated by an infinitesimal distance, would be compared. Here, there is *one global* coordinate chart covering the whole manifold space \mathbf{R}^n . One would simply parallel transport this vector from $P(x)$ to $Q(x + \delta x)$. In a general manifold however, there is no global coordinate patch, thus one has to *define*, what it is supposed to mean, if an object does not change its components when parallel-transported along a curve from the tangent space at P to the tangent space at Q . This can be visualized on the curved (non-Euclidean) surface of a sphere, cf. Fig. 3.22. We denote δA^a as the difference between the vector in Q , namely $A^a + \delta A^a$ and the vector in P , namely the original A^a . By Taylor's theorem one has to first order

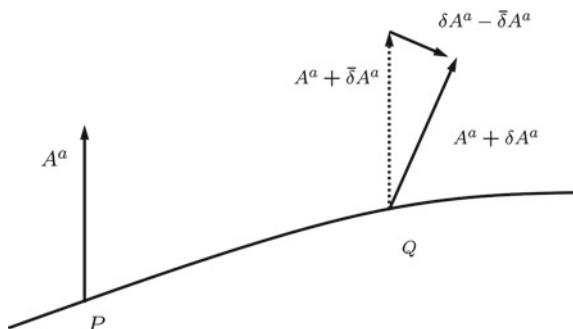
$$A^a(x + \delta x) = A^a(x) + \delta x^b \partial_b A^a. \quad (3.84)$$

Denoting the second term in (3.84) by $\delta A^a(x)$, cf. Fig. 3.21, one obtains

$$\delta A^a(x) = \delta x^b \partial_b A^a = A^a(x + \delta x) - A^a(x), \quad (3.85)$$

a quantity which cannot be tensorial as it involves subtracting tensors evaluated at two different points. We are going to define a tensorial derivative by introducing a

Fig. 3.21 An affine connection Γ_{bc}^a on a manifold allows for parallel-transporting vectors along curves in the manifold. A manifold endowed with functions Γ_{bc}^a is called an *affine manifold*



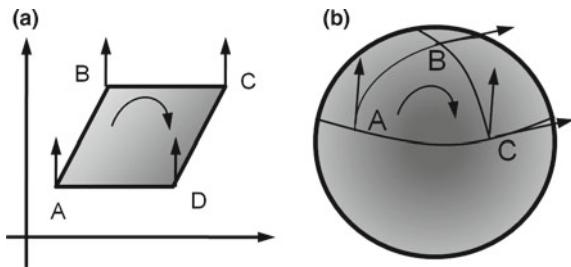


Fig. 3.22 Comparison of a parallel transport in Euclidean flat space and a non-Euclidean curved surface of the sphere. The vectors are transported along the curves in a way, that its angle with the tangent vector $\partial r / \partial x^i$ to the coordinate lines stays the same. In the Euclidean case with rectilinear coordinates ($\partial r / \partial x^i = \text{constant}$), the parallel transported vector coincides with the original one in point A. On the sphere, however, the result of a parallel transport (A-C and A-B-C) depends on the path chosen, since the tangent vector $\partial r / \partial x^i$ changes from point to point along the coordinate lines

vector at Q which in some generalized sense is “parallel” to A^a at P . Since $x^a + \delta x^a$ is close to x^a , the parallel vector only differs from $A^a(x)$ by a small amount, which is denoted as $\bar{\delta}A^a(x)$, cf. Fig. 3.21. We shall now construct the displacement δA^a in such a way as to make the difference vector

$$A^a(x) + \delta A^a(x) - [A^a(x) + \bar{\delta}A^a(x)] = \delta A^a(x) - \bar{\delta}A^a(x) \quad (3.86)$$

tensorial. It is natural to require that $\bar{\delta}A^a$ should vanish when A^a or δA^a does. Thus, one assumes a *linear* dependence in both A^a and δA^a :

$$\bar{\delta}A^a(x) = -\Gamma^a_{bc}(x)A^b\delta x^c. \quad (3.87)$$

In a space with d dimensions, the components of Γ are a set of 3^d components and due to the symmetry in the two lower indices, there are $\frac{d^2+d}{2}d$ independent coefficients, and the minus sign is introduced to agree with convention. The *covariant derivative* which is denoted by

$$\nabla_c A^a \text{ or } A^a{}_{;c} \text{ or } X^a_{|c}, \quad (3.88)$$

is defined as the following limiting process:

$$A^a_{|c} = \lim_{\delta x^c \rightarrow 0} \frac{[A^a(x + \delta x) - [A^a(x) + \delta' A^a(x)]]}{\delta x^c}. \quad (3.89)$$

Equation (3.89) expresses the difference between the vector A at point Q and the parallel transported vector at Q which is – by definition – parallel to the vector at P , in the limit as these differences approach zero. Inserting (3.84) and (3.87) into (3.89) one obtains the definition of the covariant derivative of a vector:

$$A^a_{|c} = \partial_c A^a + \Gamma^a_{bc} A^b. \quad (3.90)$$

The quantity Γ^a_{bc} is called *affine connection* or simply *connection*. It is not a tensor because its transformation law is linear inhomogeneous. A manifold with a continuous connection on it is called an *affine manifold*. The connection is used in the definition of a covariant derivative, i.e. a derivative which itself is a tensor, in

order to compensate the additional term in (3.82). A straightforward calculation (see Problem 3.7) shows that Γ_{bc}^a transforms according to:

$$\Gamma_{b'c'}^{a'} = \frac{\partial x^{a'}}{\partial x^d} \frac{\partial x^e}{\partial x^{b'}} \frac{\partial x^f}{\partial x^{c'}} \Gamma_{ef}^d - \frac{\partial x^d}{\partial x^{b'}} \frac{\partial x^e}{\partial x^{c'}} \frac{\partial^2 x^{a'}}{\partial x^d \partial x^e}, \quad (3.91)$$

or, alternatively

$$\Gamma_{b'c'}^{a'} = \frac{\partial x^{a'}}{\partial x^d} \frac{\partial x^e}{\partial x^{b'}} \frac{\partial x^f}{\partial x^{c'}} \Gamma_{ef}^d + \frac{\partial x^{a'}}{\partial x^d} \frac{\partial^2 x^d}{\partial x^{b'} \partial x^{c'}}, \quad (3.92)$$

Without the second terms on the right-hand side of (3.91) and (3.92), this would be the transformation law for a (2, 1) tensor.

Likewise, the covariant derivative of a covector is defined to be the same as its partial derivative, i.e.

$$A_{|c} = \partial_c A_a. \quad (3.93)$$

If one additionally demands that the covariant derivative obeys the Leibniz rule, one finds:

$$A_{a|c} = \partial_c - \Gamma_{ac}^b A_b. \quad (3.94)$$

For the covectors, the Γ -term enters with a minus sign, again compensating the extra term when building the partial derivative in (3.93).

With (3.90) and (3.94) the covariant derivative of vectors and covectors is defined and can be generalized to all n covariant and m contravariant indices of general (n, m) tensors.

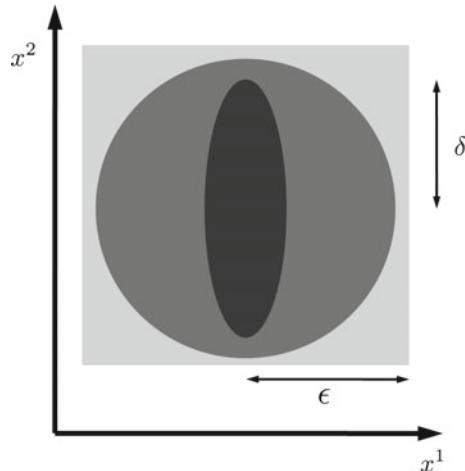
3.4 Metric Spaces and Metric Connection

In Sect. 3.2.2 a metric – or distance function – was introduced, which is a function $f : X \times X \rightarrow \mathbf{R}$ satisfying the axioms given in (3.1) on p. 119. Any symmetric covariant tensor field of rank two, i.e. any (2, 0) tensor, say $g_{ab}(x)$ defines a metric; it can be interpreted a function that takes two vectors as arguments to produce a real number, the distance d .

Example 30 (Standard Topology on \mathbf{R}^n) Consider the space $X = \mathbf{R}^n$. A point in \mathbf{R}^n is denoted by an n -tuple of real numbers (x^1, x^2, \dots, x^n) , the coordinates. After introducing a metric $d(x, y)$ in \mathbf{R}^n we can calculate the distance between any two points $x = (x^i)$ and $y = (y^i)$ with $i = 1, 2, \dots, n$. We obtain the *standard topology* on \mathbf{R}^n when the metric is defined by absolute values of differences, i.e. for $p \geq 1$, $x, y \in \mathbf{R}^n$,

$$d_p(x, y) = \left[\sum_{i=1}^n |x^i - y^i|^p \right]^{1/p}. \quad (3.95)$$

Fig. 3.23 Examples for different metrics which induce the same topology. An open circular neighborhood of radius ϵ entirely contains an ellipsoidal neighborhood of radius δ if $\delta < \epsilon$. For $\delta > 2\epsilon$, the inclusion is inverted. The open set defined by a Manhattan metric (3.96) is indicated by the quadratic domain



Let $\tilde{d}(x, y) = [4(x^1 - y^1)^2 + (x^2 - y^2)^2]^{1/2}$ and $d_2(x)$ the Euclidean distance function. Then \tilde{d} and d_2 induce the same topology as is illustrated in Fig. 3.23. \tilde{d} defines an open neighborhood in \mathbf{R}^2 which is the interior of the disk bounded by an ellipse. Also the use of a Manhattan metric⁴

$$d'(x, y) = \text{maximum}(|x^1 - y^1|, |x^2 - y^2|, \dots, |x^n - y^n|) \quad (3.96)$$

does not change the topology. All metrics define open neighborhoods of points and the key point is that any set which is open according to the metric d' is also open according to \tilde{d} or d_2 . Thus, all these metrics induce the same topology. Their neighborhoods are illustrated in Fig. 3.23. The topology is not very dependent upon the specific form of the metric d – it is a more primitive concept, cf. Tables 3.1 and 3.2.

The infinitesimal distance, or *interval* between two points in $T_p(\mathfrak{M})$ in the theory of relativity, is defined as

$$(ds)^2 = ds^2 = g_{ab}(x)dx^a dx^b. \quad (3.97)$$

Note that the components of g_{ab} depend on the specific position x in tangent space $T_p(\mathfrak{M})$. Equation (3.97) is known as *line element*, and g_{ab} is also called the *metric form* or *first fundamental form*. If the metric is indefinite (as in relativity theory), then there exist vectors which are orthogonal to themselves, called *null vectors*, i.e.

$$g_{ab}dx^a dx^b = 0. \quad (3.98)$$

If the metric is non-singular, i.e. if $g \neq 0$, the inverse of g_{ab} is given by g^{ab} and

$$g_{ab}g^{bc} = \delta_a^c = g_a^c. \quad (3.99)$$

⁴The Manhattan metric plays an important role in automatic data classification by means of clustering algorithms, which are used e.g. in genome research to analyze microarray data [8, 217].

Since δ_a^c is a scalar, it follows from this definition that g^{ab} is a contravariant tensor of rank 2. The metric tensor can be used to lower and rise indices, i.e. to switch between the dual spaces at a point P of a manifold.

The immediate importance of the metric in classical physics lies in the expression of the squared velocity v^2 of a classical point particle

$$v^2 = \left(\frac{ds}{dt} \right) = g_{ab} \frac{dx^a}{dt} \frac{dx^b}{dt}, \quad (3.100)$$

which is needed for the construction of the kinetic energy part of the classical Lagrange-function, cf. Box 2.4 on p. 71:

$$L = \frac{m}{2} v^2 = \frac{m}{2} g_{ab} \frac{dx^a}{dt} \frac{dx^b}{dt} = \frac{m}{2} g_{ab} \dot{x}^a \dot{x}^b. \quad (3.101)$$

Solving the Euler–Lagrange Equations, (2.44)

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}^c} - \frac{\partial L}{\partial x^c} = 0 = \frac{d}{dt} m g_{ab} \dot{x}^a - \frac{\partial L}{\partial x^c} \quad (3.102)$$

yields

$$g_{ab} \ddot{x}^a + (\partial_b g_{ab}) \dot{x}^a \dot{x}^b - \frac{1}{2} (\partial_c g_{ab}) \dot{x}^a \dot{x}^b = 0. \quad (3.103)$$

It follows

$$(\partial_b g_{ab}) \dot{x}^a \dot{x}^b = \frac{1}{2} (\partial_b g_{ac} + \partial_a g_{bc}) \dot{x}^a \dot{x}^b. \quad (3.104)$$

Multiplying (3.103) with g^{cd} , summing over c and using (3.99), one obtains:

$$\ddot{x}^d = \Gamma_{ab}^d \dot{x}^a \dot{x}^b = 0, \quad (3.105)$$

with the abbreviation

$$\Gamma_{ab}^d = \frac{1}{2} g^{dc} (\partial_b g_{ac} + \partial_a g_{bc} - \partial_c g_{ab}). \quad (3.106)$$

The equations in (3.105) are the equations of motion of a force-free point particle and the symbols in (3.106) are called *Christoffel symbols* or *metric connection*. The Lagrange-Equations are the solution to a variation problem, i.e. they minimize the integral of least action, which, interpreted in a manifold, gives the shortest path of a particle, i.e. the shortest connection between two points in different tangent spaces $T_P(\mathfrak{M})$, connected by a curve in \mathfrak{M} . As the result is neither dependent on a specific choice of coordinates, nor on the dimension of \mathfrak{M} , (3.106) is the *geodesic equation* or *metric geodesic*; Hence, physically speaking, force-free particles move along shortest paths in spacetime and this result is valid for any d -dimensional space.

Example 31 (Christoffel Symbols of Spherical Coordinates) Using the Euler–Lagrange Equations (2.44) for the Lagrange-function in spherical coordinates

$$L = \frac{m}{2} (\dot{r}^2 + r^2 \dot{\theta}^2 + r^2 \sin^2 \theta \dot{\phi}^2) \quad (3.107)$$

yields:

$$\ddot{r} - r\dot{\theta}^2 - r \sin^2 \theta \dot{\phi}^2 = 0 , \quad (3.108a)$$

$$\ddot{\theta} + \frac{2}{r}\dot{r}\dot{\theta} - \sin \theta \cos \theta \dot{\phi}^2 = 0 , \quad (3.108b)$$

$$\ddot{\phi} + \frac{2}{r}\dot{r}\dot{\phi} + 2 \cot \theta \dot{\phi}\dot{\theta} = 0 . \quad (3.108c)$$

It is left to the reader as Problem 3.8 to determine the nine different Christoffel symbols that are not zero.

3.5 Riemannian Manifolds

The term “Riemannian Manifolds” or “Riemannian Geometry” dates back to 10th June 1854 when Bernhard Riemann (1826–1866) held his inaugural speech “On the hypotheses which underlie the foundations of geometry” in Göttingen. In this lecture Riemann introduced the concept of a “n-fold extended manifold”, in which each point is denoted by n numbers but in which no measure for “distance” has been defined which would enable the comparison of the distance of point pairs at different locations. Such a measure of distance is introduced by a distance function which is called *metric*.

Definition 3.27 (Riemannian Metric) In modern language, a *Riemannian metric* on a manifold \mathfrak{M} is a symmetric (2,0) tensor, which has the additional property of being positive definite. Hence, for each point $P \in \mathfrak{M}$ there is a bilinear mapping $g_P : T_P(\mathfrak{M}) \times T_P(\mathfrak{M}) \rightarrow \mathbf{R}$ such that $g_P(v, w) = g_P(w, v) = \langle v, w \rangle \forall v, w \in T_P(\mathfrak{M})$ and $g_P(v, v) > 0 \forall v \neq 0$. In other words, g_P is a Euclidean scalar product on $T_P(\mathfrak{M})$.

In physics, the metric is not positive definite, due to the opposite sign of time and space coordinates. For such a metric one demands the weaker condition of *non-degeneracy* and one speaks of a *Semi-Riemannian* or Pseudo-Riemannian Metric.

Definition 3.28 (Riemannian Manifold) A manifold which is endowed with a (pseudo-) Riemannian metric is called (pseudo-) Riemannian manifold.

The main purpose of a metric function in a manifold is to allow for notions of distance, angle, parallel lines or straight lines, which – in the Euclidean case – are trivial. In Sect. 3.3.8 we have seen, that the idea of parallel lines in a topological differentiable manifold seems meaningless. In fact, the status of the parallel axiom, also called *Euclid's Fifth Postulate* in Euclidean geometry has been disputed for centuries and many mathematicians tried to “prove” it by means of other axioms. The discovery of non-euclidean geometries, i.e. geometries in which the *Fifth Postulate* is not valid, ended this dispute. Hence, in general geometries, there is no “natural

extension” of *global parallelism*, as known in Euclidean geometry. Gauss was the first, who developed the idea of *inner* geometric properties of a geometry, e.g. a surface in space, which are properties that can be measured on the surface alone, without making reference to the embedding space of the surface; he formulated the idea of curvature of surfaces in his *Theorema Egregium* which states that the curvature of a surface only depends on the first fundamental form, i.e. the metric tensor. The generalization of this idea to general curved spaces (i.e. manifolds) leads to the concept of Riemannian curvature on manifolds, which we will shortly discuss in the following.

3.5.1 Riemannian Curvature

We have seen that force-free particles follow a geodesic, a straightest line in a manifold according to (3.105). We consider two different tangent vectors of $T_P(\mathfrak{M})$ on two geodesic curves parameterized with u and v , and distance $w^i \, dv$, i.e.

$$t^i = \frac{\partial x^i}{\partial u}, \quad (3.109)$$

and

$$w^i = \frac{\partial x^i}{\partial v}. \quad (3.110)$$

The relative acceleration of these two particles following a geodesics, their *geodesic deviation* $\frac{D^2 w^i}{Du^2}$, can be taken as a measure of curvature. The covariant differential operator $\frac{D}{Du}$ is defined as:

$$\frac{D}{Du} := \frac{\partial}{\partial u} + \Gamma_{kl}^i \frac{\partial x^k}{\partial u}. \quad (3.111)$$

The straightforward calculation of the relative acceleration yields:

$$\begin{aligned} \frac{D^2 w^i}{Du^2} &= \frac{\partial}{\partial u} \left(\frac{Dw^i}{Du} \right) + \Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{Dw^l}{Du} \\ &= \frac{\partial}{\partial u} \left(\frac{Dw^i}{Du} \right) + \Gamma_{kl}^i \frac{\partial x^k}{\partial u} \left(\frac{\partial w^l}{\partial u} + \Gamma_{jm}^l \frac{\partial x^j}{\partial u} w^m \right) + \frac{\partial}{\partial u} \left(\Gamma_{kl}^i \frac{\partial x^k}{\partial u} w^l \right) \\ &= \frac{\partial^2 x^i}{\partial u \partial v} + \frac{\partial x^k}{\partial u} w^l \Gamma_{kl,m}^i \frac{\partial x^m}{\partial u} + \Gamma_{kl}^i \frac{\partial^2 x^k}{\partial u^2} w^l + \Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial^2 x^l}{\partial u \partial v} + \\ &\quad + \Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial^2 x^l}{\partial v \partial u} + \Gamma_{jl}^i \Gamma_{km}^l \frac{\partial x^j}{\partial u} \frac{\partial x^k}{\partial u} w^m. \end{aligned} \quad (3.112)$$

Using the geodesic equation of motion of force-free particles

$$\frac{\partial^2 x^i}{\partial u^2} = -\Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial x^l}{\partial u} \quad (3.113)$$

it follows:

$$\begin{aligned}
\frac{D^2 w^i}{Du^2} &= \frac{\partial}{\partial v} \left[-\Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial x^l}{\partial u} \right] + \Gamma_{kl,m}^i + \Gamma_{kl,m}^i \frac{\partial x^m}{\partial u} \frac{\partial x^k}{\partial u} w^l + \\
&\quad + \Gamma_{kl}^i \left[\Gamma_{jm}^k \frac{\partial x^j}{\partial u} \frac{\partial x^m}{\partial u} \right] w^l + \Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial^2 x^l}{\partial u \partial v} + \\
&\quad + \Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial^2 x^l}{\partial u \partial v} + \Gamma_{jl}^i \Gamma_{km}^l \frac{\partial x^j}{\partial u} \frac{\partial x^k}{\partial u} w^m \\
&= -\Gamma_{kl,j}^i \underbrace{\frac{\partial x^j}{\partial v} \frac{\partial x^k}{\partial u} \frac{\partial x^l}{\partial u}}_{=w^j} - \underbrace{\Gamma_{kl}^i \frac{\partial^2 x^k}{\partial u \partial v} \frac{\partial x^l}{\partial u}}_{-2A} - \underbrace{\Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial^2 x^l}{\partial u \partial v}}_{+2A} + \Gamma_{kl,m}^i \frac{\partial x^m}{\partial u} \frac{\partial x^k}{\partial u} w^l - \\
&\quad - \Gamma_{kl}^i \Gamma_{jm}^k \frac{\partial x^j}{\partial u} \frac{\partial x^m}{\partial u} w^l + \underbrace{2\Gamma_{kl}^i \frac{\partial x^k}{\partial u} \frac{\partial^2 x^l}{\partial u \partial v}}_{+2A} + \Gamma_{jl}^i \Gamma_{km}^l \frac{\partial x^j}{\partial u} \frac{\partial x^k}{\partial u} w^m. \tag{3.114}
\end{aligned}$$

Renaming appropriate dummy indices finally leads to the expression:

$$\frac{D^2 w^i}{Du^2} = \left[\Gamma_{km,l}^i - \Gamma_{kl,m}^i + \Gamma_{km}^n \Gamma_{nl}^i - \Gamma_{kl}^n \Gamma_{nm}^i \right] \frac{\partial x^k}{\partial u} \frac{\partial x^l}{\partial u} w^m = R_{klm}^i t^k t^l w^m, \tag{3.115}$$

with the following definition of the **Riemannian curvature tensor**:

$$R_{klm}^i = \Gamma_{km,l}^i - \Gamma_{kl,m}^i + \Gamma_{km}^n \Gamma_{nl}^i - \Gamma_{kl}^n \Gamma_{nm}^i. \tag{3.116}$$

Hence, if all components of the Riemannian curvature tensor are zero, the geodesic deviation (3.115) vanishes. This corresponds to a flat Euclidean space, where a global Cartesian coordinate system can be used to label events. Decisive for the curvature are thus the first and second derivatives of the first fundamental form g_{ab} . The following linear combinations of (3.116) are also tensors: The Ricci-tensor

$$R_{im} = R_{ilm}^l = g^{kl} R_{kil}^l, \tag{3.117}$$

and the curvature scalar

$$R = R_i^i = g^{ik} R_{ik}. \tag{3.118}$$

The curvature tensor has the following symmetries

$$R_{ijkl} = R_{klji}, \tag{3.119a}$$

$$R_{ijkl} = -R_{jikl} = -R_{ijlk} = R_{jilk}, \tag{3.119b}$$

$$R_{ijkl} + R_{iljk} + R_{ijlk} = 0, \tag{3.119c}$$

and the number of independent components c_d in d dimensions is

$$cd = \lambda \frac{d^2(d^2 - 1)}{12}, \tag{3.120}$$

i.e.

$$c_1 = 0, \quad c_2 = 1, \quad c_3 = 6, \quad c_4 = 20. \tag{3.121}$$

Example 32 (R_{ijkl} in 2d-space) In two dimensions the curvature tensor is determined by one quantity and can be written as:

$$R_{ijkl} = R_{1212} = -R_{2112} = -R_{1221} = R_{2121} = (g_{il}g_{jk} - g_{ik}g_{jl}) \frac{R_{1212}}{g}, \quad (3.122)$$

with $g = \text{Det}(g_{ik}) = (g_{11}g_{22} - g_{21}g_{12})$. It follows

$$R_{jl} = g^{ik} R_{ijkl} = g_{ip} \frac{R_{1212}}{g}, \quad (3.123)$$

and

$$R = \frac{2 R_{1212}}{g}. \quad (3.124)$$

Inserting the metric of a S^2 -sphere

$$ds^2 = a^2 d\theta^2 + a^2 \sin^2 \theta d\phi^2 \quad (3.125)$$

one yields

$$R = \frac{2}{a^2}, \quad (3.126)$$

the curvature of a sphere.

Remark 30 Some books on relativity theory make heavy use of a coordinate-free formulation of definitions and notations, e.g. Wald [203]. In such an approach, the coordinates are hidden and one can better realize the general structure of the developed theory, just like in Hamiltonian formulation of mechanics versus the original Newtonian formulation. However, as we have seen in the above example, for *explicit* calculations and for solving specific problems, what one almost always has to do first, is to chose an appropriate coordinate system and represent the geometric objects in a detailed low-level work by coordinates.

3.6 The Problem of Inertia and Motion: Coordinate Systems in Physics

In physics, events in space–time are denoted with three spacial coordinates and one time coordinate. The mathematical starting point for a description of events in spacetime is the notion of a linear vector space, which is a number of objects, called vectors, endowed with an algebraic structure. This “linear structure” manifests itself in the physical world as the principle of superposition, e.g. if one thinks of force vectors, this principle means that there is a resulting sum of force vectors that are acting at the same point in space. Usually time is just a parameter which is necessary to allow for a description of the dynamic evolution of a system, i.e. the motion of particles or atoms. Hence, coordinates are numbers that are used for a unique description of the considered objects in space. Space itself in Newtonian mechanics is ascribed an independent existence and objects are mathematically described with

respect to coordinate systems fixed in space. This is needed for a purely *kinematic* description of motion, i.e. without reference to the origin of the movement of bodies (which are forces). In physics, a dynamic description of motion makes use of the concept of forces acting on mass points. However, the forces are proportional to acceleration and thus appear as *absolute* quantities. This is the reason, why the Newtonian laws of motion

$$m_i \frac{d^2 \mathbf{r}^i}{dt^2} = \mathbf{F}_i(|\mathbf{r}^i - \mathbf{r}^j|) \quad (i, j = 1, 2, \dots, N) \quad (3.127)$$

only have their simple form in *unaccelerated* reference systems, which are called inertial systems (IS). In IS, the motion of particles are straight lines (geodesics). In *accelerated* systems, so-called *inertial forces* appear, which are of the well-known following form:

$$m_i \frac{d^2 \mathbf{r}^i}{dt^2} = \mathbf{F}_i(|\mathbf{r}^i - \mathbf{r}^j|) - m_i \frac{d\boldsymbol{\omega}_i}{dt} \times \mathbf{r}_i - 2m_i \boldsymbol{\omega}_i \times \mathbf{v}_i - m_i \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \mathbf{r}) . \quad (3.128)$$

Thus, an observer in a linearly accelerated and rotated system⁵ with respect to an inertial system measures a curvilinear path for a system and attributes this observation to the acting of several forces. As a result, just by changing the *kinematic* description of a system (by changing from an IS to a non-IS) additional forces appear in the equations with no real dynamic cause. Thus, with respect to the discussion of fundamental forces in Chap. 2 on p. 57, the question arises, to which fundamental interaction the inertial forces pertain. Standard textbooks in mechanics give the standard answer that inertial forces pertain to *no* dynamic fundamental interaction at all, but rather are a *kinematic* consequence of motion relative to absolute space, respectively to the system of fixed stars.

3.6.1 The Special and General Principle of Relativity

In Newtonian mechanics, the notion of an *inertial system* is used for defining the *Galilean Principle of Relativity*: All mechanical processes proceed in the same way in all inertial systems. Inertial systems are connected with each other via the *Galilean group*:

$$\mathbf{r}' = \mathbf{R}\mathbf{r} + \mathbf{v}t + \mathbf{x}_0, \quad (3.129a)$$

$$t' = t + t_0, \quad (3.129b)$$

where $\mathbf{R}\mathbf{R}^T = 1$. The matrix \mathbf{R} is an orthogonal rotation matrix and \mathbf{R}^T is its transposed matrix. The translations in space ($\mathbf{r}' = \mathbf{r} + \mathbf{r}_0$) and time ($t' = t$) express homogeneity of space⁶ and rotations in space (\mathbf{r} when rotations are $\mathbf{r}' = \mathbf{R}\mathbf{r}$)

⁵With experiments on Earth, one can usually neglect the additional force terms, as the angular velocity $\boldsymbol{\omega}$ of Earth is $|\boldsymbol{\omega}| = 7.27 \cdot 10^{-5} \text{ s}^{-1}$.

⁶Homogeneity means, that no point in space is preferred in any way over a different point. All points are equal. In particular, this means, that the position of the origin of a coordinate system is irrelevant, as all possible points are equivalent.

express isotropy⁷ of space. When rotations are left out, setting $t_0 = 0$, and $\mathbf{r}_0 = 0$, the *Galilei transformations* are:

$$\mathbf{r}' = \mathbf{r} + \mathbf{v}t, \quad (3.130\text{a})$$

$$t' = t. \quad (3.130\text{b})$$

When taking the first and second derivatives of (3.130a) with respect to time t one obtains

$$\dot{\mathbf{r}}' = \dot{\mathbf{r}} + \mathbf{v} \quad (3.131)$$

and, as the relative velocity \mathbf{v} between the two systems is constant,

$$\ddot{\mathbf{r}}' = \ddot{\mathbf{r}}. \quad (3.132)$$

Thus, acceleration appears to be absolute in Newton's theory – there is no reference made to any coordinate system as a specific representation of space as in the case of position and velocity vectors. The Newtonian equations of motion for an N -particle system are thus *form invariant* under Galilean transformations, i.e. they become

$$m'_i \frac{d^2 \mathbf{r}^{i'}}{dt'^2} = \mathbf{F}'_i(|\mathbf{r}_{i'} - \mathbf{r}^{j'}|) \quad (i, j = 1, 2, \dots, N) \quad (3.133)$$

Newton's equation of motion have the same mathematical form in all IS, i.e. they are invariant under Galilei transformations. In an accelerated frame of reference however, there are additional forces, e.g. centrifugal or Coriolis forces. IS are distinguished from other frames of reference by the absence of inertial forces. This means, that laws of nature formulated in an IS have a simpler mathematical form than in accelerated frames of reference. This prominent role of inertial systems in classical mechanics was seen as a proof for the existence of absolute space. The famous bucket experiment performed by Newton was seen as further evidence.

Newton's Bucket Experiment and Mach's Principle

The bucket experiment is a device for detecting an *absolute acceleration* relative to absolute space. The experiment consists of suspending a bucket filled with water by a rope in an inertial frame. The rope is twisted and the bucket is released. Then there are four distinct phases of motion described in the caption of Fig. 3.24. The relative rotation of the water with respect to the bucket is the same in phases **a** and **c**, as well as in phases **b** and **d**, but the surface is flat in one case, and concave in the other. Newton argued that this proves that the shape of the water surface is only determined by the absolute rotation of the water with respect to absolute space.

In late 19th century, Ernst Mach suggested an explanation of this effect in terms of *relative motion*, namely the relative motion of the water and bucket together with respect to the rest of the universe. That is, according to Mach, all masses in the universe, and their distribution and motion, determine the inertia of the water and bucket. This means that a particle in an otherwise empty universe would experience no inertial effects. After introduction of a small particle it does not appear conceivable that

⁷Isotropy means, that no direction in space is preferred over any other direction.

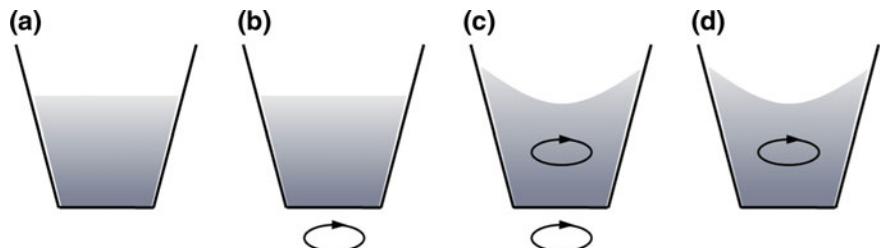


Fig. 3.24 Newton's bucket experiment. **a** At the first stage the bucket filled with water is at rest relative to the water. In **b** the bucket is set spinning. At first, the water surface stays flat and is at rest relative to the spinning bucket. After a while the water will start to rotate as well and the water surface becomes concave. At some point the water will be spinning just as fast as the bucket, i.e. the water is at rest relative to the bucket (same kinematic situation as in **a**), but the surface is curved (**c**). In **d** the bucket is instantaneously brought to rest while the water continues to spin relative to the bucket (same kinematic situation as in **b**) but the water surface is curved)

this introduction of a very small mass would restore the full inertia effects for the first particle. In particular it appears, that the inertia effects should be direction dependent if the universe were not isotropic. Experiments were carried out by Huge and Dreher around 1960 to test this hypotheses which established that mass is isotropic to at least one part in 10^{18} . The idea, that without mass there is no inertia, is called *Mach's Principle*, a term which was coined by Einstein in 1918 [218]. Einstein later tried in vain to incorporate Mach's idea into his completed Theory of General Relativity by introducing the infamous *cosmological constant*, which was to ensure that the equations do not admit any solution without mass. The cosmological constant prescribed boundary conditions for Einstein's partial differential equations, which led to a static universe solution. However, with the discovery of large-scale expansion of the universe by Hubble in the 1920s, the cosmological term was discarded. Einstein's vacuum field equations in fact admit Minkowski spacetime solutions even without mass. This means, that a test body in an otherwise empty universe would possess inertial properties even though (as a boundary condition) there is no matter to produce the inertia. Thus, Mach's principle has not really been integrated in Einstein's theory. The fundamental problem with Mach's principle is, that only space acts on matter, but it does not tell, how matter acts back on space. In [219] a collection of many articles devoted to Mach's principle discussing its current status in physics can be found.

Special Relativity Principle

The theory of special relativity of 1905 emerged at the end of a long series of theoretical and experimental investigations to identify absolute space as a frame of reference in which physical laws are formulated. The notion of "absolute space" was primarily based on Newton's authority who introduced it in his Principia in 1687. With the advance of new physical theories, in particular, Maxwell's equations which are not based on the concept of particles, and which did not obey the classical Galilei transformations, and the null result of Michelson's 1887 attempt to measure a

relative motion of the Earth with respect to “absolute space” [220] (which in turn was supposed to be at rest relative to a hypothetical medium, called ether), the problem of the compatibility of the behavior of the electromagnetic field with the motions of its sources, which are ponderous bodies, became the outstanding problem in physics toward the end of the 19th century.

The problem was solved with Einstein’s famous paper of 1905 “On the electrodynamics of moving bodies” [117] and the theory of special relativity (SRT), which was developed therein. It can be succinctly stated in axiomatic form. The two fundamental axioms of SRT are

1. All inertial observers are equivalent.
2. The velocity of light is the same in all inertial systems.

The Principle of Equivalence and of General Covariance

Physical events are described in an abstract mathematical model of space, a Riemannian manifold. For a *physical* description of events one additionally needs Field equations which describe the dynamics of matter (expressed as an energy-momentum tensor T^{ab}) in this mathematical framework. Einstein started his quest for GRT in order to remove the special status of inertial systems. In this quest he was guided by some heuristic principles. He fundamentally argued that general laws of nature should be the same in all coordinate systems. This can be stated along with several other principles that played a role in discovering the final field equations:

- **Principle of General Relativity:** All observers are equivalent.
- **Covariance Principle:** The equations of physics should have tensorial form.
- **Principle of Equivalence:** No local experiment can distinguish between non-rotating free fall in a gravitational field from uniform motion in space in the absence of a gravitational field.
- **Principle of Minimal Gravitational Coupling:** Matter only couples to the gravitational potential g , but not to the gradient of g , i.e. the curvature tensor.

The principle of General Relativity means that any coordinate system in curved spacetime may be used to formulate physical theories. In SRT, the Minkowski spaces are a canonical, or preferred system which does not exist in a manifold with a non-flat metric. All coordinate systems should lead to physical laws that are forminvariant, or covariant, which means, that the form of a law does not change under arbitrary coordinate transformations. This leads to the covariance principle which states that equations should have tensorial form. The principle of equivalence is important for making transitions from Minkowski space to a situation in an accelerated coordinate system within curved spacetime, i.e. when gravitational forces are present, as this principle says, that a non-rotating free falling reference system in a gravitational field is equivalent to an accelerated system far away from any sources of gravitation. The principle of minimal coupling was implicitly used by Einstein; it reduces the mathematical complexity of possible field equations which couple matter with curvature.

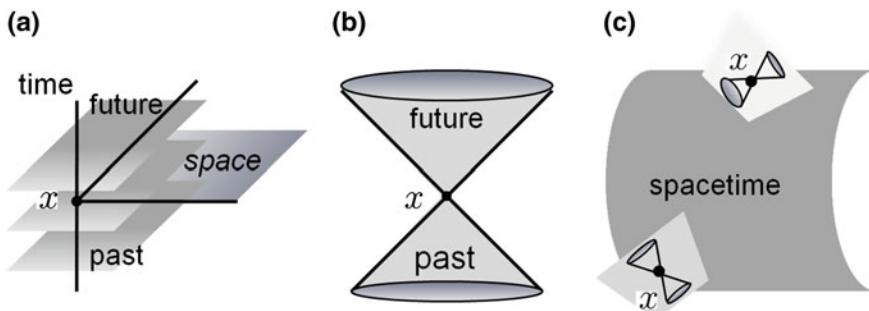


Fig. 3.25 The different structures of spacetime. **a** In prerelativistic theories, space is “sliced” by the global time coordinate which is the same for all observers. **b** The light cones in special relativity. Events outside the light one are not accessible from point x . **c** The local Minkowski tangent spaces of a Riemannian manifold in general relativity. Each local tangent space has its own light cone and the orientations are all different from each other

3.6.2 The Structure of Spacetime

In pre-relativistic Newtonian theory, the spacetime structure is very simple. Time and space are not connected in the Galilei transformation when changing between inertial systems. Thus, the time coordinate is the same parameter for all observers, cf. Fig. 3.25a. This allows for a definition of a *global* coordinate system valid for all observers. With the discovery of special relativity it became evident, that this concept of separated space and time is not realized in nature. Rather, all inertial frames are connected by a Lorentz-transformation which treats the three space parameters on a par with the time coordinate. Thus, events in spacetime are connected by time-like intervals, as all massive objects can only travel at a speed less than c . Light travels at speed c and thus, a light signal emerging from a point $x = (ct, \mathbf{x})$, displayed in a spacetime diagram with two spacial coordinates displayed, gives rise to a light cone, which displays the (future) part of spacetime that is accessible from point x , cf. Fig. 3.25b. In SRT, spacetime is a flat Minkowski space, i.e. the metric used for the description of spacetime events can always be transformed to a Euclidean metric. That is, because in SRT, all observers are located in IS. The theory of general relativity introduced the concept of curved spacetime in Riemannian manifolds. Hence, the spatio-temporal framework of all physical theories is a spacetime structure (ct, \mathbf{x}) which is mathematically represented by a four-dimensional differentiable Riemannian manifold, which is locally Euclidean,⁸ and in which each point can be addressed by a set of four real numbers in a sufficiently small region, cf. Fig. 3.25c. Physical theories parameterized by spacetime variables are considered more basic than those that are not. Many areas of physics have been successfully formulated within this spacetime framework, e.g. electrodynamics, mechanics, optics,

⁸That is, locally homeomorph to \mathbf{R}^n .

quantum electrodynamics, or elementary particle physics. For other areas, a general description within the framework of relativistic spacetime has not yet been completed. For example, in non-equilibrium thermodynamics or in elasticity theory, there are hardly any measurable relativistic phenomena, except for astrophysical applications, such that a decision between different formulations would be hard to make.

3.7 Relativistic Field Equations

Albert Einstein's quest for the field equations of the General Theory of Relativity (GRT) came to an end in the *mens mirabilis* November 1915 with four lectures that were delivered by Einstein at the Berlin Academy of Science. In this series of lectures [221–223] he approached step-by-step the final and correct field equations [133] which we just mention here:

$$R_{ab} - \frac{1}{2}g_{ab}R = \kappa T_{ab}, \quad (3.134)$$

where $\kappa = 8\pi G/c^4$ is a constant, c is the velocity of light and G is Newton's gravitational constant. The Eq. (3.134) are a system of 10 partial differential equations for determining the 10 metric functions g_{ab} .

The minutes of the Berlin academy containing Einstein's presentations have been collected and republished in [224]. Recent research on the history of the development of the GRT, analyzing newly discovered original sources [226–230], has shown that the derivation of the final and correct form of the relativistic field equations is in fact due to David Hilbert who submitted a manuscript "Die Grundlagen der Physik" [231]⁹ on 20 November 1915 – five days before Einstein gave his talk in Berlin – where he deduced the correct relativistic field equations based on an elegant variational principle. Einstein, in contrast, found his field equations based on a combination of mathematical deduction and physical heuristic assumptions on a relativistic theory of gravitation [232]. English translations of all papers of Albert Einstein pertaining to the general theory of relativity that were written in German in the original can be found in the book series "The Collected Papers of Albert Einstein" [224, 233–237].

The relativistic field equations are both non-linear and inhomogeneous with respect to the field variables and their derivatives. If the field equations are linear and homogeneous then the sum of two single solutions is again a solution; this is the case e.g. with Maxwell's electromagnetic field equations in empty space. The importance of this observation lies in the fact in such a theory it is impossible to yield the interaction between objects which can be represented as isolated solutions of the field equations. Thus, in order to describe the movement of material objects which are subject to the fields it is necessary to introduce additional laws, e.g. the Lorentz force in electromagnetic theory or Newton's third axiom in classical continuum theory.

⁹For a very recent English translation of Hilbert's lecture "On the Foundations of Physics", see [231].

3.7.1 Relativistic Hydrodynamics

Classical mechanics and kinetic theory are developed using particles as fundamental constituents of matter. As a further example that also a field theory can be reformulated in a way that the fundamental equations are Lorentz-covariant, the hydrodynamic equations are considered in this section. In a relativistic hydrodynamic field theory, the equations have to be such, that a sound wave can travel only with a velocity $v < c$. Characteristic for field theories such as hydrodynamics is, that matter is described as a continuum and not as an ensemble of particles. As a consequence, the four-momentum p^μ (2.14) is substituted by the energy-momentum tensor, generally denoted as $T^{\mu\nu}$, the components of which are the energy density, momentum density, energy-current density, and stress-density.

Starting from the Newtonian equation of motion $\Delta m \mathbf{v}/dt = \Delta \mathbf{K}$ for a mass element $\Delta m = \rho \Delta V$ of a fluid one introduces the mass density $\rho = \Delta m / \Delta V$ and the force density $\mathbf{k} = \Delta \mathbf{K} / \Delta V$. The force density $\mathbf{k} = -\nabla P + \mathbf{f}$ is split into two parts; one contribution comes from the pressure gradient $-\nabla P(\mathbf{x}, t)$ and another one comes from other external forces $\mathbf{f}(\mathbf{x}, t)$. For one selected mass element there are two contributions to the change of velocity: $(\partial_t \mathbf{x})dt$, due to the time dependence of \mathbf{v} , and the movement of the mass element by $d\mathbf{x}$. Thus, the Newtonian equation of motion yields the *Euler equation*:

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\nabla \mathbf{v}) \mathbf{v} \right) = -\nabla P + \mathbf{f}. \quad (3.135)$$

Together with the *continuity equation*

$$\frac{\partial \rho}{\partial t} = \nabla(\rho \mathbf{v}) = 0 \quad (3.136)$$

Equation (3.135) is the non-relativistic field equation for *ideal* fluids. Ideal, or perfect fluids are fluids with no friction (viscosity) and no heat conduction, which is a reasonable model for ordinary fluids and gases.

Incoherent Matter (Dust)

For the simplest kind of matter field, called *dust*, which is a pressure-less distribution of non-interacting particles (i.e. world lines that do not cross each other), one starts by writing down the four-velocity (see (3.36)):

$$u^a = \frac{x^a}{d\tau}, \quad (3.137)$$

where τ is the proper time along the worldline of a dust particle. The proper density $\rho = \rho_0(x)$ of the flow is the density measured by a co-moving observer in the flow. As the Euler equation is quadratic in \mathbf{v} and proportional to density, the simplest tensor of rank 2 that can be constructed using \mathbf{v} and ρ is:

$$T^{ab} = \rho_0 u^a u^b. \quad (3.138)$$

Using (2.14) one can express all components of T^{ab} by the components of \mathbf{v} :

$$T^{ab} = \rho \gamma^2 c^2 \begin{pmatrix} 1 & v^1/c & v^2/c & v^3/c \\ v^1/c & v^1/v^1 & v^1 v^2/c^2 & v^1 v^3/c^2 \\ v^2/c & v^2/v^1 & v^2 v^2/c^2 & v^2 v^3/c^2 \\ v^3/c & v^3/v^1 & v^3 v^2/c^2 & v^3 v^3/c^2 \end{pmatrix}. \quad (3.139)$$

The zero-component $\bar{\rho} = T^{00}/c^2 = \rho_0 \gamma^2$ has a simple interpretation: It is the relativistic energy-mass density of the matter field. Taking the divergence of T^{ab} yields:

$$\partial_b M^{0b} = c (\partial_t \bar{\rho} + \partial_c (\bar{\rho} v^c)), \quad (3.140)$$

and

$$\partial_b M^{ib} = \partial_t (\bar{\rho} v^i) + \partial_c (\bar{\rho} v^i v^c) = \bar{\rho} (\partial_t v^i + v^k \partial_c v^i) + v^i (\partial_t \bar{\rho} + \partial_c (\bar{\rho} v^c)). \quad (3.141)$$

Comparison of (3.140) with (3.136) and (3.141) with (3.135) yields the result:

$$\partial_b T^{0b} = 0, \text{ Equation of continuity}, \quad (3.142)$$

and

$$\partial_b T^{ib} = 0, \text{ Force free Euler equation}. \quad (3.143)$$

Both equations may be summarized to give the final equations governing the force-free motion of a matter field of dust particles:

$$\partial_b T^{ab} = 0. \quad (3.144)$$

This equation expresses the conservation of the four-momentum density. Equation (3.142) is replaced by its covariant counterpart, if one uses a non-flat metric:

$$\nabla_b T^{ab} = 0. \quad (3.145)$$

Perfect Fluid

For a perfect (ideal) fluid one has to add a scalar pressure field at the right-hand side of (3.144). For an ideal fluid pressure is isotropic, and in the limit of zero pressure, the ideal fluid reduces to incoherent matter.

$$T^{ab} = \rho_0 u^a u^b + p S^{ab}, \quad (3.146)$$

with some symmetric tensor which is given by

$$S^{ab} = \lambda u^a u^b + \mu g^{ab}. \quad (3.147)$$

Proceeding in the same way as for dust, one calculates the conservation law $\partial_b T^{ab} = 0$ and demands, that it reduces to (3.136) and (3.135) in an appropriate limit. This calculation is left to the reader as an exercise. As a result one obtains $\lambda = 1$ and $\mu = -1$ and thus

$$T^{ab} = (\rho_0 + p) u^a u^b - p g^{ab} \quad (3.148)$$

is the energy-momentum tensor of a perfect fluid.

3.8 Suggested Reading

An excellent introduction into set theory from a naive point of view is Halmos [238]. For a rigorous treatment of Zermelo–Fraenkel set theory, see Mendelson [239] and compare Appendix B. The book by Klingenberg [240] is a highly readable introduction into Riemannian geometry. Some excellent textbooks that treat the differential geometric aspects presented in this chapter in more detail are e.g. Darling [241], Hicks [242], Bishop and Goldberg [243], Kobayashi and Nomizu [244], Schouten [245], Hermann [246], Flanders [247], Wallace [248], or Schutz [249]. A book aimed at physicists that are mathematically literate, is Choquet-Bruhat [250]. Warner [251] is an excellent introduction of Lie groups on an introductory, undergraduate level. A comprehensive book with many exercises and lots of detail is the book by Spivak [252]. Probably one of the best introductions into manifold theory, differential geometry and the index-free tensor approach is the voluminous book by Misner, Thorne and Wheeler [213].

The three volumes by Duscheck and Hochrainer [253] are classic texts for engineers which introduce the component representation in a very thoughtful, pedagogical style from scratch. The books by Wills [254], Sygne and Schild [255], and Coburn [256] are comprehensive texts on vector and tensor analysis on an engineering level.

There are many excellent textbooks that teach the general theory of relativity and the corresponding mathematics. Among the many we mention Bergmann [211], who, in the 1940s was assistant of Einstein, French [257], or Rindler [258]. The very readable book by Schrödinger explains in a very pedagogical way the different algebraic structures that may gradually be introduced in a manifold. Wald [203] is an advanced textbook that is enjoyable, when one has digested most of the theory. Stephani [214, 259] is very readable, and Schutz [260] is an excellent textbook with many exercises, very adapted for self-study. The very readable classic by Weyl [212]—with the first edition published in 1918—is probably the first comprehensive book on the theory of relativity, which treats many fundamentals in great detail and in a way which is usually not found any more in modern texts on the subject. Last but not least, Schilpp [76] and Pais [93, 261] are very authoritative scientific biographies of Einstein, which also explain a good deal of relativity theory.

Problems

3.1 Sets

Which of the following terms are sets according to the notation introduced in Sect. 3.2?

- (a) $\{0, 11, 15, 16, 0, 3\}$, (b) (x, y, z) , (c) $\{A\}$, (d) $\{\{\emptyset\}\}$, (e) $\{\emptyset, \{1, 2\}, a\}$,
- (f) $[4, Z, w]$, (g) $\{1, 7, 9, m\}$.

3.2 Sets

Let $X = \{1, 2\}$, $Y = \{2, 3, 4\}$. Which of the following statements are correct?

- (a) $Y \subset X$, (b) $Y \supset X$, (c) $X \neq Y$, (d) $X = Y$, (e) $\{2, 4\} \subset Y$,
 (f) $Y \supset 3$, (g) $2 \in Y$, (h) $\{\{3, 4\}, 2\} \subset Y$.

3.3 Sets

Let the set X consist of n objects. Prove that the set $P(X)$ has exactly $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ subsets of X of k elements each. Hence, show that $P(X)$ contains 2^n members.

3.4 Topology

How many distinct topologies does a finite set having two or three points admit?

3.5 Tensors

Let

$$\begin{pmatrix} 1 & 0 & -1 \\ 0 & -1 & 1 \\ -1 & 1 & 1 \end{pmatrix} \quad (3.149)$$

the matrix of the coordinates A_j of a mixed tensor t of rank 2 with respect to a basis $\mathbf{B} = \{e_1, e_2, x_3\}$ and the dual basis $\mathbf{B}^* = \{e^1, e^2, x^3\}$. Calculate the coordinates of this tensor upon a change of basis:

$$\bar{e}_1 = e_2 + e_3, \quad \bar{e}_2 = e_1 + e_3, \quad \bar{e}_3 = e_2 + e_3. \quad (3.150)$$

3.6 Contraction

Show, that (3.65) is a tensor by transforming the components into a different (primed) coordinate system.

3.7 Affine Connection

Show, that Γ_{bc}^a is not a tensor.

Hint: Write down the transformation law for the three components when changing the coordinate system and use the chain rule of differentiation.

3.8 Christoffel Symbols

Establish (3.91) by assuming that the quantity defined by (3.90) has the tensor character indicated. Take the partial derivative of

$$\delta_{c'}^{a'} = \frac{\partial x^{a'}}{\partial x^{c'}} = \frac{\partial x^{a'}}{\partial x^d} \frac{\partial x^d}{\partial x^{c'}} \quad (3.151)$$

with respect to $x^{b'}$ to establish the alternative form (3.91).

3.9 Metric Connection

Write down the transformation law of g_{ab} and g^{ab} . Show directly, that the metric connection (3.106) indeed transforms like a connection.

3.10 Christoffel Symbols in orthonormal coordinates

The line elements of \mathbf{R}^3 in Cartesian, cylindrical polar, and spherical coordinates are given respectively by

1. $ds^2 = dx^2 + dy^2 + dz^2$,
2. $ds^2 = dr^2 + r^2 d\phi^2 + dz^2$,
3. $ds^2 = dr^2 + r^2 d\theta^2 + r^2 \sin^2 \theta d\phi^2$. Calculate in each case g_{ab} , g^{ab} , $g = \text{Det}(g_{ab})$, Γ_{bc}^a , Γ_{abc} .

3.11 Minkowski Coordinates

The Minkowski line elements is given by

$$ds^2 = dt^2 - dx^2 - dy^2 - dz^2, \quad (3.152)$$

with Minkowski coordinates

$$(x^a) = (x^0, x^1, x^2, x^3) = (t, x, y, z). \quad (3.153)$$

1. What is the signature?
2. Is the metric non-singular?
3. Is the metric flat?

3.12 Schwarzschild Coordinates

Taking the following coordinates

$$x^a = (x^0, x^1, x^2, x^3) = (t, r, \theta, \phi), \quad (3.154)$$

the four-dimensional spherically symmetric line element is

$$ds^2 = e^\nu dt^2 - e^\lambda dr^2 - r^2 d\theta^2 - r^2 \sin^2 d\phi^2, \quad (3.155)$$

with arbitrary functions $\nu = \nu(t, r)$ and $\lambda = \lambda(t, r)$.

1. Calculate g_{ab} , g^{ab} and g .
2. Calculate Γ_{bc}^a .
3. Calculate R_{bcd}^a
4. Calculate R_{ab} , R .

The Schwarzschild coordinates constitute a spherically symmetric solution of Einstein's field equations. In fact, this was the first approximate solution of the equations found by Karl Schwarzschild in 1916 [262].

3.13 Levi-Civita Tensor Density

Proof, that the tensor density ε_{abcd} is a Lorentz-invariant. Hint: Write down the transformation law for the components of ε as a determinant.

3.14 Covariant Derivative

In a plane polar coordinate system $r = x^1$ $\theta = x^2$ a vector filed is given by

$$v^1 = A \cos x^2, \quad v^2 = -(A/x^1) \sin x^2. \quad (3.156)$$

Calculate the covariant derivatives.

Fundamentals of Numerical Simulation

4

Abstract

In this chapter we discuss the prerequisites that are necessary to understand how problems in the natural sciences are formulated using (differential) equations and how to solve them numerically. For this, we discuss the basics of ordinary and partial differential equations in Sect. 4.1. We learn about the Lipschitz condition for ordinary differential equations and about initial and boundary conditions with partial differential equations, where we distinguish between three basic types of equations. A discussion of plane and spherical wave solutions of partial differential equations leads us to the study of numerical procedures in Sect. 4.2. Here, we discuss mesh-based and mesh-free methods, finite difference, finite volume, and finite element methods. We also get to know the most important algorithm used for numerical integration in molecular dynamics methods – the velocity Verlet algorithm. Finally, in Sect. 4.3 some tricks of the trade and important principles in software design and engineering which are not found in other books on computational science are presented. The chapter ends with a couple of problems for the interested reader.

Many of the fundamental physical laws that are essential for a description of fluid or solid systems in materials science are formulated most conveniently in terms of differential equations. Deriving and solving differential equations are thus common tasks when modeling material systems. Hence, this chapter first provides an overview of differential equations and then several numerical solution techniques – frequently used in materials science – are discussed.

4.1 Basics of Ordinary and Partial Differential Equations in Physics

There are physical quantities which - with appropriately chosen units - can be expressed by a single real-valued number. Very often however, one is looking for an unknown function, the derivation of which yields $f(x)$. In physics, the quantity x will most often be the time parameter t and thus $f(t)$ describes a process in time. In standard notation this problem can be written as

$$\frac{dy}{dx} = f(x). \quad (4.1)$$

Equations involving one or more scalar or tensorial dependent variables, unknown functions of these variables, as well as their corresponding derivatives, are called *differential equations*. For a modern and comprehensive treatment of differential equations see e.g. [263, 264]. Equations such as (4.1) with only *one* dependent and independent variable and their respective derivatives are called *ordinary differential equations* (ODE)s. The most general notation is the implicit form:

$$f\left(y^{(n)}(x), y^{(n-1)}(x), \dots, y'(x), y(x), x\right) = 0. \quad (4.2)$$

The *order* of a differential equation (DE) is the highest order of any of the derivatives occurring in the equation. For example, the relation between the motion of a point mass m under the influence of an external force \mathbf{F} is a second order differential equation. In one dimension, according to Newton's third axiom

$$ma = m \frac{d^2x(t)}{dt^2} = F(x, t). \quad (4.3)$$

A *linear* DE is a DE which is only linear in its unknown function $y(x)$ and its derivatives y', \dots . The most general form of such a DE is obtained when using the linear differential operator

$$L^{(n)}(x) = \sum_{k=0}^n f_k(x) \frac{d^k}{dx^k} \quad (4.4)$$

in the form:

$$L^{(n)}(x)y(x) = f(x). \quad (4.5)$$

The great importance and abundance of *linear* differential equations in the natural sciences has some good reasons. Physically, linearity means that phenomena occurring in nature exhibit superposition, e.g. electromagnetic waves in empty space. The realization of the fact that nature itself, to a good deal, "behaves linear" is neither self-evident nor in any way "obvious". The linearity of many natural phenomena in any case reduces the possible number of *useful* equations that might be used for theoretical model building in physics and thus leads to a powerful heuristic approach. However, there are also examples of fundamental equations which are highly non-linear, e.g.

Einstein's field equations which were treated in Sect. 3.7 where the gravitational field interacts with the masses which are themselves the sources of the gravitational field.

Linearity mathematically speaking, in essence means that for these equations the superposition principle applies, i.e. linear combinations of solutions are also solutions of the differential equation. These solutions even form a (linear) vector space (for the definition of a linear vector space see Chap. 3). This means that for obtaining all possible solutions to a linear problem, it is sufficient to obtain one particular solution that forms a basis (a “fundamental system”) of this vector space.

An *explicit* DE is one which can be resolved for the highest occurring derivative, i.e. for which $f^{(n)}$ can be written explicitly as a function of $y^{(n-1)}, \dots, y$ and x .

We now return to the simple DE of the motion of a point mass in (4.3). If the external force $F(x, t)$ is given by the gravitational force $m'g$ where g is the gravitational acceleration directed at the direction of fall, and m' is the *gravitational* mass, we obtain a one-dimensional ODE of motion:

$$m\ddot{x}(t) = m'g. \quad (4.6)$$

All experimental evidence (see the discussion on p. 174) shows that the *inertial* mass m equals the gravitational mass m' , thus:

$$\ddot{x}(t) = g. \quad (4.7)$$

Integrating this equation yields

$$\int_{\dot{x}_0}^{\dot{x}} dx'(t) = \int_{t_0}^t g dt', \quad (4.8)$$

and

$$\dot{x}(t) - \dot{x}_0 = g(t - t_0). \quad (4.9)$$

A second integration of (4.9) finally yields the equation of motion. Resolved for $x(t)$ the solution reads

$$x(t) = \frac{g}{2}(t - t_0)^2 + \dot{x}_0(t - t_0) + x_0, \quad (4.10)$$

where \dot{x}_0 is the initial velocity at time $t = t_0$, t_0 is the initial time and x_0 is the initial position of the point mass at $t = t_0$.

In this very simple example from classical point mechanics we see that a general solution of an ODE still contains some undetermined constants. Thus, the set of

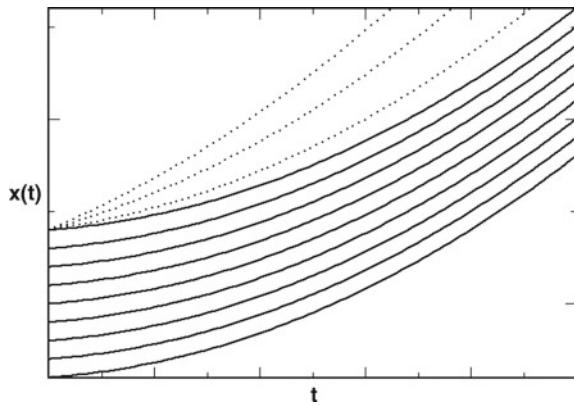
Box 4.1 Lipschitz condition.

Consider a function $f(x, y)$ in a domain $D \subset \mathbf{R}^2$. If for all pairs $(y, y_1), (y, y_2) \in D$ the inequality

$$|f(x, y_1) - f(x, y_2)| \leq L |y_1 - y_2| \quad (4.11)$$

is valid for any value $L > 0$, then $f(x, y)$ is said to satisfy a Lipschitz condition in y . The number L is called Lipschitz constant. One form of the Picard–Lindelöf theorem states that the initial value problem has a unique solution if $f(x, y)$ satisfies a Lipschitz condition in y on the interval D .

Fig. 4.1 A sample set of curves within $D = \{a, b\}$ for the solution (4.10) of (4.6) with different initial conditions of the initial position x_0 . The dotted lines are samples for curves with different initial velocity \dot{x}_0 and the same x_0



functions $y(x)$ which solve the ODE builds a family of curves within a domain $D \subset \mathbf{R}$, cf. Fig. 4.1.

Which one of these solutions is chosen is determined by initial and boundary conditions.

With a special solution these constants are already determined. A function $y(x)$ which fulfills the initial and boundary value problem of a DE in an interval $I \subset \mathbf{R}$; and which is continuously differentiable is called *solution* of the DE. Hence, with ODEs the complete task of solving an *initial-value problem* as in the example of the point mass above needs three specifications:

Initial Value Problem:

- Differential equation, e.g. $y'(x) = f(x, y)$,
- Initial condition, e.g. $y(x_0) = y(t = 0) = y_0$,
- Domain of the unknown function, i.e. the set of all points $(x, y) \in D \subset \mathbf{R}$.

Can a differential equation *always* be solved? This question is answered by Peano's theorem, see e.g. [265], which states that an initial value problem of ODEs such as in (4.3) has at least one solution on a planar domain D if $F(x, y)$ is continuous on D . The advantage of this theorem is that it has weaker premises than required by the Picard–Lindelöf theorem (see Box 4.1) which guarantees the existence and uniqueness of the solution to an initial value problem.

Example 33 Consider the initial value problem

$$y' = x\sqrt{y}, \quad y(0) = 0, \quad D = \{x \in \mathbf{R}, y \geq 0\}. \quad (4.12)$$

It is easy to verify that this equation has two solutions, namely $y = 0$ and $y = x^4/16$. Thus, not every initial value problem has a *unique* solution. If one simply relies on a computer code, e.g. in some simulation run, then this code will produce a solution, but one does not know which one.

The Picard–Lindelöf theorem (see Box 4.1) provides a sufficient condition for the existence and uniqueness of solutions which avoids the situation in the above example. This theorem states, that the initial value problem of ODEs has a unique solution if the function $f(x, y)$ obeys a Lipschitz condition on a domain $D \in \mathbf{R}^2$. A Lipschitz condition is a smoothness condition for functions which is stronger than regular continuity. Intuitively, a Lipschitz continuous function is limited in how fast it can change; a line joining any two points on the graph of this function will never have a slope steeper than a certain number called the Lipschitz constant of the function. For more details see e.g. [266].

Another aspect of the solution of ODEs is the question of stability of the solution with respect to small changes in the initial conditions. There are systems of ODEs where small changes in the initial conditions lead to an exponentially growing deviation of different solutions.¹ These systems are “chaotic” in nature. In particular when using numerical solution methods the question of stability of solutions is of great relevance.

The same equations of motion as in (4.3) are obtained for a mass point m when using the canonical Hamilton equations:

$$\dot{p}_i = -\frac{\partial H}{\partial q^i}, \quad \dot{q}^i = \frac{\partial H}{\partial p_i}, \quad (4.13)$$

where

$$H = H(p_i, q^i, t) = E(p_i) + \Phi(q^i, t) \quad (i = 1, 2, 3) \quad (4.14)$$

is the total energy of mass point m and p_i and q^i are the i -th components of the generalized momenta and coordinates in phase space. The Hamilton function of a simple point mass is $H = 1/2mq^2 + \Phi(q, t)$. Inserting this into (4.13) yields Newton's law $F_i = -\frac{\partial \Phi}{\partial q_i}$ for the i -th component of the force.

The physical world is not simply one-dimensional. Most problems in computational physics lead in their mathematical formulation to “partial differential equations” (PDEs) which involve derivatives with respect to several (space and time) variables. The Dirac equation of relativistic quantum mechanics is an example of a PDE of first order. The Schrödinger equation of non-relativistic quantum mechanics is an example of a second order PDE. While for the solution of PDEs of first order there is a number of different standard procedures, for the solution of second or higher order PDEs one has to adopt an individual approach. PDEs of up to second order can be written in a general form as:

$$\sum_{i,j=1}^k \alpha_{ij} \partial_i \partial_j F(x_1, \dots, x_k) = g(x_1, \dots, x_k), \quad (4.15)$$

¹For example, in molecular dynamics simulations, particle trajectories, which were initially very close, diverge exponentially with time. This well-known deviation of trajectories in MD is called Lyaponov instability.

where $\alpha_{ij} = \alpha_{ji}$ are the elements of a symmetric matrix. With this matrix one can write down the quadratic form

$$F(x_1, \dots, x_k) = \sum_{i,j=1}^k \alpha_{ij} x_i x_j. \quad (4.16)$$

If F is a constant then (4.16) describes higher order curves in k dimensions. Whether these curves assume an elliptic, hyperbolic or parabolic character, depends on the signs of the always existent Eigenvalues of the matrix α_{ij} . Depending on the relative signs of their respective derivatives, PDEs are associated with either one of the three categories of higher order curves. The most common partial differential equations in physics contain derivatives up to second order and can be grouped into and are named after one of the three categories:

- elliptic,
- parabolic,
- hyperbolic.

In the following sections the different types of PDEs are shortly discussed.

4.1.1 Elliptic Type

A prototype of an elliptic PDE in physics is the

$$\text{Laplace equation} \quad \Delta\phi = 0, \quad (4.17)$$

which, in Cartesian coordinates (x, y, z) and in three dimensions reads:

$$\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \phi(x, y, z) = 0 \quad (4.18)$$

In electrostatics, this equation describes the electric potential in a domain without charges. The distribution of charges is given by boundary conditions – the values of the potential at the boundary of the domain. In thermodynamics, the solution of (4.17) characterizes the temperature distribution at equilibrium in a domain without heat sinks or sources; thus it is the stationary solution of the heat diffusion equation. If one introduces explicit charges, i.e. sources of the electric potential, into (4.17) one obtains the

$$\text{Poisson equation} \quad \Delta\phi(\mathbf{x}) = \rho(\mathbf{x}) \quad (4.19)$$

in SI-units with $\rho(\mathbf{x})$ being the charge density. In the often used cgs-system there is an additional factor 4π at the right side of (4.19).² Point charges are denoted with delta functions.

Elliptic PDEs describe pure boundary value problems, i.e. the solution function ϕ obeys certain conditions at the boundary of the considered domain D . The solution

²For a thorough discussion of systems of units in classical electrodynamics, see e.g. [267].

function is typically independent of time, thus elliptic equations describe stationary situations. The specific type of boundary conditions is determined by the PDE and depends on the specific mathematical or physical problem. Two main types of boundary conditions are distinguished:

Dirichlet boundary conditions.

The values of the function ϕ are given at the boundary ∂D :

$$\phi(\partial D) = \dots, \quad (4.20)$$

and von Neumann boundary conditions.

Here, the values of the derivative of the function ϕ are given normal to the boundary curve or boundary area:

$$\nabla \phi \cdot \mathbf{n}|_{\partial D} = \dots. \quad (4.21)$$

Also mixed type boundary conditions are possible. It can be shown that the solution of an elliptic PDE is uniquely determined by specification of either von Neumann or Dirichlet boundary values at each point of the total boundary of the domain.

Remark 31 A general formal solution of the Poisson equation (4.19) can be obtained by using the Green's function $G(\mathbf{x} - \mathbf{x}')$, which obeys (4.19) for a unit point charge:

$$\Delta(G(\mathbf{x} - \mathbf{x}')) = \nabla^2(G(\mathbf{x} - \mathbf{x}')) = -4\pi\delta(\mathbf{x} - \mathbf{x}'). \quad (4.22)$$

$G(\mathbf{x} - \mathbf{x}')$ is given by

$$G(\mathbf{x} - \mathbf{x}') = \frac{1}{|\mathbf{x} - \mathbf{x}'|} + F(\mathbf{x}, \mathbf{x}'), \quad (4.23)$$

where the function $F(\mathbf{x}, \mathbf{x}')$ has to obey the Poisson equation *within* the considered integration volume V . The first term in (4.23) is the potential of a unit point charge. Thus $F(\mathbf{x}, \mathbf{x}')$ is the potential of an *external* charge distribution, i.e. a charge source outside of V , cf. Fig. 4.2.

Using definition (4.23) and Green's theorem, the formal solution of (4.19) can be written as [267]:

$$\phi(\mathbf{x}) = \int_V \rho(\mathbf{x}') G(\mathbf{x}, \mathbf{x}') d^3 x' + \frac{1}{4\pi} \oint_A \left[G(\mathbf{x} - \mathbf{x}') \frac{\partial \phi}{\partial n'} - \phi(\mathbf{x}') \frac{\partial G(\mathbf{x} - \mathbf{x}')}{\partial n'} \right] da'. \quad (4.24)$$

While the particular solution $\phi(\mathbf{x}, \mathbf{x}') = \frac{1}{|\mathbf{x} - \mathbf{x}'|}$ satisfies the Poisson equation (4.22), none of the boundary conditions are satisfied, as the surface A lies at infinity. However, by the definition of the Green's function in (4.23) and its additional term $F(\mathbf{x}, \mathbf{x}')$ one has the freedom to eliminate one of the two surface integrals in (4.24) by an appropriate choice of F , either corresponding to von Neumann or to Dirichlet boundary conditions. In practice however, it is often very difficult, if not impossible, to determine $G(\mathbf{x} - \mathbf{x}')$ as it depends on the shape of the considered integration surface A .

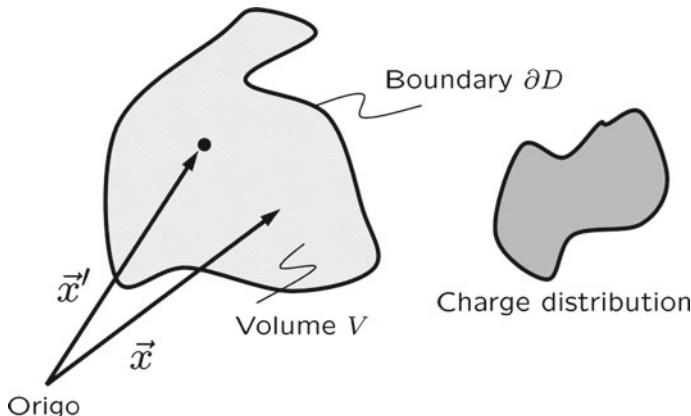


Fig. 4.2 Physical meaning of the function $F(\mathbf{x}, \mathbf{x}')$ in (4.23). $F(\mathbf{x}, \mathbf{x}')$ solves the Laplace equation within the domain on the left and thus is the potential of a charge distribution outside the volume V , which is chosen in such a way that $F(\mathbf{x}, \mathbf{x}')$ along with the potential $\frac{1}{|\mathbf{x}-\mathbf{x}'|}$ of a unit point charge located at \mathbf{x}' leads to a value of the Green's function $G_D(\mathbf{x}, \mathbf{x}') = 0$ and $\frac{\partial}{\partial n'} G_N(\mathbf{x}, \mathbf{x}') = -4\pi/A$ at the boundary (∂D) for which $\mathbf{x}' = \mathbf{x}'(\partial D)$. A is the area of the considered volume, subscripts D and N stand for Dirichlet and von Neumann boundary conditions and the value $-4\pi/A$ for the latter case is due to the Gaussian law of electrodynamics. The function $F(\mathbf{x}, \mathbf{x}')$, i.e. the external charge distribution depends on parameter \mathbf{x}' , which points to the point charge distribution in V

4.1.2 Parabolic Type

Here, aside from the spacial derivatives in the Laplace operator there is an additional derivative of first order; In most typical physical applications this will be a derivative with respect to time t .

$$\text{Diffusion equation} \quad \Delta \phi - \frac{1}{\kappa} \frac{\partial \phi}{\partial t} = 0, \quad (4.25)$$

with Δ being the Laplace operator in k dimensions and ϕ being dependent of k spacial variables and of time t . The solution of this equation might be a temperature distribution that depends on time t . In the limit $t \rightarrow \infty$ a stationary state will be reached, thus in the limit of $t \rightarrow \infty$ the ordinary Laplace equation has to be fulfilled: $\lim_{t \rightarrow \infty} \partial \phi / \partial t = 0$. Most often the physical situation requires a statement on the initial value of ϕ at time $t = 0$ in the total domain D :

$$\phi(x, y, \dots, t_0) = \dots \quad (4.26)$$

Additionally one may restrict the solution further by providing geometric restrictions to the domain D for points in time with $t > t_0$. The values of the solution at some time $t_1 > t_0$, $\phi(t = t_1)$ are part of the solution and need not be provided. Considering the total domain of $k + 1$ dimensions, boundary conditions are needed only for an open subset of the surface, namely

$$\phi(D, t = t_0), \quad (4.27)$$

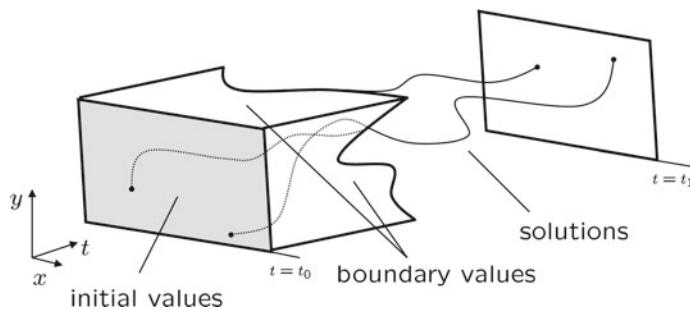


Fig. 4.3 Parabolic differential equations need as input an initial value, e.g. at time $t = t_0$. If additional boundary values are provided, the equation constitutes is a mixed initial/boundary value problem

$$\phi(\partial D, t_0 < t < t_1), \quad (4.28)$$

but no specification of $\phi(t_1)$ is needed in order to definitely determine the solution. If no boundary values are given the problem is an initial value problem, cf. Fig. 4.3.

4.1.3 Hyperbolic Type

A hyperbolic PDE contains – besides the Laplace operator – a second derivative with respect to time, which has the opposite sign with respect to the spacial derivatives. In one dimension, this equation reads:

$$\text{Wave equation} \quad \Delta\phi - \frac{1}{c^2} \frac{\partial^2 \phi}{\partial t^2} = g(x, t). \quad (4.29)$$

In physics this equation describes the motion of waves in spacetime. Parameter c is the wave propagation speed. If $g = 0$ the equation is called *homogeneous*, otherwise *inhomogeneous*. The function $\phi(x, t)$ in one dimension might describe the elongation of a point of an oscillating linear chain of mass points. In three spacial dimensions $\phi(x)$ can describe sound waves (density fluctuations) or the components of the electromagnetic field.

A hyperbolic PDE describes a typical initial value problem. A solution requires the initial values of both ϕ and its derivatives (**Cauchy-problem**) and - depending on the specific problem - boundary values at the boundary of the domain.

Plane Waves

Consider the homogeneous wave equation in one dimension:

$$\frac{\partial^2 \phi(x, t)}{\partial x^2} - \frac{1}{c^2} \frac{\partial^2 \phi(x, t)}{\partial t^2} = 0. \quad (4.30)$$

One can proof easily by insertion that any two times differentiable function of the form $f(x - ct)$ or $f(x + ct)$ is a solution of (4.30), see Problem 4.1. The function

$f(x - ct)$ describes a function which, without changing its shape, moves with constant velocity in positive x direction and $f(x + ct)$ describes a similar function moving in negative x direction. The solution ϕ depends on the phase $\varphi_{\pm} = x \pm ct$ and moves with velocity c in positive (φ_-) or negative (φ_+) x -direction.

Remark 32 Also non-linear PDEs may have special solutions which propagate through space with constant velocity. Such solutions are called solitary. In contrast to the linear wave equation (4.30) which is solved by *any* function that moves with velocity c , solitary solutions always have a special form $\phi(u)$. In some special cases solitary solutions might even superimpose without interfering with each other. In this case one speaks of *solitons*.

The general solution of the wave equation in three dimensions due to the above said is a function

$$\phi(\mathbf{x}, t) = f_+(\mathbf{x} - ct) + f_-(\mathbf{x} + ct). \quad (4.31)$$

As any function $f(\mathbf{u}_{\pm})$ solves the wave equation, so do the periodic functions

$$f_{\pm}(\mathbf{x}, t) = C_{\pm} e^{i(k\mathbf{x} - \omega t)}, \quad (4.32)$$

with a complex prefactor $C_{\pm} = \tilde{C}_{\pm} e^{i\alpha}$ and the phase α which is determined only up to modulo 2π . This solution has the property of being spatially periodic with wave length $\lambda = 2\pi/k$ ($k = |\mathbf{k}|$) for a fixed value t_0 of time and it is temporally periodic with the period $T = 1/\nu = 2\pi/\omega = \lambda/c$ for a fixed position \mathbf{x}_0 . It is

$$\nu = \frac{1}{T} = \frac{c}{\lambda} = \frac{ck}{2\pi}, \quad (4.33)$$

and

$$\omega = 2\pi\nu = kc, \quad (4.34)$$

where ν is the frequency, k is the wave number and $\omega = |\omega| = |\mathbf{k}|$ is the angular frequency of the solution.

We consider in the following only the partial solution f_+ . If the phase φ_+ is constant, then the surfaces of constant phase are given by the condition $\mathbf{k}\mathbf{x} = \text{const}$. This is the equation of a surface (wavefront) perpendicular to \mathbf{k} . For all points \mathbf{x} for which the projection $\mathbf{k}\mathbf{x}$ onto the direction of \mathbf{k} has the same value, the solution f_+ is a constant and is called a *plane wave solution*, cf. Fig. 4.4.

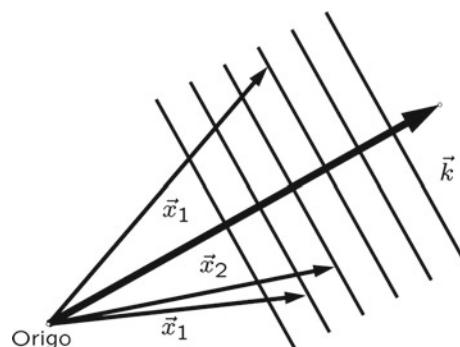
Spherical Waves

Another important solution of the wave equation (4.29) are *spherical waves*. Such a solution is of the form (see Problem 4.2)

$$\phi_{\pm} = \frac{A_{\pm}}{r} e^{i(kr \pm \omega t)} \quad (4.35)$$

with $r = |\mathbf{x}|$ and the following properties:

Fig. 4.4 Plane wave fronts as planes of constant phase, described with the condition $\mathbf{kx} = \text{const}$. For all points \mathbf{x} with the same value of the projection \mathbf{kx} into the direction \mathbf{k} , the phase φ_+ has the same value



1. The phase $\varphi_{\pm} = kr \pm \omega t$ only depends on the absolute value of the position vector $r = |\mathbf{x}|$. For a fixed time $t = t_0$ the points of equal phase, i.e. points for which ϕ has the same value, are the ones that have equal distance from the origin. These points lie on a sphere with radius r .
2. The amplitude of a spherical wave decreases with increasing distance from the origin as $1/r$.
3. The phase velocity u_{\pm} of the spherical wave front is given by the velocity of all points with equal phase, i.e. $u_{\pm} = \frac{d\varphi_{\pm}}{dt} = \frac{dr}{dt} \pm \omega = 0$. The solution (4.35) thus is an undulation with spherical wave fronts that propagate with phase velocity u_{\pm} , where u_+ is called *outbound spherical wave* and u_- is called *inbound spherical wave*.

4.2 Numerical Solution of Differential Equations

An analytic closed-form solution of differential equations is only possible in very rare cases. Sometimes certain symmetry conditions allow for a simplification of a problem, e.g. by reducing the number of dimensions. A good overview of many analytically solvable ODEs can be found in Kamke [268] or Boas [269]. For PDEs, Courant and Hilbert [266] and Morse and Feshbach are classics [270]. Farlow [271] discusses many analytic approaches to PDEs, including the separation of variables. The general solutions of PDEs have to take into account boundary values. Very often, it is possible to formulate an equivalent integral equation which implicitly contains the boundary conditions. The reformulation of PDEs as equivalent integral equations is discussed in Courant and Hilbert [266] and Porter and Stirling [272]. However, for most applications of practical interest with complex geometries involved, analytic solutions of PDEs are very rarely possible, and one has to make use of numerical methods. For a good overview of numerical solution methods of ODEs and PDEs see e.g. Morton and Mayers [273], Stoer and Bulirsch [274], Press et al. [275], Burden and Faires [276], Gould and Tobochnik [277] or Cohen [278]. For a compact and

succinct presentation of numerical methods, see e.g. Abramovitz and Segun [279] or Bronstein and Semendjajew [280]. As an example for a program package devoted to the numerical and analytical treatment of ODEs the reader is referred to Wolfram [281]. A corresponding detailed discussion of ODEs can be found in Abell and Braselton [282] and PDEs are treated in Vvedensky [283].

In the next section, we describe a number of different numerical techniques which are commonly used in simulation codes to obtain approximate solutions to boundary and initial value problems. Such problems are abundant in materials science, e.g. when solving the classical Newtonian equations of motion (discussed in detail in Sect. 6.4.1), the Navier–Stokes equations or some other basic physical equations.

4.2.1 Mesh-Based and Mesh-Free Methods

When numerically solving differential equations on a computer one has to handle two principal problems: First, the evaluation of derivatives which are actually defined by a limiting process, and second, the mapping of a continuous variable time onto a discrete scheme with intervals of length $h = \Delta t$. Hence, all numerical techniques for solving differential equations are based on the discretization of the time and space variables and the replacement of differential operators by difference operators.

One possible classification of methods could be done by distinguishing methods that are either based on discrete particles, so-called *mesh-free methods* – typically used in basic research in physics, chemistry, biology or genomics – and methods with a *mesh-based discretization* of the domain for which a solution is sought. The latter methods, based e.g. on *finite elements* (FE) have been used almost exclusively in engineering applications for more than sixty years since their first introduction which is generally attributed to A. Hrennikoff [284] and R. Courant [285]. Finite element methods (FEM) – discussed in Sect. 4.2.4 – are designed to solve numerically both, complex boundary-value, and initial-value problems. The common feature of all FEM is the reformulation of the investigated differential equation as a variational problem. Variational problems have solutions in a special class of function spaces, the so-called Sobolev spaces, cf. the discussion in Chap. 3. For a numerical treatment, a minimization of the variational problem is done in finite-dimensional subspaces, the so-called *finite element spaces*. Characteristic for the numerical treatment is a discrete division of the spacial domain Ω into a number of sub-domains, the so-called finite elements. Within these elements one considers polynomial trial functions as an approximation of the “true” solution.

Finite difference methods and even more finite element methods – in particular within typical engineering applications – are generally associated with the treatment of (spacial) large-scale problems. While finite element methods are in fact prevailing in virtually all engineering applications on the meso- and macroscale in industry and science (and are often in fact the *only* modeling tools available to engineers) it is very important to understand that there is *no intrinsic length or time scale* associated with FEM, or generally, with any numerical mesh-based method. The same is true for particle methods such as molecular dynamics (MD), smooth particle hydrodynamics

(SPH) or discrete element methods (DEM). Thus, in principle, one could also use finite elements for the modeling of atomic systems in thermodynamics or one could as well use a particle based approach – solving Newtonian equations of motion – for any large-scale structural problem. It is the underlying physics that determines a length (and associated time-scale) of a problem, whereas the different numerical schemes for the solution of basic equations are not intrinsically coupled to any specific scale. This is the reason why we will discuss the FEM method as such separately in this chapter in Sect. 4.2.4 and not in the corresponding chapters devoted to meso- and macroscales in Part II of this book. In those chapters we will focus on the physical foundations of continuum theory and the phenomenological assumptions (so-called constitutive equations) on macroscopic material behavior that are used along with finite element approaches.

One of the main limitations of a finite element approximation is that it only works on a prearranged topological environment – a mesh – which is a rather artificial constraint to ensure compatibility of the finite element interpolation. A mesh itself is not compatible, and in fact is often in conflict with the real physical compatibility condition that a continuum possesses. As soon as conflicts between the mesh and a physical compatibility occur, a FEM simulation is stopped due to numerical instabilities, and re-meshing becomes inevitable. This again not only is a time consuming process but also leads to a degradation of computational accuracy, finally tainting the numerical results. A too large distortion of a mesh usually leads to numerical instabilities which is often the case e.g. in shock wave applications which are discussed in Chap. 8. Since the beginning 90s of the last century there has been an increasing interest in the development of meshfree interpolant schemes that can relieve the burden of successive re-meshing and mesh generation, which have led to serious technical obstacles in some FE procedures such as adaptive refinement and simulations of solid objects moving in a fluid field. The investigation of meshfree interpolants by e.g. Needleman [286], Liberski et al. [287], Nayroles et al. [288], Belytschko et al. [289, 290], Lui et al. [291] and others have led to the adaptation of meshfree methods – which were originally developed in other research areas – such as *Smooth Particle Hydrodynamics* or *discrete element methods* (DEM) (see Chap. 7) to solid mechanics problems. During the past decade, meshfree approaches have attracted much attention due to their applications to computational failure mechanics, crack growth, multiscale computations and even nano-technology. As a growing field within engineering applications, meshfree methods have shown promising potential to become a major numerical methodology in engineering computations.

Generally speaking, it seems obvious that a good simulation program requires a good algorithm to integrate the underlying basic equations, e.g. Newton's equation of motion. At first sight, one might think, that speed of an algorithm is important, but this is usually not the case, as the time spent with integrating the equations of motion is small compared to the time needed to search for interacting particles, cf. Sect. 6.4.1. Although it seems easy to recognize a *bad* algorithm, it is not directly clear what criteria a *good* algorithm should satisfy. The main demands on an integrator are:

- Energy conservation
- Accuracy for large timesteps
- Numerical efficiency
- Time reversibility
- Low memory requirements

Energy conservation is an important issue when solving physical equations of motion, described e.g. using Hamiltonian dynamics. Sophisticated higher order integration schemes often have an excellent energy conservation for short times while they show an undesired and sometimes catastrophic long term energy drift [49]. Accuracy of an algorithm for large time steps is important, because the larger the time step one can use, the fewer calculations of the interactions have to be done per unit of simulation time. Higher order schemes achieve this by storing information on the higher-order derivatives of the particle coordinates. It is important to understand that too high an accuracy of the integration algorithm is not necessary in MD simulations, as with MD, one is primarily interested in thermodynamic *averages*, i.e. statistical properties of observables and not in an exact prediction of particle trajectories.³ The trajectories, no matter how close at the beginning of a simulation, will diverge exponentially from their “true” trajectory which is caused by the numerical rounding errors. This well-known instability of particle trajectories is called *Lyapunov instability*. However, there is considerable numerical evidence (but no *formal proof*), that the results of MD simulations are representative of a “true” trajectory in phase space, albeit one cannot tell a priori which one [292, 293]. Thus, the confidence in molecular dynamics simulations is based largely on numerical evidence.

True Hamiltonian dynamics also leaves the size of a volume Ω in phase space, spanned by the coordinates \mathbf{r}_i and momenta \mathbf{p}_i , invariant.⁴ Many non-symplectic, i.e. time-reversible integration schemes do not reproduce this area-preserving property. Area-preserving, time-reversible algorithms do not change the phase space volume. Last but not least, the limited precision of representing real numbers on a computer by bits of memory, cf. Fig. 4.5 results in unavoidable rounding errors, which is why *no* implementation of a symplectic algorithm is truly time-reversible.

The basic equations of motion are given in Cartesian coordinates by the canonical equations

$$\dot{\mathbf{r}}_i = \frac{\partial H}{\partial \mathbf{p}_i} \quad \dot{\mathbf{p}}_i = -\frac{\partial H}{\partial \mathbf{r}_i}, \quad (4.36)$$

where the partial derivatives $\partial \mathbf{r}_i$ and $\partial \mathbf{p}_i$ are a symbolic (phycisists’) notation for the derivative with respect to each of the $i = N$ components of the generalized coordinates \mathbf{r}_i and momenta \mathbf{p}_i . $H = K + \Phi$ is the Hamiltonian, the sum of kinetic

³In this respect, MD simulations differ fundamentally from other numerical schemes for, say, predicting the trajectory of satellites through space. Here, of course, one is interested in the exact ballistic curve of the considered object and cannot afford to just make statistical predictions on their whereabouts.

⁴This is Liouville’s theorem, cf. Theorem 7 on p. 268.

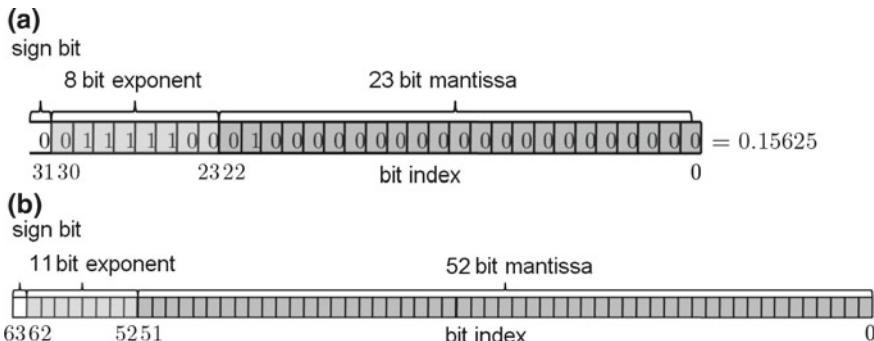


Fig. 4.5 Single precision representation of a real number on a computer. **a** On 32-bit systems, for a single precision variable 32 bits of memory are allocated. Thus, there are 8 significant figures and a maximum exponent of +127. **b** Double precision increases accuracy by allocating 64 bits of memory. In this case there is a maximum of 16 significant figures and a maximum exponent of ± 324 .

and potential energy for an ergodic system. Starting point for a discretization and subsequent MD simulation of a N -body system are the *Newtonian equations of motion*. The second axiom states these equations as

$$m_i \ddot{\mathbf{r}}_i = -\frac{\partial \Phi}{\partial \mathbf{r}_i} = \mathbf{f}_i, \quad (4.37)$$

where f_i are the forces, Φ is the total potential and m_i the particles' masses.

Several schemes for an integration of this system of coupled differential equations exist [53, 294]. Generally, one likes to make the time step during the integration as large as possible in order to quickly sample the phase space of the system. This is the reason why a low order integration scheme is more advantageous to this purpose, i.e. an integration scheme that does not involve calculating and storing higher order derivations of the positions or velocities. Such an integrator allows for a higher time step without jeopardizing energy conservation. If the system has reached equilibrium, measurements of observables are done. Due to the ergodic and chaotic nature of classical trajectories, it is useless to follow the detailed trajectory of single atoms or molecules for a longer time, as initially nearby trajectories of particles diverge from each other exponentially. Instead, the propagation of particle positions serves primarily for a sufficient sampling of the system's phase space.

Standard Hamiltonian dynamics, cf. e.g. [295], describes the flow of a representative N -particle systems through phase space with the classical statistical distribution function $\rho(\mathbf{r}^N, \mathbf{p}^N, t)$, which obeys the *Liouville equation*

$$\frac{\partial \rho}{\partial t} = \sum_i \left[\dot{r} \frac{\partial}{\partial r_i} + \dot{p}_i \frac{\partial}{\partial p_i} \right] =: -i \mathcal{L}. \quad (4.38)$$

$i\mathcal{L}$ is the *Liouville operator* which can be split in two parts [296], the momentum part $i\mathcal{L}_p$ and the coordinate part $i\mathcal{L}_r$ such that

$$\begin{aligned} i\mathcal{L} &= i\mathcal{L}_p + i\mathcal{L}_r \\ &= \sum_i \dot{\mathbf{p}}_i \frac{\partial}{\partial \mathbf{p}_i} + \sum_i \dot{\mathbf{r}}_i \frac{\partial}{\partial \mathbf{r}_i} \\ &= \sum_i f_i \frac{\partial}{\partial \mathbf{p}_i} + \sum_i m_i^{-1} \mathbf{p}_i \frac{\partial}{\partial \mathbf{p}_i}. \end{aligned} \quad (4.39)$$

The time evolution of a dynamic variable $A = A(\mathbf{r}^N, \mathbf{p}^N, t)$ in Hamiltonian dynamics can be written as:

$$\dot{A} = \sum_{i=1}^N \dot{\mathbf{r}}_i \frac{\partial A}{\partial \mathbf{r}_i} + \dot{\mathbf{p}}_i \frac{\partial A}{\partial \mathbf{p}_i} = i\mathcal{L}A \quad (4.40)$$

$$\Leftrightarrow A(t) = \exp(i\mathcal{L}t)A(0). \quad (4.41)$$

The exponential operator $\exp(i\mathcal{L}t)$ is called a *propagator*, due to the following identity:

$$\exp(i\mathcal{L}t) = \lim_{k \rightarrow \infty} [\exp(i\mathcal{L}_p t/2) \exp(i\mathcal{L}_r t) \exp(i\mathcal{L}_p t/2)]^k. \quad (4.42)$$

Using a small discrete timestep, Eq. 4.42 can be rewritten as an approximation to $\exp(i\mathcal{L}\Delta t)$ using the Trotter identity:

$$\begin{aligned} \exp(i\mathcal{L}\Delta t) &= \exp(i\mathcal{L}_p + i\mathcal{L}_r)\Delta t \\ &= \exp(i\mathcal{L}_p \Delta t/2) \exp(i\mathcal{L}_r \Delta t) \exp(i\mathcal{L}_p \Delta t/2). \end{aligned} \quad (4.43)$$

Thus, the Liouville operator propagates the equation of motion in several steps which ignore, in turn, the kinetic and the potential part of the Hamiltonian. Applying first the operator $\exp(i\mathcal{L}_p \Delta t/2)$ to the observable A one obtains:

$$\exp(i\mathcal{L}_p \Delta t/2) A [\mathbf{r}_i(t), \mathbf{p}_i(t)] = A [\mathbf{r}_i(t), \mathbf{p}_i(t) + \mathbf{p}_i(t + \Delta t/2)]. \quad (4.44)$$

The next part of the operator applied to A yields

$$\begin{aligned} &\exp(i\mathcal{L}_r \Delta t) A [\mathbf{r}_i(t), \mathbf{p}_i(t) + \mathbf{p}_i(t + \Delta t/2)] \\ &= A [\mathbf{r}_i(t) + \mathbf{r}_i(t + \Delta t), \mathbf{p}_i(t) + \mathbf{p}_i(t + \Delta t/2)]. \end{aligned} \quad (4.45)$$

Finally one obtains with the third step

$$\begin{aligned} &\exp(i\mathcal{L}_p \Delta t/2) A [\mathbf{r}_i(t) + \mathbf{r}_i(t + \Delta t), \mathbf{p}_i(t) + \mathbf{p}_i(t + \Delta t/2)] \\ &= A [\mathbf{r}_i(t) + \mathbf{r}_i(t + \Delta t), \mathbf{p}_i(t) + \mathbf{p}_i(t + \Delta t/2) + \mathbf{p}_i(t + \Delta t/2)]. \end{aligned} \quad (4.46)$$

Algorithm 4.1 The velocity Verlet algorithm

$$1. p(t + \frac{1}{2}\Delta t) = p(t + \frac{1}{2}\Delta t)f(t) \quad (4.47)$$

$$2. r(t + \Delta t) = r(t) + \Delta t p(t + \frac{1}{2}\Delta t)/m_i \quad (4.48)$$

$$3. f(t + \Delta t) = fr(t) + \Delta t, p(t + \Delta t/2) \quad (4.49)$$

$$4. p(t + \Delta t) = p(t + \frac{1}{2}\Delta t) + \frac{1}{2}\Delta t f(t + \Delta t) \quad (4.50)$$

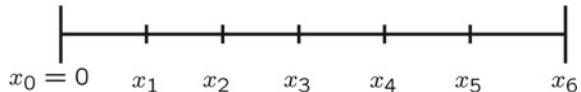
From the previous steps one has obtained one of the most commonly used algorithm to integrate the classical Hamiltonian equations of a system, namely the *Verlet velocity algorithm*. Written in standard notation (dropping the vector symbol and index i), this algorithm consists of four steps, cf. Algorithm 4.2.1.

The Verlet algorithm exists in various, essentially equivalent versions, based on a Taylor expansion of the position vector, including the originally proposed method [37, 297] (see also Problem 4.3), the above velocity version and a leapfrog form [48], see Sect. 4.2.3. For almost all applications using molecular dynamics simulations, the Verlet algorithm is usually the best and the algorithm of choice. Occasionally however, one might want to use a larger time step without loss of accuracy. As mentioned before, higher order schemes or often not reversible or area conserving and require more memory. To evaluate the different merits of algorithms, the reader is referred to a review by Berendsen and van Gunsteren [298].

4.2.2 Finite Difference Methods

The numerical solution of a differential equation involves the *discretization* of the independent variables of the equation (usually time and space coordinates). In numerical analysis the term *discretization* refers to passing from a continuous problem with an infinite number of degrees of freedom to one considered at only a *finite* number of points which may or may not be located on a regular discrete lattice. Thus, the discretization reduces the differential equation to a system of algebraic equations. These equations determine the values of the solution at a finite number of points within the domain Ω where a solution is sought. One of the simplest techniques to discretize a differential equation is based on a discrete approximation of the differential operator in the equation. Thus, this method substitutes the temporal and spacial differential operators of an equation by their finite parts of the infinite Taylor Series of the respective differential operators. Such a technique is called *finite difference method*. The variety of finite difference methods currently in use for many different problems in engineering and science [49, 57, 273, 278, 279] is evidence that there is no single algorithm superior under all different conditions encountered in practice. Thus, experience is needed to do the right choices for certain problems.

Fig. 4.6 Equispaced one-dimensional grid with $i_{\max} = 6$



Example 34 (Finite difference discretization in one dimension) In the one-dimensional case, the domain Ω on which a differential equation is to be solved is an interval $\Omega := [0, a] \subset \mathbf{R}$. The interval is subdivided into i_{\max} subintervals of equal size $\delta x = a/i_{\max}$. Thus, one obtains a one-dimensional grid as depicted in Fig. 4.6.

The same scheme applies for a discretization of the time variable in the case $\delta x = \Delta t$. Recalling the derivative of a function $u : \mathbf{R} \rightarrow \mathbf{R}$

$$\frac{du}{dx} = \lim_{\delta x \rightarrow 0} \frac{u(x + \delta x) - u(x)}{\delta x}, \quad (4.51)$$

one can approximate the differential operator at grid point x_i by its difference counterpart

$$\left[\frac{du}{dx} \right]_i^f = \frac{u(x_{i+1}) - u(x_i)}{\delta x} \text{ (forward difference)}, \quad (4.52)$$

where $x_{i+1} = x_i + \delta x$ is the right neighboring grid point of x_i . Instead of this *forward difference* one could also use the *backward difference*

$$\left[\frac{du}{dx} \right]_i^b = \frac{u(x_i) - u(x_{i+1})}{\delta x} \text{ (backward difference)}, \quad (4.53)$$

or the *central difference*

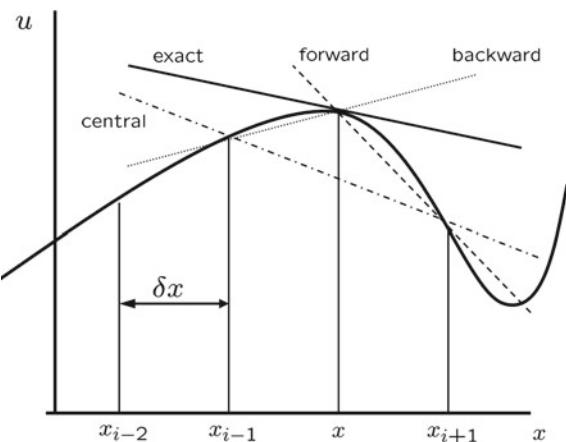
$$\left[\frac{du}{dx} \right]_i^c = \frac{u(x_{i+1}) - u(x_{i-1})}{2\delta x} \text{ (central difference)}. \quad (4.54)$$

From (4.52) to (4.54) – which are illustrated in Fig. 4.7 – it is immediately clear that a refinement of the grid, i.e. reducing the stepsize δx , results in a better approximation of the differential quotient.

The *discretization error* or *truncation error*,⁵ i.e. the difference between the “true” solution and the approximate finite difference solution for the forward and backward difference is of the order $\mathcal{O}(\delta x)$. Thus, in order to halve the error one has to halve the stepsize δx . The error of the central difference is of the order $\mathcal{O}(\delta x^2)$. In this case, halving the stepsize δx reduces the error by a factor of four.

⁵This term reflects the fact that a finite difference operator is simply a finite part of a Taylor expansion of the differential operator.

Fig. 4.7 The finite difference scheme as an approximation of a derivative



Algebraic Equations

A finite difference approximation scheme provides an algebraic equation at each node; this equation contains the variable value at this node as well as values at neighboring nodes and may also include some non-linear terms. The general form of this equation is:

$$A_P \Phi_P + \sum_l A_l \phi_l = Q_P , \quad (4.55)$$

where P denotes the node at which the partial differential equation is approximated and index l runs over the neighbor nodes involved in finite difference approximations. The coefficients A_l involve geometrical quantities, fluid properties and, in the case of non-linearity, the values themselves. Q_P contains all those terms which do not contain unknown variable values. The numbers of equations and unknowns must be equal, i.e. there has to be one equation for each grid node, which results in a large set of linear algebraic equations, that is *sparse* – each equation contains only a few unknowns. The algebraic system can be summarized as follows:

$$A\Phi = Q , \quad (4.56)$$

where $A \in \mathbf{R}^{(n \times n)}$ is the square sparse coefficient matrix, Φ is a vector containing the variable values at the grid nodes, and Q is the vector, containing the terms on the right hand side of (4.55). The structure of matrix A is dependent upon the ordering of variables in the vector Φ . For *structured grids*, a lexicographic ordering leads to a poly-diagonal structure. The variables can be most efficiently stored in one-dimensional arrays. The usual conversion between the different node locations on the grid, cf. Fig. 4.8, and the storage location in computer memory, see Table 4.1.

For unstructured grids, the coefficient matrix remains sparse, but it has no longer the diagonal structure.

Fig. 4.8 Example of computational nodes and their compass notation. In three dimensions, *B* and *T* for *bottom* and *top* are added

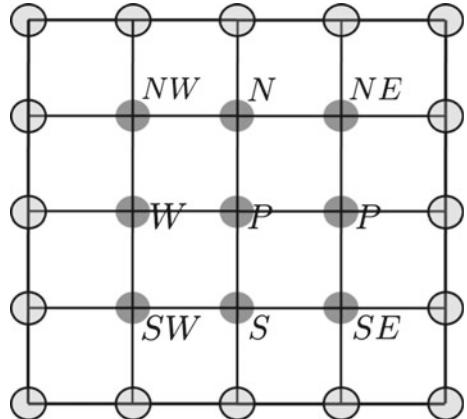


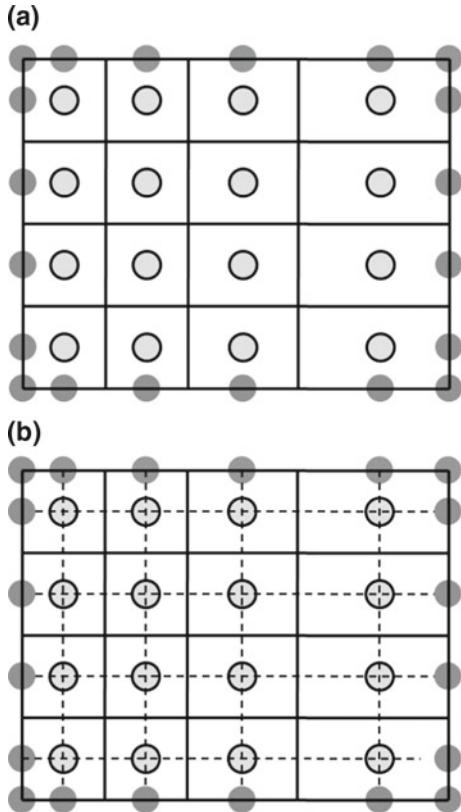
Table 4.1 Storage location of grid indices obtained from compass notation of grid nodes, cf. Fig. 4.8. N_i and N_j are the number of grid points in the two coordinate directions

Grid location	Compass notation	Memory location
i, j, k	P	$l = (k-1)N_j N_i + (i-1)N_j + j$
$i, j+1, k$	N	$l + 1$
$i-1, j, k$	W	$l - N_j$
$i, j-1, k$	S	$l - 1$
$i+1, j, k$	E	$l + N_j$
$i, j, k-1$	B	$l - N_i N_j$
$i, j, k+1$	T	$l + N_i N_j$

4.2.3 Finite Volume Method

The finite volume method (FV) is very common in computational fluid dynamics simulations when one wants to approximate the fluxes propagating through surfaces of volumes. Fluid dynamics is a field theory based on the general conservation equation of for a quantity ϕ in their integral formulation, cf. Chap. 8. Here, the solution domain is subdivided into a grid of a finite number of contiguous control volumes (CVs), and the conservation equations are applied to each CV. At the centroid of each CV lies a computational node at which the variable values are to be calculated. Interpolation is used to express variable values at the CV surface in terms of the nodal (CV-center) values. Surface and volume integrals are simply approximated employing suitable quadrature formulae. Hence, one obtains as a result an algebraic equation for each CV, in which the nodal values of neighbor nodes appear. An advantage of this method is that it can accommodate any type of grid, so this grid-based method can be applied to any type of geometry. The grid itself defines only the control volume boundaries and need not be related to a coordinate system. The method is conservative by construction, so long as surface integrals which represent the con-

Fig. 4.9 Two different types of Cartesian finite volume grids. **a** The grid nodes are centered in control volumes (CVs). **b** CV faces are centered between nodes



vective and diffusive fluxes, are the same for the CVs sharing the boundary. The FV approach is probably one the simplest approaches possible and very straightforward to program. This method is especially popular in computational engineering applications fundamentally based on meshing the considered solution domain with a finite number of small control volumes, cf. Fig. 4.9.

The approximation of surface integrals is straightforward. The CV surface in 2d (3D) can be subdivided into four (six) plane faces, with respect to the node pertaining to the volume enclosed by the surfaces. The net flux through the surfaces is the sum of integrals over the four (six) CV faces:

$$\int_{\partial\Omega} f \, dS = \sum_k \int_{\partial\Omega_k} f \, dS , \quad (4.57)$$

where f is the component of the convective or diffusive vector in the direction normal to CV face. For the validity of conservation equations it is important that CV's do not overlap; each CV is unique to two CV's which lie on either side of it. To calculate the surface integral in (4.57) exactly, one had to know the integral kernel f everywhere

on the surface S . However, since only the nodal values are calculated, one has to introduce an approximation:

- The integral is approximated in terms of the variable values at one or more locations on the cell face,
- The cell face values are approximated in terms of the nodal (CV center) values.

The simplest access to the approximate the calculation of the integral (4.57) is the *midpoint rule* which approximates the integral as a product of the integrand at the cell face center and the cell face area:

$$F = \int_{\partial\Omega} f \, dS = \bar{f} S \approx f S . \quad (4.58)$$

This approximation is of second order. The values of f have to be obtained by interpolation. In order to preserve the second order accuracy of the midpoint rule approximation of the surface integral, e.g. by the use of higher order schemes. A different simple method is the trapezoid rule:

$$F = \int_{\partial\Omega} f \, dS \approx \frac{S}{2}(f_a + f_b) , \quad (4.59)$$

where f_a and f_b are the values of the integrand in opposite directions with respect to the control volume. If the variation of f is assumed to have some particular simple shape (e.g. an interpolation polynomial), the integration is easy. The accuracy of the approximation then depends on the order of shape functions.

Euler Methods

The most simple integration scheme is probably obtained by a truncated Taylor expansion of the particle coordinates r :

$$r_{t+\Delta t} = r_t + v_t \Delta t + \frac{a_t}{2m} \Delta t^2 + \dots \quad (4.60)$$

When the expansion is truncated as in (4.60) one obtains the *Euler algorithm*. This algorithm is known to suffer from a huge energy drift and it is neither symplectic,⁶ nor area preserving. Hence, this algorithm should not be used for solving Newton's equation of motion. In fact, it is mostly used in engineering contexts, as an *implicit* integration scheme.

Leap-Frog Method

The *leap-frog method* is equivalent to the Verlet algorithm. With this algorithm, the velocities are calculated at half-integer time steps and the velocities are used to compute the new positions as follows:

$$v_{t-\Delta t/2} = \frac{r_t - r_{t-\Delta t}}{\Delta t} , \quad (4.61a)$$

$$v_{t+\Delta t/2} = \frac{r_{t+\Delta t} - r_t}{\Delta t} , \quad (4.61b)$$

⁶Time reversible.

Resolving (4.61b) for the position $r_{t+\Delta t}$ one obtains:

$$r_{t+\Delta t} = r_t + \Delta t v_{t+\Delta t/2}. \quad (4.62)$$

From (4.48) one gets the following equation for the updated velocities:

$$r_{t+\Delta t/2} = v_{t-\Delta t/2} + \Delta t \frac{f_t}{m}. \quad (4.63)$$

The leapfrog algorithm is derived from the Verlet algorithm; thus, the obtained particle trajectories in a simulation are the same. A disadvantage of the leap-frog scheme is, that the positions and velocities are not calculated at the same time, and thus, one cannot calculate directly the total energy using this scheme.

4.2.4 Finite Element Methods

In the numerical treatment of elliptic and parabolic DEs, the finite element method is very often used; in particular in engineering applications this has long been *the* discretization method that has been used to solve stress analysis, heat transfer and other types of engineering problems. The FE method is very well suited to be used within the framework of a variational formulation of differential equations and their discretized algebraic counterparts of the problem under investigation. This is the reason why FEM – in contrast to finite difference or finite volume methods – is more flexible also for the treatment of difficult problems with complex geometries and boundary conditions. The development of the FE method has been restricted to engineering applications for a long time, until in the 70s the method was standardized as a theory by mathematicians. The discretization with the FE method may lead to very large systems of equations for which the effort is of the order $\mathcal{O}(n^2)$. In the last two decades, some new procedures such as multigrid-methods or conjugate gradient methods (which are also used in particle-based codes, e.g. in quantum chemistry) have been developed, which allow for the implementation of very efficient solvers. In Chap. 7, we discuss shortly the basics of the theory of finite elements and present some typical application. The finite element method is based on solving a system of equations that describe some parameter (e.g. displacement) over the domain of a continuous physical system, (such as a part's surface). The true power of FEM lies in its ability to solve problems that do not fit any standard formula. For example, prior to the use of FEM, stress analysis problems were usually matched to a handbooks formula, which was derived for a standard shape part. As the name FEM implies, this method involves the partitioning, or discretization of a structure into a finite number of elements. Elements are connected to one another at their corner points. These corner points are called nodes or nodal points. Each element is a simple geometric shape, such as triangle or quadrilateral. Being a standard shape facilitates the development of the governing equations that relate the displacement and stress behavior within the element. For the completion of a finite element model, nodal points, elements, loads, supports and element related data must be defined. An FEM code formulates the equilibrium equations corresponding to each degree of freedom at each nodal point. The forces on each nodal point are calculated once the element is subject to

loading and begins to deform. Hence, a finite element model, in essence, acts like a large system of springs which deflect until all forces balance. The FE method is well established in the engineering community and there is a multitude of commercial program packages to chose from. The algorithm shown in Fig. 4.10 summarizes the crucial steps of a FEM simulation. While conventional analytic variational methods aim to find a solution by using a single polynomial expression which is valid throughout the whole domain considered, FEM tries to find approximate solutions in each subdomain (each element). The subdomains are connected at joints which are called nodes. The (mostly linear or simple polynomial) interpolation functions used for an approximation, describe the course of the state variable in an element in terms of its node values. Most finite element methods use these functions to map both, the state variable and the topology of the element, which, by definition, is then referred to as an *isoparametric element*. The trial functions of isoparametric elements are thus also referred to as “shape” or “form functions”. The FE method respresents a continuum type approach, that is, it does not incorporate the genuine dynamics of single lattice defects such as dislocations, but rather uses averaging constitutive laws for representing the materials reaction to load. For example, as a two-dimensional isoparametric element with four nodes, the following set of linear shape functions can be used (Fig. 4.10):

$$K_1(\xi, \kappa) = \frac{1}{2}(1 + \xi)(1 + \kappa), \quad (4.64a)$$

$$K_2(\xi, \kappa) = \frac{1}{2}(1 + \xi)(1 - \kappa), \quad (4.64b)$$

$$K_3(\xi, \kappa) = \frac{1}{2}(1 - \xi)(1 + \kappa), \quad (4.64c)$$

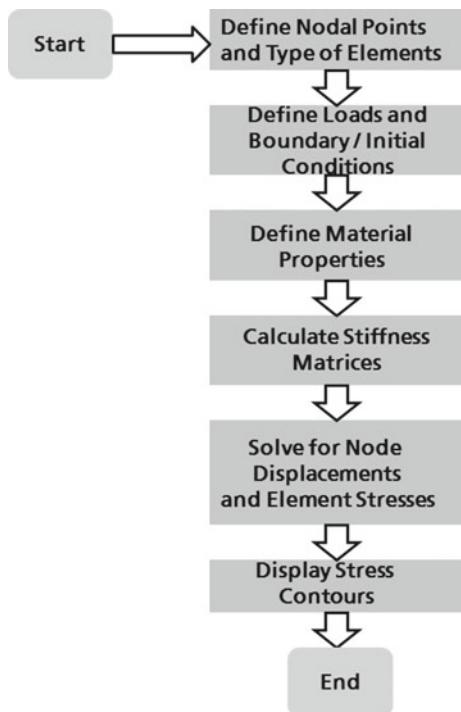
$$K_4(\xi, \kappa) = \frac{1}{2}(1 - \xi)(1 - \kappa), \quad (4.64d)$$

where the variables ξ and κ lie in the range $[-1 \leq \xi]$ and $[\kappa \leq +1]$. The approximated values of the considered field variable in one element may be calculated as:

$$f(\xi, \kappa) = \sum_{i=1}^{n=4} K_i(\xi, \kappa) f_i, \quad (4.65)$$

where n is the number of nodes and f_i are the values of the field variable at the nodes. The application of shape functions to state variables allows one to map each element into a master generic element with a fixed length by using an isoparametric transformation. By taking the derivatives $K_{i,j} = \partial K / \partial x_j$ along x_j of the n ansatz functions in (4.65), one can finally calculate the corresponding stiffness matrix. In Chap. 7 several applications of the FE method are shown. One important difference between FEM and other methods for solving PDEs, such as FDM, is, that FEM approximates the *solution* to a differential equation, whereas FDM approximates the differential equation itself. Sophisticated implementations of FEM use *adaptive finite elements*, i.e. they assess the quality of the mesh during the solution aiming at achieving an approximate solution within some bounds from the exact solution of the continuum problem. The most common mesh-adaptivity methods are:

Fig. 4.10 Basic flow scheme of a FEM analysis



- moving nodes (r-adaptivity),
- refining and elements (h-adaptivity),
- changing order of base functions (p-adaptivity),
- combinations of the above (hp-adaptivity).

4.3 Elements of Software Design

Although leading-edge software-development practice has advanced rapidly in recent years, common practice hasn't. Many programs written in software industry are still very buggy, late and over budget, and many fail to satisfy the needs of their users. On the other hand, many researchers who are active in computer simulation haven't had any formal education in computer science, software design or software engineering, let alone have any professional experience in leading a team of software experts in an industrial large-scale environment, but instead they develop research codes in a "heuristic" process. This explains the abundance of passed on ancient Fortran 77 or Fortran 90 codes for e.g. solving Schrödinger's wave function, performing Monte Carlo simulation runs or for fluid and structural dynamics

applications in engineering; It explains also to a certain degree the common practice in academia that consecutive generations of research students often have to write and invent software packages for similar purposes over and over again. This is an unfortunate situation as researchers in both the software industry and in academic settings have long since discovered effective practices that help to eliminate many programming problems and that makes codes or bits of codes reusable. These practices, when applied to scientific code development, may help considerably not only to reduce the number of bugs in codings and to be more efficient but also to make the research codes more readable and understandable to other scientists and thus allow for concentrating on the scientific parts of computational science and spending less time on implementation details. These practices or different stages in the so-called “software-development life cycle” include:

- Problem definition
- Requirements analysis
- Implementation planning
- High-level design or architecture
- Detailed design
- Implementation
- Integration
- Unit testing
- System testing.

Probably more than 80 % of the time in software development are spend with implementation. Implementation focuses on coding and debugging but also includes some detailed design and some unit testing. It comes after architectural design and before system testing and maintenance within the software-development life cycle. Many researchers – due to a lack in formal training – write their codes in a somewhat “intuitive” style, with not much architectural design, commenting of code, or planning ahead, just going along as certain needs arise. This is probably fine as long as one does not have to meet any specific goals or deadlines in scientific projects, and the code is only used by one person, but often this practice results in “spaghetti-programs” that cannot really be handled by anybody else than the author. A good piece of code should be largely self-explanatory and “readable” just like an ordinary book.

As long as software projects are small (maybe only containing a few thousand lines of code and involving only one or a handful of people working with the code), an extended architectural design would be overdone. However, a large project in software industry will most likely involve many people with many different responsibilities. Writing a parallelized molecular dynamics simulation code can easily extend to more than 30,000 lines of code which – in an industrial setting – would be a medium sized software project. Probably more than 95 % of a research code is only data management, input/output and error handling and only a very small piece of the code does actually contain “science”, i.e. a physical model of the system under

investigation. Large software projects (e.g. writing operating systems) may contain millions of lines of code.

The author of this book has written probably more than 500,000 lines of coding in mostly academic but also in industrial research environments and in software industry during the last 10 years. Scientific research codes usually have a very simple architecture and thus, as a rule of thumb, object oriented methods are way overdone for the (usually) very simple data structures involved. The advantage of object-oriented concepts is the re-usability of pieces of code which is made easy and supported by e.g. container classes and many other implicit language elements; also, combining some piece of code with a GUI is very easy in this case as most object-oriented languages provide appropriate GUI container classes. While the specific choice of programming language has nothing to do with the algorithms that are to be implemented to help solving a scientific problem, it is quite clear, that any software written today by a professional programmer would not be written e.g. using Fortran but rather in some object oriented language such as Java or C++, or even in C which is probably the best alternative to the overdone constructs which are possible in object oriented languages. Java has the advantage of being interpreted during runtime, so that one could change simulation parameters on-the-fly and directly watch the consequences in a Java Applet, instead of going through the process of re-compiling the complete code and visualizing the results using additional programs. However, with scientific programs, usually speed is the most important limiting factor which favors compiling a code into machine language. Most installations of super computers only have C, C++ and Fortran compilers.

A good place to start learning how to manage large software projects is Gilb [299] which focuses on the points a software manager should (be able to) do during the development stage. DeMarco [300] or Maguire [301] are good alternatives.

Maybe the first book on software architecture and design was Yourdon [302] who originally developed many key ideas of a structured design approach to software development. A more modern book that describes object oriented design stages is Booch [303]. He was a pioneer in the use of Ada and in object-oriented methods, and this experience shows in this book.

Testing software is an important part in software-development. Myers [304] is a classic devoted to the topic, Hetzel [305] being a good alternative.

It is sometimes astounding how less knowledge and experience researchers in academia or industry who write simulation codes or who are responsible for managing scientific software projects actually have to show in project planning, computer science, algorithms or data structures. We have discussed some basics of the latter three topics in Sect. 2.6 of Chap. 2. One may consider these things as “unscientific”, however when it comes to scientific computing, the scientific results which are published in specialized physics or engineering journals usually rely heavily and often exclusively on the results of numerical simulations. Some good books on the subject not mentioned in Chap. 2 are Sedgewick [306], Reingold/Hansen [307], Maguire [308], McConnell [309] and of course the classics by Donald Knuth [310–312].

In the next few sections we will touch on some of the most important issues in software development. We start with the first level of software development which is software design.

4.3.1 Software Design

The phrase “software design” means the planning, conception or invention of a scheme for turning a specification for a computer program into an operational program. Design is the activity that links requirements specifications to coding and debugging. It is a heuristic process rather than a deterministic one, requiring creativity and insight. A large part of the design process is appropriate only for a specific project at hand.

On large, formal projects, design is often distinct from other activities such as requirements analysis and coding. It might even be performed by several different people. A really large project is often split into several stages of design, beginning with the software architecture, a high-level module design and detailed implementation design. On small, informal projects, most of the design is done while the program is written. It might just be writing a routine in *program design language* before writing it in some programming-language or it might just involve sketching a few diagrams for a few routines before writing them. In any case, small projects usually benefit from careful design just as larger projects do.

Structured Design

A structured program design is made up of the following parts:

- Strategies for developing designs.
- Criteria for evaluating designs.
- Problem analysis as a guide to the solution to the problem.
- System organization into routines that have a well-defined, narrow interface and whose implementation details are hidden from other routines.
- Graphical tools for expressing design, including structure charts and program design language.

In Fig. 4.11 the main levels (or hierarchies) of code design are displayed which have proved useful in practice. On the first level, a system is split up into subsystems. The main point at this level is to identify all major subsystems and to divide the program into major components with respective interfaces. In a large project, some subsystems might be too big to be directly identified as modules on the first design level.

On a second level, the subsystems are further divided into modules, e.g. input/output functionality, exception handling, etc. As indicated in Fig. 4.11 different design methods might be used within different parts of the system.

On a third design level each module is subdivided into the services offered by the module. This includes identifying routines for different tasks and at the same time specifying respective functions. This level of decomposition and design is needed

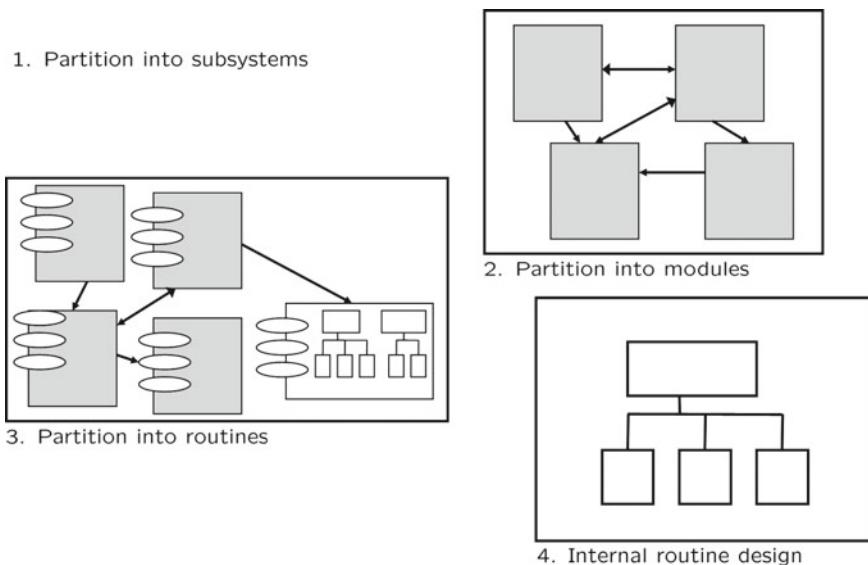


Fig. 4.11 Different software design levels

on any project that takes more time than a few hours. It does not have to be done in a formal process, but it at least needs to be done mentally.

A fourth design level specifies the detailed functionality of the individual routines. In an industrial environment this internal routine design is usually left to the individual programmer working on an individual routine. This level of design is always done in projects of any size, consciously or unconsciously, by every programmer.

In this short discussion we have only introduced a top down approach which consists in stepwise refinement. Sometimes this approach is so abstract that it hard to get started at all. In this case, also a bottom-up approach can be tried which identifies useful routines first and this usually results in a robust design.

Object-Oriented Design

Object-oriented design is characterized by the identification of real-world and abstract objects which are then represented by programming-language objects. It is based on the assumption that the more closely a program models the real-world problem, the better the program will be. In many cases, the data definitions in programs are more stable than functionality, therefore a design based on the data model, as object oriented design, is a more stable approach. This approach involves identifying objects, the operations on these objects and classes and then building a system from those objects, classes and operations. Object-oriented design uses key ideas of modern programming such as:

- Abstraction
- Encapsulation (information hiding)
- Modularity

- Hierarchy and inheritance
- Objects and classes

Object-Oriented Versus Structured Design

With a top-down structured design as depicted in Fig. 4.11 one basically breaks a program into routines. This approach has an *emphasis on functionality* and does not stress data. A simple example of a system based on functionality would be a program that reads in data in batch mode. This is a predictable process on the data which performs in predictable order. An object-oriented design on the other hand is mainly a way of designing *modules* as collections of data and operations on the data. This method is great for decomposing a system from the highest level. It is applicable to any system which acts as objects in the real world do. Examples of such systems are highly interactive programs using windows, dialog boxes, and other constructs. Most of the work being done with object-oriented techniques is focused on the successful implementation of systems from at least 100,000 to multi-million lines of code. Empirically, structured techniques have often failed on such large projects and object-oriented design was developed to specifically allow for error-free programming of very large codes [313]. Figure 4.12 contrasts the two discussed design approaches.

The main difference between structured and object-oriented design is that the latter works well at a higher level of abstraction. If data is likely to change, then the object-oriented design is appropriate because it isolates the data likely to change into individual objects (modules). However, if functionality is likely to change, the object-oriented approach is disadvantageous because the functionality is spread throughout

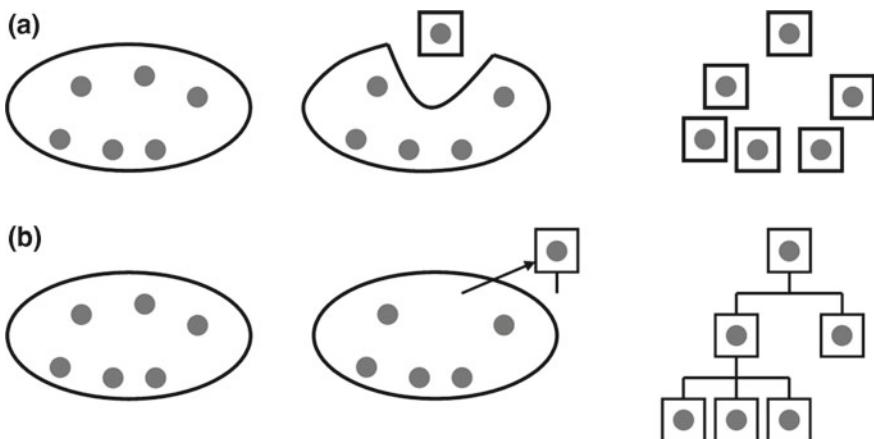


Fig. 4.12 Contrasting the object-oriented approach (a) and the structured design approach (b). The *circles* represent operations or data structures. In **a** objects are identified until no more useful partition into objects can be done. In **b** first the *top level* of code design is identified. Then, consecutively, *lower level* design structures are identified until the whole system has been partitioned

many objects. A structured design approach on the other hand will always be a useful method of choice for all but the largest software projects. In scientific computing, large programs that exceed 50,000 lines of code are extremely rare. In addition, the number of different basic data types is usually extremely small, e.g. properties of particles, atoms, finite elements or fields, compared to, e.g. the thousands of different objects involved in an object-oriented design for controlling a modern aircraft. Thus, in science, in almost all cases a structured approach will be sufficient and there will be no real considerable advantage when using an object-oriented design, simply because the complexity of a scientific code usually is not very large. Examples for this will be provided in later sections.

Next we discuss the core task of programming, the building of a routine.

4.3.2 Writing a Routine

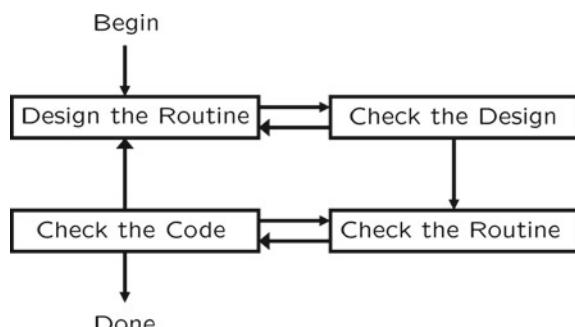
There are many low-level details that go into building a routine which are not done in any particular order, but usually there are four major activities, cf. Fig. 4.13. It is good practice to start a routine by writing down the specification demands in a program design language (PDL) [314] which is roughly the same as ordinary language. An example of a bad description of a routine is given in Algorithm 4.3.2. What is the intend of this block of PDL? Because it is poorly written, it is hard to tell. This piece of PDL is bad because it includes coding details such as

```
*hRsrcPtr
```

Algorithm 4.2 A code example of badly written PDL.

```
1 increment resource number by 1
2 allocate a dig struct using malloc
3 if malloc() returns NULL then return 1
4 invoke OSrsrc_init to initialize a resource for the
   operating system
5 *hRsrcPtr = resource number
6 return 0
```

Fig. 4.13 Various steps in writing a routine



Algorithm 4.3 A code example of improved PDL.

```

1 Keep track of current number of resources in use
2 if another resource is available
3   Allocate a dialog box structure
4   If a dialog box structure could be allocated
5     Note that one more resource is in use
6     Initialize the resource
7     Store the resource number at the location
       provided by the caller
8   Endif
9 Endif
10 Return TRUE if a new resource was created; else
     return FALSE
  
```

in specific C-language pointer notation, and

`malloc()`,

a specific C-language function. This PDL block focuses on how the code is supposed to be written rather than on the meaning of the design. It also gets into coding details - whether a routine returns a 1 or a 0.

Algorithm 4.3.2 shows an example for a much improved PDL version. This PDL is better because it does not use any syntactic elements of the target language and is easy to understand and to transfer into commands of some language.

Some good practice tips in the design of a routine are:

- **Avoiding code duplication.** Probably the most important reason for creating a routine is to avoid duplicate code. With code in *one* place one not only saves space but also modification are much easier done, as one only needs to change code at one location.
- **Limiting effects of changes.** Areas in a code which are likely to change, such as hardware dependencies, input/output handling, complex data structures, etc. should be such designed that the effects of a change are limited to the scope of a single routine or at least not more than a few routines.
- **Hiding data structures.** Routines that hide implementation details of data structures provide a valuable level of abstraction which reduces a program's complexity. They centralize data structure operations in one place and make it easy to change the structure without changing most of the program.
- **Hiding pointer operations.** Pointer operations tend to be hard to read error prone. By isolating them in routines one can concentrate on the intend of the operation rather than the mechanics of pointer manipulation.
- **Avoiding global data structures.** Beginners in programming love global data structures and global variables as they make programming very easy. One never has to pass any variables to routines, one does not have to worry about pointer arith-

metic and one can easily access all variables from all scopes. However, anybody who has ever tried to understand a program that makes extensive use of global variables will always try to avoid them. They make a code completely unreadable and completely destroy any architectural design, i.e. it is impossible to understand the purpose of routines as there is no interface – everything can be globally accessed. Worse, such a code is usually impossible to be extended or optimized, as there is no control as to where some variable might be changed or used in the code. Thus, in any good piece of code there are as less global variables as possible.

- **Saying Farwell to “goto” statements.** The general argument against *gos* is that code without *gos* is higher-quality code. The original controversy on this issue was sparked by Dijkstra [315] in 1968. He observed that the quality of code was inversely proportional to the number of *gos* the programmers used. This *goto* controversy erupted when Fortran was the most popular language, particularly in science. Fortran originally had no presentable loop structures, and in the absence of good advice on programming structured loops (using *for*, *do*, *while* constructs), programmers wrote a lot of spaghetti-code. Such *goto*-laden code was undoubtedly correlated with the production of low-quality programs; however its careful use to make up for a gap in a structured language’s capabilities might have been acceptable in the 1970s. Today, no high-quality code should contain any *gos*. Some of the main arguments against it are:

- Code containing *gos* is hard to format using indentation which should actually represent logical structure.
- Use of *gos* defeats compiler optimizations. Many optimizations depend on a program’s flow of control within a few statements. Unconditional *gos* make the flow difficult to analyze and reduce the ability of the compiler to optimize code.
- Use of *gos* violates structured programming principles.

- **Promoting code reuse.** Programs written in Fortran 77 or Fortran 90 are often written in one monolithic huge file. Writing one huge file with probably 50,000 lines and several subroutines written in the same file makes a code virtually unreadable. This practice should be strictly avoided and instead a program should be given a modular code structure where certain routines that do similar things are collected in one file. Modular routines can be reused in other programs more easily than the same code embedded in one larger routine. A modular structure also makes a program more readable and understandable.

Often it is advisable to hide details of code in a routine which is then called from the main program. This often makes code less complex and automatically documents it. The following piece of code illustrates this:

```
if ( Node <> NULL )
    while ( Node.Next <> NULL ) do
        Node = Node.Next
```

```
LeafName = Node.Name  
else  
    LeafName = '...'
```

These six lines of code could easily be put in a routine as follows:

```
LeafName = GetLeafName ( Node )
```

This new routine is short and all that it needs for documentation is a good name.

How Long Should a Routine Be?

The theoretical best maximum length is often described as one or two screen pages of program listing, about 60 to 120 lines. In Frenkel and Smit [57] the authors say that it is generally assumed that in every 200 lines of code there is one error. This is however not supported by many studies. A study by Basili [316] found that routine size was inversely correlated with errors; as the size of routines increased (more than 200 lines of code), the number of errors per line of code decreased. Card et al. [317] found in a study that small routines of 32 lines of code or fewer were not correlated with lower cost or fault rate. The evidence suggested that larger routines of 65 lines of code or more were cheaper to develop per line of code. Another study (Lind and Vairavan [318]) found that code needed to be changed least when routines averaged 100–150 lines of code. Jones et al. [319] found that routines beyond 500 lines were most error prone and that the error rate tended to be proportional to the size of the routine. A university study of graduate computer science students [320] found that their comprehension of a program that was extremely modularized into routines of about 10 lines of code was no better than their comprehension of a program that had no routines at all. Moreover, when the program was broken into routines of moderate length of about 25 lines, students scored 65 % better on a test of comprehension.

4.3.3 Code-Tuning Strategies

In the early days of programming computer resources were severely limited, and efficiency was the prime concern. As computers became more powerful in the 1970s, programmers realized how much the focus on performance had hurt readability and maintainability, and code tuning received less attention. With the return of performance limitations with the microcomputer revolution of the 1980s, performance issues were considered again. Today, at the high-end are super computer systems with at least hundreds and even thousands of processors which allow for high-performance parallel computing, usually involving a fast network that links the CPUs. As for scientific computing, high-performance and in particular speed are of paramount importance. Therefore usually a lot of time is spent for code tuning in high-end parallelized scientific programs. Generally speaking, code efficiency may have to do with one or several of the following aspects:

- Hardware
- Operating-system interactions
- Code compilation
- Program design
- Module and routine design

One theory for faster and more efficient code is that one should optimize as one writes each routine. This, however is not true. It is generally much better to write a code completely first and then start optimizing when the whole code and all routines run correctly. Before a program is really working completely it is almost impossible to identify any bottlenecks. Thus, the above strategy would lead to many *micro*-optimizations but in the end the whole code might exhibit inferior performance. Focusing too much on optimization during initial development tends to detract from achieving other program objectives, such as correctness, modularity, information hiding and readability. Optimizations done after a system is complete can identify each problem area and its relative importance so that optimization time is allocated effectively.

The Pareto Principle

This principle, also known as the 80/20 rule, states that you can get 80 % of the result with 20 % of the effort. This principle applies to many other areas than programming, but it does definitely apply to program optimization.

Knuth [321] found in an empirical study that less than 4 % of a program usually accounts for more than 50 % of its run time. Thus, since small parts of a program usually take a disproportionate share of run time, one should always measure the code to find the hot spots and then put one's resources into optimizing the few percent that are used most. An example of this can be seen in Fig. 4.14 which shows the profiling output of one of the authors molecular dynamics programs. The total run-time was

%	cumulative	self		self	total		
time	seconds	seconds		calls	ms/call	ms/call	name
34.14	28.15	28.15		10001	2.81	3.65	ForceCalcGhosts
18.47	43.38	15.23	229175856		0.00	0.00	ran2
11.84	53.14	9.76		957	10.20	10.20	BuildNTabGhosts
10.39	61.71	8.57	180000000		0.00	0.00	gasdev
8.59	68.80	7.08		10000	0.71	8.23	IntOneStep
6.93	74.51	5.71	39762734		0.00	0.00	LcCalcGhosts
3.29	77.22	2.71		10001	0.27	0.27	SummonGhosts
3.24	79.89	2.67	24002400		0.00	0.00	FeneCalcGhosts
1.46	81.09	1.20		10000	0.12	0.12	UpdateGCoords
1.27	82.14	1.05		957	1.10	1.10	VerletSetCoords
0.22	82.32	0.18		957	0.19	0.19	SortIntoCells

Fig. 4.14 Output of a profiler *gprof* on a UNIX system for one of the author's molecular dynamics simulation codes

84 seconds and a total of 51 different functions are called in the program. One can see that less than 4 % of the program accounts for more than 50 % of total run time. In this case, a Langevin-thermostat was simulated which uses random numbers. There are no general-purpose rules to optimization as the rules change each time one changes languages, hardware, compilers, or even compiler versions. This means that techniques which improve performance in *one* environment can degrade in others. Often optimizations are counter-intuitive. A simple example for this is the piece of pseudo-code given in Algorithm 4.3.3.

Algorithm 4.4 Example of a straightforward piece of code for the summation of matrix elements.

```

1 CalcSum = 0;
2 for ( Row = 0; Row < RowCount; Row++ )
3 {
4   for ( Coulomn = 0; Column < ColumnCount; Column++ )
5   {
6     CalcSum += Matrix[ Row ][ Column ];
7   }
8 }
```

This code sums up matrix elements. Generally speaking, it is always advisable to only use *one*-dimensional arrays, as the computer internally translates arrays into one-dimensional ones anyway. But this is actually not the point here. In an attempt to optimize this matrix element addition which – for a 10×10 matrix consists of 100 multiplications and additions plus the loop overhead for incrementing the loop counter, one might try out pointer notation. Converting Algorithm 4.4 into pointer notation with only 100 relatively cheap pointer increment operations, one obtains the piece of code displayed in Algorithm 4.5.

Algorithm 4.5 Example of an optimization strategy for the summation of matrix elements using pointers.

```

1 CalcSum = 0;
2 ElementPointer      = Matrix;
3 LastElementPointer = Matrix[ RowCount - 1 ][
   ColumnCount - 1 ] + 1;
4 while ( ElementPointer < LastElementPointer )
5 {
6   Sum += *ElementPointer++;
7 }
```

When testing this program's performance in C compiled with a standard C-compiler on a UNIX system, it might come as a surprise that there is no gain in speed at all compared to the un-optimized version. The reason is that the compiler's

optimizer already optimizes the first code version well enough. Thus, all that was gained here, would be turning a readable code into a harder to read code with no measurable gain in speed. Probably the best way to prepare for code tuning at the initial coding stage is to write clean code that is easy to understand and modify. As a summary of the short discussion of code tuning, we give some key points of an optimization strategy:

1. Use a good modular program design that is easy to understand and modify.
2. If performance is poor and the need arises, measure the systems performance to find hot spots.
3. Evaluate whether the performance comes from poor design, inadequate data structures or algorithms. If this is the case go back to step 1.
4. Tune the bottleneck identified in step 3 and then measure each improvement; if it does not improve, take it out.
5. Repeat steps 2–5.

4.3.4 Suggested Reading

The discussion of structured code design has been necessarily short and superficial here; in particular a graphical representation of relations is very powerful. An excellent book to start with is Yourdon [302] which is written with obvious care. Another excellent book which is succinct, to the point and less technical is Myers [304]. The original article on structured design by Stevens et. al. [322] is a classic although many books today make a better presentation of the material. The fine volume by Meyer [323] and Coad [324] discusses virtually all aspects of object-oriented design, with the latter volume being a probably easier introduction to the subject. The classic article by Parnas [325] describes the gap between how programs are often really designed and how one wished they were designed; the main point of this book is that no one ever goes through a rational, orderly design process but that aiming for it makes for better designs in the end. The very readable book by Jackson [326] explains the full data-view design methodology. A discussion of creative thought process in programming can be found in Adams [327]. A very readable book by Simon [328] discusses the “science of design”. A discussion of quality considerations during all phases of software development is provided in Glass [329]. Some good books on performance engineering are Smith [330] which includes many examples and case studies or Bentley [331] which is an expert treatment of code tuning and optimization. Weinberg [332] is probably one of the most readable books on software development. This classic also contains many entertaining anecdotes about the human side of programming.

Problems

4.1 Plane waves as solution of the wave equation

Show that any function of the form $f(x + ct)$ or $f(x - ct)$ is a solution of the wave equation (4.30).

4.2 Spherical waves as solution of the wave equation

Show that a spherically symmetric solution of the wave function (4.30) is of the form $\Phi_{\pm} = \frac{A_{\pm}}{r} e^{i(kx \pm \omega t)}$.

4.3 Verlet algorithm

Derive the Verlet algorithm by Taylor expansion of the coordinates r and show that it is of order $\mathcal{O}(\Delta t^2)$.

Part II

Computational Methods on Multiscales

Summary of Part I

Before the works of Kurt Gödel, the idea of a complete formalization and axiomatization of all mathematics, introduced by David Hilbert at the beginning twentieth century (Hilbert's program), was predominant. Mathematics was supposed to be described within the framework of a “logical calculus,” a formal, non-contradictory, and closed system of axioms. The basic idea of this formalization was to derive all mathematical theorems, which are recognized as true, by logical deduction from some prefixed axioms. In essence, this means that all true mathematical theorems can be generated by a systematic, algorithmic process.

In the 1930s, Kurt Gödel managed to show that Hilbert's idea was doomed by proving that any formal logical calculus, which is sufficiently complex, i.e., which contains theorems on the natural numbers (such as number theory), contains an infinite number of theorems which can neither be proved nor disproved within this logical system. Such theorems are called *undecidable*. In other words, Gödel managed to prove that any sufficiently complex logical system based on a set of axioms is incomplete, and he showed this explicitly for the axioms of number theory. As a consequence, for example, there exists no algorithm, which is capable of solving any complex mathematical problem, and there is no algorithm that can be used to proof the correct functioning of a different algorithm.

Roughly during the same period of time, Allan Turing provided a simple definition of a machine, called Turing machine, that can execute an algorithm. The concept of a Turing machine was developed in order to provide a fundamental notion of algorithmic computability. Despite its simple construction, a Turing machine is universal, that is, any intuitively calculable function, i.e., a function that can be solved using an algorithm, can be calculated by a Turing machine. This is underpinned by Church's thesis which states that the Turing machine is an adequate model for computability that includes all other possible models.

The formal developments in mathematical logic and the formalization of a notion of algorithm and computability were accompanied by two major physical theories, discovered and formulated during the first quarter of twentieth century: quantum theory and the theory of relativity. Based on the duality of particle and wave concepts as fundamental principles of physical theory, quantum theory changed profoundly

the general notion of what is to be considered reality in physical theories. With the extension of quantum theory to relativity theory and to the quantization of fields, along with the application of symmetry principles of group theory, it became possible to describe three of the four known fundamental interactions within one theoretical system. The mathematical ontology of modern physics is thus based on symmetries, fields, and particles.

On the other hand, with the discovery of the theory of special relativity, it became evident, that the distinctive status of inertial systems is not only valid for the mechanical equations of Newton's theory, but for all physical processes. The mathematical framework which is used in this theory is based on the invariance of the space-time interval in Minkowski space for all observers in inertial systems. Each observer uses his own system of synchronized clocks to measure time and length intervals. The theory of general relativity extends this invariance of a general line element to observers, which are not in an inertial frame of reference. This is a major difference to pre-relativistic Newtonian theory according to which there exists a global coordinate time which is the same for all observers. As a consequence of this invariance principle, one has to formulate physical theories in tensorial form which ensures their invariance against arbitrary nonlinear transformations between the frames of reference. This naturally leads to the use of curvilinear coordinate systems and to a concept of non-Euclidean curved space-time in which the components of the fundamental metric tensor are coordinate dependent. Einstein's field equations of general relativity theory (GRT) provide the connection between the sources of mass and electromagnetic fields given in the energy-momentum tensor, and the metric tensor which is a tensor of rank 2. The differential geometric concepts on which GRT is based were developed in mid-nineteenth century, by Carl Friedrich Gauss, Janos Bolyai, and N. Iwanowitsch Lobatschewski, and were finally generalized to a theory of manifolds by Bernhard Riemann in his inaugural speech in 1856, when he introduced a generalized concept of space, which allowed for a separate treatment of the underlying space (topology) and its geometry. This concept was later further formalized using basic set theory. These mathematical formalizations allow for a deeper understanding of different concepts, or levels of space, depending on the amount of algebraic structure that is provided on the respective level. In this sense, relativity theory is a framework theory which provides the basic spatio-temporal structure for the natural sciences, and theories formulated using this space-time structure in Riemannian geometry are considered to be more fundamental than those that are not.

With the development of the first electronic computers in the mid-1940s, a new way of performing scientific research became accessible, namely the use of computers to investigate in detail the behavior of N -particle systems, with differential equations too complex to be solved analytically. This development of computational science has not come to an end, and the scope of possible applications is steadily increasing with the discovery of new efficient algorithms and the steady increase of hardware capabilities.

Abstract

After an introduction in Sect. 5.1 we begin with a discussion of basic ab initio methods in Sect. 5.2, which are designed to solve the many-particle Schrödinger equation. In Sect. 5.3 we shortly discuss some basics of quantum theory, the Hilbert space and the Born–Oppenheimer approximation, which separates the motion (i.e. the coordinates) of the heavy nuclei from the fast motion of the light electrons. We take a look at density functional theory in Sect. 5.4 and study Car–Parinello MD in Sect. 5.5. In Sect. 5.6 we learn about the Hartree–Fock approximation which in essence reduces the problem of solving the N -particle Schrödinger equation to the problem of solving an effective one-particle Schrödinger equation, with the interaction between the particles taken into account on average (mean-field approach). We derive the Hartree–Fock and Hartree–Roothaan equations and also discuss shortly perturbation methods. In Sect. 5.7 we look at the different forms of chemical bonding that play a role on a phenomenological level (above the scale of sub-atomic elementary particles). In Sects. 5.8 and 5.9 we take a look at semi-empirical methods and the chapter ends with some concluding remarks in Sect. 5.10.

5.1 Introduction

Classical Newtonian physics fails to describe natural phenomena within the realms of atoms ($<10^{-9}$ m), at high velocities and also when investigating the interaction of matter with electromagnetic radiation. For high velocities, the classical Newtonian theory of mechanics of mass points has to be replaced by special relativistic mechanics and in the case of radiation and atoms the classical theory has to be replaced by quantum theory.

In this chapter we are going to discuss computational methods for simulating materials on an atomic length scale. During the last decades, numerical ab initio electronic structure calculations have become possible as a tool for investigating the structural, dynamic and electronic properties of complex molecules, clusters, polymers, and solids on the atomic level. Chemical and physical atomic processes nowadays may be simulated on a computer. A visible sign of these developments can be seen in the 1998 Nobel prize of Chemistry award to the quantum chemist J.A. Pople and to the theoretical physicist W. Kohn.

Dealing with the atoms that constitute a material means that one has to use a theory which correctly describes their dynamics. This theory was gradually established during the first decades in the 20th century by several physicists and culminated in the formulation of a final form of the theory in 1926 by Max Born, Werner Heisenberg and Pascual Jordan [333]. This formulation was based on matrix calculations. Hardly a year later, Erwin Schrödinger managed to derive his wave equation, a partial differential equation of second order¹ for the wave function $\Psi(\mathbf{x}, t)$ which bears his name. He was able to show, that his equation was mathematically equivalent to the formulation of the theory derived in the “three-man paper” earlier. This shows that the two formulations are only different manifestations of the same, underlying abstract mathematical theory. This theory was first formulated by Paul A.M. Dirac. Today, in modern terminology, one speaks of the *Heisenberg picture*, the *Schrödinger picture* and the *Dirac picture*, depending on which formulation of the theory one prefers to make use of. Generally, in modern textbooks of quantum mechanics, the theory is preferably presented in Dirac formulation as this allows to appreciate the underlying structure of it and also allows to introduce the theory in an axiomatic manner. The mathematical foundations of quantum mechanics are presented using an axiomatic approach, simply introducing the elements of the theory in the mathematical framework of (linear) Hilbert space. Beginning with Sect. 5.2 several widely used computational methods on the atomic level are introduced and discussed.

The world exhibits many levels of scale and varying complexity which require very different descriptions. The classical and quantum world are two such levels. In principle, one can agree that all systems, no matter how complex, are made up of subatomic elementary particles. However, this does not a priori guarantee that the methods and concepts that work on the smallest scales can equally well be applied to all systems. Hence, the believed universal applicability of quantum theory merely means that every physical system can in principle be decomposed into its atomic (or subatomic) constituents, which can all be represented by appropriately chosen quantum states. However, this does not imply that a macroscopic object as such can be usefully represented in this way. At least so far, there is no universally unified theory that accounts for all levels of complexity in a practical manner.

¹Actually, Schrödinger at first had derived an equation that was of first order in time – the Klein-Gordon equation – but then had dismissed it because it didn't reproduce the correct energy levels. The reason for this was that the electron spin was still unknown and not included in the theory.

5.2 Ab-Initio Methods

Atomistic simulations and in particular numerical electronic structure calculations have become increasingly important in the fields of physics and chemistry during the past decade – especially because of the advent of easily accessible high-performance computers. Provided that the types of atoms of which a given material is made of, are known, a computational approach on the level of atoms allows for answering two fundamental questions:

- What is the electronic structure of the material?
- What is its atomic structure?

The ultimate goal of atomistic calculations in materials science is to virtually “design” materials with predefined properties by changing their specific atomic bonding and electronic structure. A whole host of theoretical methods were developed to address the above questions. Nowadays, chemicals and molecules that are of interest in research and technology, including drugs, are often designed first on the computer using molecular orbital (MO) calculations (Molecular Modeling), before the molecules are really made. In MO calculations, the Hartree–Fock (HF) approximation is often used to describe the electron-electron interaction effects which excludes the correlation effects. The *local density* approximation is a much better approximation for ground state properties, but cannot be applied to excited and highly correlated states. The numerical methods for computing structural properties of materials may be divided into two classes:

1. Methods that use empirically, i.e. experimentally determined parameters, and
2. Methods that do not make use of any empirical quantities.

The former methods are often called “empirical” or “semi-empirical”, whereas the latter are called “ab initio” or “first-principles methods”. Several tight-binding schemes fall into the former category. The idea of these keywords is to treat many-atom systems as many-body systems composed of electrons and nuclei, and to treat everything on the basis of first principles of quantum mechanics, without introducing any empirical parameters from measurements. Ab initio methods are often useful for predicting the electronic properties of new materials, clusters or surfaces, and for predicting trends across a wide range of materials. We note, that even with the most sophisticated methods of modern physics and quantum chemistry, one still has to rely on some approximative theories on which ab initio methods are based (e.g. the Born–Oppenheimer approximation). It is remarkable however, that the fundamental principles on the atomic scale which determine all structural and functional properties of materials, are very simple:

1. The atom nuclei, due to their large mass compared to electrons ($m_N \approx 1836 m_e$), can usually be treated as classical particles with given mass and positive charge.
2. The electrons are spin 1/2 particle that obey the Pauli principle and Fermi–Dirac statistics.

3. The only relevant interaction is electrodynamics and the repulsion and attraction of charges is given by Coulomb's law.

Based on these principles one can treat the interaction between atoms in solids or small molecules using quantum mechanics. As a result one can predict structural and electronic properties of materials with high precision. One of the main differences between classical (empirical potential methods) and quantum mechanical methods is that in the former approach the potential energy between atoms is described by an analytical function of atom coordinates, whereas in the majority of the former methods, the energy is calculated by solving the Schrödinger equation for electrons in the field of many atom cores. Unfortunately, N -body quantum mechanical equations cannot be solved exactly, due to the astronomical number of degrees of freedom and the resulting complexity of the electromagnetic interaction between the electrons and atoms. As the repulsive electromagnetic interaction between electrons is infinite in range, the motion of each single electron depends on the motion of all others and the numerical effort for diagonalizing the Eigenwert equation scales with N^3 ; thus, the exact many-body Hamiltonian operator cannot be calculated and is usually replaced with a parameterized Hamiltonian matrix,² the parameters of which are chosen in such a way as to reproduce some reference system. For example, chemists are often interested in calculating the structure and properties of very small molecules with very high accuracy using quantum chemistry techniques, whereas physicists often treat the problem of a periodic crystalline solid, where some simplifications due to the long range crystal symmetry are possible. Very accurate methods that have been developed for small systems usually cannot readily be applied to large systems composed of many atoms. Computational physics on the atomic scale is thus always a compromise between the desired accuracy and the possible system size that can still be treated³ and an understanding of the limitations of different methods is necessary for making a useful choice. For example, the effort for self-consistent solutions of the Schrödinger equation grows with $\mathcal{O}(N^3)$, whereas for semi-empirical atomistic models based on pair potentials, the effort is $\sim\mathcal{O}(N)$.

Figure 5.1 exhibits several common methods employed for atomic scale simulations along with their estimated maximum system size and the typical time scales that can be treated. The highest precision and transferability of results of methods is achieved with self-consistent first principles calculations. Self-Consistent Field Theory (SCF) is fundamentally based on the Hartree–Fock (HF) method which itself uses the mean field approximation (MFA); due to MFA, the energies in HF calculations

²For example, the Schrödinger equation as a partial differential equation with proper boundary conditions can in principle be solved “by brute force” when introducing a finite-difference differential operator on a grid. Considering only a single atom of size $\sim 1 \text{ \AA}$, the grid should be fine enough to resolve the motion of this atom, say at least 10 \AA in each direction with a step size of at least 0.1 \AA . Thus, the total number of grid points to be considered is $100^3 = 10^6$, and the number of matrix elements to be stored is 10^{12} . Using double precision variables, this would amount to 8TByte of required storage capacity just for the dynamics of electrons in one single atom.

³Inappropriately small system for example can also result in systematic errors.

are always larger than the exact energy values. The second approximation in HF calculations is that the wave function has to be expressed in functional form; such functionals are only known exactly for very few one-electron systems and therefore, usually some approximate functions are used instead. The approximative basis wave functions which are used are either plain waves, Slater type orbitals $\sim \exp(-ax)$ (STO), or Gaussian type orbitals $\sim \exp(-ax^2)$ (GTO). Correlations are treated with Møller–Plesett perturbation theory (MPn), where n is the order of correction, Configuration Interaction (CI), Coupled Cluster theory (CP), or other methods. As a whole, these methods are referred to as *correlated calculations*.

Density Functional Theory (DFT) is an alternative ab initio method to SCF. In DFT, the total energy of a system is not expressed in terms of a wave function but rather in terms of an approximative Hamiltonian and thus an approximative total electron density. This method uses GTO potentials or plane waves as basis sets and correlations are treated with Local Density Approximations (LDA) and corrections.

So-called *semi-empirical methods* such as Tight Binding (TB) (e.g. Porezag et al. or Pettifort) approximate the Hamiltonian \mathcal{H} used in HF calculations by approximating or neglecting several terms (called Slater–Koster approximation), but re-parameterizing other parts of \mathcal{H} in a way so as to yield the best possible agreement with experiments or ab initio simulations. In the simplest TB version, the Coulomb repulsion between electrons is neglected. Thus, in this approximation, there exists no correlation problem, but there is also no self-consistent procedure.

Classical multi-body potentials of Tersoff or Stillinger–Weber type, and two-body potentials, e.g. in the *Embedded Atom Method* (EAM) or generic Lennard–Jones (LJ) potentials allow for a very efficient force field calculation, but hardly can be used for systems other than the ones for which the potentials were introduced, e.g. systems with other types of bondings or atoms. Classical methods are however very important to overcome the complexities of some materials. They can be used with parameterized potentials (e.g. Finnis–Sinclair) obtained from experiment or ab initio calculations. Although the border line between the different methods displayed in Fig. 5.1 cannot be sharply defined, they can principally be divided into the following groups:

1. Hartree–Fock (HF). This method scales $\sim \mathcal{O}(N^4)$, depending on how the correlations are treated.
2. Density Functional (DF). This method scales $\sim N^3$ ($\sim N$ for *some* problems).
3. Tight-Binding (TB). This method scales $\sim N^3$ ($\sim N$ for *some* problems).
4. Quantum Monte Carlo (QMC). This method scales nearly exponentially (depending on the number N of particles and temperature).

All of the above schemes, except for TB, are ab initio methods. In practice, the scaling of the computational effort does not depend on N but rather on the number n_b of basis functions; however, one can assume that $N \sim n_b$. QMC methods pertain to the area of quantum chemistry, and thus only small molecules can be simulated.

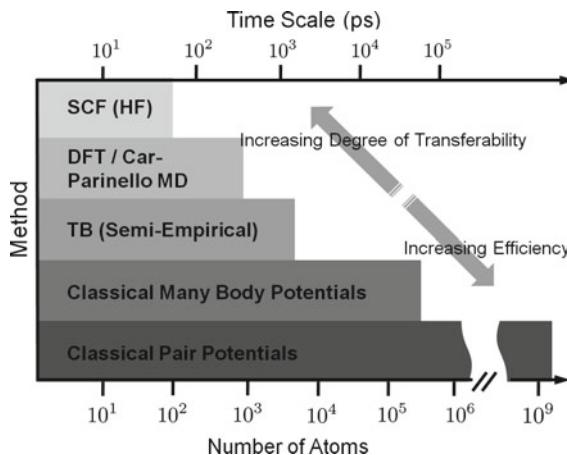


Fig. 5.1 Different atomistic methods used in materials science, along with their corresponding level of transferability, and a rough estimate of the number of atoms that can be simulated within a couple of days on present day supercomputing systems. Note that the available size of the systems can be increased considerably by using domain decomposition on parallel computing facilities, however the time scale cannot be “decomposed” and distributed on many CPUs. Therefore, the available time scale with atomic type simulations using empirical pair potentials is roughly a few hundred nanoseconds for the longest runs on the largest super computers, see also Fig. 1.11 on p. 27. SCF and DFT methods calculations yield the electronic structure of atoms and molecules, whereas calculations with Tight Binding and Car-Parinello MD yield the trajectories of a few hundred atoms for a few picoseconds. Semi-empirical methods such as EAM, bond order type (Tersoff) or cluster type (Stillinger-Weber) potentials, typically are used to obtain structural and energetic properties of clusters, crystals, surfaces or defects. Classical force field MD is often used for calculating phase transitions, diffusion, growth phenomena or surface processes

5.3 Physical Foundations of Quantum Theory

The date of birth of Quantum Theory can be given by exact day. It was in a presentation on 14th December 1900 given by Max-Planck at one of the fortnightly meetings of the German Physical Society in Berlin when the term *quantum of action* (henceforth denoted by \hbar) was first used within the realms of physics. Planck had been in search of the radiation law of heated bodies for more than 6 years at that time. During this search he had found his famous interpolation formula which accounted for all experimentally known facts about the energy density $u(\omega, T)$ of a black body radiator at that time. The formula which Planck had derived “in an act of despair” reads:

$$u(\omega, T) = \frac{\hbar\omega^3}{\pi^2 c^2} \frac{1}{\exp[\frac{\hbar\omega}{kT}] - 1}. \quad (5.1)$$

This formula describes the functional form of the behavior of radiation, emitted by bodies at different temperatures. As one can easily see in Eq. 5.1, this law does not

depend on the specific material chosen - there are no material parameters needed to correctly describe the observed radiation. Thus, it is a *universal* law. Albert Einstein re-derived this formula 16 years later by anticipating the principle according to which a laser functions. A short discussion of Eq. 5.1 yields the known Rayleigh-Jeans law for low frequencies ($\hbar\omega \ll kT$) and Wien's law for high frequencies, thus this formula interpolates between these limiting cases and is valid for the whole frequency spectrum. In an attempt to interpret (5.1), Planck had to assume that energy is only exchanged with the black body in discrete portions, which he originally called "Energieelemente" (energy elements) [3]. This was the birth of quantum theory.

Objects in everyday life differ from quantum objects in that the former are "large" and the latter are "small". The point is, that macroscopic objects are not simply blown-up microscopic objects, but they are composed of many microscopic constituents. Larger objects on the macroscopic length scale simply do not contain larger but *more* atoms; they are *many-atom*, or many particle systems. We will not spend our time with repeating all the introductory chapters of some of the many excellent text books on quantum theory, such as those by Landau and Lifshitz [334], Dirac [335], Messiah [336], Sakurai [337], or Schiff [338], but instead, in the next section the mathematical structure and fundamentals are very succinctly stated in axiomatic form.

5.3.1 A Short Historical Account of Quantum Theory

At the origin of quantum mechanics there was the so-called "old quantum theory" which started with the introduction of the quantum of action by Max Planck in 1900. Niels Bohr's model of atomic structure of 1913 [132, 339, 340] which was extended by Arnold Sommerfeld (1868–1951) in 1915 to what was called the Bohr–Sommerfeld theory, today is only of historical importance. In 1923 Born and Heisenberg could show that the Bohr–Sommerfeld theory already fails to explain the energy states of the two-electron atom helium. In their conclusions, they state that either the quantum conditions are wrong, or the electron motions do not, even in the stationary states, satisfy the mechanical equations [341]. This crisis of the "old quantum theory" aggravated in the following two years. In spring 1925 Heisenberg finally achieved the breakthrough to the modern formulation of quantum theory by means of non-commuting dynamic variables (in modern terminology called the *Heisenberg picture*) in his publication "Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen" [342]. Following this paper, Dirac worked out a complete mathematical formulation of quantum mechanics based on quantities which he called "q-numbers". This theory, "The fundamental equations of quantum mechanics" [343], was published in 1925 and contains the essence of the "bra" and "ket" formulation in terms of elements, i.e. vectors of a linear Hilbert space, which is nowadays found almost exclusively in modern textbooks on quantum mechanics. Schrödinger, working independently, published another formulation

of quantum theory [344–347]⁴ based on wave mechanics and he introduced the ψ -wavefunction into the theory, the absolute square $|\psi|^2$ of which was interpreted as the probability of experimental outcomes by Max Born [349].⁵ The approaches of Heisenberg and Schrödinger were soon shown to be equivalent. In 1932, the mathematician John von Neumann put quantum theory on a mathematically more rigorous axiomatic foundation in his influential classic “The mathematical foundations of quantum mechanics” [350].

We consider the pure state of a single physical system and observables with discrete, non-degenerate spectra. Quantum systems are usually microscopic, i.e. very small, but size alone is not a sufficient criterion. One could regard the quantum world as where classical mechanics fails. This categorization puts a tentative limit on the validity of quantum mechanics; there is yet no evidence that this theory really applies directly to classical objects. The limit however is fuzzy and there does not seem to be any satisfactory criterion that separates quantum systems from classical systems.

One can formulate the axioms of quantum mechanics in the following way, see e.g. standard textbooks, e.g. Sakurai [4, 337] or Landau and Lifshitz [334]:

1. A state of a quantum mechanical system is represented by a unit vector $|\psi\rangle$ in Hilbert space \mathcal{H} .
2. The temporal evolution of the state is governed by the Schrödinger equation: $i\hbar \frac{\partial |\psi\rangle}{\partial t} = \mathcal{H}|\psi\rangle$.
3. An observable associated with a quantum system is represented by a unique self-adjoint operator A on its Hilbert space. The spectrum of the operator comprises all possible values that can be found if the observable is measured.

The topic of quantum mechanics is an /isolated system, whose interactions with the rest of the world, including instruments for its measurement, are neglected. A *quantum state* is the complete and maximal summary of the characteristics of the quantum system at a moment in time. The collection of all states permissible for a quantum system is theoretically represented by its *state space*, which is a complex Hilbert space.

Pre-Hilbert Spaces, Hilbert Spaces

It is most straightforward to start with a definition:

Definition 6 (Pre-Hilbert Space) A *pre-Hilbert space* is a possibly infinite-dimensional linear, complex vector space *b*f V , endowed with a scalar product

$$\langle , \rangle : \mathbf{V} \times \mathbf{V} \rightarrow \mathbf{C} \quad (5.2)$$

⁴For an English translation of most “historical” articles on quantum theory, see e.g. [348].

⁵Max Born received a Nobel Prize in 1952 for his statistical interpretation of $|\psi|^2$.

of two variables on \mathbf{V} . Equation (5.2) is called a **hermitian inner product** if it holds the following properties:

$$\begin{aligned}\langle x, y \rangle &= \langle y, x \rangle^* \text{ (hermitian-symmetry property)} \\ \langle x + x', y \rangle &= \langle x, y \rangle + \langle x', y \rangle \text{ (additivity in first argument)} \\ \langle x, y + y' \rangle &= \langle x, y \rangle + \langle x, y' \rangle \text{ (additivity in second argument)} \\ \langle x, y \rangle &\geq 0 \text{ (positivity)} \\ \langle \alpha x, y \rangle &= \alpha \langle x, y \rangle \text{ (linearity in first argument)} \\ \langle x, \alpha y \rangle &= \alpha^* \langle x, y \rangle \text{ (conjugate-linearity in second argument)}\end{aligned}\tag{5.3}$$

The associated norm on \mathbf{V} is defined by

$$|x| = \langle x, x \rangle^{1/2}\tag{5.4}$$

with the non-negative square-root. Sometimes such spaces \mathbf{V} are called *inner product spaces* or *hermitian inner product spaces*.

Example 35 (l^2 Spaces) In quantum mechanics, there is actually just one *infinite-dimensional* but countable⁶ Hilbert space occurring in practice, namely the space l^2 constructed as follows: Let l^2 be the collection of sequences $a = \{a_i : 1 \leq i < \infty\}$ of complex numbers meeting the constraint

$$\sum_{i=1}^{\infty} |a_i|^2 < +\infty.\tag{5.5}$$

For two such sequences $a = \{a_i\}$ and $b = \{b_i\}$, the inner product is

$$\langle a, b \rangle = \sum_{i=1}^{\infty} a_i b_i^*.\tag{5.6}$$

This construction can be generalized by replacing the countable set $\{a_i\}$ by an arbitrary set A . Let A be an arbitrary index set, and let $l^2(A)$ be the collection of complex-valued functions f on A such that

$$\sum_{\alpha} \langle f(\alpha), g(\alpha) \rangle^2 < +\infty.\tag{5.7}$$

For two such functions f, g is defined an inner product, by

$$\langle f, g \rangle = \sum_{\alpha} f(\alpha) g(\alpha)^*.\tag{5.8}$$

⁶Most infinite-dimensional Hilbert spaces occurring in practice have a countable dense subset, and this itself is because the Hilbert spaces are completions of spaces of continuous functions on topological spaces with a countable basis to the topology.

Definition 7 (Hilbert Spaces) If a pre-Hilbert space is *complete* with respect to the metric arising from its inner product (and norm), then it is called a **Hilbert Space**. An arbitrary pre-Hilbert space can be completed as metric space. Since metric spaces have countable local basis for their topology (e.g. open balls of radii $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \dots$) all points in the completion are limits of Cauchy sequences. The completion inherits a hermitian inner product defined by a limiting process

$$\langle \lim_m x_m, \lim_n y_n \rangle = \lim_{m,n} \langle x_m, y_n \rangle . \quad (5.9)$$

5.3.2 A Hamiltonian for a Condensed Matter System

The many particle non-relativistic Hamiltonian of a system composed of k electrons of charge $-e$ with mass m and l atomic nuclei (N) of mass M may be written as:

$$\mathcal{H} = \mathcal{K}_N(\mathbf{R}) + \mathcal{K}_e(\mathbf{r}) + \mathcal{P}_{Ne}(\mathbf{R}, \mathbf{r}) + \mathcal{P}_{NN}(\mathbf{R}) + \mathcal{P}_{ee}(\mathbf{r}), \quad (5.10)$$

where the operators of kinetic (\mathcal{K}) and potential (\mathcal{P}) energy are given by:

$$\mathcal{K}_N(\mathbf{R}) = -\hbar^2 \sum_I \frac{\nabla_I^2}{2M_I} \text{ (kinetic energy of the nuclei)}, \quad (5.11a)$$

$$\mathcal{K}_e(\mathbf{r}) = -\hbar^2 \sum_I \frac{\nabla_i^2}{2m_e} \text{ (kinetic energy of the electrons)}, \quad (5.11b)$$

$$\mathcal{P}_{Ne}(\mathbf{R}, \mathbf{r}) = -\frac{1}{2} \sum_{I,i} \frac{Z_I e^2}{R_{ii}} \quad (5.11c)$$

(potential energy of the electron-nucleus interaction),

$$\mathcal{P}_{NN}(\mathbf{R}) = \frac{1}{2} \sum_{I \neq J} \frac{Z_I Z_J e^2}{R_{IJ}}, \quad (5.11d)$$

(potential energy of the nucleus-nucleus interaction),

$$\mathcal{P}_{ee}(\mathbf{r}) = \frac{1}{2} \sum_{i \neq j} \frac{e^2}{R_{ij}}, \quad (5.11e)$$

(potential energy of the electron-electron interaction). (5.11f)

The integers Z_I and Z_J in (5.11c) and (5.11e) are the charges of the nuclei.

The many particle wave function $\Psi = \Psi(r_1, r_2, \dots, r_k, R_1, R_2, \dots, R_l)$ depends on both, the electron coordinates $\mathbf{r} = \{r_1, r_2, \dots, r_k\}$ and the coordinates of the nuclei $\mathbf{R} = \{R_1, R_2, \dots, R_l\}$.

5.3.3 The Born–Oppenheimer Approximation

In the vast majority of ab initio molecular dynamics methods, the *Born–Oppenheimer approximation* (BOA) is used which is one the most important approximations in

materials science. This approximation is based on the large mass difference between electrons and atom nuclei.⁷ Due to their large weight, the atomic nuclei move relatively slowly and can be treated as stationary, while the electrons move fast in the field of the nuclei,⁸ i.e. they follow the motion of the nuclei adiabatically. The lattice oscillations of electrons and nuclei differ from each other by several orders of magnitude. Thus, one can separate the motion of electrons and nuclei and treat them independently. The wave function can accordingly be separated as follows:

$$\Psi(\{\mathbf{R}_I\}, \{\mathbf{r}_i\}) \approx \underbrace{\chi(\{\mathbf{R}_I\})}_{\text{nuclei}} \cdot \underbrace{\psi(\{\mathbf{R}_I\}, \{\mathbf{r}_i\})}_{\text{electrons}}. \quad (5.12)$$

The Schrödinger equation for the electrons reads

$$\mathcal{H}_e(\psi(\mathbf{r})) = [\mathcal{K}_e(\mathbf{r}) + \mathcal{P}_{Ne}(\mathbf{R}, \mathbf{r}) + \mathcal{P}_{ee}(\mathbf{r})] \psi(\mathbf{r}) = E_e(\mathbf{R}) \psi(\mathbf{r}). \quad (5.13)$$

The total energy of the system is the sum of electron and nuclei energies. The total eigenvalues are sums of individual eigenvalues of the separated terms.

We note, that the Born–Oppenheimer approximation does not always work. The approximation fails in those cases where the correlations between ionic and electronic motion are important, e.g. for processes where excited states are important such as photodissociation, cf. [351, 352]. The BOA also does not work for metals, where the electronic energy levels in the electron gas are very dense; thus, the adiabatic terms are not small any more, which, however is a necessary condition for BOA to work⁹. Due to BOA, the kinetic energy of the nuclei vanishes from the Hamiltonian and the distance of nuclei appears only as a parameter in the eigenvalues and eigenvectors. The potential energy surface in the Born–Oppenheimer approximation is given as a function of the relative coordinates of the nucleus, i.e.

$$E^{BO} = \frac{1}{2} \sum_{I \neq J} \frac{Z_I Z_J e^2}{R_{IJ}} + E_e(\{\mathbf{R}\}). \quad (5.14)$$

In this equation it is assumed that the electrons are in their ground state, i.e. the system is not excited. Quite frequently, the main goal of atomistic simulations is to determine the energy minimum of a system with respect to nuclear coordinates, i.e. to optimize the system geometry. A geometry optimization procedure consists of sampling points on the potential energy surface, searching for a minimum. In general, local minima on the potential surface correspond to metastable configurations, and the absolute minimum is the most stable configuration. If the energy is known as a function of nucleus coordinates, one can always calculate the energy gradient. It provides the direction in which the energy goes downhill most steeply, along with

⁷For example, for the lightest nucleus in the Hydrogen atom, the difference of mass is $m_{\text{proton}} \approx 1836 m_e$.

⁸There are some electronic motions which occur independent of atomic motion such as plasma oscillations $\mathcal{O}(10^{16})$ Hz, electron tunneling, photoabsorption or ionization processes.

⁹To show this, one has to calculate the diagonal matrix elements in perturbation theory of quantum mechanics. To first order, these matrix elements are zero. To second order, they are negligibly small, due to a mass factor of $m_{e^-}/m_{\text{nuclei}} \leq 0.0005$.

the steepness of the downhill slope. In the minima, but also at saddle points, the gradient is zero. However, the second derivative of the energy function can always be calculated to distinguish absolute form relative minima. A whole host of different numerical techniques – so-called *optimization algorithms* – have been developed to search for the energy minimum, e.g.

- Conjugate Gradients
- Steepest Descents
- Molecular Dynamics
- Monte Carlo Algorithms
- Genetic Algorithms

The steepest descent method is the simplest of all gradient methods which search for a minimum of a function f . This method approaches the minimum in a zig-zag fashion where the new direction is always orthogonal to the previous. The choice of direction is where f decreases most quickly, i.e. opposite to $\nabla f(\mathbf{x}_i)$. The search starts at an arbitrary point \mathbf{x}_0 and then slides down the gradient until one is close enough to the solution. In other words, the iterative procedure is

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \lambda_k \nabla f(\mathbf{x}_k) = \mathbf{x}_k - \lambda g(\mathbf{x}_k), \quad (5.15)$$

where g_k is the gradient at one given point.

$$\frac{d}{d\lambda_k} f(\mathbf{x}_{k+1}) = \nabla f(\mathbf{x}_{k+1})^T \frac{d}{d\lambda_k} \mathbf{x}_{k+1} = -\nabla f(\mathbf{x}_{k+1})^T \frac{d}{d\lambda_k} g(\mathbf{x}_k). \quad (5.16)$$

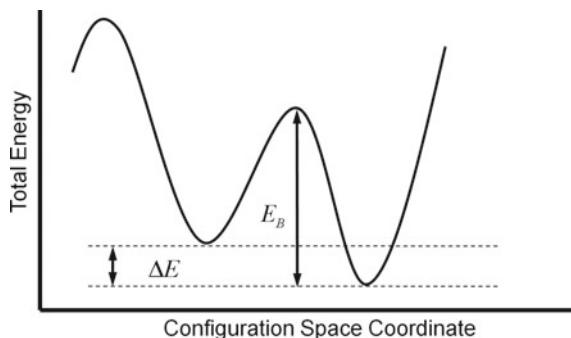
Setting this expression to zero one realizes that $f(\mathbf{x}_{k+1})$ and $g(\mathbf{x}_k)$ are orthogonal. The next step is then taken in the direction of the negative gradient of this new point and the iteration is continued until the extremum has been determined with a chosen accuracy ε .

Algorithm 5.1 shows a float chart of the conjugate gradient algorithm [353].

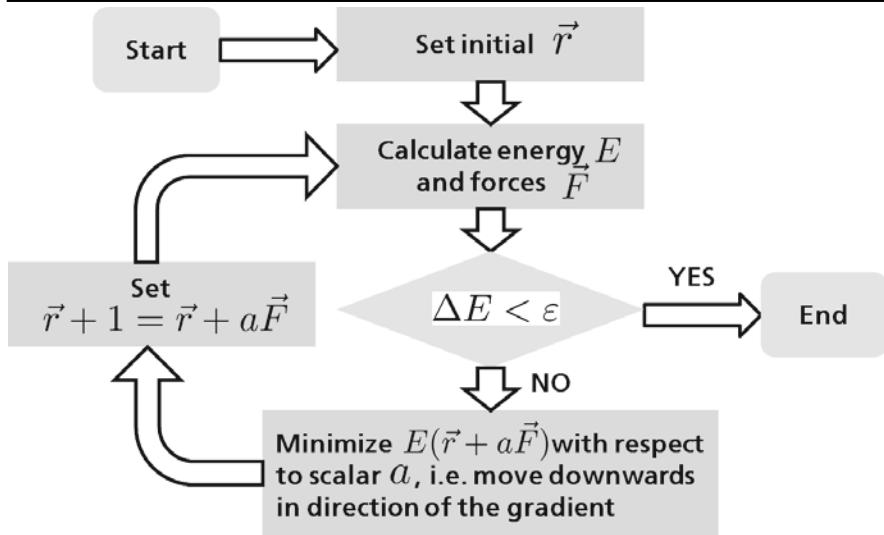
Molecular Dynamics

The Molecular Dynamics (MD) method with classical or quantum mechanical force models follows the dynamics of particles in their configuration space under the influence of external forces and forces acting between the particles. Using the Born–Oppenheimer approximation means that the particles are nuclei. MD monitors the motion of the nuclei over the potential energy surface with addition of the kinetic energy of the particles. The dynamics is governed by the Newtonian equations of motion, which are solved numerically, but the forces are calculated quantum-mechanically. The MD method can be very useful for finding local minima of the system's energy surface by gradually removing kinetic energy from the system. Most often however, a system is heated up, followed by a slow cooling, which allows the system to proceed from one local minimum to a different or the global one. This procedure is called *simulated annealing*, cf. Fig. 5.2. Figure 5.1 shows a flow chart of the conjugate gradient algorithm [353]. The MD method can also be used to determine the electronic ground state of a system for a fixed nuclei configuration by assigning

Fig. 5.2 Simulated annealing heats a system to artificially high temperatures, thereby making it possible for the system to overcome the potential barrier E_B between two energy minima



Algorithm 5.1: Flow chart of the conjugate gradient algorithm



fictitious masses to the electronic degrees of freedom. This technique is discussed below in the context of *Car–Parinello* simulations.

5.4 Density Functional Theory

As an alternative to the Hartree–Fock approximation see Sect. 5.6.1, one may consider replacing the electron-electron interactions by some effective potential acting on the electrons. Such an alternative method can be exactly formulated on the basis of the work done on *density functional theory* (DFT) by Hohenberg and Kohn in the

year 1964 [354] and extended by Kohn and Sham [355]. In their work, Hohenberg and Kohn considered the ground state of the electron-gas system in the external potential $v(\mathbf{r})$ and were able to prove that the following density functional theorem holds exactly:

Theorem 6 (Hohenberg–Kohn) *There is a universal functional – i.e. a functional independent of the external potential $v(\mathbf{r}) - F[\rho(\mathbf{r})]$ of the electron charge density distribution $\rho(\mathbf{r})$ that defines the total energy of the electronic system by the expression*

$$E = \int v(\mathbf{r})\rho(\mathbf{r}) d^3r + F[\rho(\mathbf{r})]. \quad (5.17)$$

The total energy E has a minimum when the charge density $\rho(\mathbf{r})$ coincides with the “true” charge density distribution in the external potential $v(\mathbf{r})$. This density functional theory is exact, if the ground state energy has no degeneracy.

In this theory, the exchange correlation energy is expressed as a functional of the resulting electron charge density distribution and the electronic states are solved self-consistently as in the Hartree or Hartree–Fock approximation. Because the electron density depends on only three coordinates (as opposed to the $3N$ coordinates of N electrons), the computational effort required to solve the equations of DFT is comparable with that required for Hartree–Fock theory, thus rendering DFT highly attractive for computational applications. The correct functional of the energy is not known and has to be constructed using heuristic approximations. Initial functionals, based principally on behavior of the electron gas were lacking in the accuracy required for chemical applications, but today, functionals capable of remarkable accuracy and breadth of applicability across the periodic table are available [355–357].

5.5 Car–Parinello Molecular Dynamics

The total energy of a system can be written as a functional depending on the spin-orbitals Ψ_k and of the nuclear coordinates $\{R_K\}$:

$$E_{\text{tot}} = E_{\text{el}}(\{\Psi_k\}, \{R_K\}) + U_{\text{rep}}(\{R_K\}) = E_{\text{tot}}(\{\Psi_k\}, \{R_K\}), \quad (5.18)$$

where the orbitals Ψ_k form an orthonormal set, i.e.

$$\langle \Psi_i | \Psi_j \rangle = \delta_{ij}. \quad (5.19)$$

Using HF and DFT, this energy is minimized with respect to the orbitals Ψ_k , according to the variational principle. Usually, a finite basis set $\{\chi_i\}$ is used, in terms of which the orbitals are given as

$$\Psi_k = \sum_i C_i^k \chi_i(r), \quad (5.20)$$

so that the energy can be written in terms of the C_i^k and $\{R_K\}$,

$$E_{\text{tot}} = E_{\text{tot}}(\{C_i^k\}, \{\mathbf{R}_K\}). \quad (5.21)$$

Car and Parinello used (5.21) with the constraint (5.19) as a starting point for determining the minimal energy configuration by calculating the minimum of the total energy as a function of C_i^k and the nuclear coordinates $\{R_K\}$ [358]. This procedure has the consequence that the electronic structure does not have to be calculated exactly for each atomic configuration, as both the nuclear positions and the electronic orbitals are varied simultaneously in order to locate the minimum. As a consequence, the energy minimization problem can be considered as an abstract numerical problem, and any minimization algorithm can in principle be used for minimization. Car and Parinello (CP) chose the molecular dynamics method as the minimization algorithm. Since the appearance of their paper, ab initio MD techniques have developed rapidly [353]. The treatment of the wave function and the atomic coordinates by integrating Newton’s equation of motion was a completely new idea, contrary to the previously common-sense idea that the electronic structure must be calculated by means of matrix diagonalization and self-consistent iterations. The basic equations of motion in the CP approach are derived from a Lagrangian which leads to a classical mechanics problem in which the total energy (5.21) acts as a potential. This approach is particularly suitable for microcanonical simulations. With the CP method, one no longer has to treat huge eigenvalue problems and it is possible to reduce significantly both the computational memory and the computation time.

The basic equation of motion of the CP method is given by the Lagrangian \mathcal{L} as a functional of the wavefunction Ψ_k^{10} and the atomic positions $\{\mathbf{R}_K\}$:

$$\mathcal{L}(\{\Psi_k\}, \{R_K\}) = \frac{\mu}{2} \sum_k \dot{\Psi}_k^2 + \sum_n \frac{M_n}{2} \dot{\mathbf{R}}_K^2 - E_{\text{tot}}(\{\Psi_i^k\}, \{\mathbf{R}_K\}) + \sum_{kl} [\Lambda_{kl} \langle \Psi_k | \Psi_l \rangle - \delta_{kl}], \quad (5.22)$$

where μ is a fictitious small electron mass associated with the electronic orbital and governing the motion of the wavefunctions, and M_n is the actual mass of the n -th nucleus with position \mathbf{R}_n . The last term on the right-hand side is necessary to ensure orthonormality of Ψ_k . The importance of the existence of a Lagrangian is, that the system has no energy dissipation, and the energy, including the fictitious kinetic energy of the electrons $\mu/2 \sum_k \dot{\Psi}_k^2$. From (5.22), one may derive the equation of motion for the wavefunctions

$$\mu \ddot{\Psi}_k = -\frac{\partial E_{\text{tot}}}{\partial \Psi_k} + \sum_l \Lambda_{kl} \Psi_l(\mathbf{r}) = \mathcal{H} \Psi_k + \sum_l \Lambda_{kl} \Psi_l(\mathbf{r}), \quad (5.23)$$

where \mathcal{H} is the single-particle Hamiltonian of the system (e.g. the Kohn–Sham Hamiltonian). The equations of motion for the nuclei read:

$$M_n \ddot{\mathbf{R}}_n = \frac{-\partial E_{\text{tot}}}{\partial \mathbf{R}_n}. \quad (5.24)$$

¹⁰ k denotes the level number of electronic states.

When including friction into the equations of motion,

$$\mu \ddot{\Psi}_k = -\frac{\partial E_{\text{tot}}}{\partial \Psi_k} + \sum_l \Lambda_{kl} \Psi_l(\mathbf{r}) - \gamma \dot{\Psi}_k. \quad (5.25)$$

then the solution will become stationary after some time, and the left-hand side vanishes, i.e. $\ddot{\Psi}_k = 0$. The values of the Lagrange parameters depend on time and thus must be calculated at each molecular dynamics time step, such that they guarantee the orthonormality constraint (5.19). The integration procedure can be done using the Verlet algorithm, or, as in their original paper [358], the SHAKE-algorithm [359]. For some very readable reviews the Car–Parinello approach, see Kremler and Mad-den [360] or Tuckerman [296, 361].

5.5.1 Force Calculations: The Hellmann–Feynman Theorem

The forces f_I acting on ion I are given by the derivative of the energy with respect to the position of the ion:

$$\mathbf{f}_I = -\frac{dE}{d\mathbf{R}_I}, \quad (5.26)$$

where the negative sign is convention, the total energy E is given by $E = \langle \psi | H | \psi \rangle$, and the notation of the derivative in (5.26) is a symbolic way (common in physics) of writing down the derivative with respect to all components of \mathbf{R}_I . The simplest way to estimate the force \mathbf{f} is to perform the calculation numerically by moving the ion in all directions. For a system composed of N nuclei there are $6N$ calculations of the energy required and N may be as large as 1000.

The principle idea of estimating the force is to consider that, as the nucleus moves, the wave function changes, so that the full derivative of the energy can be expanded in terms of the wave functions as follows:

$$\frac{dE}{d\mathbf{R}_I} = \frac{\partial E}{\partial \mathbf{R}_I} + \sum_i \frac{\partial E}{\partial \psi_i} \frac{\partial \psi_i}{\partial \mathbf{R}_I} + \sum_i \frac{\partial E}{\partial \psi_i^*} \frac{\partial \psi_i^*}{\partial \mathbf{R}_I}, \quad (5.27)$$

where ψ_i are the eigenfunctions of the system. Using standard ket-notation, (5.27) can be re-written as:

$$\frac{dE}{d\mathbf{R}_I} = \sum_i \langle \psi_i | \frac{\partial \mathcal{H}}{\partial \mathbf{R}_I} | \psi_i \rangle + \sum_i \langle \frac{\partial \psi_i}{\partial \mathbf{R}_I} | \mathcal{H} | \psi_i \rangle + \sum_i \langle \psi_i | \mathcal{H} | \frac{\partial \psi_i}{\partial \mathbf{R}_I} \rangle. \quad (5.28)$$

For the eigenstates $H\psi_i = \lambda_i \psi_i$ one obtains:

$$\sum_i \langle \frac{\partial \psi_i}{\partial \mathbf{R}_I} | \lambda_i \psi_i \rangle + \sum_i \langle \psi_i | \lambda_i | \frac{\partial \psi_i}{\partial \mathbf{R}_I} \rangle = \sum_i \lambda_i \frac{\partial}{\partial \mathbf{R}_I} \langle \psi_i | \psi_i \rangle = 0, \quad (5.29)$$

because $\langle \psi_i | \psi_i \rangle$ is a constant. Thus, the electronic degrees of freedom do not contribute to the derivatives of the energy and one obtains:

$$f_I = \frac{dE}{dE_I} = \frac{\partial E}{\partial \mathbf{R}_I} = \frac{\partial \langle \psi_i | \mathcal{H} | \psi_i \rangle}{\partial \mathbf{R}_I} = \langle \psi_i | \frac{\partial \mathcal{H}}{\partial \mathbf{R}_I} | \psi_i \rangle. \quad (5.30)$$

When each ψ_i is an eigenstate of the Hamilton operator, the partial derivative of the energy with respect to the position of an ion gives the real physical force on the ion. This theorem holds for any derivative of the total energy and is referred to as the **Hellman–Feynman theorem** [362, 363].

Forces calculated using (5.30) are usually very sensitive to errors in the wave function. The error in the force is of first order and the error in the energy is of second order with respect to errors in the wave functions. Thus, it is easier to calculate an accurate total energy than it is to calculate the exact forces; the latter can only be achieved when the wave functions are already close to the exact eigenstates of the system. The Hellmann–Feynman theorem of (5.30) is important for practical purposes: The subroutine that calculates the total energy by solving the expensive eigenvalue problem only has to be called once, which can result in a considerable speed-up, given that the number of considered particles may be as large as $\sim 10^3$. The forces can then be readily computed according to (5.30) using the Hamiltonian derivative matrix.

5.5.2 Calculating the Ground State

There are many first principles approaches for calculating the ground state of many-atom systems which can be classified into three main categories: The first approach is based on the Hartree–Fock approximation, which offers a rigorous one-electron approximation; the second approach is based on DFT which offers an exact background for the many-body problem but can only be solved approximately and the resulting equation is one-electron-like. For both approaches, the deviation from the exact solution is called *electron correlation* and there are several ways to correct the approximative solution. The third approach is quite different using quantum Monte Carlo methods. In this section we introduce yet another approximative method of calculating ground states which is based on the variational principle. For a system in state Ψ , the average of many measurements of the energy is given by

$$E[\Psi] = \frac{\langle \Psi | \mathcal{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle}. \quad (5.31)$$

One may expand Ψ in terms of the normalized eigenstates ϕ_k of \mathcal{H} :

$$\Psi = \sum_k c_k \phi_k. \quad (5.32)$$

Due to the orthogonality of ϕ_k the energy is:

$$E[\Psi] = \frac{\sum_k |c_k|^2 E_k}{\sum_k |c_k|^2}, \quad (5.33)$$

where E_k is the energy for the k -th eigenstate of \mathcal{H} . $E[\Psi]$ is minimal only for $\Psi = c_0 \phi_0$, i.e. $E[\Psi_0] = E_0 = \min_{\Psi} E[\Psi]$. Hence, the ground state of a system may be calculated by minimization of the functional $E[\Psi]$ with respect to all allowed N -electron wave functions:

$$\delta[\langle \Psi | \mathcal{H} | \Psi \rangle - E \langle \Psi | \Psi \rangle] = 0. \quad (5.34)$$

Many approximative methods in quantum chemistry including HF methods are based on the variational principle of (5.34).

Example 36 (Ground State Energy for a Hydrogen Atom) If one already knows some properties of Ψ one can choose it in functional form e.g. in one dimension as $\Psi(x) = a_1x^2 + a_2x^3 + a_3$ with unknown coefficients $\{a_i\}$. For an H -atom one may choose the following Gaussian-type form of the trial function:

$$\phi^G = \exp(\beta r^2). \quad (5.35)$$

The average energy is

$$E^G = \frac{\langle \phi^G | \mathcal{H} | \phi^G \rangle}{\langle \phi^G | \phi^G \rangle} = 3/2 \beta - \frac{2\sqrt{2\beta}}{\sqrt{\pi}}. \quad (5.36)$$

From the condition $\partial E^G / \partial \beta = 0$ one obtains $\beta^0 = 8/9\pi$ and thus:

$$E_0^G(\beta^0) = -\frac{4}{3\pi} = -0.424 \text{ a.u.} \quad (5.37)$$

Using a Slater trial function $\phi^S = \exp(-\beta r)$ and the same variational method one obtains the exact result in atomic units (a.u.)

$$E_0^S(\beta^0 = 1) = -\frac{1}{2} = -0.5 \text{ a.u.} \quad (5.38)$$

Variational methods have several drawbacks such as:

- The variational ground state is always an upper bound on the exact value and it is difficult to estimate how close it is to the true value.
- The method works badly for excited state energies.

Atomic Units

In fundamental equations such as the Hamiltonian for a condensed matter system (5.10), there are several constants the numerical values of which depend on the chosen system of units. In order to avoid numerical calculations with either very small or very large numbers which introduces numerical rounding errors, one usually employs reduced “simulation units” in which some basic units (usually length and energy) are chosen in such a way that some other quantities can be set equal to one. In quantum chemistry it is very common to use as reduced system the atomic units (a.u.), in which $e = \hbar = m_e = 1$. The unit of length is the Bohr radius a_B of the H -atom

$$a_B = \frac{\hbar^2}{m_e e^2} = 0.5292 \times 10^{-10} \text{ m} \approx 0.5 \text{ \AA}, \quad (5.39)$$

and the energy is given in hartrees (H), where

$$1H = \frac{e^2}{a_B} = \frac{m_e e^4}{\hbar^2} = 2 \text{ Rydberg} = 27.21 \text{ eV} = 4.36 \times 10^{-18} \text{ J}. \quad (5.40)$$

In these units, the Schrödinger equation has a very simple form:

$$\left[-\frac{1}{2} \nabla^2 - \frac{1}{r} \right] \psi(r) = E\psi(r). \quad (5.41)$$

Important energy conversion factors can be found in Appendix E.2.

5.6 Solving Schrödinger's Equation for Many-Particle Systems: Quantum Mechanics of Identical Particles

The n th eigenfunction Ψ_ξ of a quantum mechanical system with Hamiltonian \mathcal{H} , composed of N identical particles can be written as $\Psi_n(\xi_1, \xi_2, \dots, \xi_N)$, where the particle coordinates ξ_i depends on the coordinates $\{\mathbf{r}_i\}$ and spins $\{\sigma_i\}$, i.e. $\xi_i = (\{\mathbf{r}_i\}, \{\sigma_i\})$. The time-independent Schrödinger equation reads

$$\mathcal{H}\Psi_n(\xi_1, \xi_2, \dots, \xi_N) = E_n\Psi_n(\xi_1, \xi_2, \dots, \xi_N), \quad (5.42)$$

where E denotes the energy eigenvalue of the n th state. Since \mathcal{H} is Hermitian, all eigenvalues are real numbers. Since electrons are identical particles which are indistinguishable (in contrast to classical particles) the wave function has a gauge symmetry (cf. Chap. 3) with respect to permutation of two particles, i.e.:

$$\Psi(\xi_1, \dots, \xi_i, \dots, \xi_j, \dots, \xi_N) = e^{i\alpha}\Psi(\xi_1, \dots, \xi_j, \dots, \xi_i, \dots, \xi_N), \quad (5.43)$$

where $\alpha \in \mathbf{R}$ is a constant. Repeating the operation in (5.43) yields

$$\Psi(\xi_1, \dots, \xi_i, \dots, \xi_j, \dots, \xi_N) = e^{2i\alpha}\Psi(\xi_1, \dots, \xi_i, \dots, \xi_j, \dots, \xi_N). \quad (5.44)$$

Hence, $\exp(2i\alpha) = 1$, and $\exp(i\alpha) = \pm 1$. Thus,

$$\Psi(\xi_1, \dots, \xi_i, \dots, \xi_j, \dots, \xi_N) = \pm\Psi(\xi_1, \dots, \xi_j, \dots, \xi_i, \dots, \xi_N), \quad (5.45)$$

that is, the wave function is either symmetric or antisymmetric with respect to particle permutations. Particles are called bosons, if their wave function is symmetric and fermions if it is antisymmetric, cf. Fig. 2.11 on p. 61. Electrons are fermions with spin $1/2$; because of the antisymmetry of their wave function, it is easily seen that in many-electron systems there can never be more than one electron in the quantum state, because then the fermion wave function becomes identically zero. This is called Pauli's exclusion principle [364], cf. Sect. 2.5.2 on p. 61. For bosons no such principle exists. The Hamiltonian for an N -electron system can be written in reduced atomic units as

$$\mathcal{H} = -\frac{1}{2} \sum_{i=1}^N \nabla_i^2 + \sum_{i>j}^N \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} + \sum_{i=1}^N v(\mathbf{r}_i), \quad (5.46)$$

were the different terms are the electron kinetic energy, the electron-electron Coulomb interaction and the Coulomb potential caused by the nuclei. Let us consider the simplest system, composed of N electrons moving independently. In this case, the total electron Hamiltonian is additive for all electrons, i.e.

$$\mathcal{H} = \mathcal{H}_1 + \mathcal{H}_2 + \dots + \mathcal{H}_N. \quad (5.47)$$

Likewise, the total wave function Ψ for N independent particles can in general be written as the product of N one-particle wave functions (also called *spin-orbitals* in chemistry) $\psi_{p1}, \psi_{p2}, \dots, \psi_{pN}$. For the eigenvalue equation one obtains

$$\mathcal{H}\Psi_{\lambda_1, \lambda_2, \dots, \lambda_N}(\xi_1, \dots, \xi_N) = E_{\lambda_1, \lambda_2, \dots, \lambda_N}(\xi_1, \dots, \xi_N)\Psi_{\lambda_1, \lambda_2, \dots, \lambda_N}(\xi_1, \dots, \xi_N), \quad (5.48)$$

and one finds that the antisymmetric multi-particle wave function can be written as a determinant, i.e. as the symmetrized product of spin-orbitals

$$\Psi(\xi_1, \xi_2, \dots, \xi_N) = \frac{1}{\sqrt{N!}} \begin{vmatrix} \psi_{p1}(\xi_1) & \psi_{p2}(\xi_1) & \cdots & \psi_{pN}(\xi_1) \\ \psi_{p1}(\xi_2) & \psi_{p2}(\xi_2) & \cdots & \psi_{pN}(\xi_2) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_{p1}(\xi_N) & \psi_{p2}(\xi_N) & \cdots & \psi_{pN}(\xi_N) \end{vmatrix} \quad (5.49)$$

The form of the wave function in (5.49) is the same as the determinant of a matrix and is called *Slater determinant*. The symmetry properties of the non-relativistic many-particle wave function which determines the occupation of quantum states by fermions and bosons, also affects significantly the statistics of quantum systems, particularly at low temperatures. The statistics obeyed by fermions and bosons in the occupation of quantum states are called, respectively, *Fermi–Dirac statistics* [365, 366] and *Bose–Einstein statistics* [367, 368].

There are many quantum states λ_i having different energy eigenvalues and each of these states is called an energy level. In the ground state, the quantum states are occupied by electrons one by one, from energetically lower levels to higher levels, due to Pauli's principle. The highest occupied level is called the Fermi level and the eigenvalue energy is called the Fermi energy E_F . All levels below E_F are occupied, all levels above E_F are empty. This is also the case in solids.

5.6.1 The Hartree–Fock Approximation

The Hartree–Fock (HF) method is widely used in electronic structure calculations and it is particularly popular in (quantum) chemistry, because it allows for calculating the characteristics of small molecules with high accuracy. Typical systems treated with HF are:

- Fixed atomic nuclei with many electrons (e.g. a crystal or small molecules).
- Fixed atomic nucleus with many electrons (calculation of the energy values of a single atom).

In principle, the HF method is an approximation that was introduced by D.R. Hartree (1897–1958) in 1928 [369] to treat approximately N -electron systems as a product of one-electron wave functions, and extended by W.A. Fock (1898–1974) in 1930 [370] by using the Slater determinant as ansatz for the wave function Ψ , and thus taking into account the correct antisymmetry of the wave function. The first approximation in HF methods is the Born–Oppenheimer approximation, i.e. the Schrödinger equation

is solved for the electrons in the field of static nuclei. The second approximation is the replacement of the many-electron Hamiltonian with an effective one-electron Hamiltonian which acts on the one-electron wave functions, which are called *orbitals* in chemistry. The Coulomb repulsion between electrons is represented in an averaged way. The matrix elements of the effective Hamiltonian, however, depend on the wave function, i.e., the Schrödinger equation becomes non-linear and is solved using a self-consistent (SC) numerical scheme, in which the starting point is a simple ansatz for the potential, say $V_a^{(0)} = 0$. Then, using the one-particle Schrödinger equation, the wave function $\Psi(x)$ is calculated and the new potential $V_a^{(1)}$ is determined. If this potential finally converges, one obtains a self-consistently determined additional potential $V_a = \lim_{n \rightarrow \infty} V_a(n)$, which generally describes the potential of all the other electrons.

In this section we discuss first the Hartree approximation and in the next section we derive the Hartree–Fock equations. The starting point of the Hartree method is the total Hamiltonian of a system of N electrons. The total Hamilton operator \mathcal{H} is divided into the one-electron part $\mathcal{H}_0(i)$ and the electron-electron Coulomb interaction $\mathcal{U}(i, j)$ as follows:

$$\mathcal{H} = \sum_i \mathcal{H}_0(i) + \frac{1}{2} \sum_{i \neq j} \mathcal{U}(i, j), \quad (5.50)$$

where

$$\mathcal{U}(i, j) = \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|}, \quad (5.51)$$

and

$$\mathcal{H}_0(i) = -\frac{1}{2} \nabla_i^2 - \sum_j \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|}. \quad (5.52)$$

Here, Z_j is the nucleus charge of the j th atom and the interaction operator \mathcal{U} is the term which renders the solution of the Schrödinger equation difficult.

Ignoring the interaction term, one makes use of the fact, that solving the time independent Schrödinger equation is equivalent to the following variation problem, see e.g. [336]:

$$\langle E \rangle = \frac{\langle \Psi | \mathcal{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \text{minimum}, \quad (5.53)$$

respectively

$$\langle E \rangle = \langle \Psi | \mathcal{H} | \Psi \rangle = \text{minimum}, \quad (5.54)$$

with the additional condition

$$\langle \Psi | \Psi \rangle = 1. \quad (5.55)$$

If one could neglect the interaction term (5.51) then the time independent Schrödinger equation $\mathcal{H}|\Psi\rangle = E|\Psi\rangle$ could be solved by a simple product of one-electron wave functions in the form:

$$\Psi(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) = \psi_1(\mathbf{r}_1) \psi_2(\mathbf{r}_2) \cdots \psi_N(\mathbf{r}_N). \quad (5.56)$$

Inserting this product into (5.54) yields:

$$\langle E \rangle = \sum_k \langle \psi_k | \mathcal{H}_0 | \psi_k \rangle + \frac{1}{2} e^2 \sum_k \sum_{k' \neq k} \langle \psi_k \psi'_{k'} | \frac{1}{|\mathbf{r}_k - \mathbf{r}_{k'}|} | \psi_k \psi'_{k'} \rangle \quad (5.57)$$

with the constraint

$$1 - \langle \psi_k | \psi_k \rangle = 0. \quad (5.58)$$

After multiplying (5.58) with the Lagrange multiplier E_k and adding the result to $\langle E \rangle$ yields the variation

$$\delta \left[\langle E \rangle - \sum_k E_k \langle \psi_k | \psi_k \rangle \right] = 0. \quad (5.59)$$

Performing the variation in (5.59) yields

$$\langle \delta \psi_j | \mathcal{H}_0 | \psi_j \rangle + e^2 \sum_{k \neq j} \langle \delta \psi_j | \psi_k | \frac{1}{|\mathbf{r}_k - \mathbf{r}_j|} | \psi_j \psi_k \rangle - E_j \langle \delta \psi_j | \psi_j \rangle \quad (5.60a)$$

$$= \langle \delta \psi_j | \left[\mathcal{H}_0 + e^2 \sum_{k \neq j} \langle \psi_k | \frac{1}{|\mathbf{r}_k - \mathbf{r}_j|} | \psi_k \rangle - E_j \right] | \psi_j \rangle = 0. \quad (5.60b)$$

As all variations $\delta \psi_j$ are independent, one obtains the one-particle equation

$$\left[\mathcal{H}_0 + e^2 \sum_{k \neq j} \int \frac{|\psi_k(\mathbf{r}')|^2}{|\mathbf{r} - \mathbf{r}'|} d^3 r' \right] \psi_j(\mathbf{r}) = E_j \psi_j(\mathbf{r}), \quad (5.61)$$

which is called **Hartree Equation**. Equation (5.61) is an equation for the one-particle wave function $\psi_j(\mathbf{r})$. The potential of this one-particle wave function is given by:

$$\mathcal{P} = - \sum_j \frac{Z_j}{|\mathbf{r}_i - \mathbf{R}_j|} + \underbrace{e^2 \sum_{k \neq j} \int \frac{|\psi_k(\mathbf{r}')|^2}{|\mathbf{r} - \mathbf{r}'|} d^3 r'}_{V_a(\mathbf{r})}. \quad (5.62)$$

The additional potential V_a takes into account all the other particles. To calculate V_a one has to know all the other one-particle wave functions.

Derivation of the Hartree–Fock Equations

For the derivation of the HF equations one applies the variational principle of Sect. 5.5.2 and of the previous section to the Schrödinger equation using the Born–Oppenheimer Hamiltonian (5.14). One assumes the electrons to be independent and the trial function $\Psi(\xi_1, \xi_2, \dots, \xi_N)$ to be a product of spin-orbitals (one-electron wave functions) ψ_i in the form of a Slater determinant according to (5.49), having the correct antisymmetry, and the total energy is minimized with respect to the spin-orbitals in the determinant. The orbital $\psi_k(\xi_i)$ denotes the one-electron wave function of the k th level, which forms an orthonormal set, i.e.

$$\langle \psi_k | \psi_l \rangle = \sum_{s_i} \int \psi_k^\star(\xi_i) \psi_l(\xi_i) d^3 r_i = \delta_{kl} \quad (5.63)$$

Using (5.49) one can calculate the expectation value of the energy $\langle E \rangle = \langle \Psi | H | \Psi \rangle$ and obtains:

$$\langle E \rangle = \sum_k \langle \Psi_k | \mathcal{H}_0(i) | \psi_k \rangle + \frac{1}{2} \sum_{k,l} \langle \psi_k \psi_l | \mathcal{U} | \psi_k \psi_l \rangle - \frac{1}{2} \sum_{k,l} \langle \psi_k \psi_l | \mathcal{U} | \psi_l \psi_k \rangle. \quad (5.64)$$

For the matrix elements of the one-electron part one obtains:

$$\langle \Psi | \sum_i \mathcal{H}_0 | \Psi \rangle = N \frac{(N-1)!}{N!} \sum_k \langle \psi_k | \mathcal{H}_0 | \psi_k \rangle = \sum_k \langle \psi_k | \mathcal{H}_0 | \psi_k \rangle \quad (5.65a)$$

$$= \sum_k \int d\mathbf{x}_i \psi_k^*(\mathbf{x}) \mathcal{H}_0(\mathbf{r}) \psi_k(\mathbf{x}), \quad (5.65b)$$

where the integral means the integration over the spacial coordinates and summation over the spin-degrees of freedom.

Analogously, for the matrix elements of the two-electron term $\mathcal{U}(i,j)$ one obtains (as final expression)

$$\langle \Psi | \sum_{i,j} \mathcal{U}(i,j) | \Psi \rangle = \sum_{kl} \langle \psi_k \psi_l | \mathcal{U} | \psi_k \psi_l \rangle - \sum_{kl} \langle \psi_k \psi_l | \mathcal{U} | \psi_l \psi_k \rangle, \quad (5.66)$$

where the following notation is used:

$$\langle \psi_k \psi_l | \mathcal{U} | \psi_m \psi_n \rangle = \int \int d\mathbf{x}_1 d\mathbf{x}_2 \psi_k^*(\xi_1) \psi_l^*(\xi_2) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \psi_m(\mathbf{x}_1) \psi_n(\mathbf{x}_2). \quad (5.67)$$

Defining the *Coulomb operator* J and the *exchange operator* K as

$$J = \sum_{k=1}^N J_k, \quad \text{and} \quad K = \sum_{k=1}^N K_k, \quad (5.68)$$

where the operators J_k and K_k act on the spin-orbitals $\psi_k(\xi_i)$ as follows

$$J_k \psi(\xi_i) = \sum_{s_j} \int d\mathbf{r}_j^3 \psi_k^*(\xi_j) \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \psi_k(\xi_j) \psi(\xi_i) \quad (5.69a)$$

$$K_k \psi(\xi_i) = \sum_{s_j} \int d\mathbf{r}_j^3 \psi_k^*(\xi_j) \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \psi(\xi_j) p\psi_k(\xi_i), \quad (5.69b)$$

one may write the energy functional for the Slater determinant as

$$E = \sum_k \langle \psi_k | \mathcal{H}_0 + \frac{1}{2}(J - K) | \psi_k \rangle. \quad (5.70)$$

As a next step, the minimization of the energy functional with respect to the spin-orbitals ψ_k has to be done with the additional condition constraint

$$\langle \psi_k | \psi_l \rangle = \delta_{kl}. \quad (5.71)$$

Hence, one has to solve a minimization problem with constraints. Introducing the Lagrange multipliers for the constraints (5.71) one obtains

$$\delta E - \sum_{kl} \lambda_{kl} [\langle \delta \psi_k | \psi_l \rangle - \langle \psi_k | \delta \psi_l \rangle] = 0, \quad (5.72)$$

where the variation of the energy is

$$\delta E = \sum_k \langle \delta\psi_k | \mathcal{H}_0 | \psi_k \rangle + c.c. \quad (5.73a)$$

$$+ \frac{1}{2} \sum_{kl} [\langle \delta\psi_k \psi_l | \mathcal{U} | \psi_l \psi_k \rangle + \langle \psi_l \delta\psi_k | \mathcal{U} | \psi_l \psi_k \rangle] \quad (5.73b)$$

$$- \langle \delta\psi_k \psi_l | \mathcal{U} | \psi_l \psi_k \rangle - \langle \psi_l \delta\psi_k | \mathcal{U} | \psi_l \psi_k \rangle] + c.c. \quad (5.73c)$$

$$= \sum_k \langle \delta\psi_k | \mathcal{U} | \psi_k \rangle + c.c. \quad (5.73d)$$

$$+ \sum_{kl} [\langle \delta\psi_k \psi_l | \mathcal{U} | \psi_l \psi_k \rangle - \langle \delta\psi_k \psi_l | \mathcal{U} | \psi_l \psi_k \rangle] + c.c. \quad (5.73e)$$

In the second step (5.73b) the symmetry property of the two-electron matrix elements was used:

$$\langle \psi_k \psi_l | \mathcal{U} | \psi_m \psi_n \rangle = \langle \psi_l \psi_k | \mathcal{U} | \psi_n \psi_m \rangle. \quad (5.74)$$

Thus, (5.73) can be re-written as

$$\delta E = \sum_k \langle \delta\psi_k | \mathcal{F} | \psi_k \rangle + \langle \psi_k | \mathcal{F} | \delta\psi_k \rangle, \quad (5.75)$$

with

$$\mathcal{F} = \mathcal{U} + J - K. \quad (5.76)$$

The Hermitian operator \mathcal{F} in (5.76) is called the *Fock operator*. Using $\delta E = 0$ one finally arrive at the equation

$$\langle \delta\psi_k | \mathcal{F} | \psi_k \rangle + \langle \psi_k | \mathcal{F} | \delta\psi_k \rangle + \sum_l \lambda_{kl} (\langle \delta\psi_k | \psi_l + \langle \psi_l | \delta\psi_k \rangle) = 0. \quad (5.77)$$

The variation $\delta\psi$ is arbitrary and given that $\lambda_{kl} = \lambda_{kl}^*$ due to the symmetry of the constraint equations, one obtains from (5.77):

$$\mathcal{F}\psi_k = \sum_l \lambda_{kl}\psi_l. \quad (5.78)$$

The Lagrange parameters λ_{kl} must be chosen in a way, such that the solution ψ_k form an orthogonal set. An obvious solution of (5.77) is found by taking ψ_k as eigenvectors of the Fock operator with eigenvalues ε_k and $\delta_{kl} = \varepsilon_k \delta_{kl}$. Thus,

$$\mathcal{F}\psi_k = \varepsilon_k \psi_k. \quad (5.79)$$

Equation (5.79) and the approximations used to derive it, i.e. the BO approximation along with the use of a single Slater determinant to express the many electron wave function are called, respectively, the *Hartree–Fock Equation* and the *Hartree–Fock approximation*. The Hartree–Fock equation is a non-linear equation which has to be solved by a self-consistent procedure as described above. Because an electron is thought to move in an averaged field generated by all other electrons and nuclei, the name “Self-Consistent Field theory” (SCF) is used for this approach. To find the ground state, one takes the lowest N eigenstates of the spectrum as the electron spin orbitals. These orbitals are used to build the new Fock operator, then the procedure

is repeated over and over again until convergence is reached. In summary, the HF approach means that the many-electron problem is approximated by a sequential calculation of the energy states of one electron in the average potential generated by the rest of the electrons.

Basis Functions

For molecules, the HF approximation has been used mainly in the field of quantum chemistry, where each one-electron wave function is approximated by a superposition of atomic orbitals. Thus, in order to solve the eigenvalue problem (5.79) for the Fock operator, one has to expand the spin orbitals $\psi_k(\xi_i)$ in a basis set, i.e. one has to represent the orbitals as linear combination of basis states χ_p :

$$\psi_k(\xi_i) = \sum_{p=0}^N C_{pk} \chi_{\xi_i} \quad (5.80)$$

The basis set should be carefully chosen, since the computation time for matrix diagonalization scales with the third power of the number of basis functions and a small basis set is desirable. At the same time this set should be physically well motivated to provide a good approximation of the Fock equation. Most widely used are three different types of basis sets:

- Plane waves (LCAO, MO) $\sim e^{ikr}$,
- Slater-type orbitals (STO) $\sim r^n e^{-\alpha r}$, see e.g. [371, 372]
- Gaussian-type orbitals (GTO) $\sim e^{-\alpha r^2}$, see e.g. [373, 374]

The plane wave approach originates from solid state physics and works very well for crystalline solids were one can employ several simplifications due to the periodicity of a crystal. It can also be applied to amorphous and finite systems such as clusters or defect configurations. The plane wave method expresses the one-electron wave function in terms of a linear combination of atomic (LCAO) or molecular orbitals (MO), where the latter term is mostly used for isolated clusters and molecules in chemistry. The Slater-type orbitals (STO) are the most natural basis set to choose for relatively light atoms as Slater-like functions are typical solutions of the radial Schrödinger equation for hydrogen-like atoms. When using STO basis sets, two-center integrals between different atomic orbitals which are centered at different positions can be evaluated analytically [375]. Numerical integration can be used for small basis sets, but it is impractical for large systems. Gaussian-type (GTO) basis sets avoid this problem. In calculations, Slater and Gaussian functions used as basis sets are often written as product of the exponent and a spherical function. We note that plane waves form a complete basis set; this means that the number of basis functions is increased until convergence occurs, the solution of the Fock equation has converged to the true value. Gaussian or Slater orbitals however, are not complete basis sets, which is why it is much more difficult to control the accuracy of a numerical calculation.

Hartree–Fock Roothan Equations

The use of Gaussian and Slater functions as basis sets has one crucial disadvantage: functions localized on different atoms A and B are non-orthogonal to each other. Thus, the integral

$$\int dV \exp[-\beta(\mathbf{r} - \mathbf{R}_A) \times \exp[-\beta(\mathbf{r} - \mathbf{R}_B)] \quad (5.81)$$

is nonzero. Representing the solution Ψ of the Schrödinger equation as a series over non-orthogonal functions according to (5.80) and projecting the eigenwert equation onto the basis states χ_p yields:

$$H \sum_q C_p \chi_q(x) = E \sum_q C_p \chi_q(x) . \quad (5.82)$$

Hence, one yields the following matrix equation:

$$HC = ESC , \quad (5.83)$$

where H is the Hamilton matrix, S is the overlap matrix, the elements of which are given by

$$H_{pq} = \langle \chi_p | H | \chi_q \rangle , \quad S_{pq} = \langle \chi_p | \chi_q \rangle . \quad (5.84)$$

This equation is called *generalized eigenvalue problem*. Except for the additional matrix S on the right-hand side, this equation is an ordinary eigenvalue problem. It can be transformed completely to the ordinary problem

$$H'C' = EC' \quad (5.85)$$

by introducing

$$C' = V^{-1}C ; \text{ and } H' = V^\dagger HV , \quad (5.86)$$

with

$$V^\dagger SV = I . \quad (5.87)$$

There are efficient algorithms to perform this orthogonalization, e.g. Gram–Schmidt transformation, see Algorithm 5.2.

Using the non-orthogonal basis set $\{\chi_p(r)\}$ in the Fock-equation (5.78) one obtains the following matrix equation:

$$C_k = \varepsilon_k SC_k , \quad (5.88)$$

where S is the overlap matrix for the orbital basis. This generalized eigenvalue equation is called the **Roothaan** equation. The matrix F is given by

$$F_{pq} = h_{pq} = \sum_k \sum_{rs} C_{rk}^* C_{sk} (2 \langle pr | g | qs \rangle - \langle pr | g | sq \rangle) , \quad (5.89)$$

where

$$h_{pq} = \langle p | h | q \rangle = \int d3r \chi_p^*(\mathbf{r}) \left[-\frac{1}{2} \nabla^2 - \sum_n \frac{Z_n}{|\mathbf{R}_n - \mathbf{r}|} \right] \chi_q(\mathbf{r}) . \quad (5.90)$$

Algorithm 5.2: The Gram–Schmidt Algorithm

Let $\mathbf{S} = \{v_n : n = 1, 2, 3, \dots\}$ be a well-ordered set of vectors in pre-Hilbert space \mathbf{V} . For simplicity, it is also assumed that \mathbf{S} is countable. Let V_c be the collection of all finite linear combinations of \mathbf{S} , and suppose that \mathbf{S} is dense in \mathbf{V} . Then one can obtain an *orthonormal basis* from \mathbf{S} by the following procedure:

1. Let the first v_{n_1} be the first of the v_i which is non-zero, and put $e_1 = \frac{v_{n_1}}{\|v_{n_1}\|}$.
2. Let v_{n_2} be the first of the v_i which is *not* a multiple of e_1 .
3. Put $f_2 = v_{n_2} - \langle v_{n_2}, e_1 \rangle e_1$ and
4. $e_2 = \frac{f_2}{\|f_2\|}$.

Inductively, suppose one has chosen e_1, \dots, e_k which form an orthonormal set. Let $v_{n_{k+1}}$ be the first of the v_i not expressible as a linear combination of e_1, \dots, e_k .

5. Put $f_{k+1} = v_{n_{k+1}} - \sum_{1 \leq i \leq k} \langle v_{n_{k+1}}, e_i \rangle e_i$ and

$$6. e_{k+1} = \frac{f_{k+1}}{\|f_{k+1}\|}.$$

By induction on k one can prove that the collection of all finite linear combinations of e_1, \dots, e_k is the same as the collection of all finite linear combinations of $v_1, v_2, v_3, \dots, v_{n_k}$. Thus, the collection of all finite linear combinations of the orthonormal set e_1, e_2, \dots, e_n is *dense* in \mathbf{V} , so this is an orthonormal basis.

The sum over k runs over the occupied orbitals, and the sums over p, q, r, s run over the functions in the basis set. The product in the second term on the right-hand side is given by:

$$\langle pr|g|qs\rangle = \int \int d^3r_1 d^3r_2 \chi_p^\star(\mathbf{r}_1) \chi_r^\star(\mathbf{r}_2) \frac{1}{\mathbf{r}_1 - \mathbf{r}_2} \chi_q(\mathbf{r}_1) \chi_s(\mathbf{r}_2). \quad (5.91)$$

Introducing a density matrix

$$P_{pq} = 2 \sum_k C_{pk} C_{qk}^\star = 2 \sum_k^{\text{occupied}} C_{pk} C_{qk}^\star \quad (5.92)$$

with factor 2 being due to the spin, the matrix F_{pq} can be written as

$$F_{pq} = h_{pq} + g_{pq} = h_{pq} + \frac{1}{2} \sum_{rs} P_{rs} (2\langle pq|g|qs\rangle - \langle pr|g|sq\rangle). \quad (5.93)$$

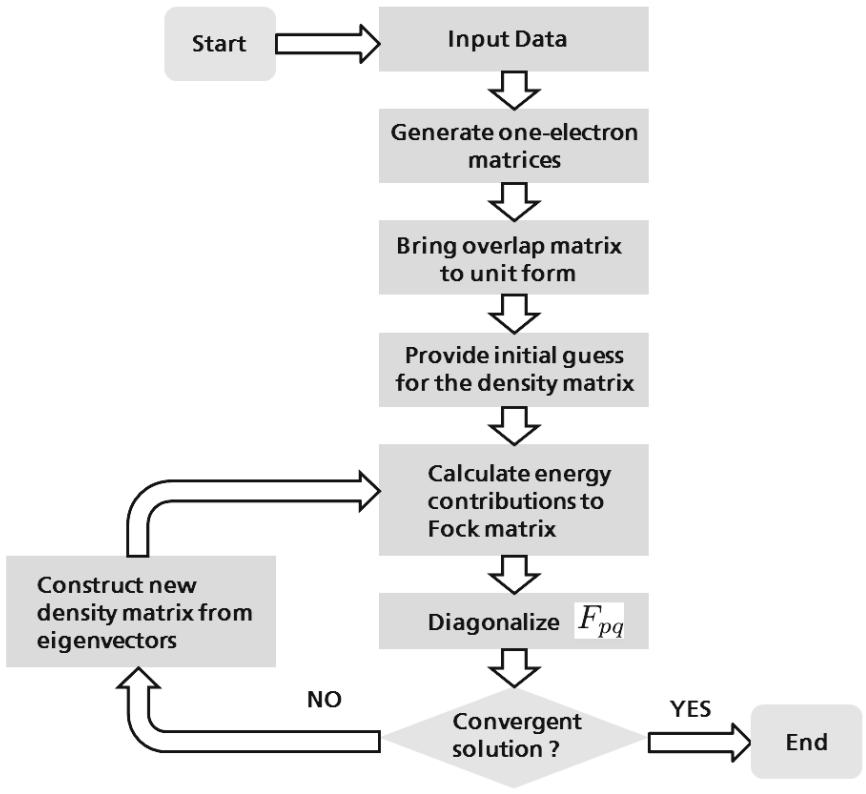
The energy is given by

$$E = \sum_{pq} P_{pq} h_{pq} + \frac{1}{2} \sum_{pqrs} P_{pq} P_{rs} \left[\langle pr|g|qs\rangle - \frac{1}{2} \langle pr|g|sq\rangle \right]. \quad (5.94)$$

In practical implementations, real orbital basis functions are used, so that complex conjugates can be removed in (5.89) to (5.92). A flow chart of the Hartree–Fock algorithm is displayed in Fig. 5.3 and in Algorithm 5.4 filling in some details.

Perturbation Methods

It is possible to treat the difference between the Hartree–Fock operator \mathcal{F} and the true Hamiltonian as a perturbation. This treatment of the Hartree–Fock approximation is called *Møller–Plesset (MP) perturbation theory* [377] which is very popular in

Algorithm 5.3: Flow chart of the Hartree–Fock algorithm

chemistry. Any $N \times N$ Slater determinant composed of the solution $\psi_k(\xi_i)$ is a valid N -body wave function in the Hartree–Fock theory. The ground state wave function is $\Psi_0^{(0)}$ with energy $E_0^{(0)}$, and the remaining Ψ_i^0 correspond to the excited states, if the ground state is not degenerate, and can be used in the perturbation expansion. Assuming that one wants to calculate the correction to the ground state total energy, the first-order correction is identical zero $E_0^{(1)} = 0$, because the expectation value of the true Hamiltonian $\langle \Psi_0^{(0)} | \mathcal{H} | \Psi_0^{(0)} \rangle$ is the same as that of the Hartree–Fock operator. The second-order correction to the ground-state total energy is given by

$$E_0^{(2)} = \sum_{i \neq 0} \frac{\left| \langle \Psi_i^{(0)} | \mathcal{U} | \Psi_0^{(0)} \rangle \right|^2}{E_0^{(0)} - E_i^{(0)}} \quad (5.95)$$

$$= \sum_{k' < l'}^{\mathcal{E}} \sum_{k < l}^{\mathcal{O}} \frac{|\langle \Psi_{\lambda'} \Psi_{\nu'} | \mathcal{U} | \Psi_{\lambda} \Psi_{\nu} \rangle - \langle \Psi_{\lambda'} \Psi_{\nu'} | \mathcal{U} | \Psi_{\nu} \Psi_{\lambda} \rangle|^2}{\varepsilon_{\lambda} + \varepsilon_{\nu} - \varepsilon'_{\lambda} - \varepsilon'_{\nu}}, \quad (5.96)$$

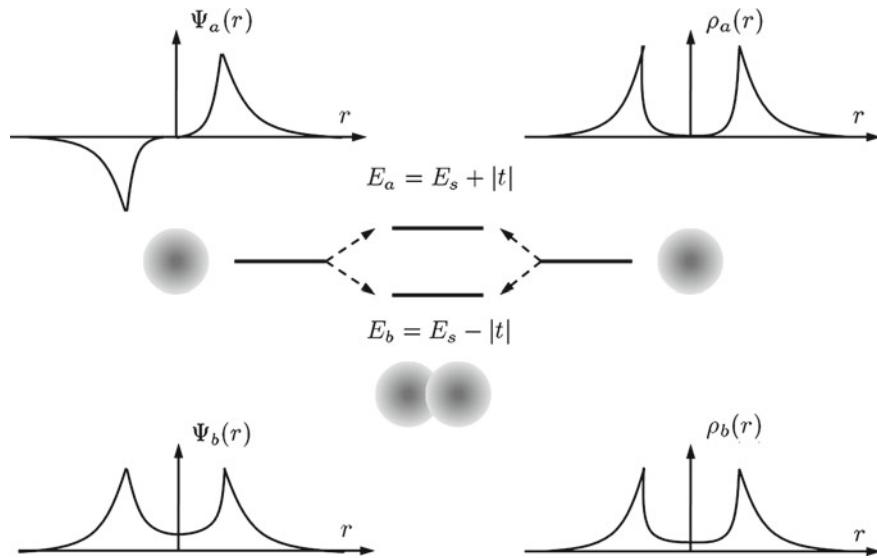


Fig. 5.3 Schematic splitting of the degenerate energy level E_s of the isolated atoms into symmetric (E_b) and antisymmetric (E_a) bonding states, the wave functions Ψ_a and Ψ_b along with the corresponding electron charge densities

Algorithm 5.4: Hartree–Fock algorithm

1. Input the atomic coordinates, the atomic number Z_n of the atoms, the total number of electrons and the basis set to be used.
2. Calculate all matrices that do not depend on eigenvectors, i.e. the overlap matrix S_{pq} , the one-electron Hamilton matrix H_{pq} in (5.90), and the two-electron integrals $\langle pr|g|qs \rangle$ in (5.91).
3. Bring overlap matrix V_{pq} to unit form.
4. The convergence speed of the scheme may depend crucially on the initial density distribution; therefore, a good initial guess is important.
5. Calculate the Coulomb energy contributions h_{pq} and exchange energy contributions g_{pq} to the Fock matrix according to (5.93).
6. Solve the generalized eigenwert problem (5.88) and construct a new density matrix (5.92) from the eigenvectors.
7. Perform a convergence test of the iterative solution. As a simple criterion one can use the change of energy levels or of density in subsequent iterations. Some more sophisticated methods for accelerating the convergence of the algorithm were proposed [366].

where the summations with symbols \mathcal{E} and \mathcal{O} run over the empty and occupied states, respectively. The calculation scheme in (5.95) is referred to as *MP3* and, similarly, the third- and fourth-order schemes are referred to as *MP3* [378] and *MP4* [379], respectively. It is not guaranteed that the perturbation series rapidly converge; in particular, when the difference between the “true” ground state and the Hartree–Fock ground state is large; additionally, the MP perturbation method is not

generally applicable to the calculation of excited states [380]. For excited states, the *configuration interaction* (CI) is used, which represents the many-electron wave function in terms of several Slater determinants, taking the electron correlation into account.

Other methods considering the electron correlation are the *independent electron-pair* (IEP) and *coupled cluster* (CC) approximation. Both theories can be reformulated in the framework of the *linked-cluster expansion* in perturbation theory. For a review see e.g. [381].

Limitations and Accuracy of the HF Method

Numerical implementations of the HF method have been widely used in quantum chemistry for molecules such as H_2 , H_2O , CH_4 , C_2 , H_6 , CO and NH_3 . For these molecules, the predicted equilibrium bond angles and inter-atomic distances are within a few percent of experiment, i.e. the geometry optimization usually works very well with the HF method. Also vibrational frequencies are often found to be within 10 % of the experimental data. Calculated values for the total energy of these molecules are less satisfactory, in particular for F_2 or O_3 molecules, as it turns out that they are very sensitive to the quality of the Gaussian basis set, even for a very large number of basis functions. The HF theory has also been used for solid state applications in different research codes, e.g. CRYSTAL 92. The HF method here turned out to be best suited for small organic molecules and compounds of main group elements which do not contain a large number of electrons and where the electrons are fairly localized. HF theory is less appropriate for systems with delocalized metallic-like states or for transition metals with high electron density.¹¹ In the *random phase approximation* (RPA) one can include correlations to correct the results, but this task becomes overly complicated for inhomogeneous systems.

The practical limit for geometry optimizations using the HF method is in the range of about 100 carbon-like atoms. SCF-based energies can be calculated for molecules containing several hundred atoms. The inclusion of correlations reduced the available system size to well below 100 atoms to just a few dozen atoms for the most accurate correlation methods. HF calculations are too slow for performing real dynamic simulations, even on the pico-second time range. The computational effort increases in essence with the fourth power of the number n_B of basis functions. However, for larger and more open structures, many of these integrals vanish and can be eliminated before diagonalizing the Hamiltonian matrices in the SCF scheme, so that for large molecules the effective effort scales with $\sim \mathcal{O}(n_B^3)$.

¹¹In fact, the HF theory breaks down for perfect metals, as it fails to account for the collective Coulomb screening in a completely delocalized electron system.

5.7 What Holds a Solid Together?

In solids, there are five types of bondings

- Ionic (e.g. $NaCl$)
- Covalent (e.g. diamond, silicon)
- Metallic (metals)
- Hydrogen (e.g. ice)
- Molecular or Van der Waals (Noble gas crystals at low temperatures)

All of these bonding types can be understood in terms of quantum mechanics – they arise naturally from the solution of the Schrödinger equation. Actually, the difference between ionic and covalent bonds is quantitative rather than qualitative.

5.7.1 Homonuclear Diatomic Molecules

In order to understand the origin of chemical bonding solids, we consider as an example a very simple system: the hydrogen molecule. Even for this simple system, there exists no analytical solution. In order to provide an insight into the physical behavior of this system, we provide in the following a simple molecular orbital analysis without using numerical methods. Let $|1\rangle$ and $|2\rangle$ be the electron ground states¹² in the first and second atom, respectively. These states can be written:

$$\mathcal{H}_1|1\rangle = E_s^0|1\rangle \text{ and } \mathcal{H}_2|2\rangle = E_s^0|2\rangle , \quad (5.97)$$

where \mathcal{H}_1 and \mathcal{H}_2 are the Hamiltonians for the isolated atoms 1 and 2.

In the following we assume that:

- States $|1\rangle$ and $|2\rangle$ form an adequate basis set to expand the ground state Ψ of the hydrogen molecule, i.e.

$$|\Psi\rangle = c_1|1\rangle + c_2|2\rangle . \quad (5.98)$$

- The states $|1\rangle$ and $|2\rangle$ are orthonormal, so that

$$\langle 1|1\rangle = \langle 2|2\rangle = 1 \text{ and } \langle 1|2\rangle = \langle 2|1\rangle = 0 . \quad (5.99)$$

This assumption breaks down when the atoms get close to each other. Actually, we can take the state overlap into account. However, it will complicate the calculations, whereas already the simplified model gives reasonable results.

- Coulomb interactions between the electrons are neglected.

¹²Actually, these are the $1s$ states.

The problem is to find the complex constants c_1 and c_2 and the energy E for the molecule. The Schrödinger equation for the molecule reads:

$$\mathcal{H}|\Psi\rangle = E|\Psi\rangle . \quad (5.100)$$

Inserting (5.98) in (5.100) yields:

$$\mathcal{H}(c_1|1\rangle + c_2|2\rangle) = E(c_1|1\rangle + c_2|2\rangle) . \quad (5.101)$$

Projecting (5.101) onto the basis states $|1\rangle$ and $|2\rangle$ and taking into account (5.99) one yields the following two equations:

$$E_s c_1 + \mathcal{H}_{12} c_2 = E c_1 , \quad (5.102a)$$

$$\mathcal{H}_{21} c_1 + E_2 c_2 = E c_2 , \quad (5.102b)$$

where $E_s = E_s^0 + \langle 1|V_2|1\rangle$ is not the same as E_s^0 , because the presence of the potential from the nearby second atom contributes an additional term. For nontrivial solutions of c_1, c_2 , one requires that the secular determinant be zero

$$\begin{vmatrix} E_s - E & \mathcal{H}_{12} \\ \mathcal{H}_{21} & E_s - E \end{vmatrix} = 0 \quad (5.103)$$

Expanding the determinant and solving the corresponding quadratic equation, one finds the solutions:

$$E_b = E_s + t , E_a = E_s - t , \quad (5.104a)$$

with the normalized eigenstates

$$|\Psi_b\rangle = \frac{1}{\sqrt{2}} (|1\rangle + |2\rangle) , \quad (5.105a)$$

$$|\Psi_a\rangle = \frac{1}{\sqrt{2}} (|1\rangle - |2\rangle) , \quad (5.105b)$$

The Hamilton operator for an electron in the molecule is

$$\mathcal{H} = -\frac{\nabla^2}{2} + V_1(r) + V_2(r) , \mathcal{H} = H_2 + V_1 , \quad (5.106)$$

where $V_1(r)$ is the electrostatic potential of the nucleus in atom 1 and $V_2(r)$ is the electrostatic potential of the nucleus in atom 2. As $\mathcal{H}_2|2\rangle = E_s|2\rangle$ and $\langle 1|2\rangle = 0$, one has

$$t = \mathcal{H}_{12} = \langle 1|H|2\rangle = \langle 1|V_1|2\rangle . \quad (5.107)$$

The potential V_1 is attractive to an electron and hence negative, whereas $|1\rangle$ and $|2\rangle$ are $1s$ states which are never negative. Thus, t is negative and the molecular state $|\Psi_b\rangle$ corresponds to the lowest energy state. It corresponds to a symmetrical combination of basis states, cf. Fig. 5.3. The two electrons with opposite spins in the H_2 -molecule occupy this molecular energy state. In the molecule the electronic contribution to the total energy is lower by $2|t|$ than for two isolated atoms which gives rise to cohesion. Thus, $|\Psi_b\rangle$ is called *bonding state*. In contrast, the molecular state $|\Psi_a\rangle$ has energy $E_a + |t| > E_s$. This corresponds to an antisymmetrical combination of basis states, cf. Fig. 5.3; since an electron in this state has an energy higher than E_s , its occupation

would not lead to bonding and thus, $|\Psi_a\rangle$ is called *antibonding* state. In Fig. 5.3 the electron densities ρ_b and ρ_a of the bonding and antibonding state are shown. These are given by:

$$\rho_b = 2|\Psi_b(r)|^2 = \rho_1(r) + \rho_2(r) + \rho_{\text{bond}}(r), \quad (5.108a)$$

$$\rho_a = 2|\Psi_a(r)|^2 = \rho_1(r) + \rho_2(r) - \rho_{\text{bond}}(r), \quad (5.108b)$$

with

$$\rho_{\text{bond}}(r) = 2\Psi_1(r)\Psi_2(r). \quad (5.109)$$

Thus, the interference of the single atomic wave functions $\Psi_1(r)$ and $\Psi_2(r)$ in the bonding state of the two atoms leads to a probability density of the electrons in between the atoms which is not zero. This is a purely quantum mechanical effect; in a classical calculation one would expect $\rho = (\rho_1(r) + \rho_2(r))/2$ for the total electron charge density. The interference of quantum mechanical wave functions delivers an explanation for the chemical homeopolar (covalent) bonding. Detailed calculations of the H_2 -molecule starting with the Hamiltonian (5.10) on p. 224 and solving the Schrödinger equation for the electrons in the effective potential of the stationary nuclei in terms of, show that there exists a minimum of the effective potential which is composed of the Coulomb repulsion of the nuclei and two attractive contributions, the so-called Coulomb integral and the exchange integral which is due to the interference of the electron's wave functions. In fact, the analysis carried out in this section shows how the tight-binding approach works, which is a semi-empirical method, because the forces acting on each atom are based on quantum mechanics, but in the construction of the Hamiltonian, empirical parameters are used. In essence, the multi-electron Hamiltonian is replaced with a parameterized matrix the elements of which (E_s, t) depend on the atom separation. The solution of the eigenvalue problem of this matrix provides the energies of occupied states. These energies provide a negative contribution to the total energy, whereas the Coulomb repulsion of nuclei counterbalances the attraction at an equilibrium separation of nuclei.

5.8 Semi-empirical Methods

Despite steady progress and developments in computer hardware and efficient algorithms, the simulation of large systems of particles with ab initio methods is still limited to a few hundred particles. For a treatment of larger systems using molecular dynamics, one can either use *tight-binding* (TB) or classical molecular dynamics. The TB method has the advantage of being quantum mechanical, i.e. it still provides information about the electronic structure of the system and therefore has higher accuracy. There are several simplified quantum mechanical computational techniques which are commonly called *semiempirical*. In the following, we will shortly discuss semi-empirical methods which are very common in the field of quantum chemistry, such as:

- MNDO
- AM1
- PM3
- PM5

The majority of these methods have been incorporated in several research program packages, available for free, e.g. into the popular code “Molecular Orbital PACkage” (MOPAC). All the above methods roughly have the same structure and are based on a *modified neglect of differential overlap* (MNDO). They are all self-consistent field methods – very similar to Hartree–Fock methods – but instead of rigorously calculating the Fock matrix elements, the computations of the Hamiltonian and overlap matrix elements are based on semi-empirical formulae with predefined coefficients, which take into account the electrostatic repulsion and the exchange interaction. Semi-empirical methods are a compromise between computational efficiency and the physical correctness of the applied approximations. The usefulness of these methods comes from an interplay between speed and accuracy, mathematical rigor and pragmatism, and a specific parameterization and generality. Semi-empirical methods allow for breaking of bonds and allows for calculating geometric structures, binding energies, electrostatic potentials, dipole moments and infrared spectra. A number of algorithmic simplifications such as the elimination of difficult integrals by mathematical transformations or by using them as parameters, or the use of minimal basis sets, avoid the expensive computational effort of first principles calculations, but also limits the transferability of semi-empirical results. The most important approximations introduced in semi-empirical methods are:

1. Only valence electrons are taken into account,
2. Three and Four center integrals are neglected,
3. Overlap integral in the Hartree–Fock Roothaan equation (5.88) are ignored,
3. Most matrix elements are approximated by analytical functions of interatomic separation and atom environment; parameters are chosen such that they reproduce the features of reference systems,
5. The Core–core Coulomb interaction is replaced with a parameterized function.

Most of the efforts in MNDO and related methods have been devoted to the description of organic molecules in chemistry. Hence, this approach works well for functional organic molecules but is sometimes confusing for other systems due to large scatter of the results obtained with different variants of MNDO. When applying this method to a new system one never knows how reliable the results are. Additionally, since the matrix elements have a rather complicated form, it is difficult to analyze the contributions of different parts of the Hamiltonian. The tight-binding methods, which has mostly been used in physical applications of carbon, silicon, and transition metals such as copper, provides – albeit being simpler – more insight into the physical behavior of the considered system.

5.8.1 Tight-Binding Method

The semi-empirical methods discussed in the previous section have a serious drawback: They usually have dozens of parameters which are chosen in a way as to reproduce known experimental results for reference systems. Sometimes the choice of parameters is not done with reasonable physical motivation which renders the evaluation of numerical results very difficult, in particular, when systems other than the reference ones are simulated. On the other hand, despite recent major development in computer hardware and in algorithms, the simulation of large systems of particles by ab initio methods is still limited to about a few hundred particles. In Sect. 2.3.1 of Chap. 2 the principle of Occam's razor was mentioned which, in essence, states that always the simplest representation of a model should be tried first which – at the same time – accounts for as many facts as possible, and that all obsolete parameters in a model should be avoided completely. Thus, the ultimate goal of an appropriate computational model is:

- To be as simple as possible and to have a minimum set of parameters,
- To be physically motivated,
- To be predictive enough to provide a qualitative and quantitative description of a system.

In light of these demands, the TB approach seems to be the optimal computational technique for simulating materials. The TB method works similarly to the MNDO method, replacing the many-body Hamilton operator \mathcal{H} with a parameterized Hamiltonian matrix according to formulae obtained from fits to ab initio data. One adopts a localized-orbital basis set which is called Linear Combination of Atomic Orbitals (LCAO). The analytic form of these basis functions is unknown; however, one can calculate the matrix elements of the Hamiltonian and thus derive the total energy and the electronic eigenvalues. A small number of basis functions is usually used which correspond roughly to the atomic orbitals in the energy range of interest. For example, when modeling graphite or diamond, the $1s$ orbitals are neglected, and only the $2s$ and $2p$ orbitals are considered. Next, the forces on the atoms are calculated using the Hellman–Feynman theorem (5.30), i.e. by taking the partial derivatives of the total energy with respect to the coordinates of the atoms. Finally, knowing the forces on the atoms, one can perform MD simulations using the predictor-corrector or the Verlet scheme. The limiting factor of these MD simulations is the time needed to diagonalize the Hamiltonian matrix and to compute the forces. The TB approach has been shown to work very well for covalently bonded systems (*Ga*, *Ge*, *Si*, *C*, *In*,) etc. and for transitional d-electron metals. The TB method is the minimum model that allows for a calculation of the atomic geometry of clusters or amorphous solids along with the electronic structure. The main advantage of the TB methods is that it is physically motivated and computationally efficient.

In principle, the TB model can be rigorously derived from density functional theory, i.e. a Kohn–Sham type Hamiltonian, see e.g. [382]. This means, that it is in essence a one-particle mean-field theory and was widely used after the pioneering work of Slater and Koster [383], who devised this method for calculations of band

structure energies of simple crystals. With the advent of DFT in the early 1960s and the advance of computer hardware at that time, ab initio calculations became the methods of choice. With new and accurate parameterizations based on ab initio calculations, one is able to make more quantitative predictions.

Similarly to the HFR method, the total energy E_{tot} is represented as a sum of the electronic energy and pair terms, i.e.:

$$\mathcal{H} = \sum_{ik} \varepsilon_{ik} c_{ik}^\dagger c_{ik} + \sum_{ij,kl} t_{ik,jl} c_{ik}^\dagger c_{ik}. \quad (5.110)$$

The indices i and j refer to the atoms and the Greek letters α and β to the atoms and the Greek letters α and β to the orbitals of these atoms. The on-site term $\varepsilon_{i\alpha}$ includes the atomic contribution, the effect of neighboring atoms, and, to some extend, the exchange-correlation potential. Strictly speaking, ε_{ij} is not the atomic eigenvalue, and includes, on average, the effects of the nearest neighbors. This is the source of one of the major problems with this theory, namely its reduced *transferability*. The second term represents hopping between the nearest neighbors and is parameterized as a function of the nearest-neighbor distance and according to the type of orbitals k and l involved. It contains contributions from the kinetic energy, the potential due to the ions and other electrons, and again some part of the exchange-correlation potential. In practice, the off-diagonal terms $t_{ik,jl}$ of \mathcal{H} are set to zero after a cutoff distance. The diagonal terms ε_{ij} and $t_{ik,jl}$ are usually determined by a fit of the TB eigenvalues to the ab initio ones [383].

Total TB Energy

In the usual TB formalism, the total energy and force calculation follow ab initio principles. In DFT formalism, the total energy can be written as:

$$E_{DFT} = 2 \sum_{\lambda} n_{\lambda} \varepsilon_{\lambda} + V_{ii} - \frac{1}{2} \int \frac{\rho(r)\rho(r')}{|r-r'|} d^3r d^3r' + E_{exc}[\rho] - \int v_{exc}(r)\rho(r)d^3r, \quad (5.111)$$

where V_{ij} is the ion–ion repulsive interaction and $n_{\lambda} = \theta(E_F - \varepsilon_{\lambda})$ is the occupation number of the eigenstate $|\lambda\rangle$ with θ being the Heaviside step function. Factor 2 accounts for spin degeneracy. The total energy in TB theory is written as the sum of the band energy E_{band} (sum of the electronic eigenvalues up to the Fermi level), and a short-range repulsive interaction replacing the repulsive interaction between the ions and their core electrons and the term correcting the double counting of the electronic Coulomb repulsion:

$$E_{TB} = 2 \sum_{\lambda} n_{\lambda} \varepsilon_{\lambda} + E_{repulsive}(R_1, \dots, R_N). \quad (5.112)$$

The band term is usually attractive and is responsible for the cohesion of molecules and solids, the only exception being for closed-shell atoms such as Helium, where there is no energy gain even after hybridization of $1s$ orbitals. In general, as atoms get closer to each other, the hybridization becomes stronger, and since the lowest orbitals are occupied, the energy of the bonding orbitals decreases, hence the attractive nature

of the band term. Expanding the eigenstates in the localized basis $|\lambda\rangle = \sum_i c_i^\lambda |i\rangle$ one obtains:

$$\begin{aligned} E_{band} &= 2 \sum_\lambda n_\lambda \varepsilon_\lambda = 2 \sum_\lambda \langle \lambda | H | \rangle \rangle \\ &= 2 \sum_\lambda \sum_{ij} n_\lambda c_i^{\lambda\star} c_j^\lambda \mathcal{H}_{ij} = 2 \sum_{ij} \rho_{ij} \mathcal{H}_{ij} = 2 \text{Tr}(\rho \mathcal{H}), \end{aligned} \quad (5.113a)$$

where the elements of the density matrix are defined as

$$\rho_{ij} = \sum_\lambda n_\lambda c_i^{\lambda\star} c_j^\lambda. \quad (5.114)$$

All ground state quantities can in fact be expressed with the density matrix, since it contains as much information as the eigenvectors.

Forces

For the calculation of the forces one can employ the Hellmann–Feynman theorem (5.30). In a matrix notation the Schrödinger equation is written as follows:

$$\sum_j H_{ij} c_j^\lambda = \varepsilon_\lambda \sum_j S_{ij} c_j^\lambda, \quad (5.115)$$

where S represents the overlap matrix.

To compute the force on an atom located at R , one must take the derivative of (5.112) with respect to R , i.e. $F_i = -\partial E / \partial R_i$. The repulsive term is a classical potential and taking its derivative is straightforward; thus, we discuss only the band term $2 \sum_\lambda n_\lambda \varepsilon_\lambda$. From (5.115) one obtains

$$\varepsilon_\lambda = \sum_{ij} c_i^{\lambda\star} \mathcal{H}_{ij} c_j^\lambda. \quad (5.116)$$

Taking the derivative with respect to R yields

$$\frac{d\varepsilon_\lambda}{dR} = \sum_{ij} c_i^{\lambda\star} \left(\frac{d\mathcal{H}_{ij}}{dR} - \varepsilon_\lambda \frac{dS_{ij}}{dR} \right) c_j^\lambda. \quad (5.117)$$

From this relation one obtains the contribution of the band term to the force:

$$F_{band} = -2 \sum_{ij} \left(\frac{d\mathcal{H}}{dR} \rho_{ij} - \frac{dS_{ij}}{dR} w_{ij} \right), \quad (5.118)$$

where w is the energy-weighted density matrix $w_{ij} = \sum_\lambda n_\lambda \varepsilon_\lambda c_i^{\lambda\star} c_j^\lambda$. Thus, if the matrix elements of the Hamiltonian and overlap matrices are known explicitly as a function of the atomic coordinates R , one can calculate the forces on the atoms, provided that the vectors c are the exact eigenvectors of \mathcal{H} . For further details on the TB method concerning special parameterizations using self-consistent DFT calculations, e.g. Porezag's et al. non-orthogonal tight-binding, the reader is referred to the primary literature [384]. The *Density Functional Tight Binding* (DFTB) method has been demonstrated to work extraordinarily well for a broad range of materials, e.g. polar semiconductors, or heteronuclear molecules [385].

Linear Scaling Algorithms

The “brute force” and direct way to solve the Schrödinger equation for electrons is the direct diagonalization of the eigenwert matrix. This task is of order $\mathcal{O}(N^3)$, where N is the number of considered atoms, i.e.

$$T_{\text{CPU}} \propto N^3. \quad (5.119)$$

An alternative way to find the eigenvalues and eigenfunctions of the electronic Hamiltonian is to minimize the total energy with respect to the electronic degrees of freedom by steepest descent, cf. Algorithm 5.1, or conjugate-gradient methods, cf (5.15). These algorithms are faster than direct diagonalization, these are still $\mathcal{O}(N^3)$ methods and require orthogonalization of the eigenfunctions. Only in the case of a band or trigonal matrix, efficient libraries can be used for the task, which reduce the effort to $\mathcal{O}(N^2)$ [353]. Many different order- N methods have been put forth during the past 20 years [63, 386–388]. Thus, there exists a crossover point N^* below which using a linear scaling algorithm is not advantageous any more. This corresponds to point n_O in Fig. 2.30 on p. 103. For TB models, this crossover point is about $N^* \approx 100$ atoms. Hence, for systems with more than N^* atoms, the $\mathcal{O}(N)$ methods are superior in terms of efficiency and thus should be used. Recent publications have shown evidence, that the linear scaling algorithms can successfully be implemented into the DFTB model [389], which should increase the applicability of this method. The different linear scaling methods have been reviewed by Goedecker and Colombo [390].

Example: Tight-Binding Simulation of Nanophase Ceramics

A simple example for an efficient application of the tight binding method is the simulation of nanophase Silicon-carbide (n-Sic) ceramics. Due to their unique properties such as larger fracture toughness and higher sinterability than conventional ceramics, these materials have rapidly gained a good deal of attention. Understanding the atomic organizations and the electronic properties of grain boundaries in n-SiC are very promising for both, electronic and structural applications under extreme conditions. A theoretical investigation of these material properties requires methods for multi-level descriptions of the physical hierarchy in nanophase systems, from local electronic/atomistic dynamics to the mesoscopic grain dynamics. Usually, the grain dynamics cannot be resolved with atomistic methods, and on the other hand, simulating grain dynamics with continuum methods cannot gain insight into the diffusivity behavior at grain boundaries. In Fig. 5.4 a tight-binding molecular dynamics (TBMD) simulation of neck formation during a collision between aligned and misaligned SiC-clusters is shown. A velocity Verlet algorithm with a timestep corresponding to 1 fs was used. The system was first relaxed by the steepest descend method. The detailed dynamics of neck formation at high temperatures is an important issue because it is assumed that this influences not only the mechanical stability of the grain boundaries but also carrier transport properties.

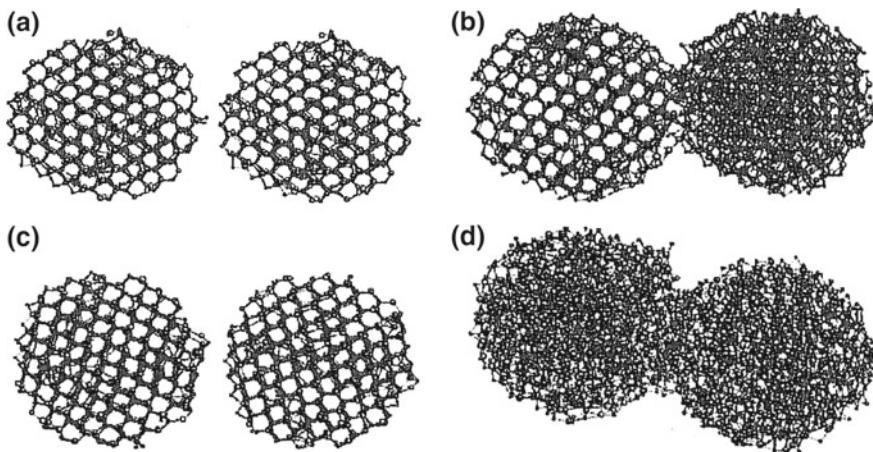
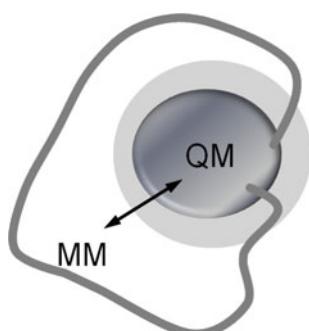


Fig. 5.4 Tight-binding simulation of aligned SiC nanospheres with 1008 atoms (504 Si atoms and 504 C atoms); **a** initial configuration, **b** Neck formation after collision (6 ps) at $T = 1500\text{ K}$. **c** Same as in (a) but with misaligned nanospheres. **d** Configuration after collision (8 ps) at 1500 K . Figures republished with permission. Compiled from [391]

5.9 Bridging Scales: Quantum Mechanics (QM) - Molecular Mechanics (MM)

With biomolecules such as proteins, the chemical reactions are very often restricted to an active center, cf. Fig. 5.5. However, the rest of the molecule has to be taken into account, in order to provide a suitable surrounding for the chemical reaction to take place and to produce the desired products. For such systems, often so-called *QM-MM* methods are used. Here, one describes the chemical reactions quantum mechanically, and the rest of the molecule is described using classical equations. The coupling of both methods is done by using *bridging atoms*, the movements of which is governed by both, the quantum mechanical equations, and the classically calculated forces.

Fig. 5.5 QM-MM coupling principle



Sometimes, one used pseudo-potentials for the coupling region instead of particles. Thus, this region has to be far enough away from the reactive center.

5.10 Concluding Remarks

In this chapter, we discussed the most fundamental aspects of computational chemistry and physics on the atomic scale. In general, there is a great many different methods that have been successfully applied and which are implemented in several ab initio research codes or commercial program packages. On the quantum scale, it is necessary for an efficient use of the various methods to understand very well the approximations made and their intrinsic limitations. This is very contrary to commercial standard software packages¹³ treating the macroscale with implementations of the finite element methods. Here, many standards in the material models based on continuum theory, have been established. With quantum chemistry methods, such simple standard do not exist. Over the past five decades, ab initio methods which are primarily used in quantum chemistry, have became an essential tool for the study of atoms and molecules, and – in recent years – also a modeling tool for the complex structures that arise in biology and in solid states in materials science. Ab initio methods are fundamentally based on the solution of the non-relativistic Schrödinger equation and the adiabatic Born–Oppenheimer approximation which allows for separating the dynamics of the electrons and the nuclei. The Schrödinger equation is a many-body problem and thus, a brute-force solution is intractable.

Two main theoretical approaches are used: Wavefunction-based approaches write the electronic wavefunction as a sum of Slater determinants whose orbitals and coefficients are optimized using different numerical procedures. Hartree–Fock methods provide solutions using a mean field approach, but do not provide a robust description in the treatment of electron correlations.

The second class of theoretical approaches are based on density functional theory (DFT), following the Hohenberg–Kohn theorem, which allows for expressing the total energy of the system as a functional of the electron density. Here, the class of gradient-corrected functionals such as BYLYP, or mixed functionals, such as BLYP3 [392] have been extensively developed. Møller–Plesset perturbation theory of second order (MP2) treats the electron correlation as small perturbation of the undisturbed HF-system. The various orders of perturbation scale as

- MP2 $\propto N^2$,
- MP3 $\propto N^6$,
- MP4 $\propto N^7$,
- MP5 $\propto N^8$,
- MP6 $\propto N^9$.

¹³For example LSDyna, pamcrash or Autodyne.

MP4 typically yields up to 98 % of the correlation energy; thus, in practice, higher order schemes are uninteresting and also tend to diverge.

For investigations of reactive chemistry in medium to large systems, DFT is at present the best approach. MP2 methods are generally useful for studying non-bonded interactions in small large systems and in developing molecular force fields for biological and materials science problems. Numerically, there are two main implementation approaches for solving the Kohn–Sham equations. One approach, primarily common in the physics community, represents the electron density orbital by a plane wave basis set. This approach is naturally suited for periodic condensed-phase systems, such as solids or fluids, in which periodic boundary conditions (implicit in the plane wave formulation) can be used. Plane-wave approaches have been implemented by two main algorithmic approaches. Car–Parinello methods perform ab initio molecular dynamics and evolve the system in phase space by means of an extended Lagrangian. This approach is very suited for studying liquids and molten solids. On the other hand, conjugate gradient methods have been implemented in the commercial program CASTEP [393] (Cambridge Serial Total Energy Package) and in the research code VASP (Vienna Ab initio Simulation Package), which is available with a free licence.

The use of localized, atom-centered Gaussian basis sets which have been used for years in solving the HF equations, is an alternative to the use of plane wave methods. These methods are e.g. implemented in the commercial package GAUSSIAN. Typical properties that can be calculated using these methods in materials science, are e.g. energy bandgaps, optical properties, structures, spectroscopic quantities, or enzymatic systems in biology.

Abstract

On the atomistic and microscopic scale, the possible conformations (the spatial arrangements) of particles – besides their energy – begins to play a fundamental role. This is why after an introduction in Sect. 6.1 we discuss statistical physics and thermodynamics in Sect. 6.2. Classical interaction potentials (that may or may not be derived from quantum mechanical ab initio methods) are treated in Sect. 6.3, which brings us to the important MD method, the nuts and bolts of which are discussed in detail in Sect. 6.4. Here, we provide many tricks of the trade of this method. A discussion of liquids, soft matter and polymers in Sect. 6.5 is then followed by a short discussion of the Monte Carlo method in Sect. 6.6. A couple of problems are provided at the end of this chapter.

6.1 Introduction

In this chapter we are going to discuss numerical methods for the simulation of many particle systems that are based on the classical dynamics of many atoms or molecules without taking into account the electronic structure of the constituents. For many materials, fluids and gases this is an excellent approximation.¹ When studying the properties of many particle systems, one has to be aware that most properties that are measured in real experiments cannot be obtained directly from simulations. For example, a simulation of a molecular liquid provides all positions x^i , momenta p^i and forces f^i acting on all particles at each point in time of the simulation; this

¹Whether or not a quantum mechanical treatment of a system is necessary, depends on the time and energy scale of the processes one is interested in. For example, in the case of C – C-bond fluctuations or the rotational motion of light molecules – such as H₂ – with high frequencies $h\nu > k_B T$ one should use a quantum treatment of molecular motion.

detailed information of a system's state however is never available in experimental systems and thus the simulation data cannot directly be compared to experiment. On the other hand, when conducting real measurements, one must perform repeated experiments and average the results, i.e. one has to assume probability distributions for the measured macroscopic observables. If one performs k independent measurements of some observable A , the mean value is computed as

$$\langle A \rangle = \frac{1}{k} \sum_{i=1}^k A_i. \quad (6.1)$$

All experiments therefore measure *average* properties and in order to benefit from simulations, one has to adopt the same approach as in the experiment, i.e. one has to be aware of which kind of averages one wants to compute in a numerical experiment.

In essence, two procedures for N -particle systems were developed: Monte-Carlo (MC) which is aimed at generating the important probability distributions which are in physics, and Molecular Dynamics (MD) simulations of different ensembles usually based on the classical equations of motion for individual particles (cf. our general discussion in Chap. 1). For ab-initio MD methods based on the Schrödinger equation, see Chap. 5. Both simulation techniques are based on methods of statistical physics which are discussed beginning with Sect. 6.1.1.

With MD, a number of particles is contained in a (usually) cubic box and periodic boundary conditions are often applied. As a rule, it is generally assumed that the total potential of the particles can be written as a sum of pair potentials. The dynamics of such an N -particle system is contained in the Hamiltonian equations of motion in the phase space of generalized coordinates q^i and momenta p_i , which are written as vectors in tangent space $T_P(\mathcal{M})$ and as one-forms in the dual space $T_P M^*$, respectively.² The MD method for classical systems is discussed in Sect. 6.4.

With MC calculations a series of configurations of the N -particle system (microstates of the system) is constructed. Starting with an initial configuration the system is developed from state to state by an algorithm that calculates a certain acceptance rate based on the energy difference of the system in the old and new (trial) state. In the simplest form of this algorithm, only trial moves which lead to an improvement of the energy condition, i.e. which lead to a decrease in energy, are accepted, cf. Algorithm 2.6 on p. 89. This method is discussed in Sect. 6.6.

The N -body problem leads to a coupled system of ordinary differential equations and is analytically insoluble for $N \geq 3$. In the following Example 37 a short discussion of the general three-body problem is provided.

Example 37 (The general three-body problem)

The classical Newtonian three-body problem occurs in nature exclusively in an astronomical context. It is most convenient to work in the *center-of-mass* (CM) system, with \mathbf{r}^i denoting the position of mass m_i . The classical equations of motion

²For mathematical details concerning one-forms and tangent vectors, see Sect. 3.3 in Chap. 3 on p. 126.

can be written in this system as:

$$\ddot{\mathbf{r}}^i = -Gm_j \frac{\mathbf{r}^i - \mathbf{r}^j}{|\mathbf{r}^i - \mathbf{r}^j|^3} - Gm_k \frac{\mathbf{r}^i - \mathbf{r}^k}{|\mathbf{r}^i - \mathbf{r}^k|^3}, \quad (6.2)$$

where indices i, j, k stand for 1, 2, 3 and the two ordered permutations of these indices, thus obtaining 18 first order scalar differential equations. The CM condition

$$\sum_{i=1}^3 m_i \mathbf{r}^i = 0 \quad (6.3)$$

and its first derivative

$$\sum_{i=1}^3 m_i \dot{\mathbf{r}}^i = 0 \quad (6.4)$$

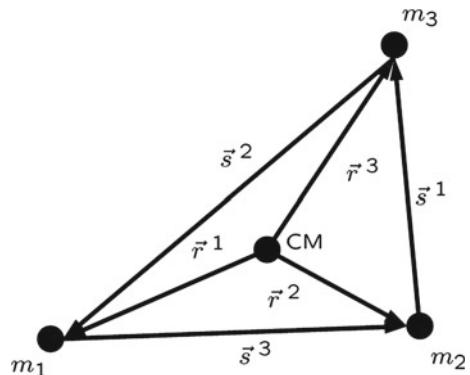
are six constraints that reduce the order of the system to 12. If there are no external forces or torques, the energy and angular momentum are conserved, i.e. they are integrals of motion. Thus, the order of the system is further reduced to $12 - 4 = 8$. Even if the motion of the system was restricted to a plane fixed in space, the order is to 6 which renders this problem unsolvable in general. The unsolvability of the three-body problem in the sense that it does not admit enough analytic integrals that can be integrated, was proved by Poincaré. Equation (6.2) can be written in a canonical (symmetric) form when introducing the relative position vectors $\mathbf{s}^i = \mathbf{r}^j - \mathbf{r}^k$, labeled in such a way, that \mathbf{s}^i is the side opposite to the vertex of the triangle containing the mass m_i (cf. Fig. 6.1) and that

$$\mathbf{s}^1 + \mathbf{s}^2 + \mathbf{s}^3 = 0. \quad (6.5)$$

Expressed with the relative position vectors, the equations of motion (6.2) adopt the symmetrical form

$$\ddot{\mathbf{s}}^i = -GM \frac{\mathbf{s}^i}{s^{i3}} m_i \mathbf{G}, \quad (6.6)$$

Fig. 6.1 Position and relative position vectors of the three-body problem in the CM system



where $M = m_1 + m_2 + m_3$ is the total mass and vector \mathbf{G} is a shorthand for

$$\mathbf{G} = \sum_{i=1}^3 \frac{\mathbf{s}^i}{s^i} \cdot \frac{s^i}{s^i} \cdot \frac{s^i}{s^i}. \quad (6.7)$$

The first term on the right-hand side of (6.6) is the usual term in the standard two-body Kepler problem, which leads to conical sections as orbital solutions. The second term however couples the equations for s^i and renders an analytical solution impossible.

If all particles are collinear, then all vectors \mathbf{r}^i , \mathbf{s}^i and \mathbf{G} are proportional to each other. Assuming that m_2 lies in between the other two masses, then one can write:

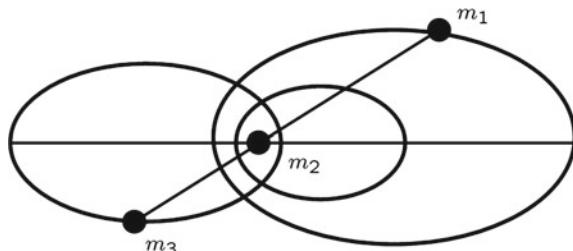
$$\mathbf{s}^1 = \lambda \mathbf{s}^3; \quad \mathbf{s}^2 = (-1 + \lambda) \mathbf{s}^3, \quad (6.8)$$

where λ is some positive scalar. After some algebra, one obtains as equation of motion for e.g. \mathbf{s}^3 the following two-body equation

$$\ddot{\mathbf{s}}^3 = -\frac{m_2 + m_3(1 + \lambda)^{-2} GM \mathbf{s}^3}{m_2 + m_3(1 + \lambda)(s^3)^3}, \quad (6.9)$$

and likewise for \mathbf{s}^2 and \mathbf{s}^1 . Parameter λ obeys a fifth degree polynomial with one single positive real root as a function of the three masses. As a result, the particles move along confocal ellipses of the same eccentricity and the same orbital period around the common center of mass; the masses are always lined up by distances obeying (6.8), cf. Fig. 6.2. From a principal point of view, the most important points of a differential equation (e.g. for a vector field) are its rest points or its zero solutions. The N -body problem does not have such rest points, that is no “static” solution where all or some points just stay motionless in space. Due to symmetries (e.g. due to rotation or scaling), there are solutions such that the all interbody distances remain constant. These solutions are called *central configurations*. For the three-body problem there are exactly five such configurations up to symmetry. The motion on collinear ellipses discussed above and found by Euler (1767) are three of these solutions and they were the first and simplest periodic exact solution of the three-body problem (6.6). The other two central configurations were discovered by Lagrange (1772) and they have the shape of equilateral triangles, see e.g. Goldstein [295]. In the 20th century these so-called *Lagrange points* were discovered to actually exist in the solar system. The collinear and the equilateral triangle solutions are the only explicit solutions known for arbitrary masses and a handful of solutions for special cases are also known [395].

Fig. 6.2 Euler’s collinear solution for masses $M_1 : m_2 : m_3 = 1 : 2 : 3$ [394]



6.1.1 Thermodynamics and Statistical Ensembles

Usually, the quantities of interest with many particle systems, that is $N \sim 10^{23}$, are not the particle trajectories themselves but rather (macroscopic) structural, static, dynamic and thermodynamic quantities, expressed in so-called *state variables* and obtained from the atomic particle dynamics that characterize *uniquely* the macroscopic state of a system as a whole, e.g. temperature T , entropy S , pressure p , or the internal energy E . Relations between these state variables are called *equations of state* which – in thermodynamic theory – can only be determined empirically, i.e. by measurements, and which are only valid for specific physical systems.

The physical theories that explain how to derive classical macroscopic state variables and equations of state, are classical statistical mechanics and (equilibrium) thermodynamics, the foundations of which were developed by Gibbs, Carnot, Boltzmann, Ehrenfest, and others in 19th century. The formalism of statistical mechanics allows the straightforward analytic treatment of only a few “simple” systems. The main motivation behind molecular simulation is to apply the principles of statistical mechanics and thermodynamics to the prediction of the macroscopic behavior of “complicated” many particle systems.

*Thermodynamics*³ provides a *phenomenological* description (as classical electrodynamics or continuum mechanics), which is very general and therefore used in many areas of physics and technical engineering applications, i.e.:

1. There are only quantities used as elements of the theory, which are directly measurable and no speculations on the microscopic origin of these quantities are done.
2. There are only relations between measurable quantities.

Thus, thermodynamics is based exclusively on (macroscopic), experimental, empirical facts which are formulated in basic *Laws of Thermodynamics*:

- 0th Law: The existence of a state variable called temperature T is postulated such that if two bodies B_1 and B_2 are at thermal equilibrium with a third body B_3 , then B_1 is at thermal equilibrium with B_2 (transitivity property of temperature).
- 1st Law: Heat Q is a form of energy; along with other forms of energy W in a system, the law of conservation of energy E is valid in the form $dE = \delta W + \delta Q$.
- 2nd Law: Heat is a special form of “unordered” energy which cannot be transformed completely into other forms of energy, while on the other hand all other forms of energy can be transformed completely into heat.
- 3rd Law: The entropy $S = 0$ at $T = 0$.

³Actually, this term is misleading, as thermodynamics is an equilibrium theory which has little to say about processes far away from equilibrium.

Note that heat δQ and work δW which are exchanged with the system's surrounding, are denoted with δ and not a complete differential d , as the result of such an exchange depends on the path of the process. The laws of thermodynamics have the status of *basic axioms* from which all basic relations of thermodynamics can be recovered deductively. Thermodynamic theorems are only valid in the *thermodynamic limit*, i.e. for *many* particle systems and for systems at equilibrium. For microscopic systems, where the dynamics of individual particles becomes important, thermodynamics is not valid any more, due to local fluctuations.

The second possible approach to treat N -particle systems is *statistical mechanics*.⁴ Here, it is tried to justify the thermodynamic laws – valid for macroscopic systems – by considering microscopic particle variables. In the terminology of classical mechanics, one has not only to solve the equations of motion for at least 10^{23} particles, but also the same number of particles in a system would have to be prepared in their initial state and then be measured. For quantum mechanical many particle systems, the same problems arise: here, one has to determine the state vector $|\psi\rangle$ for a system with N particles. Hence, the detailed initial state, based on the phase space coordinates of a classical system and the state vector for a quantum mechanical system respectively, is unknown.

Therefore, instead of trying to investigate only *one* system, one considers *many* systems for which certain *macroscopic* observables, the state variables such as energy E , pressure p , temperature T , particle number N , or volume V have been fixed identically. The entity of all these systems is called an *ensemble* or *Gibb's ensemble* and one is interested in their mean values, i.e. the most likely values of certain state variables for a given ensemble.

The axiomatic basis of statistical thermodynamics is given by the axioms of classical mechanics and the axioms of probability calculus. Based on these foundations⁵ one can derive theorems on *statistical ensembles*. The methods of statistical mechanics can as a matter of principle be applied to both, equilibrium and non-equilibrium states.

Before we start with a simple MD case study in Sect. 6.4, we briefly review some of the basics of statistical mechanics and thermodynamics in the next section. In particular, the following questions will be answered:

1. What does it mean to measure a quantity in an observation?
2. How does one calculate probabilities?

⁴Also often denoted as *statistical thermodynamics*.

⁵Note that due to the time reversibility of the microscopic mechanical equations, the question arises why at all a physical equilibrium state of a system is reached.

6.2 Fundamentals of Statistical Physics and Thermodynamics

6.2.1 Probabilities

Measurements of equilibrium properties from simulations using the molecular dynamics or Monte Carlo techniques, which are thermodynamic in nature can be considered as exercises in numerical statistical mechanics. The link between simulation and experiment is done by treating the results of computer experiments in the same way as data of a real experiment, i.e. one performs an error analysis of fluctuating results of measurements. Typical properties, that are measured in simulations are

- Transport Coefficients,
- Correlation Functions,
- Pair Distribution Functions,
- Structural Parameters.

Considering a series of n measurements of some fluctuating property A with a distribution P_A , respectively $P(A)$ in the continuous case; then the average of A is given by

$$\langle A \rangle = \sum A P_A , \quad (6.10)$$

and

$$\sum P_A = 1 \quad (6.11)$$

in the discrete case and

$$\langle A \rangle = \int A P(A) dA , \quad (6.12)$$

with

$$\int P(A) dA = 1 . \quad (6.13)$$

in the continuous case. The integration, respectively the summation has to be done for all values of A , for which the distribution P_A , respectively $P(A)$ is defined. Fluctuations are defined as the deviations from the average value of a statistical quantity $A(t)$. For the average of A , one obtains

$$\langle A \rangle = \frac{1}{n} \sum_{i=1}^n A_i , \quad (6.14)$$

and if each measurement A_i is independent, it has variance

$$\sigma^2(A) = \frac{1}{n} \sum_i (A_i - \langle A \rangle)^2 = \langle A^2 \rangle - \langle A \rangle^2 . \quad (6.15)$$

The variance of the mean is

$$\sigma^2(\langle A \rangle) = \sigma^2(A)/n . \quad (6.16)$$

If, as is usually the case in MD (and also other) simulations, the assumed dependence of the A_i is unwarranted, $\sigma^2(\langle A \rangle)$ is liable to be underestimated because the effective number of independent measurements is considerably less than n . This effect of autocorrelated measurements can be seen by rewriting the variance as

$$\sigma^2(\langle A \rangle) = \frac{1}{n} \sigma^2(A) \left[1 + 2 \sum_i \left(1 - \frac{i}{n}\right) C_i \right], \quad (6.17)$$

with the *autocorrelation function*

$$C_i = \frac{\langle A_i A_0 \rangle - \langle A \rangle^2}{\langle A^2 \rangle - \langle A \rangle^2}. \quad (6.18)$$

The determination of statistical averages using MD simulations, is particularly advantages also for *non-equilibrium* phenomena, because in a non-equilibrium situation, averages can still be obtained from (6.10). Interesting examples of non-equilibrium MD simulations are investigations of crack propagation in solids [396] or shock-wave structures [43, 397, 398].

6.2.2 Measurements and the Ergodic Hypotheses

In a typical measurement, the value of the observable $A(q^i, p_i)$ is monitored and measured during a certain time interval Δt . If, in a time interval $\tau = k \Delta t$ the system has been measured k times in time intervals of length Δt , the arithmetic average of A is given by

$$\langle A \rangle_\tau = \frac{1}{\tau} \sum_{n=1}^{n=k} A(k \Delta t). \quad (6.19)$$

The mechanical state of a system which is to be considered at time t can be described by the position of all particles pertaining to this system. Suppose we want to describe an N -particle system at a time $t = t_1$ in a Cartesian coordinate system. A snapshot of this system yields a distribution of mass points in the coordinate system. Each point can be assigned to a vector $\mathbf{r}_i(t)$. With N points index i is running from 1, 2, ..., N . Thus, the position of mass points of the mechanical system at time $t = t_1$ is fully characterized but not so for other points in time. As the quantities $\mathbf{r}_i(t)$ are time dependent, one needs the additional information about the velocities $\dot{\mathbf{r}}$, respectively momenta \mathbf{p} . In analogy to our consideration above, in statistical physics the mechanical state of a system is described in a N -dimensional configuration space ($\{\mathbf{q}^N \mathbf{p}_N\} = (\mathbf{q}^1, \mathbf{q}^2, \dots, \mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)$, called phase space which is spanned by the coordinate axes q^i and the momenta axes p_i .

When calculating averages of the measured quantity A in (6.19), the duration of measurement and the stepsize Δt have to be small enough, such that all relevant values of $A(t)$ are considered when averaging. The resulting value is called the *time average* of A :

$$\langle A \rangle_t = \lim_{t \rightarrow \infty} \frac{1}{\tau} \int_0^\tau A(q^i(t), p_i(t)) dt. \quad (6.20)$$

On the other hand, the *ensemble average* of a system, according to (6.10) is given by:

$$\langle A \rangle_e = \int \int A(q^i(t), p_i(t)) \rho(q^i(t), p_i(t)) dq dp . \quad (6.21)$$

According to the *Ergodic Hypotheses*, if a system is at equilibrium, it comes arbitrarily close to any point in phase space and thus the time average obtained by repeatedly measuring the same system, and the average obtained from measuring each system of an ensemble just once, is the same, i.e.:

$$\langle A \rangle_t = \langle A \rangle_e . \quad (6.22)$$

The Ergodic Hypotheses is used in both, MC and MD calculations of averages in different ensembles. In Fig. 6.3 a simulation of an ideal gas consisting of hard spheres is shown. After removing the barrier between the two boxes the gas irreversibly

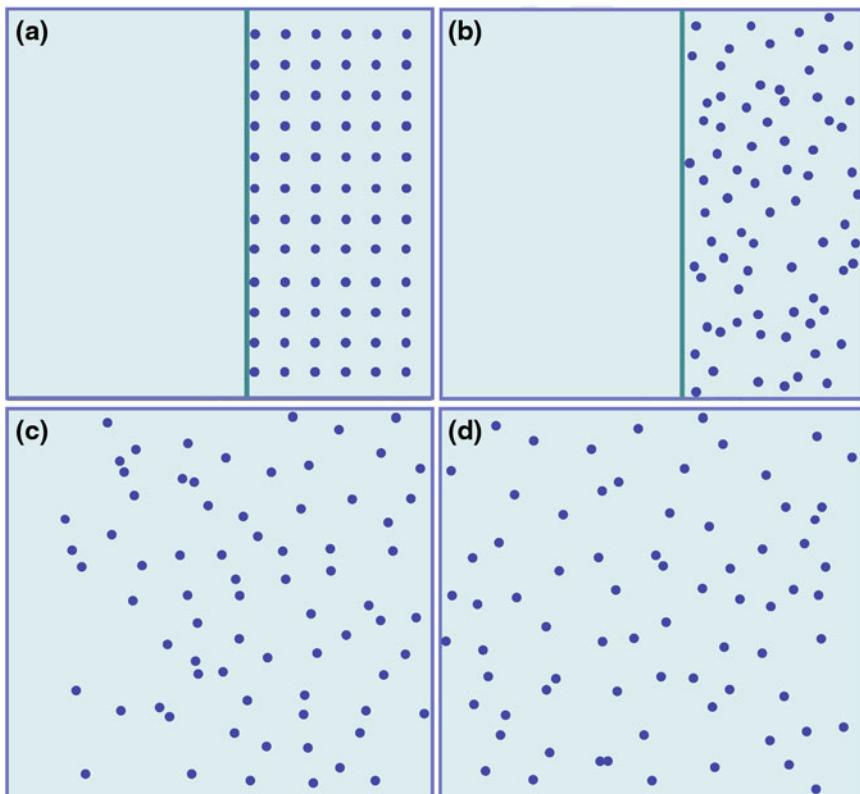


Fig. 6.3 Illustration of the irreversible expansion of an ideal gas in a 2-dimensional MD simulation of hard spheres. **a** Initial configuration with all particles aligned in the *left* chamber of the simulation box. **b** Simulation after 1 reduced LJ timestep. **c** Expansion of the molecular gas after removal of the separating wall. **d** Situation after 1 reduced LJ timestep after wall removal

expands into the full volume and after while, the density and the distribution of particles in the volume is no different from the original situation where the gas was confined in the left part of the box (situation (b) in Fig. 6.3). Thus, the system is at equilibrium, and many consecutive measurements of the system's properties can be made to provide information about the system in terms of mean values.

The number of states in a microcanonical ensemble for a classical system where the energy is characterized by a delta-function, is given by

$$\Omega(N, V, E) = \frac{1}{h^{3N} N!} \int \int d\mathbf{q}^N d\mathbf{p}_N \delta[\mathcal{H}(\mathbf{q}^N, \mathbf{p}_N) - E]. \quad (6.23)$$

A *microstate* of a system in *quantum theory* is given by a full specification of all of its degrees of freedom. The system may be defined, e.g. as having N degrees of freedom in a volume V . Microstates are generally denoted by quantum numbers and the energy eigenstates (the expectation values) are calculated by solving the time independent Schrödinger equation, which is nothing else than the following eigenvalue equation of the Hamiltonian \mathcal{H}

$$\mathcal{H}|\nu\rangle = E_\nu|\nu\rangle. \quad (6.24)$$

$|\nu\rangle$ is a shorthand notation for the system's wave function in state ν , and E_ν is the energy of state ν . The number of distinct microstates of a system with $N \sim 10^{23}$ particles, is very large (and impossible to prepare explicitly). This impossibility of preparing all degrees of freedom of a system in a defined initial state is the reason why one has to treat this system statistically. The number of (micro)states that are compatible with a certain value of the energy E is called a microcanonical ensemble and is denoted as

$$\Omega(N, V, E) = \left\{ \begin{array}{l} \text{number of microstates with } N \text{ and } V \\ \text{and energy between } E \text{ and } E + dE. \end{array} \right. \quad (6.25)$$

6.2.3 Statistics in Phase Space and Statistical Ensembles

Statistical thermodynamics uses theorems of probability calculus for the treatment of N -particle systems which move according to the physical laws of classical mechanics. The mechanical state of a system which is to be considered at time t can be described by the position of all particles pertaining to this system. Suppose we want to describe an N -particle system at time $t = t_1$ in a Cartesian coordinate system. A snapshot of this system yields a distribution of mass points in the coordinate system. Each point can be assigned to a position vector $q^i(t)$. Thus, the position of mass points of the mechanical system at time $t = t_1$ is fully characterized but not so for other points in time. As the quantities $q^i(t)$ are time dependent, one needs the additional information about the velocities $q^i(t)$, respectively momenta $p^i(t)$. In statistical physics, the two variables q^i and p_i span the mechanical configuration space of a system. One distinguishes two kinds of phase spaces in statistical thermodynamics, depending on whether one considers the *individual molecules* or the *whole system* as statistical elements.

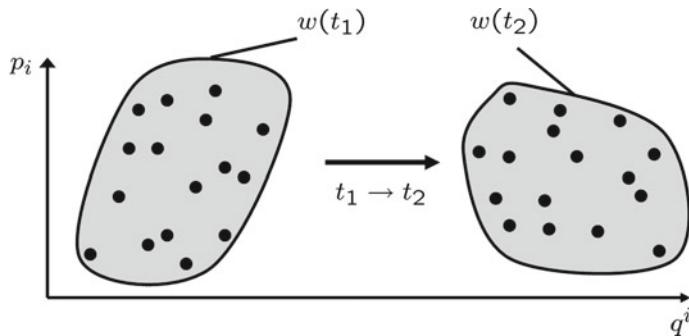


Fig. 6.4 A phase space volume w used in statistical thermodynamics evolving from a point in time t_1 to t_2 . When describing a system in μ -space, index i runs from 1 to f . Each point in this case represents one single atom or molecule of a system. If the motion of the particles of a system is restricted to a one-dimensional curve by some constraints, then $f = 1$ and thus the points of the system move in a two-dimensional phase space. In the general case with no constraints, $f = 3$ and μ -space has 6 dimensions. When using a Γ -space description of a system, one single point represents a complete N -particle system with N degrees of freedom: thus, index i in this case runs from 1 to fN and if the systems are not restricted in their motion in phase space, $f = 3$. The total dimensionality of Γ -space is therefore $6N$ and the above cloud of points corresponds to a Gibb's ensemble of many physically equivalent systems

μ -Space

If one considers the individual particles of a system (e.g. individual molecules or atoms in a fluid), then the generalized coordinates q^i and momenta p_i span the so-called μ -space. cf. Fig. 6.4, in which a complete N -particle system is described by the paths of N points in a $2f$ -dimensional space. Parameter f is the number of degrees of freedom; in the case of a *freely* moving particle or molecule in a system, $f = 3$ and thus the dimension of μ -space is 6.

If the points in phase space are located sufficiently dense, one can introduce an N -particle distribution function $\rho = \rho(q^i, p_i, t)$ of states. ρ is determined by separating phase space into single cells and then counting the number of phase points in each cell, i.e. one counts the number of microstates available to the system in phase space. This counting of microstates however is only valid for systems which are free of interactions.⁶ As the N points of a system in μ -space interact with each other, it is difficult to find an expression for ρ ; the correlations between individual points in phase space have to be taken into account. A set of points in this case represents a Gibb's ensemble of identically prepared systems.

Γ -Space

A different description of a mechanical system is a description in Γ -space. freedom is *one* “supermolecule” in a $2f$ -dimensional phase space, spanned by f generalized

⁶For example, an ideal crystal or an ideal gas.

coordinates and f generalized momenta, cf. Fig. 6.4. This means that a complex system is represented by one single point in Γ -space and a Gibb's ensemble of identically prepared (that is, physically equivalent) systems is given by a set of points. A number of systems is considered to be as physically equivalent, when the Hamiltonian $H = H(q^i, p_i, t)$ of each single system depends in the same way on the generalized coordinates and momenta. To have something specific in mind, one could think of a gas container of volume V containing N particles with energy E .

In Sect. 6.2.3 on p. 266 we have introduced a phase space distribution function $\rho(q^i, p_i, t)$ which specifies the probability of finding a system with N particles in a Γ -space interval $(q^i, \dots, q^i + dq^i, p_i, \dots, p_i + dp_i, t)$ at a certain time t . In order to obtain proper probabilities, $\rho(q^i, p_i, t)$ has to satisfy the following relations for a specific time t :

$$\rho(q^i, p_i) \in \mathbf{R} \quad (6.26a)$$

$$\rho(q^i, p_i) \geq 0 \quad (6.26b)$$

$$\int \rho(q^i, p_i) d^{3N}q d_{3N}p = 1. \quad (6.26c)$$

The probability density $\rho(q^i, p_i, t)$ of phase space states obeys *Liouville's theorem*.

Theorem 7 (Liouville's Theorem) *The dynamic development of a system in corresponds to a curve $(q^i(t), p_i(t))$ in phase space which is called phase space trajectory. For this trajectory the following equation holds:*

$$\frac{d\rho}{dt} = \frac{\partial\rho}{\partial t} + \{\rho, H\}, \quad (6.27)$$

where H is the classical Hamiltonian and $\{\rho, H\}$ is the Poisson bracket for ρ and H :

$$\{\rho, H\} = \sum_{k=1}^{3N} \left(\frac{\partial\rho}{\partial q^i} \frac{\partial H}{\partial p_i} - \frac{\partial\rho}{\partial p_i} \frac{\partial H}{\partial q^i} \right). \quad (6.28)$$

Proof The Hamiltonian equations are valid, i.e.

$$\dot{p}_i = -\frac{\partial H}{\partial q^i}, \quad (6.29a)$$

$$\dot{q}^i = \frac{\partial H}{\partial p^i}. \quad (6.29b)$$

Let

$$\mathbf{r} = (p_1, p_2, \dots, p_{3N}, q^1, q^2, \dots, q^{3N}) \quad (6.30)$$

a position vector of a point in Γ -space. Then the velocity of this system in the considered phase space volume w is given by

$$\mathbf{v} = (\dot{p}_1, \dot{p}_2, \dots, \dot{p}_{3N}, \dot{q}^1, \dot{q}^2, \dots, \dot{q}^{3N}). \quad (6.31)$$

The rate at which systems flow out of the finite volume w is given by

$$\frac{\partial \rho}{\partial t} \int_w dw \rho = - \int_{\sigma(w)} \rho(\mathbf{v} \mathbf{n}) d\sigma, \quad (6.32)$$

where the minus sign comes from the outbound direction of the normal vector. Using the divergence theorem, (6.32) can be written as

$$\int_w dw \left(\frac{\partial}{\partial t} \rho + \operatorname{div}(\rho \mathbf{v}) \right) = \sum_{i=1}^{3N} \frac{\partial (\rho q^i)}{\partial p_i} + \sum_{i=1}^{3N} \frac{\partial (\rho \dot{p}_i)}{\partial q^i} = 0. \quad (6.33)$$

Using (6.29a) and (6.29b) it follows (6.27). ■

Remark 33 Liouville's theorem means that the total number of systems in a Gibb's ensemble is preserved and that the ensemble moves like an incompressible fluid. For stationary ensembles ($\partial \rho / \partial t = 0$), the phase space density ρ is a constant of motion, i.e. it is conserved if $\rho(q^i, p_i, t) = \rho(H(q^i, p_i, t))$.

Using the phase space density ρ of a system at equilibrium, i.e. $\rho = \rho(q^i, p_i)$, an average of some observable A at time t is calculated as

$$\langle A \rangle = \int d_{3N} p d_q^{3N} \rho(q^i, p_i) a(q^i, p_i), \quad (6.34)$$

and the fluctuations of this observable are given by

$$(\Delta A)^2 = \langle (a - \langle A \rangle)^2 \rangle = \langle a^2 \rangle - \langle A \rangle^2. \quad (6.35)$$

The relative fluctuation, or *variance*, as a measure for the deviation of A from its mean value is given by

$$\frac{\Delta A}{\langle A \rangle}. \quad (6.36)$$

When measuring the fluctuations of two systems A and B at the same time, one obtains the *pair correlation* of A and B :

$$K_{AB} = \langle (A - \langle A \rangle)(B - \langle B \rangle) \rangle. \quad (6.37)$$

A probability distribution $\chi(a)$ for an observable $A(q^i, p_i)$ is obtained with

$$\chi(a) = \langle \delta(a - A) \rangle = \int d_{3N} q d_{3N} p \rho(q^i, p_i) \delta(a - A(q^i, p_i)), \quad (6.38)$$

where $\delta(a - A)$ is Dirac's δ -function and a is a possible value of A . Likewise, for two observables A and B one obtains

$$\chi(a, b) = \langle \delta(a - A) \delta(b - B) \rangle. \quad (6.39)$$

Integration of (6.39) with respect to b (a possible value for B) yields:

$$\int \chi(a, b) db = \int \langle \delta(a - A) \delta(b - B) \rangle db = \langle \delta(a - A) \rangle = \chi(a). \quad (6.40)$$

For quantities A and B which are *statistically independent*, the distribution χ separates in a product, i.e. $\chi(a, b) = \chi_A(a) \times \chi_B(b)$ and the correlation vanishes.

Proof

$$K_{AB} = \int da db (a - \langle A \rangle) (b - \langle B \rangle) \chi_A(a) \chi_B(b) \quad (6.41a)$$

$$= \int da (a - \langle A \rangle) \chi_A(a) \int db (b - \langle B \rangle) \chi_B(b) \quad (6.41b)$$

$$= (\langle A \rangle - \langle A \rangle) (\langle B \rangle - \langle B \rangle) = 0. \quad (6.41c)$$

■

6.2.4 Virtual Ensembles

In Sect. 2.3.1 of Chap. 2 it was discussed that the most important progress in the natural sciences was achieved by the crucial idea of *isolating* a system from its surrounding. In thermodynamics this principle can be seen in the distinction of different *ensembles*, depending on their degree of isolation from their surrounding.

- Isolated systems, i.e. systems without exchange of energy or matter.
- Closed systems, i.e. systems that exchange energy but no matter with their surrounding.
- Open systems, i.e. systems that exchange energy and matter with their environment

In experiments, for example, it is usually much easier to control temperature as a state variable during a measurement, than, e.g. the entropy or the chemical potential. The different physical situations are depicted in Fig. 6.5.

6.2.5 Entropy and Temperature

Irreversible processes (see e.g. Fig. 6.3) develop a system from an initial state to a final state in a sequence of non-equilibrium states whereas reversible processes only use equilibrium states; thus, equilibrium processes are an idealization: if a system is at equilibrium, all state variables are not time dependent and no process takes place. Quantum mechanically speaking, this means that all states are equally probable with likelihood

$$P_\nu = \frac{1}{\Omega(N, V, E)}. \quad (6.42)$$

This equation is also referred to as the *hypotheses of the equal a priori probabilities in phase space*. According to Ludwig Boltzmann, the link between statistical mechanics and classical thermodynamics is given by the following definition of entropy

$$S = k_B \ln \Omega. \quad (6.43)$$

This is an extensive quantity; thus, when considering two different systems A and B there common entropy is additive: $S_{AB} = S_A + S_B$. The thermodynamic definition

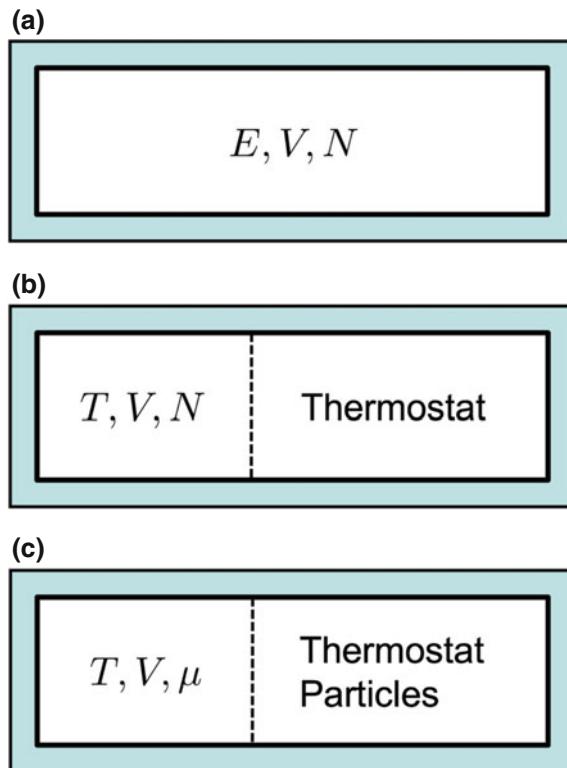


Fig. 6.5 Different Gibb's ensembles. **a** Microcanonical ensemble which is completely *isolated* from its surrounding. This ensemble represents many isolated systems, with the same energy E , the same volume V , and the same number of particles N . The *characteristic function* which characterizes this ensemble, is the entropy $S(E, V, N)$. The analytic form of the phase space density is $\rho \sim \delta(\mathcal{H} - E)$. **b** Ensemble which is coupled to a heat bath (a thermostat). For these systems, the temperature T , volume V , and the number of particles N is kept constant. The characteristic function for describing the physical situation is the free energy $F(T, V, N)$. The analytic form of the phase space density is $\rho(p^i, q_i) \sim \exp(-\beta\mathcal{H})$ with $\beta = 1/k_B T$. **c** Ensemble coupled to a heat bath and a reservoir of particles. Here, the system is in contact with a reservoir of particles for keeping the chemical potential μ constant. Additionally, the system is coupled to a heat bath which keeps the temperature T constant. The volume V of the simulation box is kept constant as well. The characteristic function describing this physical situation is $\rho(q_i, p^i) \sim \exp(-\beta(\mathcal{H} - \mu N))$

of temperature is then given by

$$\frac{1}{T} = (\partial S / \partial E)_{N,V}, \quad (6.44)$$

where

$$\beta = (k_B T)^{-1} = (\partial \ln \Omega / \partial E). \quad (6.45)$$

6.3 Classical Interatomic and Intermolecular Potentials

In order to study longer-timescale phenomena of systems composed of a larger number of particles, it becomes necessary to introduce much simpler methods. The issue here is how to reduce the amount of necessary computation in such methods, and also, how to introduce parameterizations or fittings into the interatomic potentials without losing too much accuracy and reliability. One way of doing this, is to construct realistic classical potentials based on ab initio calculations. One methodology here is to fit parameterized classical potentials to contour maps of the total energy, which may be obtained with an ab initio method by changing the position of one atom while fixing the coordinates of all other atoms. Interatomic potentials of atoms can, in principle, be determined at absolute zero temperature by estimating the ground state total energy of the corresponding electronic system with given atomic positions. This is the most reliable way to generate classical potentials, albeit prohibitive for larger systems. For these systems, it much more convenient to fit the ab initio total energies to simple *analytic* potential functions, which are commonly called *molecular dynamics* or *semi-empirical potentials*.

6.3.1 Charged Systems

In the case of *ionic solid state crystals*, the total energy can be estimated accurately by summing the isotropic Coulomb interaction energies between the ions with lattice position vectors \mathbf{r} as follows:

$$E = \frac{1}{2} \sum_{ij}^N \frac{\pm Z_i Z_j e^2}{\mathbf{r}^i - \mathbf{r}^j}. \quad (6.46)$$

This equation can be re-written as

$$-\frac{NZ^2}{2} \frac{\alpha}{a}, \quad (6.47)$$

where a is a lattice constant. The coefficient α is called a “Madelung constant”, which is a dimensionless parameter which is only conditionally convergent as it depends on the lattice and the way of ionic alignments but not on the spacial scale a . The Madelung constant can be determined by identifying (6.47) with the Madelung energy which may be evaluated by means of the Ewald sum.⁷

Example 38 (Madelung Constant of a Linear Chain) Consider a linear chain with N alternating ion charges:

$$+ - + - + - \underbrace{+ - +}_{2a} - \dots$$

⁷For example, $\alpha = 1.6381$ for a ZS-type structure, or $\alpha = 1.7476$ for a NaCl-type crystal. For general crystal structures or noncrystalline solids, the electrostatic energy has to be calculated for every given set of atomic coordinates by means of the Ewald sum.

The electrostatic energy is calculated as:

$$-\frac{e^2}{R} \cdot 2 \underbrace{\left(\frac{1}{a} - \frac{1}{2a} + \frac{1}{3a} \pm \dots \right)}_{=\alpha}. \quad (6.48)$$

Thus,

$$\alpha = 2 \sum_{n=1}^N \frac{(-1)^{n+1}}{n} \underset{N \rightarrow \infty}{\approx} 2 \log 2. \quad (6.49)$$

In systems composed of charged *molecules*, e.g. in biological systems, the electrostatic interactions play a crucial role. In such a case, the electrostatic potential at position \mathbf{R}_i is given by

$$\Phi(\mathbf{R}_i) = \sum_n \frac{q_n(\mathbf{r}_n)}{|\mathbf{R}_i - \mathbf{r}_n|}, \quad (6.50)$$

where $q_n(\mathbf{r})$ is the charge distribution around the j th molecule. If one looks at the charge distribution from greater distance, i.e. if $\mathbf{R}_{ij} = (\mathbf{R}_i - \tau_j) \gg 1$, (6.50) can be expanded in a multipole expansion:

$$\Phi(\mathbf{R}_{ij}) = \frac{Q_j}{R_{ij}} + \sum_{\alpha} P_j^{(\alpha)} \frac{\partial}{\partial X_{ij}^{\alpha}} \frac{1}{R_{ij}} + \frac{1}{2!} \sum_{\alpha\beta} \Theta_j^{(\alpha\beta)} \frac{\partial^2}{\partial X_{ij}^{\alpha} \partial X_{ij}^{\beta}} \frac{1}{R_{ij}} + \dots, \quad (6.51)$$

where the following quantities are defined as

$$Q_j = \sum_n q_n, \quad (6.52a)$$

$$P_j^{(\alpha)} = \sum_n q_n x_n^{\alpha}, \quad (6.52b)$$

$$\Theta_j^{(\alpha\beta)} = \sum_n q_n x_n^{\alpha} x_n^{\beta}. \quad (6.52c)$$

Q_j is the total charge of the j th molecule, P_j is its dipole moment, and $\Theta_j^{(\alpha\beta)}$ the quadrupole moment. If each molecule is electrically neutral, then the most dominant energy may be due to the dipole-dipole interaction between molecules, which can be expressed as

$$D_{ij} = \frac{(P_i P_j) R_{ij}^2 - 3(P_i R_{ij})(P_j R_{ij})}{R_{ij}^5}. \quad (6.53)$$

If the charge distribution of a molecule is not altered, i.e. not affected by the environmental configuration of other molecules, the potential form discussed here can be used without change. Usually, the charge distribution of a molecule depends on the environment. If such an effect is not negligible, one should consider it explicitly. One such effect is the induction of multipoles [399]. The induction, dispersion and dipol-dipol interaction of molecules have a range of $\sim 5 \text{ \AA}$. This is rather small compared to the range of, e.g. the interaction between ions ($\sim 500 \text{ \AA}$).

6.3.2 Ewald Summation

Polyelectrolytes are electrically charged macromolecules, which are usually solvable in water. For molecules with only slowly decreasing potentials such as the Coulomb potential

$$\Phi_C = \frac{\lambda \varepsilon \sigma}{2} \sum_{n=0}^{\infty} \left(\sum_{i=1}^N \sum_{j=1}^N \frac{Z_i Z_j}{|\mathbf{r}_{ij} + \mathbf{n}|} \right) \quad (6.54)$$

the minimum image convention (see Sect. 6.4.4) cannot be applied. In (6.54), the sum over n means that the summation is done in the central simulation box and over all periodic images, where N is the number of particles. This means that the summation of the forces has to be done with an effort $\mathcal{O}(N^2)$. \vec{n} is the vector by which particles in the central box have to be translated, in order to obtain the corresponding image. In the sum over n the self-energy term $i = j$ is left out. A direct summation of the terms in (6.54) is slowly and only conditional convergent. Independent of the number of terms being taken into account, the limiting value of the sum depends on the order of terms in the sum. Several methods were devised to improve the convergence behavior of the sum in (6.54). Most methods are based on introducing a screening function ρ_{screen} , based on a Debye–Hückel approach. ρ_{screen} is 1 for small arguments and which zero for arguments that are larger than half the box-length. Physically speaking, this means, that the point charges are screened by an oppositely charged spherical cloud. To remove this drastic intervention in the system, one substitutes in a second step each point charge by a like-charged spherical cloud. This second contribution is then calculated in reciprocal space, i.e. in Fourier space. The screening function in direct space is thus approximated by several discrete Fourier components. As a result, this procedure leads to a convergent sum, independent of the order of terms.

The *Ewald Summation* [400, 401] solves the summation problem by choosing a particular screening function in direct (position) space, namely a Gaussian function. Thus, for the summation of the slowly converging sum (6.54) one uses the following identity:

$$\frac{1}{r} = \frac{f(r)}{r} + \frac{1 - f(r)}{r}. \quad (6.55)$$

Function $f = \rho_{screen}$ is chosen as

$$\rho_{screen}(r)q \frac{\kappa^3}{\sqrt{\pi^3}} \exp(-\kappa^2 r^2). \quad (6.56)$$

For the screened potential one obtains

$$\Phi_{screen}(\mathbf{r}) = q \left(\frac{1}{r} - \frac{\kappa^3}{\sqrt{\pi^3}} \int \frac{\exp(-\kappa^2 r'^2)}{|\mathbf{r} - \mathbf{r}'|} d^3 r' \right). \quad (6.57)$$

Calculating the integral finally leads to the error function and the total potential can be written as a sum of four terms:

$$\Phi_{total} = \Phi_{direct} + \Phi_{reciprocal} + \Phi_{surface} - \Phi_{self}. \quad (6.58)$$

The first term is the *direct* sum:

$$\Phi_{direct} = \frac{\lambda\varepsilon\sigma}{2} = \sum_{i=1}^N \sum_{j=1}^N \sum_{\mathbf{n}=0}^{\infty} q_i q_j \frac{erfc(\kappa|\kappa\mathbf{r}_{ij} + \mathbf{n}|)}{|\mathbf{r}_{ij} + \mathbf{n}|}. \quad (6.59)$$

If the damping constant κ in the screening function is chosen large enough, then the direct sum with $\mathbf{n} = 0$ contributes exclusively to the potential. Thus, this term can be calculated in this sum using the minimum image convention in the central simulation box. The other long-range terms are calculated in the reciprocal sum:

$$\Phi_{reciprocal} = \lambda \frac{2\pi\varepsilon\sigma}{L_{box}^3} \sum_{i=1}^N \sum_{j=1}^N q_i q_j \sum_{\mathbf{k} \neq \mathbf{0}}^{\infty} \frac{1}{k^2} \exp(-\frac{k^2}{4\kappa^2}) \cos(\mathbf{k}\mathbf{r}_{ij}). \quad (6.60)$$

The surface term only depends on the chosen boundary conditions at infinity. If the system is surrounded by vacuum, then it gets polarised. Thus, on the surface of the system there is a dipole layer which contributes to the energy as follows:

$$\Phi_{surface} = \lambda \frac{2\pi\varepsilon\sigma}{3L^3} \left| \sum_{i=1}^N Z_i \mathbf{r}_i \right| \quad (6.61)$$

Finally, the self energy term has to be subtracted, as it is contained in the reciprocal term:

$$\Phi_{self} = \lambda \frac{\kappa\varepsilon\sigma}{\sqrt{\pi}} \sum_{i=1}^N q_i^2. \quad (6.62)$$

6.3.3 The P^3M Algorithm

The Fast Fourier Transform (FFT) Algorithm is a very fast algorithm to calculate sum in Fourier space [35], which reduces the effort of calculation to $\mathcal{O}(N \log N)$. The P^3M -method (Particle-Particle-Mesh) [48] is a faster method to calculate the sum (6.54) which uses FFT. Here, Φ_{self} , $\Phi_{surface}$ and Φ_{direct} are calculated in the same way as in the standard Ewald summation. Only the calculation of the reciprocal term is split into several steps which are displayed in Algorithm 6.1.

Algorithm 6.1 P^3M -Algorithm to calculate the reciprocal sum

1. Distribute the charges on a lattice.
 2. Fourier-transform (FFT) the discrete charges into reciprocal space.
 3. Solve the Poisson-equation in reciprocal space.
 4. Calculate the electric field in reciprocal space.
 5. Re-Fourier-transform the electric field to direct space.
 6. Re-distribute the forces to the particles in direct space.
-

6.3.4 Van der Waals Potential

For inert gas elements such as He , N , Ar , K , the LJ potential is a very good approximation:

$$\Phi_{ij}(R_{ij}) = K \left[\left(\frac{R_0}{R_{ij}} \right)^{12} - 2 \left(\frac{R_0}{R_{ij}} \right)^6 \right]. \quad (6.63)$$

Here, R_{ij} is the interatomic distance, and U_0 and R_0 are the binding energy and equilibrium distance of the two atoms, respectively. The second term in (6.63) describes the van-der-Waals type attractive interaction between atoms, neither of which has any intrinsic dipole moment. The quantum mechanical origin of the attractive part of the van der Waals (attractive) interaction is due to the fluctuations of the electrons at zero temperature. The size of the effect is calculated using 2nd order perturbation theory in quantum mechanics. The result is $\Phi_{ij} \sim 1/R_{ij}^6$ and was first derived by London in [402]. The short-range repulsive part in the LJ potential can be considered to be a result of interaction of higher-order terms in the perturbation expansion of the energy shift. For very short distances however, the nature of this inter-particle repulsion can be more attributed to the overlapping electron clouds.

6.3.5 Covalent Bonds

In the work of Abell [403] it was realized that the potentials between neutral atoms exhibit *universal features at short ranges*. Although this study was confined to pair potentials, it was shown, that for most atoms, the repulsive and attractive pair interactions can be parameterized as simple exponential functions and that nearest neighbor interactions may be ignored without losing much accuracy. The universal form of pairwise exponential potentials, proposed by Abell (the Abell potential) is

$$\Phi(r) = A \exp(-\Theta r) - B \exp(-\lambda r). \quad (6.64)$$

Covalent bonds have a strong angle dependence and cannot be appropriately be expressed by pair potentials. Tersoff [404] and Abell thus proposed an extension of the potential to *three-body* interactions:

$$\Phi(r_{ij}) = \sum_{i < j} f_c(r_{ij}) [A_{ij} \exp(-\lambda_{ij} r_{ij}) - b_{ij} \exp(-\mu_{ij} r_{ij})], \quad (6.65)$$

where r_{ij} is the distance between atoms i and j and $f_c(r_{ij})$ is a cutoff function given by

$$f_c(r_{ij}) = \begin{cases} 1 & : R_{ij} < R_{ij}, \\ \frac{1}{2} + \frac{1}{2} \cos [\pi(r_{ij} - R_{ij})/(S_{ij} - R_{ij})] & : R_{ij} \leq r_{ij} \leq S_{ij}, \\ 0 & : r_{ij} > S_{ij}. \end{cases} \quad (6.66)$$

Parameter b_{ij} in (6.65) represents the bond strength due to many-body effects and is given by

$$b_{ij} = B_{ij} \chi_{ij} (1 + \beta_i^{n_i} \xi_{ij}^{n_i})^{-1/2 n_i}, \quad (6.67)$$

where

$$\xi_{ij} = \sum_{k \neq i, j} f_c(r_{ik}\omega_{ik}g(\Theta_{ijk})) , \quad (6.68a)$$

$$g(\Theta_{ijk}) = 1 + (c_i/d_i)^2 - c_i^2 / [d_i^2 + (h_i - \cos \Theta_{ijk})^2] , \quad (6.68b)$$

$$\omega_{ik} = \exp [\mu_{ik}^3 (r_{ij} - r_{ik})^3] . \quad (6.68c)$$

Here, Θ_{ijk} is the bond angle between the ij bond and the ik bond; $\beta_i, n_i, c_i, d_i, h_i$ are parameters which have to be determined for different atomic species, see e.g. [405]. For the other parameters, usually the following simple relations are assumed:

$$A_{ij} = (A_i A_j)^{1/2} , \quad (6.69a)$$

$$B_{ij} = (B_i B_j)^{1/2} , \quad (6.69b)$$

$$R_{ij} = (R_i R_j)^{1/2} , \quad (6.69c)$$

$$S_{ij} = (S_i S_j)^{1/2} , \quad (6.69d)$$

$$\lambda_{ij} = (\lambda_i \lambda_j)^{1/2} , \quad (6.69e)$$

$$\mu_{ij} = (\mu_i \mu_j)^{1/2} . \quad (6.69f)$$

6.3.6 Embedded Atom Potentials

It is well-known, that the use of pair-potentials leads to the Cauchy-relation $C_{12} = C_{44}$. This relation between the elastic constants, however, is not always true with metals. Thus, simple pair potentials, such as the LJ potential, provide no good description of metals, particularly of transition and noble metals, in which d -electrons contribute to the bonding. An additional problem comes in because the vacancy formation energy E_V is not always equal to the cohesive energy E_C , but with pair potentials one always has $E_V = E_C$. Moreover, pair potentials cannot reproduce the particular properties of metal surfaces. The interlayer distance between the first and second layers near the surface is experimentally smaller than the bulk interlayer distance. The use of pair potentials gives an unrealistically high vaporization rate near the melting temperature.

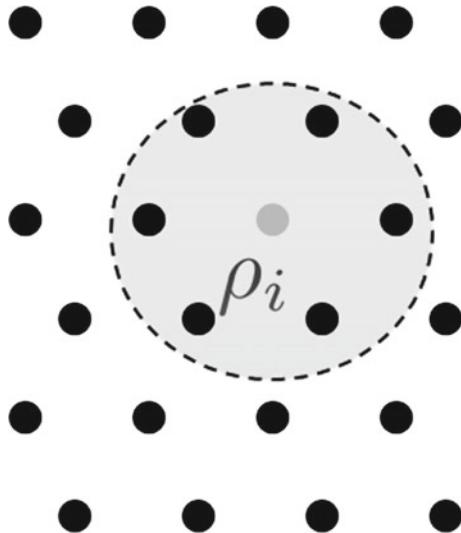
As a solution of these problems, Daw and Baskes [386] proposed to describe the energy of an atom as the sum of pairwise terms $\Phi(r_{ij})$ and many body terms which are expressed in terms of the electron density ρ_i at each atomic center.

$$\Phi = \sum_{i>j} \Phi(r_{ij}) + \sum_i^N E(\rho_i) . \quad (6.70)$$

This potential description is called *embedded atom method* (EAM) as the term $E(\rho_i)$ represents the energy needed to embed the atom at a point of electron density ρ_i . The electron density ρ_i is determined by superposition of electron densities of the surrounding atoms, cf. Fig. 6.6:

$$\rho_i = \sum_{i \neq j} \rho_j^{\text{atom}}(r_{ij}) . \quad (6.71)$$

Fig. 6.6 Embedding of an atom within the electron density of the surrounding atoms. The energy at a certain position depends on the accumulated electron densities of the surrounding atoms



ρ_j^{atom} is the electron density of the j th atom, which may be determined, for example, by solving the Hartree-Fock equation, see e.g. [406]. The forces in an EAM system are given by

$$\mathbf{F}_i = \sum_{j \neq i} \left[F'_i(\rho_i) \rho_j^{\text{atom}'}(r_{ij}) + F'_j(\rho_j) \rho_i^{\text{atom}'}(r_{ij}) + \Phi'_{ij}(r_{ij}) \right] \frac{\mathbf{r}_j - \mathbf{r}_i}{r_{ij}}. \quad (6.72)$$

For a numerical calculation, appropriate descriptions for the embedding part $E(\rho)$, the effective charge $Z(r)$ (contained in the pair potential $\Phi_{ij}(r_{ij})$) and the electronic density $\rho_j^{\text{atom}}(r)$ have to be found. For example, this can be done by using splines [407] or by fitting the functions to the second-moment approximations in the tight-binding method, as proposed by Finnis and Sinclair [408]. Since the electron densities of an isolated atom and of an atom in a solid are generally different, ρ_j^{atom} should not simply be set equal to the density of the corresponding isolated atom. Usually, it is modified empirically to

$$\rho_j^{\text{atom}}(r) = \begin{cases} N_s \rho_s^{\text{atom}} + (N - N_s) \rho_{p/d}^{\text{atom}} - \rho_c & : r < r_c \\ 0 & : r \geq r_c \end{cases}, \quad (6.73)$$

where N is the number of valence electrons of the atom, and ρ_s^{atom} and ρ_d^{atom} denote the electron densities of s , p and d electrons of the isolated atom. To make the potentials finite, a cut-off radius r_c is introduced which limits the range of the electron density. Constant ρ_c is chosen such that the density ρ^{atom} is continuously differentiable at the cut-off radius.

The embedded-atom method has been successfully applied to many fcc- and bcc-type metals and other transition metals whose d -band is almost fully occupied.

6.3.7 Pair Potentials

For a N -body system the total energy Φ_{nb} , i.e. the potential hypersurface of the non-bonded interactions can be written as

$$\Phi_{nb}(\mathbf{R}) = \sum_i^N \phi_1(\mathbf{r}_i) + \sum_i^N \sum_{j>i}^N \phi_2(\mathbf{r}_i, \mathbf{r}_j) + \sum_i^N \sum_{j>i}^N \sum_{k>j>i}^N \phi_3(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots, \quad (6.74)$$

where ϕ_1, ϕ_2, ϕ_3 , etc. are the interaction contributions due to external fields, e.g. the effect of container walls, and pair, respectively triple interactions of particles with index i, j or k , respectively. In practice, the external field term is usually ignored, while all the multi-body effects are incorporated into Φ_2 in order to reduce the computational expense of the simulations. Hence, the general structure of a non-binding pair potential

$$\Phi_{nb} = \Phi_0 + \frac{1}{2} \sum_{i,j \neq i}^N \Phi(|\mathbf{r}_i - \mathbf{r}_j|) \quad (6.75)$$

can be characterized as:

- Repulsive at short distances,
- Attractive at intermediate and long distances,
- Usually applied with a cutoff.
- Energy and length scale are the minimum set of parameters.

Analytic forms of pair potentials are usually based on fundamental notions of physics. Fitting of potential parameters however often impairs the physical meaning of the analytic form.

The simplest general ansatz for a non-binding potential for spherically symmetric systems, i.e. $\Phi(\mathbf{r}) = \Phi(r)$ is a potential of the following form:

$$\Phi_{nb}(r) = \Phi_{Coulomb}(r) + \left(\frac{C_1}{r}\right)^{12} + \left(\frac{C_2}{r}\right)^6. \quad (6.76)$$

The term

$$\Phi_{Coulomb}(r) = \sum_i \sum_{j>i} \frac{Z_i Z_j e^2}{\epsilon |\mathbf{r}_i - \mathbf{r}_j|} \quad (6.77)$$

is the electrostatic energy between the particles with position vectors \mathbf{r}_i and \mathbf{r}_j , respectively and C_1 and C_2 are parameters of the attractive and repulsive interaction. If two atoms are closer to each other than the so-called bond length, then a very sudden growth of the repulsive force of the electron clouds occurs. There are several analytic ways to describe this physical phenomenon. As early as 1924, Jones proposed the following potential function to describe pairwise atomic interactions [409]:

$$\Phi(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} + \left(\frac{\sigma}{r}\right)^6 \right]. \quad (6.78)$$

Today, this potential with $C_1 = C_2 = \sigma$ is known as the *Lennard-Jones (LJ) (6–12) potential*. Parameter ϵ determines the energy scale and σ the length scale.

Using an exponential function instead of the repulsive r^{-12} term, one obtains the *Buckingham potential* [410]:

$$\Phi(r) = b \exp(-ar) - \frac{c}{r^6} - \frac{d}{r^8}. \quad (6.79)$$

This potential however has the disadvantage of using a numerically very expensive exponential function and it is unrealistic for many substances at small distances r and has to be modified accordingly. Another analytic form often employed for metals is the *Morse potential*

$$\Phi(r) = D \exp(-2\alpha(r - r_0)) - 2D \exp(-\alpha(r - r_0)). \quad (6.80)$$

For reasons of efficiency, a classical MD potential should be short-ranged in order to keep the number of force calculations between interacting particles at a minimum. Therefore, instead of using the original form of the potential in Eq. 6.78, it is common to use a modified form, where the potential is cut off at its minimum value $r = r_{\min} = \sqrt[6]{2}$ and shifted to positive values by ϵ such that it is purely repulsive and smooth at $r = r_{\text{cut}} = \sqrt[6]{2}$:

$$\Phi_{LJ}(r) = \begin{cases} 4 \epsilon \left\{ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right\} + \epsilon & r \leq 2^{1/6}\sigma, \\ 0 & \text{otherwise.} \end{cases} \quad (6.81)$$

The bulk modulus of a general LJ material is given by the curvature in the potential minimum, i.e.

$$B_{LJ} = \Phi \frac{\partial^2 E}{\partial V^2}. \quad (6.82)$$

Pure pair potentials neglect the specific organization of bonds in materials; thus, it is very difficult to stabilize e.g. a diamond-cubic structure for *Si* only with using pair potentials. They also do not include any angular dependence which is a problem for e.g. covalently bonded materials and a pure pair potential has no stability against shear. Thus, pair potentials are limited in their transferability when fitted for one particular coordination environment, e.g. a pair potential fitted to the bulk behavior of some material cannot be used without modifications to model surface effects. For metals or covalent solids it can be shown, that the energy difference between an fcc and bcc configuration is a “fourth-order” effect. Thus, pair potentials fundamentally cannot predict such crystal structures. Some typical quantities to which pair potentials may be fitted to are:

- Equation of state,
- Cohesive energy,
- Lattice constants,
- Phonon frequencies,
- Forces,
- Surface and relaxation energies,
- Liquid pair correlation functions,

- Bulk modulus,
- Lattice constants.

One of the first examples for applying a pair potential for an atomistic modeling of a material, is the study of radiation damage in *Cu* conducted by Gibson et al. [411].

6.4 Classical Molecular Dynamics Simulations

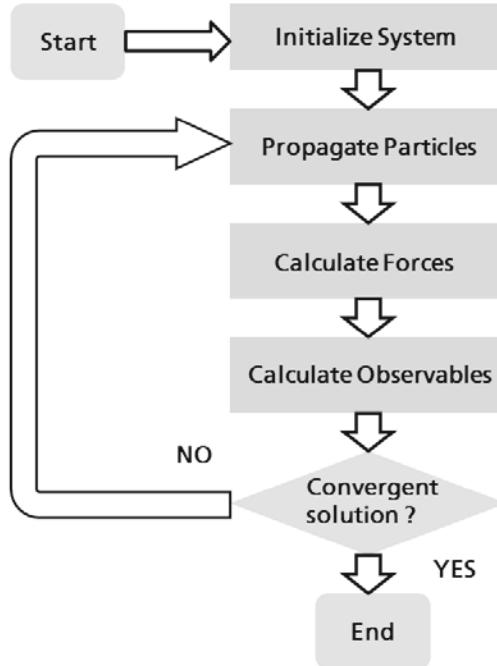
In this section we discuss a case study for classical molecular dynamics (MD) simulations, namely a simple program for simulating a molecular fluid in a microcanonical ensemble in a periodic, cubic simulation box. In doing so, we refine the general scheme of a computer simulation presented in Chap. 1 on p. 3 for a discussion of the molecular dynamics method. The basic ingredients of a working classical MD program are considered and explicit code samples are given in the language C. Classical in this context means that we do not consider any quantum states of the atoms which are thus described as classical particles. First-principle-based (or ab initio) MD as well as semi-empirical MD are discussed in Chap. 5. The classical MD method allows the prediction of static and dynamic properties of matter in the fluid, gaseous or solid state, based on a direct modeling of classical interaction potentials between the atoms or molecules. In the simplest case, the MD method solves the classical Newtonian equations of motion of a N -particle system and follows the time-dependent trajectories of the particles. We note that with MD methods, the particle trajectories are *not* what one is interested in due to the Lyapunov instability, cf. p. 188 and [57], but rather to advance the system to an equilibrium state, make measurements at equilibrium and calculate statistical properties. The MD method is probably *the* most important and most widely used method based on particles in scientific and engineering fields [49, 50, 54, 412, 413] and there are some excellent textbooks which treat this method in some depth, see e.g. [54, 57, 414]. The quality and reliability of the results obtained with this method heavily depend on a correct or at least useful choice of the underlying potentials which fully determine the material properties in the simulation model.

6.4.1 Numerical Ingredients of MD Simulations

A Lennard-Jones fluid is the simplest model of a molecular fluid, based on interacting spherical particles, which allows for illustrating the basic MD techniques. The interactions are the simplest possible, occurring between pairs of particles. The reason for the preferred shape of spheres in contrast to, e.g. the discrete element method (DEM), is obvious. It allows for efficient search strategies for identifying potentially interacting particles during a simulation run. In contrast to this, with DEM [415], one often makes use of convex polygons to represent microstructural features of granular structures as seen on the microscale. The MD method is still

open for new numeric developments and its theoretical foundations rest in analytical mechanics and the justified believe that the simulated trajectories of particles represent the “true” development of a system. Algorithm 6.2 illustrates the principal flow diagram of a MD program, cf. also Fig. 1.5. The basic idea of MD is to proceed in the same way as in a typical macroscopic experiment: One first prepares a sample of the material one wishes to study; This is the initialization procedure. The material sample is connected to a measuring instrument (e.g. a thermometer or viscosimeter, etc.) which controls some macroscopic variables of the system, e.g. pressure, temperature, Cauchy tension, and one measures the properties of interest during a certain time interval. The setting of external conditions for the state variables of a system is achieved by the equations of motion which are integrated. In the simplest case considered here, simply the temperature, which is defined by the particle velocities, is kept at a constant value. This corresponds to a system in a microcanonical ensemble, that propagates in phase space on a hypersurface of constant energy which is $\propto \delta(\mathcal{H} - E)$ where \mathcal{H} is the total Hamiltonian of the system. After a while, the system will have reached equilibrium, where the systems macroscopic state variables, such as pressure, temperature, energy, only fluctuate about the equilibrium value. Using the ergodic hypotheses, if the measurements are long enough the system will come arbitrarily close to any point on the hypersphere in phase space, and if at equilibrium,

Algorithm 6.2 Principal flow diagram of a MD simulation



every measurement of a system's observable should be statistically independent, i.e. the correlations (6.41) between measurements vanish. The measurements are subject to statistical noise, as all experimental measurements. From measurements at equilibrium, average statistical properties of the system can be calculated, based on the particles positions, velocities and forces. Most often, one is only interested in structural properties which can be extracted from the coordinates.

In a first step, the system has to be initialized. This means setting up and initializing all variables, allocating memory, and assigning initial particle coordinates in the simulation box. Algorithm (C) in Appendix C shows a fragment of a C-code which shows in principle how the main routine in MD can be handled. Figure 6.7 shows a possible strategy of combining header and implementation files into a working program when using the C-language. A principal flow scheme of consecutive tasks in an MD simulation run is provided in Algorithm 6.3

In a molecular dynamics program, first a model system of N particles is prepared. Preparation means setting up the initial configuration of particles, setting up the initial conditions for positions and velocities of all particles, and choosing a force model for the interaction between the particles.

The declarations of functions (done in header (.h)-files) should not be mixed up with the specific implementation of a function (which is done in (.c)-files). In Algorithm (C) the different functions on the top level in the main function make use of the information stored about the particles and the simulation parameters. This information is conveniently stored in C-structures which are made accessible to other modules by including the header file "defs.h", cf. the code sample in Algorithm 6.4.

In the simplest case of a MD scheme, the Newtonian equations of motion for the system of N particles are solved until the properties of the system do not change anymore with time; this is the case, when the system has reached its *equilibrium*

Algorithm 6.3 A general MD simulation scheme

The bonded interactions in step 3 may be e.g. the connectivity in a molecule

```
1 /* A simple MD Program */
2 1. INPUT initial conditions
3     a) Set all simulation parameters
4     b) Set positions of all particles
5     c) Set velocities of all particles
6     d) Reset total momentum to zero
7
8 2. REPEAT Steps (3-5) until last last timestep
9
10 3. Calculate Forces from non-bonded and bonded
11     potentials
12 4. Integrate Forces
13 5. Calculate Observables
14 6. END
```

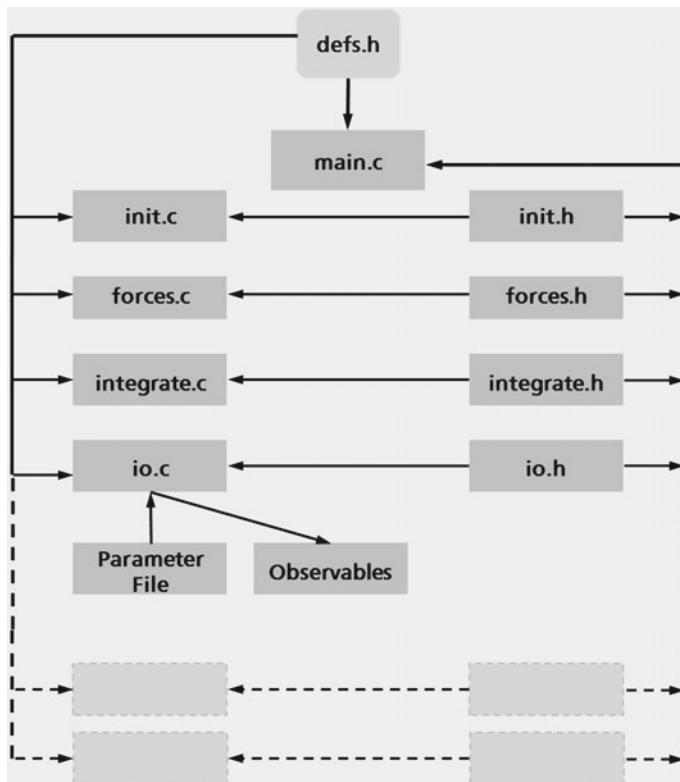


Fig. 6.7 Example for a structured program design keeping the declarations of functions (in (.h)-files) separate from their specific implementations (in (.c)-files). The “io.h” file handles everything that has to do with reading and writing data, e.g. with reading in simulation parameters from an external file. Control parameters for a simulation should not be hard-coded, but rather kept separate from the main code. This is a concept which is realized in virtually all commercial scientific program packages. Sometimes the whole initialization stage of a simulation (setting up particles and boundary conditions, etc.) is done with a separate program and then only the ready-to-use parameter files are read in when starting the simulation

state in which the measured macroscopic state variables, obtained as averages over microscopic quantities, do not change anymore. In fact, many of the basic errors, that one can make in a computer experiment are similar to the ones that can be made in real experiments, e.g. the sample is not large enough, or prepared incorrectly, or the measurement time is too short, the system is not at equilibrium during the measurement, or one does not measure what one thinks. The fundamental observables of a MD simulation are the particles’ velocities and momenta (the forces are only rarely needed for calculating observables); Hence, in order to measure an observable in an MD simulation, one must be able to express this quantity in terms of the particles

Algorithm 6.4 Code fragment showing a convenient organization of data within C-structures

These structures are provided by including the file “defs.h” into other modules ((.c)-files). That way, all structural data are organized at one place and be changed or extended easily

```

1 /* Code fragment for organizing of particle and
   parameter data */

2
3 #ifndef _DEFS_H
4 #define _DEFS_H

5
6 /* List particle properties */
7 struct Particle
8 {
9     double pos_x, pos_y, pos_z;
10    double velo_x, double velo_y, double velo_z;
11    double force_x, double force_y, double force_z;
12    int charge;
13    int moleculeType;
14
15 /* Other types of particle properties */
16 };
17 typedef struct Particle T_Particle;
18
19 /* List of parameters controlling the simulation flow
20 struct Parameters
21 {
22     int MDCycles;
23     int IntervalObservables;
24     int IntervalSafetyFiles;
25     int NumParticles;
26
27 /* More parameters */
28 };
29 typedef struct Parameters T_Parameters;
30 #endif

```

positions or momenta. For example, temperature in MD simulations, according to the *classical equipartition theorem*, is defined as the average kinetic energy per degree of freedom:

$$\left\langle \frac{1}{2}mv_{\alpha}^2 \right\rangle = \frac{1}{2}k_B T . \quad (6.83)$$

Algorithm 6.5 Rescaling the velocity of particles to a predefined temperature T in a microcanonical ensemble

```

1 #include "defs.h"
2 /* Maybe more includes */
3 double ScaleVelocities(T_Particle particle,\n
4                         T_Parameters *parameter){
5     /* Local variables */
6     int i,k;
7     double T, sum_v_squared, scale_factor, v_x,v_y,v_z;
8     sum_v_squared = 0.;
9     scale_factor = 0.;
10    v_x           = v_y = v_z = 0.;
11
12    /* Compute the total kinetic energy of the system */
13    for (i = 0; i < parameter->NumParticles; i++)
14    {
15        v_x = particle[i].velo_x;
16        v_y = particle[i].velo_y;
17        v_z = particle[i].velo_z;
18        sum_v_squared += v_x * v_x + v_y * v_y + v_z * v_z;
19    }
20    scale_factor = ((parameter->dim * parameter->\n
21    T * (double)(parameter->NumParticles)) / sum_v_squared);
22    scale_factor = sqrt(scale_factor);
23    sum_v_squared = 0.0;
24
25    /* Scale the velocities and recompute the total energy */
26    for (i = 0; i < parameter->NumParticles; i++)
27    {
28        v_x = particle[i].velo_x *= scale_factor;
29        v_y = particle[i].velo_y *= scale_factor;;
30        v_z = particle[i].velo_z *= scale_factor);
31        sum_v_squared += (v_x * v_x + v_y * v_y + v_z * v_z );
32    }
33    /* Return the total kinetic energy */
34    return (0.5 * sum_v_squared);
35 }
```

In practice, one measures the total kinetic energy of a system and divides this by the number of degrees of freedom f .⁸ If the total kinetic energy of a system during a simulation fluctuates, so does the temperature is

$$T(t) = \frac{m_i v_i^2(t)}{k_B f} . \quad (6.84)$$

The unit of temperature is ε/k_B and since each translational degree of freedom contributes $k_B T/2$ to the kinetic energy, in reduced units (with $k_B = 1$) (6.84) reduces in d dimensions to

$$T = \frac{1}{dN_a} \sum_i v_i^2 , \quad (6.85)$$

⁸For a system of N particles with fixed momenta $f = 3N - 3$.

where d in virtually all practical cases will be $d = 3$. Algorithm 6.5 shows a code sample for rescaling the particle velocities as described in (6.83) and (6.84). In this code sample, the total energy is returned to the calling function. Note, that the particle coordinates and the simulation parameters are given to this function as argument and with the point operator one can access those parts of the structures that are needed.

6.4.2 Integrating the Equations of Motion

We start with the classical equations of motion of a force-free point particle of mass m moving with velocity v which may be obtained from the Lagrange function which in this case is identical with the kinetic energy of the mass.

$$L = \frac{m}{2} v^2 = \frac{m}{2} g_{ab} \frac{dx^a}{dt} \frac{dx^b}{dt} = \frac{m}{2} g_{ab} \dot{x}^a \dot{x}^b. \quad (6.86)$$

The corresponding Lagrange equations are the solution to the variational problem:

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{x}^a} - \frac{\partial L}{\partial x^a} = 0. \quad (6.87)$$

Applying (6.87) in Cartesian coordinates recovers Newton's law of motion

$$m \ddot{\mathbf{r}}_i = \sum_{j \neq i}^N \mathbf{F}_{ij}. \quad (6.88)$$

The usually best integrator for a energy-conserving microcanonical ensemble, is the widely used symplectic Verlet-algorithm, cf. the discussion in Sect. 4.2.1 on p. 186. Algorithm 6.6 shows how the integration routine may be implemented in an actual MD program. The integration function "Integrate" additionally returns the total energy of the system. Note that in C function calls in return statements can be done. The calculation is most conveniently done in this part of the total program and is usually necessary, in order to to check for energy conservation of the integration.

For other ensembles, many different algorithms have been devised, see e.g. [416, 417, 359]. One main difference in the treatment of such boundary conditions in particle based methods that integrate Newton's classical equations of motion and the treatment of boundary conditions in mesh-based engineering methods, is, that with methods based on particles, the boundary conditions are incorporated into the equations of motion of the particles. This means, that e.g. for the simulation of a canonical ensemble where energy may fluctuate, but temperature is kept constant, the equation of motion is changed. In this case, instead of Newton's equation, a *Langevin Equation of motion* is integrated which reads in reduced simulation units

$$\ddot{\mathbf{r}}(t) = -\gamma \mathbf{v}(t) + \mathbf{f}(t) + \boldsymbol{\Gamma}(t). \quad (6.89)$$

with $-\gamma \mathbf{v}(t) = \mathbf{F}_\gamma/m$, $\mathbf{f}(t) = \mathbf{F}/m$, $\boldsymbol{\Gamma}(t) = \mathbf{F}_S/m$ and γ being the friction coefficient. This kind of particle dynamics is also celled *Stochastic Dynamics* (SD) or

Algorithm 6.6 Code fragment of the all-particle search

```

1 #Code fragment of a Verlet integration
2 double Integrate(T_Particle particle, T_Parameter *
3     parameter){
4     int i;
5     double t_2, t_2Squared, pos_i_x, pos_i_y, pos_i_z,\ 
6         v_i_x, v_i_y, v_i_z, f_i_x, f_i_y, \
7         f_i_z, EkinTotal;
8
9     t_2           = 0.5 * parameter -> tStep;
10    t_2Squared   = parameter -> tStep * parameter -> t_2;
11    E_kin        = 0.0;
12    for (i = 0; i < parameter -> NumParticles; i++) {
13        pos_i_x    = particle[i].pos_x;
14        pos_i_y    = particle[i].pos_y;
15        pos_i_z    = particle[i].pos_z;
16        v_i_x      = particle[i].v_x;
17        v_i_y      = particle[i].v_y;
18        v_i_z      = particle[i].v_z;
19        f_i_x      = particle[i].force_x;
20        f_i_y      = particle[i].force_y;
21        f_i_z      = particle[i].force_z;
22        particle[i].pos_x += tStep * v_i_x + t_2Squared * 
23            f_i_x;
24        particle[i].pos_y += tStep * v_i_y + t_2Squared * 
25            f_i_y;
26        particle[i].pos_z += tStep * v_i_z + t_2Squared * 
27            f_i_z;
28    }
29    for (i = 0; i < parameter -> NumParticles; i++) {
30        v_i_x = (particle[i].v_x += t_2 * particle[i].
31                  force_x);
32        v_i_y = (particle[i].v_y += t_2 * particle[i].
33                  force_y);
34        v_i_z = (particle[i].v_z += t_2 * particle[i].
35                  force_z);
36        E_kin += v_i_x * v_i_x + v_i_y * v_i_y +\
37                  v_i_z * v_i_z;
38    }
39    return (0.5 * E_kin + ForceCalc(particle, parameter) );
40 }
```

Brownian Dynamics Simulation (BD). The properties of the Langevin-force $\boldsymbol{\Gamma}_i(t)$, follow from the *fluctuation-dissipation theorem* [418]:

$$\gamma = \frac{1}{6k_B T} \int_{-\infty}^{\infty} \langle \boldsymbol{\Gamma}_i(0) \boldsymbol{\Gamma}_j(t) \rangle dt. \quad (6.90)$$

Equation (6.90) elucidates the connection between the energy dissipation of a system due to a frictional force with coefficient γ and the thermal fluctuations due to the influence of the surrounding.

Reduced Units

Computers can only process bits that are combined to represent numbers, cf. Fig. 4.5. In order to avoid processing very small or very large numbers, which tends to introduce huge numerical errors in calculations, and to present results independent of specific systems of unit, one introduces dimensionless, so-called *reduced units*. Reduced units simplify the basic equations. With a Lennard-Jones potential, all lengths are scaled with the characteristic length σ , i.e. the scaled length \mathbf{r}^* is:

$$\mathbf{r}^* = \mathbf{r}/\sigma . \quad (6.91)$$

The energy is scaled according to ε , i.e.

$$E^* = E/\varepsilon . \quad (6.92)$$

The reduced force is introduced by

$$\mathbf{F}^* = \sigma/\varepsilon \mathbf{F} . \quad (6.93)$$

Newton's equation of motion thus becomes:

$$\varepsilon/\sigma \mathbf{F}^* = m\sigma \frac{d^2 \mathbf{r}^*}{dt^2} . \quad (6.94)$$

Time is rescaled according to

$$t^* = t/C , \quad (6.95)$$

where C is chosen as

$$C = \sqrt{\frac{m\sigma^2}{\varepsilon}} , \quad (6.96)$$

such that in Newton's equation, the force is identical with the acceleration (thus m can be formally set to 1). As a result, time in reduced units is:

$$t^* = \sqrt{\frac{\varepsilon}{m\sigma^2}} t . \quad (6.97)$$

The reduced density of a system is given by

$$\rho^* = \rho\sigma^3 , \quad (6.98)$$

and temperature is

$$T^* = \frac{k_B T}{\varepsilon} = \frac{1}{\varepsilon} , \quad (6.99)$$

as $k_B T$ has the dimension of an energy. Thus, temperature is also scaled with $1/\varepsilon$.

Hence, using reduced units, the LJ force is computed as:

$$\mathbf{F}_i = \ddot{\mathbf{r}}_i = \begin{cases} 24\varepsilon \left[2\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right] \frac{r}{r^2} & r \leq r_{\min}, \\ 0 & \text{otherwise.} \end{cases} \quad (6.100)$$

In Algorithm 6.7 a code fragment exhibits the force calculation for a LJ fluid.

Algorithm 6.7 Code fragment of a brute force $\mathcal{O}(N^2)$ LJ force calculation

```

1 void ForceCalculation (T_Parameter *parameter, T_Particle particle){
2     for (i = 0; i < (n-1); i++){
3         for (j=i+1; j<n; j++){
4             /*calculation of the distance-vector */
5             r_ij_x = particle[i].pos_x - particle[j].pos_x;
6             r_ij_y = particle[i].pos_y - particle[j].pos_y;
7             r_ij_z = particle[i].pos_z - particle[j].pos_z;
8             /* set the distance-vector back into the simulation box */
9             r_ij_x -= base_x * Backfold(r_ij_x/base_x);
10            r_ij_y -= base_y * Backfold(r_ij_y/base_y);
11            r_ij_z -= base_z * Backfold(r_ij_z/base_z);
12            r_ij_2      = r_ij_x * r_ij_x + r_ij_y * r_ij_y + \
13                           r_ij_z * r_ij_z;
14            r_ij_2_inv  = 1.0 / r_ij_2;
15            r_ij_6_inv  = r_ij_2_inv * r_ij_2_inv * r_ij_2_inv;
16            r_ij_12_inv = r_ij_6_inv * r_ij_6_inv;
17            r_ij_3_inv  = r_ij_inv * (1.0/r_ij_2);
18            if(r_ij_2 < cutoff_2){
19                f_per_r = 24.0 * ((2.0 * r_ij_12_inv) - r_ij_6_inv) \
20                           * r_ij_2_inv;
21                f_ij_x          = f_per_r * r_ij_x;
22                f_ij_y          = f_per_r * r_ij_y;
23                f_ij_z          = f_per_r * r_ij_z;
24                particle[i].force_x += f_ij_x;
25                particle[i].force_y += f_ij_y;
26                particle[i].force_z += f_ij_z;
27                /* actio == reactio */
28                particle[j].force_x -= f_ij_x;
29                particle[j].force_y -= f_ij_y;
30                particle[j].force_z -= f_ij_z;
31            }
32            particle[i].force_x += (factor * r_ij_3_inv * r_ij_x);
33            particle[i].force_y += (factor * r_ij_3_inv * r_ij_y);
34            particle[i].force_z += (factor * r_ij_3_inv * r_ij_z);
35            particle[j].force_x -= (factor * r_ij_3_inv * r_ij_x);
36            particle[j].force_y -= (factor * r_ij_3_inv * r_ij_y);
37            particle[j].force_z -= (factor * r_ij_3_inv * r_ij_z);
38        } } }
```

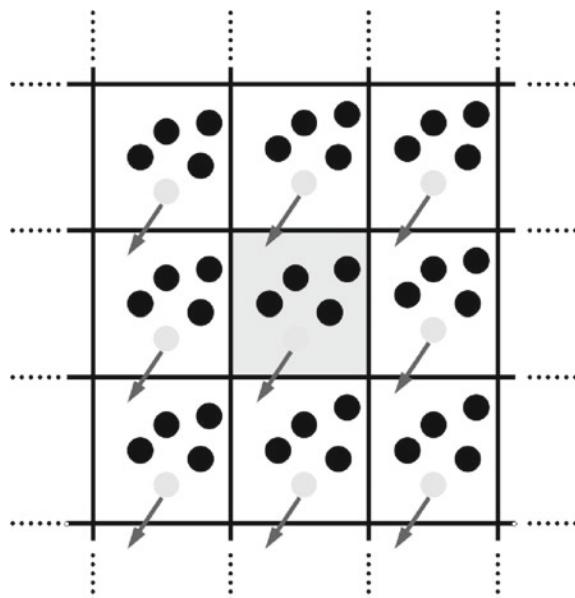
6.4.3 Periodic Boundary Conditions

In a MD simulation, the behavior of only a small number of particles compared with the number of atoms in a microscopic or larger system can be simulated. Therefore, in order to obtain the bulk properties of the behavior of materials, one introduces *periodic boundary conditions* (PBC) for the considered simulation box. By introducing PBC edge effects in a finite system are suppressed. The atoms of the system to be simulated are put into a space-filling simulation box, which is surrounded by translated copies of itself, cf. Fig. 6.8. This means, that the artefact caused by unwanted boundaries in an isolated cluster of particles is now replaced by the artefact of periodicity. For the simulation of a solid state, such as a crystal, periodic boundary conditions are desired, as they reflect the symmetry of the system, albeit the dynamics is restricted to periodic motions within wavelengths fitting into the box. In Algorithm 6.7 an example for periodic backfolding can be seen in the three lines of code making use of the C-function “Backfold”

$$\text{Backfold}(x)((\text{double})(\text{(int)}((x) + 0.5) - ((x) < -0.5)))) , \quad (6.101)$$

which returns the next positive integer. Once a particle has crossed the boundaries it is periodically back-folded into the simulation box. Note, that this periodic wrap-around is done in the *innermost* loop of the force calculation and therefore is extremely computationally expensive. For non-periodic systems, such as liquids or solutions, the introduced periodicity causes errors by itself. These errors however can be estimated by simulating successively larger systems and comparing the obtained results. Errors resulting from periodicity in general are less severe than errors resulting e.g. from

Fig. 6.8 Two-dimensional illustration of periodic boundary conditions in a cubic simulation box. The particle coordinates in the central simulation box are copied in every direction. When calculating observables, the particle coordinates are backfolded into the central box, and analyzed using the minimum image convention according to which the real distance between any two particles is given by the shortest distance of any of their images



an unnatural boundary with vacuum. The shape of a space-filling simulation box with unit cells is not restricted to a cube and also rhombic dodecahedra⁹ or truncated octahedra are used [419]. The main advantage of the latter shape compared to the cube is, that its volume is only about 71 % of the volume of a cube having image cells at the same distance.

6.4.4 The Minimum Image Convention

The question whether the properties of a small, infinitely extended system and of a macroscopic system, which is represented by the smaller system, are the same will depend on the range of the potential and on the characteristics of the investigated phenomenon. For a LJ-potential according to (6.81) the small cut-off radius¹⁰ ensures that no particle may interact with one of its image particles and thus, “realize” the artificial periodicity of the lattice. When calculating interactions among particles, one can only take particles into account which are separated at the most by Boxlength/2. This approach for calculating interactions is called *minimum image convention*.

6.4.5 Efficient Search Strategies for Interacting Particles

The most crucial part of an MD simulation, no matter whether the code is parallelized or not, is the force calculation, i.e. the search for potential interacting neighbors of particles. A big difference between the MD method and mesh-based methods, e.g. the finite element method, is, that all node points are firmly attached to each other within a mesh at the beginning of the simulation. In this case, by the use of abstract data types, one can very efficiently search for nodes and their neighbors. In MD simulations, the particles move freely about in the volume of the simulation box and thus, potential interacting partners have to be determined at each timestep. In contrast to systems with long-range forces, for short-range potentials, there is no problem in the force calculation, because the potential is cut off at a certain distance and one can conveniently use a neighbor list.¹¹ This type of calculation is called in coordinate space is called *particle-particle method* and the computational effort scale ideally with the number of particles N . Figure 6.9 exhibits different search strategies for the determination of potentially interacting particles.

The least efficient search method is a “brute force” approach which simply calculates the interactions of each single particle with all other particles, which is very inefficient and only useful for very small system up to $N \sim 1000$, cf. Algorithm 6.7. With this method the central simulation box is divided into sub-cells which have

⁹This is actually the smallest and most regular space filling unit cell.

¹⁰Compared to the subcell box size.

¹¹Generally, it is more advantageous to use only one list, also in the case of several different cut-offs in the system. In this case, the list should be sorted according to cutoff, and the largest cutoff should be used to determine the sub-cell size for the sorting of particles into cells.

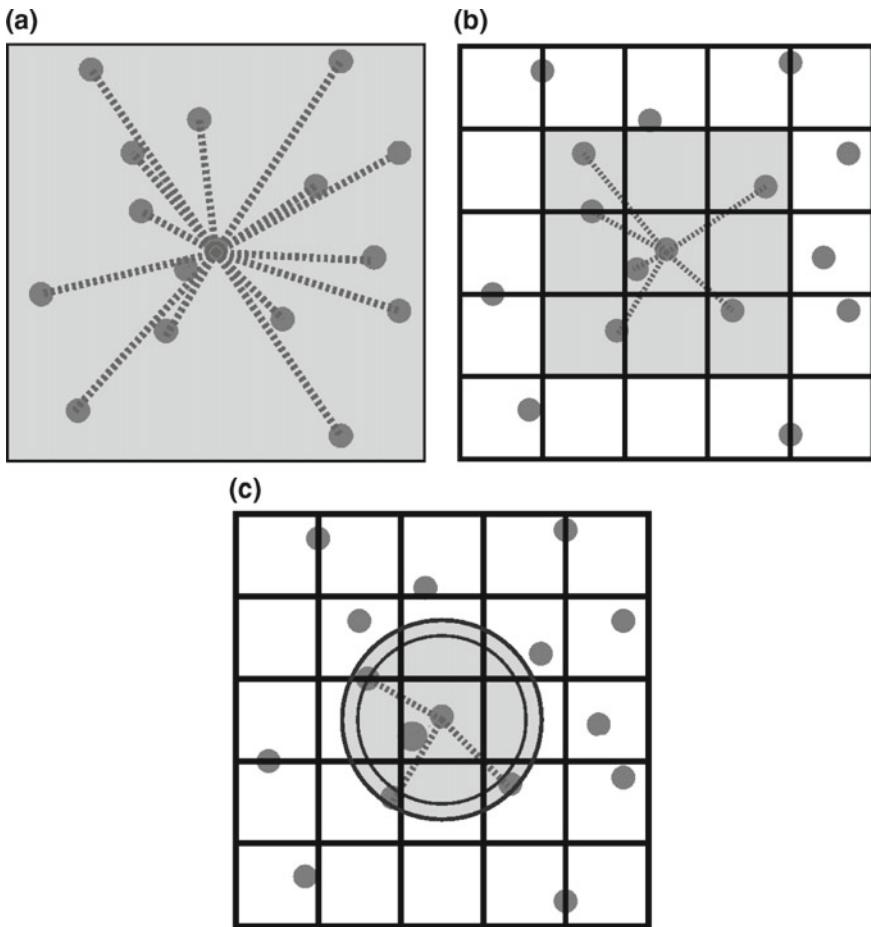


Fig. 6.9 Different search strategies for identifying potentially interacting particles. **a** $\mathcal{O}(N^2)$ all particle approach. **b** Linked cell method with effort $\mathcal{O}(N)$. **c** Linked cell method combined with neighbor lists. The shaded area displays the size of the region which is searched for potential interacting partners

a length of at least one interaction cutoff r_{cut} . Thus, the number of distance calculations is restricted to those particle pairs of neighboring cells only. Assuming the average particle density in the simulation box as $\langle \rho \rangle$ then the number of particles in each one of the subcells is $\langle \rho \rangle r_{\text{cut}}^3$. The total number of subcells is $N / \langle \rho \rangle r_{\text{cut}}^3$ and the total number of neighbor cells of each subcell is 26 in a cubic lattice in three dimensions. Due to Newton's third law only half of the neighbors actually need to be considered. Hence, the order to which the linked-cell algorithm reduces the effort is of $13(\langle \rho \rangle r_{\text{cut}}^3)^2 N$. For this method to function, the size of the simulation box has to be at least $3r_{\text{cut}}$, cf. Fig. 6.9. For simulations of e.g. dense polymer melts or with ordinary many particle fluids, this requirement is usually met.

Ghostparticles

Another method of gaining speed in a MD simulation is to remove any mentioning of periodic boundaries in the force calculations. This can be done by using the concept of *ghost particles*, see e.g. [54]. With this concept, there are two sets of coordinates. The back-folded *periodic coordinates* which are used to sort the particles into the sub-cells and the *free coordinates* which are not back-folded at all and which are the ones used for later calculations of observables. The idea with the ghost particles is the following: All particles that are in sub-cells which are on the surface of the simulation box are being duplicated right away into the extra ghost cells surrounding the whole box. These ghost particles are now used for distance calculations instead of the original coordinates. As a result, the periodic back-folding only has to be done for a relatively small number of surface particles in the outermost sub-cells, but not anymore for the many particles in the force calculations. As a result of Newton's third axiom, one needs only half of the ghost cells surrounding the original box. With the use of ghost cells, the number of adjacent neighbor cells now depends upon the location of the considered cell. In a cube there are 18 different cases and again due to Newton's third axiom only half of them need to be considered. The sub-routine that contains the search algorithm which examines adjacent cells makes provision for all 9 different cases. The *number* of adjacent cells for these different cases is fixed, see Fig. 6.10, and can be written in a static array, which is computationally efficient.

The additional advantage of using ghost cells lies in the fact, that the effort of setting up the cells and the effort of book-keeping *decreases* with system size, as the number of cells in the outermost layers of the simulation box *decrease*. E.g., for a system with $N = 1000$ particles, on average $\approx 73\%$ of all particles are ghosts, whereas this number has decreased to an average value of $\approx 13\%$ for a system with the same density, but $N = 200,000$. Using this technique for larger system can result in a overall speed-up of up to a factor of 2. The individual ghost layers in each direction are set up in the in three steps: An individual sub-cell can be identified by the three integers (m_x, m_y, m_z) . First of all, all particles of sub-cells with $(m_x = 1, \dots, n_x, m_y = 1, \dots, n_y, m_z = 1, \dots, n_z)$ are duplicated into the appropriate ghost cells. The second ghost layer contains particles pertaining to $(m_x = 1, \dots, n_x, m_y = 1, m_z = 1, \dots, n_z)$, *including* the ghost particles of the first layer. Finally, the third ghost-layer contains particles with $(m_x = 1, \dots, n_x, m_y = 1, \dots, n_y, m_z = 1)$ *including* the ghost particles of layers 1 and 2. The scanning through neighbor cells when adding up the interactions can be done in exactly the same way without using ghost particles. Only all ghost cell pairs of *the same ghost layer* have to be left out in the interaction calculation, e.g. in the two layers shown in Fig. 6.10 this applies to cells 4, 8, 12, 13 – 16 and cells 20, 24, 28, 29 – 32. In the force summation of the interactions one has to take care not to double-count any interaction between particle pairs by making sure *not* to add interactions between ghost particles of the same ghost layer.

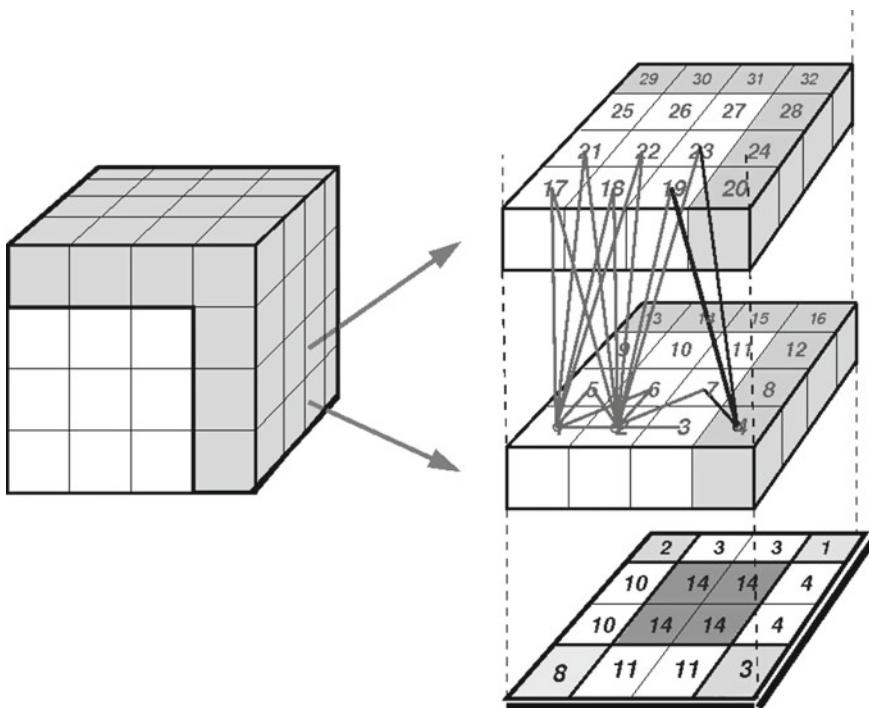


Fig. 6.10 Illustration of the search-algorithm for ghost particles with a fixed number of neighbor cells depending upon the location of the considered cell. The lowest square depicts the 9 possible cases of cell locations and displays the respective number of neighbor cells, always including the cell itself. Two layers of the simulation box are displayed along with the numbering of the sub-cells. The ghost cells are coded in dark color. As an example, the locations of the adjacent sub-cells that are scanned by the search-algorithm are displayed for three different sub-cells (cells 1, 2 and 4)

6.4.6 Making Measurements

Once the system has been run for a while and the energy and other variables fluctuate around a mean value, measurements can be done. Sample snapshots of a molecular fluid both in the initial setup (which is far away from equilibrium) and after a micro-canonical simulation run for 4 and 1 LJ time, respectively, are shown in Fig. 6.11. In this particular initial set-up the particles are arranged at their equilibrium distances and at time step $t = 0$ the full LJ potential acts on the particles which are assigned random initial velocities. It is not useful to waste much time generating on the initial configuration, as it is artificial anyway and the system will very quickly approach a Maxwell (equilibrium) velocity distribution.

This, however is different when simulating macromolecules – such as polymers – due to their particular connectivity and entropic effects that came into play. The decision as to when a system is at equilibrium is not an easy question to answer in a simulation. Usually one checks the decay of correlation of consecutive snapshots

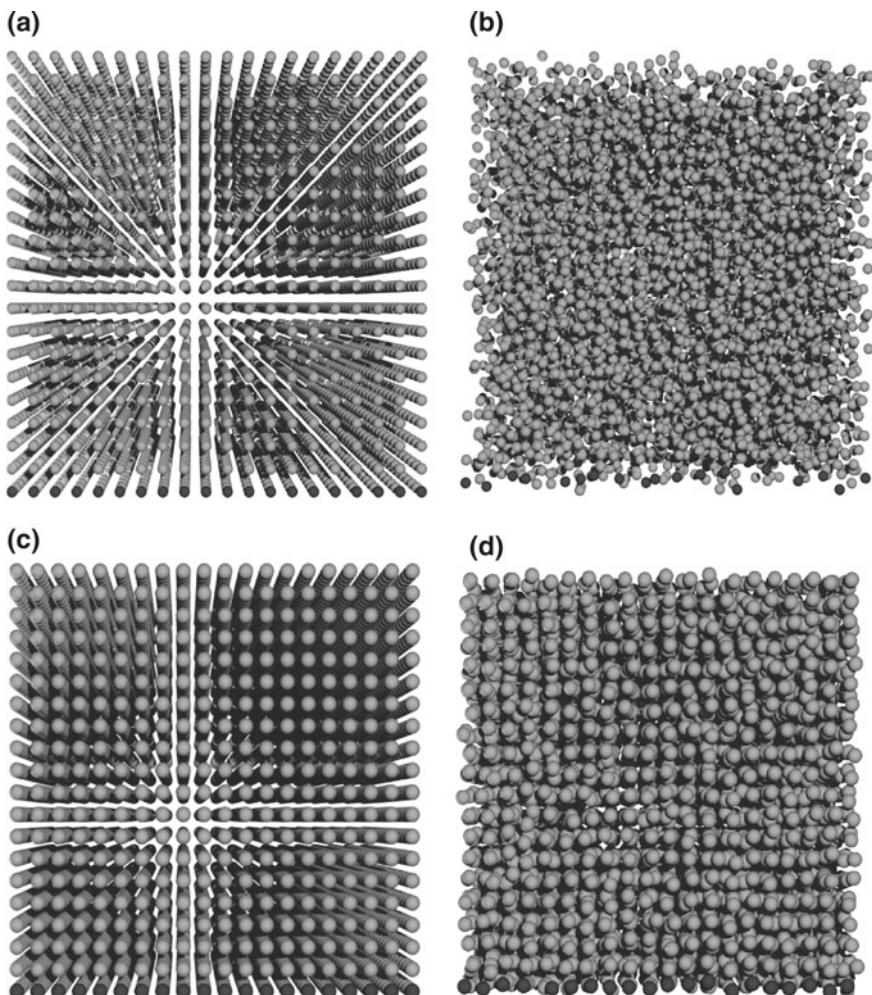
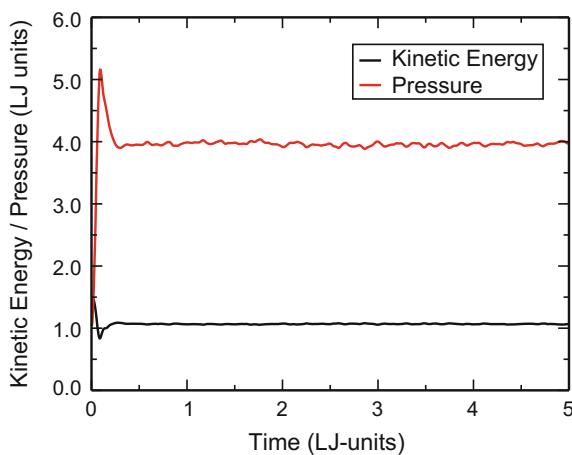


Fig. 6.11 Simulation snapshots of a micro-canonical molecular dynamics simulation of a Lennard-Jones fluid, starting from a cubic initial configuration with $N = 8000$ particles. **a** Initial cubic configuration of fluid particles at density $\rho = 0.4$. **b** System snapshot after 4 reduced time units. **c** Initial configuration of particles in a simulation box with $\rho = 0.8$. **d** System snapshot after 1 reduced time unit. The lowest front row of particles in (a) and (c) were darkly colored to show their diffusion in the fluid

as well as typical configuration properties, such as coordination numbers, energy, pressure, or – in the case of molecules – the average extension, e.g. the radius of gyration of the molecules – as an indication for the system to have reached an equilibrium state. In polymer melt simulations, one usually lets the system evolve for several times the typical relaxation time τ of molecules in a melt. The pressure

Fig. 6.12 Kinetic energy and pressure of a molecular LJ fluid. After a short initial equilibrium phase, the values fluctuate about their mean values



p of a fluid is defined in terms of a virial expansion. In reduced units this writes as:

$$pV = NT + \frac{1}{d} \left\langle \sum_{i=1}^N \mathbf{r}_i \mathbf{F}_i \right\rangle. \quad (6.102)$$

In Fig. 6.12 the energy and the pressure of a Lennard-Jones fluid are displayed. After an initial equilibration phase, both quantities have reached their equilibrium value and fluctuate gradually about their mean value.

Scattering Experiments with Fluids

Scattering experiments are performed with electromagnetic radiation, neutrons, or electrons. In all cases, the principle of a scattering experiment is always the same: A primary beam with frequency ω_0 , intensity I_0 and wave vector \mathbf{k} hits a target specimen and initiates scattering waves. The scattered intensity I of the secondary waves emerging from the target, is measured. I in general depends on the direction of observation and the angle θ by which the direction of observation diverges from the direction of the incoming beam, is called *Bragg angle*. In order to analyze the structure of a specimen, one has to choose a wave length of the incoming radiation or particles which is comparable to a typical length scale of the investigated structure, e.g. a lattice constant in the case of a solid state. Other nanoscopic structures, e.g. molecular fluids typically have structural dimensions on the order of (1–10) Å. Hence, for high resolution microscopy electrons can be used which have may wave lengths below 1 Å according to (2.2). Also thermal neutrons, with energies in the range of ~ 10 meV may be used. With colloidal systems (soft matter) much larger typical length scales up to μm may arise. Thus, for such systems, photons (light) may be used.

Electrons interact with matter via the electromagnetic force. This is the reason, why specimen investigated by X-ray microscopy have to be small – the electrons interact with the electrons of the atoms in the material and dissipate energy in the specimen, which is why the scattered intensity will be rather low. Neutrons however

can be used with large specimen, as they are electrically neutral and thus can penetrate deeper in a material. Information on the structure of the specimen is obtained by measuring the scattered intensity from different Bragg angles. Differences in intensity are a consequence of the interference of emerging waves from the particles of the specimen. Using elementary electrodynamics, the outgoing primary beam can be described as plane wave with amplitude E_0 :

$$E_0 \exp(-i\omega_0 t + i\mathbf{k}\mathbf{r}) . \quad (6.103)$$

This wave hits the specimen and initiates oscillations of atoms and molecules. Due to these oscillations, secondary waves are emitted from the specimen in direction \mathbf{k}' . To simplify calculations one introduces the scattering vector

$$\mathbf{q} = \mathbf{k}' - \mathbf{k} . \quad (6.104)$$

The total amplitude E of the scattered waves is obtained by summing over all contributions of the scatterers in the material, i.e. one has to calculate the sum

$$E = E(\mathbf{q}) \sim \sum_j f_j \exp(-i\mathbf{q}\mathbf{r}_j) . \quad (6.105)$$

where parameter f_j is the so-called *form factor* which is introduced, as the scattered waves of molecules are different from the ones scattered by electrons. Most often however, scattering experiments are done with specimen where the scatterers are only made of one kind of particle. In this case, the amplitude E only depends on the phase factor of the scattered wave, i.e.

$$E = E(\mathbf{q}) \sim \text{sum}_j \exp(-i\mathbf{q}\mathbf{r}_j) = C(\mathbf{q}) . \quad (6.106)$$

Any detector is not able to follow the fast changing electromagnetic field vector $\mathbf{E}(\mathbf{r}, t)$. Rather, all measurements are averages of some quantities, in this case the average of the different intensities of the quickly changing field vector. This average is called *intensity* I and this is what is measured in a scattering experiment:

$$I(\mathbf{q}) \sim \langle |E_0(\mathbf{q})|^2 \rangle = \langle |C(\mathbf{q})|^2 \rangle \quad (6.107)$$

The total intensity measured in a scattering experiment is proportional to the number N of scattering particles. Hence, one introduces a function $S(\mathbf{q})$ as

$$S(\mathbf{q}) = \frac{1}{N} \sum_{j,k=1}^N \langle \exp[-i\mathbf{q}(\mathbf{r}_j - \mathbf{r}_k)] \rangle . \quad (6.108)$$

$S(\mathbf{q})$ is called *Structure factor*¹² $S(\mathbf{q})$ does not depend on the nature of the scatterers, is universally valid and can thus be applied to any condensed matter system. For molecular fluids, the scattering function can be written using the pair correlation function $g(\mathbf{r})$ as:

$$S(\mathbf{q}) = \frac{1}{N} \left(N + N \int_V \exp(-i\mathbf{q}\mathbf{r}) g(\mathbf{r}) d^3 r \right) = 1 + \int_V \exp(-i\mathbf{q}\mathbf{r}) g(\mathbf{r}) d^3 r . \quad (6.109)$$

¹²Sometimes $S(\mathbf{q})$ is also called scattering function or interference function.

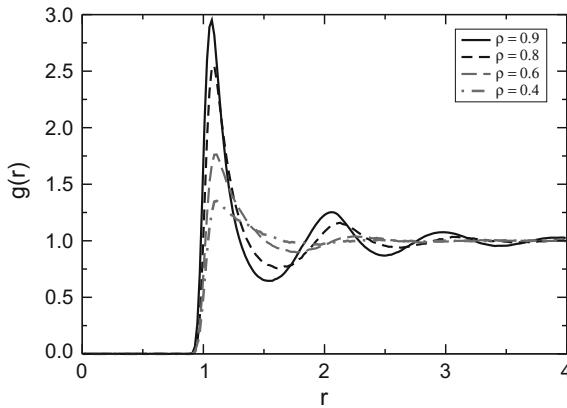


Fig. 6.13 The radial distribution function (RDF) $g(r)$ of a molecular fluid with different densities as displayed. For a visualization of the structures, see Fig. 6.11. The RDF counts the number of particles found in a spherical shell of radius r in the surrounding of a picked particle. The denser the systems, the more structure (pronounced peaks in $g(r)$) can be seen and the more extended is the function $g(r)$

The first term on the right hand side of (6.109) is the contribution to sum by all terms for which $i = j$. The second term is the sum over all contributions from particles with $j \neq k$. The integral in (6.109) is the Fourier transform of $g(\mathbf{r})$. For $r \rightarrow \infty$, the limiting value of $g(r)$ is the particle density ρ . This limiting value has to be subtracted from $S(\mathbf{q})$ to obtain:

$$S(\mathbf{q}) - 1 = \int_V \exp(-i\mathbf{qr})g(\mathbf{r} - \rho) d^3r + \rho \int_V \exp(-i\mathbf{qr})d^3r . \quad (6.110)$$

The second term is the Fourier transform of a macroscopic volume which has only contributions for $\mathbf{q} \approx 0$. This forward scattering has to be avoided in practical experiments to avoid detecting primary waves. Hence, this term can usually be neglected. For fluids, $S(\mathbf{q})$ is isotropic which means that

$$g(\mathbf{r} = g(|\mathbf{r}| = r)) , \quad (6.111)$$

and

$$S(\mathbf{q}) = S(|\mathbf{q}|) . \quad (6.112)$$

For isotropic functions, the Fourier transform may be written as

$$S(q) - 1 = \int_{\infty}^{\infty} (g(r) - \rho) 4\pi r^2 \frac{\sin(qr)}{qr} dr \quad (6.113)$$

and the inverse transformation is

$$g(r) - \rho = \left(\frac{1}{2\pi} \right)^3 \int_{r=0}^{\infty} (S(q) - 1) 4\pi r^2 \frac{\sin(qr)}{qr} dr . \quad (6.114)$$

In Fig. 6.13 some exemplary measurements of the MD simulation of a molecular fluid are displayed for four different densities of the fluid. The maxima of $g(r)$ correspond to different shells of neighbors.

6.5 Liquids, Soft Matter and Polymers

When looking closely at an ordinary fluid, e.g. water, under a microscope, one realizes that this is an inhomogeneous, structured material, which is made up of atoms or molecules. However, above a resolution of ~ 1 nm, such fluids look entirely homogeneous, and no structure is detectable any more, cf. also Fig. 6.11. This is very different when looking at so-called *complex fluids*, e.g. suspensions of particles in oil or water. For such systems, one still observes complex structures at a resolution of several 100 nm. As this structuring is of a size that is beyond the size of the involved particles and molecules, these structures are also called *supramolecular aggregates*. Only on a scale of micrometers or even larger, these fluids look homogeneous. The structural length, which surmounts the size of atoms by several orders of magnitude, has important consequences for the properties of these systems. Particularly, materials made of extremely large molecules, so-called polymers, are easy to deform. This is the reason why such materials are called *soft matter* which has displaced the outdated denotation “complex fluids”. Polymers generally are a paradigm for *soft matter* materials. Roughly speaking, soft matter consists of materials, whose constituents have atomic size, but which organize – or self-assemble – to much larger entities, which exhibit different structural features on the micro- and mesoscale. Examples other than polymers are liquid crystals, colloidal suspensions, fluid membranes or microemulsions [421].

The time scale and size scale problems of simulations are particularly abundant in polymer physics and cover many orders of magnitude. A polymer is a large molecular chain, a macromolecule, which is composed of many smaller chemical repeating units which are covalently bound to each other. The largest repeating unit which is contributed by a single molecule in a polymerization process is called a monomer and the number of repetitive units is called the degree of polymerization N . Experimentally, the chain length N is large, typically $10^3 \leq 10^5$ and the size of a polymer chain ($\mathcal{O}(10^3 \text{ \AA})$) exceeds that of a monomer ($\mathcal{O}(1 \text{ \AA})$) by several orders of magnitude. In fact, the largest known naturally occurring macromolecule is DNA, cf. Fig. 6.14, which is packaged in the cell nuclei of all organisms. This packaging is mediated by proteins and protects DNA from physical damage and from free radicals, which are extremely reactive molecules. Different species package DNA differently, but the mechanisms of how different proteins package DNA into distinct shapes are poorly understood.

Due to their large molecular masses, polymers possess an enormous variety of chain conformations with different sizes and different topological connectivities of the monomer units, e.g. as linear chains, star polymers, brush polymers or dendrimers. The study of these conformations is one key to understanding the diverse properties of macromolecules. Polymers show a wide spectrum of properties and of possible material applications, because of their chemical variability and the interplay of entropy and energy which determines their structure. For a very readable review on the physical chemistry of polymers, see [422].

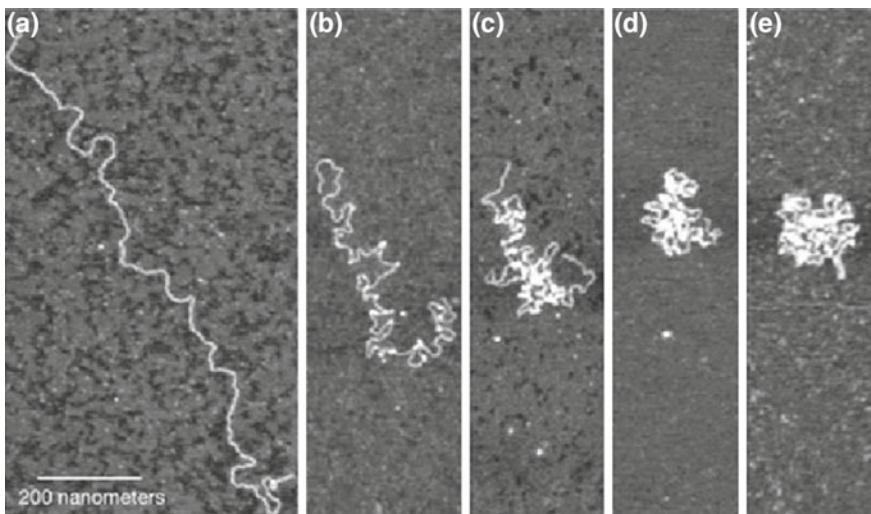


Fig. 6.14 Progressive images from atomic force microscopy which show the compaction of DNA in yeast caused by a protein called AbF2. Taken from an article edited by Arnie Heller and Ann Parker [420]. Republished with permission granted by the University of California, Lawrence Livermore National Laboratory, and the Department of Energy under whose auspices the work was performed

Bonded Interactions

With molecular fluids, no particular bonded interaction has to be taken into account, i.e. the fluid particles are not bound to each other and move freely within the simulation box, cf. Fig. 6.3. With molecules or very long polymers, their particular connectivity, i.e. the linear or branched connectivity of molecules or monomers has to be taken into account in a simulation model. In particular with polymer systems, many properties can be successfully simulated by only taking into account the basic and essential features of the chains and neglecting the detailed *chemical* structure of the molecules. Such models, which disregard the explicit atomic, or chemical details of a structure, are called *coarse grained models*. Such models are used because atomistic models of long molecules, e.g. polymer chains are usually intractable for the time scales that need to be simulated to allow for a comparison with real experiments. Also, the fractal properties of polymers renders this type of model very useful for the investigation of scaling properties. Fractality means, that some property P of a system (with polymers, e.g. the radius of gyration R_g) obeys a relation $P \propto N^k$, where N is the size of the system and the fractal dimension k with $k \in \mathbb{Q}$ is of the form $k = p/q$ with $p \neq q$, $q \in \mathbb{N}$ and $p \in \mathbb{N}$.

The basic properties that are sufficient to extract many polymer properties are:

- Connectivity of monomers in a chain

- Topological constraints (e.g. non-crossability of different chains and chain bonds of a single chain)
- Flexibility (or stiffness) of monomer segments

The bonded interactions Φ_{bonded} for polymeric systems with internal degrees of freedom (due to the specific monomer connectivity) can be treated by using some or all parts of the following potential term:

$$\begin{aligned}\Phi_{bonded}(r, \theta, \phi) = & \frac{\kappa}{2} \sum_i (|\mathbf{r}_i - \mathbf{r}_{i-1} - l_0|)^2 \\ & + \frac{\alpha}{2} \sum_k (\theta_k - \theta_0)^2 + \frac{\beta}{2} \sum_m (\phi_m - \phi_0)^2.\end{aligned}\quad (6.115)$$

Here, the summation indices sum up the number of bonds i , the number of bond angles k between consecutive monomers along a chain and the number of torsion angles m along the polymer chain. A typical value of κ is 5000 which ensures that the fluctuations of bond angles are very small (below 1 %). The terms l_0 , θ_0 and ϕ_0 are the equilibrium distance, bond angle and torsion angle, respectively.

Sometimes also a finitely extensible non-linear elastic (FENE) potential is used which restricts the maximum bond length of a polymer bond to a prefixed value:

$$\Phi_{FENE}(r) = \begin{cases} -\frac{1}{2}\kappa R_0^2 \ln(1 - \frac{r^2}{R_0^2}) & r < R_0, \\ \infty & \text{otherwise.} \end{cases} \quad (6.116)$$

However, this potential is not as advantageous when considering breakage of polymer bonds, e.g. in the simulation of shock waves in polymer systems where failure of bonds has to be taken into account as well. The exact analytic form of the terms in Eq. 6.115 is not really decisive for coarse-grained models as long as the basic properties mentioned above are modeled correctly.

When modeling biological molecules, e.g. DNA or proteins, their characteristic primary, secondary, ternary and quaternary structures have to be taken into account. Proteins are heteropolymer molecules, that consist of 20 different monomeric units, called amino acid residues, or simply amino acids. These units are connected in a chain-like topology resulting in a *universal polypeptide backbone chain*, where different amino acids form different side groups at every chain monomer. The mechanisms responsible for the folding of proteins, by which the heteropolymer chain reaches its folded native structure, is not yet completely understood in molecular biology and is thus a current active topic of research. When modeling protein structures, several different types of force fields are used to account for the variety of structures that arise in these systems. The total energy of a biomolecular system is thus composed of many contributions:

$$\Phi = \sum_i \phi_i^B + \sum_j V_j^W + \sum_k \phi_k^D + \sum_l \phi_l^E + \sum_{ij} (\Phi_{i,j}^C + \Phi_{i,j}^P + \Phi_{i,j}^{vdW}). \quad (6.117)$$

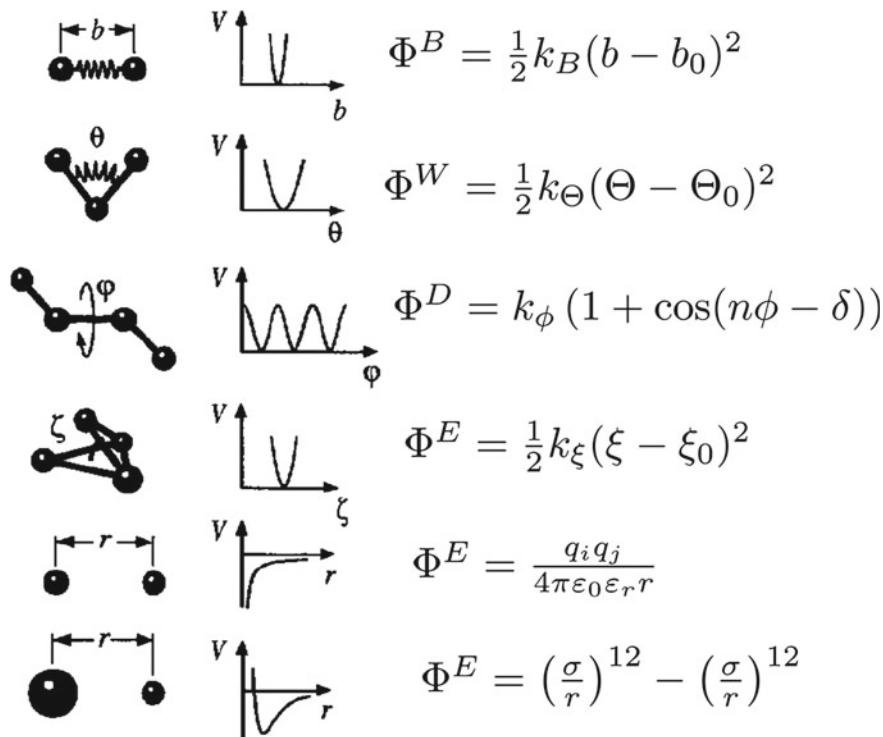


Fig. 6.15 Overview of typical force fields in polymer and biological (DNA/protein) applications. Graphics adapted from [423]. Bond stretch vibrations are described by a harmonic potential Φ^B . Bond angles and dihedral angles are taken into account by a harmonic respectively periodic potential term Φ^W and Φ^D . Out-of-plane angles can be accounted for by a harmonic potential Φ^E , where ξ_0 denotes the respective equilibrium angle. Non-bonded forces are described by the standard LJ potential and the Coulomb interaction

These contributions have been implemented in several force fields of research codes available with a free licence, e.g. CHARMM,¹³ AMBER¹⁴ or GROMOS.¹⁵ In Fig. 6.15 the different contributions to Φ are exhibited.

6.5.1 Scaling and Universality of Polymers

A simple characteristic quantity of a polymer chain is its spacial extension. For polymer chains, common quantitative measures are the (averaged squared) end-to-

¹³<http://www.charmm.org/>.

¹⁴<http://ambermd.org>.

¹⁵<http://www.gromos.net>.

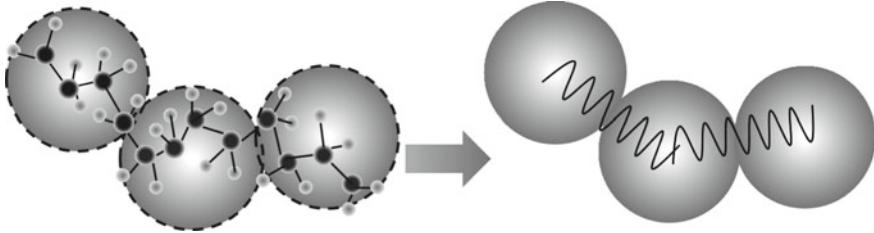


Fig. 6.16 The coarse-grained bead-spring model

end vector R_e^2 and the radius of gyration R_g^2 .

$$R_e^2 = \langle (\mathbf{r}_N - \mathbf{r}_1)^2 \rangle = \left\langle \left(\sum_{i=1}^N \mathbf{r}_i \right) \cdot \left(\sum_{j=1}^N \mathbf{r}_j \right) \right\rangle = \sum_{i=1}^N \sum_{j=1}^N \langle \mathbf{r}_i \mathbf{r}_j \rangle , \quad (6.118a)$$

$$R_g^2 = \frac{1}{N} \sum_{i=1}^N \langle (\mathbf{r}_i - \mathbf{R}_{cm})^2 \rangle , \quad (6.118b)$$

where \mathbf{R}_{cm} is the position vector of the center of mass of the chain. The radius of gyration can also be used to characterize the extension of *branched chains*, whereas R_e is limited to linear chains (Fig. 6.16). Let b_i denote the unit vector associated with the bond b_i of fixed length l_0 , then R_e can be written as

$$R_e^2 = l_0^2 \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \langle b_i b_j \rangle = 2l_0^2 \sum_{i=1}^{N-1} \sum_{k=0}^{N-1-i} \langle b_i b_{i+k} \rangle - (N-1)l_0^2 . \quad (6.119)$$

Hence, the large-scale behavior obviously depends on the range of orientational correlations between bond vectors. It is well-known, that polymers are fractal objects, which exhibit both, criticality,¹⁶ and universality on long length scales. For detailed discussions, the reader is referred to the excellent treatments of the subjects in e.g. Flory [424], Doi and Edwards [418] or de Gennes [425]. Due to the above said, one has

$$R_e \sim R_g = b N^\nu . \quad (6.120)$$

If $\langle b_i b_j \rangle$ in (6.119) decays more rapidly than $1/k$ for large k the second term converges in the $N \rightarrow \infty$ -limit and one obtains:

$$R_e^2 = N l_0^2 \left[2 \sum_{k=0}^{\infty} \langle b_1 b_{1+k} \rangle - 1 \right] = N l_0^2 \left[2 \frac{l_p}{l_0} - 1 \right] . \quad (6.121)$$

The term l_p is called *persistence length* – it measures the intrinsic stiffness of a chain, i.e. the persistence of the orientational correlations along the backbone. It is

¹⁶The existence of a critical temperature $T_c = T_\Theta$ for which repulsive and attractive interactions cancel each other.

the shortest length scale on which polymer chains still exhibit flexibility. Short-range orientational correlations only affect the prefactor – they renormalize the bond length to the *effective bond length*

$$b = l_0[2(l_p/l_0) - 1]^{1/2}, \quad (6.122)$$

but do *not* change the scaling of R_e with N . The scaling is always $R_e \sim N^{1/2}$ [418]. A polymer chain, exhibiting this random-walk-like behavior is generally termed *ideal chain*. Microscopic degrees of freedom only determine the prefactor b , the effective bond length, but not the overall global scaling with N . This fact is the reason for the interest in so-called *coarse-grained* models. When studying large-scale properties, one can simply replace the chemical details by a much simpler bead-spring model where N *effective* beads are connected by harmonic springs, cf. Fig. 6.16. Such a model is microscopically incorrect, but reproduces correctly the large-scale behavior. The general potential of a bead-spring model may be written as

$$\Phi(\mathbf{r}) = \sum_{i=1}^{N-1} \Phi_0(\mathbf{b}_i, \dots, \mathbf{b}_j, \dots, \mathbf{b}_{i+i_{\max}}) + \Phi_1(\mathbf{x}, \text{solvent}), \quad (6.123)$$

where $\mathbf{b}_i = \mathbf{r}_{i+1} - \mathbf{r}_i$ denotes the bond vector from the i th to $(i+1)$ th monomer. Φ_0 depends on the explicit chemical nature of the polymer molecule and Φ_1 depends strongly on the quality of the solvent.

If the correlation $\langle b_i b_j \rangle$ in (6.119) decays more slowly as $1/k^\alpha$ ($\alpha < 1$), then the scaling behavior is altered – the chains swell and one obtains from (6.119):

$$R_e^2 \sim Nl_0^2 \int^N dk \langle \mathbf{b}(k) \mathbf{b}_0 \rangle \sim \begin{cases} N^{2-\alpha} (\alpha < 1), \\ N \ln N (\alpha = 1). \end{cases} \quad (6.124)$$

In Fig. 6.17 some simulation snapshots of linear polymer chains for the transition from a good to a bad solvent, using a bead-spring model with a potential according to (6.127) are displayed. One can see the gradual forming of globules when changing the quality of a solvent from good to bad. Also Θ -conditions are displayed.

Typically, the bond correlations are very short ranged - thus, distant monomers along a polymer backbone are orientationally decorrelated, but still these monomers can come close to each other. In a coarse-grained spring model, this effect is accounted for by the repulsive interaction Φ_1 in (6.123) which can be expanded in terms of the monomer density $\rho(r)$ [418]:

$$\Phi_1 = \int d^3r \left(\frac{1}{2} k_B T v \rho(r)^2 + \frac{1}{6} k_B T w \rho(r)^3 + \dots \right), \quad (6.125)$$

where v is the *excluded volume* parameter and $\rho(r)$ is the monomer density. v measures the binary repulsion between two beads. From (6.125) one realizes, that the excluded volume parameter is identical with the second virial coefficient, which is generally temperature dependent. It vanished at the so-called Θ -point, where repulsive and attractive forces cancel each other. The corresponding temperature is called Θ -temperature T_Θ .

In the phase diagram of polymers, cf. Fig. 6.18, depending on the temperature, one can distinguish different regimes for polymers in solution.

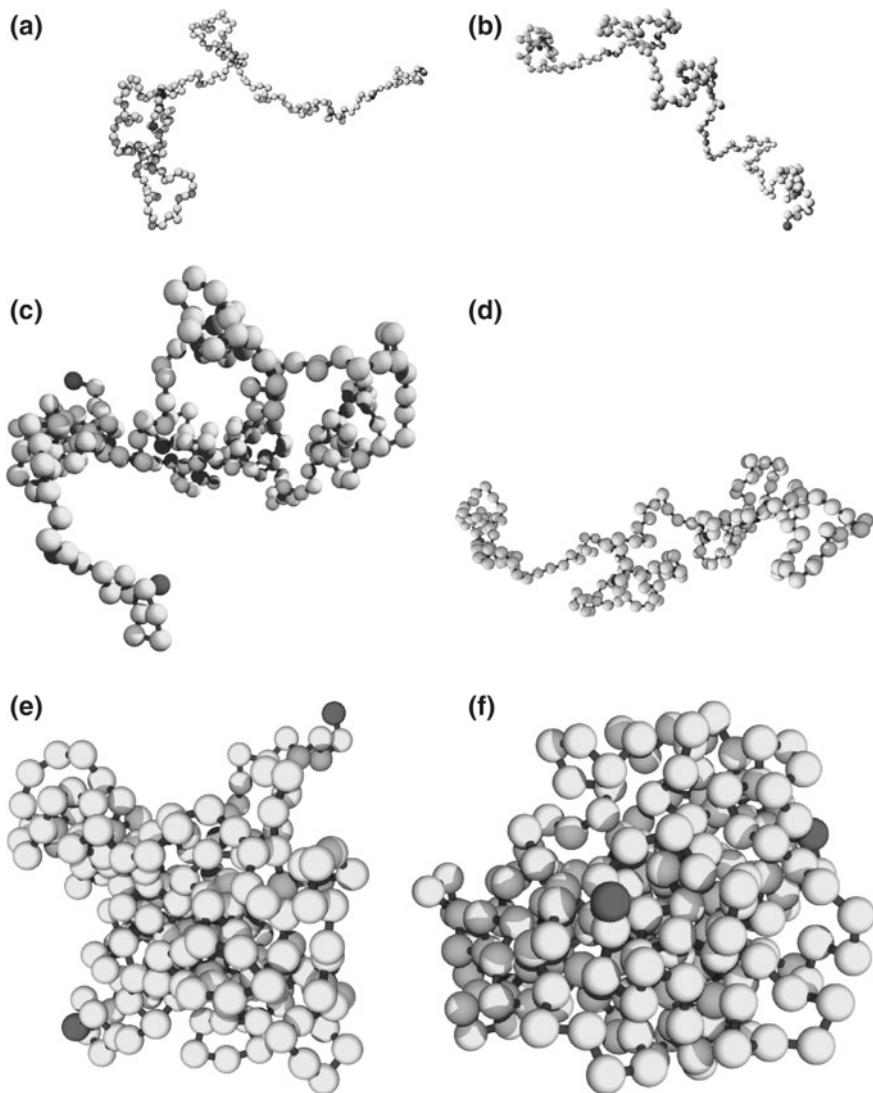


Fig. 6.17 Simulation snapshots of equilibrium configurations showing the coil-globule transition of linear polymer chains. **a** Athermal chain. **b** Good solvent, $\xi = 0.2$. **c** Good Solvent, $\xi = 0.4$. **d** Θ -solvent, $\xi = 0.6$. Bad solvent, $\xi = 0.8$. Bad solvent, $\xi = 1.0$

- Good solvent: In a good solvent the chains are swollen and scale according to $R_e \sim R_g \sim N^\nu$ with $\nu = 0.588 \approx 3/5$.
- Θ -Solvent: In a Θ -solvent, the chain has (nearly) ideal configuration, thus $R_e \sim R_g \sim \sqrt{N}$.

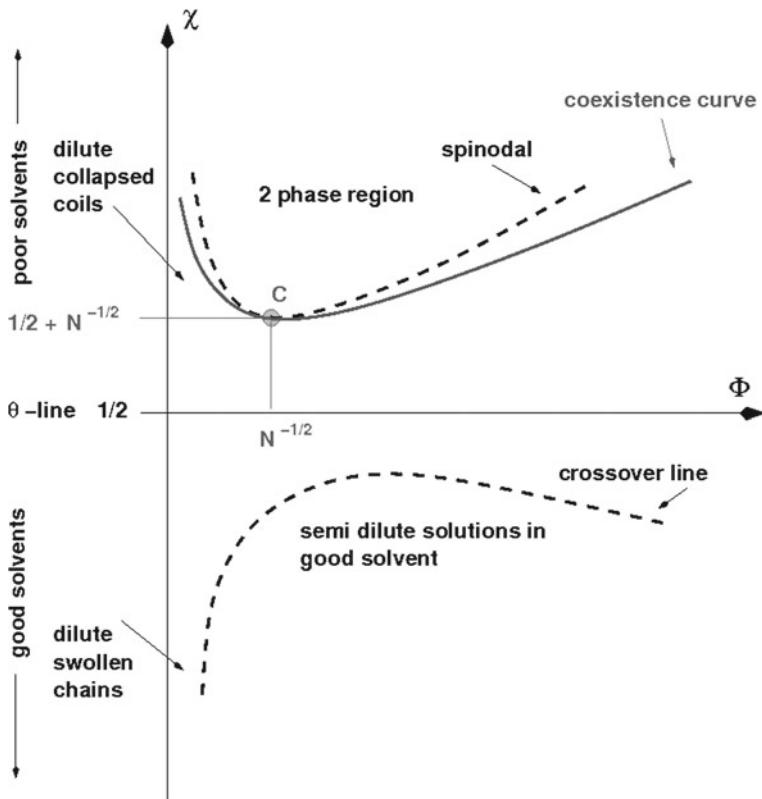
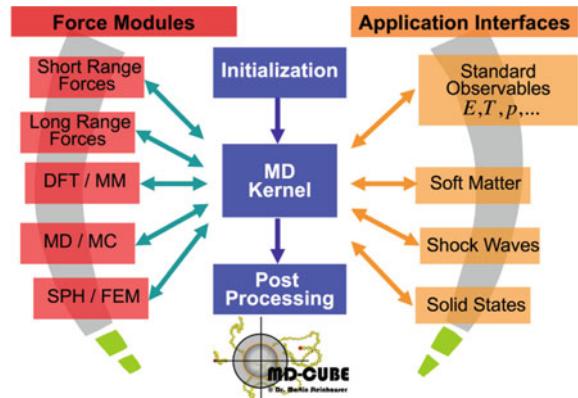


Fig. 6.18 Schematic phase diagram of a polymer. Adapted from [425]. A solvent is referred to as “good” if the effective interaction between chain segments is repulsive. The effective interaction is summarized in one dimensionless parameter χ in *Flory-Huggins-Theorie* [425]. At lower χ -values, steric interaction dominates. The chains tend to swell. One enters this good solvent regime when one crosses a certain crossover line which defines a region of crossover between ideal and swollen chains. This crossover line is not a sharp boundary and is approximately defined by the condition $\Phi < 3(1 - 2\chi)$ [425]. In usual cases such as polystyrene-cyclohexane, χ is a decreasing function of temperature T ; high temperatures correspond to the lower part of the figure. When $\chi > 1/2$, the solvent is referred to as being “poor” and there is a two-phase region and a critical point C as indicated in the figure. The critical point occurs at very low concentrations. Polymers and solvents segregate and the polymer chains collapse to compact coils. The condition $\chi = 1/2$ defines the “ Θ -condition” at “ Θ -temperature” $T = T_\Theta$, also called “ Θ -point” which corresponds to an exact cancellation between steric repulsion and van der Waals attraction between monomers (except logarithmic corrections, as there are still ternary interactions present). At $T = T_\Theta$ polymer chains possess a Gaussian conformation at all concentrations

Fig. 6.19 Scheme of one of the author's software suites MD-CUBE [426], which exhibits the modular software design, with a separate kernel and modules for calculated observables and implementing different force modules, that are coupled to the kernel by several interfaces



- Bad solvent: In a bad solvent, the chains attract each other and form globules with $R_e \sim R_g \sim N^\nu$ with $\nu = 1/3$.

Figure 6.19 shows the principal design of the software suite MD-CUBE written by the author, that has been used for a variety of applications on different scales, from the simulation of simple Lennard-Jones fluids or polymers in solution or in a melt, charged molecular systems, to shock wave applications in solid state physics. Another extension of the potential in (6.78) is given in a review [426], which allows for including different solvent qualities¹⁷ of the solvent surrounding the polymer by an additional smooth attractive term

$$\Phi_{cos}(r) = \left[\frac{1}{2} \cdot \cos(\alpha r^2 + \beta) + \gamma \right] \epsilon \quad (6.126)$$

which is added to Eq. 6.78. This additional term adds an attractive part to the potential and at the same time – by appropriately choosing parameters α , β and γ – keeps the potential cutoff at r_{cut} smooth, cf. Fig. 6.20. Further details can be found in [426].

The total unbonded potential then reads:

$$\Phi_{UB}(r) = \begin{cases} \Phi_{LJ}(r) - \xi \epsilon & 0 < r < 2^{1/6} \sigma, \\ \xi \Phi_{cos}(r) & 2^{1/6} \sigma \leq r < r_{cut}, \\ \infty & \text{otherwise,} \end{cases} \quad (6.127)$$

where ξ is a new parameter of the potential which determines the depth of the attractive part. Instead of varying the solvent quality by changing temperature T directly, one can achieve a phase transition in polymer behavior by changing ξ accordingly.

The assumption of a short ranged interaction is usually fulfilled very well for all polymeric fluids. However, as soon as charged systems are involved this assumption

¹⁷Often, polymer simulations are only done in the extreme solvent conditions of an athermal (ideally good) solvent, close to the Θ -point, or in the collapsed regime.

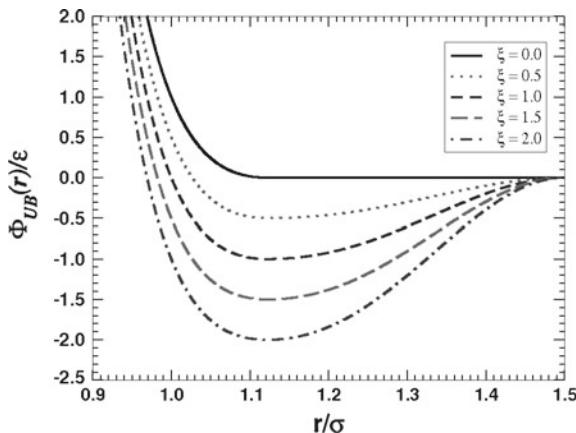


Fig. 6.20 Sample potential which is suitable for simulating different solvent qualities of a dilute polymer solution by variation of parameter ξ , which determines the depth of the attractive part of $\Phi(r)$

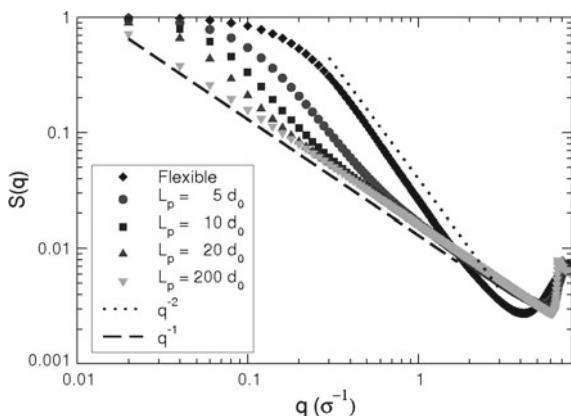


Fig. 6.21 Structure functions from simulation runs with one the author's MD codes for the transition of linear polymer chains from fully flexible to stiff chains. Different persistence lengths are displayed. Variable d_0 is the radius of one bead. Corresponding snapshots of simulated chains are displayed in Fig. 6.22

breaks down and the calculation of the Coulomb force requires special treatment due to its infinite range.

An example for using a bond angle potential Φ^W which allows for simulating stiffness of polymers is displayed in Fig. 6.21. Here the scaling behavior of the corresponding structure function is displayed upon a transition of the chains from fully flexible to rather stiff. Corresponding snapshots of these systems with different persistence lengths are displayed in Fig. 6.22.

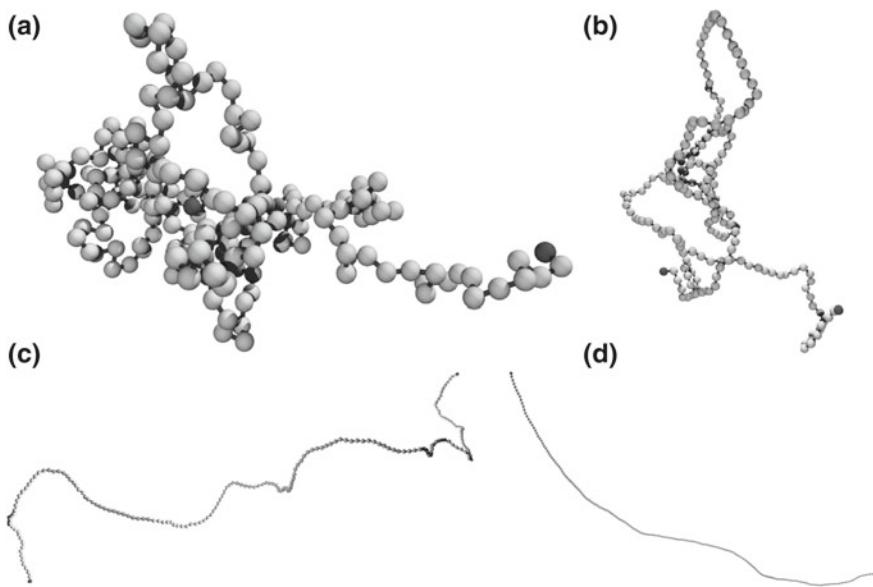


Fig. 6.22 Simulation snapshots of equilibrium configurations showing the transition of linear polymers from fully flexible to semi-flexible and stiff chains. **a** Fully flexible chain. **b** Persistence length $L_p = 5 d_0$. **c** Persistence length $L_p = 20 d_0$. **d** Persistence length $L_p = 200 d_0$. d_0 is the radius of one bead

6.6 Monte Carlo Simulations

In equilibrium statistical mechanics thermodynamic properties are calculated as ensemble averages over all points (p, q) in a high-dimensional configuration space.

In the canonical (NVT) ensemble average for example, the average of an observable $A(p, q)$ is given by

$$\langle A \rangle = \int dp dq A(p, q) P_{eq}(p, q) = \frac{1}{Z} \int dp dq A(p, q) \exp[-\beta \mathcal{H}(p, q)] , \quad (6.128)$$

where Z is the partition function

$$Z = \int \exp[-\beta \mathcal{H}(p, q)] dq dp . \quad (6.129)$$

The basic idea of the Monte Carlo scheme is to substitute the high-dimensional phase space integrals that arise when calculating thermodynamic averages by sums, and to sample the phase space according to a given probability distribution P . Denoting the phase space coordinates as X , one thus obtains:

$$\langle A \rangle \approx \bar{A} = \frac{\sum_{i=1}^N A_i(X) P^{-1}(X_i) \exp(-\beta \mathcal{H}(X_i))}{\sum_{i=1}^N P^{-1} \exp(-\beta \mathcal{H}(X_i))} . \quad (6.130)$$

The N points were chosen according to a predefined probability distribution which is called *importance sampling*. If all phase space points have the same probability, the sampling is called *simple sampling*.

According to Metropolis a chain of configurations in phase space is constructed as follows:

$$P(X_i) = P_{eq}(X_j) \propto \exp(-\beta \mathcal{H}(X_i)) . \quad (6.131)$$

As the probability P_{eq} is unknown, one constructs a random walk of points $\{X_i\}$ in phase space (a Markov-Chain) with the property

$$P(X_i) \rightarrow P_{eq}(X_i) \quad (6.132)$$

for $N \rightarrow \infty$. A sufficient condition for this to be possible is that the principle of *detailed balance* holds, which states that the transition probability $T(X_i \rightarrow X'_i)$ for $X_i \rightarrow X'_i$ obey the relation

$$P_{eq}(X_i)T(X_i \rightarrow X'_i) = P_{eq}(X'_i)T(X'_i \rightarrow X_i) . \quad (6.133)$$

From this it follows:

$$\frac{T(X_i \rightarrow X'_i)}{T(X'_i \rightarrow X_i)} = \exp(-\beta \delta \mathcal{H}) , \quad (6.134)$$

with

$$\delta \mathcal{H} = \mathcal{H}(X'_i) - \mathcal{H}(X_i) , \quad (6.135)$$

According to Metropolis, one chooses for T the simplest form which is compatible with the detailed balance condition (6.133):

$$T(X_i \rightarrow X'_i) = \begin{cases} \exp(-\beta \delta \mathcal{H}) & \delta \mathcal{H} \geq 0 , \\ 1 & \text{else} . \end{cases} \quad (6.136)$$

The Metropolis Monte Carlo scheme is depicted in Algorithm 6.8.

Algorithm 6.8 Metropolis Algorithm (Version II)

1. Initialize the starting configuration Set $\mathbf{r} :=$ Positions of all particles in phase space
 2. Repeat steps 3. to 7. until the maximum number of trials
 3. Choose at random a particle \mathbf{r}_i with index i
 4. Change the position of this particle, i.e. $\mathbf{r}_i \rightarrow \mathbf{r}'_i$
 5. Calculate δH according to (6.135)
 6. Accept the trial move of Step 3 if $\delta H \leq 0$
 7. Generate a random number $z \in [0, 1]$
 8. Accept trial move if $\exp(-\beta \delta H) > z$ Otherwise go back to Step 2
-

An example for an application of a Monte Carlo method in polymer physics is displayed in Fig. 6.23. Here, several Pivot-moves are done with a linear chain of length $N = 100$. A Pivot move moves a randomly chosen part of a chain by a random angle in configuration space. The acceptance rate of Pivot moves (and Monte Carlo moves in general) depends on the density of the considered system. Thus, this method is best for simulating isolated chains in good solvents but is not applicable in dense melt systems.

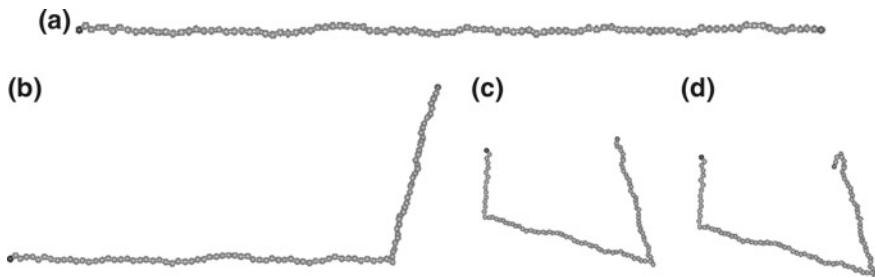


Fig. 6.23 Demonstration of Pivot moves with a linear polymer chain in solution. Pivot moves may be used in polymer simulations for a fast equilibration of long length scales in the chain. The local scales are then equilibrated using MD. The acceptance rate of Pivot moves depends on the density of the system

Problems

6.1 Three-body Problem

Starting from (6.6) using (6.8) fill in the missing algebra leading to (6.9).

6.2 Gibbs phase rule

Consider a closed system at equilibrium. How many intensive state variables (or degrees of freedom) are needed to uniquely characterize the state of this system?

Hint: Write down the first law of thermodynamics and consider the number of equations if the system has C chemical components (species of particles) and P phases. Then consider the additional conditions obtained for the intensive variables T , p and μ (chemical potential) owing to the equilibrium condition.

Computational Methods on Mesoscopic/Macroscopic Scale

7

Abstract

The meso- and macroscopic scales, due to their complexity and the difficulties to prepare systems in reproducible initial conditions is usually treated with phenomenological approaches from engineering. Yet, more and more scientists increasingly deal with model building and simulation on these scales due to the progress in computing power during the past 10 years, which for example allow routinely simulating multi-billion atom systems of the size of microns. In this chapter, we begin our discussion with a short introduction before we look at concrete examples for modeling and simulation on these scales in Sects. 6.2 and 7.2. In Sect. 7.3 we introduce the important dissipate particle dynamics method and in Sect. Ginzburg we discuss Ginzburg–Landau type modeling approaches. In Sect. 7.5 we introduce a scale-bridging example from the field of shock wave physics that shows how to perform computer simulations coupling the mesoscale with the macroscale. Section 7.6 provides further examples for scale bridging applications between MD and FEM and in Sects. 7.7–7.9 we discuss the continuum theory and elasticity theory which form the underlying mathematical basis for many computational approaches on the macroscale. In engineering, the world is usually treated as if it was a continuum, completely neglecting the discrete quantum mechanical structure and in these sections we will see by using the linear chain model of a solid state, how the transition from a description based on discrete particles to a continuum description can be done. This chapter ends with another application example from the area of crack propagation.

The purpose of this chapter is to briefly review some of the most important methods for modeling and simulating materials properties on the meso- and macroscale and to provide some applications from recent research projects. We first introduce statistical methods used on this length scale and then discuss in various sections different particle methods commonly used on the mesoscopic/macrosopic scale. Finally, in

Sect. 7.7, the fundamentals of continuum and elasticity theory are reviewed in some detail in tensorial notation. We do not discuss plasticity models based on continuum theory and it is not the aim of this chapter to provide an exhaustive overview of the large number of phenomenological, empirical and statistical models that dominate materials research on this scale. Rather, we will shortly discuss several typical methods and illustrate a few examples as typical applications. Often, these methods are combined with traditional finite element simulation schemes., i.e. they serve for taking into account parts of structural features into a conventional finite element analysis.

The derivation and an understanding of structure-property relationships at the meso- and macroscale constitute probably the most challenging subdiscipline in computational materials science. Particle simulations based on truly atomistic models are not possible on this scale and also conventional MD with classical coarse-grained models fails here, as the number of particles needed to be simulated is simply too large. Therefore, on this scale one has to further coarsen the description with particle models if one does not want to start using purely phenomenological models right away, which are based on continuum theory. In engineering science, such models, based on the conservation equations of energy-mass and momentum are abundant and the finite element method, cf. Sect. 4.2.4 on p. 207 is the de-facto standard on this scale. Also, a numerical treatment of the Navier–Stokes equation for fluid applications on Eulerian grids are well-established in the engineering community. These methods discretize the corresponding PDEs on grids and use one of the difference methods introduced in Sect. 4.2.1 for their solution, or some advanced techniques that were not discussed in Chap. 4, e.g. Godunov-Schemes [427] or coupled multiple grid methods on Cartesian or non-Cartesian grids. Modeling a material completely as a continuum is most often insufficient for determining any properties that are based

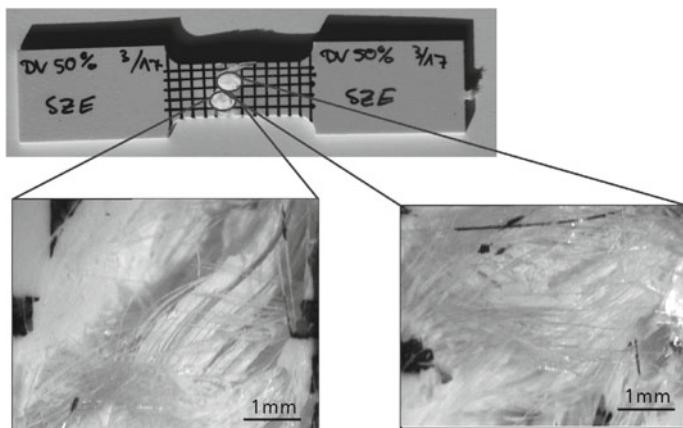
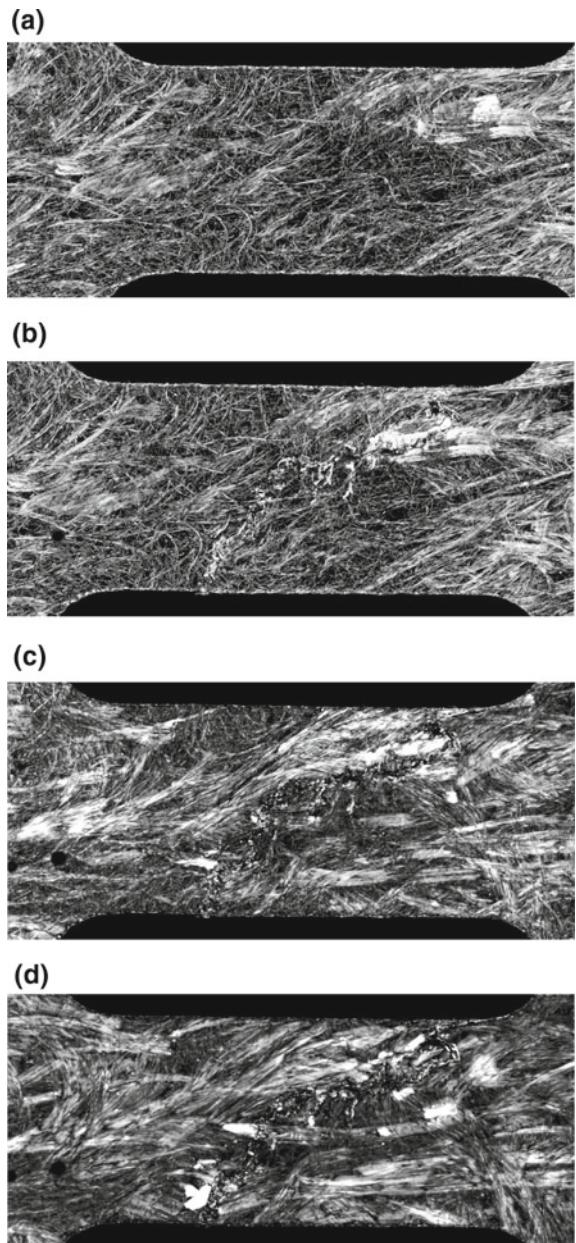


Fig. 7.1 Standard uni-axial static experiment with a fiber reinforced Sheet Moulding Compound (SMC). Two sections of the material along the developed crack upon loading are shown to a larger scale. Photos are © Martin O. Steinhauser

Fig. 7.2 **a** Ultrasound microscopic photos of a fiber-reinforced Sheet Moulding Compound (SMC) before the application of uni-axial load, 0.07 mm below the surface. **b-d** The same specimen after uni-axial loading. The consecutive photos are taken at increasing specimen depths with an interval of 0.07 mm. One can see the extension of a crack through the fiber-reinforced material. Photos are © Martin O. Steinhauser



on micro- and mesostructural features such as failure, crack initiation or crack propagation. Typical examples for such mesostructural features that have to be integrated into a simulation model, are displayed in Figs. 7.1 and 7.2. Thus, methods taking into account the explicit micro- and mesoscopic structural features, have to be used. This can be achieved by appropriately meshing the integration domain and assigning different material properties to different elements. Finite element methods are fundamentally based on a mesh-connectivity of elements, i.e. these methods are not very well suited to model failure or, e.g. very strong loading as in shock waves, which leads to very strong deformations of elements.

For the calculation of highly dynamic properties of materials, such as shock and failure, the topological connection of nodes within a mesh becomes a severe disadvantage when strong mesh distortions and corresponding discontinuities occur, as the numerical solution may become unstable. This is the reason, why during the last 25 years there has been much interest into suitable continuum alternatives to mesh-based methods on the meso-/macroscale. One such alternative is the SPH-method, which has the great advantage of working with elements that are called particles and which are not pre-fixed in a mesh but may move about freely within the simulation domain. The SPH method is based on the continuum approximation, formulated as conservation equation of mass, energy and momentum of freely moving point-elements, called particles. We do not discuss

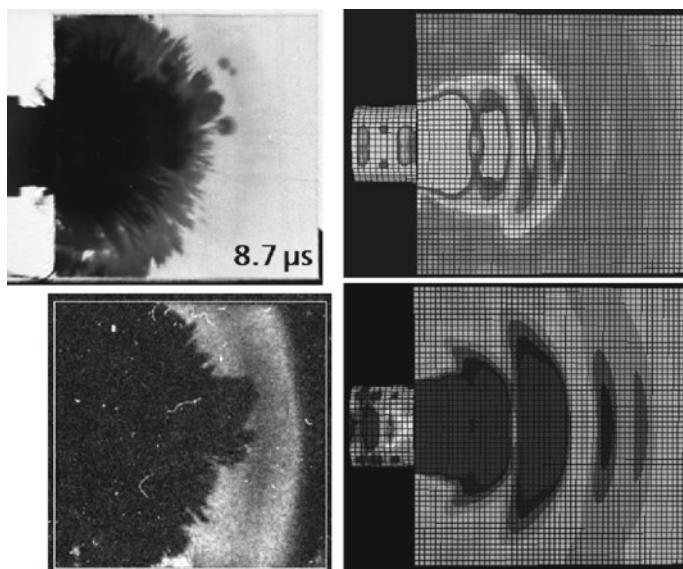


Fig. 7.3 Comparison of a high-speed impact experiment with a FEM simulation which models the whole structure as an elastic continuum. In the simulation, the von-Mises stress (*top*) and the stress in S_{11} (*bottom*) direction are displayed. While the propagation of elastic waves can be modeled sufficiently with such a simple computational approach, the numerical model is incapable of reproducing any failure in the material based on an understanding of micro- and mesostructural features. Snapshots and photos are © Martin O. Steinhauser

this method in detail here and refer the interested reader to the detailed literature [287, 291, 428, 429]. Stability problems with highly dynamic processes as described above do not arise *in principle* when using a genuine particle discretization of the integration domain, i.e. a discretization where the dynamics of the particles is governed by Newton's equation of motion.

7.1 Example: Meso- and Macroscale Shock-Wave Experiments with ceramics

In this section we provide a typical example of an experiment in material science which involves crack formation and failure; phenomena which cannot be simulated appropriately by without taking into account several microstructural features.

On the macroscopic scale, many materials (such as concrete or ceramics) are homogeneous solid states. However on the scale of a few microns to a few hundred microns many materials exhibit an inhomogeneous polycrystalline grain structure. As yet there is no proper understanding of how the micro-structure of a material specimen ultimately determines its macroscopic mechanical properties in terms of strength or resistance against high-velocity impact [430, 431]. These mechanical properties can be measured by means of high-speed impact experiments with typical high-performance ceramics (HPC)s such as Al_2O_3 and SiC in the standard set-up of the edge-on impact (EOI), where a fast impactor hits the edge of a target. A necessary prerequisite for a focused design and for an understanding of the structure-property relationship in brittle materials such as ceramics, is to understand the failure mechanisms occurring during the first 10 μs after impact. Using high-speed photographic techniques allows for visualizing dynamic fracture in brittle material during an EDI test. In an EOI test the projectile hits one edge of a specimen of typical dimensions $(100 \times 100 \times 10) \text{ mm}^3$ and wave and fracture propagation are observed e.g. by means of a Cranz-Schardin high-speed camera. The experimental set-up for two different configurations is schematically shown in Fig. 7.4.

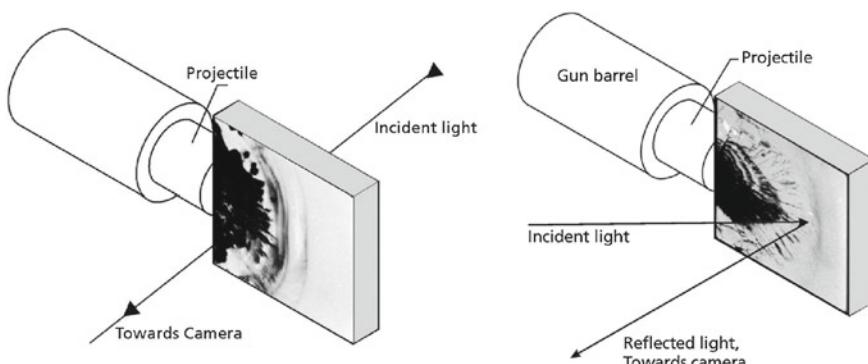


Fig. 7.4 Experimental edge-on impact standard configurations

In the velocity range up to 400 m/s a gas gun using compressed air is being employed for the acceleration of the projectiles. The ceramic specimens are placed at a distance of 1 cm in front of the muzzle of the gas gun in order to achieve reproducible impact conditions and a high precision. In this set-up the rear of the projectile is still guided by the barrel of the gun when the front hits the target. The projectile velocity is measured by means of two light barriers through openings in the barrel 15 cm behind the muzzle. For higher impact velocities of up to about 1000 m/s, a 30 mm caliber powder gun is used for the acceleration of the projectiles. In this case, due to the muzzle flash and the fumes, the specimens cannot be placed directly in front of the muzzle. When the projectile hits the edge of the ceramic tile a shock wave is generated which propagates through the material. For example, in *SiC* the longitudinal wave speed is typically in the range from 12 to 13 km/s. Hence, the shock wave will arrive at the rear edge of the specimen after 7.5–8.5 μ s. In general, in order to visualize cracks, the surface of the specimen has to be polished mirror-like, because otherwise, the intensity of the reflected light is not sufficient. With Al_2O_3 -ceramics a mirroring surface cannot be accomplished by polishing only. In this case an additional coating with a thin layer of silver or aluminum is required. At the positions where the surface is disturbed by the crack edges, no light is directly reflected to the camera. Thus, fracture appears as dark lines on the photograph, cf. Fig. 7.5. Usually Steel cylinders of 30 mm diameter and 23 mm length are employed as projectiles with impact velocities in the range from 20 to 1100 m/s. Figure 7.5 shows a selection of ten high-speed photographs of a recent EMI experiment with Al_2O_3 impacted at 140 m/s. A field of cracks can be recognized which propagates at a speed of around 4700 m/s. The area immediately in front of the impact zone appears black on the high-speed photographs, which means that no light is reflected to the camera from this area. However, the ceramic is not totally destroyed in this zone. From the analysis of many experiments it could be concluded that the black zone in front of the impacting projectile is mainly due to a deformation of the specimen which is sufficient to deflect the light.

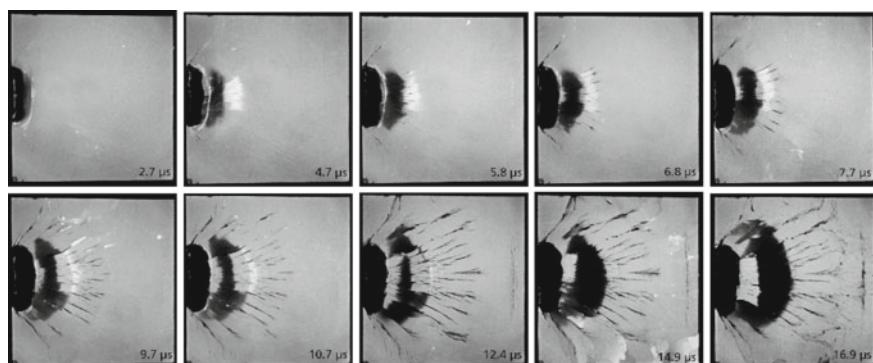


Fig. 7.5 Series of high-speed photographs of Al_2O_3 impacted at a speed of 140 m/s

7.2 Statistical Methods: Voronoi Tesselations and Power Diagrams for Modeling Microstructures of Ceramics

Generally, modern engineering materials often have constituents which differ in their individual physical properties. The constituents are often of random shapes and also of varying sizes. With ceramics, the specific shape and size of these polycrystalline grain structures is formed in a sintering process where atomic diffusion plays a dominant role. The sintering process usually results in a porous microstructure with grain sizes of several hundred micrometers, cf. Fig. 7.6. For ceramics, experimentally, there is no simple connection between grain size and hardness (a typical macroscopic property) of a polycrystalline material. With numerical investigations taking explicitly into account the details of the microstructure one can expect to achieve a considerably enhanced understanding of the structure-property relationships of such materials. Characteristic for HPCs are extremely low porosity, high purity and high hardness of the finished ceramic, cf. Fig. 7.7. An additional beneficial property of HPCs is the fact that, depending on the final grain size and other process parameters, the ceramics exhibit translucency or even complete transparency which renders these materials the prime source for future engineering applications. Typical applications of HPCs that benefit from high hardness at low weight are e.g. wear resistant brake disks, protection shields in the automobile industry or bone substitutes in medical devices. Future applications especially for translucent ceramics could be e.g. housings of high-pressure discharge lamps in medical technology. Today, one is compelled to search for the optimal micro structure for a specific application by intricate and expensive experimental “trial-and-error” studies. In order to overcome this situation by numerical simulations a detailed and realistic modeling of the available experimental structures is a basic requirement.

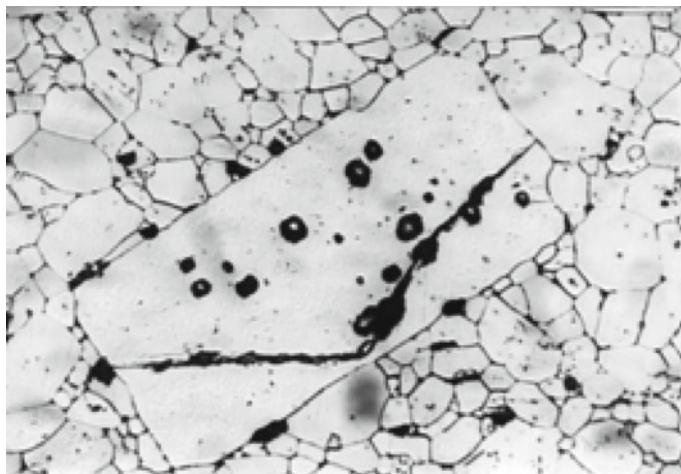


Fig. 7.6 Damaged microstructure of TiB_2 after edge-on impact at speed $v = 784$ m/s. The largest grain is about 0.1 mm in size. Photograph is © Martin O. Steinhauser

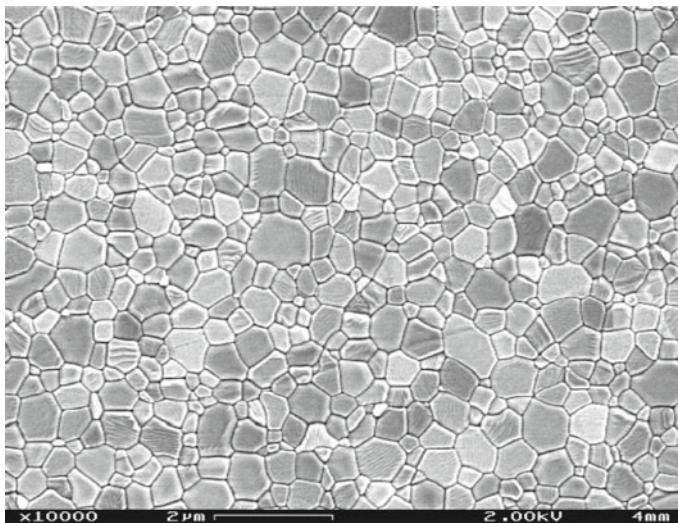


Fig. 7.7 Typical densely sintered microstructure of a HPC ceramic. Photograph is © Martin O. Steinhauser

Classical Voronoi diagrams in d -dimensional Euclidean space and their duals – the Delaunay tessellations – can be used for describing space-filling, mosaic-like structures resulting from growth processes. For a finite set of generator points $M \in \mathbf{R}^d$ the Voronoi diagram maps each point $p \in M$ onto its Voronoi region consisting of all points $x \in \mathbf{R}^d$ that are closer to p than to any other point in M . Voronoi tessellations have been used in many fields of materials science, e.g. for the description of biological tissues or polymer foams. Ghosh et al. [432] utilized Voronoi cells to obtain stereologic information for the different morphologies of grains in ceramics and Espinoza and Zavattieri [433, 434] used random Voronoi tessellations for the study of wave propagation models that describe various mechanisms of dynamic material failure at the micro scale. However, these models have major drawbacks such as limitations to two dimensions or a generic nature of the structures as they are very often not matched and validated against actual experimental data. Full-resolution 3D structures of ceramics are generally not available and only recently first reports on experimentally measured microstructures of steel were published [435, 436]. Generally, most of the micrographs obtained from densely sintered ceramics represent a tessellation of convex polyhedra in very good approximation. As a micrograph section is just *one* 2D sample of the real microstructure in 3D, the value of its explicit rendering is very questionable. Thus, in the application of geometric models one concentrates on a statistical characterization of the grains, e.g. in terms of the grain area and grain perimeter distribution. The results of this preliminary investigation are histograms of the area and the perimeter distribution as shown in Fig. 7.8. Contrary to what was claimed by Zhang et al. [437], neither the experimental area nor the perimeter distribution in Fig. 7.8 obey a Gaussian statistics.

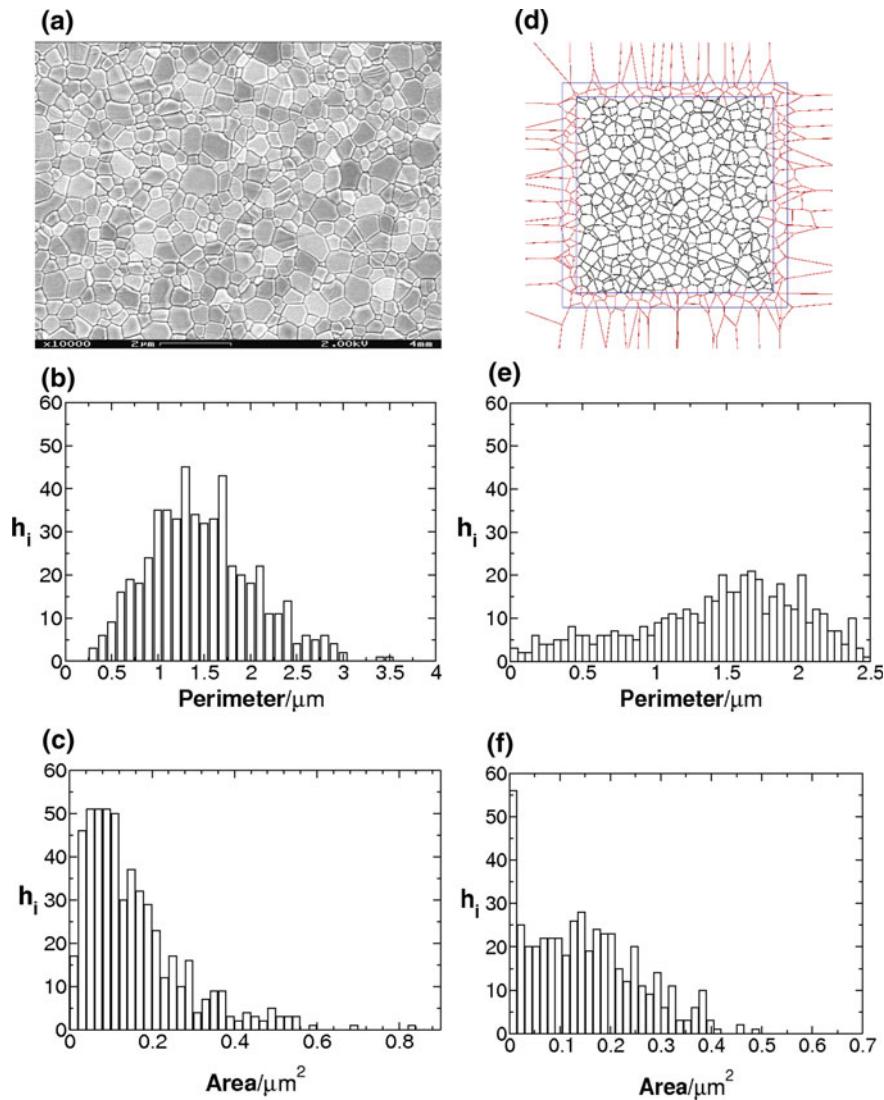


Fig. 7.8 Micrograph of a ceramic and its perimeter and area distribution (left). At the right the same (non-optimized) quantities are displayed for a 2D cut of a power diagram

7.2.1 Reverse Monte Carlo Optimization

In order to represent the polyhedral cell complex numerically one may choose a power diagram (PD) which contains the Voronoi diagram as a special case. The PD extends the Voronoi diagram by adding a scalar weight to each generator point. PDs have several advantages compared to classical Voronoi diagrams:

- Any simple polyhedral cell complex in d -space with $d \geq 3$ can be represented as a PD [438].
- Any slice of a PD is also a PD (while a slice of a Voronoi diagram is not a Voronoi diagram in general).
- The slices of a PD in d -space can be calculated directly by projecting the generator points into $(d-1)$ space.

For the calculation of power diagrams, efficient worst case optimal algorithms are available in 2D ($\mathcal{O}(N \log N)$) and in 3D $\mathcal{O}(N^2)$, e.g. the *iterative incremental topological flipping algorithm* [439]. The general flow of a computational procedure for calculating PDs in 3D is displayed in Fig. 7.9. The basic idea of a reverse MC optimization of polycrystalline structures is to compare all cells (typically at least ~ 10000) inside a cube of a given PD in 3D with the available experimental data in 2D. This comparison is performed for each coordinate axis by generating a set of parallel, equidistant 2D slices (typically about 500 slices for each one of the three coordinate directions) through the cube and perpendicular to the respective axis, cf. Fig. 7.10. For each 2D slice the grain area sizes A can be calculated and combined into one histogram. The same is done with perimeter P . For a comparison of the calculated histograms with the experimental histograms one employs the first apparent central moments of the area and perimeter distribution. A single figure of merit q of conformity is defined:

$$q = \sum_i^k \left(\frac{P_i - P_i^{exp}}{P_i^{exp}} \right)^2 + \left(\frac{A_i - A_i^{exp}}{A_i^{exp}} \right)^2. \quad (7.1)$$

All 2D slices through the cube can be calculated very efficiently in parallel using the MPI standard. A typical time scale after which a useful agreement with experimental

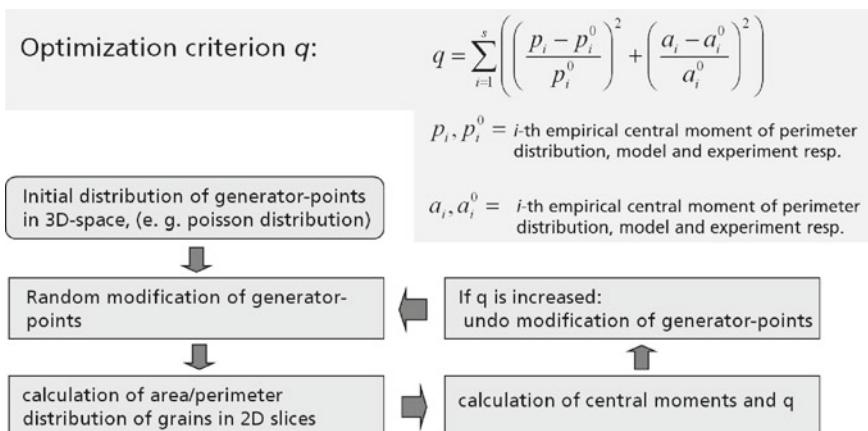


Fig. 7.9 Illustration of a reverse Monte Carlo optimization scheme for the generation of power diagrams in three dimensions based on an optimization of the measured central moments of two-dimensional micrographs of appropriate structures

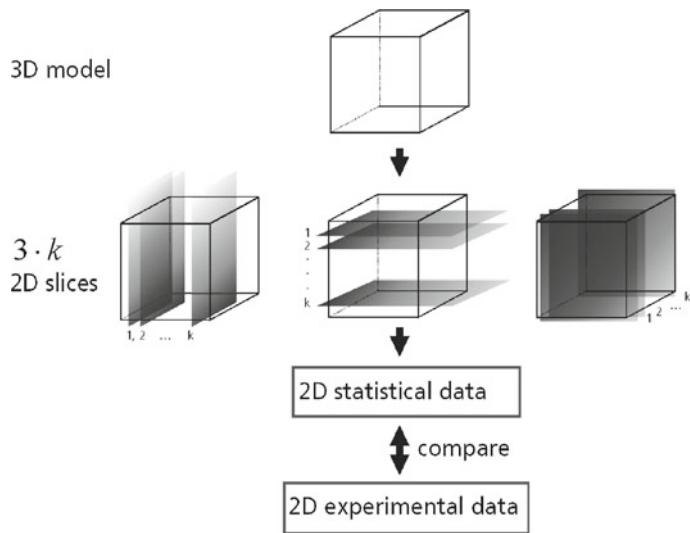


Fig. 7.10 Illustration of the cuts through the 3D power diagram generated in a cube. The obtained statistical information is compared with data obtained from etched micrographs

values is reached is about 15 days on processors on a small super computing facility for a system of $n_G = 10000$ grains and $n_S = 1500$ slices at each optimization step. The optimization algorithm generally scales as $\mathcal{O}(n_G^{4/3} \log n_G)$ under the assumption that n_S is chosen as $n_S \sim \sqrt[3]{n_G}$. Examples of the result of this statistical optimization of microstructures can be seen in Fig. 7.11

Scale Problem with FEM

Mesh-based methods such as FEM are not intrinsically scaled to any length, i.e. *in principle* one could use the function approximation of the solutions of a PDE also for the simulation of micro- and sub-microstructures. Different fundamental problems arise in this context, namely the data basis of the materials models which are used in FEM codes. These are usually derived from experiments with idealized loading situations and are then extended for other types of loading and more complex geometries. Thus, such *constitutive* models only show a very low transferability to other situations. In Fig. 7.12 the dilemma of numerical simulation on a mesoscale can be viewed: If one wants to include some mesostructural features in a mesh-based simulation scheme, then one has to make a compromise with the total domain that can be simulated. In the example of the EOI above, due to computational limitation in the number of mesh-nodes that can still be usefully calculated, one is thus restricted to the small cut-out of the material as indicated in Fig. 7.12.

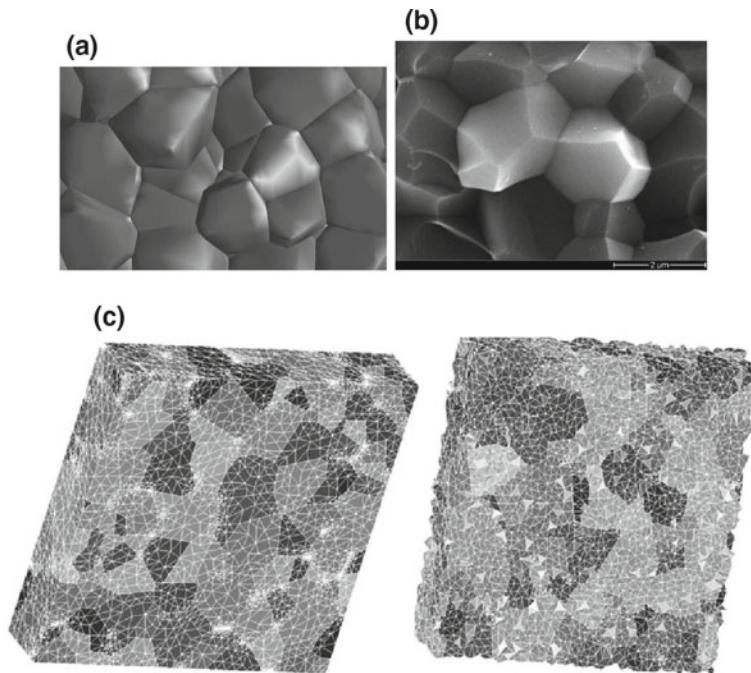


Fig. 7.11 Direct comparison of a computer generated 3-dimensional rough surface (a) with grains sticking out and a X-Ray picture (b). c Different material samples generated using the PD method described above. Note that the optimization of PDs represents the rough granular surface structure very well in comparison with X-ray pictures

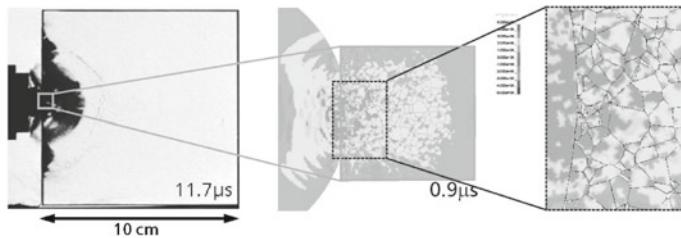


Fig. 7.12 Illustration of the scale problem when using finite element methods based on a meshed solution domain. Of the whole macroscopic ceramic tile used in the EOI experiment, only the indicated small area can be modeled using a FE mesh. Thus, including meso- and microstructural details in FE discretizations is a compromise between computational computability and the desired resolution

Mesoscale Monte Carlo Potts Models

One well-known application of the Monte Carlo method is its application to the Ising lattice model for modeling magnetic spins. This statistical model was extended to model lattice growth in materials science in 1952 [109]. The Potts model deviates from the Ising model in, that it generalizes the spin concept and makes use of a different Hamiltonian. The original spin variable, where only two states were admissible (“spin up”, “spin down”) is replaced by a generalized spin variable S_i , which can assume one of q possible discrete set of states (“ q ”-state Potts model). This spin variable accounts for the interaction between dissimilar neighbors and the Hamiltonian can be written as

$$H_{Potts} = -J_{int} \sum_{i,j} (\delta_{S_i S_j} - 1) \quad (7.2)$$

where J is the anisotropic interaction energy. The most characteristic property of this energy operator is that it defines the interaction energy between nodes with like spins to be zero and with unlike spins to one. This rule allows the identification of interfaces and the quantification of the interfacial energies for each boundary segment as a function of the abutting domains. The particular properties of the Potts model make it very suitable for describing coarsening phenomena. Materials related Potts-type simulations have been devoted to nucleation [440], recrystallization [101] or grain growth [111, 441].

7.3 Dissipative Particle Dynamics

The microscopic phenomena of soft matter systems can be treated with the standard methods on that scale, MD and MC. However, applications of soft matter systems and other materials extends to the mesoscopic/macrosopic scale. Systems of interest with this respect are, e.g. liquid crystals, colloids or biomembranes. One particularly appealing technique for simulating mesostructural features is *dissipative particle dynamics* (DPD) [442, 443], which we will shortly discuss in this section.

Dissipate Particle Dynamics is a computational method that allows the simulation of Newtonian and non-Newtonian fluids, including polymer melts and blends, on mesoscopic length and time scales. Just as MD or BD, it is a particle-based method, but in DPD, the elementary unit is not an atom or molecule but a point particle that is meant to represent a fluid element containing many molecules or molecular groups. Hence, DPD, in a sense, is a progression of MD and BD on larger scales. Although course-graining in principle can be done as well with MD and BD, DPD offers the explicit advantage of a proper description of hydrodynamic modes significant in the physical approach toward a system’s equilibrium. This is achieved in DPD by implementing a thermostat in terms of pairwise conservative (C), random (R), and dissipative (D) forces in a way that the total momentum of the system is conserved:

$$\mathbf{F}_{ij}^{DPD} = \mathbf{F}_{ij}^C + \mathbf{F}_{ij}^D + \mathbf{F}_{ij}^R, \quad (7.3)$$

where the different forces are:

$$\text{Conservative Force } \mathbf{F}_{ij}^C = \Pi_0 \omega_C(r_{ij}) \mathbf{e}_{ij} , \quad (7.4a)$$

$$\text{Dissipative Force } \mathbf{F}_{ij}^D = -\gamma \omega_D(r_{ij}) (\mathbf{e}_{ij} \mathbf{p}_{ij}) , \quad (7.4b)$$

$$\text{Random Force } \mathbf{F}_{ij}^R = \sigma \xi_{ij} \omega_R(r_{ij}) \mathbf{e}_{ij} . \quad (7.4c)$$

Here, $\mathbf{r}_{ij} = (\mathbf{r}_i - \mathbf{r}_j)$, $|\mathbf{r}_{ij}|$, $\mathbf{e}_{ij} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$, $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$. Π_0 is a constant related to fluid compressibility, γ is a friction coefficient, σ is a noise amplitude and ξ_{ij} is a symmetric Gaussian noise term with zero mean and unit variance and different ξ_{ij} are independent for different pairs of particles and different times, i.e.

$$\langle \xi_{ij}(t) \xi_{kl}(t') \rangle = (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \delta(t - t') . \quad (7.5)$$

The momentum conservation requires $\xi_{ij} = \xi_{ji}$

The conservative force term \mathbf{F}_{ij}^C is not further specified by the DPD formulation and is independent of the other two forces. Hence, it can be chosen to include any forces that are appropriate for a given system, e.g. van der Waals interactions. The specific form of all interaction potentials is governed by the expression for the weight functions ω_C , ω_D and ω_R . Usually, the conservative force is chosen to be a soft repulsive LJ-type potential. The random and dissipative forces are coupled through a fluctuation-dissipation relation which is due to the requirement that in thermodynamic equilibrium the system must have a canonical distribution. This requirement sets two conditions linking the random and dissipative forces:

$$\omega^D(r_{ij}) = \left(\omega^R(r_{ij}) \right) , \quad (7.6a)$$

$$\sigma^2 = 2\gamma k_B T . \quad (7.6b)$$

Equation (7.6) couples the weight functions and (7.6) couples the strength of the random and dissipative forces. Beyond a certain cut-off separation r_c , the weight functions and thus all forces are zero. The weight function are normalized such that

$$\int_V \omega(r) d\mathbf{r} = \frac{V}{N} , \quad (7.7)$$

where V is the volume of the simulation box and N is the number of considered particles. The instantaneous temperature is obtained just as in MD from the equipartition theorem

$$3k_B T = \left\langle \frac{\mathbf{p}_i^2}{m_i} \right\rangle . \quad (7.8)$$

The DPD system can be integrated using a velocity-Verlet integration scheme, cf. (4.7).

$$d\mathbf{r}_i = \mathbf{v}_i dt \quad (7.9a)$$

$$d\mathbf{v}_i = \frac{1}{m} (\mathbf{F}_i^C dt + \mathbf{F}_i^D dt + \mathbf{F}_i^R \sqrt{dt}) . \quad (7.9b)$$

The velocity increment due to the random force is \sqrt{dt} instead of dt . This arises from the discretization of a stochastic differential equation, which models a Wiener process, i.e. the intrinsic thermal noise in a system, cf. [444]. For several recent applications of this method to the simulation of flow around spheres or polymer simulations in shear flow, see e.g. [445, 446].

7.4 Ginzburg–Landau/Cahn–Hilliard Field Theoretic Mesoscale Simulation Method

Predicting the equilibrium and non-equilibrium phase transformation phenomena on the mesoscopic scale, particularly of the liquid-solid and solid-liquid types, is among the most challenging problems in materials science. A detailed knowledge of the morphological and structural characteristics of materials on the mesoscale which arise from such transitions, form the basis of many advanced models for the description of mesoscopic structures. The Ginzburg–Landau model is a kind of a “metamodel” to formulate thermodynamic potentials near critical points and was originally suggested for magnetic systems by Landau in 1937. The Landau–Ginzburg model elucidates the behavior of systems near critical points and shows how this behavior depends on the dimensionality d of the underlying lattice, and on the dimensionality of the order parameter D . Indeed, the model tries to eliminate the lattice altogether and assumes a continuous order parameter D in a field $\Phi(x)$. In essence, Ginzburg–Landau theory relates the time evolution of one or several spatio-temporal order parameters to the derivatives of a free energy that is a functional of these order parameters. While the the thermodynamics of phase transition phenomena only prescribe the general direction of microstrucure evolution, the kinetics of the various lattice defects determines the actual microstructural path. As a result, it is not those microstructures that are close to equilibrium, which provide the most promising advantageous material properties, but those, which are in a non-equilibrium state. In this sense, microstructure optimization on the meso/macroscale level in materials science focuses on the prediction of kinetics. With the advent of powerful computer in recent time, it has become possible to render some fundamental state variable approaches, such as the Ginzburg–Landau theory, into numerical schemes which treat solid- and liquid phase transformation kinetics in a discretized fashion, considering the spacial and temporal changes in the chemical and structural fields. These field variables are often called *phase field variables*. The Ginzburg–Landau theory of critical phenomena, originally formulated for calculating magnetic second-order phase-transitions, has been adapted to metallurgical variants, which are formulated e.g. in the phase field models by Cahn–Hilliard (CH) in 1958 [106], that have been known for a long time, are nowadays commonly used in computational materials science. The CH phase-field model is based on the non-linear diffusion equation for a binary mixture which begins with a continuity equation for each component i of the mixture that relates the spatio-temporal concentration or density $\Phi_i(\mathbf{r}, t)$ of that component to the mass current $\mathbf{j}_i(\mathbf{r}, t)$, and expresses the conservation of mass in the system. In general, for an n -component system, an equation of motion can be written for each component as

$$\frac{\partial \Phi_i(\mathbf{r}, t)}{\partial t} = -\nabla \cdot \mathbf{j}_i(\mathbf{r}, t) . \quad (7.10)$$

The mass current of component i is related to the chemical potential μ_j through

$$\mathbf{j}_i(\mathbf{r}, t) = - \sum_j^{n-1} M_{ij} \nabla \mu_j + \mathbf{j}_T(\mathbf{r}, t) , \quad (7.11)$$

where M_{ij} is the mobility of component i due to j , and $j_T(\mathbf{r}, t)$ is the mass current arising from thermal noise. The chemical potential μ_j is related thermodynamically to the free energy functional

$$F[\Phi_i(\mathbf{r}, t), \Phi_j(\mathbf{r}, t), \dots, \Phi_{n-1}(\mathbf{r}, t)] \quad (7.12)$$

by

$$\mu_j = \frac{\delta F}{\delta \phi_j} \quad (7.13)$$

and the constraints

$$\Phi_i + \Phi_j + \Phi_k + \dots + \Phi_n = 1. \quad (7.14)$$

For an incompressible binary blend with particle types A and B , there results only one equation of motion:

$$\frac{F[\Phi(\mathbf{r})]}{k_B T} = \int d\mathbf{r} \left[\frac{f_{FH}[\Phi(\mathbf{r})]}{k_B T} + \kappa(\Phi) |\Phi(\mathbf{r})|^2 \right], \quad (7.15)$$

where $f_{FH}(\Phi)$ is the Flory–Huggins (FH) free energy of mixing:

$$\frac{f_{FH}(\Phi(\mathbf{r}))}{k_B T} = \frac{\Phi}{N_A} \ln \Phi + \frac{1-\Phi}{N_B} \ln(1-\Phi) + \chi \Phi(1-\Phi), \quad (7.16)$$

where χ is the enthalpic interaction parameter between the two components. For small molecules ($N = 1$) the free energy of mixing may be approximated by a Taylor expansion around the critical point by the Landau form

$$\frac{f_{FH}(\Psi)}{k_B T} = r\Psi^2 + g\Psi^4. \quad (7.17)$$

The Ginzburg–Landau free energy functional is then the CH functional with the Landau form of the bulk free energy

$$\frac{f_{GL}[\Psi(\mathbf{r})]}{k_B T} = \int d\mathbf{r} [r\Psi^2 + g\Psi^4 + \kappa|\nabla\Psi(\mathbf{r})|^2]. \quad (7.18)$$

This method has been applied in materials science for simulating spinodal decomposition or grain growth of microstructures, see e.g. [447].

7.5 Bridging Scales: Soft Particle Discrete Elements for Shock Wave Applications

This section presents an application of the *Discrete Element Method* (DEM) which is very commonly used on mesoscopic length scales. It is similar to the DPD method in that it uses “super-particles” which do not represent atoms or molecules but it does not necessarily use a special form of the equations of motion as in (7.9).

The original idea to use discrete elements goes back to Cundall and Strack [415] who introduced this modeling for geotechnical problems. The core element of the DEM is the subdivision of the integration domain into discrete elements which may

or may not be of a spherical shape. In the first application of this method, based on a simple linear force-displacement law, Cundall and Strack could derive the inner stresses and forces of a cohesionless agglomeration of disks, respectively spheres. In a later work [448] they could show the existence of shear bands in arrangements of polydisperse disks. Later applications of this method differ from each other mainly in the kind of considered forces (normal, shear or frictional forces) and the complexity of the employed force models. Thus, many models for modeling granular matter using different particle discretizations and potentials exist in the literature [449–453]. Often, instead of using spheres which are computationally efficient, one uses convex polyhedra as elements in an attempt to model as closely as possible the actual granular structure of materials as seen in micrographs, cf. the discussion in Sect. 7.2.

We present here a simple method of simulating shock wave propagation and failure in materials which are modeled as solid states using soft particles that approximate the polycrystalline microstructure. The question whether a material under load displays a ductile, metal-like behavior or ultimately breaks irreversibly, in principle depends on the atomic crystal structure and on the propagation of defects in the material. From a principal point of view, broken atomic bonds on the nanoscale which eventually can be seen as macroscopic cracks in the material and dislocations on the nano-microscale are the two major defects determining mechanical properties. Many-particle molecular dynamics (MD) simulations taking into account the degrees of freedom of several billion atoms are feasible nowadays [454, 455], which have led to a basic understanding of the processes that govern failure and crack behavior, such as the dynamics of dislocations [456, 457], the stability of crack propagation [458–460], the limiting speed of crack propagation [42, 461, 462], or the universal features of energy dissipation in fracture [463]. However, these investigations are still limited to the microscopic scale and follow the dynamics of the considered systems only on time scales of nanoseconds,¹ even on the largest supercomputers of present time [46]. Other theoretical models were developed principally based on the Newtonian equations of motion, e.g. Holian and Lomdahl developed a *mesodynamical model* for the description of metals which extends the classic equations of motion by a damping term that accounts for energy dissipation. However, this model was not applied to failure and crack propagation [97]. For simulations of macroscopic material behavior, as seen in the EOI shock wave experiments described in Sect. 7.1, techniques based on a continuum approximation, such as the finite element (FE) method or smooth particle hydrodynamics (SPH) are almost exclusively used. In a continuum approach the considered grain structure of the material is typically subdivided into smaller (finite) elements, e.g. triangles or tetrahedra in three dimensions. Upon failure, the elements are separated according to some predefined failure modes, often including a heuristic Weibull distribution [464, 465] which is imposed upon the system. In FE simulations of polycrystalline materials, in order to be able to simulate a sufficient number of grains, often only two dimensions are considered.

¹Only the spatial domain can be scaled by being decomposed and distributed to a large number of processors – time however, cannot.

Results using these methods are often strongly influenced by mesh resolution and mesh quality [466, 467].

Using Occam's Razor one can use a very simple model for simulating failure of brittle materials which is based on simple Newtonian particle dynamics. The model is based on the following unbonded LJ-type pair potential which models overlapping discrete elements, cf. Fig. 7.13.

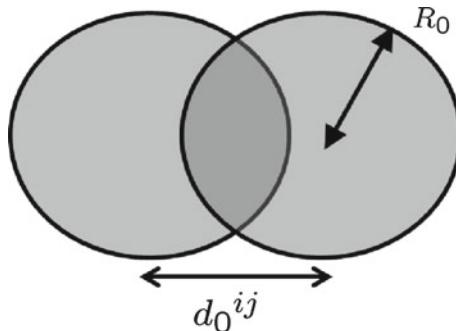
$$\Phi_{rep}^{ij}(d^{ij}) = \begin{cases} \gamma R_0^3 \left(\left(\frac{d_0^{ij}}{d^{ij}} \right)^{12} - 2 \left(\frac{d_0^{ij}}{d^{ij}} \right)^6 + 1 \right) & : 0 < d^{ij} < d_0^{ij} \\ 0 & : d^{ij} \geq d_0^{ij} \end{cases}. \quad (7.19)$$

γ scales the energy density and the prefactor R_0^3 ensures the correct scaling behavior of the calculated total stress $\Sigma_{ij}\sigma^{ij} = \Sigma_{ij}F^{ij}/A$ which is independent of N . Φ_{rep}^{ij} acts on every pair of discs $\{i,j\}$ only when the degree of overlap d^{ij} decreases during the simulation compared to the initial situation, i.e. only if $d^{ij} < d_0^{ij}$: The degree of initial particle overlap in this simple model is a measure for the force that is needed to detach particles from each other. The force is imposed on the particles by elastic springs. Additionally, a failure threshold for both, extension and compression of the springs that connect the initial particle network is introduced. R_0 is the constant radius of the particles, $d^{ij} = d^{ij}(t)$ is the instantaneous mutual distance of each interacting pair $\{i,j\}$ of particles, and $d_0^{ij} = d^{ij}(t=0)$ is the initial separation which the pair $\{i,j\}$ had in the starting configuration. Every single pair $\{i,j\}$ of overlapping beads is associated with a different initial separation d_0^{ij} and hence with a different force. The initial distances d_0^{ij} of each individual bead pair $\{i,j\}$ are chosen such that the body is force-free at the start of the simulation. The cohesive potential, exhibited under tensile load reads:

$$\Phi_{coh}^{ij}(d^{ij}) = \lambda R_0 (d^{ij} - d_0^{ij})^2, \quad d^{ij} > 0. \quad (7.20)$$

In this equation, λ (which has dimension [energy/length]) determines the strength of the potential and the prefactor R_0 again ensures proper scaling behavior of the material response. Hence, systems with all parameters kept constant, but only N varied, lead to the same slope (Young's modulus) in a stress-strain diagram. The

Fig. 7.13 Illustration of the model solid state consisting of overlapping soft spherical discrete elements



total potential of the model reads:

$$\Phi_{tot} = \Sigma_{ij} \left(\Phi_{rep}^{ij} + \Phi_{coh}^{ij} \right). \quad (7.21)$$

The repulsive part of Φ_{tot} acts only on disc pairs that are closer together than their mutual distance d_0^{ij} in the starting configuration, whereas the harmonic potential Φ_{coh} either acts repulsively or cohesively, depending on the actual distance d^{ij} .

Finally, failure is included by two breaking thresholds for the springs with respect to compressive and to tensile failure, respectively. If either of these thresholds is exceeded, the respective spring is considered to be broken and is removed from the system. A tensile failure criterium is reached when the overlap between two particles vanishes, i.e. when the distance d^{ij} of disc centers exceeds the sum of their radii:

$$d^{ij} > (2R_0). \quad (7.22)$$

Failure under pressure load occurs in our model when the actual mutual particle distance is less by a factor α (with $0 < \alpha < 1$) than the initial mutual particle distance, i.e. when

$$d^{ij} < \alpha \cdot d_0^{ij}. \quad (7.23)$$

Note that particle pairs without a spring after pressure or tensile failure still interact via the repulsive potential and cannot move through each other.

This simple model only has three free parameters, γ and λ for the interaction potentials and α for failure. These model parameters can be adjusted to mimic the behavior of specific materials. In particular, the phenomenological stress strain curve can be obtained for specific ceramics [468, 426]. Note that the maximum tensile strength does not appear as a parameter – it follows from the initial particle configuration. Decisive for this quantity is the average value of the pair distribution function $\langle d_0^{ij} \rangle$. The closer $\langle d_0^{ij} \rangle$ is to $2R_0$, the less is the maximum tensile strength. The distribution of initial particle distances determines the system's stability upon load, as well as its failure behavior and the initiation of cracks.

Figure 7.14 exhibits some examples for particle configurations for different Θ -values where Θ is a density parameter for the system. For values of Θ smaller than Θ_1 , one obtains a disordered distribution with coordination numbers ranging from zero to three (Fig. 7.14a). At the critical value $\Theta = \Theta_1$, the particles assume an ordered structure with a predominant coordination number of six. This corresponds to a hexagonal packing of disks in Fig. 7.14b and c. For Θ -values that increase well beyond Θ_2 , the structure of the system changes from hexagonal to a mainly quadratic arrangement, where main coordination number is eight, cf. Fig. 7.14d.

The factors R_0 and R_0^3 which are used in the expressions of the potentials (7.19) and (7.20) respectively are chosen such, that the model properties are independent of the system size N . This property arises from the consideration that the physical properties of a system should be independent of the arbitrary resolution with which the system is described. To show this, we consider as a solid state model M_A a square box with fixed edge length, filled with n_A mono-disperse beads that are allowed to overlap. Next, we want to reduce the number of beads in such a way that the model

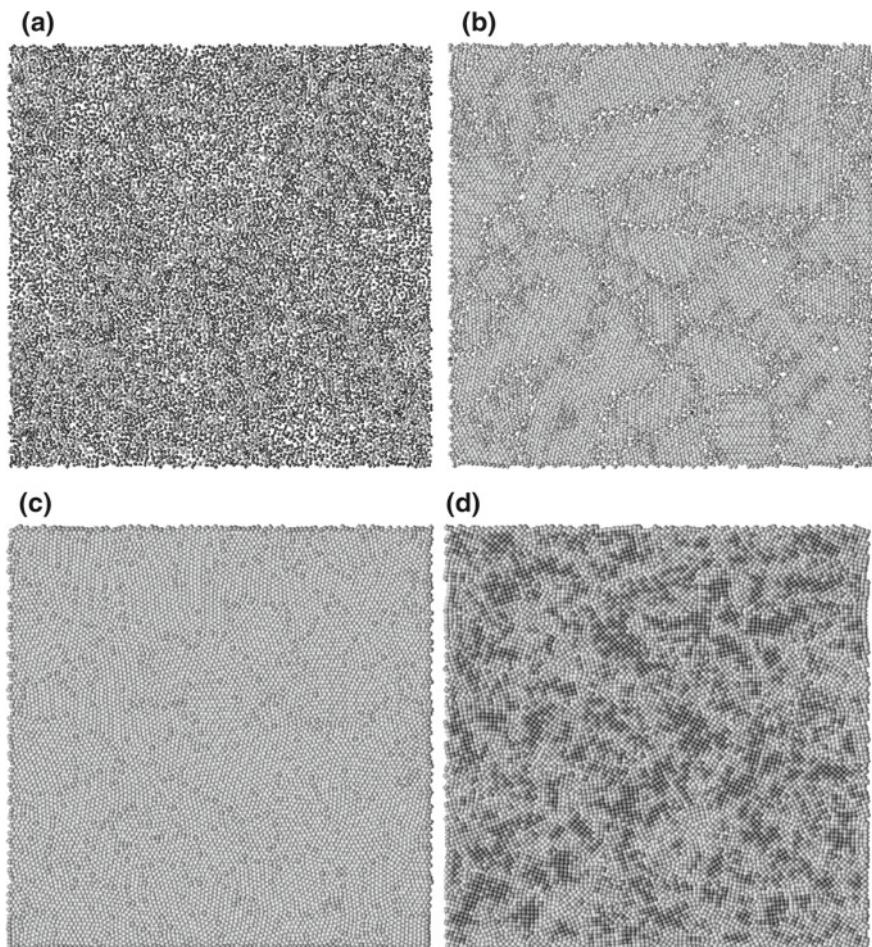


Fig. 7.14 Sample initial configurations with $N = 10^4$ discs and different Θ -values after the warm-up of the system. The color code displays the coordination number. Blue (no contact with particles within a sphere of radius $2R_0$), green (4 next neighbors), yellow (6) and red (8). **a** $\Theta = 0.7$. System loose packing and only few overlapping particles, i.e. coordination numbers within the range of 0–3. **b** $\Theta = 0.9$. Transition to a predominantly hexagonal packing with 4–6 next neighbors. **c** $\Theta = 1.3$. Dense system with predominantly hexagonal packing with coordination number 6. **d** $\Theta = 1.7$. Transition from hexagonal to predominantly quadratic packing of overlapping discs with coordination number 8

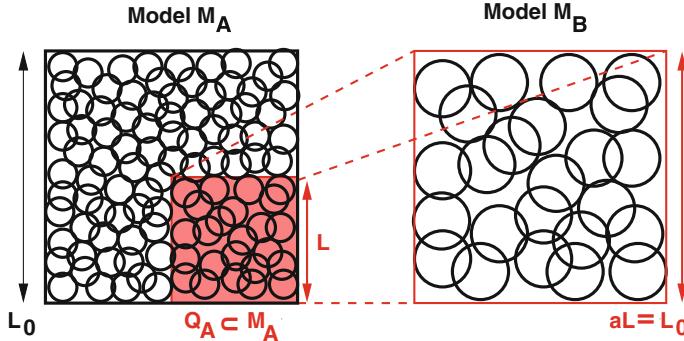


Fig. 7.15 Sketch of the proposed model system and its scaling properties. For sake of simplicity, only the two-dimensional case is shown. *Left* The original system M_A with edge length L_0 , constant particle radii R_0 and the sub-system $Q_A \subset M_A$ (shaded area) with edge length L . *Right* The scaled system M_B with particle radii $S_0 = aR_0$ and edge length $L_0 = aL$. Both models represent the same macroscopic solid

keeps its spacial dimensions. For this purpose, we chose a square sub-box $Q_A \subset M_A$ with $n_B < n_A$ beads and scale this box with a scaling factor a , such that it has the same dimension as M_A . This new model is called model M_B , cf. the sketch in Fig. 7.15. We note that both models M_A and M_B represent the same macroscopic solid.

We now require that the energy Φ which is stored in the bondings of the beads is the same in both models, i.e. $\Phi(M_A) = \Phi(M_B)$ and that the energy is distributed homogeneously in system M_A . From the last condition it follows that the total energy density in M_A and Q_A is identical, i.e. it follows that:

$$\begin{aligned} \Phi_{tot}(M_A) / Vol(M_A) &= \Phi_{tot}(Q_A) / Vol(Q_A) \\ &= \Phi_{tot}(M_B) / Vol(M_B). \end{aligned} \quad (7.24)$$

Here, the volume scales with a^3 , and thus $Vol(M_B) = Vol(Q_A) \cdot a^3$. Using (7.19) and (7.20) we have as a general form for the total potential energy Φ_{tot} in System Q_A :

$$\Phi_{tot}(Q_A) = \frac{1}{2} \sum_{i,j} \kappa \left(d^{ij} - d_0^{ij} \right)^2 + \epsilon \left[\left(\frac{d^{ij}}{d_0^{ij}} \right)^{-6} - 1 \right]^2 \cdot \Theta_H \left(d_0^{ij} - d^{ij} \right). \quad (7.25)$$

where $\Theta_H(x)$ is the Heaviside step function and the parameters κ and ϵ have yet to be fixed. We now require (7.24) to hold. As we show in the following, it is a simple matter to verify that the choice $\kappa = \lambda R_0$ and $\epsilon = \gamma R_0^{-3}$ (with γ and λ independent of R_0) render the density $\Phi_{tot}(Q_A) / Vol(Q_A)$ invariant under a change of scale, and hence independent of N . In this way, we recover the expressions for the potentials given in (7.19) and (7.20), respectively. Considering the situation in the up- (or down-) scaled

model M_B . Setting now $s^{ij} = a \cdot d^{ij}$, $s_0^{ij} = a \cdot d_0^{ij}$ and $S_0 = a \cdot R_0$, one has

$$\begin{aligned}\phi_{tot}(M_B) &= \frac{1}{2} \sum_{i,j} \left\{ \lambda S_0 \left(s^{ij} - s_0^{ij} \right)^2 \right. \\ &\quad \left. + \gamma S_0^3 \left[\left(\frac{s^{ij}}{s_0^{ij}} \right)^{-6} - 1 \right]^2 \cdot \Theta_H \left(s_0^{ij} - s^{ij} \right) \right\} \\ &= \frac{1}{2} \sum_{i,j} \left\{ \lambda a R_0 \cdot a^2 \left(d^{ij} - d_0^{ij} \right)^2 \right. \\ &\quad \left. + \gamma a^3 R_0^3 \left[\left(\frac{d^{ij}}{d_0^{ij}} \right)^{-6} - 1 \right]^2 \cdot \Theta_H \left(d_0^{ij} - d^{ij} \right) \right\} \\ &= a^3 \cdot \phi_{tot}(Q_A),\end{aligned}\tag{7.26}$$

and from $Vol(M_B) = a^3 Vol(Q_A)$ (7.24) follows.

Considering the corresponding applied forces F_{tot} in the up- or downscaled model M_B , we obtain the following result:

$$\begin{aligned}F_{tot}(M_B) &= -\nabla \Phi_{tot}(M_B) \\ &= \sum_{ij} \lambda S_0 \left(s^{ij} - s_0^{ij} \right) - 6\gamma S_0^3 \left[\left(s^{ij} - s_0^{ij} \right)^{-6} - 1 \right] \\ &\quad \left(\frac{s^{ij}}{s_0^{ij}} \right)^{-7} \left(\frac{1}{s_0^{ij}} \right) \cdot \Theta_H \left(s_0^{ij} - s^{ij} \right) \\ &= \sum_{ij} \lambda a R_0 \left(ad^{ij} - ad_0^{ij} \right) - 6\gamma a^3 R_0^3 \left[\left(\frac{d^{ij}}{d_0^{ij}} \right)^{-6} - 1 \right] \\ &\quad \cdot \left(\frac{d^{ij}}{d_0^{ij}} \right)^{-7} \cdot \left(\frac{1}{ad_0^{ij}} \right) \cdot \Theta_H \left(d_0^{ij} - d^{ij} \right) \\ &= a^2 \cdot F_{tot}(Q_A).\end{aligned}\tag{7.27}$$

The area of Q_A and M_B upon which the force in (7.27) is applied, is $Area(Q_A) \cdot a^2 = Area(M_B)$. Hence, one obtains for the stresses σ :

$$\begin{aligned}\sigma(M_A) &= \sigma(Q_A) = F_{tot}(Q_A) / Area(Q_A) \\ &= F_{tot}(M_B) / Area(M_B) \\ &= \sigma(M_B)\end{aligned}\tag{7.28}$$

As a result of the above considerations, the DEM model for a brittle solid state can be scaled without having to calibrate it anew for different system sizes N .

Results of applying this model to shock wave simulations, are displayed in the picture series of Fig. 7.16, which shows the dynamics of crack propagation in the material.

As all information about the particles in the system are available at all times, one can study in detail the initiation of meso-cracks and their propagation through the material until catastrophic failure. Upon increasing external load particle pair bonds start to fail. This leads to crack tips where local tensions accumulate until further breaking of bonds occurs. Note that with the proposed model it is possible to capture particularly well the catastrophic behavior of brittle materials. At first, many micro-cracks are initiated all over the material by failing particle pair bonds, cf. Fig. 7.16a. These meso-cracks lead to local accumulations of stresses in the material until a macroscopic crack originates (Fig. 7.16b) which quickly leads to a complete rupture of the macroscopic sample (Fig. 7.16c and d).

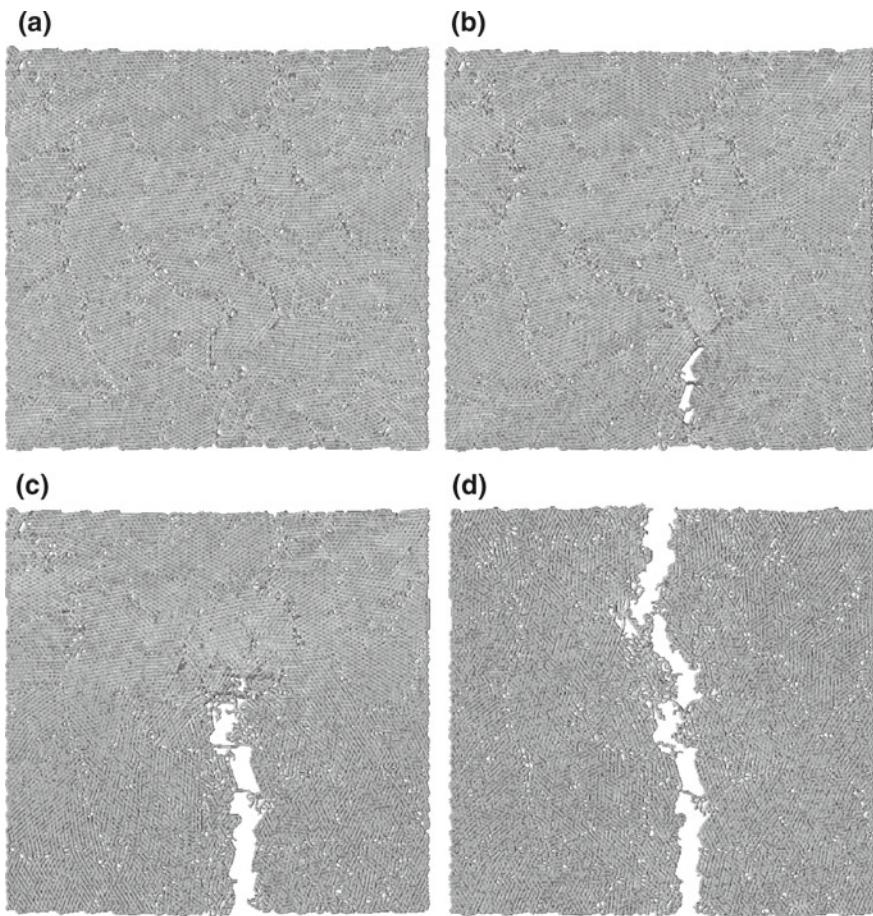


Fig. 7.16 Crack initiation and propagation in the material upon uni-axial tensile load. Color code is: *Light colors*: forcefree bonds. *Dark*: tension. **a** After applying a force at both sides of the specimen, local tensions occur. **b** Initiation of a crack tip with local tensions concentrated around this tip. **c** Propagation of this crack through the material. **d** Failure. The simulated material sample is of the same macroscopic size as in the experiment of Sect. 7.1

One advantage of the DEM mesoscale approach is, that one is able to simulate many systems with the same set of parameters (and hence the same macroscopic properties) but statistically varying micro-structure (i.e. initial arrangement of particles). Hence, by way of performing many simulations, one is able to obtain a good statistics for corresponding observables. The results of uni-axial shear simulations are presented in Fig. 7.17a–c. Starting from the initially unloaded state, we shift the top and bottom layer of discs in opposite directions. At first, in Fig. 7.17a, the tension accumulate in the whole system. Then, as can be seen from Fig. 7.17b and c, shear bands form and stresses accumulate, until failure due to a series of broken bonds occurs. The same detail of information is, if at all, very hard to obtain by extensive polarization pictures of a real system under load. In a simulation, the complete detailed information on local tensions and compressions is readily available. In addition, the influence of specific parameters on the model behavior can be studied in detail in the simulations, as these can be repeated ad libitum.

Irreversible deformations of the particles such as plasticity or heat are not considered in this simple form of the DEM model, i.e. energy is only removed from the system by broken bonds. Therefore, the development of damage in the material is slightly overestimated compared to the experiment. The advantage of DEM simulations for investigating impact phenomena is evident from the simulation snapshots in Fig. 7.18 obtained from non-equilibrium MD simulations. One can study in great detail the physics of shock waves traversing the material and easily identify strained or compressed regions by plotting the potential energies of the individual pair bonds. Also the failure in the material can be conveniently visualized by plotting only the failed bonds as a function of time, cf. the bottom series (g, h, i) of snapshots in Fig. 7.18. These figures display the failed area in the material over time. The impactor hits the target at the left edge. This leads to a local compression of the discs in the impact area. The top series of snapshots (a, b, c) in Fig. 7.18 shows the propagation of a shock wave through the material. The shape and also the distance traveled by this shock front correspond very well to the photographs (d, e, f) in the

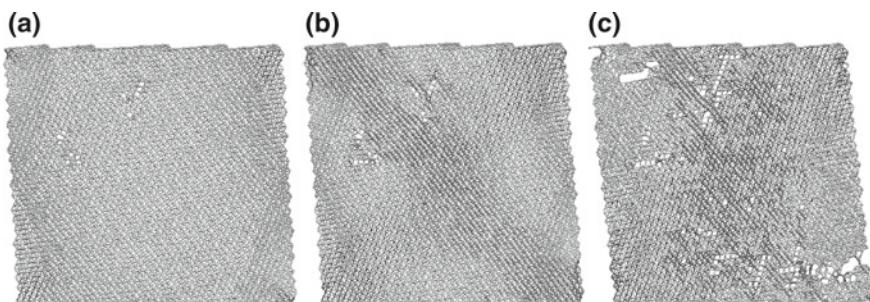


Fig. 7.17 Results of a shear simulation, where the *top* layer and the *bottom* layer of particles move in opposite directions. Color code as in Fig. 7.16 **a** Occurrence of local tensions. **b** Shear bands have occurred, spreading through the whole system. **c** Shear bands have moved with the external load and failure occurs

middle of Fig. 7.18. These snapshots were taken at comparable times after the impact had occurred in the experiment and in the simulation, respectively. After a reflection of the shock wave at the free end of the material sample, and its propagation back into the material, the energy stored in the shock wave front finally disperses in the material without causing further damage, i.e. further broken bonds. Generally speaking, impact simulations using MD are numerically very efficient and scale linearly

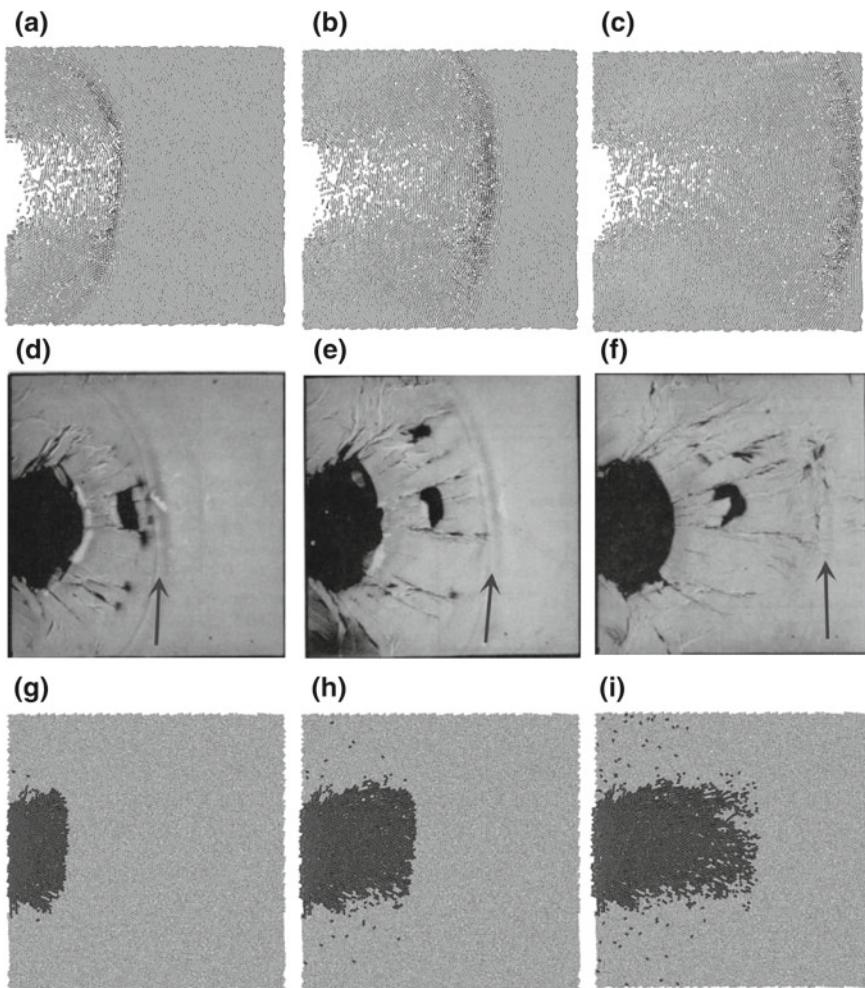


Fig. 7.18 **a–c** Simulation results of an EOI simulation at $V = 150 \text{ m/s}$ with SiC . The material is hit at the left edge. A pressure wave (color-coded in blue) propagates through the system. The time interval between the snapshots from *left to right* is $2\mu\text{s}$. **d–f** The same experiment with a real SiC specimen. The time interval between the photographs is comparable with the ones in the *top* row. Time interval after impact from *left to right* 9.5 , 11.5 and $13.5\mu\text{s}$. **g–i** The same computer simulation, this time displaying the occurring damage in the material with respect to broken bonds

with N . The generation and propagation of shock waves leads to a non-equilibrium evolution of the considered system.

7.6 Bridging Scales: Energetic Links between MD and FEM

On the mesoscopic length scale, many systems exhibit complex properties which can neither be subjected to a rigorous ab initio approach, nor can they be appropriately described by a phenomenological theory. Typical examples are soft matter systems, such as the mechanical properties of DNA, or RNA, the mechanics of epitaxial thin films, the initiation of shear bands in materials upon load or crack initiation in ceramics, or glass, and the motion of a dislocation in a ductile material. Thus, it is a logical step to try to couple the simulations on different scales concurrently, either by performing simulations on different scales separately and then combining the results and using them as input for a simulation on a different scale, or as a combined simulation run within one code. One pioneering multi-scale approach was the work done by Abraham et al. [42] who tried to link dynamically tight-binding, molecular dynamics, and finite elements together in a unified approach called *Macroscopic, Atomic, Ab initio Dynamics* (MAAD). Dynamic linking here means, that all three simulations run at the same time, and dynamically transmit necessary information to and receive necessary information from the other simulations. In such a method, the FE mesh is downgraded to the order of the atomic scale, at which point, the dynamics is governed by MD. At crack tips, which are dynamically most interesting, TB is used to correctly simulate the dynamics of the braking of atomic bonds. All interactions are governed by conservation of energy in the system and the Hamiltonian reads:

$$\mathcal{H}_{\text{tot}} = \mathcal{H}_{\text{TB}} + \mathcal{H}_{\text{TB/MD}} + \mathcal{H}_{\text{MD}} + \mathcal{H}_{\text{MD/FEM}} + \mathcal{H}_{\text{FE}} . \quad (7.29)$$

The Hamiltonian of the total system may be written as

$$\mathcal{H}_{\text{MD}} = \sum_{i < j} \Phi_2(r_{ij}) + \sum_{i, j < k} \Phi_3(r_{ij}, r_{ik}, \Theta_{ijk}) + K, \quad (7.30)$$

where the summation is over all atoms in the system, K is the kinetic energy of the system, r_{ij} and r_{jk} indicate the distance between two atoms and Θ_{ijk} is the bonding angle between three atoms. The finite element Hamiltonian may be written as the sum of the kinetic and potential energies in the elements, i.e.

$$H_{\text{FE}} = \Phi_{\text{FE}} + K_{\text{FE}} , \quad (7.31)$$

where

$$\Phi_{\text{FE}} = \frac{1}{2} \int_{\Omega} \varepsilon_{ij} \varepsilon_{kl} C^{ijkl} d\Omega , \quad (7.32a)$$

$$K_{\text{FE}} = \frac{1}{2} \int_{\Omega} \rho(\mathbf{r})(\dot{\mathbf{v}})^2 d\Omega , \quad (7.32b)$$

where ε is the strain tensor and C is the stiffness tensor, ρ is the material density, and $\dot{\mathbf{v}}$ are the nodal velocities. The contribution to H_{FE} by the potential energy Φ_{FE} , is the

integral of the strain energy and the kinetic energy depends on the nodal velocities. Finally, the tight-binding energy Φ_{TB} may be written as

$$\Phi_{TB} = \sum_{n=1}^{N_{occ}} \varepsilon_n + \sum_{i < j} \Phi^{rep}(r_{ij}) . \quad (7.33)$$

Equation (7.33) shows that Φ_{TB} has contributions from an attractive part ε_n and a repulsive part $\Phi^{rep}(r_{ij})$. The number of occupied states is given by N_{occ} . The overlapping regions which combine (FE/MD and MD/TB) are called “handshake” regions, and each region makes a contribution to the total energy of the system. The handshake potentials are combinations of the potentials given above, with weight factors chosen depending on whether the atomic bond crosses over a given interface. All three equations of motion (TB, MD, FE) are integrated forward using the same timestep. This method was successfully applied to the simulation of brittle fracture [42].

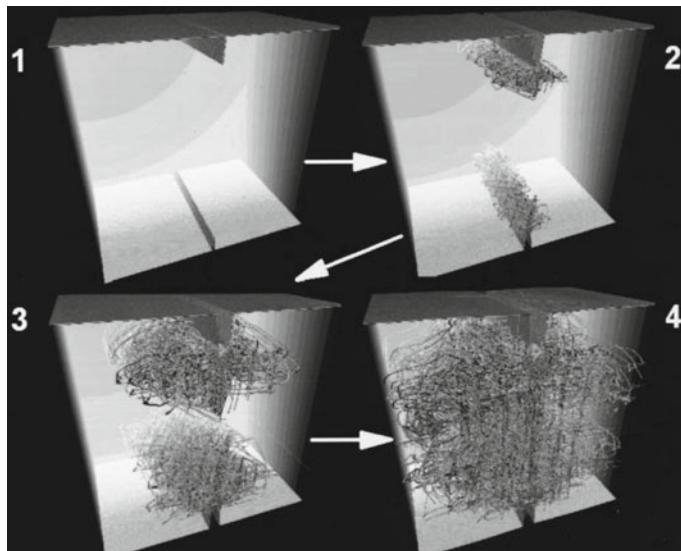


Fig. 7.19 Early-time sequence of the propagating dislocations is shown growing as partial landscapes and the subsequent collision of the dislocations from the opposing notches. The reduced times in a clockwise sequence are 0, 22.5, 45, 67.5 in reduced time units. Only atoms with a potential energy less than 97 % of the bulk value are displayed, resulting in the selected visualization of atoms neighboring surfaces and dislocations. Republished from [10] with permission. Copyright (2007) National Academy of Sciences, U.S.A.

7.6.1 Bridging Scales: Work-Hardening

If brittle failure of materials, as observed e.g. in ceramics, where the only mechanism of failure, then the world would be rather fragile. There are actually two generic types of material failure: ductile and brittle failure. Generally, microcracks are the source of both, ductile and brittle failure. It has been known for long that the strength of most crystalline materials, including metals, derives from the motion, multiplication and interaction of defects, called dislocation lines. These are believed to be responsible for the macroscopically observed phenomenon of strain hardening.

In recent large-scale atomistic simulations, the capabilities of the molecular dynamics method was demonstrated. In Figs. 7.19 and 7.20 one can actually “see” plasticity at work. In these large scale simulations, a detailed understanding of the atomistic origin of strength, brittleness and ductility as typical materials properties could be achieved. Thus, MD simulation methods are gradually approaching a scale, where traditionally purely macroscopic continuum theory would have been applied, and detailed atomistic simulations can provide insight into the fundamental properties of materials.

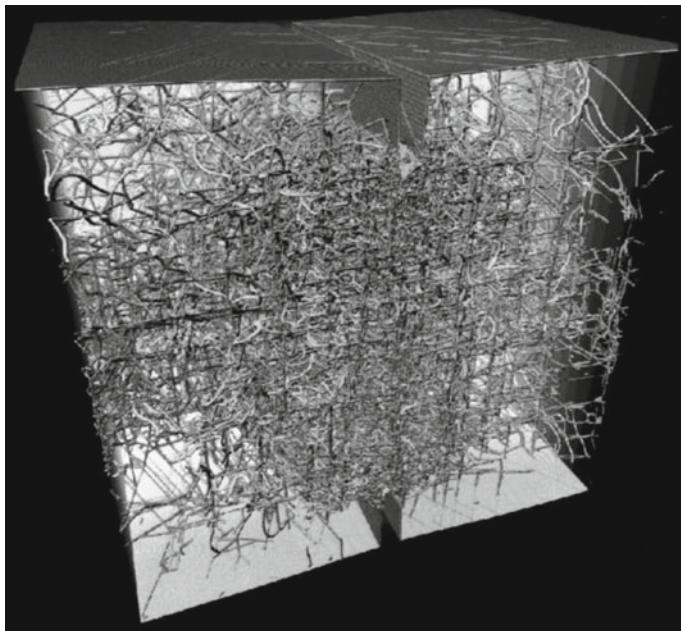


Fig. 7.20 Late-time snapshot picture of the dislocation network of the billion atom slab. A coarse grain cubical skeleton of rigid junctions (sessile dislocations) becomes apparent from a distant view. The reduced simulation time is 700. Republished from [10] with permission. Copyright (2007) National Academy of Sciences, U.S.A.

7.7 Physical Theories for Macroscopic Phenomena: The Continuum Approach

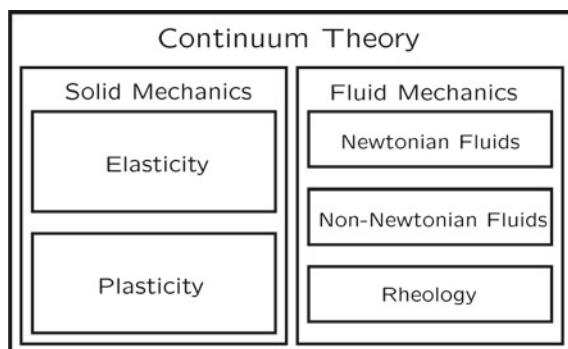
Computational modeling of macroscopic structures, engineering materials and components is usually focused on the improvement of material performance in industrial applications. One tries to better understand and to control material processing, structure, durability, chemistry and their relationships. On the coarse level of a macroscopic length scale one can usually neglect the discrete nature of all matter, i.e. the fact that materials consists of atoms, which allows for the introduction of a continuum model for the considered systems. Consequently, in this section classical continuum field theory is reviewed. In Chap. 3 we already encountered another classical Lorentz invariant field theory, the Maxwell theory of electromagnetism. Field theories treat the considered system as a continuum in space time and derivatives with respect to different independent variables occur in the equations. Thus, physical field theories mathematically lead to the treatment of partial differential equations.

The numerical treatment of the fluid state of matter in the framework of a continuum theory is called computation fluid dynamics (CFD). While generally speaking the number of exact (analytical) solutions of problems in CFD is truly small, the engineering communities have long mastered technical problems that deal with complex flow situations, even in many practical instances where a fundamental physical understanding was lacking. Important recent developments in the subject for practical applications have been achieved - such as boundary layer effects, well-developed turbulence, or the existence of coherent structures in highly turbulent flows (of which the flow around the Jupiter spot is a very impressive example).

As we have already encountered in previous chapters, there is a whole hierarchy of disciplines for a continuum treatment of fluids and solids as depicted in Fig. 7.21.

In elasticity theory, the atoms can only move very small distances about their equilibrium positions (neglecting diffusion effects), otherwise the body breaks. In contrast to this, an arbitrary movement of particles is allowed in the description of fluids and gases. With this respect the term “hydrodynamics” is a bit too confining as the concepts presented here apply as well to gases, as long as they are not too dilute

Fig. 7.21 Hierarchy of disciplines that make use of a mechanical continuum theory



and the flows are slow enough. With high-velocity gases, compressibility effects have to be taken into account and fluid dynamics has to be coupled to thermodynamics by means of the equation of state (EOS). Fluid mechanics today is a branch of mechanics taught within specialized sub-disciplines, well versed in applied mathematics and strongly coupled to the industrial world. The foundations of this discipline were laid in 18th century by the works of Stokes (1819–1903), Navier (1785–1836) and others.

7.7.1 The Description of Fluid Motion

A fluid is considered as a continuum which means that we assume a space filled continuously with matter. This treatment is valid as long as one can use the concept of small representative fluid elements of volume V such that the size $a \propto V^{1/3}$ of one element is:

- Very small relative to the length scale L that is characteristic of the flow, e.g. the size of an obstacle, the radius of a tube or the width of a channel,
- Very large compared to the mean free path l of a molecule.

Within the representative volume of a fluid particle the physical quantities such as temperature, energy or pressure can be considered as constant. A prerequisite for this assumption to be valid is that the dimension of the representative fluid elements (RVE)s is small compared to the average mean squared free path length l . A fundamental problem in hydrodynamics is the representation of the geometric coordinates of RVEs with time. In a simple analogy to classical mechanics of mass points we could simply keep track of the path of the individual RVEs in time. This approach is called *Lagrangian description*.

Lagrangian Description

We consider the total volume of a fluid which is divided into N fluid elements. Each element could be uniquely indexed with a number; however we chose a different way of indexing, cf. [469]. In a fixed coordinate system we introduce at time t_0 N position vectors \mathbf{r}_0 such that each particle is assigned one unique position vector \mathbf{r}_0^i ($i = 1, \dots, N$). The motion of the total fluid is thus completely described by the time dependence of the N position vectors of all fluid elements. Hence, the position vector \mathbf{r} is a function of \mathbf{r}_0 and of time t :

$$\mathbf{r} = F(\mathbf{r}_0, t) \quad (7.34)$$

This apparently simple method of describing the motion of a fluid is for all practical applications turns out to be very laborious and awkward. Generally speaking, when modeling fluids one usually is not really interested into the dynamics of the individual fluid element. Rather one is interested into the general flow condition which is characterized by thermodynamic variables such as temperature T , pressure p or internal energy e at each point of the fluid. This less far-reaching description of fluids is called *Eulerian Description*.

Eulerian Description

In an Eulerian description one is not interested into the individual paths of single fluid elements. Rather, the physical quantities are treated as field variables, i.e. each point in the fluid is assigned a time dependent scalar or tensorial field. In particular, the velocity field \mathbf{v} reads:

$$\mathbf{v} = \mathbf{f}(\mathbf{r}, t) \quad (7.35)$$

In contrast to a Lagrangian description, (7.35) means that at time t the fluid flows with velocity \mathbf{v} at position \mathbf{r} . In this context, the individual path of the fluid element which is at position \mathbf{r} at time t is not considered. This fluid element moves with velocity \mathbf{v} at time t , thus we have:

$$\frac{d\mathbf{r}}{dt} = \mathbf{v} = \mathbf{f}(\mathbf{r}, t) \quad (7.36)$$

Equation (7.36) is a coupled system of ODEs of first order in t . The solution contains three constants which can be interpreted as components of a vector. As a result one can determine a specific position, e.g. \mathbf{r}_0 at a specific time, e.g. t_0 . With this one ends up with Eq. (7.34) again. This transition from the Eulerian to the Lagrangian description and vice versa is always possible in principle. However, the solution of the occurring coupled system of differential equations can be very complex.

Exercise 6 (Eulerian and Lagrangian Description) Consider the velocity field \mathbf{v} in polar coordinates:

$$\mathbf{v}(\mathbf{r}, t) = \frac{k^2 t}{r} \mathbf{e}_r + l \mathbf{e}_\phi + \frac{a}{r} (r_0 - a t_0) \mathbf{e}_r.$$

Calculate the position of fluid elements at time t .

If one looks at a velocity field of a fluid at a certain point in time $t = t_1$ one can draw curves the direction of which in each point agrees with the direction of the velocity field in that point. These curves are called *streamlines*. Hence, an Eulerian description of a fluid yields a series of instantaneous snapshots of the respective flow conditions at that time. There is no relation between the fluid elements and the streamlines as - in general - the streamlines at different times are due to different elements.

7.8 Continuum Theory

When a piece of polymer or rubber is exerted to an external force \mathbf{F} it will be deformed and the mutual distance of the mass points of this body changes. In this case the standard mechanical model of a rigid body is not appropriate any more for a description of the material properties upon loading, cf. our discussion of model hierarchies in Chap. 2. Materials which return into their undeformed initial state after

the external force has vanished are generally called elastic or deformable media and mark a sub area of classical mechanics. In this case, instead of considering many atoms or molecules, one introduces a deformable continuum which is treated within the theory of elasticity, a classical field theory. This theory was formerly an essential part of classical mechanics courses, while today it is considered as a limiting case of classical solid state physics. This limiting case can be nicely illustrated with the model of a linear chain as is done in Sect. 7.9.

A similar situation arises in the discussion of the properties of (classical) fluids and gases within the continuum model. These branches of classical mechanics, termed *hydrodynamics* and *aerodynamics*, are treated as continuum limiting cases in condensed matter physics because for a consistent description of fluids and gases thermodynamic relations are needed.

7.8.1 The Continuum Hypothesis

Continuum Mechanics is based on the assumption of the existence of a continuum, in engineering textbooks sometimes called “Continuum Hypothesis”. Mathematically, a notion of a continuum is given in real analysis by the fundamental limiting process used in a Cauchy series. This expresses the idea, that the real numbers are “arbitrarily dense”, that is, there is no “space left” in between two real numbers for another number. Another way of expressing this idea in the language of sets is to say that the reals are an uncountable infinite set.

The existence of mathematical objects does not necessarily mean, that these objects also exist (or have any meaning) in the physical world. Physics is an experimental science and all theorems and ideas ultimately have to pass the experimental test. Using the real numbers as an example we want to expand on this idea.

One of the most fundamental notions used in mathematics is the axiom of the existence of real numbers,² denoted with the letter **R**. Real numbers are called “real” because they seem to provide the magnitudes needed for the measurement of distance, angle, time, or numerous other physical quantities. Hence, are real numbers “real”?

Real numbers refer to a *mathematical idealization*, a free invention of human mind, rather than to any actual physical objective reality. For example, the system of real numbers has the property, that between any of them, no matter how close, there lies a third.³ It is very questionable, whether physical distances or time intervals actually have this property. If one continues to divide up the physical distance between two points, one eventually reaches scales so small that the whole concept of distance, in the intuitive primitive sense, could cease to have any meaning. In this context it is interesting to mention that as early as 1899, Max Planck realized that the fundamental

²It is attributed to Leopold Kronecker (1823–1891) that he once said: “The natural numbers are God-given. The rest is the work of man” [470].

³This property is a consequence of Dedekind’s *Schnittaxiom*, see e.g. [35] or the original sources by R. Dedekind [471] (1892) and [472] (1893).

physical constants G (Newton's gravitational constant), c (velocity of light), and $\hbar = \frac{h}{2\pi}$ (later called Planck's quantum of action) could be combined in such a way as to build a length, a time and a mass [2]. These fundamental values are given by:

$$\text{(Planck length)} \quad l_P = \sqrt{\frac{\hbar G}{c^3}} \approx 1.62 \times 10^{-35} \text{ m}, \quad (7.37a)$$

$$\text{(Planck time)} \quad t_P = \frac{l_P}{c} = \sqrt{\frac{\hbar G}{c^5}} \approx 5.40 \times 10^{-44} \text{ s}, \quad (7.37b)$$

$$\text{(Planck mass)} \quad m_P = \frac{\hbar}{l_P c} = \sqrt{\frac{\hbar G}{c}} \approx 2.17 \times 10^{-8} \text{ kg}. \quad (7.37c)$$

It is anticipated that on this "quantum gravity scale" or "Planck scale", due to Heisenberg's uncertainty principle, the ordinary notion of distance and also the notion of a continuum become meaningless. At this scale, one will start to notice the quantum nature of time which is connected with energy by an uncertainty relation $\Delta E \Delta t \geq \hbar$. Obviously, distances smaller than the order of magnitude of Planck's constant do not make any physical sense anymore, because the uncertainty principle as a fundamental physical law of nature prevents any useful measurements of length at this scale. But in order to really mirror the real numbers one would have to go far beyond that scale, e.g. to distances 10^{300} th l_P or $10^{10^{200}}$ th l_P . Thus, are the real numbers to be considered as "real"?

The example shows that theoretical constructs or models are only "real" to the extend of their leading to falsifiable consequences. The real numbers have been chosen in physics because they are known to agree with physical continuum concepts that are used for e.g. distance and time but not because they are known to agree with physical concepts over *all* scales. The applicability of real numbers for the description of a continuum in the natural sciences is however very large and has been applied to the smallest subatomic scales (quarks) $\sim 10^{-19}$ m, and to the large scale structures in the universe $\sim 10^{25}$ m, thus extending the applicability range of the order of $\sim 10^{44}$. Assuming that reals are a useful concept down to the Planck scale, their range of applicability and thus their "reality" is extended to at least $\sim 10^{60}$. Wigner writes about "The unreasonable effectiveness of mathematics in the natural sciences" [473]:

It is true, of course, that physics chooses certain mathematical concepts for the formulation of the laws of nature, and surely only a fraction of all mathematical concepts is used in physics. It is true also that the concepts which were chosen were not selected arbitrarily from a listing of mathematical terms but were developed, in many if not most cases, independently by the physicist and recognized then as having been conceived before by the mathematician. It is not true, however, as it is often stated, that this had to happen, because mathematics uses the simplest possible concepts and these were bound to occur in any formalism.[...] the concepts of mathematics are not chosen for their conceptual simplicity [...], but for their amenability to clever manipulations and to striking, brilliant arguments.

(Eugene P. Wigner, 1960, p. 7)

In mathematics, the *continuum hypothesis* was proposed by Georg Cantor [474] after he showed that the real numbers cannot be put into one-to-one correspondence with

the natural numbers, cf. our discussion in Chap. 3 on p. 119. Cantor suggested that the cardinality of real numbers is the next level of infinity above the number of natural numbers. He used the Hebrew letter “aleph” to name the different levels of infinity: \aleph_0 is the number (or cardinality) of the natural numbers or any countable infinite set, and the next levels of infinity are $\aleph_1, \aleph_2, \aleph_3, \dots$. Since the reals form the quintessential notion of continuum, Cantor named the cardinality of the reals “ c ”, for continuum. The continuum hypotheses is the assumption that there is no set with a cardinal number in between the number of natural numbers and the number of reals (the continuum), or simply stated: $\aleph_1 = c$.

In the year 1938 Kurt Gödel [174, 475] proved that the continuum hypothesis cannot be disproved within the generally accepted system of axioms for set theory, the Zermelo–Fraenkel (ZF) axioms, see Definition 8 in Appendix B. Later, in the year 1963, Paul Cohen [476, 477] was able to prove, that within the ZF system of axioms, also the validity of the continuum hypotheses cannot be proved. Thus, within the ZF system of set axioms, the continuum hypothesis is *undecidable*.⁴

7.9 Theory of Elasticity

In contrast to the theory of rigid bodies, the theory of elasticity investigates the mutual movements of the constituent particles of a body. If the mutual movements of particles are confined to a few lattice spacings one can ignore the discreteness of the body and make use of a continuum model. The advantage of continuum models is that the number of relevant equations is reduced considerably. We illustrate this with the model of a linear chain, cf. Fig. 7.22.

The linear chain consists of N mass points m connected with springs with constant λ . The chain is translation invariant in one dimension. In a particle treatment one obtains as equation of motion the following equation:

$$m\ddot{u}_i + \lambda(2u_i - u_{i+1} - u_{i-1}) = 0. \quad (7.38)$$

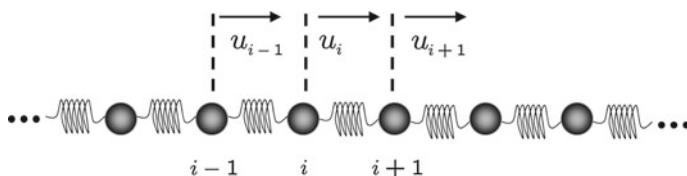


Fig. 7.22 A linear chain as a simple discrete model for a solid state

⁴For a definition of decidability see Definition 2.3 on p. 100.

These are N coupled ordinary linear differential equations. Assuming that the movement of neighboring mass points is almost identical, one can set:

$$u_i = u(x_i, t) = u(x, t), \quad (7.39)$$

with t as time parameter. It follows:

$$2u_i - u_{i+1} - u_{i-1} = 2u(x) + u(x + \Delta x) - u(x - \Delta x) = -\frac{\partial^2 u}{\partial x^2}(\Delta x)^2, \quad (7.40)$$

with Δx being the mutual particle distance. In the limit of $\Delta x \rightarrow 0$, and at the same time keeping the product $\Delta x \cdot \lambda = f$ constant, one obtains with $c^2 = f/\rho = \lambda(\Delta x)^2/m$ the wave equation

$$\frac{\partial^2 u}{\partial x^2} - \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} = 0. \quad (7.41)$$

Thus, instead of having to solve $3N$ coupled equations, in a continuum description there is only *one* but partial differential equation.

Exercise 7 (Linear Chain) Solve the differential equation Eq. (7.38).

Solution 5 The total force on the chain is given by

$$F_s = \sum_n f_n(u_{s+1} - u_s). \quad (7.42)$$

A possible ansatz for the solution is

$$u_{s+1} = u \exp(\pm i(qa - \omega t)) \quad (7.43)$$

with wave vector q , distance a of neighboring atoms on the lattice and the angular frequency ω . Inserting this ansatz into (7.38) yields

$$\omega^2 M = f(2 - \exp(iqa) + \exp(-iqa)) = 2f(\cos(qa) - 1) = 4f \sin^2 \frac{qa}{2}. \quad (7.44)$$

Thus, as a result, one obtains for the angular frequency of the wave that is propagating in the crystal, the *dispersion relation*

$$\omega = 2\sqrt{\frac{f}{M}} \left| \sin \frac{qa}{2} \right|. \quad (7.45)$$

Looking at the elongations of neighboring particles

$$\frac{u_{s+1}}{u_s} = \frac{u \exp(i(s+1)qa)}{u \exp(isqa)} = \exp(iqa) \quad (7.46)$$

one realizes, that the useful range of the exponential function with phase qa is restricted to the interval ($-\pi = qa = \pi$). A larger difference in phase than $\pm\pi$ makes physically no sense, as all behavior can always be reduced to this interval by consecutive subtraction of 2π in the phase. Thus, physically useful values of wave vector q are restricted to the *Brillouin-zone*:

$$-\frac{\pi}{a} \leq q \leq \frac{\pi}{a}. \quad (7.47)$$

From (7.45) it follows that in the limit $q \ll 1/a$, i.e. for wave lengths ω that are large compared to a lattice constant a , the angular frequency is directly proportional to q , i.e.:

$$\omega = \sqrt{\frac{fa^2}{M}} q . \quad (7.48)$$

In this case, the crystal lattice behaves like a continuum and deviations from continuum behavior are the stronger, the closer the wave vector are to the edges $\omega \pm \pi/a$ of the Brioullin-zone. The group velocity v_g of the wave packet which transports energy through the lattice is given by

$$v_s = \frac{d\omega}{dq} = \sqrt{\frac{fa^2}{M}} \cos \frac{qa}{2} . \quad (7.49)$$

In the case $q \ll a$, v_g is independent of the wave number q . The expression of v_g then corresponds to the propagation of a longitudinal sound wave. Assuming for the velocity of sound $v_s = 4 \times 10^3 \text{ ms}^{-1}$ and a lattice constant $a = 2 \times 10^{-10}$, one obtains for the phonon frequency

$$\omega \approx v_s q = v_s \frac{\pi}{a} = 2\pi \times 10^3 \text{ Hz} . \quad (7.50)$$

To experimentally probe the properties of lattice oscillations of solids and fluids, different elastic scattering techniques are used, e.g. X-rays, photons or neutrons; the different scattering techniques only differ by the amount of energy that is carried by the different scattering particles. In Fig. 7.23 a schematic overview of the different experimental scattering techniques for probing properties of liquids or solids is provided.

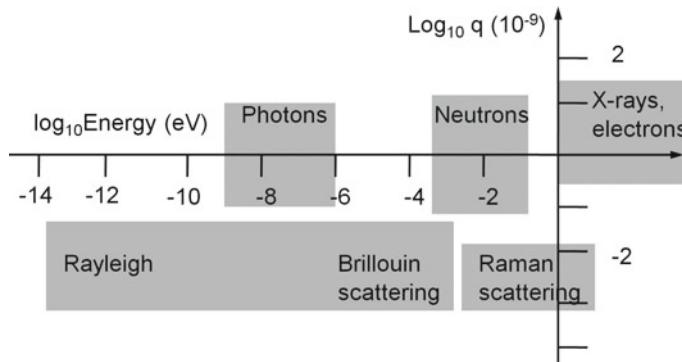


Fig. 7.23 Regions of energy and wave-vector transfer for various experimental probes used in the spectroscopic analysis on a equidistant log-scale

7.9.1 Kinematic Equations

An abstract notion of a continuous body B can be obtained by considering points P in a compact Hausdorff space H , cf. our discussion in Chap. 3. As the points $P \in \mathbf{R}^3$, their number is uncountable. The Hausdorff property ensures that the points in the body are arbitrarily “dense”. In order to have something specific in mind as a visualization, we imagine as body B a rubber blanket which is fixed at two sides. We consider this body before and after deformation, e.g. due to a force applied at one of the edges, cf. Fig. 7.24.

In the undeformed body we establish an arbitrarily dense coordinate system x^i (so-called *Gaussian coordinates* in two dimensions) and affix the values of its coordinates to each material point (particle). This means that such points will also be known by the same values x^i even after deformation, although moved to a different place. In other words, the coordinate system undergoes the same deformation as the body. Coordinates used in this way are called *material (particle)* or *convected coordinates*. Before the deformation we have base vectors \mathbf{b}_i and a metric tensor g_{ij} such that a line element can be written as

$$d\mathbf{r} = \mathbf{b}_i dx^i , \quad (7.51)$$

and its square

$$dr^2 = d\mathbf{r} \cdot d\mathbf{r} = \mathbf{b}_i \mathbf{b}_j dx^i dx^j = g_{ij} dx^i dx^j . \quad (7.52)$$

After the deformation, the line element $d\hat{\mathbf{r}}$ connecting the same material points is different in length and may be written

$$d\hat{\mathbf{r}} = \hat{\mathbf{b}}_i dx^i . \quad (7.53)$$

The square of the line element is now

$$d\hat{r}^2 = d\hat{\mathbf{r}} \cdot d\hat{\mathbf{r}} = \hat{g}_{ij} dx^i dx^j . \quad (7.54)$$

Here $\hat{\mathbf{b}}_i$ is the vector into which the original base vector \mathbf{b}_i has been deformed, and \hat{g}_{ij} is the metric tensor in the deformed coordinate system. The degree of deformation

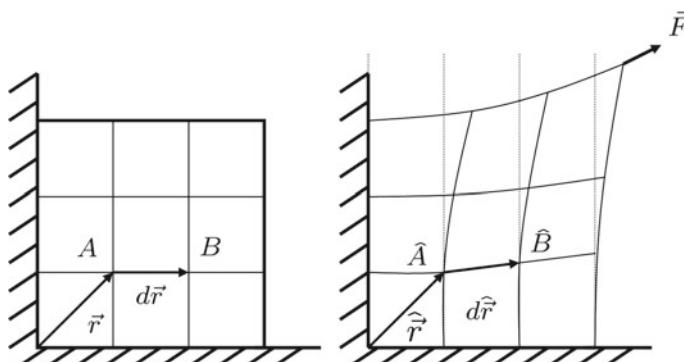


Fig. 7.24 Illustration of the deformation of an elastic body

of a body can be described by the change of the metric tensor, i.e. by the following equation:

$$\gamma_{ij} = \hat{g}_{ij} - g_{ij} . \quad (7.55)$$

Both \hat{g}_{ij} and g_{ij} are components of tensors, but in different reference frames, and it is not evident whether their difference γ_{ij} is a tensor in the undeformed reference frame \mathbf{b}_i . To clarify the situation, we transform everything from the coordinate system x^i to another one, $x^{i'}$, in which

$$d\mathbf{r} = \mathbf{b}_{i'} dx^{i'} , \quad (7.56)$$

and

$$d\hat{\mathbf{r}} = \hat{\mathbf{b}}_{i'} dx^{i'} . \quad (7.57)$$

The undeformed base vectors are related by a Jacobian matrix α_j^i :

$$\mathbf{b}_{i'} = \alpha_{i'}^i \mathbf{b}_i . \quad (7.58)$$

Similarly, for the deformed state:

$$\hat{\mathbf{b}}_{i'} = \hat{\alpha}_{i'}^i \hat{\mathbf{b}}_i . \quad (7.59)$$

Since we are using convective coordinates, there is no difference between x^i and \hat{x}^i and one obtains:

$$\hat{\alpha}_{i'}^i = \frac{\partial x^i}{\partial x^{i'}} = \alpha_{i'}^i . \quad (7.60)$$

Hence, both terms on the right hand side in (7.55) are transformed the same way:

$$\gamma_{i'j'} = \gamma_{ij} \alpha_{i'}^i \alpha_{j'}^j . \quad (7.61)$$

Equation (7.61) proves the tensor character of γ_{ij} ; the entity of the nine components γ_{ij} are called *strain tensor*. In order to derive contravariant strain components γ^{ij} , we now straightforwardly define the quantities

$$\xi^{ij} = \hat{g}^{ij} - g^{ij} . \quad (7.62)$$

The components \hat{g}_{ij} and \hat{g}^{ij} of the metric tensor of the deformed medium are related by the equation

$$\hat{g}_{ij} \hat{g}^{jk} = \hat{\delta}_i^k = \delta_i^k , \quad (7.63)$$

and using (7.55) and (7.62), one obtains

$$(g_{ij} + \gamma_{ij})(g^{jk} + \xi^{jk}) = \delta_i^k + \gamma_i^k + \xi_i^k + \gamma_{ij}\xi_{jk} = \delta_i^k . \quad (7.64)$$

The mixed term in the previous equation is quadratic in strain-like quantities. Neglecting this term one obtains simply

$$\xi_i^k = -\gamma_i^k , \quad (7.65)$$

hence

$$\xi^{ik} = -\gamma^{ik} , \quad (7.66)$$

and finally

$$-\gamma^{ij} = \hat{g}^{ij} - g^{ij} . \quad (7.67)$$

The deformation of a body is completely known when, for each of its points, one knows the displacement vector \mathbf{u} which extends from the position before deformation to that position described by the same material point after deformation. In other words, the components γ_{ij} may be expressed in terms of the displacement \mathbf{u} . Next, we derive this relation. In doing so, we restrict ourselves to rectilinear coordinates, such that the base vectors $\mathbf{b}_i = \partial \mathbf{r} / \partial x^i$ are constant. Thus, for the differential of the displacement \mathbf{u} one obtains:

$$d\mathbf{u} = \mathbf{b}^i \frac{\partial u_i}{\partial x^j} dx^j = \mathbf{b}^i u_{i,j} dx^j . \quad (7.68)$$

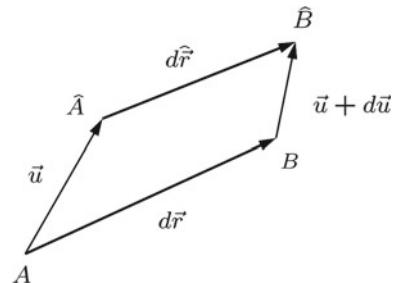
We now return to the consideration of two material points A and B in Fig. 7.24 before and after deformation of the body. A magnified cutout of the situation is displayed in Fig. 7.25.

From Fig. 7.25 and (7.68) one has

$$d\hat{\mathbf{r}} = d\mathbf{r} + d\mathbf{u} = d\mathbf{r} + \mathbf{b}^i u_{i,j} dx^j . \quad (7.69)$$

In the following, several index techniques are demonstrated which are very common in the use of tensor equations. Writing down the square of the deformed line element

Fig. 7.25 Displacement and strain of an elastic body.
During the deformation, A undergoes a displacement \mathbf{u} and moves to \hat{A} , while B experiences a slightly different displacement $\mathbf{u} + d\mathbf{u}$ when moving to \hat{B}



yields:

$$\hat{dr} \cdot \hat{dr} = (\mathbf{b}_i dx^i + \mathbf{b}^k u_{k,i} dx^i) \cdot (\mathbf{b}_j dx^j + \mathbf{b}^l u_{l,j} dx^j) \quad (7.70a)$$

$$= (g_{ij} + \mathbf{b}_i \mathbf{b}^l u_{l,j} + \mathbf{b}^k \mathbf{b}_j u_{k,i} + g^{kl} u_{k,i} u_{l,j}) dx^i dx^j \quad (7.70b)$$

$$\stackrel{l \rightarrow k}{=} (g_{ij} + g_i g^k u_{k,j} + g_j g^k u_{k,i} + g^{kl} u_{k,i} u_{l,j}) dx^i dx^j \quad (7.70c)$$

$$\stackrel{i \rightleftarrows j}{=} (g_{ij} + g_i g^k u_{k,j} + g_i g^k u_{k,j} + g^{kl} u_{k,j} u_{l,i}) dx^i dx^j \quad (7.70d)$$

$$= (g_{ij} + 2 \cdot g_i g^k u_{k,j} + g^{kl} u_{k,j} u_{l,i}) dx^i dx^j \quad (7.70e)$$

In (7.70a)–(7.70e) all indices are dummy indices, i.e. they are all summed over. In the second term of (7.70c) index l was changed to k and in the third term of (7.70d) index i was interchanged with j . With the help of (7.54) and (7.55), the result (7.70e) can be written as

$$\gamma_{ij} dx^i dx^j = (2u_{i,j} + g^{kl} u_{k,j} u_{l,i}) dx^i dx^j . \quad (7.71)$$

It follows

$$2\gamma_{ij} dx^i dx^j = (\gamma_{ij} + \gamma_{ji}) dx^i dx^j \quad (7.72a)$$

$$= (2u_{i,j} + 2u_{j,i} + \underbrace{g^{kl} u_{k,j} u_{l,i} + g^{kl} u_{k,i} u_{l,j}}_{=(g^{kl} u_k)_{,j} u_{l,i} + (g^{kl} u_k)_{,i} u_{l,j}}) dx^i dx^j \quad (7.72b)$$

$$\stackrel{i \rightleftarrows j}{=} 2 \left(u_{i,j} + u_{j,i} + (g^{kl} u_k)_{,i} u_{l,j} \right) dx^i dx^j . \quad (7.72c)$$

In (7.72b) the metric tensor components g^{kl} – which are constants in rectilinear coordinates have been included in the partial derivatives. In (7.72c) indices i and j have been interchanged in the third term. As (7.70) is valid for any product $dx^i dx^j$, it finally follows:

$$\gamma_{ij} = u_{i,j} + u_{j,i} + u^k_{,j} u_{k,j} . \quad (7.73)$$

The last term on the right-hand side of (7.73) is quadratic in the displacement \mathbf{u} and therefore negligible when the displacement is sufficiently small. In order to show that the components γ_{ij} are identical with the quantities ϵ_{ij} which are commonly called *strain*, we specialize (7.73) to Cartesian coordinates (i.e. co- and contravariant components are the same) using

$$dx^1 = dx , \quad dx^2 = dy , \quad dx^3 = dz , \quad (7.74)$$

and

$$u_1 = du , \quad u_2 = dv , \quad u_3 = dw . \quad (7.75)$$

From (7.73) we thus yield

$$\gamma_{11} = 2 \frac{\partial u}{\partial x} + \left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 + \left(\frac{\partial w}{\partial z} \right)^2, \quad (7.76)$$

$$\gamma_{12} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} + \frac{\partial u}{\partial x} \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \frac{\partial v}{\partial y} + \frac{\partial w}{\partial x} \frac{\partial w}{\partial y}. \quad (7.77)$$

These expressions are equal to $2\epsilon_x$ and γ_{xy} respectively, if one adopts a strain definition based on the change of the *square* of the line element. In this case they are not second-order approximations, but are exact for any amount of deformation. This is not the case, if the strain ϵ_x is defined as the increment of length divided by the original length (*linear strain*). As a measure of strain we will use the tensor $\epsilon_{ij} = \frac{1}{2}\gamma_{ij}$, mainly because products of the ϵ_{ij} with the stresses are the work done during the deformation. For later reference we write the relations in explicit linearized form

$$\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \text{ small displacements,} \quad (7.78)$$

and the exact formula

$$\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i} + u^k,_i u_{k,j}) \text{ large displacements.} \quad (7.79)$$

Equations (7.78) and (7.79) are independent of the forces acting, and of the elastic or inelastic character of the material. They are concerned with the geometry of the motion which leads from the undeformed state to the deformed position and are thus known as *kinematic relations*.

The validity of (7.78) and (7.79) is restricted to coordinate systems in which every component of the metric tensor is a constant, i.e. for which $g_{ij,k} = 0$. Because of the presence of partial derivatives $u_{i,j}$, these equations are not tensor equations. Thus, we substitute the partial derivatives by covariant derivatives and obtain:

$$\epsilon_{ij} = \frac{1}{2} (u_{i|j} + u_{j|i}) \quad (7.80)$$

for the linear version of the kinematic equation and

$$\epsilon_{ij} = \frac{1}{2} (u_{i|j} + u_{j|i} + u^k,_i u_{k|j}) \quad (7.81)$$

for the exact equation. In this form they are tensor equations and can be carried over to any arbitrary coordinate system. Equation (7.80) is the symmetric part of the tensor of the displacement derivatives $u_{i|j}$. The antimetric part $\frac{1}{2} (u_{i|j} - u_{j|i})$ may be associated with a vector

$$-2\omega^k = u_{i|j} \epsilon^{ijk}. \quad (7.82)$$

Using \mathbf{b}_k as a base vector in an explicit coordinate system one obtains in ordinary vector notation

$$\omega^k \mathbf{b}_k = \frac{1}{2} \nabla \times \mathbf{u} . \quad (7.83)$$

Compatibility Conditions

There are three different displacement components u_i , but six different strain components ϵ_{ij} . If one wants to use (7.80) to calculate the strains from the displacements, one has one equation for every ϵ_{ij} , but if one considers them as differential equations for the u_i , one has three more equations than unknowns. Thus, there must be some conditions to which these functions are subjected as one cannot expect that they are compatible with each other for any given set of six functions $\epsilon_{ij}(x^k)$. Calculating the second derivatives of (7.80) yields

$$2\epsilon_{ij|kl} = u_{i|jkl} + u_{j|ikl} . \quad (7.84)$$

We note that the first term on the right-hand side of (7.84) is symmetric with respect to the indices j and l . Hence, its product with the antimetric permutation tensor ε (or in components: ε^{jln}) vanishes. Similarly, the product of the second term and ε^{ikm} is zero, and when one multiplies the equation with both factors, one eliminates the displacements from (7.84) and obtains

$$\epsilon_{ij|jk} \varepsilon^{ikm} \varepsilon^{jln} = 0 . \quad (7.85)$$

This equation is called the *compatibility conditions*. There are six essentially different component equations for each one of the choices $mn = \{11, 22, 33, 12, 23, 31\}$.

7.9.2 The Stress Tensor

We consider a small volume element of an elastic body, e.g. the rubber blanket in Fig. 7.24 and consider some forces acting on this volume element. One has to distinguish two types of forces, *volume* forces and *surface* forces. In Fig. 7.26 an infinitesimal volume with parallel faces is displayed which is cut from a fluid or solid material supposed to be in equilibrium. The edges of this block are given by $dr^i \mathbf{b}_i$, $ds^j \mathbf{b}_j$, $ds^k \mathbf{b}_k$. The grey shaded face on one of its sides has the area $dA_l = ds^j dt^k \epsilon_{jkl}$. In first approximation the force $dA_i \sigma^{lm} \mathbf{b}_m$ is transmitted through this face and through the face opposite it. These two forces – due to the separation of the faces by dr^i – differ from each other in second approximation by

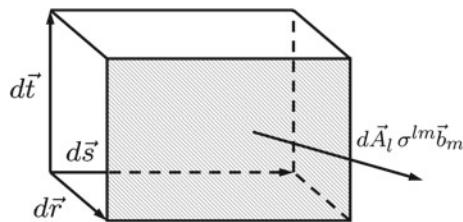
$$(dA_i \sigma^{lm})_{|i} dr^i \mathbf{b}_m = dA_i \sigma^{lm}_{|i} dr^i \mathbf{b}_m = ds^j dt^k \epsilon_{ijk} \sigma^{lm}_{|i} dr^i \mathbf{b}_m . \quad (7.86)$$

For the other two opposite faces one obtains similar expressions for the unbalanced forces, and all contributions thus add up to

$$dr^i ds^j dt^k \left(\sigma^{lm}_{|i} \epsilon_{jkl} + \sigma^{lm}_{|j} \epsilon_{kil} + \sigma^{lm}_{|k} \epsilon_{ijl} \right) \mathbf{b}_m . \quad (7.87)$$

Fig. 7.26 Derivation of the equilibrium condition.

Displayed are the stresses acting on a volume element $dV = dr^i dr^j dr^k \epsilon_{ijk}$



This equation is equal to the expression

$$dr^i ds^j dt^k \sigma^{lm} \mid_l \epsilon_{ijk} \mathbf{b}_m . \quad (7.88)$$

Exercise 8 (Show the equivalence of (7.88)–(7.87))

Proof In (7.87) all indices are dummy indices and the dot product of the differentials is commutative, i.e. independent of the order of terms; when closely examining the three terms in the sum of (7.87) one realizes that the sum over l in each term only contains at most one single term because l must be different from the other two indices in the ϵ -factor. If these two indices of ϵ are the same, then all terms of the sum drop out altogether. As a typical example, consider the case ($i, j, k = 1, 2, 3$). In this case (7.87) becomes

$$dr^1 ds^2 dt^3 (\sigma^{1m} \mid_1 \epsilon_{231} + \sigma^{2m} \mid_2 \epsilon_{312} + \sigma^{3m} \mid_3 \epsilon_{123}) \mathbf{b}_m = dr^1 ds^2 dt^3 (\sigma^{im} \mid_i \epsilon_{123}) \mathbf{b}_m . \quad (7.89)$$

If two indices are equal, e.g. ($i, j, k = 1, 1, 2$) then one obtains

$$dr^1 ds^1 dt^2 (\sigma^{3m} \mid_1 \epsilon_{123} + \sigma^{3m} \mid_1 \epsilon_{213} + 0) \mathbf{b}_m = 0 . \quad (7.90)$$

Thus, one sees that the expression in parentheses in (7.88) is only nonzero, if ($i \neq j \neq k$) and due to the result in (7.89), one obtains (7.88) ■

Equation (7.88) expresses the resultant of all stresses acting on the infinitesimal volume element. It can be written in the more succinct form $\sigma^{lm} \mid_l dV \mathbf{b}_m$ or, when changing dummy indices, as $\sigma^{ji} \mid_j dV \mathbf{b}_i$.

Additionally, there may be a volume force acting on the volume element which can be written as $X^i dV \mathbf{b}_i$. In equilibrium, the sum of the volume force and of the unbalanced stresses must equal zero:

$$\left(\sigma^{ij} \mid_j + X^i \right) dV \mathbf{b}_i = 0 . \quad (7.91)$$

Equation (7.91) holds for each component, thus, after dropping the constant factor dV one obtains:

$$\sigma^{ij} \mid_j + X^i = 0 \text{ (Static equilibrium condition).} \quad (7.92)$$

This is the condition of equilibrium. The equilibrium of moments is taken care of by the symmetry of the stress tensor, i.e. $\sigma^{ij} = \sigma^{ji}$.

If the body is in motion, one can choose from two formulations of the problem. In the particle or *Lagrangian formulation* one attaches coordinates, that is, unique labels to the individual particles, and write down the dynamic equations in terms of these coordinates, which are moved and deformed with the material. In the *Eulerian* or field *formulation* one uses a fixed, rigid coordinate system and lets the material move in it. In the Langrangian case the derived equations consider the behavior of a single particle and in the Euler case the equations describe what happens at a certain point in space. Note that for *small* deviations of the constituent particles of a solid about their equilibrium positions, both formulations coincide. In this case, the velocity is simply the partial derivative of the displacement $\mathbf{u} = \mathbf{u}(t, \mathbf{x})$, i.e. $\dot{\mathbf{u}}_i = \partial u_i / \partial t$.

To obtain the dynamics equations, when the material moves, one has to add d'Alembert's inertia forces (per unit volume) as additional forces to the previous volume forces X^i . This leads to the dynamic equation

$$\mathbf{S}\ddot{\mathbf{u}}^i \text{ (dynamic equation).} \quad (7.93)$$

7.9.3 Equations of Motion of the Theory of Elasticity

The basic ingredients for the treatment of condensed matter within the framework of continuum theory, are the *kinematic relations* (7.79), the equilibrium conditions (7.92) and, when considering d'Alembert's inert force, the above equations of motion (7.93). In order to calculate something one has to make an assumption on the behavior of a system under load, i.e. one has to come up with a model for the stress-strain relationship, the so-called constitutive equations. In the next section, these equations are to be discussed for the simplest case, a linear elastic material.

7.9.4 Constitutive Equations

The mechanics of continua draws its physical information from three sources:

- conditions of equilibrium
- kinematic relations
- stress-strain relations

The stress-strain relations are in essence algebraic equations and describe the mechanical properties of the material upon loading - its constitutive equations. There are as many kinds of such equations as there are different materials. The sheer number of different constitutive laws already provides an indication of the general validity of constitutive models: Usually they are of very limited use and can only be applied within the boundaries of the experimental conditions with which the phenomenological material relations were obtained. We start out with a linear elastic

but not necessarily isotropic material. The relation between stress σ^{ij} and strain ϵ_{kl} is:

$$\sigma^{ij} = E^{ijkl} \epsilon_{kl} . \quad (7.94)$$

This is Hooke's law for a general anisotropic solid.

The stress-strain relationships are the mathematical description of the mechanical properties of a material. In fact, in engineering, the purely empirical determination of *material models* is generally constraint to the determination of the complex stress-strain relationships under a multitude of external loading conditions in terms of temperature, pressure, shear rate, strain rate, or uni-axial and multi-axial load. The intricate experimental efforts to determine a system's answer to external load then eventually results in a constitutive equation which can be used in a FE analysis of the material. For a linear elastic material one has a linear relationship between stress σ^{ij} and strain ϵ_{ij} . As both quantities are tensors (defined in different tangent spaces at the same point), the proportionality constant (the elastic moduli) has to be a tensor of rank four. Thus, the relation is

$$\sigma^{ij} = C^{ijkl} \epsilon_{kl} . \quad (7.95)$$

This is Hooke's law for a general anisotropic solid. In total there are $3^4 = 81$ components of E , but they are not all different from each other. Since $\sigma^{ij} = \sigma^{ji}$, there must be $C^{ijkl} = C^{jikl}$ and due to $\epsilon_{lm} = \epsilon_{ml}$ one has

$$C^{ijkl} = C^{jikl} = C^{jilk} = C^{ijlk} . \quad (7.96)$$

Within each pair of subscripts $\{ij\}$ and $\{lm\}$, one can interchange the order. Hence, there are six different pairs left ($\{1, 1\}; \{1, 2\}; \{1, 3\}; \{2, 2\}; \{2, 3\}; \{3, 3\}$) and $6^2 = 36$ different values C^{ijlk} . By taking into account a further reduction of components due to a symmetry (the existence of a strain energy), there are in general 21 different elastic constants, that characterize the elastic properties of a general anisotropic solid. This number is further reduced, if the crystal has additional symmetries, cf. Table 7.1.

Table 7.1 Elastic constants of different crystal systems

System	Crystallographic point group	Elastic Constants
Triclinic	all	21
Monoclinic	all	13
Orthorhombic	all	9
Tetragonal	$C_4, C_{4h}, S_4, C_{4v}, D_{4v}, D_{4h}, D_{2d}$	7
Rhombohedral	$C_3, S_6, C_{3v}, D_{3v}, D_{3d}$	6
Hexagonal	all	5
Cubic	all	3

Hence, according to Table 7.1 a cubic crystal for example has only 3 different independent elastic constants which can conveniently be displayed in a matrix as

$$\begin{pmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{pmatrix} \quad (7.97)$$

Example 39 (Strain Energy) In Cartesian coordinates one has for the strain energy density a :

$$a = \frac{1}{2}(\sigma_x \varepsilon_x + \sigma_y \varepsilon_y + \sigma_z \varepsilon_z + \tau_{xy} \gamma_{xy} + \tau_{yz} \gamma_{yz} + \tau_{zx} \gamma_{zx}) . \quad (7.98)$$

This equation can be written in tensor form – valid in all coordinate systems – as

$$a = \frac{1}{2} \sigma^{ij} \varepsilon_{ij} . \quad (7.99)$$

For small increments of stress and strain one follows that

$$da = \sigma^{ij} d\varepsilon_{ij} = \frac{\partial a}{\partial \sigma^{ij}} d\sigma^{ij} + \frac{\partial a}{\partial \varepsilon_{ij}} d\varepsilon_{ij} \quad (7.100)$$

Inserting (7.95) into (7.100) one obtains:

$$\begin{aligned} da &= \frac{\partial a}{\partial \sigma^{ij}} \frac{\partial \sigma^{ij}}{\partial \varepsilon_{lm}} d\varepsilon_{lm} + \frac{\partial a}{\partial \varepsilon_{ij}} d\varepsilon_{ij} = \frac{\partial a}{\partial \sigma^{lm}} \frac{\partial \sigma^{lm}}{\partial \varepsilon_{ij}} d\varepsilon_{ij} + \frac{\partial a}{\partial \varepsilon_{ij}} d\varepsilon_{ij}, \\ &= \frac{1}{2} (\varepsilon_{lm} C^{lmi j} + \sigma^{ij}) d\varepsilon_{ij} . \end{aligned} \quad (7.101)$$

Since $\varepsilon_{ij} = \varepsilon_{ji}$, the quantities $d\varepsilon_{ij}$ cannot be chosen entirely arbitrarily but the right hand side of (7.99) and (7.101) have to be the same, i.e.

$$\sigma^{ij} + \sigma^{ji} = \frac{1}{2} [\varepsilon_{lm} (C^{lmi j} + C^{lmi j}) + \sigma^{ij} + \sigma^{ji}] \quad (7.102)$$

Using (7.96) and the symmetry of σ^{ij} one obtains:

$$\sigma^{ij} = \varepsilon_{lm} C^{lmi j} , \quad (7.103)$$

and comparison with (7.95) shows that

$$C^{ijlm} = C^{lmi j} . \quad (7.104)$$

7.10 Bridging Scale Application: Crack Propagation in a Brittle Specimen

One landmark simulation was performed in the year 2000 concerning the manner by which materials fail, namely either ductile or brittle, cf. Fig. 7.27. In this simulation, it was shown that there are cracks traveling faster than the speed of sound in a material, which is, of course, not predicted by continuum theory, according to which the limiting speed is the speed of sound in the material. The simulation involved 20 million atoms in a brittle material. Usually, in experiments, it is impossible to see the dynamic development of cracks and fracture in a material. Thus, in this case, the simulation acts as a microscope, determining what happens upon crack formation on the atomic level. In another study using 1 billion atoms, the creation and interaction of hundreds of dislocations could be studied. In ductile failure, metals bend instead of shatter as a result of plastic deformation, which occurs when rows of atoms slide past one another on slip planes (dislocations), cf. also Figs. 7.19 and 7.20. In this study the interaction of moving dislocations in a piece of matter of almost the size $1 (\mu\text{m})^3$ could be analyzed in detail. After a while, dislocations become rigid as they stick to one another. This rigidity causes the material to change from ductile to brittle, a phenomenon also called *work hardening*. This phenomenon had been known on the *macroscopic scale* for a long time, but had never been understood on the atomic level. This kind of atomic large-scale simulations in fact are able to bridge the

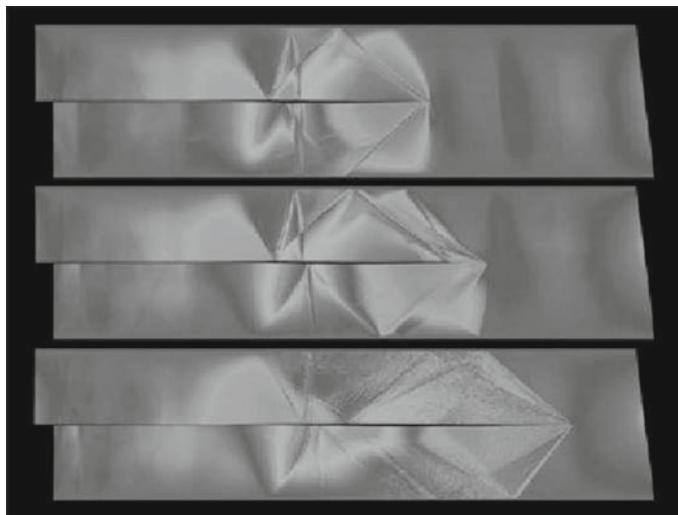


Fig. 7.27 A multimillion-atom simulation to study crack propagation in rapid brittle fracture performed on a 12.3 teraflops ASC White supercomputer. The snapshot pictures represent a progression time (from *top to bottom*) of a crack traveling in a harmonic wave. Taken from [478]. Credit is given to the University of California, Lawrence Livermore National Laboratory, and the Department of Energy under whose auspices the work was performed

gap between the atomic and macroscopic scale, explaining the macroscopic effect of work hardening by dislocations on the atomic scale with a simulated specimen of material that has microscopic dimensions.

Abstract

The research area of computational science and multiscale modeling of materials has not completely come of age yet. The prevalent procedure of multiscale modeling is to find designed solutions for each individual problem to be solved. There are no real standard procedures that can be applied for any model on any scale. This has to do with the structural complexity of systems on different length scales, but also with the artificial human separation into several scales and methods applicable on these scales. Nature itself does not make such a distinction and the theoretical development of methods is thus far from being exhaustive. Perhaps the most interesting progress in multiscale computational science can be expected in the area of biological systems where more and more physicists, computer scientists and mathematicians are entering the field of genomics, drug design, molecular modeling of enzymes, proteins and other molecular structures. What is really needed, is to overcome the traditional boundaries of knowledge and research and their artificial partition into “biology”, “chemistry”, “physics” and so forth. This is necessary, as multiscale computational science truly requires knowledge and experience in different disciplines of research.

In order to truly link mesoscopic or even macroscopic phenomena to a detailed molecular description in light of a missing molecular-based, quantitative theory, requires the bridging of models and simulation techniques across the huge range of length and associated time scales separating the atomic from the macroscopic world. One of the first breakthrough examples of multiscale modeling of materials was the linking of quantum and classical MD method with continuum methods to study crack propagation in silicon by Abraham et al. [41].

The main goal of materials science is the rapid and accurate prediction of properties of new materials before their development and manufacturing. Despite the advances made in modeling of the structural, mechanical, thermal, and transport

properties of materials at the macroscopic level (FEM applied to complicated structures) there still remains considerable uncertainty about how to predict many properties of industrial interest. With the further increase of computational capacities within the next decade, it is to be expected, that many methods which are cutting-edge science today, will become routine applications. Development in several directions will be important or become feasible:

- Development of new, faster algorithms to solve well-known equations.
- New ideas for extending and coupling the time and length scales in simulations.
- Emerging applications in biological (gene analysis, microarrays, biomembranes) and medical systems (e.g. simulations of the interaction of cells with the environment, FEM simulation of organs exerted to pressure waves).
- Fast simulation of equilibrium states of macromolecules in polymer physics and biology.

It is foreseeable that ab initio methods will be extended by at least one order of magnitude, thus enabling to calculate molecular structures of up to several thousand atoms with ab initio based numerical methods. Thus, ab initio methods will start to compete with classical force-field methods on the small length scales. With increasing system size however, the more expensive ab initio procedures will always become prohibitive which means that, in particular for biomolecular and macromolecular applications, the coarse-grained force-field methods will still remain the method of choice. Also the investigation of large molecules, such as proteins and nucleic acids, where simplification have to be introduced to be able to simulate even system for nanoseconds, simplified empirical and half-empirical methods will remain the major modeling methodologies. Quantum mechanical methods have undergone remarkable advances in the past 10 years, allowing for the simulation of systems containing several hundred atoms. Molecular mechanics is faster and a more approximate method for computing the structure of simple biological systems.

An important issue in the near future will be *time-scale* bridging methods. The problem with time is, that it cannot be parallelized, i.e. one cannot split the temporal scale on many processors as can be done with the spatial domain. Therefore, here, new methods will have to be developed to extend the simulated time scales into the experimental regime, e.g. when investigating diffusion or considering biomolecules.

The mathematical treatment of molecular geometries and energies based on a classical description (Molecular Mechanics) is one the oldest known concepts in chemistry and physics. A principal restriction in the application of these methods are the limitations of the procedures for energy minimizations, e.g. by gradient methods. These methods are not capable, starting from an arbitrary configuration, to predict the *absolute*, i.e. the *global* minimum of the energy surface of the system in phase space. Thus, one has to start with many different initial configurations which extends the computational effort in practice.

With the emerging possibilities of ever stronger coupling of different length scales, it becomes quite evident how important it is, not to be restricted in one's modeling and simulation capabilities to just one specific topic, tool, or method. On the smallest scales one deals only with physical theory, then, with molecules, chemistry comes

into play, and with macromolecular aggregates (macromolecules, or polymers) one enters the field of *molecular biology*. Going yet to larger scales, one encounters typical engineering methods which approximate materials as a continuum and solve the evolving PDEs on a grid. The predictability of materials behavior with FEM is impressive, however one may not forget that this method usually does not lead to a *fundamental* understanding, i.e. an understanding that is based on the atomic crystalline structure of all condensed matter. The material models in engineering are obtained as macroscopic averages from ingenious experimental tests under many different variations of load conditions, where the atomic degrees of freedom have all been averaged out. In recent years, therefore, one tries to incorporate into FEM, DEM or SPH the meso- and microstructural features of materials by, e.g. taking into account the granular structure as seen micrographs. Examples for this modern approach to materials modeling were presented in Chap. 7. It is in essence the mesoscale, where several scientific and engineering disciplines meet in an attempt to understand materials behavior. Thus, the increasing capabilities of computer hardware will make it even more necessary for researchers in this field to become acquainted with the basic methods and theoretical constructs of several different research areas such as mathematics, physics, chemistry, biology and engineering. This book wants to contribute to this purpose.

Further Reading

A

General Physics

- J. von Neumann: *Collected Works of J. von Neumann*, Pergamon Press, 1963
E.A. Bender: *An Introduction to Mathematical Modeling*, John Wiley & Sons, 1978
R. Penrose: *The Emperor's New Mind*, Oxford University Press, New York, 1989
H. Genz: *Wie die Naturgesetze Wirklichkeit schaffen. Über Physik und Realität*, Rowolt TB Verlag, Reinbek bei Hamburg, 2004
H. Weyl: *Symmetry*, Princeton University Press, Princeton, NJ, 1952
M.H. Shamos (ed.): *Great Experiments in Physics*, Dover, New York, 1987
P. Yourgrau: *A World Without Time*, Basic Books, New York, 2005
R. Penrose: *The Emperor's New Mind*, Oxford University Press, 1989
D.R. Hofstadter: *Gödel, Escher, Bach: an Eternal Golden Braid*, Basic Books, New York, 1979
R. Feynman: *The Pleasure of Finding Things Out*, Helix Books, Cambridge, MA, 1999
M. Born: *The Born-Einstein Letters 1916-1955*, Macmillan Press Ltd, 2005

Programming Techniques

- T. Ericson: *Computers in Materials Technology*, Pergamon Press, 1981
Meilir Page-Jones: *The Practical guide to Structured Systems Design*, Englewood Cliffs, Yourdon Press NJ, 1988
L.J. Peters: *Handbook of Software Design: Methods and Techniques*, New York, Yourdon Press, 1981
D. King: *Creating Effective Software: Computer Program Design Using the Jackson Methodology*, New York, Yourdon Press, 1988
G. Polya: *How to Solve It: A New Aspect of Mathematical Methods*, 2nd ed, Princeton NJ, Princeton University Press, 1957
N. Wirth: *Algorithms and Data Structures*, Englewood Cliffs, NJ, Prentice Hall, 1986

Journals and Conferences on Multiscale Materials Modeling and Simulation

- International Journal for Multiscale Computational Engineering, Editor-in-Chief: Jacob Fish (Rensselaer Polytechnic Institute)
- International Journal of Multiscale Materials Modeling, Editor-in-Chief: Luo Yun-hua (Univ. of Manitoba)
- Multiscale Modeling and Simulation (SIAM), Editor-in-Chief: Jack Xin (University of California Irvine)
- MMM Conference on Multiscale Materials Modeling (biannual)

Definition 8 (*Zermelo–Fraenkel Axioms*) The Zermelo–Fraenkel Axioms are the basis for *Zermelo–Fraenkel set theory*. The system of axioms 1–8 is called *Zermelo–Fraenkel set theory*, denoted “ZF”. The original publication is [479]. The system of axioms 1–6 plus 8 is called *Zermelo set theory*, often denoted “Z”. The complete set of axioms 1–9 including the axiom of choice, is usually denoted “ZFC”. For a short discussion of the axiom of choice, see Chap. 3. There does not seem to be a general agreement about what axioms constitute “Zermelo set theory”. For example, Enderton [480] includes axioms 8 and 9, but *excludes* axiom 7, while Mendelson [239] includes axiom 7, but excludes axioms 8 and 9. In the following, the symbol “ \forall ” stands for “for all” and “ \equiv ” is short for “is equivalent to”. Taken from [481].

1. Axiom of Extensionality:

If X and Y have the same elements, then $X = Y$, or formally:

$$\forall u (u \in X \equiv u \in Y) \Rightarrow X = Y.$$

2. Axiom of the Unordered Pair (Axiom of Pairing):

For any a and b there exists a set $\{a, b\}$ which contains exactly a and b .

$$\forall a \forall b \exists c \forall x (x \in c \equiv (x = a \text{ or } x = b)).$$

3. Axiom of Subsets (Axiom of Separation):

If ϕ is a property parameterized with p , then for any X and p there exists a set $Y = \{u \in X : \phi(u, p)\}$ that contains all those $u \in X$ which have the property ϕ .

$$\forall X \forall p \exists Y \forall u (u \in Y \equiv (u \in X \text{ and } \phi(u, p))).$$

4. Axiom of the Sum Set (Axiom of Union): For any X there exists a set $Y = \bigcup X$, the union of all elements of X .

$$\forall X \exists Y \forall u (u \in Y \equiv \exists z (z \in X \text{ and } u \in z)).$$

5. Axiom of the Power Set: For any X there exists a set $Y = P(X)$, the set of all subsets of X .

$$\forall X \exists Y \forall u (u \in Y \equiv u \subset X).$$

6. Axiom of Infinity: There exists an infinite set.

$$\forall X (\emptyset \in X \text{ and } (\forall x \in X) (x \cup \{x\} \in X)).$$

7. Axiom of Replacement: If f is a function, then for any X there exists a set $Y = f(X) = \{f(x) : x \in X\}$.
8. Axiom of Foundation (Axiom of Regularity): Every non-empty set has an ε -minimal element.
 $\forall X (X \neq \emptyset \Rightarrow (\exists x \in X) S \cap x = \emptyset)$.
9. Axiom of Choice: Every family of non-empty sets has a choice function A .
 $\forall X \in a \exists A(x, y) \Rightarrow \exists y \forall x \in a A(x, y(x))$.

Sample Code for the Main Routine of a MD Simulation

C

The algorithm presented here shows several things. First of all, it shows structured coding. The declarations of functions are put into header (.h) files which are included into (.c)-files. Thus, the *declaration* and *implementation* of functions is separated and the functions can easily be used via include statements in any other module (.c)-file where needed. Code should be documented. In the sample code below, this is done in a way that works with the freely available tool doxygen,¹ which is a parser that can generate automatically documented code in latex, html and some other formats. The particle and parameter information is declared in the module “defs.h” which has to be included in any file, that needs to have information on particle properties or the parameters, that control the flow of the simulation. These parameters along with the particle properties are conveniently stored in C-structures, as shown in the sample code of Algorithm 6.4 on p. 285.

```
1      /** \file main.c
2      \brief A molecular dynamics code for the simulation
3      of a simple LJ fluid
4      \version 1.0
5      \author Author's name
6      \bug Known bugs
7      \todo Various todos
8      */
9      /* Standard headers */
10     #include <math.h>
```

¹<http://www.stack.nl/~dimitri/doxygen/>.

Algorithm C.1: Sample code for the main routine in a MD program

```
11     #include <stdio.h>
12
13     /* Personal headers */
14     #include "Defs.h"
15     #include "Init.h"
16     #include "Forces.h"
17     #include "Integrate.h"
18     #include "Io.h"
19
20     /** \fn int main(int argc, char *argv[])
21      \brief The main function
22      \param argc
23      \param *argv[]
24      \brief Standard input argument to main.
25      */
26     int main(int argc, char *argv[])
27     {
28         int MDCycles = 1;
29         /* Declare local variables here */
30
31         Init(particle, parameters);
32         while (MDCycles)
33         {
34             ForceCalc(particle, parameters);
35             Integrate(particle, parameters);
36             /* Calculate Observables and write data files */
37             if (!(MDCycles % IntervalObservables))
38                 CalcObservables(particle, parameters);
39             if ((MDCycles++) < parameters -> MDcycleMax)
40                 MDCycles = 0;
41             }
42             CalcObservables(particle, parameters);
43             return(0);
44     }
```

In any Unix — and Linux – based programming environment makefiles are commonly used to ease handling of administrative tasks when compiling the code sources. Here, a simple makefile is provided as a template for compiling the program sources of a simple MD code as presented and discussed in Chap. 6.

Algorithm D.1: Sample makefile for convenient handling of compilation

```
1 # -----
2 # Makefile sample for a simple program
3 # -----
4 # In this example use the GNU compiler
5 CC = gcc
6
7 # Compiler options
8 CFLAGS = -Wall -pedantic -O2
9
10 # Link math library
11 LIBS = -lm
12
13 # file names and dependencies
14 SRC = main.c init.c forces.c integrate.c io.c
15 OBJ = main.o init.o forces.o integrate.o io.o
16 BIN = MDcode
17
18 # command abbreviation
19 RM = rm -f
20
21 # this gets executed first of all
22 $(BIN): $(OBJ)
23     @echo Linking files...
24     @$(CC) $^ -o $@ $(LIBS)
25
26 # further dependencies
27 main.o : main.c init.h forces.h integrate.h io.h
28     $(CC) $(CFLAGS) -c $<
```

Algorithm D.2: Sample makefile for convenient handling of compilation

```
31 init.o: init.c init.h random.h
32     $(CC) $(CFLAGS) -c $<
33 forces.o: forces.c forces.h
34     $(CC) $(CFLAGS) -c $<
35 integrate.o: integrate.c integrate.h random.h forces.h
36     $(CC) $(CFLAGS) -c $<
37 io.o: io.c io.h
38     $(CC) $(CFLAGS) -c $<
39
40 .PHONY: install
41 install: $(BIN)
42     @echo installing simulation program to
43     @prefixed directory....
44     @install -s $(BIN) $(HOME)/mybin
45     @$(RM) $(BIN)
46     @echo Program '$(BIN)' installed.
```

Tables of Physical Constants

E

Gravitation constant	G	$6.627 \times 10^{-11} (N \cdot m^2/kg^2)$
Elementary charge	e	$1.602 \times 10^{-19} (C)$
Velocity of light in vacuum	c	$2.998 \times 10^8 (m/s)$
Boltzmann constant	k_B	$1.381 \times 10^{-23} (J/K)$
Avogadro constant	N_0	$6.022 \times 10^{-23} (\text{atoms/mol})$
Gas constant	$R = N_0 k_B$	$8.314 (J/mol \cdot K)$
Planck constant	h	$6.626 \times 10^{-34} (J \cdot s)$
	\hbar	$1.054 \times 10^{-34} (J \cdot s)$
Permittivity of empty space	$\epsilon_0 = 1/\mu_0 c^2$	$8.854 \times 10^{-12} (A \cdot s/v \cdot m)$
Permeability of empty space	μ_0	$4\pi \times 10^{-7} (H/m) = 1.257 \times 10^{-6} (V \cdot s/A \cdot m)$
Faraday constant	F	$9.648 \times 10^4 (C/mol)$
Mass of electron	m_{e^-}	$9.109 \times 10^{-31} (kg)$
Mass of proton	m_p	$1.673 \times 10^{-27} (kg)$
Bohr radius	a_0	$5.292 \times 10^{-11} (m)$

E.1 International System of Units (SI or mksA System)

In the SI unit system, there are four base units, the meter, the kilogram, the second, and the ampère are *defined*. Further base units are the Kelvin, the mole (for the amount of substance), and the candela (for the luminous intensity). All other units are *derived units* as shown in the listing below. Other systems are still in use as well. In theoretical physics for example, the use of cgs units (Gaussian system) is quite common.

Quantity	Name	Symbol	Base Units
Base Units			
Length	Meter	<i>m</i>	—
Mass	Kilogram	<i>kg</i>	—
Time	Second	<i>s</i>	—
Derived Units			
Force	Newton	<i>N</i>	<i>kg · m/s</i> ²
Energy, work	Joule	<i>J</i>	<i>kg · m</i> ² / <i>s</i> ²
Pressure	<i>c</i>	$2.998 \times 10^8 \text{ (m/s)}$	
Bohr radius	<i>a</i> ₀	$5.292 \times 10^{-11} \text{ (m)}$	

E.2 Conversion Factors of Energy

The following table contains several unit systems, commonly used in quantum chemistry and atomic simulations

Unit	<i>eV</i>	<i>hartree</i>	<i>kJ mol</i> ⁻¹	<i>kcal mol</i> ⁻¹	<i>cm</i> ⁻¹
<i>eV</i>	1	$3.649 \cdot 10^{-2}$	96.485	23.061	8065.5
<i>hartree</i>	27.212	1	2625.50	627.51	$2.1947 \cdot 10^5$
<i>kJ mol</i> ⁻¹	$1.0364 \cdot 10^{-2}$	$3.8088 \cdot 10^{-4}$	1	0.239011	83.593
<i>kcal mol</i> ⁻¹	$4.3363 \cdot 10^{-2}$	$1.5936 \cdot 10^{-3}$	4.1840	1	349.75
<i>cm</i> ⁻¹	$1.2398 \cdot 10^{-4}$	$4.5563 \cdot 10^{-6}$	$1.1963 \cdot 10^{-2}$	$2.8591 \cdot 10^{-3}$	1

Solutions

Problems of Chap. 2

2 Proper Time $d\tau$

In a new coordinate system $x^{\alpha'}$, the coordinate differentials of (2.8) are given as

$$dx^{\alpha'} = \Lambda_{\beta}^{\alpha} dx^{\beta}, \quad (5.1)$$

so the new coordinate time is:

$$d\tau'^2 = \eta_{\alpha\beta} dx^{\alpha'} dx^{\beta'} = \eta_{\alpha\beta} \Lambda_{\gamma}^{\alpha} \lambda_{\delta}^{\beta} dx^{\gamma} dx^{\delta} = \eta_{\gamma\delta} dx^{\gamma} dx^{\delta}. \quad (5.2)$$

Thus,

$$d\tau'^2 = d\tau^2. \quad (5.3)$$

2 Conservation Laws

- (a) Myon and electron lepton number
- (b) Electron lepton number
- (c) Energy

3 Euler–Lagrange Equations

The variation of the integral of action (2.44) taking into account (2.42) is:

$$\delta S = \int_{t_1}^{t_2} dt \sum_i \left[\frac{\partial L}{\partial q_i} \delta q_i(t) + \frac{\partial L}{\partial \dot{q}_i}(t) \delta \dot{q}_i(t) \right]. \quad (5.4)$$

Integration of the second term by parts yields with $\delta \dot{q}_i(t) = \frac{d}{dt} \delta q_i(t)$ yields

$$\int_{t_1}^{t_2} dt \frac{\partial L}{\partial \dot{q}_i} \cdot \frac{d}{dt} \delta q_i = \frac{\partial L}{\partial \dot{q}_i} \Big|_{t_1}^{t_2} - \int_{t_1}^{t_2} dt \left(\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} \right) \delta q_i = - \int_{t_1}^{t_2} dt \left(\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} \right) \delta q_i. \quad (5.5)$$

The right-hand side of (5.5) vanishes due to the boundary conditions $\delta q_i(t_1) = \delta q_i(t_2) = 0$; hence, one obtains

$$\delta S = \int_{t_1}^{t_2} dt \sum_i \left(\frac{\partial L}{\partial q_i} \right) - \frac{\partial L}{\partial q_i} \delta q_i = 0, \quad (5.6)$$

which equals 0, if the Euler–Lagrange equations are valid.

5 Abstract Data Types: LIFO Structure

We declare the class “Stack” in a header “Stack.h” in C++ as follows:

```
#ifndef _STACK_H
#define _STACK_H

class Stack{
private:
    int *someData;
    unsigned long number;
    unsigned long maxNumber;

public:
    Stack ( unsigned long );
    ~Stack(void);

    int Push (int);
    int Pop (void);
    unsigned long Size (void);
};

#endif /* _STACK_H */
```

The class declaration is done within preprocessor statements. If “_STACK_H” has not yet been defined, then it is defined and the class declaration is processed. The commentary in the last line is written using C-notation (instead of “//”) as many C++ compilers use a C-preprocessor for parsing preprocessing commands. These do not accept C++-comments.

Next we provide the file “Stack.cpp” which implements the functionality of a stack data structure.

```
#include ``Stack.h''\n\nStack::Stack ( unsigned long size){\n\n    data = new ( int[size] );\n    if ( data ){\n        number      = 0;\n        maxNumber  = size;\n    }\n    else{\n        number = maxNumber = 0;\n    }\n}\n\nStack::~Stack (void){\n    if ( data ) delete[]( data );\n}\n\nint Stack::Push (int w){\n    if ( number < maxNumber ){\n        data[number++] = w;\n        return (1);\n    }\n    else{\n        return (0);\n    }\n}\n\nint Stack::Pop (void){\n    if ( number > 0 )\n        return (data[--number]);\n    else\n        return (0);\n}\n\nunsigned long Stack::Size (void){\n    return (number);\n}
```

The class definition first includes the class declaration and then implements the push and pop methods. Additionally, a method “size” is implemented, which returns

the current number of saved data elements on the stack. These methods are self-explanatory.

Last, a main function “Main.cpp” which tests the stack is implemented as follows:

```
#include <iostream.h>
#include <string.h>
#include ``Stack.h``

void main (void){
    const unsigned long SIZE = 100;
    Stack stack (SIZE);
    char str[SIZE];
    unsigned int x;

    cout << ``Please input some string.``;
    cin.getline (str, SIZE);
    for (x = 0; x < strlen (str); x++) stack.Push (str[x]);
    cout << ``\n``;
    while ( stack.Size () ) cout << (char)(stack.Pop());
    cout << ``\n``;
}
```

Main.cpp includes the declaration of Stack from Stack.h. The advantage of this modular coding structure is the separation of the main function from the class definition. Should the definition be changed, no new compilation of Main.cpp is necessary. The explicit cast operator for the return value of *Pop()* is necessary as the stack processes *int*-values and *cout* would otherwise print them as integers.

6 A Java Program to Calculate the Ackermann Function

We provide a recursive implementation of the Ackermann function in Java as follows:

```
class ackermann{
    public static long ackermann( long n, long m ){
        if ( n == 0 ) return (m + 1);
        else
            if ( m == 0 )    return ackermann ( n - 1, 1 );
            else           return ackermann ( n - 1, ackermann (n, m - 1) )
    }

    public static void main ( String args[] ){
        int x = 2,
            y = 2;
        System.out.println( ``ackermann(`` + x + ``, `` + y + ``)` = `` \\
            + ackermann (x,y) );
```

}

}

The binary code will calculate $A(5, 0)$ in roughly about 30 seconds to a few minutes depending on the computer hardware. $A(5, 1)$ already surpasses all computational capabilities of a modern PC, workstation or any computer system available.

Problems of Chap. 3

3.13 Levi-Civita Tensor

In a moved system Σ' one has:

$$\varepsilon'_{abcd} = \Lambda_a^e \Lambda_b^f \Lambda_d^g \Lambda_e^h \varepsilon_{efgh}. \quad (5.7)$$

Because ε is only nonzero, if the numbers $(abcd)$ are a permutation of the numbers (0123) , the above equation may be written as:

$$\varepsilon'_{abcd} = \sum_P (-)^P \Lambda_a^P(0) \Lambda_b^P(1) \Lambda_d^P(2) \Lambda_e^{(3)} \varepsilon_{efgh}. \quad (5.8)$$

This is exactly the formula for calculating a determinant. Thus, choosing the velocity in x -direction $v = v\mathbf{e}_x$, one can rewrite (5.8) as

$$\varepsilon'_{0123} = \text{Det}(\Lambda(v\mathbf{e}_x)) = \left| \begin{pmatrix} -\frac{\gamma}{c}\gamma & -\frac{v}{c}\gamma & 0 & 0 \\ -\frac{v}{c}\gamma & \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \right| = \gamma^2 - \left(-\frac{v}{c}\gamma \right)^2 = \gamma^2 \left(1 - \left(\frac{v}{c} \right)^2 \right) = 1. \quad (5.9)$$

As a result, one obtains for every even (uneven) permutation of $(abcd)$ $\varepsilon'_{abcd} = +1$ (-1), since the sign of the determinant is changed when exchanging two rows. Hence,

$$\varepsilon'_{abcd} = \varepsilon_{abcd} . \quad (5.10)$$

■

3.8 Christoffel Symbols

1. $g_{ab} = \text{diag}(1, 1, 1)$, $g^{ab} = \text{diag}(1, 1, 1)$, $g = 1$
2. $g_{ab} = \text{diag}(1, r^2, 1)$, $g^{ab} = \text{diag}(1, r^{-2}, 1)$, $g = r^2 \Gamma_{r\theta}^\theta = \Gamma_{\theta r}^\theta 1/r$, $\Gamma_{\theta\theta}^r = -r$.
3. $g_{ab} = \text{diag}(1, r^2, r^2 \sin^2 \theta)$, $g^{ab} = \text{diag}(1, -r^{-2}, -r^{-2} \sin^{-2} \theta)$, $g = r^4 \sin^2 \theta$
 $\Gamma_{\theta\theta}^r = -r$, $\Gamma_{\theta\theta}^r = -r \sin^2 \theta$, $\Gamma_{r\theta}^\theta = \Gamma_{\theta r}^\theta = 1/r$, $\Gamma_{\phi\phi}^\theta = -\sin \theta \cos \theta$, $\Gamma_{r\phi}^\phi = \Gamma_{\phi r}^\phi = 1/r$, $\Gamma_{\theta\phi}^\phi = \Gamma_{\phi\theta}^\phi = \cot \theta$.

3.11 Minkowski Coordinates

1. -2
2. yes
3. yes

Problems of Chap. 4

4.1 Solution of the Wave Equation

The proof is easy.

First, one sets $u_{\pm} := x \pm ct$.

Thus,

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial u} \frac{\partial u}{\partial x} = \frac{\partial f}{\partial u} = f' \Rightarrow \frac{\partial^2 f}{\partial x^2} = f''.$$

For the derivative with respect to time one obtains

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial u} \frac{\partial u}{\partial t} = \pm c \frac{\partial f}{\partial u} = \pm c f' \Rightarrow \frac{\partial^2 f}{\partial t^2} = c^2 \ddot{f}.$$

4.3 Verlet Algorithm

The standard derivation of the Verlet algorithm is based on a Taylor expansion of the coordinates r as follows:

$$r_{t+\Delta t} = r_t + v_t + \frac{1}{2} \Delta t f_t + \mathcal{O}(\Delta t^3), \quad (5.11)$$

$$r_{t-\Delta t} = r_t + v_t - \frac{1}{2} \Delta t f_t + \mathcal{O}(\Delta t^3), \quad (5.12)$$

Addition of the above equations yields:

$$r_{t+\Delta t} + r_{t-\Delta t} = 2r_t + (\Delta t)^2 f_t + \mathcal{O}((\Delta t)^4), \quad (5.13)$$

or

$$r_{t+\Delta t} = 2r_{t-\Delta t} - r_{t-\Delta t} ((\Delta t)^2) f_t + \mathcal{O}((\Delta t)^4), \quad (5.14)$$

This is the basic equation of the Verlet algorithm, which has the disadvantage, that no velocities are explicitly calculated, but which are needed for computing the kinetic energy of a system. Substituting t by $t + \Delta t$ in (5.12) one obtains:

$$r_t = r_{t+\Delta t} - \Delta t v_{t+\Delta t} + \frac{(\Delta t)^2}{2} a_{t+\Delta t} + \mathcal{O}((\Delta t)^3). \quad (5.15)$$

Adding (5.11) and (5.15) results in:

$$\begin{aligned} r_{t+\Delta t} + r_t &= r_t + r_{t+\Delta t} + \Delta t v_t - \Delta t v_{t+\Delta t} + \\ &+ \frac{(\Delta t)^2}{2} a_t + \frac{(\Delta t)^2}{2} f_{t+\Delta t} + \mathcal{O}((\Delta t)^3). \end{aligned} \quad (5.16a)$$

Resolving this equation for $v_{t+\Delta t}$ yields:

$$v_{t+\Delta t} + \underbrace{v_t + \frac{(\Delta t)^2}{2} a_t}_{\approx v_{t+\frac{\Delta t}{2}}} + \frac{(\Delta t)^2}{2} f_{t+\Delta t} + \mathcal{O}((\Delta t)^2) . \quad (5.17)$$

Inserting this expression into (5.11) and (5.17) yields the velocity-Verlet algorithm:

$$r_{t+\Delta t} = r_t + \Delta t v_{t+\frac{\Delta t}{2}} + \mathcal{O}(\Delta t^3) , \quad (5.18a)$$

$$v_{t+\Delta t} = v_{t+\frac{\Delta t}{2}} + \frac{1}{2} a_{t+\Delta t} + \mathcal{O}(\Delta t^2) . \quad (5.18b)$$

References

1. Newton, I. (1999). *Principia: mathematical principles of natural philosophy*. Berkeley: University of California Press.
2. Planck, M. (1900). Über irreversible Strahlungsvorgänge. *Annalen der Physik*, 306(1), 69–122.
3. Planck, M. (1900). Zur Theorie des Gesetzes der Energieverteilung im Normalspectrum. *Verhandlungen der Deutschen Physikalischen Gesellschaft*, 2, 237–245.
4. Steinhauser, M. O. (2016). *Quantenmechanik für Naturwissenschaftler*. Berlin: Springer.
5. Cabibbo, N., Iwasaki, Y., & Schilling, K. (1999). High performance computing in lattice QCD. *Parallel Computing*, 25, 1197–1198.
6. Evertz, H. G. (2003). The loop algorithm. *Advances in Physics*, 52, 1–66.
7. Holm, E. A., & Battaile, C. C. (2001). The computer simulation of microstructural evolution. *The Journal of Minerals*, 53(9), 20–23.
8. Kschischio, M., Kern, R. R., Gieger, C. C., Steinhauser, M. O., & Tolle, R. (2005). Automatic scoring and quality assessment using accuracy bounds for FP-TDI SNP genotyping data. *Applied Bioinformatics*, 4, 75–84.
9. Cleri, F., Yip, S., Wolf, D., & Phillpot, S. R. (1997). Atomic-scale mechanism of crack-tip plasticity: Dislocation nucleation and crack-tip shielding. *Physical Review Letters*, 79, 1309–1312.
10. Abraham, F. F., Walkup, R., Gao, H., Duchaineau, M., De La Rubia, T. D., & Seager, M. (2002). Simulating materials failure by using up to one billion atoms and the world's fastest computer: Work-hardening. *Proceedings of the National Academy of Sciences*, 99(9), 5783–5787.
11. Komanduri, R., Chandrasekaran, N., & Raff, L. M. (2003). Molecular dynamic simulations of uniaxial tension at nanoscale of semiconductor materials for micro-electro-mechanical systems (MEMS) applications. *Materials Science and Engineering: A*, 340(1–2), 58–67.
12. Park, D. (1992). *Introduction to quantum theory* (3rd ed.). International series in pure and applied physics. New York: McGraw-Hill.
13. Ashcroft, N. W., & Mermin, N. D. (1976). *Solid state physics*. Philadelphia: W.B Saunders Company.
14. Cho, A. (2007). No twisting out of Newton's law. *ScienceNow Daily News*.
15. Abramovici, A., & Vager, Z. (1986). Test of Newton's second law at small accelerations. *Physical Review D*, 34, 3240–3241.

16. Pauli, W. (1961). Zur älteren und neueren Geschichte des Neutrinos. *Aufsätze und Vorträge über Physik und Erkenntnistheorie* (pp. 156–180). Wiesbaden: Vieweg and Teubner.
17. Reines, F., & Cowan, C. L. (1956). The Neutrino. *Nature*, *178*, 446–449.
18. Anderson, H. L. (1986). Scientific uses of the MANIAC. *Journal of Statistical Physics*, *43*(5), 731–748.
19. Burks, A. W., Goldstine, H. H., & von Neumann, J. (1946). *Preliminary discussion of the logical design of an electronic computing instrument*. Institute for Advanced Studies, Princeton, N.J.: Technical report.
20. Fermi, E., Metropolis, N., & Alei, E. F. (1954). Phase shift analysis of the scattering of negative pions by hydrogen. *Physical Review*, *95*, 1581–1585.
21. Hoffmann, F. D., Metropolis, N., Alei, E. F., & Bethe, H. A. (1954). Pion-hydrogen phase shift analysis between 120 and 217 Mev. *Physical Review*, *95*, 1586–1605.
22. Fermi, E., Pasta, J. R., & Ulam, S. M. (1955). *Studies of nonlinear problems*. Los Alamos National Laboratory, Los Alamos, NM, Los Alamos, NM: Technical report.
23. Gamow, G. (1954). Possible relation between deoxyribonucleic acid and protein structures. *Nature*, *173*, 318–318.
24. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*, 1087–1092.
25. Metropolis, N., & Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Association*, *44*, 335–341.
26. Dongarra, J., & Sullivan, F. (2000). Guest editors' introduction: The top 10 algorithms. *Computing in Science & Engineering*, *2*(1), 22–23.
27. Alder, B. J., & Wainwright, T. (1958). Molecular dynamics by electronic computers. In I. Prigogine (Ed.), *Proceedings of the International Symposium on Transport Processes in Statistical Mechanics, Brussels, 1956* (pp. 97–131). New York: Interscience Publishers Inc.
28. Wood, W. W., & Jacobson, J. D. (1957). Preliminary results from a recalculation of the Monte Carlo equation of state of hard spheres. *The Journal of Chemical Physics*, *27*(5), 1207–1208.
29. Alder, B. J., & Wainwright, T. E. (1957). Phase transition for a hard sphere system. *The Journal of Chemical Physics*, *27*(5), 1208–1209.
30. Rahman, A. (1964). Correlations in the motion of atoms in liquid argon. *Physical Review*, *136*, 405–411.
31. Kernighan, B. W., & Ritchie, D. M. (1978). *The C programming language*. Englewood Cliffs: Prentice Hall.
32. Parker, A. (2007). From sound waves to stars. *S&TR* (pp. 26–27).
33. Steinhauser, M. O., & Hiermaier, S. (2009). A review of computational methods in materials science: Examples from shock-wave and polymer physics. *International Journal of Molecular Sciences*, *10*, 5135–5216.
34. Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, *82*, 1013–1024.
35. Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, *19*(90), 297–301.
36. Greengard, L., & Rokhlin, V. (1987). A fast algorithm for particle simulations. *Journal of Computational Physics*, *73*(2), 325–348.
37. Verlet, L. (1967). Computer “experiments” on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Physical Review*, *159*(1), 98–103.
38. Needleman, A. (1988). Material rate dependence and mesh sensitivity in localization problems. *Computer Methods in Applied Mechanics and Engineering*, *67*(1), 69–85.
39. Elston, R. C., & Stewart, J. (1971). A general model for the genetic analysis of pedigree data. *Human Heredity*, *21*, 523–542.
40. Foster, I., & Kesselman, C. (1998). *The grid: Blueprint for a new computing infrastructure*. Technical report: Morgan Kaufmann.

41. Abraham, F. F., Brodbeck, D., Rafey, R., & Rudge, W. (1994). Instability dynamics of fracture: A computer simulation investigation. *Physical Review Letters*, 73, 272–275.
42. Abraham, F. F., Broughton, J. Q., Broughton, J. Q., Bernstein, N., Kaxiras, E., & Kaxiras, E. (1998). Spanning the length scales in dynamic simulation. *Computers in Physics*, 12(6), 538–546.
43. Holian, B. L., & Lomdahl, P. S. (1998). Plasticity induced by shock waves in nonequilibrium molecular-dynamics simulations. *Science*, 280(5372), 2085–2088.
44. Holian, B. L. (2004). Molecular dynamics comes of age for shockwave research. *Shock Waves*, 13(6), 489–495.
45. Buehler, M., Hartmaier, A., Gao, H., Duchaineau, M., & Abraham, F. (2004). Atomic plasticity: Description and analysis of a one-billion atom simulation of ductile materials failure. *Computer Methods in Applied Mechanics and Engineering*, 193, 5257–5282.
46. Kadau, K., Germann, T. C., & Lomdahl, P. S. (2004). Large-scale molecular-dynamics simulation of 19 billion particles. *International Journal of Modern Physics C*, 15, 193–201.
47. Langer, J. (1999). Computing in physics: Are we taking it too seriously? Or not seriously enough? *Physics Today*, 52(7), 11.
48. Hockney, R. W., & Eastwood, J. W. (1981). *Computer simulation using particles*. New York: McGraw-Hill.
49. Allen, M. P., & Tildesley, D. J. (1987). *Computer simulation of liquids*. New York: Oxford University Press.
50. Ciccotti, G., Frenkel, G., & McDonald, I. R. (1987). *Simulation of liquids and solids*. Amsterdam: North-Holland.
51. Hoover, W. G. (1986). *Molecular dynamics* (Vol. 17). Berlin: Springer.
52. Heermann, D. W. (1990). *Computer simulation methods in theoretical physics*. Berlin: Springer.
53. Haile, J. M. (1992). *Molecular dynamics simulation: Elementary methods*. New York: Wiley.
54. Rappaport, D. C. (1995). *The art of molecular dynamics simulation*. Cambridge, UK: Cambridge University Press.
55. Raabe, D. (1998). *Computational materials science*. New York: Wiley-VCH.
56. Landau, D. P., Landau, D. P., & Binder, K. (2000). *A guide to Monte Carlo simulations in statistical physics*. Cambridge: Cambridge University Press.
57. Frenkel, D., & Smit, B. (2002). *Understanding molecular simulation: From algorithms to applications*. New York: Academic Press.
58. Ray, P. (2002). *Applying molecular and materials modeling*. Berlin: Springer.
59. Engquist, P., Löstedt, P., & Runborg, O. (2005). *Multiscale methods in science and engineering*. New York: Springer.
60. Yip, S. (Ed.). (2005). *Handbook of materials modeling*. Berlin: Springer.
61. Cox, B. N., Gao, H., Gross, D., & Rittel, D. (2005). Modern topics and challenges in dynamic fracture. *Journal of the Mechanics and Physics of Solids*, 53, 565–596.
62. Steinhauser, M. O. (2006). Computational methods in polymer physics. *Recent Research Developments in Physics*, 7, 59–97.
63. Li, S., & Liu, W. K. (2002). Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1), 1–34.
64. Gates, T. S., Odegard, G. M., Frankland, S. J. V., & Clancy, T. C. (2005). Computational materials: Multi-scale modeling and simulation of nanostructured materials. *Composites Science and Technology*, 65, 2416–2434.
65. Demongeot, J., Bezy-Wendling, J., Mattes, J., Haigron, P., Glade, N., & Coatrieux, J. L. (2003). Multiscale modeling and imaging: The challenges of biocomplexity. *Proceedings of the IEEE*, 91(10), 1723–1737.
66. Lépinoux, J. (2000). *Multiscale phenomena in plasticity: From experiments to phenomenology, modeling and materials engineering*. New York: Springer.

67. Chuang, T. J., & Rudnicki, J. W. (Eds.). (2001). *Multiscale deformation and fracture in materials and structures*. New York: Springer.
68. Owen, D. R., & Del Piero, G. (2004). *Multiscale modeling in continuum mechanics and structured deformations*. New York: Springer.
69. Alt, W., Chaplain, M., Griebel, M., & Lenz, J. (2012). *Polymer and cell dynamics: Multiscale modelling and numerical simulations*. Basel: Springer Basel AG.
70. Pojman, L. P. (1998). *Classics of philosophy*. Oxford: Oxford University Press.
71. Vlastos, G. (1993). *The presocratics* (Vol. 1). Studies in Greek philosophy. Princeton, NY: Princeton University Press.
72. Dalton, J. (1808). *A new system of chemical philosophy, part I*. Manchester.
73. Smith, R. A. (2005). *Memoir of John Dalton, and history of the atomic theory up to his time*. Elibron Classics: Adamant Media Corporation.
74. Einstein, A. (1905). Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 17, 549–560.
75. Feynman, R. P. (1949). Space-time approach to quantum electrodynamics. *Physical Review*, 76(6), 769–789.
76. Schilpp, P. A. (Ed.). (1951). *Albert Einstein-philosopher-scientist* (Vol. 42). Evanston, IL: Isis.
77. Gross, D. J., & Wilczek, F. (1973). Ultraviolet behavior of non-Abelian gauge theories. *Physical Review Letters*, 30, 1343–1346.
78. Gross, D. J., & Wilczek, F. (1973). Asymptotically free gauge theories. I. *Physical Review D*, 8, 3633–3652.
79. Politzer, H. D. (1973). Reliable perturbative results for strong interactions. *Physical Review Letters*, 30(26), 1346–1349.
80. Rutherford, E. (1911). The scattering of alpha and beta particles by matter and the structure of the atom. *Philosophical Magazine*, 21, 669–688.
81. Heisenberg, W. (1973). *Der Teil und das Ganze: Gespräche im Umkreis der Atomphysik*. München: Deutscher Taschenbuch Verlag.
82. Einstein, A. (1916). *Ernst Mach. Physikalische Zeitschrift*, 17, 101–104.
83. Lakes, R. (1993). Materials with structural hierarchy. *Nature*, 361(6412), 511–515.
84. Qing, L. (1995). *Zur Frühgeschichte des Elektronenmikroskops*. Stuttgart: GNT Verlag.
85. Heidegger, M. (2005). *The essence of truth: On Plato's cave allegory and theaetetus*. Continuum impacts. Continuum International Publishing Group (new ed.).
86. Drake, S. (1967). *Dialogue concerning the two chief world systems*. Ptolemaic and Copernican. Modern library science. Berkeley: University of California Press.
87. Einstein, A., & Solovine, M. (1993). *Letters to solovine*. New York: Citadel Press.
88. Einstein, A. (1954). *Ideas and opinions*. New York: Bonanza Books.
89. Einstein, A. (1934). On the method of theoretical physics. *Philosophy of Science*, 1(2), 163–169.
90. Mach, E. (1921). *Die Mechanik in ihrer Entwicklung - Historisch kritisch dargestellt* (8th ed.). Leipzig: Brockhaus.
91. Weinberg, S. (1993). *Dreams of a final theory*. New York: Pantheon Books.
92. Hoddeson, L., Brown, L., Riordan, M., & Dresden, M. (Eds.). (1997). *The rise of the standard model particle physics in the 1960s and 1970s*. New York: Cambridge University Press.
93. Pais, A. (1982). *Subtle is the Lord. The science and life of Albert Einstein*. Oxford: Oxford University Press.
94. Einstein, A. (1919). *Time, space, and gravitation*. The Times, London: Technical report.
95. Maxwell, J. C. (1865). A dynamical theory of the electromagnetic field. *Philosophical Transactions of the Royal Society of London*, 155, 459–512.
96. Weinberg, S. (1972). *Gravitation and cosmology*. New York: Wiley.
97. Holian, B. L. (2003). Formulating mesodynamics for polycrystalline materials. *Europhysics Letters*, 64, 330–336.

98. Metropolis, N. (1987). *The beginning of the Monte Carlo method*. Los Alamos: Los Alamos Science.
99. Binder, K. (1992). *Monte Carlo methods in condensed matter physics. Topics in Applied Physics*. Berlin: Springer.
100. Binder, K. (1997). Applications of Monte Carlo methods to statistical physics. *Reports on Progress in Physics*, 60, 487–559.
101. Rollett, A. D., Srolovitz, D. J., Anderson, M. P., & Doherty, R. D. (1992). Computer simulation of recrystallization-III. Influence of a dispersion of fine particles. *Acta Metallurgica et Materialia*, 40(12), 3475–3495.
102. Moldovan, D., Wolf, D. E., Phillpot, S. R., & Haslam, A. J. (2002). Mesoscopic simulation of two-dimensional grain growth with anisotropic grain-boundary properties. *Philosophical Magazine A*, 82, 1271–1297.
103. Kobayashi, M., Takayama, Y., & Kato, H. (2001). Prediction of microstructural evolution during grain growth of a pure aluminum by means of Monte Carlo simulations. *Materials Transactions*, 42(11), 2307–2315.
104. Miodownik, M. A., Godfrey, A. W., Holm, E. A., & Hughes, D. A. (1999). On boundary misorientation distribution functions and how to incorporate them into three-dimensional models of microstructural evolution. *Acta Materialia*, 47(9), 2661–2668.
105. Holm, E. (2001). On misorientation distribution evolution during anisotropic grain growth. *Acta Materialia*, 49, 2981–2991.
106. Chan, J. W., & Hilliard, J. E. (1958). Free energy of a nonuniform system. I. Interfacial free energy. *The Journal of Chemical Physics*, 28(2), 258–267.
107. Cleri, F. (2000). A stochastic grain growth model based on a variational principle for dissipative systems. *Physica A: Statistical Mechanics and its Applications*, 282, 339–354.
108. Miodownik, M. A., & Holm, E. A. (2001). Computer simulations of abnormal subgrain growth. In G. Gottstein & D. A. Moldov (Eds.), *Recrystallization and grain growth* (pp. 309–314). New York: Springer.
109. Potts, R. B., & Domb, C. (1952). Some generalized order-disorder transformations. *Mathematical Proceedings of the Cambridge*, 48(01), 106–109.
110. Hassold, G. N., Chen, I. W., & Srolovitz, D. J. (1990). Computer simulation of final-stage sintering: I, Model kinetics, and microstructure. *Journal of the American Chemical Society*, 73, 2857–2864.
111. Srolovitz, D. J., Grest, G. S., & Anderson, M. P. (1985). Computer simulation of grain growth-V. Abnormal grain growth. *Acta Metallurgica*, 33(12), 2233–2247.
112. Grest, G. S., Anderson, M. P., & Srolovitz, D. J. (1988). Domain-growth kinetics for the Q-state Potts model in two and three dimensions. *Physical Review B*, 38, 4752–4760.
113. Saito, Y. (1997). The Monte Carlo simulation of microstructural evolution in metals. *Materials Science and Engineering: A*, 223(1–2), 114–124.
114. Nordbakke, M. W., Ryum, N., & Hunderi, O. (2002). Invariant distributions and stationary correlation functions of simulated grain growth processes. *Acta Materialia*, 50(14), 3661–3670.
115. Kunaver, U., & Kolar, D. (1998). Three-dimensional computer simulation of anisotropic grain growth in ceramics. *Acta materialia*, 46(13), 4629–4640.
116. Blumenthal, O. (Ed.). (1913). *Das Relativitätsprinzip - Eine Sammlung von Abhandlungen. Fortschritte der Mathematischen Wissenschaften in Monografien*. Leipzig, Berlin: B.G. Teubner.
117. Einstein, A. (1905). Zur Elektrodynamik bewegter Körper. *Annalen der Physik*, 17, 891–921.
118. Einstein, A. (1905). Ist die Trägheit eines Körpers von seinem Energieinhalt abhängig? *Annalen der Physik*, 323, 639–641.
119. Einstein, A. (1906). Das Prinzip von der Erhaltung der Schwerpunktsbewegung und die Trägheit der Energie. *Annalen der Physik*, 325, 627–633.
120. Pauli, W. (1921). *Relativitätstheorie*. Leipzig: BG Teubner.

121. Einstein, A. (1988). *Über die spezielle und die allgemeine Relativitätstheorie* (23rd ed.). Berlin, Heidelberg, New York: Springer.
122. Jones, P., & Wanex, L. F. (2006). The clock paradox in a static homogeneous gravitational field. *Foundations of Physics*, 19(1), 75–85.
123. Ciarlet, P. G. (1988). *Mathematical elasticity – volume I: Three-dimensional elasticity*. Amsterdam, The Netherlands: North Holland 1988.
124. Sheldrake, R. (1998). *The presence of the past*. New York: Times Book.
125. Greiner, W., & Wolschin, G. (Eds.). (1988). *Elementare Materie, Vakuum und Felder. Die Struktur des Vakuums und der Bausteine der Natur*. Spektrum der Wissenschaft Verlagsgesellschaft.
126. Griffiths, D. J. (1987). *Introduction to elementary particles*. New York: Wiley.
127. Schwinger, J. (1948). Quantum electrodynamics. I. A covariant formulation. *Physical Review*, 74, 1439–1461.
128. 't Hooft, G., (1971). Renormalizable Lagrangians for massive Yang-Mills fields. *Nuclear Physics, B*, 35(1), 167–188.
129. de Lavoisier, A. L. (1789). *Traité Élémentaire de Chimie, présenté dans un ordre nouveau et d'après les découvertes récentes*. Couchet, Paris.
130. Mendeleff, D. (1905). *The principles of chemistry* (3rd ed.). London: Longmans, Green, and Co.
131. Thomson, J. J. (1897). Cathode rays. *Philosophical Magazine Series*, 6(44), 293–316.
132. Bohr, N. H. D. (1913). On the constitution of atoms and molecules. *Philosophical Magazine*, 26(151), 1–25.
133. Einstein, A. (1915). Die Feldgleichungen der Gravitation, *Sitzungsber. Königl. Preuss. Akad. Wiss.* (pp. 844–847).
134. Dirac, P. A. M. (1930). A theory of electrons and protons. *Proceedings of the Royal Society of London*, 126(8), 360–365.
135. Dirac, P. A. M. (1931). Quantised singularities in the electromagnetic field, *Proceedings of the Royal Society of London* (pp. 60–72).
136. Chadwick, J. (1932). Possible existence of a neutron. *Nature*, 129, 312.
137. Anderson, C. D. (1933). Cosmic-ray bursts. *Physical. Review*, 43(5), 368–369.
138. Anderson, C. D., & Anderson, C. D. (1933). The positive electron. *Physical. Review*, 43, 491–494.
139. Lande, K., Booth, E. T., Impeduglia, J., Lederman, L. M., & Chinowsky, W. (1956). Observation of long-lived neutral V particles. *Physical Review*, 103, 1901–1904.
140. Gell-Mann, M. (1961). The eightfold way: A theory of strong interaction symmetry. Technical report, Caltech Synchrotron Laboratory Report No. CTSL-20.
141. Gell-Mann, M. (1964). A schematic model of baryons and mesons. *Physics Letters*, 8(3), 214–215.
142. Zweig, G. (1964). *An SU(3) model for strong interaction symmetry and its breaking*. CERN: Technical report.
143. Zweig, G. (1964). *An SU(3) model for strong interaction symmetry and its breaking: II*. CERN: Technical report.
144. Barnes, V. E., Connolly, P. L., Crennell, D. J., Culwick, B. B., Delaney, W. C., Fowler, W. B., et al. (1964). Observation of a hyperon with strangeness minus three. *Physical Review Letters*, 12(8), 204–206.
145. Glashow, S. L. (1991). *The charm of Physics*. New York: SImon & Schuster.
146. Weinberg, S. (1967). A model of leptons. *Physical Review Letters*, 19(21), 1264–1266.
147. Salam, A. (1969). Weak and electromagnetic interactions. In N. Svartholm (Ed.), *Proceedings of the 8th Nobel Symposium on "Elementary Particle Theory, Relativistic Groups and Analyticity"* (pp. 367–377). Stockholm: Sweden.
148. Fritsch, H., & Gell-Mann, M. (1972). Current algebra: Quarks and what else? In *Proceedings of the XVI International Conference on High Energy Physics* (p. 164).

149. Aubert, J. J., Becker, U., Biggs, P. J., Burger, J., Chen, M., Everhart, G., et al. (1974). Experimental observation of a heavy particle. *J. Physical Review Letters*, 33, 1404–1406.
150. Rubbia, C. (1994). Experimental observation of the intermediate vector bosons W^+ , W^- , and Z^0 . In *API Conference Proceedings, Santa Monica, CA, USA* (pp. 526–591). AIP.
151. Abe, F., Akimoto, H., Akopian, A., et al. (1995). Observation of top quark production in p^-p collisions with the collider detector at fermilab. *Physical Review Letters*, 74, 2626–2631.
152. Collaboration, T. C. (2014). Evidence for the direct decay of the 125 GeV Higgs boson to fermions. *Nature Physics*, 10, 557–560.
153. Noether, E. (1918). Invariante Variationsprobleme. *Nachrichten von der Akademie der Wissenschaften in Göttingen, II*, 235–257.
154. Klammer, R., & Schörner-Sadenius, T. (2005). Verstehen wir die starke Kraft? *Physik Journal*, 5(5), 41–47.
155. Close, F. E. (1979). *An introduction to quarks and partons*. New York: Academic Press.
156. Halzen, F., & Martin, A. D. (1984). *Quarks & leptons: An introductory course in modern particle physics*. New York: Wiley.
157. Nachtmann, O. (1990). *Elementary particle physics: Concepts and phenomena. Texts and monographs in physics*. Berlin: Springer.
158. Aitchison, I. J. R., & Hey, A. J. G. (1989). *Gauge theories in particle physics* (2nd ed.). Bristol: Adam Hilger.
159. Sakurai, J. J. (1967). *Advanced quantum mechanics*. Reading: Addison Wesley.
160. Mandl, F., & Shaw, G. (1984). *Quantum field theory*. New York: Wiley.
161. Bjorken, D. J., & Drell, S. D. (1964). *Relativistic quantum mechanics*. New York: McGraw-Hill.
162. Bjorken, D. J., & Drell, S. D. (1965). *Relativistic quantum fields. International series in pure and applied physics*. New York: McGraw-Hill.
163. Dodd, J. E. (1984). *The ideas of particle physics*. Cambridge: Cambridge University Press.
164. Okun, L. B. (1991). *Physik der Elementarteilchen*. Berlin: Akademie Verlag.
165. Fritsch, H. (1992). *Quarks*. Munich, Zurich: Piper.
166. Creutz, M. (1983). *Quarks, gluons and lattices*. Cambridge: Cambridge University Press.
167. Montvay, L., & Münster, G. (1994). *Quantum fields on a lattice*. Cambridge: Cambridge University Press.
168. Brassard, G., & Bratley, P. (1996). *Fundamentals of algorithms*. Upper Saddle River: Prentice Hall.
169. McCarthy, J. (1960). Recursive functions of symbolic expressions and their computation by machine. *Part I. Communications of the ACM*, 3(4), 184–195.
170. Hopcroft, J. (1979). *Introduction to automata theory, languages and computation*. Reading: Addison-Wesley.
171. Turing, A. (1936). On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society* (pp. 230–265).
172. Hilbert, D. (1900). Mathematische Probleme. *Nachrichten von der Akademie der Wissenschaften in Göttingen, II* (Vol. 3, pp. 253–297).
173. Skolem, T. (1922). *Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre*. Redogorelse, Akademiska Bokhandeln, Helsinki: In Matematikerkongressen i Helsingfors.
174. Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38, 173–198.
175. Kleene, S. C. (1936). Lambda-definability and recursiveness. *Duke Mathematical Journal*, 2, 340–353.
176. Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58, 345–363.
177. Church, A. (1937). Review of turing 1936. *Journal of Symbolic Logic*, 2, 42–43.
178. Matiyasevich, Y. V. (1993). *Hilbert's tenth problem*. Cambridge, MA: MIT Press.
179. Kleene, S. C. (1952). *Introduction to Metamathematics*. Amsterdam: North-Holland.

180. Peter, R. (1935). Konstruktion nichtrekursiver Funktionen. *Mathematische Annalen*, 111, 42–60.
181. Robinson, R. M. (1948). Recursion and double recursion. *Bulletin of the American Mathematical Society*, 54, 987–993.
182. Davis, M. (1973). *Computability and unsolvability*. New York: Dover.
183. Ackermann, W. (1928). Zum Hilbertschen Aufbau der reellen Zahlen. *Mathematische Annalen*, 99, 118–133.
184. Horowitz, E., Sahni, S., & Rajasekaran, S. (1998). *Computer algorithms*. New York: Computer Science Press.
185. Motwani, R. (1995). *Randomized algorithms*. Cambridge: Cambridge University Press.
186. Cook, S. (1971). The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual Symposium on Theory of Computing, ACM, New York* (pp. 151–158).
187. Sipser, M. (1997). *Introduction to the theory of computation*. Boston: PWS Publishing Company.
188. Chabert, J. L., et al. (1999). *A history of algorithms - from the Pebble to the microchip*. New York: Springer.
189. Lee, D. T. (1980). Two-dimensional Voronoi diagrams in the L_p metric. *Journal of the Association for Computing Machinery*, 27, 604–618.
190. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms*. Cambridge: MIT Press.
191. Diestel, R. (1996). *Graphentheorie*. New York: Springer.
192. Promel, H. J., & Steger, A. (2002). *The Steiner tree problem - a tour through graphs, algorithms, and complexity*. Braunschweig: Friedrich Vieweg & Sohn.
193. Gruska, J. (1997). *Foundations of computing*. London: Thomson Computer Press.
194. Markov, A. A. (1906). Extension of the law of large numbers to dependent quantities (in Russian). *Izvestia Fiz.-Mathem. Obsch. Kazan Univ.*, (2nd Ser.) (Vol. 15, pp. 135–156).
195. Seneta, E. (1996). Markov and the birth of chain dependence theory. *International Statistical Review*, 64, 255.
196. Copeland, B. J. (Ed.). (2004). *The essential turing - the ideas that gave birth to the computer age*. Oxford: Oxford University Press.
197. Ausellio, G. (1999). *Complexity and approximation - combinatorial problems and their approximability properties*. New York: Springer.
198. Cooper, S. B. (2004). *Computability theory*. Boca Raton: Chapman & Hall.
199. Garey, M., & Johnson, D. (1979). *Computers and intractability - a guide to the theory of NP-completeness*. San Francisco: Freeman.
200. Homer, S., & Selman, A. L. (2001). *Computability and complexity theory*. New York: Springer.
201. Moore, G. H. (1995). *Russel's paradox. From Dedekind to Gödel*. Dordrecht: Kluwer Academic Publishers.
202. Listing, J. B. (1848). *Vorstudien in Topologie*. Göttingen: Vandenhoeck und Ruprecht.
203. Wald, R. M. (1984). *General relativity*. Chicago: The University of Chicago Press.
204. Schrödinger, E. (1950). *Space-time structure*. Oxford: Oxford University Press.
205. Einstein, A. (1916). Die Grundlage der allgemeinen Relativitätstheorie. *Annalen der Physik*, 49, 769–822.
206. Taylor, E. F., & Wheeler, J. A. (1992). *Space time physics*. New York: W.H. Freeman and Company.
207. Goenner, H. (1996). *Einführung in die spezielle und allgemeine Relativitätstheorie*. Heidelberg: Spektrum Akademischer Verlag.
208. Landau, L. D., & Lifschitz, E. M. (1976). *Lehrbuch der Theoretischen Physik* 2. Klassische Feldtheorie: Akademie-Verlag Berlin.
209. d'Inverno, R. (1992). *Introducing Einstein's relativity*. Oxford: Oxford University Press.
210. Fließbach, T. (1990). *Allgemeine Relativitätstheorie*. BI Wissenschaftsverlag Mannheim.

211. Bergmann, P. G. (1942). *Introduction to the theory of relativity*. New Jersey: Prentice-Hall Inc.
212. Weyl, H. (1988). *Raum, Zeit, Materie* (7th ed.). New York: Springer.
213. Misner, C. W., Thorne, K. S., & Wheeler, J. A. (1973). *Gravitation*. New York: W. H. Freeman and Company.
214. Stephani, H. (1982). *General relativity: An introduction to the gravitational field*. Cambridge: Cambridge University Press.
215. Schmutzler, E. (1979). *Relativitätstheorie - aktuell*. Leipzig: B.G. Teubner.
216. Dirschmid, H. J. (1996). *Tensors and fields*. Heidelberg: Springer.
217. Kaufman, L., & Rousseeuw, P. R. (1990). *Finding groups in data - an introduction to cluster analysis. Wiley series in probability and mathematical statistics*. New York: Wiley.
218. Einstein, A. (1918). Prinzipielles zur allgemeinen Relativitätstheorie. *Annalen der Physik*, 55, 241–244.
219. Howard, D., & Stachel, J. (1995). *Mach's principle: From Newton's bucket to quantum gravity* (Vol. 6). Einstein studies. Boston, Basel, Berlin: Birkhäuser.
220. Michelson, A. A., & Morley, E. W. (1887). On the relative motion of the earth and luminiferous ether. *Science*, 24(203), 333–345.
221. Einstein, A. (1915). Zur allgemeinen Relativitätstheorie. *Sitzungsber. Königl. Preuss. Akad. Wiss.*, 98, 778–786.
222. Einstein, A. (1915). Zur allgemeinen Relativitätstheorie (Nachtrag). *Sitzungsber. Königl. Preuss. Akad. Wiss.*, 108, 799–801.
223. Einstein, A. (1915). Erklärung der Perihelbewegung des Merkur aus der allgemeinen Relativitätstheorie. *Sitzungsber. Königl. Preuss. Akad. Wiss.*, 112, 831–839.
224. Stachel, J., Cassidy, D., Schulmann, R., & Renn, J. (Eds.). (1987). *The collected papers of Albert Einstein: The early years, 1897–1902* (Vol. 1). Princeton: Princeton University Press.
225. Bjerkenes, C. J. (2003). *Anticipations of Einstein in the general theory of relativity* (Vol. 1). Downers Grove Illinois XTX Inc.
226. Sommer, K. P. (2005). Wer entdeckte die Allgemeine Relativitätstheorie. *Physik in unserer Zeit*, 36(5), 230–235.
227. Wuensch, D. (2005). *Zwei wirkliche Kerle. Neues zur Entdeckung der Gravitationsgleichungen durch Albert Einstein und David Hilbert*. Göttingen: Termessos Verlag.
228. Renn, J., & Sauer, T. (1996). Einsteins Zürcher Notizbuch: die Entdeckung der Feldgleichungen der Gravitation im Jahre 1912. *Physikalische Blätter*.
229. Renn, J., Castagnetti, G., & Damerow, P. (1998). *Albert Einstein: Alte und neue Kontexte in Berlin*. Berlin: Springer.
230. Renn, J., & Stachel, J. (2007). Hilbert's foundation of physics: From a theory of everything to a constituent of general relativity. Technical report, Preprint 118, Max-Planck-Institute for the History of Science.
231. Hilbert, D. (1915). Die Grundlagen der Physik (Erste Mitteilung). *Nachrichten von der Akademie der Wissenschaften in Göttingen, II* (Vol. 3, pp. 395–407).
232. Howard, D., & Stachel, J. (Eds.). (1989). *Einstein and the history of general relativity* (Vol. I). Birkhäuser: Boston, Basel, Berlin.
233. Stachel, J., Cassidy, D., Schulmann, R., & Renn, J. (Eds.). (1989). *The collected papers of Albert Einstein: The Swiss years: Writings, 1900–1909* (Vol. 2). Princeton: Princeton University Press.
234. Klein, M. J., Kox, A. J., Renn, J., & Schulmann, R. (Eds.). (1993). *The collected papers of Albert Einstein: The Swiss years: Correspondence, 1902–1914* (Vol. 5). Princeton: Princeton University Press.
235. Klein, M. J., Kox, A. J., & Schulmann, R. (Eds.). (1993). *The collected papers of Albert Einstein: The Swiss years: Writings, 1909–1911* (Vol. 3). Princeton: Princeton University Press.

236. Klein, M. J., Kox, A. J., Renn, J., & Schulmann, R. (Eds.). (1995). *The collected papers of Albert Einstein: The Swiss years: Writings, 1912–1914* (Vol. 4). Princeton: Princeton University Press.
237. Schulmann, R., Kox, A. J., Janssen, M., & Illy, J. (Eds.). (1998). *The collected papers of Albert Einstein. Volume 8: The Berlin years: Correspondence, 1914–1918* (Vol. 8). Princeton: Princeton University Press.
238. Halmos, P. R. (1974). *Naive set theory*. New York: Springer.
239. Mendelson, E. (1997). *Introduction to mathematical logic* (4th ed.). London: Chapman & Hall.
240. Klingenberg, K. (1982). *Riemannian geometry*. Berlin: De Gruyter.
241. Darling, R. W. R. (1994). *Differential forms and connections*. Cambridge: Cambridge University Press.
242. Hicks, N. J. (1965). *N.J. Hicks*. New York: D. Van Nostrand.
243. Bishop, M., Clarke, J. H. R., Rey, A., & Freire, J. J. (1991). Brownian dynamics simulation of linear and star polymers. *The Journal of Chemical Physics*, 95, 3804.
244. Kobayashi, S., & Nomizu, K. (1969). *Foundations of differential geometry*. New York: Interscience.
245. Schouten, J. A. (1954). *Ricci calculus*. Berlin: Springer.
246. Hermann, R. (1968). *Differential geometry and the calculus of variations*. New York: Academic Press.
247. Flanders, A. (1963). *Differential forms*. New York: Academic Press.
248. Wallace, A. (1968). *Differential topology: First steps*. Benjamin, Reading: Mass.
249. Schutz, B. F. (1980). *Geometrical methods of mathematical physics*. Cambridge: Cambridge University Press.
250. Choquet-Bruhat, Y., DeWitt-Morette, C., & Dillard-Bleick, M. (1977). *Analysis, manifolds, and physics*. Amsterdam: North-Holland.
251. Warner, F. W. (1971). *A comprehensive introduction to differential geometry*. Foresmann Glenview, III: Scott.
252. Spivak, M. D. (1970). *A comprehensive introduction to differential geometry* (2nd ed.). Wilmington, DE: Publish or Perish.
253. Duschek, A., & Hochrainer, A. (1955). *Grundzüge der Tensorrechnung in analytischer Darstellung* (Vol. 1,2,3). Wein: Springer.
254. Wills, A. P. (1931). *Vector analysis*. New York: Prentice-Hall.
255. Sygne, L., & Schild, A. (1949). *Tensor calculus*. Toronto: University of Toronto Press.
256. Coburn, N. (1955). *Vector and tensor analysis*. New York: Macmillan.
257. French, A. P. (1966). *Special relativity. MIT introductory physics series*. New York, NY: Norton.
258. Rindler, W. (1977). *Essential relativity* (2nd ed.). New York: Springer.
259. Stephani, H. (1991). *Allgemeine Relativitätstheorie*. Berlin: Deutscher Verlag der Wissenschaften.
260. Schutz, B. F. (1985). *A first course in general relativity*. Cambridge: Cambridge University Press.
261. Pais, A. (1994). *Einstein lived here*. New York, NY: Oxford University Press Inc.
262. Schwarzschild, K. (1916). On the gravitational field of a sphere of incompressible fluid according to Einstein's theory. *Sitzungsber. Königl. Preuss. Akad. Wiss.*, 189–196.
263. Zill, D. G. (1986). *A first course in differential equations with applications*. Boston: Prindle, Weber & Schmidt.
264. Farlow, S. J. (1994). *An introduction to differential equations and their applications*. New York: McGraw-Hill.
265. Kolmogorov, A. N., & Fomin, S. V. (1970). *Introductory real analysis*. New York: Dover Publications Inc.

266. Courant, R., & Hilbert, D. (1962). *Methods in mathematical physics, vol. II: Partial differential equations*. New York: Wiley.
267. Jackson, J. D. (1975). *Classical electrodynamics*. New York: Wiley.
268. Kamke, E. (1961). *Differentialgleichungen, Lösungsmethoden und Lösungen. I. Gewöhnliche Differentialgleichungen*. Leipzig: Akademische Verlagsgesellschaft Geest & Portig K.-G.
269. Boas, M. L. (1966). *Mathematical methods in the physical sciences*. New York: Wiley.
270. Morse, P. M., & Feshbach, H. (1953). *Methods of theoretical physics*. New York: McGraw-Hill.
271. Farlow, S. J. (1982). *Partial differential equations for scientists and engineers*. New York: Wiley.
272. Porter, D., & Stirling, D. S. G. (1990). *Integral equations*. Cambridge: Cambridge University Press.
273. Morton, K. W., & Mayers, D. F. (2005). *Numerical solution of partial differential equations, an introduction*. Cambridge: Cambridge University Press.
274. Stoer, J., & Bulirsch, R. (1980). *Introduction to numerical analysis*. New York: Springer.
275. Press, W. H., Flannery, B. P., Teukolski, S. A., & Vetterling, W. T. (1986). *Numerical recipes*. Cambridge: Cambridge University Press.
276. Burden, R. L., & Faires, J. D. (1973). *Numerical analysis*. Boston: PWS-Kent Publishing Company.
277. Gould, H., & Tobochnik, J. (1988). *An introduction to computer simulation methods, parts 1, 2*. Reading, MA: Addison-Wesley.
278. Cohen, A. (1962). *Numerical analysis*. London: McGraw-Hill.
279. Abramovitz, M., & Segun, I. A. (1964). *Handbook of mathematical functions*. New York: Dover Publications.
280. Bronstein, I. N., & Semendjajew, K. A. (1980). *Taschenbuch der Mathematik*. Frankfurt Main: Verlag Harri Deutsch, Thun.
281. Wolfram, S. (1997). *Mathematica*. New York: Addison-Wesley.
282. Abell, M. L., & Braselton, J. P. (1993). *Differential equations with mathematica*. Cambridge, MA: Academic Press.
283. Vvedensky, D. (1994). *Partial differential equations with mathematica*. New York: Addison-Wesley.
284. Hrennikoff, H. (1941). Solutions of problems in elasticity by the framework method. *Journal of Applied Mechanics A*, 8, 169–175.
285. Courant, R. (1943). Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49(1), 1–23.
286. Needleman, S. B., & Wunsch, C. N. (1970). A general method applicable to the search for similarities in the aminoacid sequence of two proteins. *Journal of Molecular Biology*, 48, 443–453.
287. Libersky, L. D., & Petschek, A. G. (1991). Smoothed particle hydrodynamics with strength of materials. In *The Next Language Conference* (pp. 248–257). New York: Springer.
288. Nayroles, B., Touzot, G., & Villon, P. (1992). Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10(5), 307–318.
289. Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Fracture and crack growth by element-free Galerkin methods. *Molecular Simulation of Science and Computer Engineering*, 2, 519–534.
290. Belytschko, T., Lu, Y. Y., & Gu, L. (1994). Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2), 229–256.
291. Liu, G. R., & Liu, M. B. (2003). *Smoothed particle hydrodynamics. A meshfree particle method*. Singapore: World Scientific.
292. Quinlan, G. D., & Tremaine, S. (1992). On the reliability of gravitational N-body integrations. *Monthly Notices of the Royal Astronomical Society (ISSN 0035-8711)*, 259, 505–518.
293. Gillian, R. E., & Wilson, K. R. (1992). Shadowing, rare events, and rubber bands - a variational Verlet algorithm for molecular-dynamics. *The Journal of Chemical Physics*, 97, 1757–1772.

294. Gear, C. W. (1971). *Numerical initial value problems in ordinary differential equations*. New Jersey: Englewood Cliffs.
295. Goldstein, H., Poole, C. P., & Safko, J. L. (2002). *Classical mechanics* (3rd ed.). Cambridge: Addison-Wesley.
296. Tuckerman, M. E., Berne, B. J., & Martyna, G. J. (1992). Reversible multiple time scale molecular dynamics. *The Journal of Chemical Physics*, 97, 1990–2001.
297. Verlet, L. (1968). Computer ‘experiments’ on classical fluids. *II. Equilibrium correlation functions*. *Physical Review*, 165(1), 201–214.
298. Berendsen, H. J. C., & van Gunsteren, W. F. (1986). Practical algorithms for dynamics simulations. In G. Cigotti & W. G. Hoover (Eds.), *Molecular dynamics simulations of statistical mechanics systems, Proceedings of the 97th International Enrico Fermi School of Physics* (pp. 43–65). Amsterdam: North Holland.
299. Gilb, T. (1988). *Principles of software engineering management*. Wokinham, England: Addison-Wesley.
300. DeMarco, T. (1982). *Controlling software projects*. New York: Yourdon Press.
301. Maguire, S. (1991). *Debugging the development process*. Redmond: Microsoft Press.
302. Yourdon, E., & Constantine, L. L. (1979). *Structured design: Fundamentals of a discipline of computer program and systems design*. Englewood Cliffs, NY: Yourdon Press.
303. Booch, G. (1991). *Object oriented design: With applications*. Redwood City, CA: Benjamin/Cummings.
304. Myers, G. J. (1976). *Software reliability*. New York: Wiley.
305. Hetzel, B. (1988). *The complete guide to software testing*. Wellesley, MA: QED Information Systems.
306. Sedgewick, R. (1988). *Algorithms* (2nd ed.). Reading: Addison-Wesley.
307. Reingold, E. M., & Hansen, W. J. (1983). *Data structures*. Boston, MA: Little Brown.
308. Maguire, S. (1990). *Writing solid code*. Redmond: Microsoft Press.
309. McConnell, S. (1993). *Code complete*. Redmond: Microsoft Press.
310. Knuth, D. (1973). *The art of computer programming*, vol. 1. *Fundamental algorithms of 1. Fundamental algorithms*. Reading: Addison-Wesley.
311. Knuth, D. (1973). *The art of computer programming*, vol. 3. *Sorting and searching of 3. Sorting and searching*. Reading: Addsion-Wesley.
312. Knuth, D. (1981). *The art of computer programming*, vol. 2. *Seminumerical algorithms of 2. Seminumerical algorithms*. Reading: Addison-Wesley.
313. Constantine, L. L. (1990). Objects, functions and program extensibility. *Computer Language*, 7(1), 34–56.
314. Caine, S. H., & Gordon, E. K. (1975). PDL – a tool for software design. In *National Computer Conference, AFIPS Press* (pp. 271–276).
315. Dijkstra, E. (1972). The humble programmer. Turing award lecture. *Communications of the ACM*, 10, 859–866.
316. Basili, V. R., & Weiss, D. M. (1984). A methodology for collecting valid software engineering data. *IEEE Transactions on Software Engineering SE-10*, 6, 728–738.
317. Card, D., Page, G., & McGarry, F. (1985). Criteria for software modularization. In *Proceedings of the 8th International Conference on Software Engineering, Washington DC, IEEE Computer Society Press* (pp. 372–377).
318. Lind, R. R., & Vairavan, K. (1989). An experimental investigation of software metrics and their relationship to software development effort. *IEEE Transactions on Software Engineering SE-15*, 5, 649–653.
319. Jones, T. C. (1986). *Applied software measurement*. New York: McGraw-Hill.
320. Conte, S. D., Dunsmore, H. E., & Shen, V. Y. (1986). *Software engineering metrics and models*. Menlo Park, CA: Benjamin/Cummings.
321. Knuth, D. (1971). An empirical study of FORTRAN programs. *Software Practice and Experience*, 1, 105–133.

322. Stevens, W., Myers, G., & Constantine, L. (1974). Structured design. *IBM Systems Journal*, 2, 115–139.
323. Meyer, B. (1988). *Object-oriented software construction*. New York: Prentice Hall.
324. Coad, P. (1985). *Object-oriented design*. Englewood Cliffs, NY: Yourdon Press.
325. Parnas, D. (1979). Designing software for ease of extension and contraction. *IEEE Transactions of the ACM*, 12, 1053–1058.
326. Jackson, M. A. (1975). *Principles of program design*. New York: Academic Press.
327. Adams, J. L. (1980). *Conceptual blockbuster: A guide to better ideas*. New York: Norton.
328. Simon, H. (1969). *The sciences of the artificial*. Cambridge MA: MIT Press.
329. Glass, R. L. (1992). *Building quality software*. Englewood Cliffs, NJ: Prentice Hall.
330. Smith, C. U. (1990). *Performance engineering of software systems*. Reading, MA: Addison-Wesley.
331. Bentley, J. (1982). *Writing efficient programs*. Englewood Cliffs, NJ: Prentice Hall.
332. Weinberg, G. M. (1971). *The psychology of computer programming*. New York: Van Nostrand Reinhold.
333. Born, M., Heisenberg, W. K., & Jordan, P. (1926). Zur Quantenmechanik II. *Zeitschrift für Physik*, 35, 557–615.
334. Landau, L. D., & Lifschitz, E. M. (1977). *Quantum mechanics (non-relativistic theory)* (3rd ed.). New York: Elsevier Science Limited.
335. Dirac, P. A. M. (1958). *The principles of quantum mechanics*. London: Oxford University Press.
336. Messiah, A. (1961). *Quantum Mechanics*. Amsterdam: North-Holland.
337. Sakurai, J. J. (1994). *Modern quantum mechanics*. Reading: Addison Wesley Publishing, revised ed.
338. Schiff, L. I. (1968). *Quantum mechanics*. New York: McGraw-Hill.
339. Bohr, N. (1913). On the constitution of atoms and molecules. Part II. Systems containing only a single nucleus. *Philosophical Magazine*, 26, 476–502.
340. Bohr, N. (1913). On the constitution of atoms and molecules. Part III. Systems containing several nuclei. *Philosophical Magazine*, 26, 857–875.
341. Born, M., & Heisenberg, W. (1923). Die Elektronenbahnen im angeregten Heliumatom. *Zeitschrift für Physik*, 16, 229–243.
342. Heisenberg, W. (1925). Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen. *Zeitschrift für Physik*, 33, 879–893.
343. Dirac, P. A. M. (1925). The fundamental equations of quantum mechanics. *Proceedings of the Royal Society of London*, 109, 642–653, Dec. 1925.
344. Schrödinger, E. (1926). Quantisierung als Eigenwertproblem (Erste Mitteilung). *Annalen der Physik*, 79(361–367), 1926.
345. Schrödinger, E. (1926). Quantisierung als Eigenwertproblem (Zweite Mitteilung). *Annalen der Physik*, 79(6), 489–527.
346. Schrödinger, E. (1926). Quantisierung als Eigenwertproblem (Dritte Mitteilung). *Annalen der Physik*, 80(13), 437–490.
347. Schrödinger, E. (1926). Quantisierung als Eigenwertproblem (Vierte Mitteilung). *Annalen der Physik*, 81(18), 109–139.
348. Schrödinger, E. (1982). *Collected papers on wave mechanics*. Providence, RI: AMS Chelsea Publishing.
349. Born, M. (1926). Zur Quantenmechanik der Stoßvorgänge. *Zeitschrift für Physik*, 37, 863–867.
350. von Neumann, J. (1983). *Mathematical foundations of quantum mechanics*. Princeton landmarks in mathematics and physics. Princeton, New Jersey: Princeton University Press, reprint ed.
351. Sutcliffe, B. T. (1997). The nuclear motion problem in molecular physics. *Advances in Quantum Chemistry*, 28, 65–80.

352. Combes, J. M., Duclos, P., & Seiler, R. (1981). The Born-Oppenheimer approximation. In G. Velo & S. Wightman (Eds.), *Rigorous atomic and molecular physics* (pp. 185–212).
353. Payne, M. C., Teter, M. P., Allan, D. C., Arias, T. A., & Joannopoulos, J. D. (1992). Iterative minimization techniques for ab initio total-energy calculations: Molecular dynamics and conjugate gradients. *Reviews of Modern Physics*, 64(4), 1045–1097.
354. Hohenberg, P., & Kohn, W. (1964). Inhomogeneous electron gas. *Physical Review*, 36(3B), 864–871.
355. Kohn, W., & Sham, L. J. (1965). Self-consistent equations including exchange and correlation effects. *Physical Review*, 140(4A), A1133–A1138.
356. Becke, A. D. (1993). A new mixing of Hartree-Fock and local density-functional theories. *The Journal of Chemical Physics*, 98, 1372–1377.
357. Lee, C., Yang, W., & Parr, R. (1988). Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Physical Review B*, 37, 785–789.
358. Car, R., & Parrinello, M. (1985). Unified approach for molecular dynamics and density-functional theory. *Physical Review Letters*, 55, 2471–2474.
359. Andersen, H. C. (1983). RATTLE: A "velocity" version of the SHAKE algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 52(1), 24–34.
360. Kremler, D. K., & Madden, P. A. (1990). Molecular dynamics without effective potentials via the Car-Parinello approach. *Molecular Physics*, 70(6), 921–966.
361. Tuckerman, M. E., & Parinello, M. (1994). Integrating the Car-Parinello equations. I. Basic integration techniques. *The Journal of Chemical Physics*, 101(2), 1302–1315.
362. Hellman, H. (1937). *Einführung in die Quanten Theorie*. Leipzig: Deuticke.
363. Feynman, R. P. (1939). Forces in molecules. *Physical Review*, 56(4), 340–343.
364. Pauli, W. (1925). Über den Zusammenhang des Abschlusses der Elektronengruppen im Atom mit der Komplexstruktur der Spektren. *Zeitschrift für Physik*, 31, 765.
365. Fermi, E. (1926). Zur Quantelung des idealen einatomigen Gases. *Zeitschrift für Physik*, 36(11–12), 902–912.
366. Dirac, P. A. M. (1926). On the theory of quantum mechanics. *Proceedings of the Royal Society of London*, 112(661–677), 1926.
367. Bose, S. N. (1924). Plancks Gesetz und Lichtquantenhypothese. *Zeitschrift für Physik*, 26, 178.
368. Einstein, A. (1924). Quantentheorie des einatomigen idealen Gases. *Sitzungsber. Königl. Preuss. Akad. Wiss.*, 22(261–267), 1924.
369. Hartree, D. R. (1928). The wave mechanics of an atom with a non-Coulomb central field. Part III. Term values and intensities in series in optical spectra. *Mathematical Proceedings of the Cambridge*, 24(03), 426–437.
370. Fock, V. (1932). Näherungsmethoden zur Lösung des quantenmechanischen Mehrkörperproblems. *Zeitschrift für Physik*, 61, 126–148.
371. Slater, J. C. (1930). Atomic shielding constants. *Physical Review*, 36, 57–64.
372. Slater, J. C. (1929). The theory of complex spectra. *Physical Review*, 34, 1293–1322.
373. Ditchfield, R., Hehre, W. J., & Pople, J. A. (1971). Self-consistent molecular-orbital methods. IX. An extended gaussian-type basis for molecular-orbital studies of organic molecules. *The Journal of Chemical Physics*, 51, 724–728.
374. Pople, J. A., Head-Gordon, M., Fox, D. J., Raghavachari, K., & Curtiss, L. A. (1989). Gaussian-1 theory: A general procedure for prediction of molecular energies. *The Journal of Chemical Physics*, 90, 5622–5629.
375. Roothaan, C. C. J. (1951). A study of two-center integrals useful in calculations on molecular structure. I. *The Journal of Chemical Physics*, 19, 1445–1458.
376. Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92), 577.
377. Møller, C., & Plesset, M. S. (1934). Note on an approximation treatment for many-electron systems. *Physical Review*, 46, 618–622.

378. Pople, J. A., Binkley, J. S., & Seeger, R. (1976). Theoretical models incorporating electron correlation. *International Journal of Quantum Chemistry*, 10S(S10), 1–19.
379. Krishnan, R., & Pople, J. A. (1978). Approximate fourth-order perturbation theory of the electron correlation energy. *International Journal of Quantum Chemistry*, 14(1), 91–100.
380. Wolinski, K., & Pulay, P. (1989). Generalized Moller-Plesset perturbation theory: Second order results for two-configuration, open-shell excited singlet, and doublet wave functions. *The Journal of Chemical Physics*, 90(7), 3647–3659.
381. Bartlett, R. J. (1981). Many-body perturbation theory and coupled cluster theory for electron correlation in molecules. *Annual Review of Physical Chemistry*, 32, 359–401.
382. Foulkes, W. M. C., Haydock, R., & Haydock, R. (1989). Tight-binding models and density-functional theory. *Physical Review B*, 39(1), 12520–12536.
383. Slater, J. C., & Koster, G. F. (1954). Simplified LCAO method for the periodic potential problem. *Physical Review*, 94, 1498–1524.
384. Porezag, D., Frauenheim, T., Köhler, T., Seifert, G., & Kaschner, R. (1995). Construction of tight-binding-like potentials on the basis of density-functional theory: Application to carbon. *Physical Review B*, 51, 12947–12957.
385. Elstner, M., Porezag, D., Jungnickel, G., Elsner, J., Haugk, M., Frauenheim, T., et al. (1998). Self-consistent-charge density-functional tight-binding method for simulations of complex materials properties. *Physical Review B*, 58, 7260–7268.
386. Daw, M. S. (1993). Model for energetics of solids based on the density matrix. *Physical Review B*, 47, 10895–10898.
387. Mauri, F., Galli, G., & Car, R. (1993). Orbital formulation for electronic-structure calculations with linear system-size scaling. *Physical Review B*, 47(15), 9973–9976.
388. Kohn, W. (1996). Density functional and density matrix method scaling linearly with the number of atoms. *Physical Review Letters*, 76(17), 3168–3171.
389. Frauenheim, T. (2002). Atomistic simulations of complex materials: Ground-state and excited-state properties. *Journal of Physics: Condensed Matter*, 14, 3015–3047.
390. Goedecker, S. (1999). Linear scaling electronic structure methods. *Reviews of Modern Physics*, 71, 1085–1123.
391. Tsuruta, K., Totsuji, H., & Totsujii, C. (2000). Parallel-tight binding simulations of Nanophase ceramics: Atomic and electronic transport at grain boundaries. In L. P. Kubin, R. L. Selinger, J. L. Bassani, & K. Cho (Eds.), *Multiscale modeling of materials*. Warrandale: Pennsylvania.
392. Johnson, B. G., Gill, P. M. W., & Pople, J. A. (1998). The performance of a family of density functional methods. *The Journal of Chemical Physics*, 98, 5612–5626.
393. Segall, M. D., Lindan, P. J. D., Probert, M. J., Pickard, C. J., Hasnip, P. J., Clark, S. J., et al. (2002). First-principles simulation: ideas, illustrations and the CASTEP code. *Journal of Physics: Condensed Matter*, 14, 2717–2744.
394. Hestenes, D. (1999). *New foundations for classical mechanics* (2nd ed.). Dordrecht: Kluwer Academic Publishers.
395. Montgomery, R. (2001). A new solution to the three-body problem. *Notices of the AMC*.
396. Holian, B. L., Lomdahl, P. S., & Zhou, S. (1997). Fracture simulations via large-scale nonequilibrium molecular dynamics. *Physica A: Statistical Mechanics and its Applications*, 240(1–2), 340–348.
397. Steinhauser, M. O., Grass, K., Thoma, K., & Blumen, A. (2006). A nonequilibrium molecular dynamics study on shock waves. *Europhysics Letters*, 73, 62–68.
398. Zhakhovskii, V. V., Zybina, S. V., Nishihara, N., & Anisimov, S. I. (1999). Shock wave structure in Lennard-Jones crystal via molecular dynamics. *Physical Review Letters*, 83, 1175–1179.
399. Margenau, H. (1939). Van der waals forces. *Reviews of Modern Physics*, 11, 1–35.
400. Ewald, P. P. (1921). Die Berechnung optischer und electrostatischer Gitterpotentiale. *Annalen der Physik*, 64, 253–287.
401. Heyes, D. M. (1981). Electrostatic potentials and fields in infinite point charge lattices. *The Journal of Chemical Physics*, 74(3), 1924–1929.

402. London, F. (1930). Zur Theorie und Systematik der Molekularkräfte. *Zeitschrift für Physik*, 63(3), 245–279.
403. Abell, G. C. (1985). Empirical chemical pseudopotential theory of molecular and metallic bonding. *Physical Review B*, 31, 6184–6196.
404. Tersoff, J. (1989). Modeling solid-state chemistry: Interatomic potentials for multicomponent systems. *Physical Review B*, 39(8), 5566–5568.
405. Stillinger, F., & Weber, T. (1985). Computer simulation of local order in condensed phases of silicon. *Physical Review B*, 31, 5262–5271.
406. Clementi, E., & Roetti, C. (1974). Roothaan-Hartree-Fock atomic wavefunctions: Basis functions and their coefficients for ground and certain excited states of neutral and ionized atoms, $Z = 54$. *Atomic Data and Nuclear Data Tables*, 14(3–4), 177–478.
407. Kadau, K. (2001). *Molekulardynamik-Simulationen von strukturellen Phasenumwandlungen in Festkörpern mit Nanopartikeln und ultradünnen Filmen*. Ph.D. thesis, Gerhard-Mercator-University Duisburg, Germany.
408. Finnis, M. W., & Sinclair, J. E. (1984). A simple empirical N-body potential for transition metals. *Philosophical Magazine A*, 50, 45–55.
409. Jones, J. E. (1924). On the determination of molecular fields. I. From the variation of the viscosity of a gas with temperature. *Proceedings of the Royal Society of London* (pp. 441–462). The Royal Society.
410. Buckingham, R. A. (1938). The classical equation of state of gaseous helium, neon and argon. *Proceedings of the Royal Society of London* (pp. 264–283). The Royal Society.
411. Gibson, J. B., Goland, A. N., Milgram, M., & Vineyard, G. H. (1960). Dynamics of radiation damage. *Physical Review*, 120, 1229–1253.
412. Catlow, C. R. A., Parker, S. C., & Allen, M. P. (1990). *Computer modeling of fluids, polymers and solids*. Dordrecht: Kluwer Academic Publishers.
413. Ohno, K., Esfarjani, K., & Kawazoe, Y. (1999). *Computational materials science: From ab-initio to Monte Carlo methods*. Berlin: Springer.
414. Steinhauser, M. O. (2013). *Computer simulation in physics and engineering* (1st ed.). Berlin, Boston: deGruyter.
415. Cundall, P. A., & Strack, O. D. (1979). A discrete numerical model for granular assemblies. *Geotechnique*, 29(1), 47–65.
416. Chung, J., & Lee, J. M. (1994). A new family of explicit time integration methods for linear and non-linear structural dynamics. *International Journal for Numerical Methods in Engineering*, 37(23), 3961–3976.
417. Andersen, H. C. (1980). Molecular dynamics simulations at constant pressure and/or temperature. *The Journal of Chemical Physics*, 72(4), 2384.
418. Doi, M., & Edwards, S. F. (1986). *The theory of polymer dynamics*. Oxford: Clarendon Press.
419. Adams, D. J., Adams, F. M., & Hills, G. J. (1979). The computer simulation of polar liquids. *Molecular Physics*, 38, 387–400.
420. Heller, A., & Parker, A. (2004). *Life at the nanoscale*. Lawrence Livermore National Laboratory: Technical report.
421. Kleman, M., & Lavrentovich, O. D. (2003). *Soft matter physics*. New York: Springer.
422. Lodge, T. P., & Muthukumar, M. (1996). Physical chemistry of polymers: Entropy, interactions, and dynamics. *The Journal of Physical Chemistry*, 100(31), 13275–13292.
423. Söder, G. (2000). *Molekulardynamiksimulationen der Flexibilität und Fluoreszenzanisotropie eines an ein Protein gebundenen Farbstoffs*. Ph.D. thesis, University of Göttingen.
424. Flory, P. J. (1969). *Statistical mechanics of chain molecules*. New York: Wiley.
425. de Gennes, P. G. (1979). *Scaling concepts in polymer physics*. Ithaca, London: Cornell University Press.
426. Steinhauser, M. O. (2005). A molecular dynamics study on universal properties of polymer chains in different solvent qualities. Part I. A review of linear chain properties. *The Journal of Chemical Physics*, 122, 094901-1–094901-13.

427. Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik (N.S.)*, 47(89), 271–306.
428. Libersky, L. D., Petschek, A. G., Carney, T. C., & Hipp, J. R. (1993). High strain Lagrangian hydrodynamics. *Journal of Computational Physics*, 109, 67–75.
429. Monaghan, J. J. (1988). An introduction to SPH. *Computer Physics Communications*, 48(1), 89–96.
430. Steinhauser, M. O., & Kühn, M. (2006). Modeling of shock-wave failure in brittle materials. In P. Gumbsch (Ed.), *MMM Multiscale Materials Modelling (3rd International Conference on Multiscale Materials Modeling (MMM), Freiburg, Germany, 18–22. 09. 2006)* (pp. 380–382). Fraunhofer IRB Verlag.
431. Krell, A., Blank, P., Ma, H., & Hutzler, T. (2003). Processing for high-density submicrometer Al_2O_3 for new applications. *Journal of the American Ceramic Society*, 86, 546–553.
432. Ghosh, S., & Liu, Y. (1995). Voronoi cell finite element model based on micropolar theory of thermoelasticity for heterogeneous materials. *International Journal for Numerical Methods in Engineering*, 38, 1361–1398.
433. Espinosa, H. D., & Lee, S. (1999). Modeling of ceramic microstructures: Dynamic damage initiation and evolution. In M. D. Furnish, L. C. Chhabildas, & R. S. Hixson (Eds.), *CP505. Shock Compression of Condensed Matter*.
434. Zavattieri, P. D., & Espinosa, H. D. (2001). Grain level analysis of crack initiation and propagation in brittle materials. *Acta Materialia*, 49(20), 4291–4311.
435. Lauridsen, E. M. (2006). Approaches for 3D materials characterization. *The Journal of the Minerals*, 58, 12.
436. Lewis, A. C., Suh, S., Stukowski, M., Geltmacher, A. B., Spanos, G., & Rajan, K. (2006). Quantitative analysis and feature recognition in 3-D microstructural data sets. *The Journal of the Minerals*, 12(12), 51–56.
437. Zhang, K. S., Wu, M. S., & Feng, R. (2005). Simulation of microplasticity-induced deformation in uniaxially strained ceramics by 3-D Voronoi polycrystal modeling. *International Journal of Plasticity*, 21, 801–834.
438. Okabe, A., Boots, B., & Sugihara, K. (1992). *Spatial tessellations - concepts and applications of Voronoi diagrams*. New York: Wiley.
439. Edelsbrunner, H., & Shah, N. R. (1992). Incremental topological flipping works for regular triangulations. Technical report, Department of Computer Science, University of Illinois at Urbana-Champaign, 1304 W. Springfield Avenue Urbana, IL 61801.
440. Tavernier, P., & Szpunar, J. A. (1991). Modelling of recrystallization textures. *Acta Metallurgica*, 39, 549–556.
441. Anderson, M. P., Srolovitz, D. J., Grest, G. S., & Sahni, P. S. (1984). Computer simulation of grain growth-I. Kinetics. *Acta Metallurgica*, 32(5), 783–791.
442. Hoogerbrugge, P. J., & Koelman, J. M. V. A. (1992). Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhysics Letters*, 19(3), 155–160.
443. Warren, P. B. (1998). Dissipative particle dynamics. *Current Opinion in Colloid & Interface Science*, 3, 620–624.
444. Groot, R. D., & Warren, P. B. (1997). Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of Chemical Physics*, 107, 4423–4435.
445. Kim, J. M., & Phillips, R. J. (2004). Dissipative particle dynamics simulations of flow around spheres and cylinders at finite Reynolds numbers. *Chemical Engineering Science*, 59, 4155–4168.
446. Chen, S., Phan-Thien, N., Fan, X.-J., Khoo, B. C., & Khoo, B. C. (2004). Dissipative particle dynamics simulation of polymer drops in a periodic shear flow. *Journal of Non-Newtonian Fluid Mechanics*, 118(1), 65–81.
447. Fan, Q.-H., Chen, Y.-M., Chen, X.-M., Jiang, D.-Z., Xi, F., & Chan, A. S. C. (2000). Highly effective and recyclable dendritic BINAP ligands for asymmetric hydrogenation. *Chemical Communications*, 1(9), 789–790.

448. Cundall, P. A. (1989). Numerical experiments on localization of frictional materials. *Ingenieur Archiv*, 59, 148–159.
449. Gröger, T., Tüzün, U., & Heyes, D. M. (2003). Modelling and measuring of cohesion in wet granular materials. *Powder Technology*, 133(1–3), 203–215.
450. Leszczynski, J. S. (2003). A discrete model of a two-particle contact applied to cohesive granular materials. *Granular Material*, 5, 91–98.
451. Kadau, D., Bartels, G., Brendel, L., & Wolf, D. E. (2002). Contact dynamics simulations of compacting cohesive granular systems. *Computer Physics Communications*, 147(1–2), 190.
452. Kadau, D., Bartels, G., Brendel, L., & Wolf, D. E. (2003). Pore stabilization in cohesive granular systems. *Phase Transitions*, 76(4–5), 315–331.
453. Sadd, M. H., Tai, Q. M., & Shukla, A. (1993). Contact law effects on wave propagation in particulate materials using distinct element modeling. *International Journal of Non-Linear Mechanics*, 28(2), 251–265.
454. Gao, H., Huang, Y., Huang, Y., & Abraham, F. F. (2000). Continuum and atomistic studies of intersonic crack propagation. *Journal of the Mechanics and Physics of Solids*, 49, 2113–2132.
455. Buehler, M. J., Abraham, F. F., & Gao, H. (2003). Hyperelasticity governs dynamic fracture at a critical length scale. *Nature*, 426, 141–146.
456. Abraham, F. F., Brodbeck, D., Rudge, W. E., Broughton, J. Q., Schneider, D., Land, B., et al. (1998). Ab initio dynamics of rapid fracture. *Modelling and Simulation in Materials Science and Engineering*, 6, 639–670.
457. Bulatov, V., Abraham, F. F., Kubin, L., Devincre, B., & Yip, S. (1998). Connecting atomistic and mesoscale simulations of crystal plasticity. *Nature*, 391, 669–672.
458. Fineberg, J., Gross, S. P., Marder, M., & Swinney, H. L. (1992). Instability in the propagation of fast cracks. *Physical Review B*, 45(10), 5146–5154.
459. Fineberg, J., & Marder, M. (1999). Instability in dynamic fracture. *Physics Reports*, 313, 1–108.
460. Marder, M. (1991). New dynamical equation for cracks. *Physical Review Letters*, 66(19), 2484–2487.
461. Abraham, F. F., & Gao, H. J. (2000). How fast can cracks propagate? *Physical Review Letters*, 84(14), 3113–3116.
462. Samudrala, O., Huang, Y., & Rosakis, A. J. (2002). Subsonic and intersonic mode II crack propagation with a rate-dependent cohesive zone. *Journal of the Mechanics and Physics of Solids*, 50(6), 1231–1268.
463. Gross, S. P., Fineberg, J., Marder, M., McCormick, W. D., & Swinney, H. L. (1993). Acoustic emissions from rapidly moving cracks. *Physical Review Letters*, 71(19), 3162–3165.
464. Weibull, W. (1951). A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, 18, 293–297.
465. Zavattieri, P. D., & Espinosa, H. D. (2003). An examination if the competition between bulk behavior and interfacial behavior of ceramics subjected to dynamics pressure-shear loading. *Journal of the Mechanics and Physics of Solids*, 51, 607.
466. Xu, X. P., & Needleman, A. (1996). Numerical simulations of dynamic interfacial crack growth allowing for crack growth away from the bond line. *International Journal of Fracture*, 74(3), 253–275.
467. D'Addetta, G. A., Kun, F., & Ramm, E. (2002). On the application of a discrete model to the fracture process of cohesive granular materials. *Granular Material*, 4(2), 77–90.
468. Grass, K., Blumen, A., Steinhauser, M. O., & Thoma, K. (2005). Sequential modeling of failure behavior in cohesive brittle materials. In R. García-Rojo, H. J. Herrmann, & M. Sean (Eds.), *The 5th International Conference on Micromechanics of Granular Media, Stuttgart, Germany, 18–22 July 2005* (pp. 1447–1550).
469. Hetnarski, R. B., & Ignaczak, J. (2004). *Mathematical theory of elasticity*. New York: Taylor & Francis Books Ltd.

-
- 470. Heuser, H. (1989). *Lehrbuch der Analysis* (6th ed., Vol. I). Mathematische Leitfäden Stuttgart: B.G. Teubner.
 - 471. Dedekind, R. (1960). *Stetigkeit und Irrationale Zahlen* (6th ed.). Braunschweig: Friedrich Vieweg & Sohn, Unabridged.
 - 472. Dedekind, R. (1969). *Was sind und was sollen die Zahlen/Stetigkeit und Irrationale Zahlen*. Braunschweig: Friedrich Vieweg & Sohn.
 - 473. Wigner, E. P. (1960). The unreasonable effectiveness of mathematics in the natural sciences. *Communications on Pure and Applied Mathematics*, 13, 1–14.
 - 474. Cantor, G. (1980). *Gesammelte Abhandlungen mathematischen und philosophischen Inhalts*. E. Zermelo, reprint ed.
 - 475. Gödel, K. (1940). Consistency of the axiom of choice and of the generalized continuum-hypothesis with the axioms of set theory. *Annals of Mathematics Studies*, 3, 1–66.
 - 476. Cohen, L. W., & Ehrlich, G. (1963). *The structure of the real number system*. New York: Van Nostrand Reinhold Company.
 - 477. Cohen, P. J. (1963). The independence of the continuum hypothesis. *Proceedings of the National Academy of Sciences*, 50, 1143–1148.
 - 478. Heller, A., & Parker, N. (2005). *Experiment and theory have a new partner: Simulation*. Lawrence Livermore National Laboratory: Technical report.
 - 479. Zermelo, E. (1930). Über Grenzzahlen und Mengenbereiche. *Fundamenta Mathematicae*, 16, 29–47.
 - 480. Enderton, H. B. (1977). *Elements of set theory*. New York: Academic Press.
 - 481. Jech, T. (1997). *Set theory* (2nd ed.). New York: Springer.

Index

A

Ab initio, 4, 7, 56, 60, 61, 102
calculation, 60
method, 6, 25
Algorithm
Metropolis, 18
Anti-particle, 62
Atomic scale, 5, 9
Axiom, 14
of infinity, 367

B

Basis set, 239
Besso, Michelle, 44
Bethe, Hans, 18
Bianchi's identity, 44
Bloch's theorem, 11
BLYP3, 254
Bootom-up approach, 34
Born–Oppenheimer approximation, 6, 61, 234
Bose–Einstein condensate, 11
Boson, 61
Bravais lattice, 11
BYLYP, 254

C

Cartesian Product, 117
CFD, 10
Computer
science, 90
simulation, 14
Computing, 19
Conservation Equations, *see* Equation
Contravariant
component, 51
Covariant

component, 51
Dirac equation, 74

D

DEM, 187
Deterministic, 38
DFT, 25
Differential
geometry, 77
of a function, 50, 54
Differential equation, *see* Equation
Dirac equation, 73

E

Einstein, Albert, 31
Electromagnetism, 4
Equation
conservation, 10
Klein-Gordon, 73
ordinary differential, 4, 5, 47–49, 58
partial differential, 10, 44, 48
Schrödinger, 239
Euler-Lagrange equations, 71
Event, 55

F

Fermi, Enrico, 18
Fermion, 61, 72, 74, 76
Feynman, Richard, 63
Fock, Wladimir Alexandrowitsch,
234
Four-vector, 51
Frame of reference, 55

G

Galilei transformation, 55

- Galilei, Galileo, 37, 39
Gauge
 invariance, 75, 76
 symmetry, 76
 transformation, 76
Geodesic, 37
Ghost particle, 294
Glashow, Sheldon L., 77
Gluon, 59
Graph theory, 90
Group, 66
 Abelian, 59
- H**
Hamiltonian, 10
Hamilton's principle, 70
Hartre-Fock
 operator, 241
Hartree, Douglas Rayner, 234
Hartree-Fock approximation, 241
Higgs boson, 59
- I**
Importance sampling, 18
Interaction
 strong, 59
 weak, 59
IR, 5
Isospin, 69
- K**
Kepler, Johannes, 13
Kernighan, Brian, 21
- L**
Lagrangian
 density, 75
 function, 71
Lagrangian density, 74, 76
LCAO, 239
Leibnitz, Willhelm Gottfried, 4
Length scale, 3
LISP, 81
Lyapunov instability, 188
- M**
Mach, Ernst, 31
Macroscale, 10, 49, 50
Macroscopic scale, *see* Macroscale
Manifold
 differentiable, 138
- Mean-field theory, 10
Meitner, Lise, 14
Mesh
 distortion problem, 10
Mesh-based, 10
Mesoscale, 10, 49, 50
Method
 deductive, 14
 inductive, 13
 mesh-based, 10
 particle-based, 10
Metropolis Algorithm, *see* Algorithm
Metropolis, Nicolas, 18, 106
Microscale, 49, 50
Microscopic scale, 9
Microstructure, 50
Minkowski
 coordinates, 51
 metric, 51, 52
 space, 52, 74
Minkowski space, *see* Space
Minkowski, Hermann, 51
Møller-Plesset perturbation theory, 241
- N**
Nanoscale, 49
Neumann, John von, 18
Newton, Isaak, 3, 39
NMR, 5
Noether's theorem, 74
- P**
Pair correlation function, 12
Pauli's exclusion principle, 61
Pauli, Wolfgang, 14
Planck, Max, 4
Positron, 60
Pre-processing, 16
Principia, 4, 39
Principle of relativity, 60
Proper time, 55
- Q**
Quantum
 chemistry, 9
 theory, 4
Quantum theory, *see* Theory
Quark, 30, 42, 59–61
 model, 63

R

Relativity theory, *see* Theory

Ritchie, Dennis, 21

S

Schrödinger equation, 60, 73, 102

Schwinger, Julian, 63

Semi-empirical

approach, 6

Sheldrake, Rupert, 57

Simulation

atomistic, 49

Smoothed particle hydrodynamics, *see* SPH

Sobolev space, 109

Space

Minkowski, 51

Spacetime, 4, 46, 52, 55, 74, 110

coordinate, 55

diagram, 52

interval, 55, 67

manifold, 55

SPH, 10

Spinor, 73

Strain, 353

Structure function, 12

T

Tensor

metric, 352

Ricci, 44

stress, 44

Theory

of general relativity, 44, 51, 56, 60

of special relativity, 44, 51, 52, 56

quantum, 45, 60

Timestep, 26

Top-down approach, 34

Turing machine, 32

U

UV, 5

V

Venn-Diagram, 117

Verlet, Loup, 24

W

Worldline, 52