

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

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*,

$$\left(p - \frac{a}{V^2}\right)(V - b) = Nk_B T, \quad (1.1)$$

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 $F(T, V) = -k_B T \ln Z(T, V)$ obtained from the classical partition function,

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

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

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

where $\beta = 1/(k_B T)$ 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),

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

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

$$= \frac{N k_B T}{V - b} + \frac{a}{V^2} . \quad (1.7)$$

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:

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

$$=: U_{\text{pot}} . \quad (1.9)$$

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

$$\ln Z_{\text{pot}}(\beta, V) = \ln Z_{\text{pot}}(\infty, V) - \int_0^\beta d\beta' U_{\text{pot}}(\beta', V) , \quad (1.10)$$

$$p = k_B T \frac{\partial}{\partial V} \left[N \ln \frac{V}{(\sqrt{2\pi\hbar})^3} - \int_0^\beta d\beta' U_{\text{pot}}(\beta', V) \right] . \quad (1.11)$$

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),

$$p = \frac{k_B T N}{V} - k_B T \int_0^\beta d\beta' \frac{\partial U_{\text{pot}}}{\partial V} . \quad (1.12)$$

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

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

using boldface types for three-dimensional vectors as the position vector \mathbf{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

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

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

where the integration constant vanishes at infinite volume.

We can approximately get some insight from the analytical point of view. At high temperature $\beta \rightarrow 0$ 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 ,

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

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

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

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 ($\lambda \rightarrow \infty$) before proceeding to the limit of large volume. The former gives the van der Waals law with

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

where a is determined by the constants of the interaction potential. The factor $1/2N(N-1)$ 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 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

$$v := \frac{V}{\lambda^3}, \quad (1.20)$$

$$\alpha := \frac{v_0}{k_B T \lambda} = \frac{v_0 \beta}{\lambda}, \quad (1.21)$$

$$u(\alpha, v) := \frac{\lambda}{N v_0} U_{\text{pot}}(\beta, v), \quad (1.22)$$

which leads with (1.14) to

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

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

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

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

for $T \rightarrow \infty$.

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

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

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

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.DO/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.DO
AVRHO(1:NDIV,1:NDIV,1:NDIV) = 0.DO
LOCEN = 0.DO
ERWEN = 0.DO
VAREN = 0.DO
ERWKIN = 0.DO
VARKIN = 0.DO
ERWPOT = 0.DO
VARPOT = 0.DO
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) > 0.5DO*LENGTH)
CC      )          RNEU(i,IE)=LENGTH-RNEU(i,IE)
CC      IF (RNEU(i,IE) < -0.5DO*LENGTH)
CC      )          RNEU(i,IE)=-LENGTH-RNEU(i,IE)
CCC Scatter particle at a rough wall
CC      if (DABS(RNEU(i,IE)) > 0.5DO*LENGTH) THEN
CC      call GENRAN(rannumb)
CC      if (RNEU(i,IE) > 0.5DO*LENGTH)
CC      &      RNEU(i,IE)=LENGTH/2.DO-ABS(rannumb-0.5)*STEPMAX
CC      &      RNEU(i,IE)=LENGTH*(1/2.DO-rannumb) ! totally unbiased
CC      if (RNEU(i,IE) < -0.5DO*LENGTH)
CC      &      RNEU(i,IE)=-LENGTH/2.DO+ABS(rannumb-0.5)*STEPMAX
CC      &      RNEU(i,IE)=-LENGTH*(1/2.DO-rannumb) ! totally unbiased
C Apply strict cut-off boundary conditions
      if (DABS(RNEU(i,IE)) > 0.5DO*LENGTH) THEN
MCSCHRITT = .false.
RNEU(1:3,IE) = RE(1:3,IE)
MCCOUNT = MCCOUNT + 1
      cycle lelrunpre
    end do
  end do
end do

```

```

        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.DO+ABS(rannumb-0.5)*STEPMAX
CC      &          RNEU(i,IE)=-LENGTH*(1/2.DO-rannumb) ! totally unbiased
C      Apply strict cut-off boundary conditions
        if (DABS(RNEU(i,IE)) > 0.5DO*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)
            MDCOUNT = MDCOUNT + 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
        LOECN = LOECN + 0.5DO*ep
    end do lelrn
C      energy per particle
        LOECN = LOECN/DBLE(NE)
        if (IMC == 2) then !set start values
            ERWEN = LOECN
            VAREN = 0.DO
        end if
        ERWEN = (IMC-1)/DBLE(IMC)*ERWEN+LOECN/DBLE(IMC)
        VAREN = (IMC-1)/DBLE(IMC)*VAREN +
&          1/DBLE(IMC-1)*(ERWEN-LOECN)**2
        LOECN = 0.DO
        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 $GENRAN(\text{rannumb})$. 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 $\pm \text{STEPMAX}/2$. 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.

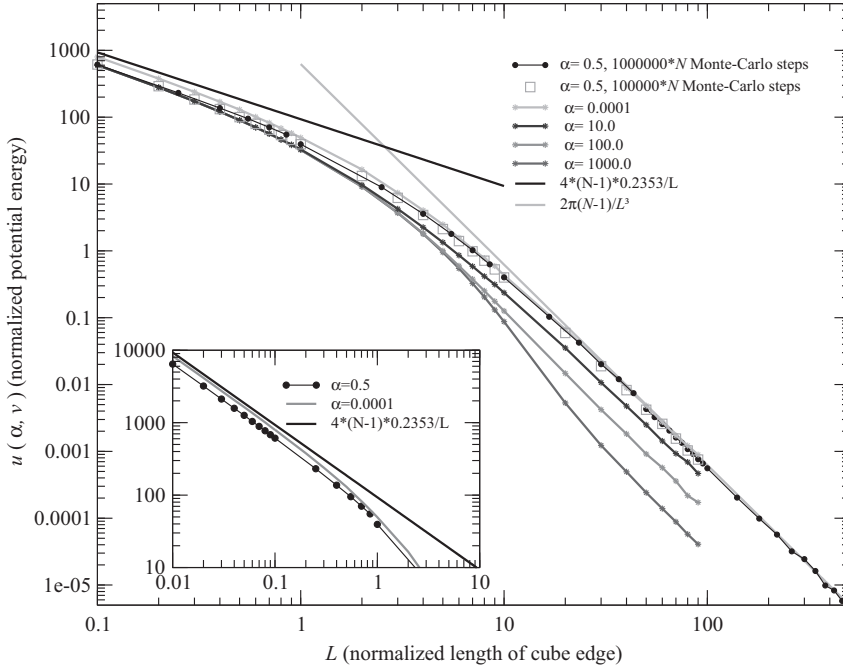


Figure 1.1 Van der Waals pressure correction to ideal gas equation of state according to (1.22) in logarithmic representation, $u(\alpha, \nu)$ vs. $L = \nu^{1/3}$. Note that here the edge length L of the cubic container volume is already scaled by the screening length λ of the interaction potential. Thick lines without symbols show analytical results $2\pi(N-1)/L^3$ and $4(N-1)0.2353/L$ for high temperature $\alpha \rightarrow 0$ at both asymptotic limits $\nu \rightarrow \infty$ and $\nu \rightarrow 0$, respectively; dots connected by thin solid lines refer to MC simulations with $\alpha = 0.5$ showing separate calculation

sets with different maximum step widths $\text{STEPMAX} = 0.001, 0.01, 0.1$, 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.DO/3.DO*EMACH
        IF (work02 .lt. EMACH) work02 = 2.DO/3.DO*EMACH
        DIST(4,IE,k) = work01
        DIST(4,k,IE) = work01
        DISTNEW(4,IE,k) = work02
        DISTNEW(4,k,IE) = work02
        work1=work1+1.DO/work02*dexp(-work02)-1.DO/work01*dexp(-work01)
        work2=work2+1.DO/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, $\alpha \rightarrow 0$. 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 $u = 0.41139$, $\sigma^2 = 8.8 \times 10^{-4}$ for $L = 10$ with $\alpha = 0.5$, $\text{MCMAX} = 10^6$, $\text{NE} = 100$ 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 0.00006 for the latter, yields for the lengths $\text{LENGTH} = 1$ and 100 a 95% probability that this and the other three generators are equivalent. For $\text{LENGTH} = 10$ 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	5.17891×10^{-4}	5.19690×10^{-4}	5.19643×10^{-4}	5.18382×10^{-4}
σ^2	6.37×10^{-7}	6.53×10^{-7}	6.52×10^{-7}	6.44×10^{-7}
$L = 10$				
u	0.37241	0.37257	0.37286	0.37316
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,

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

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 ($\text{NDIV} = 21$) 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 $\text{DENSITY}(\text{RHO})$ 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>plot "thermod_erw.dat" using 1:2:4 index 9 with lines notitle
.....
gnuplot>unset contour
gnuplot>plot "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 $v_0\beta$. 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.

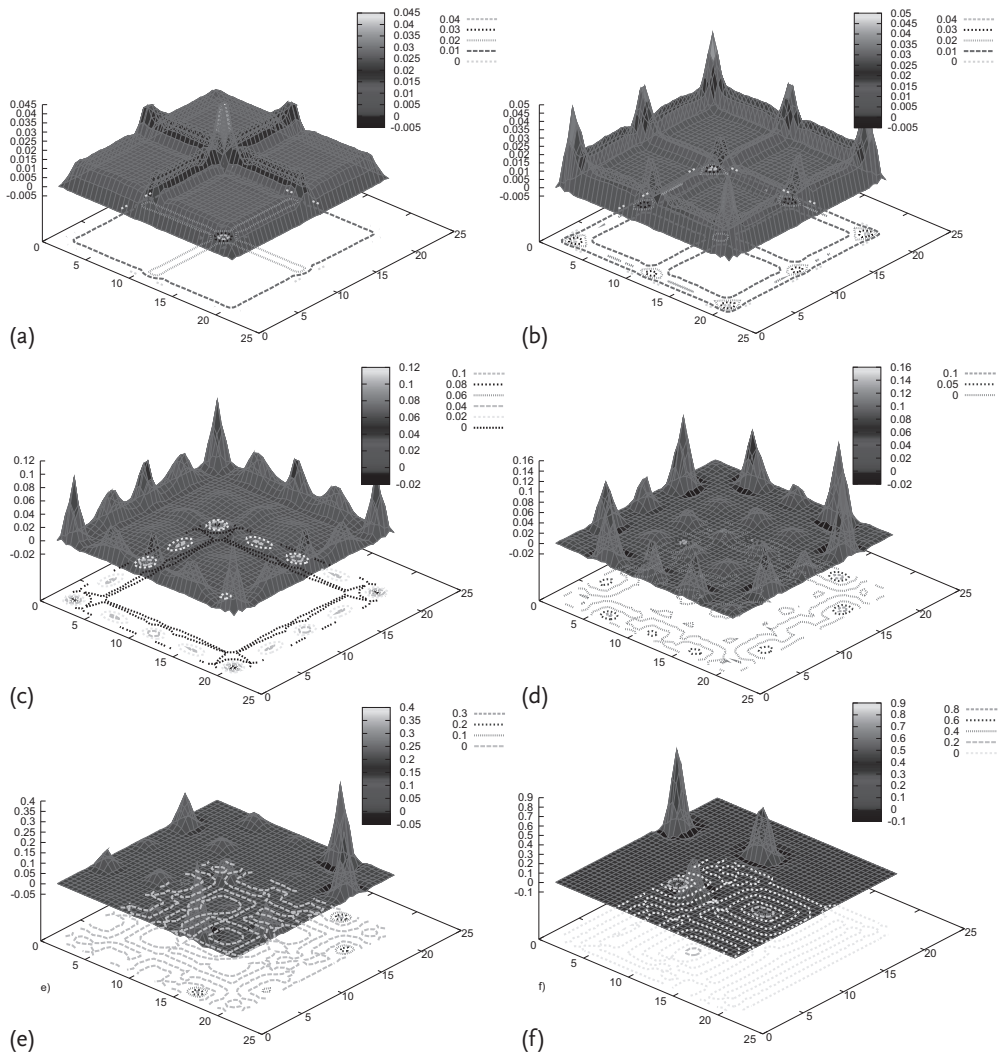


Figure 1.2 Density distributions horizontally cut through the cube center for edge length $\text{LENGTH} = 10$ with temperature parameter α equal to (a) 0.00005, (b) 10.0, (c) 100.0, (d) 500.0, (e) 1000.0 and (f) 5000.0 plotted as surface plot above and contour plot below.