as our model system. We use a truncated and shifted potential (see also section 3.2.2):

$$u^{tr-sh}(r) = \begin{cases} u^{lj}(r) - u^{lj}(r_c) & r \leq r_c \\ 0 & r > r_c \end{cases},$$

where $u^{lj}(r)$ is the Lennard-Jones potential and for these simulations $r_c = 2.5\sigma$ is used.

Case Study 4 (Static Properties of the Lennard-Jones Fluid)
Let us start a simulation with 108 particles on a simple cubic lattice. We give the system an initial temperature $T = 0.728$ and density $\rho = 0.8442$, which is close to the triple (gas-liquid-solid) point of the Lennard-Jones fluid [81–83].

In Figure 4.3 the evolution of the total energy, kinetic energy, and potential energy is shown. It is important to note that the total energy remains constant and does not show a (slow) drift during the entire simulation. The kinetic and potential energies do change initially (the equilibration period) but during the end of the simulation they oscillate around their equilibrium value. This figure shows that, for the calculation of the average potential energy or kinetic energy, we need approx. 1000 time steps to equilibrate the simulation. The figure also shows significant fluctuations in the potential energy, some of which may take several (100) time steps before they disappear.

Appendix D shows in detail how to calculate statistical error in the data of a simulation. In this example, we use the method of Flyvbjerg and Petersen [84]. The following operations on the set of data points are performed: we start by calculating the standard deviation of all the data points, then we group two consecutive data points and determine again the standard deviation of the new, blocked, data set. This new data set contains half the number of data points of the original set. The procedure is repeated until there are not enough data points to compute a standard deviation; the number of times we perform this operation is called M. What do we learn from this?

First of all, let us assume that the time between two samples is so large that the data points are uncorrelated. If the data are uncorrelated the standard deviation (as calculated according to the formula in Appendix D, i.e., correcting for the fact we have fewer data points) should be invariant to this blocking operation and we should get a standard deviation that is independent of M. In a simulation, however, the time between two data points is usually too short to obtain a statistically independent sample; as a consequence consecutive data points would be (highly) correlated. If we would calculate a standard deviation using these data, this standard deviation will be too optimistic. The effect of the block operation will be that after grouping two consecutive data points, the correlation between the two new data points will be less. This, however, will increase the standard deviation; the data will have more noise since consecutive data points no longer resemble

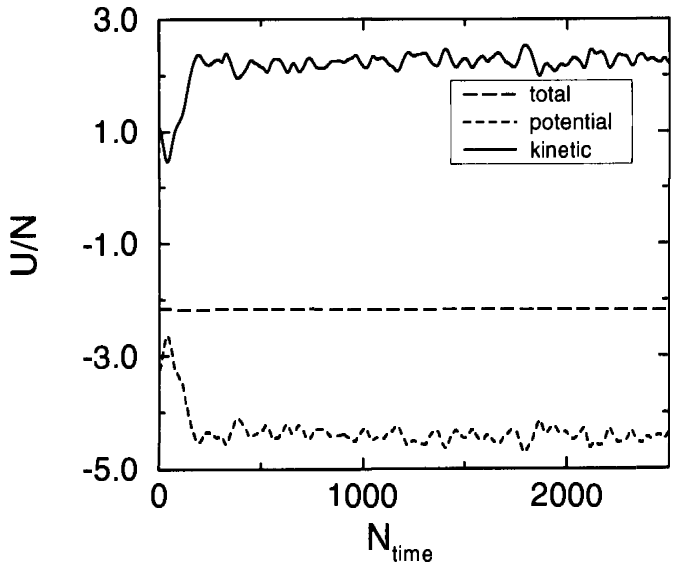

Figure 4.3: Total, potential, and kinetic energy per particle U/N as a function of the number of time steps N_{time}.

each other that closely. This decrease of accuracy as a function of the number of blocking operations will continue until we have grouped so many data points that two consecutive points are really uncorrelated. This is exactly the standard deviation we want to determine. It is important to note that we have to ensure that the standard deviations we are looking at are significant; therefore, we have to determine the standard deviation of the error at the same time.

The results of this error calculation for the potential energy are shown in Figure 4.4, as expected, for a low value of M; the error increases until a plateau is reached. For high values of M, we have only a few data points, which results in a large standard deviation in the error. The advantage of this method is that we have a means of finding out whether we have simulated enough; if we do not find such a plateau, the simulation must have been too short. In addition we find a reliable estimate of the standard deviation. The figure also shows the effect of increasing the total length of the simulation by a factor of 4; the statistical error in the potential energy has indeed decreased by a factor of 2.

In this way we obtained the following results. For the potential energy $U = -4.4190 \pm 0.0012$ and for the kinetic energy $K = 2.2564 \pm 0.0012$, the latter corresponds to an average temperature of $T = 1.5043 \pm 0.0008$. For the pressure, we have obtained 5.16 ± 0.02.

In Figure 4.5, the radial distribution function is shown. To determine this function we used Algorithm 7. This distribution function shows the characteristics of a dense liquid. We can use the radial distribution function to calculate the energy and pressure. The potential energy per particle can be

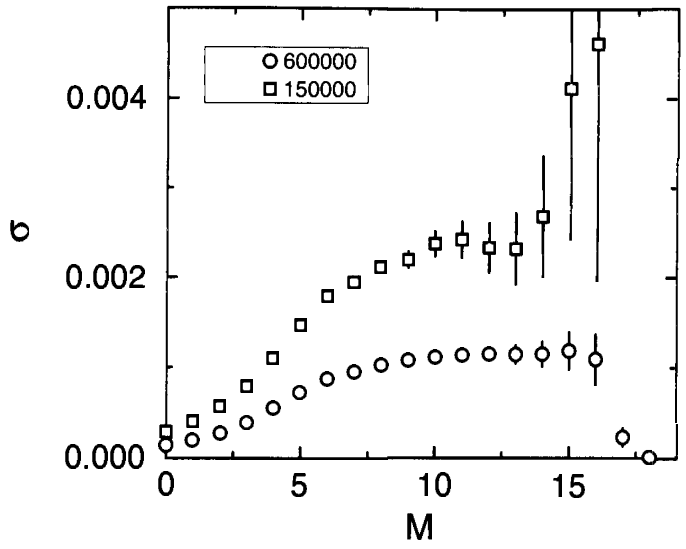

Figure 4.4: The standard deviation σ in the potential energy as a function of the number of block operations M for a simulation of 150,000 and 600,000 time steps. This variance is calculated using equation (D.3.4).

calculated from

$$\begin{aligned} U/N &= \frac{1}{2}\rho \int_0^\infty dr\, u(r)g(r) \\ &= 2\pi\rho \int_0^\infty dr\, r^2 u(r)g(r) \end{aligned} \tag{4.5.1}$$

and for the pressure from

$$\begin{aligned} P &= \rho k_B T - \frac{1}{3}\frac{1}{2}\rho^2 \int_0^\infty dr\, \frac{du(r)}{dr} r g(r) \\ &= \rho k_B T - \frac{2}{3}\pi\rho^2 \int_0^\infty dr\, \frac{du(r)}{dr} r^3 g(r), \end{aligned} \tag{4.5.2}$$

where $u(r)$ is the pair potential.

Equations (4.5.1) and (4.5.2) can be used to check the consistency of the energy and pressure calculations and the determination of the radial distribution function. In our example, we obtained from the radial distribution function for the potential energy $U/N = -4.419$ and for the pressure $P = 5.181$, which is in good agreement with the direct calculation.

Case Study 5 (Dynamic Properties of the Lennard-Jones Fluid)
As an example of a dynamic property we have determined the diffusion coefficient. As shown in the previous section, the diffusion coefficient can be determined from the mean-squared displacement or from the velocity autocorrelation function. We have determined these properties using Algorithm 8.

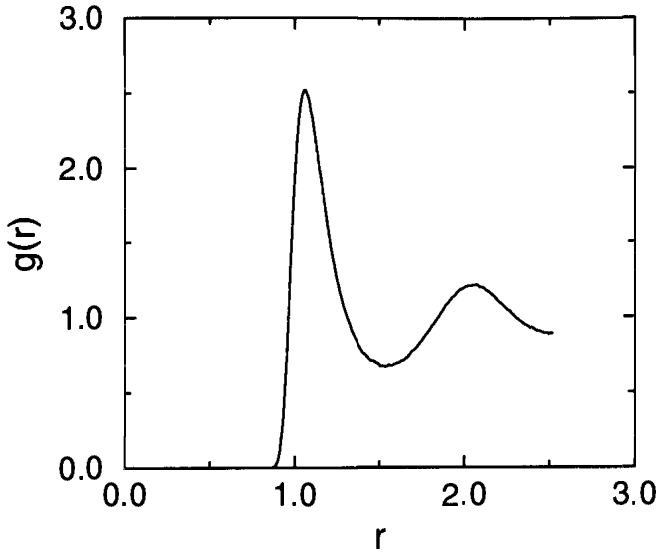

Figure 4.5: Radial distribution function of a Lennard-Jones fluid close to the triple point: $T = 1.5043 \pm 0.0008$ and $\rho = 0.8442$.

In Figure 4.6 the mean-squared displacement is shown as a function of the simulation time. From the mean-squared displacement we can determine the diffusion using equation (4.4.9). This equation, however, is valid only in the limit $t \rightarrow \infty$. In practice this means that we have to verify that we have simulated enough that the mean-squared displacement is really proportional to t and not to another power of t.

The velocity autocorrelation function can be used as an independent route to test the calculation of the diffusion coefficient. The diffusion coefficient follows from equation (4.4.11). In this equation we have to integrate to $t \rightarrow \infty$. Knowing whether we have simulated sufficiently to perform this integration reliably is equivalent to determining the slope in the mean-squared displacement. A simple trick is to determine the diffusion coefficient as a function of the truncation of the integration; if a plateau has been reached over a sufficient number of integration limits, the calculation is probably reliable.

Case Study 6 (Algorithms to Calculate the Mean-Squared Displacement) In this case study, a comparison is made between the conventional (Algorithm 8) and the order-n methods (Algorithm 9) to determine the mean-squared displacement. For this comparison we determine the mean-squared displacement of the Lennard-Jones fluid.

In Figure 4.7 the mean-squared displacement as a function of time as computed with the conventional method is compared with that obtained from the order-n scheme. The calculation using the conventional scheme could not be extended to times longer than $t > 10$ without increasing the number of

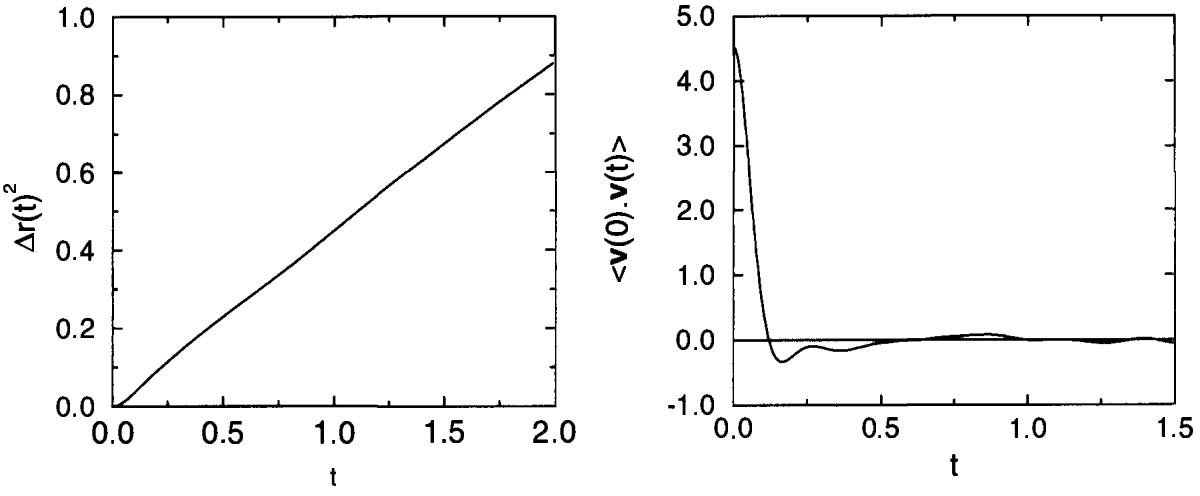

Figure 4.6: (left) Mean-squared displacement $\Delta r(t)^2$ as a function of the simulation time t. Note that for long times, $\Delta r(t)^2$ varies linearly with t. The slope is then given by 2dD, where d is the dimensionality of the system and D the self-diffusion coefficient. (right) Velocity autocorrelation function $\langle \mathbf{v}(0) \cdot \mathbf{v}(t) \rangle$ as a function of the simulation time t.

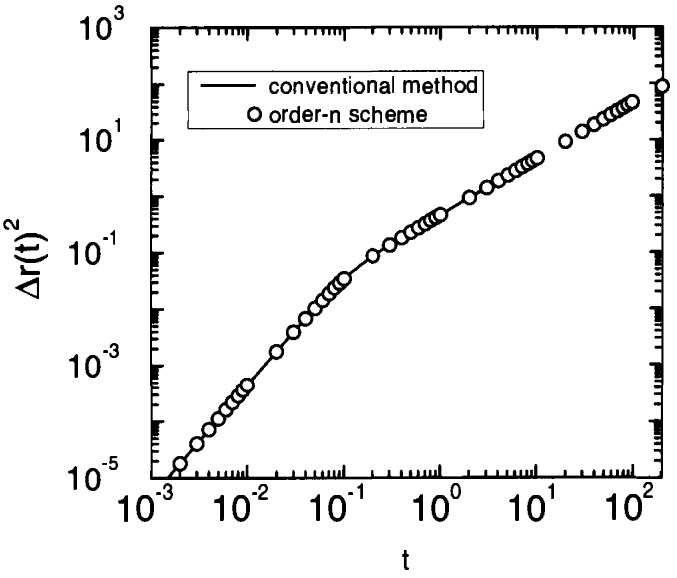

Figure 4.7: Mean-squared displacement as a function of time for the Lennard-Jones fluid (ρ = 0.844, N = 108, and T = 1.50); comparison of the conventional method with the order-n scheme .

time steps between two samples because of lack of memory. With the order-n scheme the calculation could be extended to much longer times with no difficulty. It is interesting to compare the accuracy of the two schemes. In the conventional scheme, the velocities of the particles at the current time step are used to update the mean-squared displacement of all time intervals. In

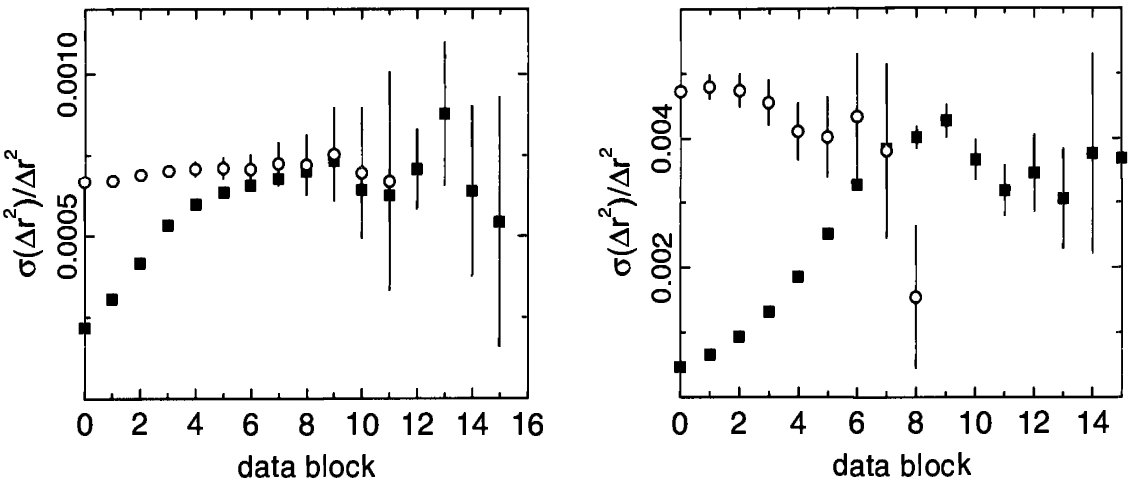

Figure 4.8: Relative error in the mean-squared displacement as a function of the number of data blocks as defined by Flyvbjerg and Petersen. The figures compare the conventional scheme (solid squares) with the order-n method (open circles) to determine the mean-squared displacement. The right figure is for $t = 0.1$ and the left figure for $t = 1.0$.

the order-n scheme the current time step is only used to update the lowest-order array of $\mathbf{v}_{\text{sum}}$ (see Algorithm 9). The block sums of level i are updated only once every n^i time step. Therefore, for a total simulation of M time steps, the number of samples is much less for the order-n scheme; for the conventional scheme, we have M samples for all time steps, whereas the order-n scheme has M/n^i samples for the ith block velocity. Naively, one would think that the conventional scheme therefore is more accurate. In the conventional scheme, however, the successive samples will have much more correlation and therefore are not independent. To investigate the effect of these correlations on the accuracy of the results, we have used the method of Flyvbjerg and Petersen [84] (see Appendix D.3 and Case Study 4). In this method, the standard deviation is calculated as a function of the number of data blocks. If the data are correlated, the standard deviation will increase as a function of the number of blocks until the number of blocks is sufficient that the data in a data block are uncorrelated. If the data are uncorrelated, the standard deviation will be independent of the number of blocks. This limiting value is the standard deviation of interest.

In these simulations the time step was $\Delta t = 0.001$ and the block length was set to $n = 10$. For both methods the total number of time steps was equal. To calculate the mean-squared displacement, we have used 100,000 samples for all times in the conventional scheme. For the order-n scheme, we have used 100,000 samples for $t \in [0, 0.01]$, 10,000 for $t \in [0.01, 0.1]$, 1,000 for for $t \in [0.1, 1]$, etc. This illustrates that the number of samples in the order-n scheme is considerably less than in the conventional scheme. The

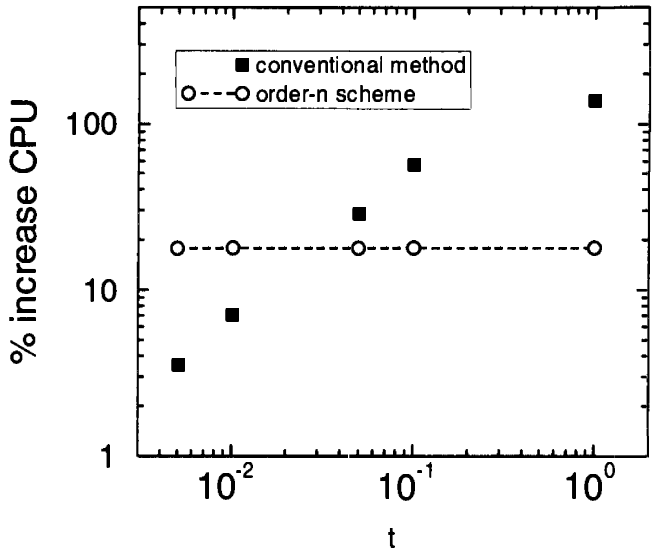

Figure 4.9: Percentage increase of the total CPU time as a function of the total time for which we determine the mean-squared displacement; comparison of the conventional scheme with the order-n scheme for the same system as is considered in Figure 4.7

accuracy of the results, however, turned out to be the same. This is shown in Figure 4.8 for $t = 0.1$ and $t = 1.0$. Since the total number of data blocking operations that can be performed on the data depends on the total number of samples, the number of blocking operations is less for the order-n method. Figure 4.8 shows that for $t = 0.1$ the order-n scheme yields a standard deviation that is effectively constant after three data blocking operations, indicating the samples are independent, whereas the standard deviation using the conventional method shows an increase for the first six to eight data blocking operations. For $t = 1.0$ the order-n method is independent of the number of data blocks, the conventional method only after 10 data blocks. This implies that one has to average over $2^{10} \approx 1000$ successive samples to have two independent data points. In addition, the figure shows that the plateau value of the standard deviation is essentially the same for the two methods, which implies that for this case the two methods are equally accurate.

In Figure 4.9 we compare the CPU requirements of the two algorithms for simulations with a fixed total number of time steps. This figure shows the increase of the total CPU time of the simulation as a function of the total time for which the mean-squared displacement has been calculated. With the order-n scheme the CPU time should be (nearly) independent of the total time for which we determine the mean-squared displacement, which is indeed what we observe. For the conventional scheme, however, the required CPU time increases significantly for longer times. At $t = 1.0$ the order-n scheme gives an increase of the total CPU time of 17%, whereas the conventional scheme shows an increase of 130%.