

III. Notes: Starting the Simulation

At this point in the course we know how to calculate forces based on model potentials. We also know how to solve the equations of motion that use those potentials to predict positions and velocities of particles in the future. At the same time, we know how to obtain thermodynamic properties from a running simulation and how to characterize structure based on positions, calculate response functions, and determine the diffusion coefficient based on velocities. However, we still do not know how to implement the technical steps that permit the initiation of a simulation. These steps are essential to establish the initial conditions of the system that will eventually yield macroscopic behavior and that will let us obtain sensible results based only on a small number of particles.

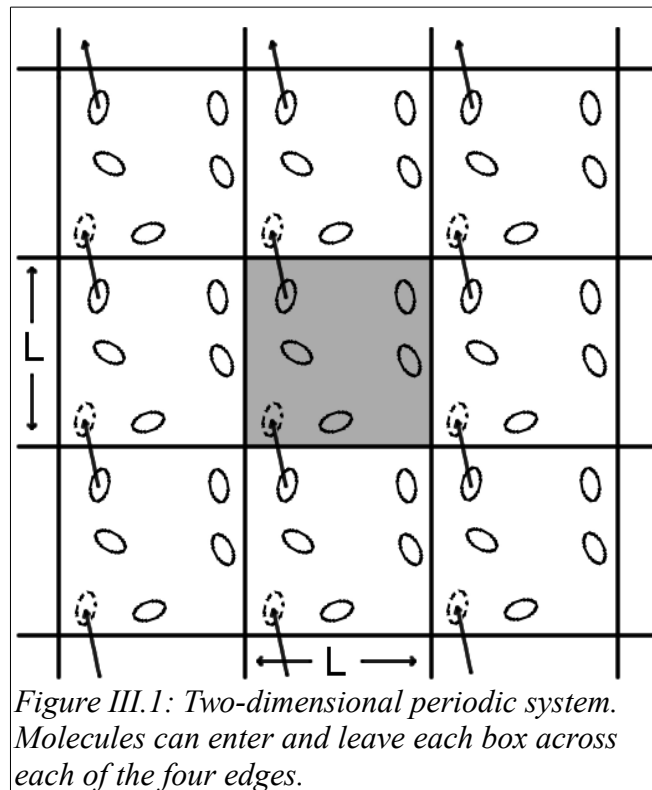
In the following sections, we address these technical issues and discuss the different possibilities and situations to which they are applicable.

1. *Periodic Boundary Conditions*

Computer simulations are by construction performed on a small number of particles, typically in the order from a few tens of thousands to much larger at several hundreds of thousands - a number very small when compared with real systems with Avogadro's number of particles. This small number comes as a consequence of limited storage on computers, but more importantly, from the relatively slow speeds of execution of our programs. Without any special consideration the time taken for a double loop used to evaluate the forces or potential energy is proportional to N^2 . Special techniques, to be discussed later, may reduce this dependence to order N for very large systems, but still the force/energy loop inevitably dictates the overall speed.

Size enters very crucially when trying to simulate a bulk system using a finite system. The number of molecules lying at or near the surface of the simulation container will always be too high, compared to the total, for the properties of the system to be close to those of the bulk. The physics to be obtained from such a system, unless we are dealing with truly confined systems, will not correspond to those desired because molecules on the surface will experience quite different forces from the molecules inside the box. Thus, surface effects overwhelm the physics of the system.

The problem of surface effects in simulations can be tackled by implementing periodic boundary conditions (PBC). Take, for example, a repeated simulation cubic box in space that forms an infinite lattice. In the course of the simulation, as a molecule moves in the original box, its periodic image in each of the neighboring boxes moves in exactly the same way. Thus, as a molecule leaves the central box one of its images will enter through the opposite face. There are no walls at the boundary of the central box and no surface molecules. This box simply forms a convenient axis system for measuring the coordinates of the N molecules. In Figure III.1, as a particle moves through a boundary, its images, also move across their corresponding boundaries. The number density in the central box is conserved. It turns out that it is not necessary to store the coordinates of all the images in a simulation, just the molecules in the central box. When a molecule leaves the box by crossing a boundary, attention may be switched to the image just entering.

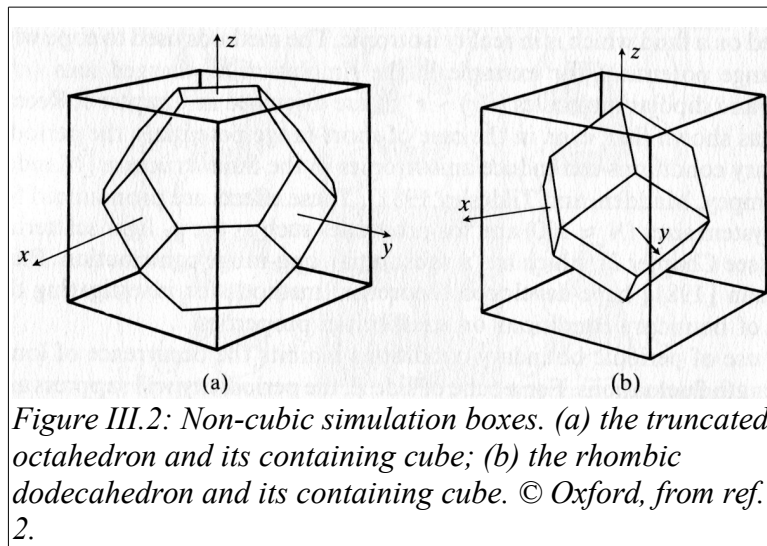


The question arises whether this technique will be able to reproduce the macroscopic system that it aims to represent. This will depend both on the range of the intermolecular potential and the phenomenon under investigation. For a fluid of LJ atoms, it should be possible to perform a simulation in a cubic box of side $L \approx 6\sigma$ without a particle being able to “feel” the symmetry of the periodic lattice.

The use of PBC, however, inhibit the very important occurrence of long-wavelength fluctuations. For a cube of side L , the periodicity will suppress any density waves with a wavelength greater than L . Thus, it would not be possible to simulate a liquid close to the gas-liquid critical point where the range of critical fluctuations is macroscopic. Furthermore, transitions which are known to be first order often exhibit the characteristics of higher order transitions when modeled in a small box because of the suppression of fluctuations.

Despite of the above caveats, experience shows that in simulations, periodic boundary conditions have little effect on the equilibrium thermodynamic properties and structures of fluids away from phase transitions and when the interactions are short-ranged.

The cubic box geometry is the most frequently used in computer simulations because of its simplicity to implement. Of the four remaining semiregular space-filling polyhedra, the rhombic dodecahedron and the truncated octahedron shown in Figure III.2, are also used. These two are more “spherical” than the cube which may be useful for simulating liquids whose structure is spatially isotropic. In addition, for a given number density, the distance between periodic images is larger than in the cube. This property is useful in calculating distribution functions and structure factors to a larger distance extent.



External fields can be accommodated to PBC provided the symmetry of this external field is reflected in the periodic lattice. For example, in Figure III.3 the PBC for a gas adsorbed to a graphite surface is shown. In (b), in a top view, we see a periodic system with a rhombic shape that conforms to the geometry of the graphite surface.

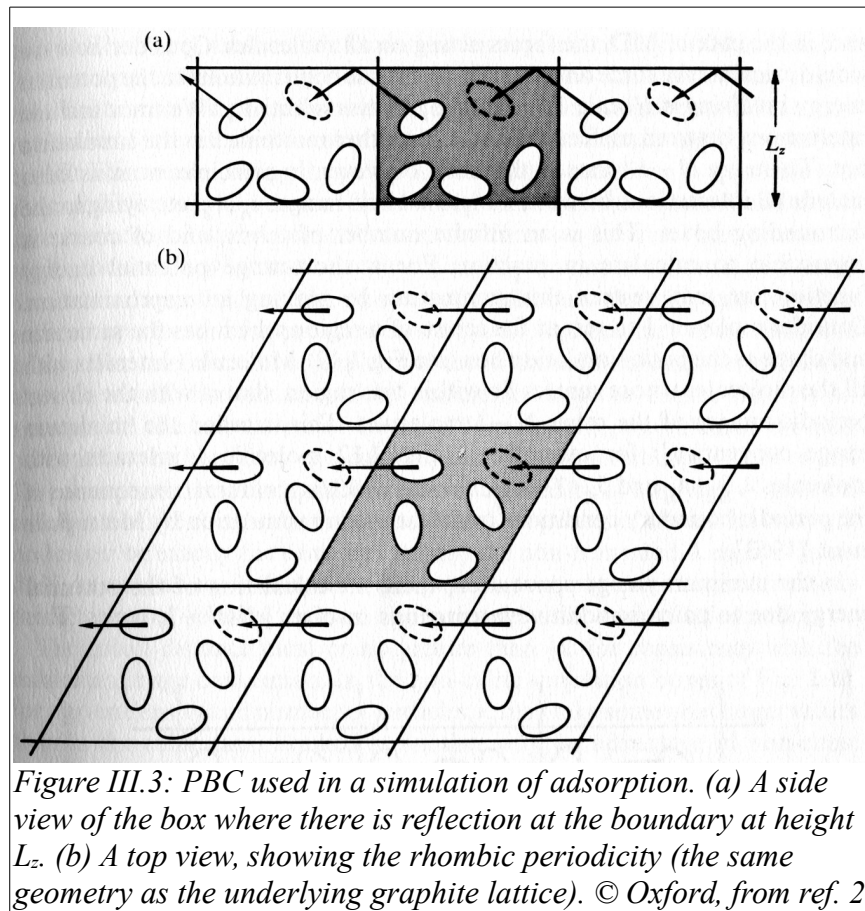


Figure III.3: PBC used in a simulation of adsorption. (a) A side view of the box where there is reflection at the boundary at height L_z . (b) A top view, showing the rhombic periodicity (the same geometry as the underlying graphite lattice). © Oxford, from ref. 2

In (a), in contrast, the geometry is not periodic, where the gas close to the surface is adsorbed while the gas farther than the surface monolayer and more mobile can move, but if it hits the top surface it is reflected where the vertical component of the velocity is reversed.

2. Initial Configurations

Random positions: The simplest method of constructing a liquid structure is to place molecules at random inside the simulation box. The difficulty with this technique is that the configuration so constructed may contain substantial overlaps. This would be totally unphysical for a hard-core system. For soft potentials, the energy for most random configurations, though high, can be calculated, but too high numbers could cause difficulties in the solution of the stiff differential equations of motion. To avoid overlap problems, then for every new particle (when constructing the system by introducing one particle at a time) one should calculate the total forces on that particle (and thus the energy) at the proposed random position and compare that number to a previously chosen threshold force (energy) beyond which that position would be rejected. In this way one can ensure that magnitudes of forces and energies are numerically manageable thus avoiding instabilities and too high velocities that would destroy the simulation. One note is that the convergence of this method (i.e. the time that it takes to assemble all particles) could be very slow in dependence of the threshold energy and also on the desired density. If the first is too small and the second too high, this scheme will not be able to accommodate all particles in the initial configuration. A benefit is that random starting positions reduce

equilibration time when simulating dense liquids.

Lattice positions: The other usually chosen method is to put the initial particles at the vertices of a lattice inside the simulation box. Almost any lattice is suitable, but historically in 3D the face-centered cubic structure lattice points has been the starting configuration of choice. During the course of the simulation the lattice structure will disappear, to be naturally replaced by a typical liquid structure. This process of 'melting' can be enhanced by giving each molecule a small random displacement from its initial lattice point along each of the space-fixed axes.

3. Initial Velocities

For a MD simulation, the initial velocities of all the molecules must be specified. It is usual to choose random velocities uniformly distributed in a range $(-v_{\max}, +v_{\max})$, with magnitudes conforming to the required temperature, corrected so that there is no overall momentum, as specified in equation 1

$$\mathbf{P} = \sum_{i=1}^N m_i \mathbf{v}_i = 0 . \quad (1)$$

Alternatively, one may choose to randomly assign velocities drawn from a Gaussian distribution, for example, the one shown in equation 2

$$\rho(v_{ix}) = \sqrt{\frac{m_i}{2\pi k_B T}} e^{(-\frac{1}{2} m_i v_{ix}^2 / k_B T)} \quad (2)$$

where $\rho(v_{ix})$ is the probability density for velocity component v_{ix} . Similar equations apply for the y and z components.

Note that even if we start from uniformly distributed random velocities, the distribution will quickly approach that given by the Maxwell distribution.

3.1. A Note on Gaussian distributions

When generating random numbers belonging to a Gaussian distribution $p(x)$ as in the probability distribution function of equation 3,

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left[-\frac{(x - \langle x \rangle)^2}{2\sigma^2} \right] \quad (3)$$

with mean $\langle x \rangle$ and variance $\sigma^2 (= \langle (x - \langle x \rangle)^2 \rangle)$, these can usually be generated by using random numbers belonging to a uniformly distributed distribution.

One option is to first generate two random numbers ξ_1 and ξ_2 between 0 and 1. The corresponding two numbers from the normal distribution are then calculated using

$$\begin{aligned}x_1 &= \sqrt{-2 \ln \xi_1} \cos(2\pi \xi_2) \\x_2 &= \sqrt{-2 \ln \xi_1} \sin(2\pi \xi_2)\end{aligned}\quad (4)$$

This particular form of the transformation, however, has two problems:

1. it is slow because of many calls to the math library,
2. it can have numerical instability problems when ξ_1 is very close to zero.

A better transformation that is both faster and more robust is given algorithmically by:

```
float x1, x2, w, y1, y2;
do {
    x1 = 2.0 * ran() - 1.0;
    x2 = 2.0 * ran() - 1.0;
    w = x1 * x1 + x2 * x2;
} while ( w >= 1.0 );

w = sqrt( (-2.0 * ln( w ) ) / w );
y1 = x1 * w;
y2 = x2 * w;
```

where `ran()` gives random numbers uniformly distributed in the range $[0,1]$. This form is faster because it does the equivalent of the sine and cosine geometrically without a call to the trigonometric function library.

Another option is to generate twelve random numbers ξ_1, \dots, ξ_{12} and then calculate equation 5:

$$x = \sum_{i=1}^{12} \xi_i - 6. \quad (5)$$

These two methods generate random numbers in the normal distribution with zero mean and unit variance. A number (x) generated from this distribution can be related to its counterpart (x') from another Gaussian distribution with mean $\langle x' \rangle$ and variance σ using:

$$x' = \langle x' \rangle + \sigma x. \quad (6)$$

4. Energy Minimization

For almost all systems that are considered in simulations, the potential energy is a complicated, multi dimensional function of the coordinates. The way in which the energy depends on the coordinates is usually referred to as the *potential energy surface* (sometimes called the *hypersurface*). For a system of N atoms the energy is thus a function of $3N$ degrees of freedom. It is therefore impossible to visualize the entire energy surface.

In MD we are especially interested in minimum points on the energy surface. Minimum energy arrangements of the atoms correspond to stable states of the system; any movement away from a

minimum gives a configuration with a higher energy. There may be a very large number of minima on the energy surface. The minimum with the very lowest energy is known as the *global energy minimum*. To identify those geometries of the system that correspond to minimum points (local or global) on the energy surface we use a minimization *algorithm*.

Features of interest in the energy surface include system transitions from one minima to another adjacent minima. These transitions could correspond to reactions or structural and conformational changes. The highest point on the pathway between two minima is of special interest and is known as a *saddle point*, with the arrangement of the atoms being the *transition structure*. Both minima and saddle points are stationary points on the energy surface, where the first derivative of the energy function is zero with respect to all the coordinates (but may differ on the sign of the second derivative in particular directions).

The minimization problem can be formally stated as follows: given a function f which depends on one or more independent variables x_1, x_2, \dots, x_i , find the values of those variables where f has a minimum value. At a minimum point, the first derivative of the function with respect to each of the variables is zero and the second derivatives are all positive;

$$\frac{\partial f}{\partial x_i} = 0; \quad \frac{\partial^2 f}{\partial x_i^2} > 0. \quad (7)$$

In MD, f corresponds to the energy function and the variables x_i correspond to Cartesian or internal coordinates of the atoms. Because of the complicated form of the energy with respect to the coordinates, minima are located using numerical methods, which gradually change the coordinates to produce configurations with lower and lower energies until the minimum is reached.

We can classify minimization algorithms into two groups: those which use derivatives of the energy with respect to the coordinates and those which do not. In this course we will use methods that use derivatives. These can provide information about the shape of the energy surface, and, if used properly, they can significantly enhance the efficiency with which the minimum is located. However, because no single minimization method has yet proven to be the best for all molecular modeling problems, most software packages offer a choice of methods.

Whenever possible, analytical forms for the derivatives are preferred. However, under some circumstances it is necessary to use numerical derivatives. These can be calculated as follows. If one of the coordinates x_i is changed by a small change δx_i and the energy for the new arrangement is computed, then the derivative $\partial E / \partial x_i$ is obtained by dividing the change in energy δE by the change in coordinate δx_i . This strictly gives the derivative at the mid-point between the two points x_i and $x_i + \delta x_i$ (i.e. at $x_i + \delta x_i / 2$). A more accurate value of the derivative at the point x_i itself may be obtained (at the cost of additional energy calculation) by evaluating the energy at two points, $x_i + \delta x_i$ and $x_i - \delta x_i$. The derivative is then obtained by dividing the difference in the energies by $2\delta x_i$.

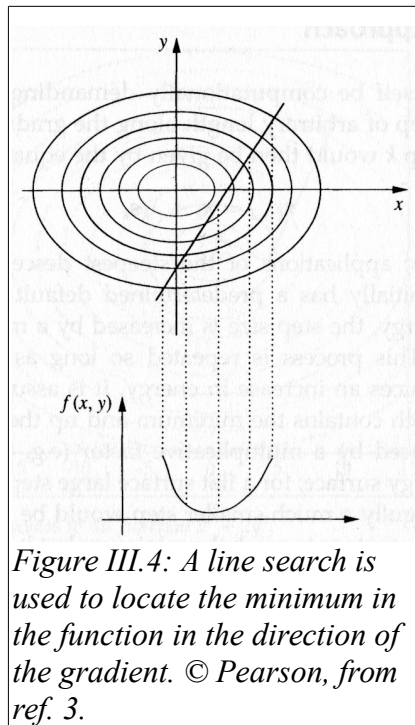
In the following, we present two frequently used first-order (in the derivative of the energy) minimization algorithms, the *steepest descents* and the *conjugate gradient*. These two methods have in common that they gradually change the coordinates of the atoms as they move the system closer and closer to the minimum points. The starting point for each iteration k is the molecular configuration obtained from the previous step, which is represented by the multidimensional vector \mathbf{x}_{k-1} . For the first

iteration the starting point is the initial configuration of the system provided by the user, the vector \mathbf{x}_1 .

4.1. Steepest Descents

The steepest descents method moves in the direction parallel to the net force. For 3N Cartesian coordinates this direction is most conveniently represented by a 3N-dimensional unit vector, \mathbf{s}_k . Thus,

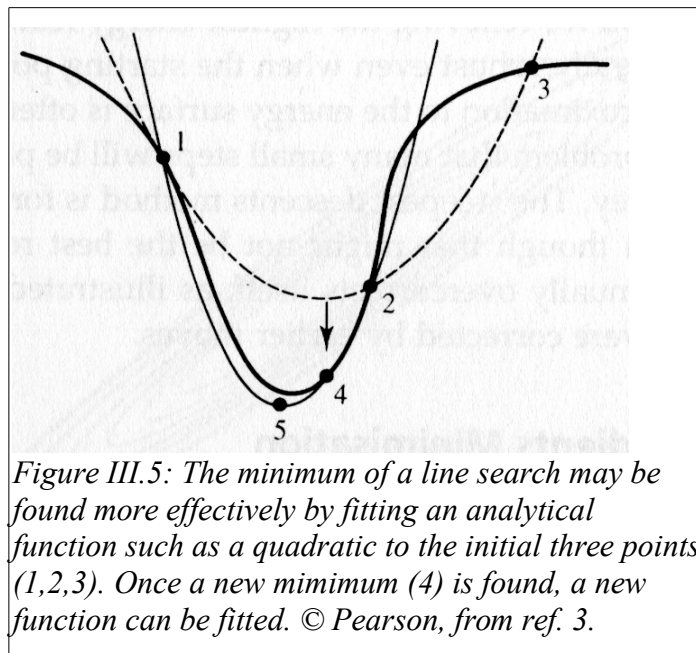
$$\mathbf{s}_k = -\frac{\mathbf{g}_k}{|\mathbf{g}_k|}, \quad (8)$$



where \mathbf{g}_k is the gradient of the potential at point k . Having defined the direction along which to move, then it is necessary to decide how far to move along the gradient. Consider Figure III.4. The gradient direction from the starting point is along the line indicated. If we imagine a cross-section through the surface along the line, the function will pass through a minimum and then increase, as shown in the figure. We can choose to locate the minimum point by performing a *line search* or we can take a step of *arbitrary size* along the direction of the force.

4.1.a Line Search in One Dimension

A line search locates the minimum along a specified direction through multidimensional space. The first stage of the line search is to *bracket* the minimum. This entails finding three points along the line such that the energy of the middle point is lower than the energy of the two outer points. If three such points can be found, then at least one minimum must lie between the two outer points. An iterative procedure can then be used to decrease the distance between the three points, gradually restricting the minimum to an even smaller region. The possibly large number of iterations of this method makes an alternative algorithm more appealing. The alternative is to fit a function, such as a quadratic, to the three bracketing points. Differentiation of the fitted function enables an approximation to the minimum along the line to be identified analytically. A new function can then be fitted to give a better estimate, as shown in Figure III.5. Higher-order polynomials may give a better fit to the bracketing points, but these



can give incorrect interpolations when used with functions that change sharply in the bracketed region.

The gradient at the minimum point obtained from the line search will be perpendicular to the previous direction. Thus, when the line search method is used to locate the minimum along the gradient, the next direction in the steepest descents algorithm will be orthogonal to the previous direction (i.e. $\mathbf{g}_k \cdot \mathbf{g}_{k-1} = 0$).

An example by line search. Consider the function $f(x,y) = x^2 + 2y^2$ illustrated in Figure III.6. We first calculate the direction, given by equation 8, of the negative of the gradient by

$$s_k = -\frac{\nabla f}{|\nabla f|} = -\frac{2x\hat{i} + 4y\hat{j}}{\sqrt{4x^2 + 16y^2}}. \quad (9)$$

If we start from point (9.0,9.0), the direction is given by $s_k = -0.45\hat{i} - 0.9\hat{j}$ that gives a search line with equation $y = 2x - 9$ (slope = 0.9/0.45 and intercept $m = -9$). To find the minimum of $f(x,y)$ along the constraint line y we simply substitute y in $f(x,y)$ to obtain $f(x) = 9x^2 + 192 - 72x$ whose x derivative should be 0 by definition of a stationary point, giving $x=4$. Substituting this value in the equation of the line, we finally obtain a point (4,-1) as the point that minimizes $f(x,y)$ along the direction of steepest descent.

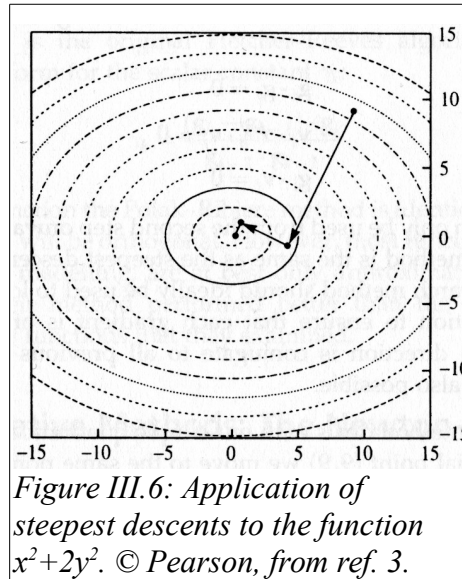


Figure III.6: Application of steepest descents to the function $x^2 + 2y^2$. © Pearson, from ref. 3.

By repeating this procedure on the new starting point of (4,-1) we get that the second next minima is found at (2/3, 2/3) and the third at (0.296, -0.074), as illustrated in Figure III.6.

Because the direction of the gradient is determined by the largest interatomic forces, the steepest descents is a good method for relieving the highest-energy features in an initial configuration. The method is generally robust even when the starting point is far from a minimum, where the harmonic approximation to the energy surface is often a poor assumption. However, it suffers from the problem that many small steps will be performed when proceeding down a long narrow valley. The steepest descents method is forced to make a right-angled turn at each point, even though that might not be the best route to the minimum. The path oscillates and continually over corrects itself, as illustrated in Figure III.7; later steps reintroduce errors that were corrected by earlier moves.

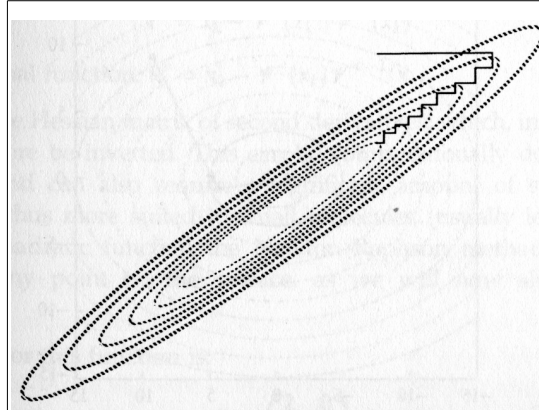


Figure III.7: The steepest descents method can give undesirable behavior in a long narrow valley. © Pearson, from ref. 3.

4.1.b Arbitrary Step Approach

An alternative to the *line search* approach is a less computationally demanding algorithm where we obtain the new coordinates by taking a step of arbitrary length along the gradient unit vector s_k . The new set of coordinates after step k would then be given by

$$x_{k+1} = x_k + \lambda_k s_k, \quad (10)$$

where λ_k is the *step size*. In most applications of the steepest descents algorithm in MD the step size initially has a predetermined default value. If the first iteration leads to a reduction in energy, the step size is increased by a multiplicative factor (e.g. 1.2) for the second iteration. This process is repeated so long as each iteration reduces the energy. When a step produces an increase in energy, it is assumed that the algorithm has jumped across the valley which contains the minimum and up the slope on the opposite face. The step size is then reduced by a multiplicative factor (e.g. 0.5). The step size depends upon the nature of the energy surface; for a flat surface large step sizes would be appropriate but for a narrow twisting region a much smaller step would be more suitable. The arbitrary step method may require more steps to reach the minimum but it can often require fewer function evaluations (and thus less computer time) than the more rigorous line search approach.

4.2. Conjugate Gradients

The conjugate gradients method produces a set of directions which does not show the oscillatory behavior of the steepest descents method in narrow valleys. In the steepest descents both the gradients and the direction of successive steps are orthogonal. In conjugate gradients, the gradients at each point are orthogonal but the directions are *conjugate*. A set of conjugate directions has the property that for a quadratic function of M variables, the minimum will be reached in M steps. The conjugate gradients method moves in a direction v_k from point x_k where v_k is computed from the gradient at the point and the previous direction vector v_{k-1} :

$$v_k = -g_k + \gamma_k v_{k-1}, \quad (11)$$

where γ_k is a scalar constant given by

$$\gamma_k = \frac{\mathbf{g}_k \cdot \mathbf{g}_k}{\mathbf{g}_{k-1} \cdot \mathbf{g}_{k-1}}. \quad (12)$$

In the conjugate gradients method all of the directions and gradients satisfy the following relationships:

$$\begin{aligned} \mathbf{g}_i \cdot \mathbf{g}_j &= 0 \\ \mathbf{v}_i \cdot V''_{ij} \cdot \mathbf{v}_j &= 0 \\ \mathbf{g}_i \cdot \mathbf{v}_j &= 0 \end{aligned} \quad (13)$$

where V''_{ij} is the second derivative matrix of the potential function. Because for the very first step of equation 11 we already need a direction, the first step is to use the steepest descents to obtain that first direction and then proceed with the conjugate gradients algorithm.

An example. Using our previous example function $f(x,y) = x^2 + 2y^2$, and given already the first direction along with the first minima at $(x,y) = (4,-1)$ we find the direction of the next move by first determining the negative gradient at the current point. This is the vector $(-8,4)$. This is then combined with the vector corresponding to minus the gradient at the initial point, $(-18,-36)$ multiplied by γ :

$$\mathbf{v}_k = \begin{pmatrix} -8 \\ 4 \end{pmatrix} + \frac{(-8)^2 + (4)^2}{(-18)^2 + (-36)^2} \begin{pmatrix} -18 \\ -36 \end{pmatrix} = \begin{pmatrix} -80/9 \\ +20/9 \end{pmatrix}. \quad (14)$$

To locate the second point we therefore need to perform a line search along the line with gradient $-1/4$ that passes through the point $(-4,1)$. The minimum along this line is at the origin, at the true minimum of the function. Note how the conjugate gradients method achieved the true minimum in just two moves, as illustrated in Figure III.8.

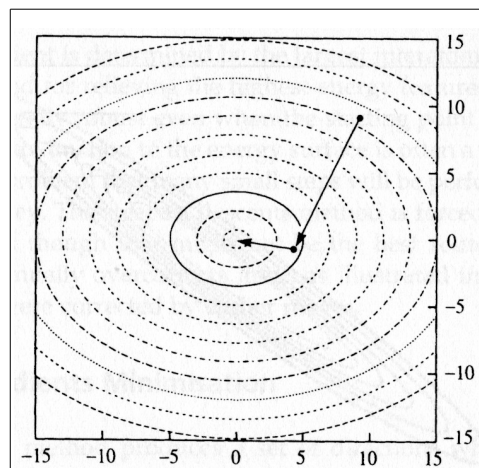


Figure III.8: Application of conjugate gradients method to the function $x^2 + 2y^2$. © Pearson, from ref. 3.

5. Equilibration

If a simulation is started from a lattice, or from a disordered configuration at a different density and temperature, it is necessary to run for a period of time so that the system can come to equilibrium at the new state point. At the end of this equilibration period, all memory of the initial configuration should have been lost.

A simple way to monitor system equilibration is to record the instantaneous values of the potential energy and pressure during this period. Examples of different thermodynamic and transport properties during equilibration are shown in Figure III.9.

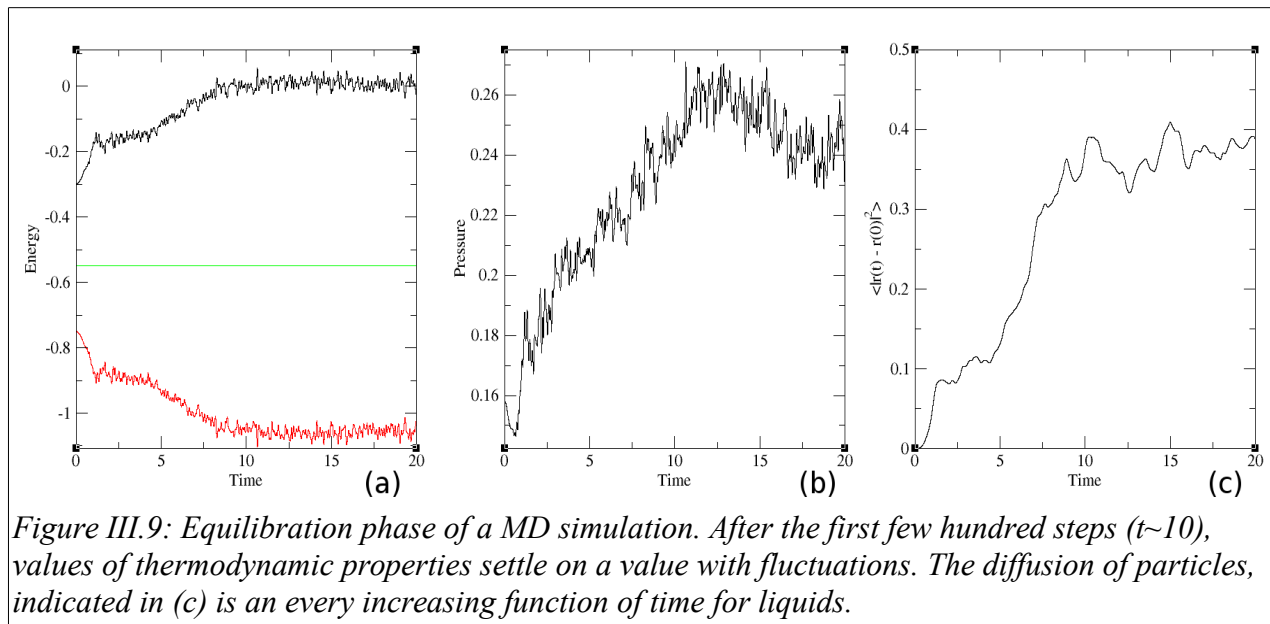


Figure III.9: Equilibration phase of a MD simulation. After the first few hundred steps ($t \sim 10$), values of thermodynamic properties settle on a value with fluctuations. The diffusion of particles, indicated in (c) is an ever increasing function of time for liquids.

The equilibration time should extend at least until the values of the thermodynamic properties cease to show a systematic drift and have started to oscillate about steady mean values and the kinetic energy is equally distributed among the three directions x , y , and z . In contrast, the mean squared displacements in Figure III.9(c) increase during the course of a liquid simulation but oscillate around a mean value in a solid. A useful rule of thumb is that when the RMS displacements per particle exceed $\sigma/2$ and is clearly increasing, then the system has “melted” and the equilibration can be declared completed.

One useful trick that may be used to decrease the time of equilibration from a lattice is to raise the kinetic temperature to a high value (e.g. $T=5$ for a LJ system) for the initial few hundred steps and then reset the temperature to the desired value during a second part of the equilibration period.