A SAMPLE MOLECULAR DYNAMICS PROGRAM

Objectives

After completing the reading of this chapter, you will be able to:
Relate experimental density to the number density in a simulation box.
Follow a simple MD simulation program step by step.
Analyze the function of each subroutine used in the given MD program.
Generate trajectories for the system.
Visualize the motion of the argon atoms in a movie or
in a snapshot series.

Keywords

Argon liquid, Simulation box, Lennard-Jones potential,
Verlet algorithm, VMD

Accompanying files with this chapter

Program file: mdnewargon.f
Input data file: ncoord.0

Introduction

We illustrate the MD simulation procedure by considering a simple example, viz., simulation of liquid argon.

Argon is an inert gas under ambient conditions. It is mostly used as an inert shielding gas in welding and other high temperature operations in various industrial processes. Gaseous argon is also used in incandescent lamps and fluorescent lighting. In high technological applications, argon makes a distinctive blue-green gas laser. Argon is produced industrially by the fractional distillation of liquid air in a cryogenic air separation unit. Argon boils at 87.3 K (to compare, boiling point of nitrogen is 77.3 K). Liquid argon is used as the target for direct dark matter searches. The interaction of a hypothetical WIMP (weakly interacting massive particle) particle with the argon nucleus produces scintillation light that is detected by photomultiplier tubes.

As mentioned in Chapter 26, a typical MD simulation involves the following steps.

1. Give initial positions, r ($t = 0$) to the atoms, choose an appropriate time step, Δt.
2. For the chosen potential function $U(r)$, calculate the force $F(r)$ for all the atoms.
3. Move the atoms using the Verlet algorithm.
4. Move time forward to ($t + \Delta t$).

5. Repeat step nos 2, 3, 4 as long as desired.

During these steps, all the properties of interest, viz., temperature, pressure, total energy etc. need to be monitored for the subsequent analysis.

The size of the simulation system is so chosen such that the number density in the simulation box corresponds to the experimental density of the system. Argon is a liquid at 85 K with density 1.40 gm/cc. Corresponding to this density at this temperature, 64 argon atoms are to be kept in a volume of 3.03×10^3 A^3.(How? See Q.1.) The simulation box is usually taken to be cubic, this makes the box size to be 14.4744 A.

In order to prepare an initial configuration for the system, one may start with a simple cube, in which the argon atoms are placed on to the lattice sites. Given these initial positions $r(t = 0)$ to the atoms, we need to choose an appropriate time step, Δt.

The potential energy function for argon is a simple Lennard-Jones 12-6 potential,

$$U(r) = 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \qquad (27.1)$$

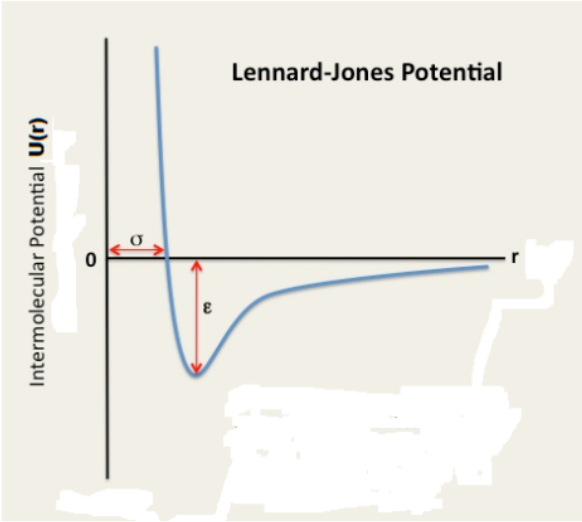


Figure 27.1: The Lennard-Jones potential defined by the two parameters, σ (particle diameter) and ε (well-depth). The deeper the well-depth ε, the stronger is the interaction between the two particles.

For the chosen potential function $U(r)$, we then calculate the force $F(r)$ for all the atoms using eq. (27.2).

$$F(\vec{r}) = -\frac{dU(r)}{dr} = 4\varepsilon\left[\frac{\sigma^{12}}{r^{14}} - \frac{\sigma^{6}}{r^{8}}\right]\vec{r} \qquad (27.2)$$

We now move the atoms using the Verlet algorithm,

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + (\Delta t)^2 a(t) \quad (27.3)$$

where, $a(t)$ is the acceleration and is conveniently replaced as, $a(t) = \frac{F(t)}{m}$. Here m is the mass of the atom.

In eq. (27.3),earlier step time is ($t - \Delta t$), current time is$t$ and the forward time is ($t + \Delta t$).

Now, open the two attached files: mdnewargon.f (program file) and 'ncoord.0 (input data file).

Description of the program: mdnewargon.f

Like any FORTRAN program, the program mdnewargon.f comprises the 'main' and the following subroutines: dynama, forcal, acomcor, atmprtr, averlet, bring, avsd, paircf, vmdwrite, gauss, savern, setrn and ranrun. The 'main' contains the input-output file specifications and the statements for the information to be generated by the program. The purpose of the accompanying subroutines has been outlined in the following table.

Subroutine	Purpose
dynama	Controls the dynamics of the system, by calling forcal, atmprtr and averlet.
forcal	Does the force and potential energy calculation for the system. Also collects the histogram data for calculating the radial distribution function.
acomcor	Checks for the centre-of-mass correction for the system.
atmprtr	Calculates kinetic energy and temperature from the velocities.
averlet	The regular Verlet algorithm for the forward step position.
bring	Brings back the coordinates of the atoms into the box, using the periodic boundary conditions.
avsd	Calculates the averages and standard deviations for the dynamic quantities (temperature, kinetic energy, potential energy, total energy and pressure).
paircf	Calculates the pair correlation function (radial distribution function).
vmdwrite	Writes the configuration for viewing in VMD (Visual Molecular Dynamics).
gauss	Prepares a Gaussian distribution of the velocities corresponding to the set temperature.
ranrn, setrn, savern	Subroutines to generate, set and save random numbers.
function rannum	Random number generator function.

In the program 'mdnewargon.f', an input file 'ncoord.0' (unit = 11) has been opened in the 'main' from which the starting parameters and positions of the atoms are read.There are several output files where different types of data are written. What follows is a short description of each of these files.

File name	Description
ncoord.0	Contains the run parameters and initial coordinates for 64 argon atoms. A "*.0" is used to denote the starting file; the subsequent files for rerun may be named as "*.1", "*.2" etc.
argoout	All the bug checks, positions, forces and velocities are written.
bindat	The earlier positions, current positions and current velocities are written in binary form. Useful for restarting the MD runs.
grdat	File with the data for $g(r)$, the radial distribution function.
positions	Positions $r(t)$ are written, to be utilized later to calculate mean squared displacements.
velocities	Velocities $v(t)$ are written, to be utilized later to calculate velocity autocorrelation functions.
arvmd.xyz	Coordinates of argon atoms in (x,y,z) format, written for 1 in 100 (or 50) MD steps. This file is to be used by the VMD (Visual Molecular Dynamics) visualization program, to show animations of the motion of the argon atoms. Atom no 15 has been given a different colour for the purpose of tracking.
testxij	A bug check file for positions.
testvij	A bug check file for velocities.

The following are the descriptions of the parts of the program in detail.

Line 14: The implicit statement makes the real variables as double precision.

Line 15 to 24: Common blocks are specified so that the variables are shared as 'common' to all the subroutines, where these set of lines appear.

Line 26 to 38: Input and output files with their unit nos and names.

Line 42 to 51: Parameters read and written.

Line 53 to 85: As explained in the comments.

Line 91 to 111: When the run starts with an input of positions and velocities of the atoms.

Line 113 to 143: When the run starts with an input of positions only (cold start). The velocities of the atoms are generated from the set temperature (85 K) as a Gaussian distribution using the subroutines 'gauss', 'savern', 'setrn' and 'ranrn'.

Line 146 to 154: The positions x01(3,ntot), x(3,ntot) and velocities v(3,ntot) are defined. Here 'ntot' is the total no of argon atoms, i.e. 64.

The array 'x01' stores the earlier positionsat time ($t - \Delta t$). Array 'x' has the present time (t) positions, array 'v' stores the present time (t) velocities. Similarly, the array 'xnew' stores the forward step positionsat time ($t + \Delta t$). In all these arrays and also in the array'f(3,ntot)', the (1,ntot) is the x-component, (2,ntot) is the y-component and (3,ntot) is the z-component of the respective variables.

Line 156 to 163: Initial temperature is calculated using the subroutine 'atmprtr', and then printed in the 'argoout' file (unit = 12).

Line 164 to 178: Initial forces are calculated using the subroutine 'forcal'and then printed in the 'argoout' file (unit = 12).

Line 180 to 183: Initialization for the histogram, needed to calculate the pair distribution function.

Line 185 to 187: Write the starting (x,y,z) coordinates of the atoms for viewing in VMD.

Line 193 to 195: Variables 'nc1' and 'nsub1' are initialized. These are needed to make cumulative sums of the quantities of interest (petot, t, ketot, etot, ptot), see in line nos 216 to 222.

Line 197 to 286: The main loop for the dynamics and collecting the properties. 'noutf' is the total no of MD steps to be run for the simulation.

Line 199 to 202: Writing the (x,y,z) coordinates of the atoms for viewing in VMD in every 50 steps.

Line 207: Subroutine 'acomcor' corrects for the movements of the centre of mass of the full system (i.e., the box containing all the 64 atoms).

Line 208: Subroutine 'dynama' handles the force calculation, Verlet algorithm and velocity rescaling, if needed.

Line 210 to 211: Bug cheks for positions and velocities of the selected atom no 1.

Line 216 to 222: Storing the quantities of interest (petot, t, ketot, etot, ptot), for calculating averages and standard deviations.

Line 224 to 226: Writing current positions, x (for calculating the mean squared displacemnets, using a separate program, not shown here) and current velocities, v (for calculating the velocity autocorrelation function, using a separate program, not shown here).

Line 228 to 236: Updating the positions with respect to time. Current position 'x(3,ntot)' becomes earlier position 'x01(3,ntot)'. Forward step position 'xnew(3,ntot)' (calculated in subroutine 'averlet') becomes current position 'x(3,ntot)'.

Line 238: Subroutine 'bring' brings back the coordinates of the atoms into the box, which might have moved out by subroutine 'averlet'.

Line 240 to 255: Writing the x01, x, v in binary form, in every 1000 steps in file 'bindat' (unit 13). This file will be utilized for restarting the MD run, later.

Line 273 to 278: Temperature is checked in every 10 steps.

Line 297 to 306: Averages and standard deviations are calculated using the subroutine 'avsd'.

Line 308 to 320: Radial distribution function g(r) is calculated using the subroutine 'paircf'. Also, the function $4\pi\rho r^2 g(r)$ (needed for calculating the running coordination number) is calculated and printed in the 'grdat' file (unit = 14).

Line 322 to 365: All the final variables and quantities (temperature, forces, positions, velocities) are printed.

Line 370 to 372: Writing the x01, x, v in binary form in file 'bindat' (unit 13). This file will be utilized for restarting the MD run, later.

Line 387 to 498: Subroutine 'forcal'.

Line 391 to 401: Double precision declaration and common block of shared variables.

Line 405 to 406: Potential energy 'petot' is initialized.

Line 408 to 413: Force array f(3,ntot) is initialized.

Line 415 to 477: Potential energy and force calculation. Loop 160 runs for n = 1 to (ntot - 1) and loop 150 runs for m = (n + 1) to ntot. This ensures that the atom pairs are not repeated. The array xxx(3) stores the distance vectors [$(r_m - r_n)$; $r_m \equiv$ (x(1,m), x(2,m), x(3,m))$\equiv (x_m, y_m, z_m)$].

Line 425: Minimum images of the distance vectors are calculated using the built-in 'anint' function.

Line 426: Variable 'rsqnm' is $|(r_n - r_m)|^2$.

Line 433 to 473: Calculation of LJ potential energy and LJ force using a spherical cut-off distance of rss.lt.rcut (where, rss is square root of rsqnm, rcut is xhl, xhl is half of the box length). In case the atom-atom distance gets close to zero, both potential energy and force become very large. In order to prevent this type of numerical blow-up, a safe distance of 0.9*sigma has been chosen.

Line 446 to 454: Calculation of LJ potential energy according to eq. (27.1).

Line 458 to 472: Calculation of LJ force according to eq. (27.2).

Line 478 to 494: Calculation of virial pressure (pvir), ideal pressure (pid) and total pressure (ptot) according to eq. (27.4).

$$P_{\text{vir}} = \frac{1}{3V}\sum_{i=1}^{N}(F_i \,.\, r_i) \quad (27.4a)$$

$$P_{\text{id}} = \frac{Nk_{\text{B}}T}{V} \quad (27.4b)$$

$$P_{\text{tot}} = P_{\text{id}} + P_{\text{vir}} \quad (27.4c)$$

Line 500 to 528: Subroutine 'dynama'. It calls subroutine 'forcal' to calculate force and potential energy. Next it calls 'atmprtr' to calculate kinetic energy and temperature. Subroutine 'averlet' is called next to move the atom into the forward step and also the calculate the velocities. It then calls 'atmprtr' again in order to recalculate temperature (in case of a velocity rescaling).

Line 531 to 560: Subroutine 'avsd'. This routine calculates the averages and standard deviations.

Line 551: sav(i) is the average of the i-th quantity (petot/t/ketot/etot/ptot).

Line 559: ssdv(i) is the standard deviation of the i-th quantity (petot/t/ketot/etot/ptot).

Line 566 to 592: Subroutine 'paircf'. This routine calculates the pair distribution function (also called the radial distribution function) from the histogram data (collected in subroutine 'forcal'). The implementation formula for the pair distribution function is explained below.

Let us suppose, we ran the program for τ_{run} steps during which, in a bin b, we collected the no of atoms data within the interval (r, $r+\delta r$) as n_{his} (b) pairs. Then the average number of atoms whose distance from a given atom in the fluid lies in theinterval (r, $r + \delta r$) is

$$n(b) = \frac{n_{his}(b)}{N\tau_{run}} \quad (27.5a)$$

where, N is the total no of atoms in the system. The average number of atoms in the same interval in an ideal gas at the same density ρ is

$$n^{id}(b) = \frac{4\pi\rho}{3}[(r+\delta r)^3 - r^3] \quad (27.5b)$$

where, ρ is defined as $\rho = \frac{N}{V}$.

Therefore, the pair distribution function is

$$g\left(r + \frac{1}{2}\delta r\right) = \frac{n(b)}{n^{id}(b)} \quad (27.5c)$$

Line 585: rlower is r.

Line 586: rupper is $(r + \delta r)$.

Line 587: fideal is $(r + \delta r)^3 - (r)^3$.

Line 588: hist(nbin) is n_{his} (b), real(nc1) is τ_{run} steps, ntot is N, const is $\left(\frac{4\pi\rho}{3}\right)$.

Line 595 to 684: Subroutines for generating Gaussian velocities (when only positions are given as input data)

Line 687 to722: Subroutine 'bring'. This is the periodic boundary condition to bring back the atoms which might have departed from the box following the forward step movement in 'averlet'. The function 'anint' (in Line 706) takes care of the minimum image convention and periodic boundary correction. Examples of anint(x) use are: anint(7.3) returns as 7.0, anint(1.49) returns as 1.0, anint(1.5) returns as 2.0, anint(-2.5) returns as -3.0.

Line 725 to 763: Subroutine 'vmdwrite'. This routine writes the (x,y,z) coordinates for visualization of the atoms using the VMD (Visual Molecular Dynamics) software. Note that atom no 15 is given a separate identification, so that it will be easier to follow its motion on the VMD screen.

Line 766 to 812: Subroutine 'acomcor'. This routine makes the correction for the centre of mass motion.

Line 794 to 798: Calculation of the momentum in each (x-,y-,z-) direction.

Line 800 to 802: Correction factors facx, facy, facz in each (x-,y-,z-) direction, respectively

Line 805 to 809: Velocities are corrected.

Line 815 to 866: Subroutine 'averlet', the regular Verlet algorithm.

Line 840 to 847: For step no 1, only positions are known x(3,ntot) and the Gaussian velocities v(3,ntot) have been generated by the subroutines (gauss, savern, stern, ranrn). Forcesf(3,ntot) are calculated using subroutine 'forcal'. Thus,

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}\frac{F(t)}{m}(\Delta t)^2 \quad (27.6)$$

Line 849 to 853: When step no is more than 1, the three time ($t - \delta t$, t, $t + \delta t$) formula applies

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \frac{F(t)}{m}(\Delta t)^2 \quad (27.7)$$

Line 858 to 863: The velocities v(3,ntot) are now calculated according to,

$$v(t) = \left(\frac{1}{2\Delta t}\right)[r(t+\Delta t) - r(t-\Delta t)] \qquad (27.8)$$

Line 869 to 928: Subroutine 'atmprtr'. This subroutine calculates kinetic energy and temperature from the velocities.

Line 886 to 891: Kinetic energy is calculated according to,

$$KE = \frac{1}{2}\sum_{i=1}^{N} m_i v_i^2 \qquad (27.9)$$

where, v_i is the velocity and mi is the mass of the i-th atom.

Line 897 to 898: Dynamic temperature is calculated according to,

$$T = \frac{2\,KE}{3Nk_B} \qquad (27.10)$$

where, k_B is the Boltzmann constant and N is the total no of atoms.

Line 901 to 905: The velocity rescaling factor tcor is defined as,

$$tcor = \sqrt{\frac{T_0}{T_{calc}}} \qquad (27.11)$$

where, T_0 is the set temperature and T_{calc} is the temperature calculated from the velocities, using eq. (27.10).

Line 908 to 914: Velocities are rescaled, only when T_{calc} deviates too much from T_0.

Line 916 to 923: Kinetic energy and temperature are recalculated from the rescaled velocities.

Results

The program mdnewargon.f has been run for 5000 MD steps using $\Delta t = 0.05$. The important output files are 'grdat' (containing the radial distribution function data) and 'arvmd.xyz' (containing the x,y,z data for the atoms saved in every 50 steps, for visualization using the VMD).

The radial distribution function

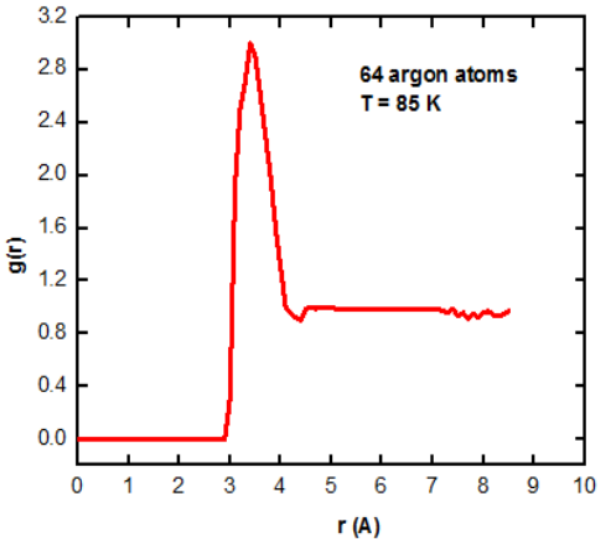
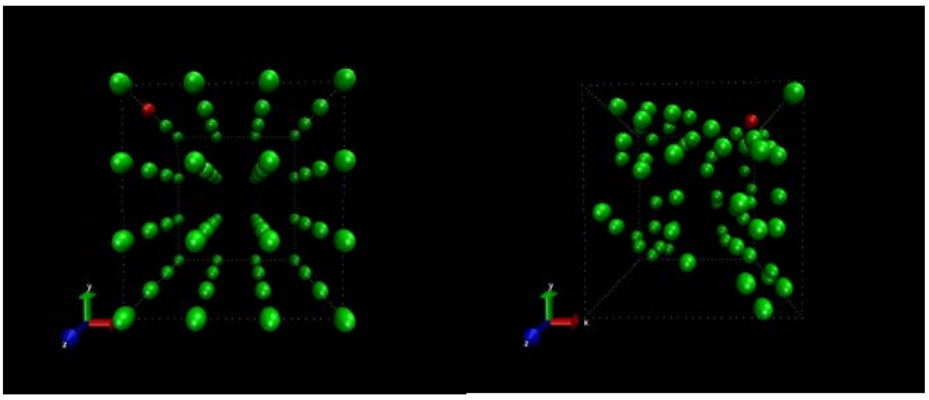


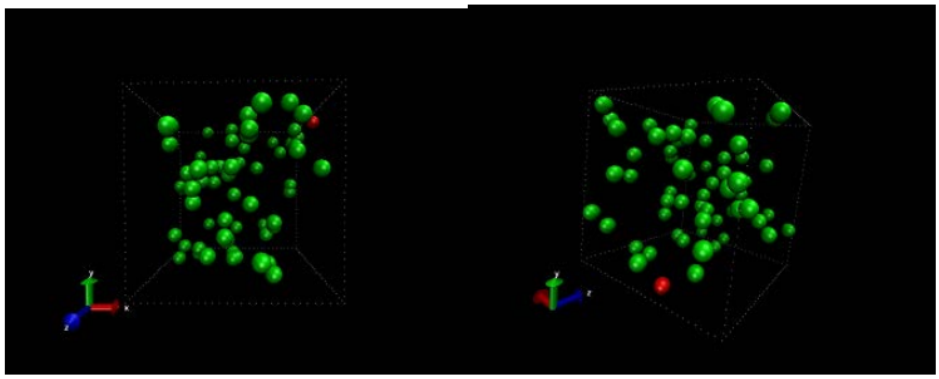
Figure 27.2: The radial distribution function for liquid argon at 85 K (using the results from the mdnewargon.f program, for 5000 MD steps, using $\Delta t = 0.05$). The first peak position at ~ 3.4 A corresponds to the first coordination shell of the atoms. The subsequent coordination shells are expected to develop at longer radial distances. This requires that the MD run is continued for several million steps and statistical averaging from multiple runs (using different starting configurations) is performed.

Snapshots of the system at different times

Figure 27.3 presents a series of snapshots generated by the VMD (Visual Molecular Dynamics, an open source software; see in Chapter #, for details) using the output file 'arvmd.xyz' as the input. Note that atom No 15 has been given a different colour for the purpose of illustration.



Starting configuration (step no 0)Configuration at step no 1000



Configuration at step no 2000 Configuration at step no 5000

Figure 27.3: Snapshots of the atoms in liquid argon at 85 K (using the results from the mdnewargon.f program, for 5000 MD steps, using $\Delta t = 0.05$). Note the position of atom no 15 (marked red) at different time.

Questions

1. Prove that for 64 argon atomsthe edge length of the cubic simulation box is 14.4744 A. Given,density of argon 1.40 gm/cc at 85 K, atomic weight of argon 39.948.
2. Why the temperature of the system needs to be checked in every 10 steps?
3. What is the origin of p_{vir}?
4. Compare the figures 26.2 and 27.2 for the RDF of argon. In which respect these are different and why?
5. From the snapshots shown in the figure 27.3, can you formulate a scheme to calculate the self-diffusion coefficient of argon?