Understanding Molecular Simulation 2010

Srikanth Sastry

Jawaharlal Nehru Centre for
Advanced Scientific Research,
Bangalore

http://www.jncasr.ac.in/sastry

IIT Kanpur
November 5, 2010

Outline

- Statistical Mechanics I: Entropy, Boltzmann weight, partition function, canonical ensemble [Frenkel Lecture 1, Das Lecture 1]
- Statistical Mechanics II: Different ensembles, Dynamics [Frenkel Lecture 2, Sastry Tutorial Lecture 1]
- Molecular dynamics simulations - basic principles, boundary conditions, neighbor lists, truncation and tail corrections [Frenkel Lecture 3]
- This Tutorial Lecture:

➢ Review basics of molecular dynamics

➢ Molecular dynamics in different ensembles

➢ Event driven MD

➢ Molecular systems

➢ Long range interactions

Molecular Dynamics – Basic concepts

- Integrate the equations of motion to generate a trajectory.
- Average over the trajectory is the average that we seek for properties of the system.
- Statistical mechanics provides formulation to extract useful quantities.
- Both equilibrium thermodynamic and dynamical quantities are obtained.
- Structure of MD simulations: Initialization, force calculation, integration of equations of motion, sampling to calculate desired averages.

Algorithm 3 (A Simple Molecular Dynamics Program)

```
program md                      simple MD program

call init                       initialization
t=0
do while (t.lt.tmax)            MD loop
   call force(f,en)             determine the forces
   call integrate(f,en)         integrate equations of motion
   t=t+delt
   call sample                  sample averages
enddo
stop
end
```

Comment to this algorithm:

1. *Subroutines* `init`, `force`, `integrate`, *and* `sample` *will be described in Algorithms 4, 5, and 6, respectively. Subroutine* `sample` *is used to calculate averages like pressure or temperature.*

Algorithm 4 (Initialization of a Molecular Dynamics Program)

```
subroutine init                initialization of MD program
sumv=0
sumv2=0
do i=1,npart
  x(i)=lattice_pos(i)          place the particles on a lattice
  v(i)=(ranf()-0.5)            give random velocities
  sumv=sumv+v(i)               velocity center of mass
  sumv2=sumv2+v(i)**2          kinetic energy
enddo
sumv=sumv/npart                velocity center of mass
sumv2=sumv2/npart              mean-squared velocity
fs=sqrt(3*temp/sumv2)          scale factor of the velocities
do i=1,npart                   set desired kinetic energy and set
  v(i)=(v(i)-sumv)*fs          velocity center of mass to zero
  xm(i)=x(i)-v(i)*dt           position previous time step
enddo
return
end
```

Algorithm 5 (Calculation of the Forces)

```
subroutine force(f,en)                 determine the force
en=0                                   and energy
do i=1,npart
   f(i)=0                              set forces to zero
enddo
do i=1,npart-1
  do j=i+1,npart                       loop over all pairs
     xr=x(i)-x(j)
     xr=xr-box*nint(xr/box)            periodic boundary conditions
     r2=xr**2
     if (r2.lt.rc2) then               test cutoff
       r2i=1/r2
       r6i=r2i**3
       ff=48*r2i*r6i*(r6i-0.5)         Lennard-Jones potential
       f(i)=f(i)+ff*xr                 update force
       f(j)=f(j)-ff*xr
       en=en+4*r6i*(r6i-1)-ecut        update energy
     endif
  enddo
enddo
return
end
```

Equations of Motion

Discretize equations of motion in order to integrate.
Desired properties: Preserve symmetries of the original equations (time reversal..), conservation of energy, permit large time steps of integration…

Verlet Algorithm: Sum Taylor series for positive and negative time
Steps:

$$r(t+\Delta t) = r(t) + v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 + \frac{\Delta t^3}{3!}\dddot{r} + \mathcal{O}(\Delta t^4)$$

$$r(t-\Delta t) = r(t) - v(t)\Delta t + \frac{f(t)}{2m}\Delta t^2 - \frac{\Delta t^3}{3!}\dddot{r} + \mathcal{O}(\Delta t^4)$$

$$r(t+\Delta t) + r(t-\Delta t) = 2r(t) + \frac{f(t)}{m}\Delta t^2 + \mathcal{O}(\Delta t^4)$$

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

Update with positions at two times. No velocity. Good accuracy.

But can define velocity in terms of adjacent time positions:

$$v(t) = \frac{r(t+\Delta t) - r(t-\Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^2)$$

Verlet algorithm equivalent to Leap-Frog and Velocity Verlet:

Leap Frog

$$r(t+\Delta t) = r(t) + \Delta t \; v(t+\Delta t/2)$$

$$v(t+\Delta t/2) = v(t-\Delta t/2) + \Delta t \; \frac{f(t)}{m}$$

Velocity Verlet

$$r(t+\Delta t) = r(t) + \Delta t \; v(t) + \Delta t^2 \; \frac{f(t)}{2m}.$$

$$v(t+\Delta t) = v(t) + \Delta t \; \frac{f(t) + f(t+\Delta t)}{2m}.$$

How do we compare different integrators?

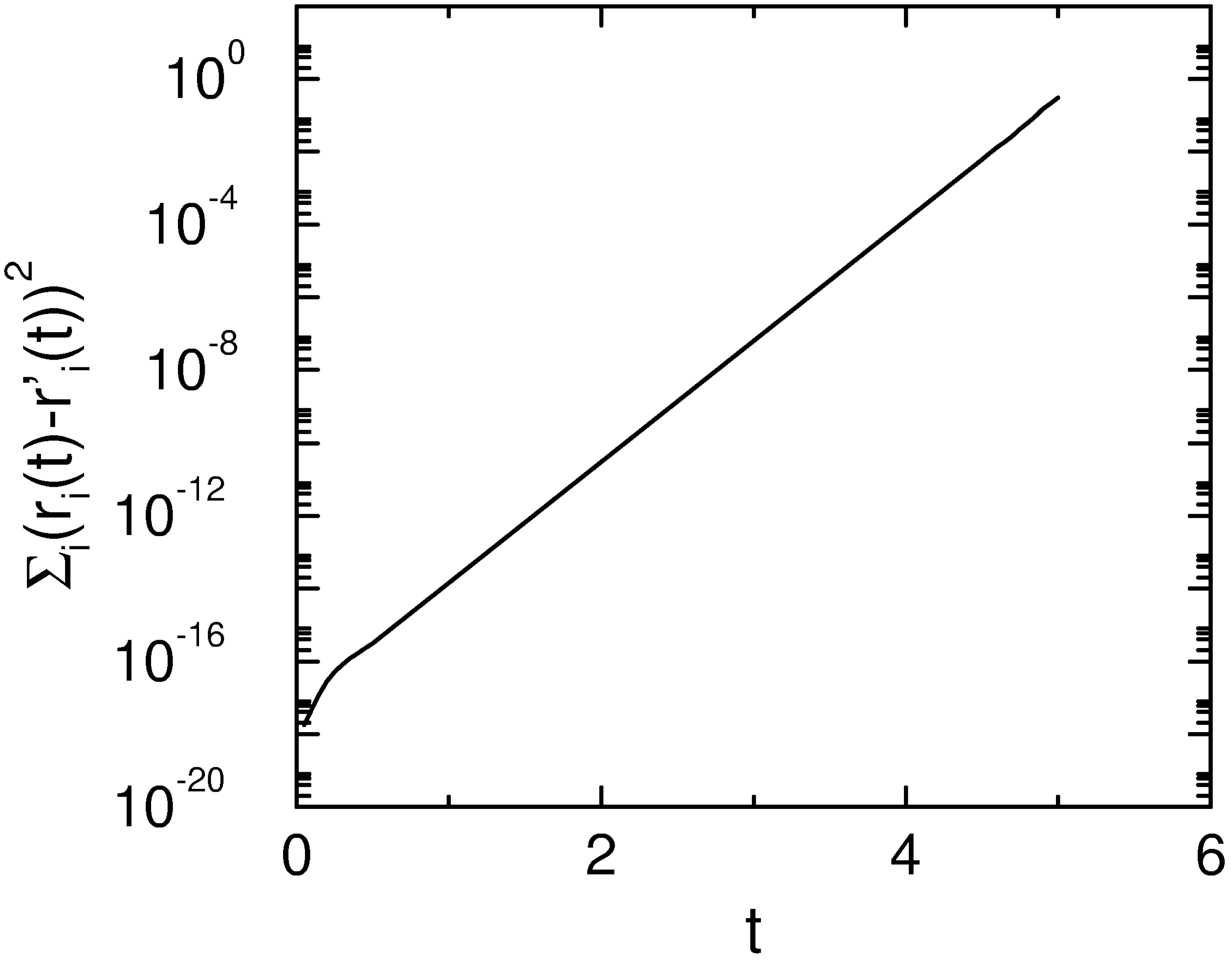
Long time accuracy is not a possible criterion. Lyapunov instability.

Symmetries, preservation of phase space volume, conservation of energy… efficient sampling of correct phase space.

Algorithm 6 (Integrating the Equations of Motion)

```
subroutine integrate(f,en)          integrate equations of motion
sumv=0
sumv2=0
do i=1,npart                        MD loop
   xx=2*x(i)-xm(i)+delt**2*f(i)     Verlet algorithm (4.2.3)
   vi=(xx-xm(i))/(2*delt)           velocity (4.2.4)
   sumv=sumv+vi                     velocity center of mass
   sumv2=sumv2+vi**2                total kinetic energy
   xm(i)=x(i)                       update positions previous time
   x(i)=xx                          update positions current time
enddo
temp=sumv2/(3*npart)                instantaneous temperature
etot=(en+0.5*sumv2)/npart           total energy per particle
return
end
```

Lyapunov Instability: Exponential divergence of nearby trajectories. True for typical many body systems.



Time Reversible Algorithms: Liouville Formulation

Time evolution of functions of coordinates and momenta:

$$\dot{f}(\mathbf{r},\mathbf{p}) = \dot{\mathbf{r}}\frac{\partial f}{\partial \mathbf{r}} + \dot{\mathbf{p}}\frac{\partial f}{\partial \mathbf{p}} \equiv iLf$$

Liouville Operator - $iL = \dot{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}} + \dot{\mathbf{p}}\frac{\partial}{\partial \mathbf{p}}$

Formal solution: $f(\mathbf{r}(t),\mathbf{p}(t)) = \exp(iLt) f(\mathbf{r},\mathbf{p})$

How to develop useful integration schemes?

Consider position and momentum parts of L separately:

$$iL_r = \dot{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}; \qquad iL_p \dot{\mathbf{p}}\frac{\partial}{\partial \mathbf{p}}$$

Operation by *iL*$_r$ leads to shift of coordinates: Similarly *iL*$_p$ leads to shift in momenta.

$$\begin{aligned} f(t) &= f(0) + iL_r t f(0) + \frac{(iL_r t)^2}{2} f(0) + \dots \\ &= \sum_{n=0}^{\infty} \frac{(\dot{\mathbf{r}}(0)t)^n}{n!} \frac{\partial^n}{\partial \mathbf{r}^n} f(0) \\ &= f(\mathbf{p}(0), (\mathbf{r}(0) + \dot{\mathbf{r}}(0)t). \end{aligned}$$

But since operators don’t commute, we cannot write

$$\exp(iLt) = \exp(iL_r t) \times \exp(iL_p t)$$

Instead, the Trotter identity is used to write approximate expressions that are time reversible:

$$\exp(A+B) = lim_{p\to\infty}\left[\exp(\frac{A}{2p})\exp(\frac{B}{p})\exp(\frac{A}{2p})\right]^p$$

$$\exp(A+B) = \left[\exp(\frac{A}{2p})\exp(\frac{B}{p})\exp(\frac{A}{2p})\right]^p \exp(\mathcal{O}(1/p^2))$$

We consider the case $A/p = iL_p t/p$ and $B/p = iL_r t/p$ and $\Delta t = t/p$.

One step of integration, with p = 2, then corresponds to:

$$\exp(iL_p\Delta t/2)\exp(iL_r\Delta t)\exp(iL_p\Delta t/2)$$

Applying step by step:

$$\exp(iL_p\Delta t/2)f(\mathbf{p},\mathbf{r}) = f(\mathbf{p}+\frac{\Delta t}{2}\dot{\mathbf{p}},\mathbf{r})$$

$$\exp(iL_r\Delta t)f(\mathbf{p}+\frac{\boldsymbol{\Delta}\mathbf{t}}{\mathbf{2}}\dot{\mathbf{p}},\mathbf{r}) = f(\mathbf{p}+\frac{\Delta t}{2}\dot{\mathbf{p}},\mathbf{r}+\Delta t\dot{\mathbf{r}}(\Delta t/2))$$

$$\exp(iL_p\Delta t/2)f(\mathbf{p}+\frac{\Delta t}{2}\dot{\mathbf{p}},\mathbf{r}+\Delta t\dot{\mathbf{r}}(\Delta t/2)) = f(\mathbf{p}+\frac{\Delta t}{2}\dot{\mathbf{p}}+\frac{\Delta t}{2}\dot{\mathbf{p}}(\Delta t),\mathbf{r}+\Delta t\dot{\mathbf{r}}(\Delta t/2))$$

Writing the next change in positions and momenta:

$$\begin{array}{rcl}
\mathbf{p}(0) & \rightarrow & \mathbf{p}+\frac{\Delta t}{2}\mathbf{F}(0)+\frac{\Delta t}{2}\mathbf{F}(\Delta t) \\
\mathbf{r}(0) & \rightarrow & \mathbf{r}+\Delta t\dot{\mathbf{r}}(\frac{\Delta t}{2}) \\
 & = & \mathbf{r}+\Delta t\dot{\mathbf{r}}(0)+\frac{\Delta t^2}{2m}\mathbf{F}(0).
\end{array}$$

This is the Velocity Verlet update!

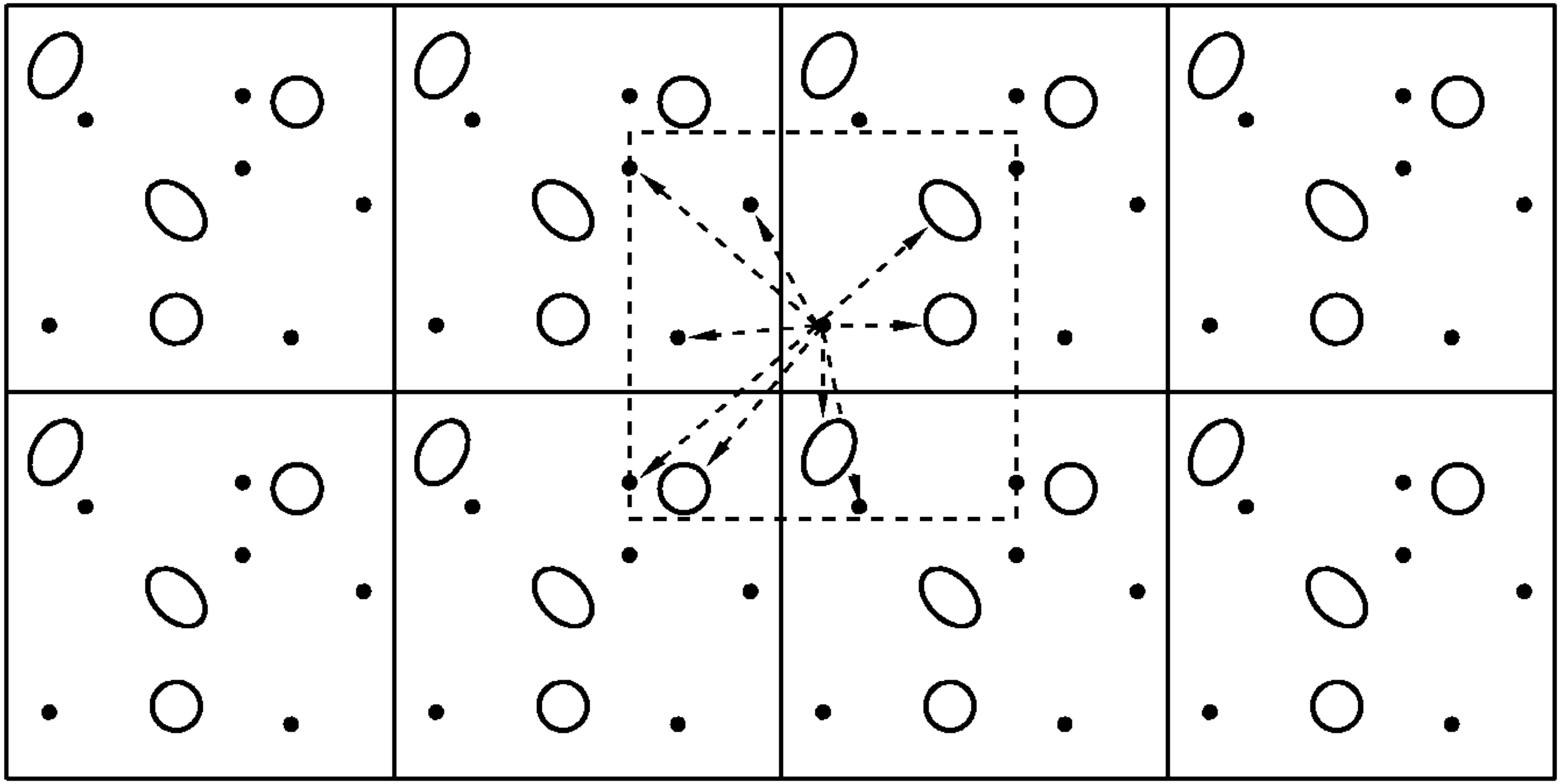
Verlet algorithm is the first member of integrators that Can be developed from this procedure.

Periodic Boundary Conditions

Number of particles in simulations 100 – 10^6 but need to simulate properties of bulk systems.

Avoid free boundaries, by using periodic boundary conditions.

Calculate interactions with periodic images of particles as well.



Truncation of the potential

Use of periodic boundary conditions not efficient if interactions with all image particles have to be calculated.

But for short ranged potentials such as the LJ potential, one truncates the potential beyond a cutoff r_c.

Thus, we use

$$\begin{aligned} U(r) = & \quad 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] & if\ r \leq r_c \\ = & \quad 0 & if\ r \geq r_c \end{aligned}$$

But long range correction needs to be calculated, or potential redefined. If we assume density of neighbors to be constant beyond cutoff:

$$U^{Corr} = \frac{N\rho}{2}\int_{r_c}^{\infty} drU(r)4\pi r^2 dr \qquad U^{Corr} = \frac{N8\pi\rho\epsilon\sigma^3}{3}\left[\frac{1}{3}\left(\frac{\sigma}{r_c}\right)^9 - \left(\frac{\sigma}{r_c}\right)^3\right]$$

For the Lennard-Jones potential

Neighbor lists

The calculation of the energy scales as N^2

But for short range interactions, finite number of neighbors for each particle, so O(N) computation should be possible.

This can be done if we know the neighbors of each particle but this keeps changing.

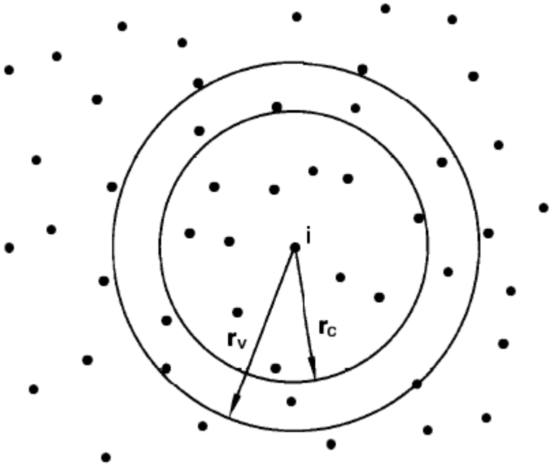
Solution: Calculate neighbors for radius $r_v > r_c$

Monitor how far particles move.

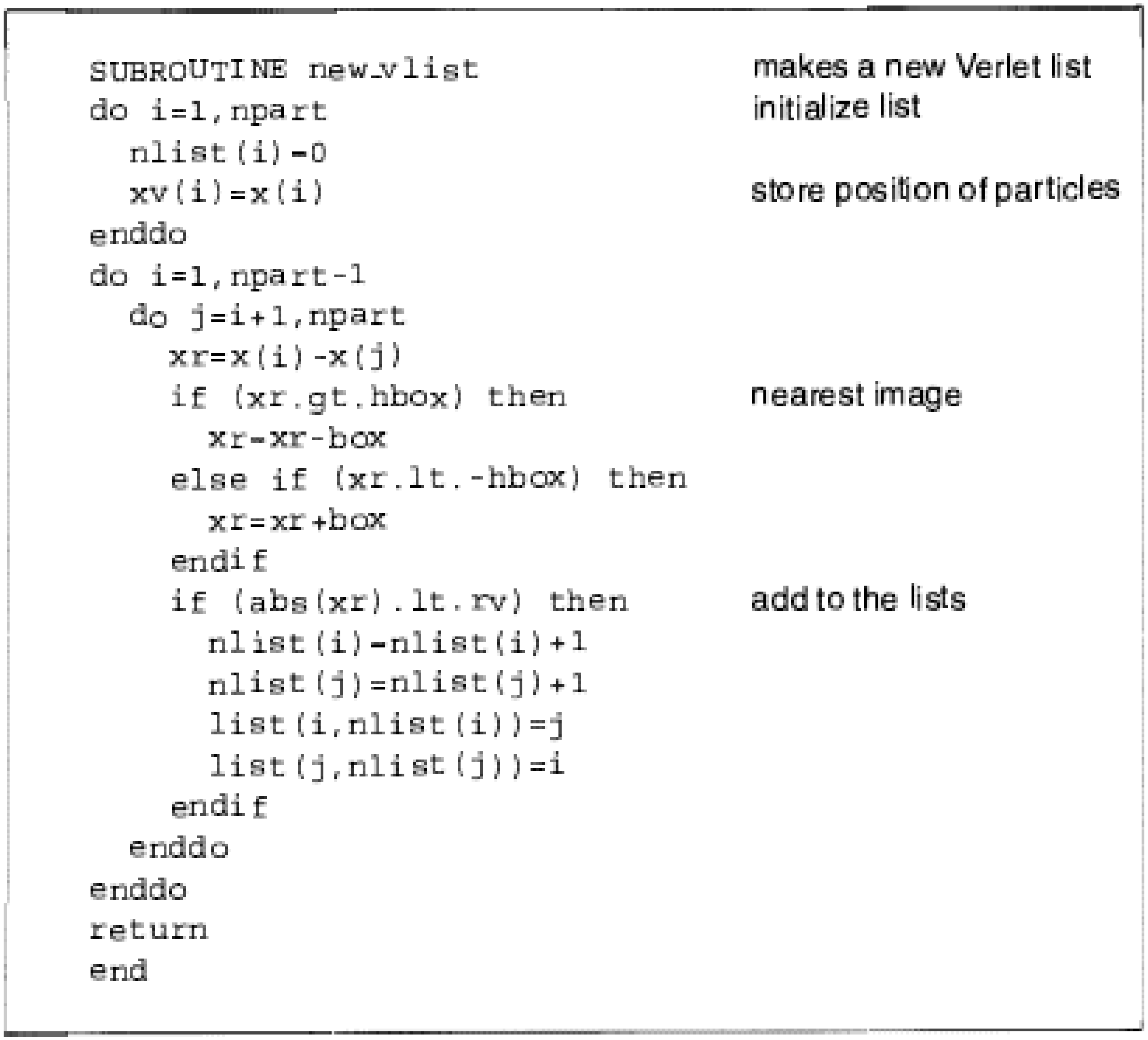
When maximum displacement exceed $(r_v - r_c)/2$, update

neighbor list.

Efficient scheme – Verlet Neighbor Lists.



Algorithm 34 (Making a Verlet List)



```
SUBROUTINE new_vlist                                 makes a new Verlet list
do i=1,npart                                         initialize list
  nlist(i)-0
  xv(i)=x(i)                                         store position of particles
enddo
do i=1,npart-1
  do j=i+1,npart
    xr=x(i)-x(j)
    if (xr.gt.hbox) then                             nearest image
      xr-xr-box
    else if (xr.lt.-hbox) then
      xr=xr+box
    endif
    if (abs(xr).lt.rv) then                          add to the lists
      nlist(i)-nlist(i)+1
      nlist(j)=nlist(j)+1
      list(i,nlist(i))=j
      list(j,nlist(j))=i
    endif
  enddo
enddo
return
end
```

Molecular Dynamics simulations in different ensembles

So far: molecular dynamics in the microcanonical ensemble (constant N, V, E) have been discussed.

Why do we need to consider other ensembles?

➢To do single phase simulations in the presence of a first order transition – e. g., liquid - gas phase transition -- (N, P, T) ensemble preferable to (N, V, T)

➢To simulate efficiently in conditions close to experiment – e. g., gas adsorption in porous adsorbent – (μ, V, T) preferable to (N, V, T)

➢Determination of fluctuations – e. g. isothermal compressibilit from volume fluctuations -- (N, P, T) ensemble preferable to (N, V, T).

➢“Unphysical ensembles” for special purposes – e. g., phase equilibria.

Molecular dynamics in the canonical (N, V, T) ensemble

In the canonical ensemble, the system is characterized by a fixed temperature, which is usually measured in MD simulations by the average kinetic energy.

One possible way of then achieving constant temperature is to constrain the kinetic energy by modifying the equations of motion:

$$\dot{r} = \frac{p}{m}; \qquad \dot{p} = f - \xi(p)\ p$$

The constraint parameter ξ is chosen such that:

$$\frac{1}{2m}\sum_i p_i^2 = constant \longrightarrow \xi = \frac{\sum_i p_i . f_{\ i}}{\sum_i p_i^2}$$

This “friction” is calculated at each integration time step.

Such modified dynamics does not generate the true canonical ensemble.

Andersen thermostat

Another possible simple thermostat is the “Andersen thermostat” wherein the coupling with the heat bath is represented by stochastic forces that act on randomly selected particles, such that the particle acquires a new velocity drawn from the Maxwell distribution.

One must specify a “collision frequency” that determines how often such a velocity change is made.

A simple method which generates the canonical ensemble, with static properties that do not depend on collision frequency.

However, not so good for dynamics, which depend on the arbitrary collision frequency that is specified.

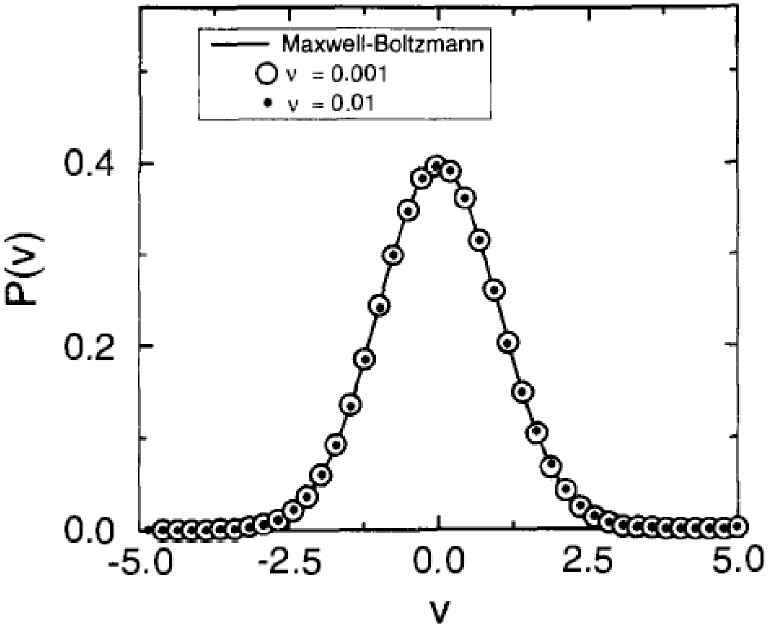
Algorithm 14 (Molecular Dynamics: Andersen Thermostat)

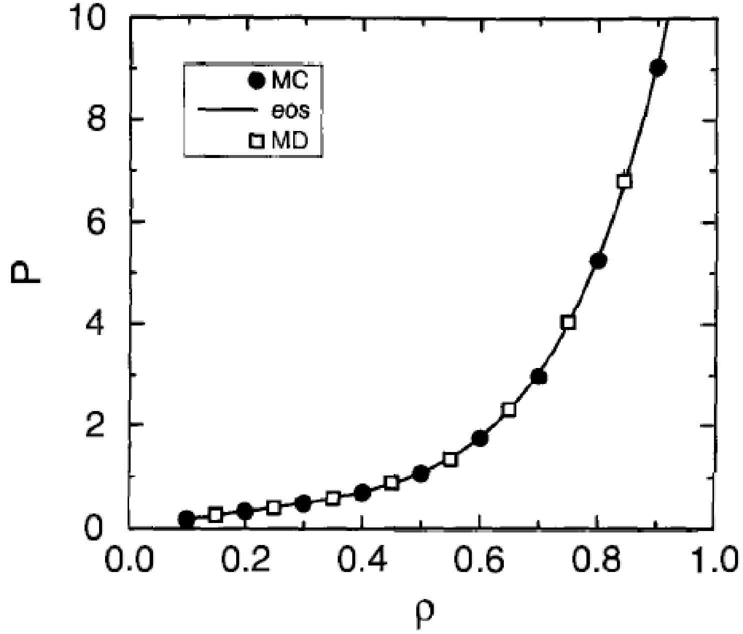
```
program md_Andersen                       MD at constant temperature
call init(temp)                           initialization
call force(f,en)                          determine the forces
t=0
do while (t.lt.tmax)                      MD loop
   call integrate(1,f,en,temp)            first part of the eqs. of motion
   call force(f,en)                       determine the forces
   call integrate(2,f,en,temp)            second part of eqs. of motion
   t=t+dt
   call sample                            sample averages
enddo
stop
end
```

Algorithm 15 (Equations of Motion: Andersen Thermostat)

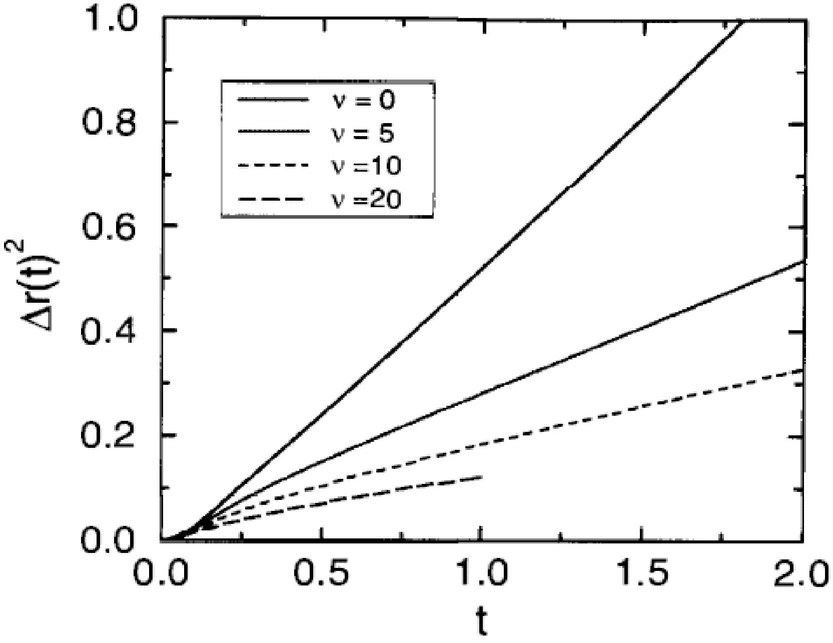
```
   subroutine integrate(switch,f          integrate equations of motion:
                    ,en,temp)             with Andersen thermostat
   if (switch.eq.1) then                  first step velocity Verlet
      do i=1,npart
         x(i)=x(i)+dt*v(i)+               update positions current time
+          dt*dt*f(i)/2
         v(i)=v(i)+dt*f(i)/2              first update velocity
      enddo
   else if (switch.eq.2) then             second step velocity Verlet
      tempa=0
      do i=1,npart
         v(i)=v(i)+dt*f(i)/2              second update velocity
         tempa=tempa+v(i)**2
      enddo
      tempa=tempa/(s*npart)               instantaneous temperature
      sigma=sqrt(temp)                    Andersen heat bath
      do i=1,npart
        if (ranf().lt.nu*dt) then         test for collision with bath
           v(i)=gauss(sigma)              give particle Gaussian velocity
        endif
      enddo
   endif
   return
   end
```

Static properties





Dynamic Properties

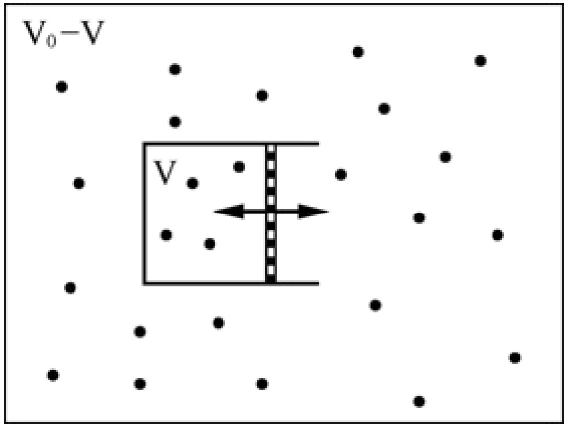


Extended Lagrangian Methods

The general idea (we will also discuss this for the constant pressure MD) is to treat an “extended” system, which includes additional degrees of freedom.

In the case that one imposes a constant pressure in this way, one may visualize the additional degree of freedom as the position of the “piston” in the set up we considered Earlier.

An extended Lagrangian is written for the full system.



Lagrangian dynamics:

$$\mathcal{L} = T - V \qquad 0 = \frac{d}{dt}\left(\frac{\partial \mathcal{L}}{\partial \dot{q}_j}\right) - \frac{\partial \mathcal{L}}{\partial q_j}$$

The Nose-Hoover thermostat

The extended Lagrangian, including an additional degree of freedom s used to obtain the canonical ensemble is:

$$\mathcal{L}_{Nose} = \sum_{i=1}^{N} \frac{m_i}{2} s^2 \dot{\mathbf{r}}_i^2 - U(\mathbf{r}^N) + \frac{Q}{2}\dot{s}^2 - \frac{L}{\beta} ln\ s$$

The conjugate momenta are

$$\mathrm{p}_i = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{r}}} = m_i s^2 \dot{\mathbf{r}}_i \qquad p_s = \frac{\partial \mathcal{L}}{\partial \dot{s}} = Q\dot{s}.$$

And the Hamiltonian is $\mathcal{H}_{Nose} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i s^2} + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + L\frac{ln\ s}{\beta}$

We write the (“microcanonical”) partition function for this system as

$$Q_{Nose} = \frac{1}{N!} \int dp_s \; ds \; d\mathbf{p}^N \; d\mathbf{r}^N \delta(E - \mathcal{H}_{Nose})$$

Using scaled variables $\mathbf{p}' = \mathbf{p}/s$

And defining $\mathcal{H} = \sum_{i=1}^{N} \frac{\mathbf{p}_i'^{\,2}}{2m_i} + U(\mathbf{r}^N)$

And using $\delta(h(s)) = \delta(s - s_0)/h'(s_0)$

For a function h(s) with a root at s_0

$$Q_{Nose} = \frac{1}{N!}\int dp_s\; ds\; d\mathbf{p}'^{N}\; d\mathbf{r}^{N}\frac{\beta\; s^{3N+1}}{L}\delta\left\{s - exp\left[-\beta\frac{\mathcal{H}(\mathbf{p}',\mathbf{r}) + \frac{p_s^2}{2Q} - E}{L}\right]\right\}$$

With the choice: $L = 3N + 1$

$$Q_{Nose} = \frac{C}{N!}\int\; d\mathbf{p}'^{N}\; d\mathbf{r}^{N}\exp\left[-\beta\mathcal{H}(\mathbf{p}',\mathbf{r})\right]$$

With $r' = r$, $p' = p/s$, $s' = s$, $p_s' = p_s/s$, and $t' = t/s$

The equations of motion for the primed (real) variables are:

$$\begin{aligned}
\frac{d\mathbf{r}_i'}{dt'} &= \mathbf{p}_i'/m_i \\
\frac{d\mathbf{p}_i'}{dt'} &= -\frac{\partial U}{\partial \mathbf{r}_i'} - (s'p_s'/Q)\mathbf{p}_i' \\
\frac{1}{s}\frac{ds'}{dt'} &= s'p_s'/Q \\
\frac{d(s'p_s'/Q)}{dt'} &= \left(\sum_i p_i'^2/m_i - \frac{3N+1}{\beta}\right)/Q
\end{aligned}$$

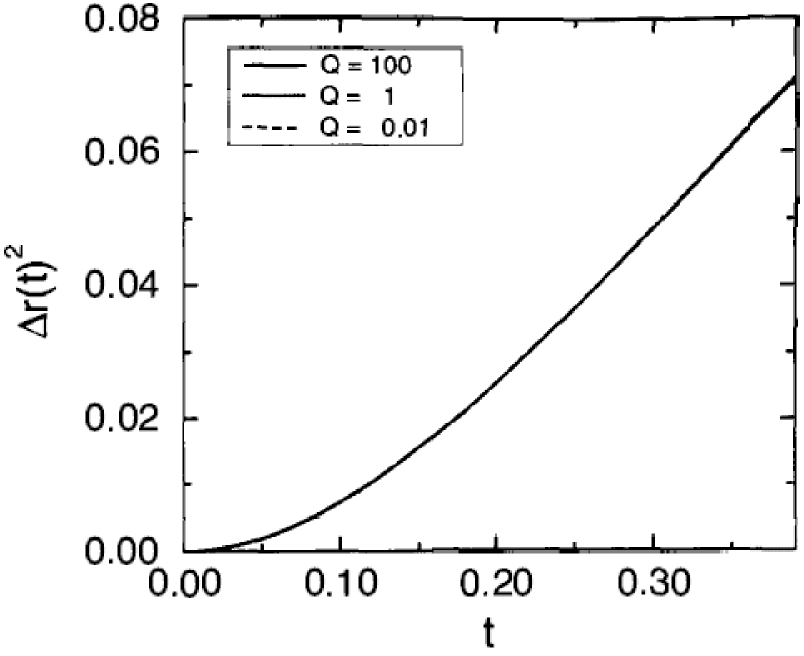
These equations conserve the Hamiltonian

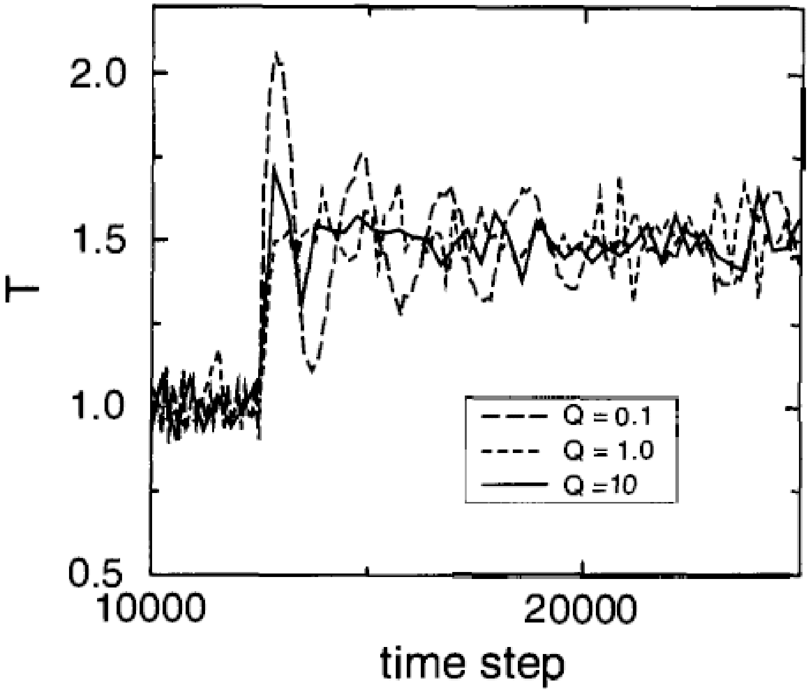
$$\mathcal{H}_{Nose} = \sum_{i=1}^{N} \frac{\mathbf{p}_i'^2}{2m_i} + U(\mathbf{r}'^N) + \frac{s'^2 p_s'^2}{2Q} + L\frac{lns'}{\beta}$$

From the relation between the scaled and unscaled variables it is apparent that the variable s scales the time (in a time dependent way).

$\xi = \dot{s}/s$ is like the friction term in the constraint method, that plays the same role

The Nose-Hoover thermostat generate dynamics that do not critically depend on the thermostat parameter Q.





In special cases, the Nose-Hoover thermostat fails to generate the canonical distribution, which arises from the conservation of another quantity in addition to H_{Nose} (we assumed in deriving the canonical distribution that it was the only conserved quantity). This can be taken care by introducing a series of thermostat variables that are coupled to each other (Nose-Hoover chains).

Constant pressure molecular dynamics

Along the same lines as constant pressure MD, in this case we write an extended Lagrangian, with two additional variables Instead of one.The equation of motion is given by:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} + \frac{p_\epsilon}{W}\mathbf{r}_i$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - (1 + \frac{1}{N})\frac{p_\epsilon}{W}\mathbf{p}_i - \frac{p_\xi}{Q}\mathbf{p}_i$$

The variable that determines volume change is $\epsilon = ln(V/V_0)$
And the corresponding equations of motion are

$$\dot{V} = \frac{d\,V p_\epsilon}{W},$$

$$\dot{p}_\epsilon = d\,V(P_{int} - P_{ext}) + \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i} - \frac{p_\xi}{Q}p_\epsilon$$

P_{int} and P_{ext} are the internal pressure (from virial) and the applied pressure

Some Special Cases and Techniques

There are special cases where specific methods that are distinct from those discussed above are used. Some of these are:

1. Discontinuous potentials (hard spheres etc): Event driven molecular dynamics

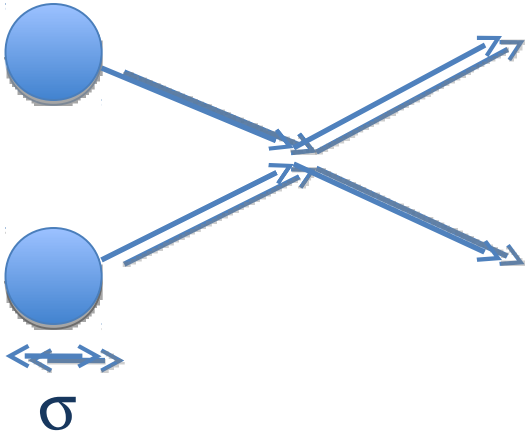
2. Systems with long range interactions: Ewald Sum

3. Molecular systems: SHAKE

Event Driven Molecular Dynamics

In systems interacting with hard core, or step potentials, the momentum change is discontinuous upon contact, and therefore the fixed time step integration methods are not useful.

Instead, the system is propagated from one collision event (hard sphere case) to the next, and the momentum change implemented at the collision, with the following steps:



1. Locate next collision. For a pair I,j, the condition for collision time is

$$\left|\mathbf{r}_{ij} + \mathbf{v}_{ij} t_{ij}\right| = \sigma$$

$$v_{ij}^2 t_{ij}^2 + 2b_{ij} t_{ij} + r_{ij}^2 - \sigma^2 = 0 \qquad b_{ij} = \mathbf{r}_{ij} . \mathbf{v}_{ij}$$

2. Cases:

i.If b_{ij} > 0, particles are moving away.

ii.Root of equation are complex if $b_{ij}^2 - v_{ij}^2 (r_{ij}^2 - \sigma^2) < 0$

iii.Else smaller of the roots is the collision time:

$$t_{ij} = \frac{-b_{ij} - (b_{ij}^2 - v_{ij}^2 (r_{ij}^2 - \sigma^2))^{1/2}}{v_{ij}^2}$$

3. Find smallest collision time among all pairs.

4. Propagate all particles up to the collision time.

5. For colliding particles, invert velocities in the direction of Collision:

$$\mathbf{v}_i = \mathbf{v}_i^o + \delta\mathbf{v} \qquad \mathbf{v}_j = \mathbf{v}_j^o - \delta\mathbf{v} \qquad \delta\mathbf{v} = -\frac{b_{ij}}{\sigma^2}\mathbf{r}_{ij}$$

6. Repeat. Keep track of time elapsed.

Since the dynamics is updated according to the collision events, this method is termed event driven molecular dynamics.

Molecular Systems

For molecular systems, we have both intermolecular and intra-molecular distances and forces to contend with.

Intramolecular vibrations are much faster than intermolecular motion, and often not of interest.

Hence, use rigid molecular models. Eg, SPC/E model of water:

OH distance : $1\ \dot{A}$

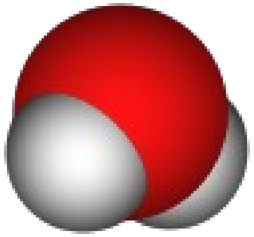
HOH angle : 109.47^0

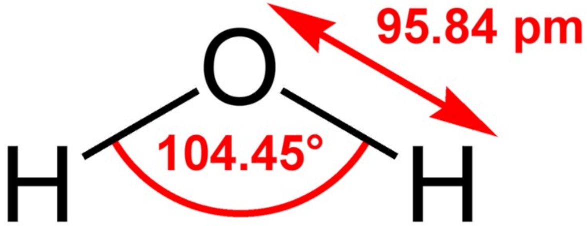
$q_H = 0.4238$ e

$q_O = -2q_H$

O-O interaction via LJ

$\sigma = 3.166\dot{A}\ \epsilon = 0.6502\frac{KJ}{mol}$





OH distances and HOH angle do not change during simulation.

How to handle such fixed distances and angles?

1.Work with centre of mass + rotational degrees of freedom.
Too complicated.

2. Add constraint forces to keep shape of molecule fixed.

$$F_{i\alpha} = F_{i\alpha}^{free} + F_{i\alpha}^{constraint}$$

Define constraint parameter $\chi_{12} = r_{12}^2 - d_{12}^2 = 0$

The force, including constraints is

$$F_1 = F_1^{free} + \frac{1}{2}\lambda_{12}\ \nabla_{r_1}\chi_{12}$$

Implementing all the constraints

$$F_1^c = \lambda_{12}\vec{r}_{12} - \lambda_{31}\vec{r}_{31}$$

$$F_2^c = \lambda_{23}\vec{r}_{23} - \lambda_{12}\vec{r}_{12}$$

$$F_3^c = \lambda_{31}\vec{r}_{31} - \lambda_{23}\vec{r}_{23}$$

The Verlet update will be

$$r_\alpha(t+\delta t) = r'_\alpha(t+\delta t) + \frac{\delta t^2 F_\alpha^c}{m_\alpha}$$

$$r_{12}(t+\delta t) = r'_{12}(t+\delta t) + \delta t^2(m_1^{-1} + m_2^{-1})\lambda_{12}r_{12}(t) - \delta t^2 m_2^{-1}\lambda_{23}(t) - \delta t^2 m_1^{-1}\lambda_{31}r_{31}(t)$$

where primed coordinates are updates without constraint.

The constraints, written as

$$|r_{12}(t+\delta t)|^2 = |r_{12}(t)|^2 = d_{12}^2$$

yield quadratic equations for the parameters λ. But instead, we linearize and iteratively solve the equation to covergence.

$$r_{i\alpha}(t+\delta t) = r_{i\alpha}(t+\delta t) + \frac{1}{2}(\frac{m_\beta}{m_\alpha + m_\beta})\left[\frac{r_{\alpha\beta}^2(t+\delta t) - d_{\alpha\beta}^2}{r_{\alpha\beta}(t+\delta t)}\right]$$

$$r_{i\beta}(t+\delta t) = r_{i\beta}(t+\delta t) - \frac{1}{2}(\frac{m_\alpha}{m_\alpha + m_\beta})\left[\frac{r_{\alpha\beta}^2(t+\delta t) - d_{\alpha\beta}^2}{r_{\alpha\beta}(t+\delta t)}\right]$$

Long Range Interactions: Ewald Sum

Normally, we trunctate the potential so that we need to consider only the minimum image.

This method will not work for long range interactions (Coulomb interactions)

In the Ewald sum method, the potential due to other charges, and all their periodic images, is carried out efficiently.

This is done by introducing a “screening” charge distribution around each point charge, and adding a second compensating charge distribution. The Coulomb interactions with screened charges are now short ranged, and can be treated as such.

The potential due to the compensating charges (and all its images) is efficiently calculated in Fourier space, IF we include also the charge distributions in the primary simulation cell.

This introduces a “self” interaction term, that needs to be compensated.

Consider point charges q_i.Their total interaction is written as

$$U_{Coul} = \frac{1}{2} \sum_{i=1}^{N} q_i \phi(r_i)$$

Where

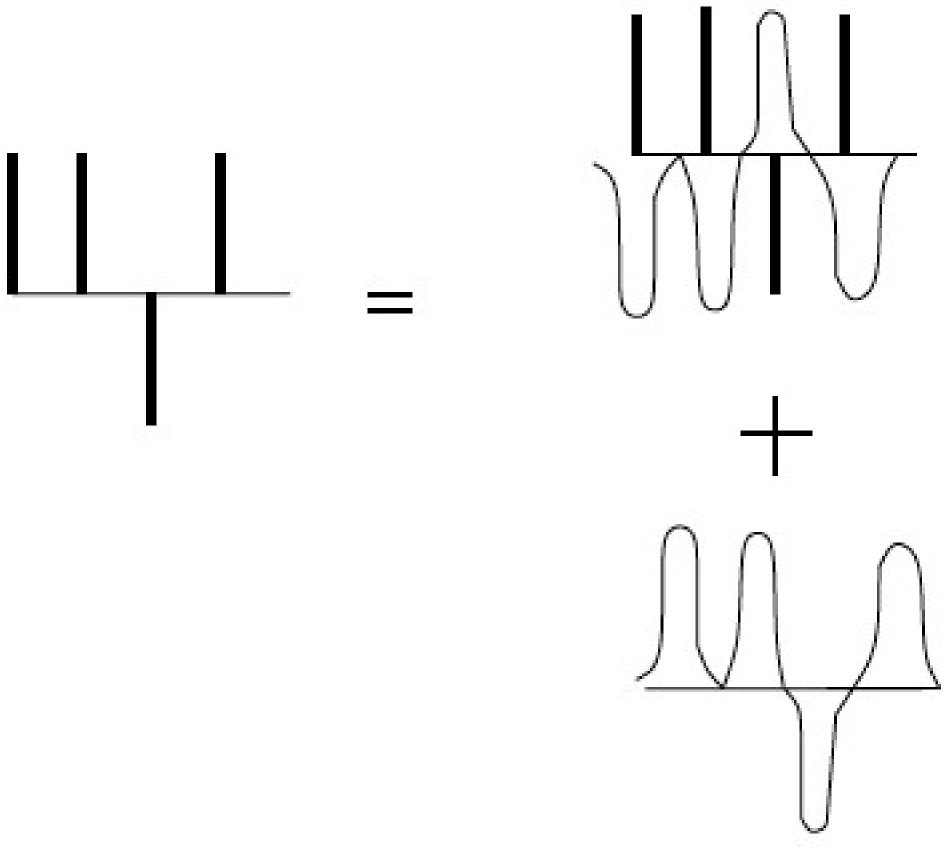
$$\phi(r_i) = \sum_{j,\mathbf{n}}{}' \frac{q_j}{|\mathbf{r}_{ij} + \mathbf{n}L|}$$

Exclusion of $j = i, \mathbf{n} = 0$ Indicated by the prime ‘

Add and subtract charge distributions for each point charge:

$$\rho(r) = -q_i (\frac{\alpha}{\pi})^{3/2} exp\left[-\alpha r^2\right]$$

Schematic:



Interaction with compensating charges

First we calculate he potential at r_i due to periodic array of compensating charges

$$\rho_{comp}(r) = \sum_{\mathbf{n}\ \mathbf{j}} q_j \left(\frac{\alpha}{\pi}\right)^{3/2} exp\left[-\alpha \mid \mathbf{r} - (\mathbf{r_j} + \mathbf{nL}) \mid^2\right]$$

The Fourier transform of the charge distribution is:

$$\begin{aligned}\rho_{comp}(\mathbf{k}) &= \frac{1}{V}\int_{allspace} \mathbf{d}\mathbf{r} exp\left[-i\mathbf{k}.\mathbf{r}\right] \sum_{j=1}^{N} q_j \left(\frac{\alpha}{\pi}\right)^{3/2} exp\left[-\alpha \mid \mathbf{r} - \mathbf{r_j} \mid^2\right] \\ &= \frac{1}{V}\sum_{j=1}^{N} exp\left(-i\mathbf{k}.\mathbf{r_j}\right) exp\left(-\frac{k^2}{4\alpha}\right)\end{aligned}$$

We obtain the electrostatic potential by solving the Poisson equation in Fourier space

$$-\nabla^2\phi(r) = 4\pi\rho(r)$$

$$k^2\phi(k) = 4\pi\rho(k)$$

$$\phi(k) = \frac{4\pi}{k^2}\rho(k)$$

The potential from the compensating charges then is:

$$\phi_{comp}(k) = \frac{4\pi}{k^2}\frac{1}{V}\sum_{j=1}^{N} q_j exp\left(-i\mathbf{k}.\mathbf{r_j}\right) exp\left(-\frac{k^2}{4\alpha}\right)$$

In real space

$$\phi_{comp}(r) = \sum_{k\neq 0}\sum_{j=1}^{N}\frac{4\pi q_j}{k^2} exp\left[i\mathbf{k}.(\mathbf{r}-\mathbf{r_j})\right] exp\left(-\frac{k^2}{4\alpha}\right)$$

The potential for all the charges is

$$\begin{aligned} U_{comp} &= \frac{1}{2}\sum_i q_i \phi_1(r_i) \\ &= \frac{1}{2V}\sum_{k\neq 0}\sum_{i,j=1}^{N} \frac{4\pi q_j q_i}{k^2} exp\left[i\mathbf{k}.(\mathbf{r_i}-\mathbf{r_j})\right] exp\left(-\frac{k^2}{4\alpha}\right) \\ &= \frac{V}{2}\sum_{k\neq 0}\frac{4\pi}{k^2} \mid \rho(k) \mid^2 exp\left(-\frac{k^2}{4\alpha}\right). \end{aligned}$$

with $$\rho(k) = \frac{1}{V}\sum_{i=1}^{N} q_i exp\left[\mathbf{ik.r_i}\right]$$

Correction for self interaction

The previous summation includes also the compensating charge distribution at the location of the charge. To subtract it, we calculate the potential due to a gaussian charge distribution at a distance r:

$$\phi_{Gauss}(r) = q_i \frac{erf(\sqrt{\alpha} r)}{r}$$

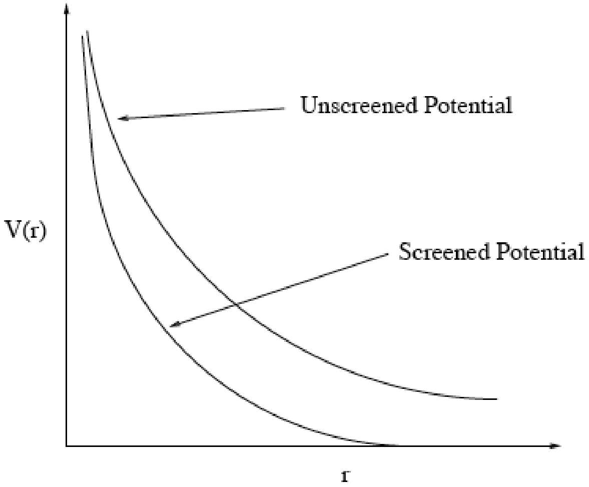
The self interaction therefore is

$$\begin{aligned} U_{self} &= \frac{1}{2} \sum_{i=1}^{N} q_i \phi_{self}(r_i) \\ &= (\frac{\alpha}{\pi})^{1/2} \sum_{i=1}^{N} q_i^2 \end{aligned}$$

Interaction with screened charges

The potential due to screened charges is

$$\begin{aligned}\phi_{screened}(r) &= \frac{q_i}{r} - \frac{q_i}{r} erf(\sqrt{\alpha} r) \\ &= \frac{q_i}{r} erfc(\sqrt{\alpha} r)\end{aligned}$$



The energy

$$U_{screened} = \frac{1}{2} \sum_{i,j} q_i q_j erfc(\sqrt{\alpha} r)/r_{ij}$$

The Ewald sum energy

Combining all these terms

$$\begin{aligned}
U_{Coul} &= \frac{V}{2}\sum_{k\neq 0}\frac{4\pi}{k^2} \mid \rho(k) \mid^2 exp\left(-\frac{k^2}{4\alpha}\right)\\
&- (\frac{\alpha}{\pi})^{1/2}\sum_{i=1}^{N} q_i^2\\
&+ \frac{1}{2}\sum_{i\neq j} q_i q_j erfc(\sqrt{\alpha} r)/r_{ij}
\end{aligned}$$