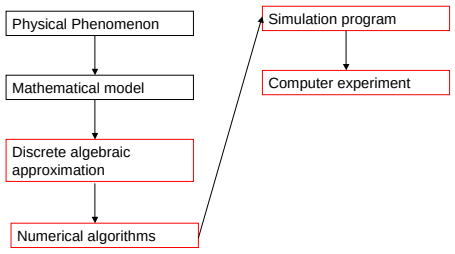
MD simulation model: P3M

Bin Fang

Contents

- Introduction
- Physical system
- PP method
- PM method
- P3M method
- How to solve the Poisson's equation

Introduction



- Methods of discretization used in obtaining simulation models:
  1. Finite-difference methods
  2. Finite-element methods
  3. particle methods.

- "particle models" is a generic term for the class of simulation models where the discrete representation of physical phenomena involves the use of interacting particles.
- In most applications,
  the particles=physical objects
  attributes: mass, charge, velocity, position.
  the state of the system is defined by the
  attributes of a finite ensemble of particles
  and the evolution of the system is determined by the laws of interaction of the particles.

Physical systems

- Correlated system, in which there exist one-to-one correspondences between physical and computer particles.
  Computer experiments with these systems do statistical mechanics the hard way, following the path of the system in sN-dimensional phase space.

- *Action at a distance* formulations

$$\mathrm{F}_i = \sum_{j \neq i} \mathrm{F}_{ij} + \mathrm{F}_i^{ext}$$

A familiar example is Coulomb's law, giving the electrostatic force between two charged particles as

$$\mathrm{F}_{ij} = \frac{q_i q_j}{4\pi\varepsilon_0} \frac{(\mathrm{x}_i - \mathrm{x}_j)}{|\mathrm{x}_i - \mathrm{x}_j|^3}$$

The potential energy of particle i,

$$\Psi_i = \sum_{j \neq i} \psi_{ij} + \Psi_i^{ext}$$

Where, for point charges

$$\psi_{ij} = \frac{q_i q_j}{4\pi\varepsilon_0} \frac{1}{|\mathrm{x}_i - \mathrm{x}_j|}$$

- *Force at a point* formulations

Forces and potentials may equally be regarded as fields

$$\mathrm{F} = -\nabla\Psi$$

$$\mathrm{F}_i = \mathrm{F}(\mathrm{x})|_{\mathrm{x}=\mathrm{x}_i} \qquad \Psi_i = \Psi(\mathrm{x})|_{\mathrm{x}=\mathrm{x}_i}$$

The potential field is related to the source distribution by the field equation, e.g. for point charges, we got Poisson's equation:

$$\nabla^2\phi = -\rho/\varepsilon_0$$

Where $\Psi_i = q_i\phi|_{\mathrm{x}=\mathrm{x}_i}$, ϕ Is the electrostatic potential and ρ is charge density

The Particle-particle method

The state of the physical system at some time t is described by the set of particle positions and velocities

$$\{\mathrm{x}_i(t), \mathrm{v}_i(t); i = 1, N_p\}$$

The timestep loop updates these values using the forces of interaction and equations of motion to obtain the state of the system at a slightly later time:

1.Compute forces
clear force accumulators
for i=1 to Np do
$\mathrm{F}_i := 0$
Accumulate forces
For i=0 to Np-1 do
For j=i+1 to Np do
Find force Fij of particle j on particle i

$$\mathrm{F}_i := \mathrm{F}_i + \mathrm{F}_{ij}$$

$$\mathrm{F}_j := \mathrm{F}_j - \mathrm{F}_{ij}$$

2. Integrate equations of motion
for i=1 to Np do

$$\mathrm{v}_i^{new} := \mathrm{v}_i^{old} + \frac{\mathrm{F}_i}{m_i} DT$$

$$\mathrm{x}_i^{new} := \mathrm{x}_i^{old} + \mathrm{v}_i DT$$

3. Update time counter
t:=t+DT

- Operation count

Calculation		Operations count
Clear force accumulator		3Np
Do for Np(Np-1)/2 particle pairs		
Compute $\mathrm{x}_i - \mathrm{x}_j$	3	
Compute $\lvert \mathrm{x}_i - \mathrm{x}_j \rvert^3$	8	
Compute F_{ij}	3	
Update F_i and F_j	6	
	20	10Np(Np-1)
Update v and x		6Np
	total	10Np*Np-Np

- Typically, a few thousand timesteps are needed to obtain useful results from a computer experiment, so the direct summation PP method is viable only for systems of up to approximately a thousand particles if forces are long ranged.
  e.g. stellar clusters
- We can do more particles if the forces are short range.
  e.g. the Lennard-Jones model of atomic liquids

The Particle-mesh method

- PM method exploits the force-at-a-point formulation and a field equation for the potential.
- The result is a much faster, but generally less accurate, force calculation than is obtained using the PP method

Physical system	PM approximation
Field quantities, which pervade all space	Values on a regular array of mesh points
Differential operator	Finite-difference approximation
Potentials and forces at particle positions	Interpolate on the array of mesh-defined values

- The timestep loop of the PM method differs from that of the PP method only in the calculation of the forces.
- The PM force calculation corresponding to the charged-particle PP example given above consists of three steps:

1. assign charge to mesh
2. Solve Poisson's equation on the mesh
3. Compute forces from the mesh-defined potential and interpolate forces at particle positions.

- Operations count
  operations count = $\alpha N_p + \beta(N)$

  Where α and β depend on the particular form of PM scheme being used

  We will take some characteristic values in order to make a comparison with PP method:

  $\alpha = 20$

  $\beta(N) = 5N^3 \log_2 N^3$ for an $(N \times N \times N)$ mesh

  e.g. taking N=32, Np=10^5, and a machine with a nominal CPU time per
  Floating point operation of 1us, we obtain

  CPU time=(20*10^5+5*32^3*15)*10^-6 seconds
  =4.5 seconds
  ~1day for the PP method

- The disadvantage of the PM method
  Loss of resolution in the potential and force fields
  only those field variations which have wavelength longer than the spacing of the mesh can be accurately represented by mesh values.

  PM method only can handle smoothly varying forces.

P3M method

- P3M method combines the advantages of the PP and PM methods and enables large correlated systems with long-range forces to be simulated.

  The trick used in the P3M method is to split the interparticle forces into two parts

$$\mathrm{F}_{ij} = \mathrm{F}_{ij}^{sr} + \mathrm{F}_{ij}^{m}$$

Where the rapidly varying short range part F_{ij}^{sr} is nonzero for only a few interparticle distances and the slowly varying part F_{ij}^{m} is sufficiently smooth to be accurately represented on a mesh.

The addition of the total short-range and total slowly varying force contributions gives the total forces on each particle which is used to update the velocities.

- Operations count

  operations count= $\alpha N_p + \beta(N) + \gamma N_n N_p$

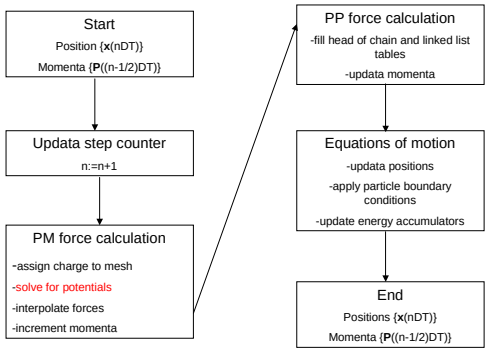
- P3M method was developed by Eastwood and Hockney in 1973.

Particle orbits are integrated forwards in time using the leapfrop scheme: (we restrict discussions to long-range forces which are purely coulombic.)

$$\mathrm{x}_i^{n+1} = \mathrm{x}_i^n + \frac{\mathrm{P}^{n+1/2}}{m_i} DT$$

$$\mathrm{P}_i^{n+1/2} = \mathrm{P}_i^{n-1/2} + (\mathrm{F}_i + \mathrm{F}_i^{sr}) DT$$

Position $\{\mathrm{x}_i\}$ are defined at integral timelevels and momenta $\{\mathrm{P}_i\}$ are defined at half-integral timelevels. The reason that we use momenta rather than velocities in P3M computations is for computational economy.



Solve for potentials

Problem: $\begin{cases} \dfrac{\partial^2\phi(x,y)}{\partial x^2}+\dfrac{\partial^2\phi(x,y)}{\partial x^2}=\rho(x,y) \\ \phi(x+kL,y)=\phi(x,y) \\ \phi(x,y+kL)=\phi(x,y) \end{cases}$

Consider a square region covered by a square 48*48 mesh, with the boundary Condition that the solution be periodically repeated in both the x and y directions

$$\begin{cases} \phi_{i-1,j}+\phi_{i+1,j}+\phi_{i,j-1}+\phi_{i,j+1}-4\phi_{i,j}=q_{i,j} \quad (i.j=0,1,...,47) \\ \phi_{i+48k,j+48k}=\phi_{i,j}, \quad q_{i+48k,j+48k}=q_{i,j} \end{cases}$$

$$B\phi=\begin{pmatrix} A & I & 0 & \dots & 0 & I \\ I & A & I & & & \vdots \\ \vdots & & \ddots & & & 0 \\ 0 & & & I & A & I \\ I & 0 & \dots & 0 & I & A \end{pmatrix}\begin{pmatrix} \phi_0 \\ \phi_1 \\ \vdots \\ \phi_{47} \end{pmatrix}=\begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{47} \end{pmatrix} \quad \text{where} \quad \phi_j=\begin{pmatrix} \phi_{0j} \\ \phi_{1j} \\ \vdots \\ \phi_{47j} \end{pmatrix}, \quad q_j=\begin{pmatrix} q_{0j} \\ q_{1j} \\ \vdots \\ q_{47j} \end{pmatrix}$$

and

$$A=\begin{pmatrix} -4 & 1 & 0 & \dots & 0 & 1 \\ 1 & -4 & 1 & & & 0 \\ \vdots & & \ddots & & & \vdots \\ 0 & & & 1 & -4 & 1 \\ 1 & 0 & \dots & & 0 & 1 & -4 \end{pmatrix}$$

Odd/even reduction

$$\phi_{j-2}+(2I-A^2)\phi_j+\phi_{j+2}=q_{j-1}-Aq_j+q_{j+1}=q^*_{j+1} \quad (1)$$
$$(j=0,2,...,46)$$

Fourier Transform:

Goal: $Q^T(2I-A^2)Q=\Lambda$

Where Q's columns are the eigenvectors of $(2I-A^2)$. Since A is symmetric and AI=IA, Q's columns are the eigenvectors of A. Λ Is the diagonal matrix whose elements are the eigenvalues of $(2I-A^2)$

Do the Fourier transform in x direction only:

$$\phi_{i,j}=\frac{1}{2}\bar{\phi}^c_{0,j}+\frac{1}{2}\bar{\phi}^c_{24,j}(-1)^i+\sum_{k=1}^{23}\left\{\bar{\phi}^c_{k,j}\cos\frac{2\pi ki}{48}+\bar{\phi}^s_{k,j}\sin\frac{2\pi ki}{48}\right\} \quad (2)$$

Substituting the expansion (2) into (1) and using the orthogonality relations, we get the finite fourier transform of equation (1)

$$\bar{\phi}_{k,j-2} + \lambda_k \bar{\phi}_{k,j} + \bar{\phi}_{k,j+2} = \bar{q}^*_{k,j} \qquad (3)$$

Where $\bar{\phi}$ and $\bar{q}^*$ refer to either the sine or cosine harmonic and

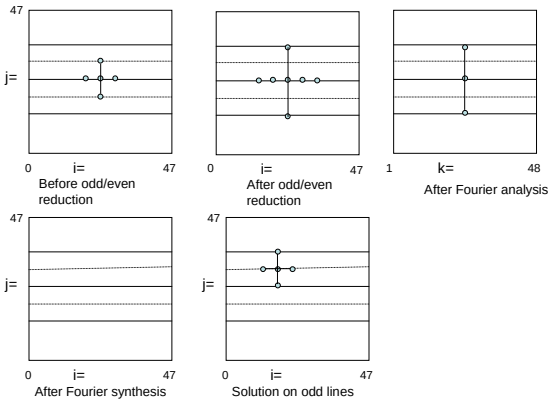
$$\lambda_k = -2\left(8 - 8\cos\frac{2\pi k}{48} + \cos\frac{4\pi k}{48}\right).$$

The equations (3) are 48 independent sets of 24 equations, one set for each of the 48 harmonic amplitudes.

After transformation, we can use several methods to solve the 48 24-equation systems to get the k-space solutions on the even lines. The method includes Recursive Cyclic Reduction.

Then Fourier synthesis would be applied to the k-space solutions on even lines to get the x-space solutions on even lines.

At last, by exact interpolation, the solution on odd lines can be obtained.



Thank you.