Introduction to Molecular Dynamics

CS 471/Math 471

Brian T. Smith, UNM

Andrew C. Pineda, UNM

Molecular Dynamics

- Simulation of the motion of molecules to gain an understanding of their interaction
  - chemical reactions
  - fluid flow
  - phase transitions
  - …..
- Reference: Ch. 15, IHPSC

Physics of MD and MC

- Connection to classical statistical mechanics
  - Average values (NVT ensemble averages) of an observable *A* (e.g. P, E, C_V) can be computed as

$$\langle A \rangle = \frac{\int d\,\vec{r}^{\,N} d\,\vec{p}^{\,N} e^{-\beta H(\vec{r}^{\,N},\vec{p}^{\,N})} A(\vec{r}^{\,N},\vec{p}^{\,N})}{\int d\,\vec{r}^{\,N} d\,\vec{p}^{\,N} e^{-\beta H(\vec{r}^{\,N},\vec{p}^{\,N})}}$$

where *H* is the Hamiltonian and $\beta = 1/k_B T$

MC/MD Introduction Cont'd.

- When *A* is a function of position only, this reduces to:

$$\langle A \rangle = \frac{\int d\vec{r}^{\,N} e^{-\beta U(\vec{r}^{\,N})} A(\vec{r}^{\,N})}{\int d\vec{r}^{\,N} e^{-\beta U(\vec{r}^{\,N})}}$$

- the term in the denominator is commonly called the configurational integral.

Physics of MD and MC continued

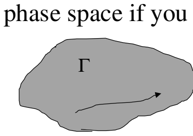
– Can also compute <*A*> as a time average:

$$< A >= \lim_{t\to\infty} \frac{1}{t}\int_0^t dt' A(t') = \lim_{t\to\infty} \frac{1}{t}\int_0^t dt' A(r^N(t'), p^N(t'))$$

where $\;^{N}(0)\;\;^{N}(0)$ are arbitrary initial conditions consistent with N,V,E. (*Ergodic hypothesis, not true in general.*) Can also average over initial conditions consistent with N,V,E.

Ergodic hypothesis

- The idea behind the ergodic hypothesis is that trajectories in phase space Γ pass near every point in phase space if you wait long enough.
- Also, the time they spend in a region is proportional to its ensemble weight.



Monte Carlo Method

- The first method forms the basis for Monte Carlo approach
  - Naïve random sampling of configurations produces terms in which the exponential is small. This leads to a large variance (error) in the computed quantity.
  - $\therefore$ need a variance reduction technique. Metropolis "importance sampling" algorithm basis for doing ensemble integral efficiently.

MD Introduction Cont'd

- 2nd Method is basis for MD.
  - Uses classical Newtonian mechanics
    - forces depend on distances between objects
    - naturally used for constant N,V,E systems
  - Utilizes the speed of computers and their ability to repeat simple computations millions of times
  - An early use of machines and continues to consume many cycles
    - initially 100's of molecules simulated
    - now millions of cycles, many time steps (10^5)

MD Introduction Cont'd

- Discrete models for potentials and forces
  - attractive and repulsive forces
  - n particles yield in general $O(n^2)$ operations
  - large simulations take advantage of forces tailing off with larger distances (beyond a cutoff, forces are assumed to be zero)
- To illustrate the principles, three simple models are used -- Hooke's law, Lennard-Jones potential, hard-sphere interaction

Three Interaction Models

- Hooke's Law:
  - forces proportional to the displacements
- Lennard-Jones Potential:
  - strong repulsive force when particles are close
  - weak attractive force when particles are far apart
- Hard-Sphere Interaction:
  - no force when separated
  - large instantaneous repulsive force when they touch

Models

- Conservative 2-body forces
  - conservation of energy
  - force on particle i is the sum of the forces exerted by all particles j from 1 to n
  - the force is a vector sum $f_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} f_{ij}$
  - Newton's third law implies the relation

$$f_{ij} + f_{ji} = 0$$

Potential Energy

- The force between particles is dependent on the distance r between the particles
- These forces can be defined in terms of the potential energy $\Phi(r)$ between the particles: $f(r) = -\nabla\Phi(r)$
- Can distinguish the three different interaction models by the potential energy used by each model

Potential Energy Definitions

- Hooke's Law Potential: $\Phi(r) = \| r \|^2$
- Lennard-Jones Potential: $\Phi(r) = \dfrac{1}{\| r \|^{12}} - \dfrac{2}{\| r \|^{6}}$
- Hard-Sphere Potential: $\Phi(r) = \begin{cases} 0, & \| r \| > \sigma \\ \infty, & \| r \| \leq \sigma \end{cases}$

Hooke's Law Model

- Hooke's law potential and forces in 3-D (assume the origin is the point of equilibrium):

$$\Phi(r) = x^2 + y^2 + z^2$$
$$f_x = -\frac{\partial \Phi}{\partial x} = -2x$$
$$f_y = -\frac{\partial \Phi}{\partial y} = -2y$$
$$f_z = -\frac{\partial \Phi}{\partial z} = -2z$$

1-D Hooke's Law

- Assume the equilibrium point is: x^{eq}
- Assume the force constant is k
- Potential energy in terms of distance from the equilibrium point x^{eq} is:

$$\Phi(x) = \frac{k}{2}(x - x^{eq})^2 + \Phi_{\min}$$

- Force is: $f(x) = -k(x - x^{eq})$
  - $f(x) > 0$ when particle is to left of equilibrium point
  - $f(x) < 0$ when particle is to right of equilibrium point

Hooke's Law -- A Chain Of Particles

- 2 molecules connected by a spring
- Let the positions be x_i
- Let the particles be a distance d apart at equilibrium
- The equation for potential energy and force

$$\Phi(x_1, x_2) = \frac{k}{2}(x_1 - x_2 + d)^2$$
$$f_1 = -k(x_1 - x_2 + d)$$
$$f_2 = k(x_1 - x_2 + d)$$

A Chain Continued

- Consider 4 particles connected by 3 springs which are at a distance d apart when in equilibrium. The springs have the same k.
- Potential energy and forces are:

$$\Phi(x^1, x^2, x^3, x^4) = \frac{k}{2}\left((x^1 - x^2 + d)^2 + (x^2 - x^3 + d)^2 + (x^3 - x^4 + d)^2\right) + \Phi^{\min}$$
$$f^1(x^1, x^2) = -k(x^1 - x^2 + d)$$
$$f^2(x^1, x^2, x^3) = -k(2x^2 - x^1 - x^3)$$
$$f^3(x^2, x^3, x^4) = -k(2x^3 - x^2 - x^4)$$

Further Simplifications

- Let q_i be the relative distances; that is,

$$x_i = (i-1)d + q_i$$

- Then the potential and forces are:

$$\Phi(q^1, q^2, q^3, q^4) = \frac{k}{2}\left((q^1 - q^2)^2 + (q^2 - q^3)^2 + (q^3 - q^4)^2\right) + \Phi^{\min}$$
$$f^1(q^1, q^2) = -k(q^1 - q^2)$$
$$f^2(q^1, q^2, q^3) = -k(2q^2 - q^1 - q^3)$$
$$f^3(q^2, q^3, q^4) = -k(2q^3 - q^2 - q^4)$$

Equations Of Motion

- The equation of motion for a chain of particles

$$ma_i = m\ddot{x}_i = f_i$$

- where *a* and *f* are acceleration and force vectors for each particle of the same mass. Substituting the forces gives:

$$\begin{aligned} m\ddot{q}_1 &= -k(q_1 - q_2) \\ m\ddot{q}_2 &= -k(2q_2 - q_1 - q_3) \\ m\ddot{q}_3 &= -k(2q_3 - q_2 - q_4) \\ m\ddot{q}_4 &= -k(q_4 - q_3) \end{aligned}$$

Numerical Solutions

– The Hooke's Law (HL) ODE is 2nd order.

– It can be changed to 2 first order ODEs by introducing a variable for the velocity

$$\begin{aligned} m\ddot{x} &= f \\ \dot{x} &= u \\ m\dot{u} &= f \end{aligned}$$

– The solution technique based on Euler's method

$$x(t+h) \approx x(t) + hu(t)$$

$$u(t+h) \approx u(t) + \frac{h}{m} f(x(t))$$

Computer Program

– Recall that Euler's method is $O(h^2)$

```
X(0) = Init_loc; U(0) = Init_vel
do T = 0, NSTEP
  X(T+1) = X(T) + H*U(T)
  U(T+1) = U(T) + (H/M)*F(X(T))
enddo
```

– This code saves all data at all time steps

– F() is a function that evaluates the forces

– Can be readily generalized to multiple particles and multiple dimensions

Verlet's Method

- Verlet's method is $O(h^4)$. It treats the equation of motion as a 2nd order ODE and approximates the 2nd derivative with finite difference

$$\ddot{x} \approx \frac{x(t+h) - 2x(t) + x(t-h)}{h^2}$$

$$x(t+h) \approx 2x(t) - x(t-h) + h^2 f(x(t))$$

Starting Verlet and The Summed Form

– We need a value for *x(h)* of order $O(h^3)$ in order to start the method

$$x(h) \approx x(0) + hx'(0) + \frac{h^2}{} f(x(0))$$

– A summed form for this method reduces rounding errors. It is developed by defining dx(t) as (x(t+1)-x(t))/h and reformulating the computation -- it avoids adding small values to large values, reducing the loss of accuracy

Comparison of Summed and Unsummed Forms

- Unsummed form:

```
do T = 1, NSTEP
  X(T+1) = 2*X(T) - X(T-1) + H**2*F(X(T))
enddo
```

- Summed form

```
do T = 1, NSTEP
  DX(T) = DX(T-1) + H*F(X(T))
  X(T+1) = X(T) + H*DX(T)
enddo
```

Exact Solutions to HL Model

- 1-D case of a single molecule on a spring
  $$m\ddot{q}(t) = -kq(t), \quad q(0) = q_0, \quad \dot{q}(0) = \dot{q}_0$$
- The solution can be shown to be:
  $$q(t) = Q\cos(\omega t + \delta)$$
  where Q and δ are determined by initial conditions and
  $$\omega = \sqrt{\frac{k}{m}}$$

Exact Solution--Multiple Molecules

$$\begin{aligned} m\ddot{q}_1 &= -k(q_1 - q_2) \\ m\ddot{q}_2 &= -k(2q_2 - q_1 - q_3) \\ m\ddot{q}_3 &= -k(2q_3 - q_2 - q_4) \\ m\ddot{q}_4 &= -k(q_4 - q_3) \end{aligned}$$

$$m\ddot{q} = -k\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} q$$

Solutions To The Vector System

- Guessing at the solution
- Results in an eigenvalue problem
- Form of the solutions
- General form of the solutions

UNM Notes on Lab Chapter 8

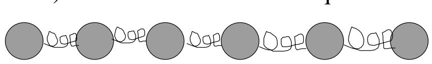
– Ignore the AVS notes and exercises; we do not have access to an AVS system on campus

– The directories containing the Euler codes, examples, and matlab directories are:

Orig. Dir.	New Location
euler	~acpineda/Labs8and9/MolecDyn/Euler
examples	~acpineda/Labs8and9/Chapter8Examples
matlab	~acpineda/Labs8and9/MatlabCode

Lab 8 Exercises

- Modeling of the dynamic behavior of molecules, using 1-D chain of particles
- Based on Hooke's law of motion
- Uses Euler's method (with $O(h)$ global error) to solve the differential equations



Purpose of Lab Chapters 8 and 9

– Solve the equations of motion numerically

– Study the approximation properties numerically and check that they follow the mathematical analysis

– Compare results with exact answers

– Display the results graphically and with motion

– Understand the numerical algorithms with an eye toward a parallel implementation

Notation

- 1-D Hooke's law model
  - particles are numbered left-to-right 1,2,…
  - particles interact with nearest neighbors
  - absolute position of the particles is x_i
  - relative position is q_i and all equations are expression in terms of the relative positions
  - assume that the particles at equilibrium are at a distance d apart
  - spring constant k/m is defined as a in the euler1_h1.f

Configuration of Three Particles

We give the exact solution for the 3 particle chain you will examine in lab. The potential energy function for the system is:

$$\Phi(x_1, x_2, x_3) = \frac{k}{2}\left[(x_1 - x_2 + d)^2 + (x_2 - x_3 + d)^2\right]$$

where I have picked the minimum energy to be zero.

Config. Of 3 particles cont'd.

- At equilibrium Φ is zero, and $x_{j+1} = x_j + d$

and the position of the 1st particle is arbitrary.

- Changing to relative coordinates, we have

$$\Phi(q_1, q_2, q_3) = \frac{k}{2}\left[(q_1 - q_2)^2 + (q_2 - q_3)^2\right]$$

and for the equations of motion

$$f_i = m\ddot{q}_i = -\frac{\partial \Phi}{\partial q_i} \text{ where } x_i = (i-1)d + q_i$$

Config. Of 3 particles cont'd.

- This leads to the following system of equations.

$$\ddot{q}_1 = -\omega^2(q_1 - q_2)$$
$$\ddot{q}_2 = \omega^2(q_1 + q_3 - 2q_2)$$
$$\ddot{q}_3 = \omega^2(q_2 - q_3)$$
$$\text{where } \omega^2 = k/m$$

Config. Of 3 particles cont'd.

- We look for solutions of the form $q_i(t) = \mathrm{Re}\left(Q_i e^{i\lambda t}\right)$

$$\begin{bmatrix} (\lambda^2 - \omega^2) & \omega^2 & 0 \\ \omega^2 & (\lambda^2 - 2\omega^2) & \omega^2 \\ 0 & \omega^2 & (\lambda^2 - \omega^2) \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

- the determinant of the matrix must vanish for a non-trivial solution.

Config. Of 3 particles cont'd.

- Setting the determinant to zero gives:

$\lambda_i^2 = \left\{0, \omega^2, 3\omega^2\right\}$ with eigenvectors:

$$Q_i = \left\{ \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} \right\}$$

Config. Of 3 particles cont'd.

- This allows us to write the solution as

$$\begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = (A+Bt)\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + C\cos(\omega t+\theta)\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} + D\cos(\sqrt{3}\omega t+\phi)\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$$

Translational mode, λ=0

2 vibrational modes

Monte Carlo Method Cont'd.

- Monte Carlo integration method uses importance sampling to speed up convergence.
- A 1-d example illustrates the ideas behind the method. Consider evaluating

$$I = \int_a^b dx f(x)$$

Monte Carlo Method Cont'd.

- We can rewrite this as *I=(b-a)<f(x)>* where *<f(x)>* is the average value of *f(x)* in the interval.
- In brute force Monte Carlo, we can evaluate *<f(x)>* as $\langle f(x)\rangle = \frac{1}{N}\sum_{j=1}^{N} f(x_j)$ where x_j

is a uniform random number on [*a*,*b*].

Monte Carlo Method Cont'd.

- The problem with the brute force approach is that it converges very slowly. From the central limit theorem we have in theory,

$$\left|I_{MC}(N) - I\right| < \lambda\sigma_I \approx O(N^{-1/2})$$

- So to get an extra digit of accuracy, the number of function evaluations must go up by 100.

Monte Carlo Method cont'd.

- The reason for the poor behavior is that the uniform distribution puts a lot of points in regions where *f(x)* is small.
- Solve problem by sampling w.r.t. another probability distribution, *w(x),* and rewrite as

$$I = \int_a^b dx w(x)\frac{f(x)}{w(x)} \text{ where } w(x) \geq 0 \text{ and } \int_a^b dx w(x) = 1.$$

Monte Carlo Method Cont'd.

- Can now change variable to $u = \int^x dx w(x)$

which defines the function *x(u),* so that

$$I = \int_0^1 du \frac{f[x(u)]}{w[x(u)]}$$

which has $\sigma_I^2 = \frac{1}{N}\left[\left\langle \left(f/w\right)^2 \right\rangle - \left\langle f/w \right\rangle^2\right]$

Monte Carlo Method Cont'd

- What have we gained? If we pick *w(x)* so that it is nearly proportional to *f(x)* we can make σ as small as we like.
- In the multi-dimensional case, we cannot do the change of variable explicitly, so instead we generate a sequence (chain) of random configurations such that their weight is proportional to the Boltzmann factor.

Monte Carlo Method Cont'd

– We generate the chain by starting the system in a state for which $e^{-\beta U(\vec{r}^N)}$ is not zero. We then make random displacements of the particles in the system and add them to the chain with probability

$$P = \begin{cases} e^{-\beta(U(n)-U(o))} & if\ U(n) > U(o) \\ 1 & if\ U(n) < U(o) \end{cases}$$

– If the new configuration is rejected, we count the old one in our average again.

Monte Carlo Method Cont'd

- In practice, the configuration we start from is not very likely, so we let the system evolve for a while before we begin computing our averages. (For example, in a MC simulation of a liquid the starting configuration is a crystal, so we have to allow the crystal to melt before we start averaging.)