

Molecular Dynamics Simulations How to Integrate

Banu Ozkan

Banu.Ozkan@asu.edu

The Nobel Prize in Chemistry 2013



© Nobel Media AB. Photo: A.

Mahmoud

Martin Karplus

Prize share: 1/3



© Nobel Media AB. Photo: A.

Mahmoud

Michael Levitt

Prize share: 1/3



© Nobel Media AB. Photo: A.

Mahmoud

Arieh Warshel

Prize share: 1/3

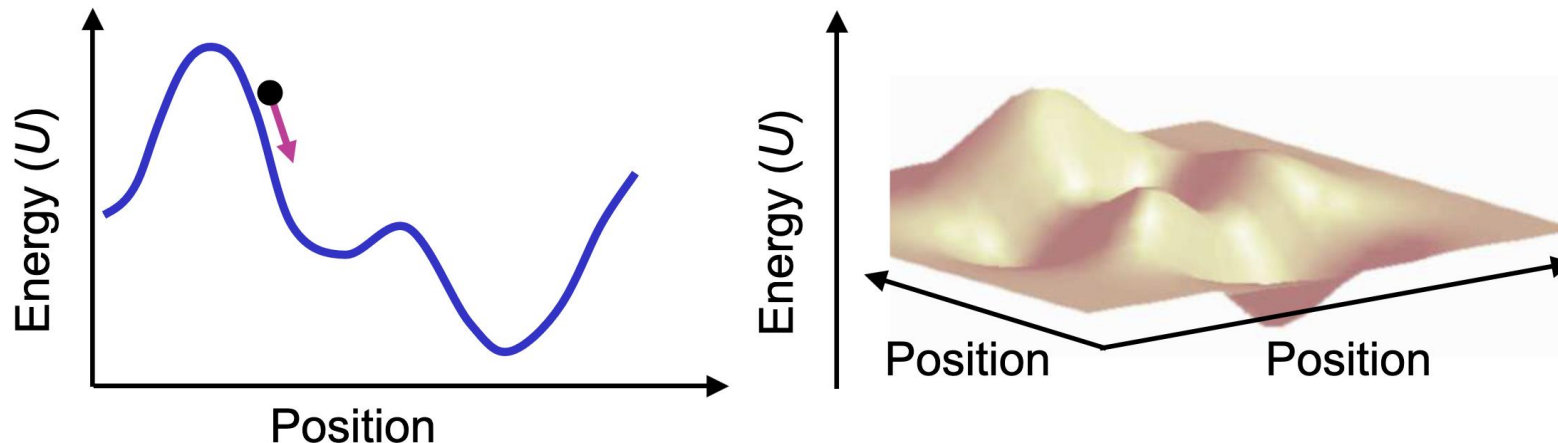
Chemists used to create models of molecules using plastic balls and sticks. Today, the modelling is carried out in computers. In the 1970s, **Martin Karplus, Michael Levitt** and **Arieh Warshel** laid the foundation for the powerful programs that are used to understand and predict chemical processes. Computer models mirroring real life have become crucial for most advances made in chemistry today

Relationship between energy and force

- Force on atom i is given by derivatives of U with respect to the atom's coordinates x_i , y_i , and z_i

$$\mathbf{F}(\mathbf{x}) = -\nabla U(\mathbf{x})$$

- At local minima of the energy U , all forces are zero



Force vector

- A single vector \mathbf{F} specifies the force acting on every atom in the system
- For a system with N atoms, \mathbf{F} is a vector of length $3N$
 - This vector lists the force on each atom in the x-, y-, and z- directions
- Notation:
 - Force on atom 1 in the x-direction: $F_{1,x}$
 - Rate of change of U as x_1 increases: $\frac{\partial U}{\partial x_1}$

$$\mathbf{X} = \begin{pmatrix} x_1 \\ y_1 \\ z_1 \\ x_2 \\ y_2 \\ z_2 \\ \vdots \\ x_N \\ y_N \\ z_N \end{pmatrix} \quad \mathbf{F} = \begin{pmatrix} F_{1,x} \\ F_{1,y} \\ F_{1,z} \\ F_{2,x} \\ F_{2,y} \\ F_{2,z} \\ \vdots \\ F_{N,x} \\ F_{N,y} \\ F_{N,z} \end{pmatrix} = - \begin{pmatrix} \frac{\partial U}{\partial x_1} \\ \frac{\partial U}{\partial y_1} \\ \frac{\partial U}{\partial z_1} \\ \frac{\partial U}{\partial x_2} \\ \frac{\partial U}{\partial y_2} \\ \frac{\partial U}{\partial z_2} \\ \vdots \\ \frac{\partial U}{\partial x_N} \\ \frac{\partial U}{\partial y_N} \\ \frac{\partial U}{\partial z_N} \end{pmatrix}$$

Note that U depends on which atoms are present in the system and the covalent bonds between them. Two different molecular systems with the same number of atoms will have different potential energy functions.

How can we formulate the potential energy
defining forces acting on a protein ?

Molecular Mechanics Force Fields: Functional Forms

The basic functional form of a force field encapsulates both bonded terms relating to atoms that are linked by covalent bonds, and nonbonded (also called "noncovalent") terms describing the long-range electrostatic and van der Waals forces. The specific decomposition of the terms depends on the force field, but a general form for the total energy can be written as

$$E_{total} = E_{covalent} + E_{noncovalent}$$

$$E_{covalent} = E_{bond} + E_{angle} + E_{dihedral}$$

$$E_{noncovalent} = E_{electrostatic} + E_{vanderWaals}$$

Potential Function-Force Field

$$\begin{aligned}
 U(\vec{R}) = & \underbrace{\sum_{bonds} k_i^{bond} (r_i - r_0)^2}_{U_{bond}} + \underbrace{\sum_{angles} k_i^{angle} (\theta_i - \theta_0)^2}_{U_{angle}} + \\
 & \underbrace{\sum_{dihedrals} k_i^{dihe} [1 + \cos(n_i \phi_i + \delta_i)]}_{U_{dihedral}} + \underbrace{\sum_i \sum_{j \neq i} 4\epsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \sum_i \sum_{j \neq i} \frac{q_i q_j}{\epsilon r_{ij}}}_{U_{nonbond}}
 \end{aligned}$$

PDB file → **geometry** (points to k_i^{bond} , k_i^{angle} , k_i^{dihe})
 Topology → **PSF file** (points to n_i , ϕ_i , δ_i)
parameters (points to ϵ_{ij} , σ_{ij} , q_i) ← **Parameter file**

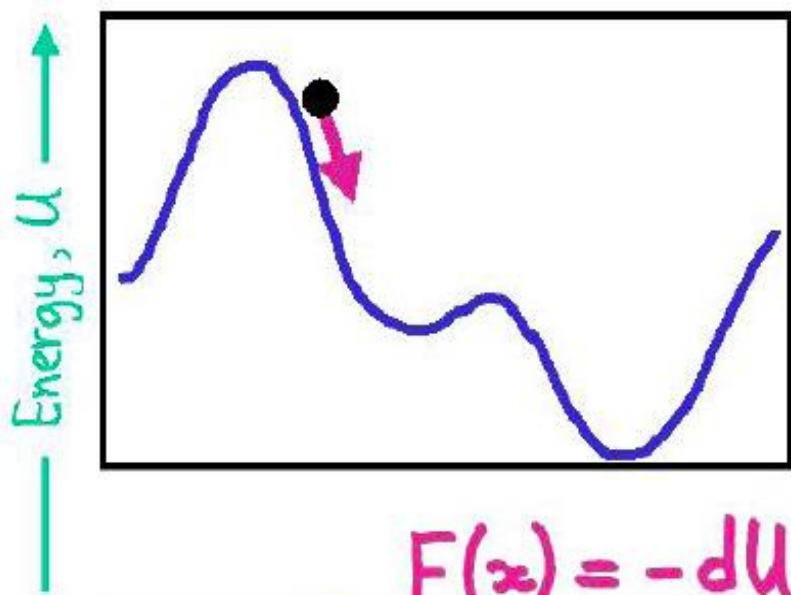
File Format/Structure

- The structure of a pdb file
- The structure of a psf file
- The topology file
- The parameter file
- Connection to potential energy terms

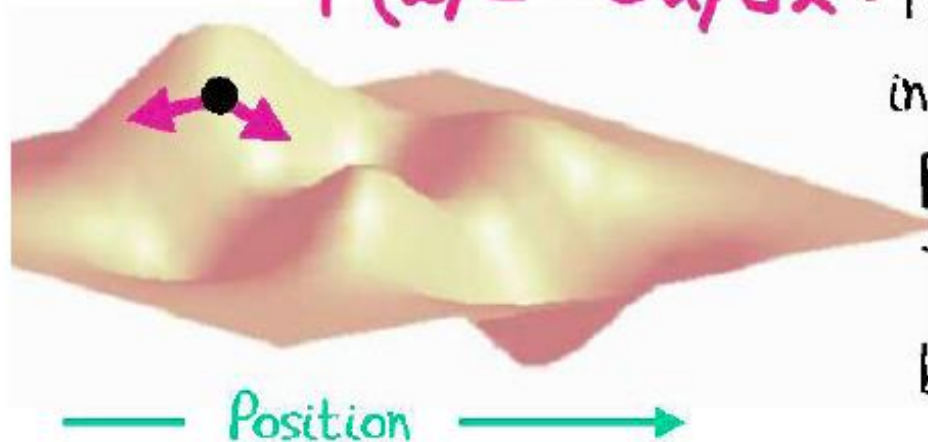
Checking file structures

- PDB file
- Topology file
- PSF file
- Parameter file

We can use Classical Mechanics to sample conformations on the energy-landscape



$$F(x) = -dU/dx$$



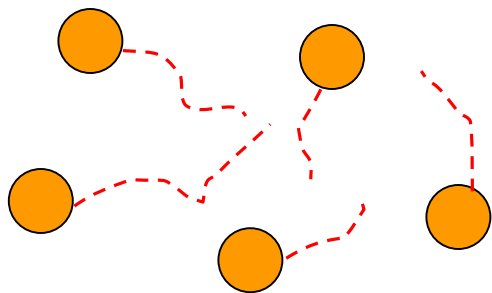
- The total potential energy or enthalpy fully defines the system, U .
- The forces are the gradients of the energy.
- The energy is a sum of independent terms for:
Bonds, Bond angles,
Torsion angles and non-bonded atom pairs.

Example 2: gradient of bond stretching potential with respect to r_i

- $E_i(\text{bs}) = (k_{\text{bs}}/2) (l_i - l_i^0)^2$
- $l_i = r_{i+1} - r_i$
 - $l_{ix} = x_{i+1} - x_i$
 - $l_{iy} = y_{i+1} - y_i$
 - $l_{iz} = z_{i+1} - z_i$
 - $l_i = [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2]^{1/2}$

$$\begin{aligned} \partial E_i(\text{bs}) / \partial x_i &= - m_i a_{ix}(\text{bs}) \text{ (induced by deforming bond } l_i) \\ &= (k_{\text{bs}}/2) \partial \{ [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2]^{1/2} - l_i^0 \}^2 / \partial x_i \\ &= k_{\text{bs}} (l_i - l_i^0) \partial \{ [(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2]^{1/2} - l_i^0 \} / \partial x_i \\ &= k_{\text{bs}} (l_i - l_i^0) (1/2) (l_i^{-1}) \partial (x_{i+1} - x_i)^2 / \partial x_i = - k_{\text{bs}} (1 - l_i^0 / l_i) (x_{i+1} - x_i) \end{aligned}$$

Molecular Dynamics Theory



- Force = $-dU/dx$ (slope of potential, U); acceleration, $m a(t) = \text{Force}$.
- All atoms move together so force between atoms change with time.
- Analytical solution for $x(t)$ and $v(t)$ is impossible; numerical solution is trivial.

$$\underset{\text{New position}}{x(t+\Delta t)} = \underset{\text{Old position}}{x(t)} + \underset{\text{Old velocity}}{v(t)\Delta t} + \underset{\text{Acceleration}}{[4a(t) - a(t-\Delta t)]\Delta t^2/6}$$

$$\underset{\text{New velocity}}{v(t+\Delta t)} = \underset{\text{Old velocity}}{v(t)} + \underset{\text{Acceleration}}{[2a(t+\Delta t) + 5a(t) - a(t-\Delta t)]\Delta t/6}$$

$$\underset{\text{Kinetic energy}}{U_{\text{kinetic}}} = \frac{1}{2} \sum \underset{\text{Atomic masses, velocities}}{m_i v_i(t)^2} = \frac{1}{2} \underset{\text{Boltzmann's Constant}}{n} \underset{\text{Temperature}}{k_B T}$$

Number of coordinates (not atoms)

Total energy ($U_{\text{potential}} + U_{\text{kinetic}}$) must not change with time

Time step, Δt , must be very small at 10^{-15} seconds or 0.001 ps.

Integrators

In the following sections we derive discrete-time numerical approximations to Newton's equations of motion. For simplicity, we will consider a single coordinate r but keep in mind that the following relations can be generalized to vector Cartesian coordinates \mathbf{r}^N .

The basic idea in each of the following is to solve for the trajectory of atoms as a function of time:

$$\mathbf{r}^N(t)$$

This is called a **molecular dynamics** simulation. We will actually be finding the positions at discrete time intervals,


$$\mathbf{r}^N(0), \mathbf{r}^N(\Delta t), \mathbf{r}^N(2\Delta t), \mathbf{r}^N(3\Delta t), \dots$$

Verlet algorithm

- Consider a Taylor expansion of the position vector in time:

$$\begin{aligned} r(t + \Delta t) &= r(t) + \frac{dr(t)}{dt} \Delta t + \frac{d^2 r(t)}{dt^2} \frac{\Delta t^2}{2} + \frac{d^3 r(t)}{dt^3} \frac{\Delta t^3}{6} + \mathcal{O}(\Delta t^4) \\ &= r(t) + v(t) \Delta t + \frac{f(t)}{m} \frac{\Delta t^2}{2} + \frac{d^3 r(t)}{dt^3} \frac{\Delta t^3}{6} + \mathcal{O}(\Delta t^4) \end{aligned}$$

$$r(t - \Delta t) = r(t) - v(t) \Delta t + \frac{f(t)}{m} \frac{\Delta t^2}{2} - \frac{d^3 r(t)}{dt^3} \frac{\Delta t^3}{6} + \mathcal{O}(\Delta t^4)$$


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

Verlet algorithm

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

This equation forms the basis of the **Verlet algorithm** for molecular dynamics. Here, we propagate a system forward in time by a **time step** Δt . To do so, we use the positions at the previous two time steps as well as the forces at the current time step. To get the forces, we use the force field and the current position set at time t :

$$f(t) = - \frac{dU(r(t))}{dr}$$

The accuracy of this equation is of order $\mathcal{O}(\Delta t^4)$. In other words, smaller time steps reproduce the accuracy of this discrete time approximation is.

Verlet algorithm

The Verlet algorithm does not use the velocity to determine a solution to the atomic positions at the next time step. We can approximate the velocities using:

$$v(t) = \frac{r(t + \Delta t) - r(t - \Delta t)}{2\Delta t} + \mathcal{O}(\Delta t^3)$$

Velocity Verlet algorithm

One disadvantage of the Verlet algorithm is that it requires us to store in memory two sets of positions, $r(t)$ and $r(t+\Delta t)$. An alternative is the so-called velocity Verlet algorithm, which is a reformulation of the Verlet algorithm (by manipulating the expansions) that uses the velocity directly:

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

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

1. Given $r(t)$ and $v(t)$ at one time point t , first compute the forces on each atom using the force field.

2. Do an update of the positions $r \leftarrow r + v\Delta t + \frac{f}{2m}\Delta t^2$.

3. Do a partial update of the velocity array based on the current forces:

$$v \leftarrow v + \frac{f}{2m}\Delta t$$

4. Compute the new forces $f(t + \Delta t)$ using the new positions $r(t + \Delta t)$.

5. Finish the update of the velocity array:

$$v \leftarrow v + \frac{f}{2m}\Delta t$$

6. Go back to 1.

Leapfrog Integrator

- The leap frog scheme is equivalent to the Verlet algorithm, but solves for the velocities at half time step intervals:

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

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

- One disadvantage of the leap frog approach is that the velocities are not known at the same time as the positions, making it difficult to evaluate the total energy (kinetic + potential) at any one point in time. We can get an estimate from:

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

Predictor-Corrector

Given velocities, forces and positions at time t , we determine velocities and positions at time $t + \Delta t$, where Δt is the integration tilmestep.

To illustrate the method, we choose the predictor-corrector algorithm

We take example of motion along a single coordinate $x(t)$

The Taylor expansion can read for the predicted coordinate, velocity and acceleration

$$x^p(t + \Delta t) = x(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2 + \frac{1}{6}b(t)\Delta t^3$$

$$v^p(t + \Delta t) = v(t) + a(t)\Delta t + \frac{1}{2}b(t)\Delta t^2$$

$$a^p(t + \Delta t) = a(t) + b(t)\Delta t$$

Predictor-corrector

The predictor-corrector algorithm introduces an additional correction step. Correct accelerations are calculated at $x^p(t + \Delta t)$ and the size of the error is estimated

$$\Delta a(t + \Delta t) = a^c(t + \Delta t) - a^p(t + \Delta t)$$

This error is now fed to the corrector step

$$x^c(t + \Delta t) = x^p(t + \Delta t) + c_0 \Delta a(t + \Delta t)$$

Similar equations apply to velocities and accelerations, e.g.,

$$v^c(t + \Delta t) = v^p(t + \Delta t) + c_1 \Delta a(t + \Delta t)$$

Harmonic oscillator

We will produce the equations of motion for a one-dimensional harmonic oscillator by using the finite difference predictor-corrector algorithm.

We first need to arrive at the exact solution of the problem to compare to approximate calculations.

If the oscillator has the initial position x_0 and the initial velocity v_0 , the solution is

$$x(t) = x_0 \cos \omega t + (v_0/\omega) \sin(\omega t)$$