

The Elements of Molecular Dynamics

Bill Smith

© 2023

Molecular Dynamics

- MD is the solution of the classical equations of motion for atoms and molecules to obtain the time evolution of the system.
- Applied to many-particle systems - a general analytical solution not possible. Must resort to numerical methods and computers
- Classical mechanics only - fully fledged many-particle time-dependent quantum method not yet available
- Maxwell-Boltzmann averaging process for thermodynamic properties (time averaging).

Classical Dynamics

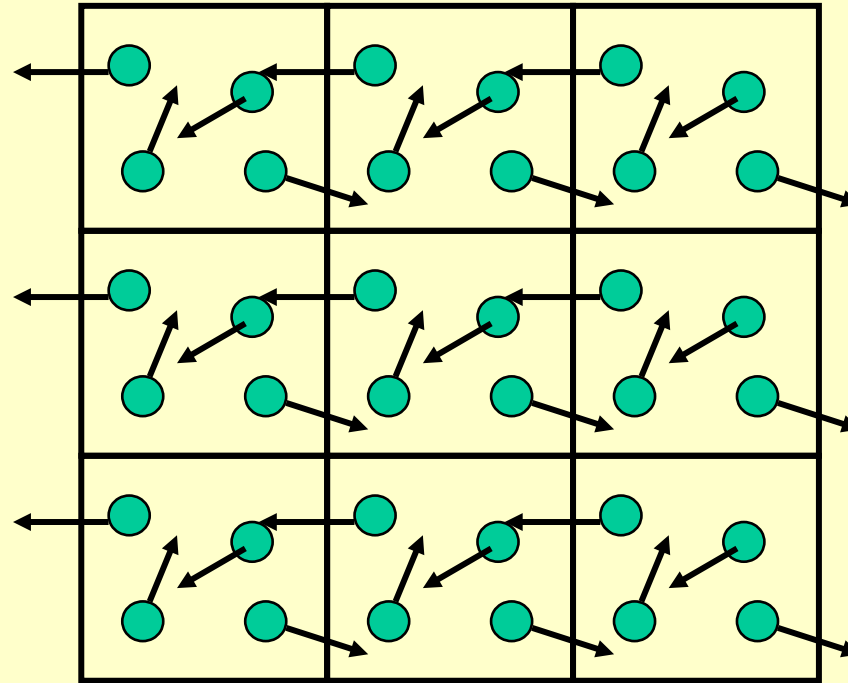
- Newton's prescription for particles:
 - If we know the position \underline{r}_i of a particle, its velocity \underline{v}_i and the force \underline{f}_i acting on it at any time t , we can compute its (translational) motion for all time.
- Euler's prescription for rigid bodies:
 - With knowledge of the centre of mass position \underline{R}_i , velocity \underline{V}_i and total force \underline{F}_i , we can compute the translational motion.
 - With knowledge of the orientation $\underline{\Omega}_i$, moment of inertia \underline{I}_i , angular velocity $\underline{\omega}_i$ and torque $\underline{\tau}_i$ of a rigid body we can compute the rotational motion.

Molecular Dynamics: Ingredients

- Initial configuration: $\{\underline{r}_i\}$
 - Crystal structure (database)
 - Amorphous structure (generate, RMC)
- Initial velocities: $\{\underline{v}_i\}$
 - Random numbers, remove COM motion, scale to T
- Interaction potentials: $\{f_i\}$
 - Potential functions (literature, database, force field)
- Boundary condition (simulation `box')
 - Cluster, periodic, hard walled?
- Integration Algorithm
 - Verlet, leapfrog or velocity Verlet

Periodic Boundary Conditions

Central cell is
simulation cell



Calculating Atomic Forces

- Forces are obtained by differentiation of the system potential energy with respect to the coordinates of the atoms.

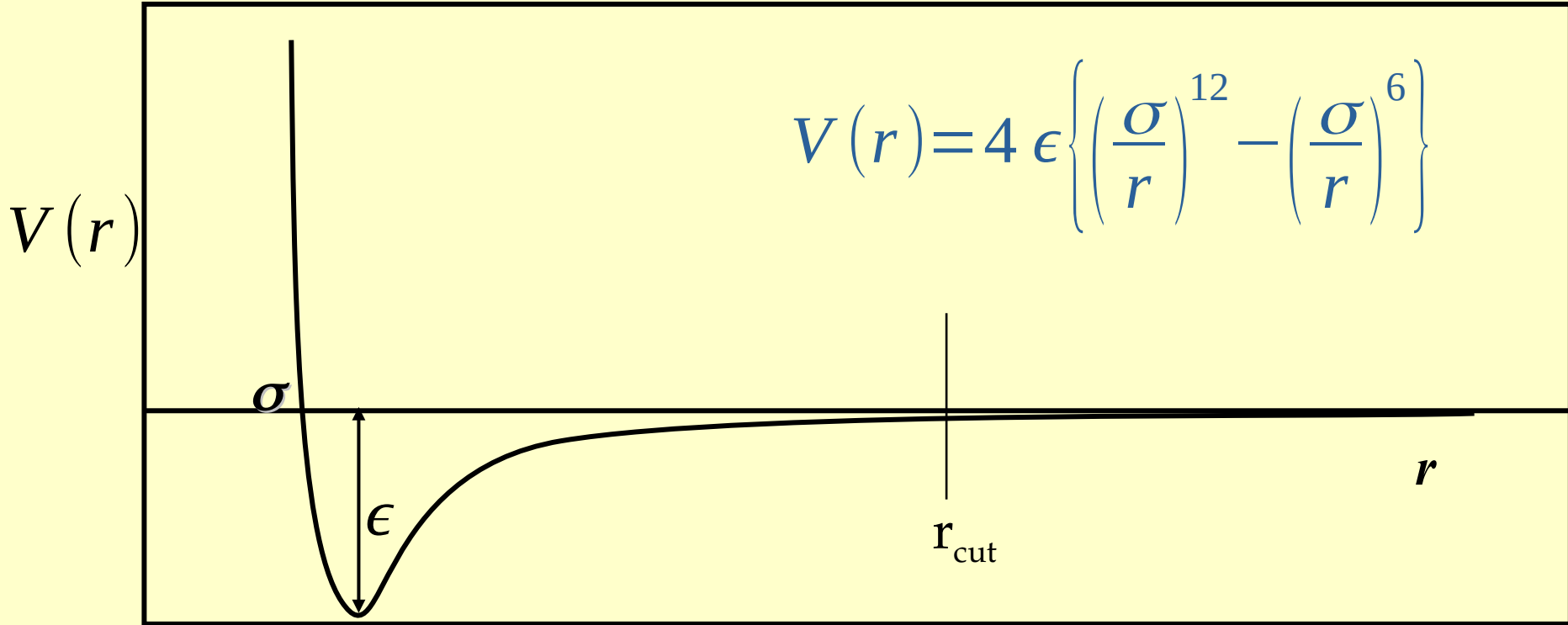
$$\vec{f}_i = -\vec{\nabla}_i U(\vec{r}^N)$$

$$\vec{r}^N = (\vec{r}_1, \vec{r}_2, \dots, \vec{r}_N)$$

$$\vec{f}_i = -\left\{ \frac{\partial}{\partial x_i} + \frac{\partial}{\partial y_i} + \frac{\partial}{\partial z_i} \right\} U(\vec{r}^N)$$

- The potential can take many mathematical forms and a cut-off is normally applied after a few Angstroms.

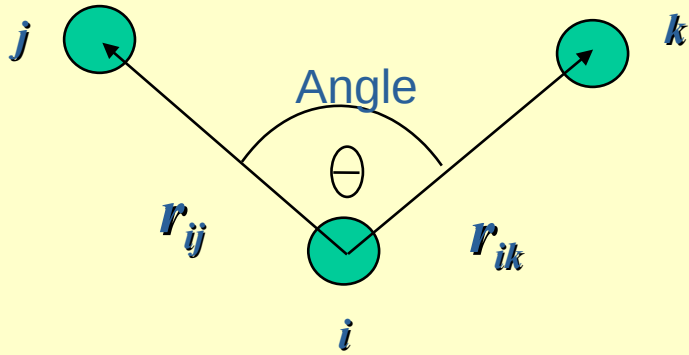
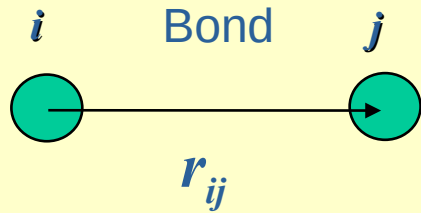
A Typical Van der Waals Potential



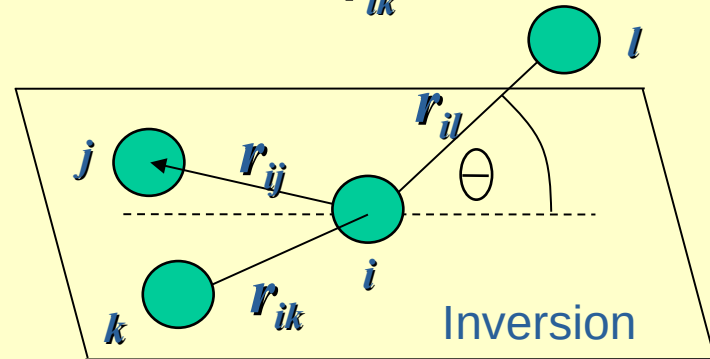
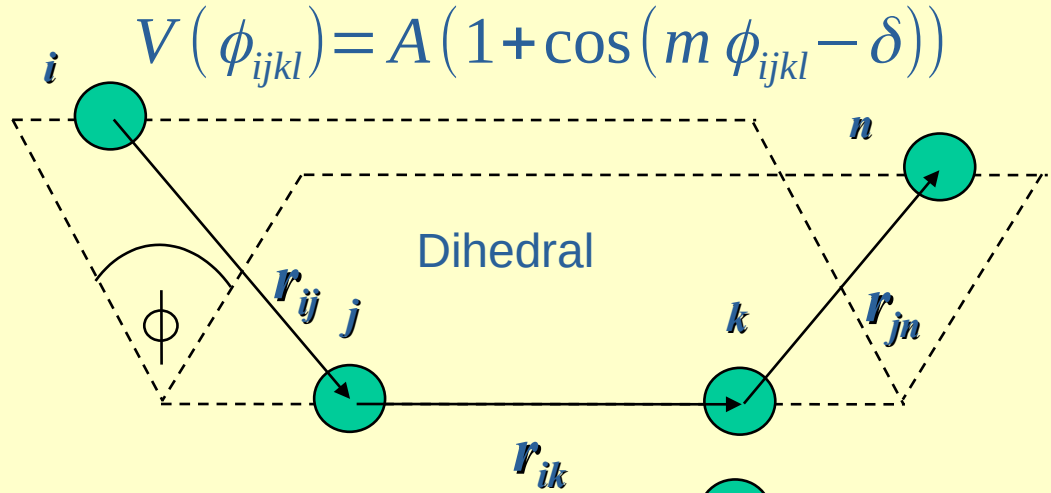
The Lennard-Jones Potential

Intra-Molecular Forces

$$V(r_{ij}) = \frac{1}{2} k (r_{ij} - r_{ij}^0)^2$$



$$V(\theta_{ijk}) = \frac{1}{2} k (\theta_{ijk} - \theta_{ijk}^0)^2$$



$$V(\theta_{ijkl}) = A(1 - \cos(\theta_{ijkl}))$$

Example: The DL_POLY Force Field

$$\begin{aligned} U(\vec{r}^N) = & \sum_{ij}^{N'} U_{ij}(r_{ij}) + \frac{1}{4\pi\epsilon_0} \sum_{ij}^{N'} \frac{q_i q_j}{r_{ij}} + \sum_{ijk} U_{3body}(\vec{r}_i, \vec{r}_j, \vec{r}_k) \\ & + \sum_{ijkl} U_{4body}(\vec{r}_i, \vec{r}_j, \vec{r}_k, \vec{r}_l) + \epsilon_{metal} \left(\sum_{ij}^{N'} \left(\frac{\alpha}{r_{ij}} \right)^n - C \sum_i^N \rho_i^{1/2} \right) \\ & + \sum_{i_{bond}}^{N_{bond}} U_{bond}(i_{bond}, \vec{r}_a, \vec{r}_b) + \sum_{i_{angle}}^{N_{angle}} U_{angle}(i_{angle}, \vec{r}_a, \vec{r}_b, \vec{r}_c) \\ & + \sum_{i_{dihed}}^{N_{dihed}} U_{dihed}(i_{dihed}, \vec{r}_a, \vec{r}_b, \vec{r}_c, \vec{r}_d) \\ & + \sum_{i_{inver}}^{N_{inver}} U_{inver}(i_{inver}, \vec{r}_a, \vec{r}_b, \vec{r}_c, \vec{r}_d) + \sum_i^N U_{external}(\vec{r}^N). \end{aligned}$$

Time Stepping

A time step is of order $10^{-15} \leq \Delta t \leq 10^{-14}$ (s). A simulation may be From 10,000 to 10,000,000 time steps. For each step:

1. Start with particle position and velocities.
2. Calculate the force on each particle.
3. Calculate new positions and velocities for one time step later.
4. Repeat steps 1-3 for required number of time steps.

The duration of the simulation (or number of time steps) will depend on the natural time scale of the phenomenon under Investigation.

Integration Algorithms

Considerations:

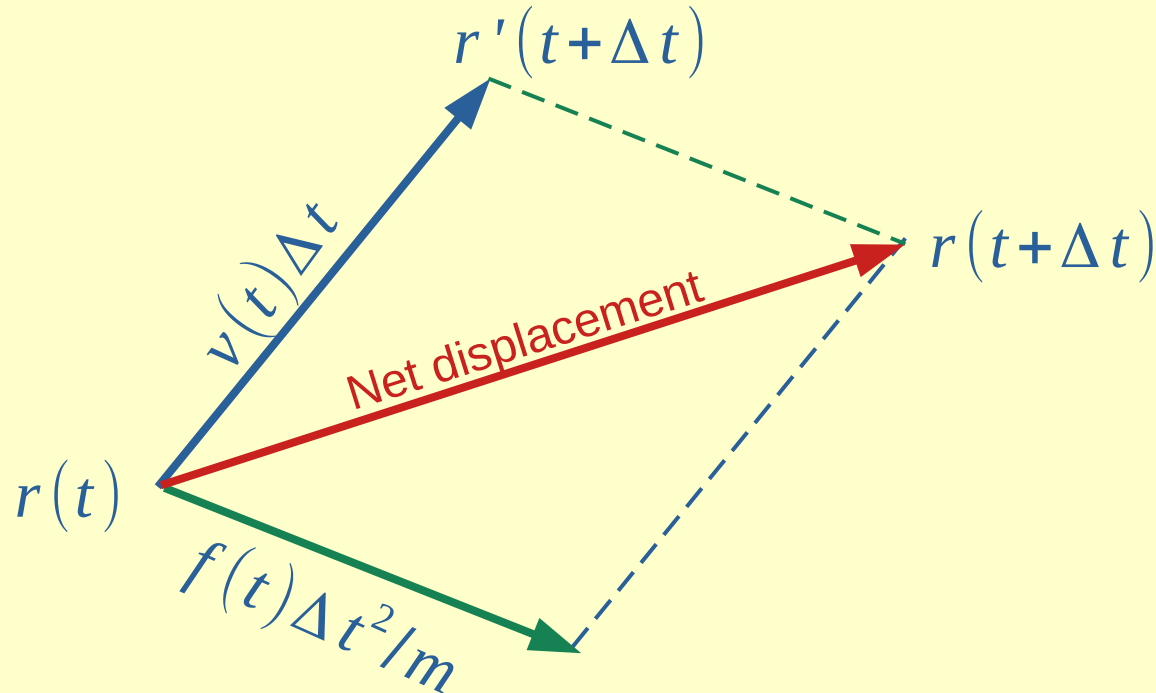
- Computational speed
- Low memory demand
- Accuracy
- Stability (energy conservation, no drift)
- Desirable - Time reversibility
- Desirable - Symplecticness

Integration: Essential Idea

$$[\vec{r}(t), \vec{v}(t), \vec{f}(t)] \longrightarrow [\vec{r}(t+\Delta t), \vec{v}(t+\Delta t), \vec{f}(t+\Delta t)]$$

Original

Updated



Verlet Algorithms

Verlet

$$\vec{r}_i^{n+1} = 2\vec{r}_i^n - \vec{r}_i^{n-1} + \frac{\Delta t^2}{m_i} \vec{f}_i^n$$

$$\vec{v}_i^n = \frac{1}{2\Delta t} (\vec{r}_i^{n+1} - \vec{r}_i^{n-1})$$

Leapfrog

$$\vec{v}_i^{n+1/2} = \vec{v}_i^{n-1/2} + \frac{\Delta t}{m_i} \vec{f}_i^n$$

$$\vec{r}_i^{n+1} = \vec{r}_i^n + \Delta t \vec{v}_i^{n+1/2}$$

Accuracy for all Verlet forms:

Position – Order $O(\Delta t^4)$

Velocity – Order $O(\Delta t^2)$

Velocity Verlet

$$\vec{v}_i^{n+1/2} = \vec{v}_i^n + \frac{\Delta t}{2m_i} \vec{f}_i^n$$

$$\vec{r}_i^{n+1} = \vec{r}_i^n + \Delta t \vec{v}_i^{n+1/2}$$

$$\vec{v}_i^{n+1} = \vec{v}_i^{n+1/2} + \frac{\Delta t}{2m_i} \vec{f}_i^{n+1}$$

System Properties

- Most properties are obtained as averages and are subject to thermodynamic fluctuation.
- Each configuration produced by MD simulation is a *sample* taken from the system *phase space*.
- Use sampled configurations to compute *average* property.
- MD is *Time Averaging* (after Maxwell and Boltzmann).
- Beware hidden time correlations!
- Structural *and* time dependent properties are available from MD.

System Properties: Static (1)

Thermodynamic Properties

Kinetic Energy:

$$\langle K \rangle = \left\langle \frac{1}{2} \sum_{i=1}^N m_i v_i^2 \right\rangle$$

Temperature:

$$T = \frac{2}{(3 N k_B)} \langle K \rangle$$

Configuration Energy:

$$U_c = \left\langle \sum_{i=1}^{N-1} \sum_{j>i}^N V(r_{ij}) \right\rangle$$

(for pair forces).

Pressure:

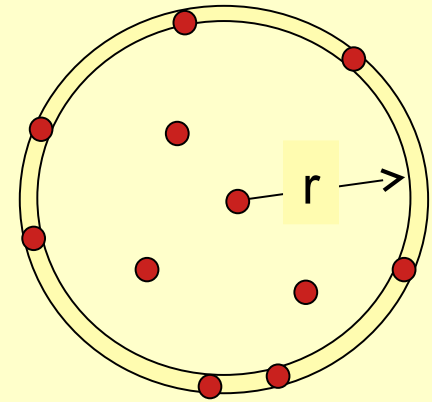
$$PV = Nk_B T - \frac{1}{3} \left\langle \sum_{i=1}^N \vec{r}_i \cdot \vec{f}_i \right\rangle$$

System Properties: Static (2)

Structural Properties

Pair correlation Function (Radial Distribution Function):

$$g(r) = \frac{\langle n(r) \rangle}{4 \pi r^2 \Delta r \rho}$$

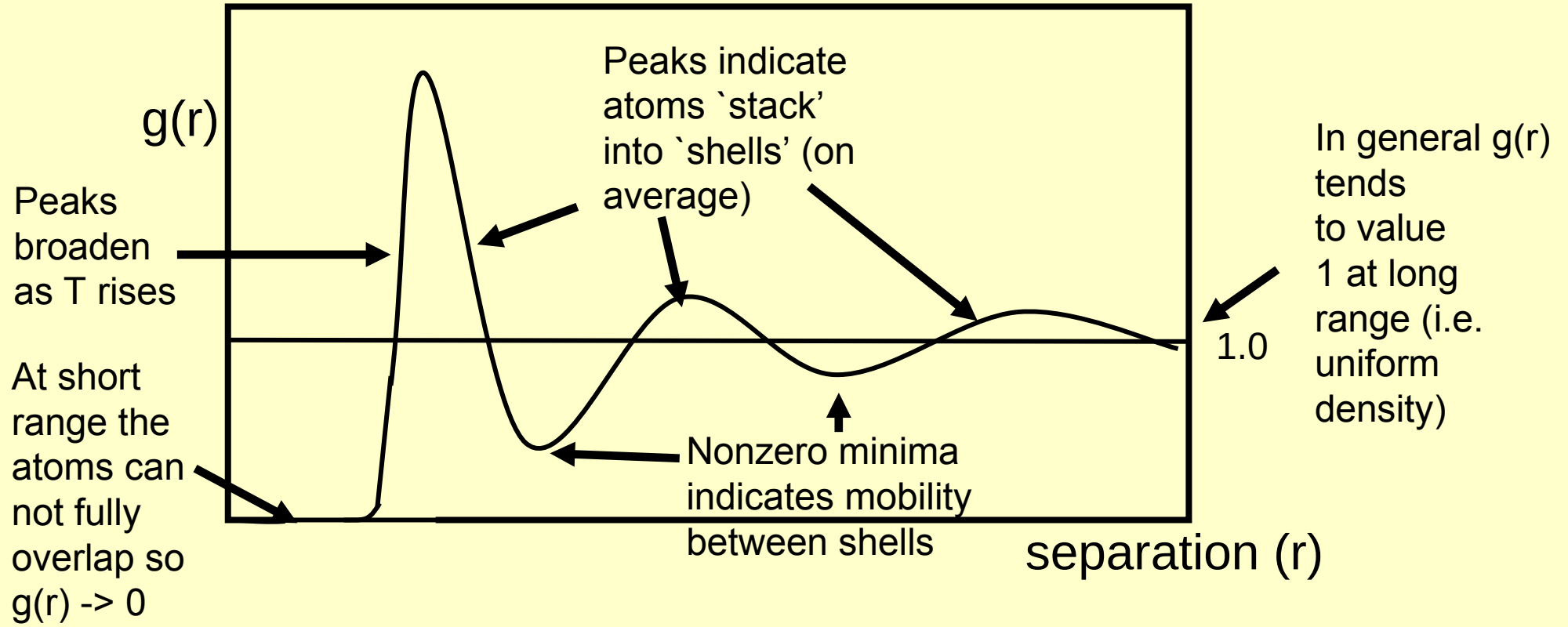


Structure Factor:

$$S(k) = 1 + 4 \pi \rho \int_0^{\infty} \frac{\sin(kr)}{kr} (g(r) - 1) r^2 dr$$

Note: $S(k)$ is available from x-ray diffraction.

Typical RDF (Liquid)



Free Energies?

All the above are calculable by molecular dynamics simulation, but **not** the free energy:

$$A(V, T) = -k_B T \log_e(Q_N(V, T))$$

where

$$Q_N(V, T) = \frac{1}{N! h^{3N}} \int \int \exp(-\beta H(\vec{r}^N, \vec{p}^N)) d\vec{r}^N d\vec{p}^N$$

is the Partition Function. However molecular dynamics can calculate a free energy **difference** using **Thermodynamic Integration**.

Thermodynamic Integration

To calculate a free energy difference we start with a **Mixed Hamiltonian**:

$$H(\lambda) = (1 - \lambda)H_1 + \lambda H_2,$$

where λ is a mixing parameter.

Next we obtain the derivative of the free energy with respect to λ :

$$\frac{\partial A(V, T)}{\partial \lambda} = \langle H_2 - H_1 \rangle_\lambda.$$

The integral of this over λ between the values 0 and 1 gives the free energy difference:

$$\Delta A_{12} = \int_0^1 \langle H_2 - H_1 \rangle_\lambda d\lambda.$$

System Properties: Dynamic

The bulk of these properties are calculated from **Correlation Functions**:

The diffusion coefficient D is obtained from the mean squared displacement using the Einstein relation:

$$2 D t = \frac{1}{3} \langle |\vec{r}_i(t) - \vec{r}_i(0)|^2 \rangle.$$

Alternatively it can be calculated from the velocity autocorrelation function:

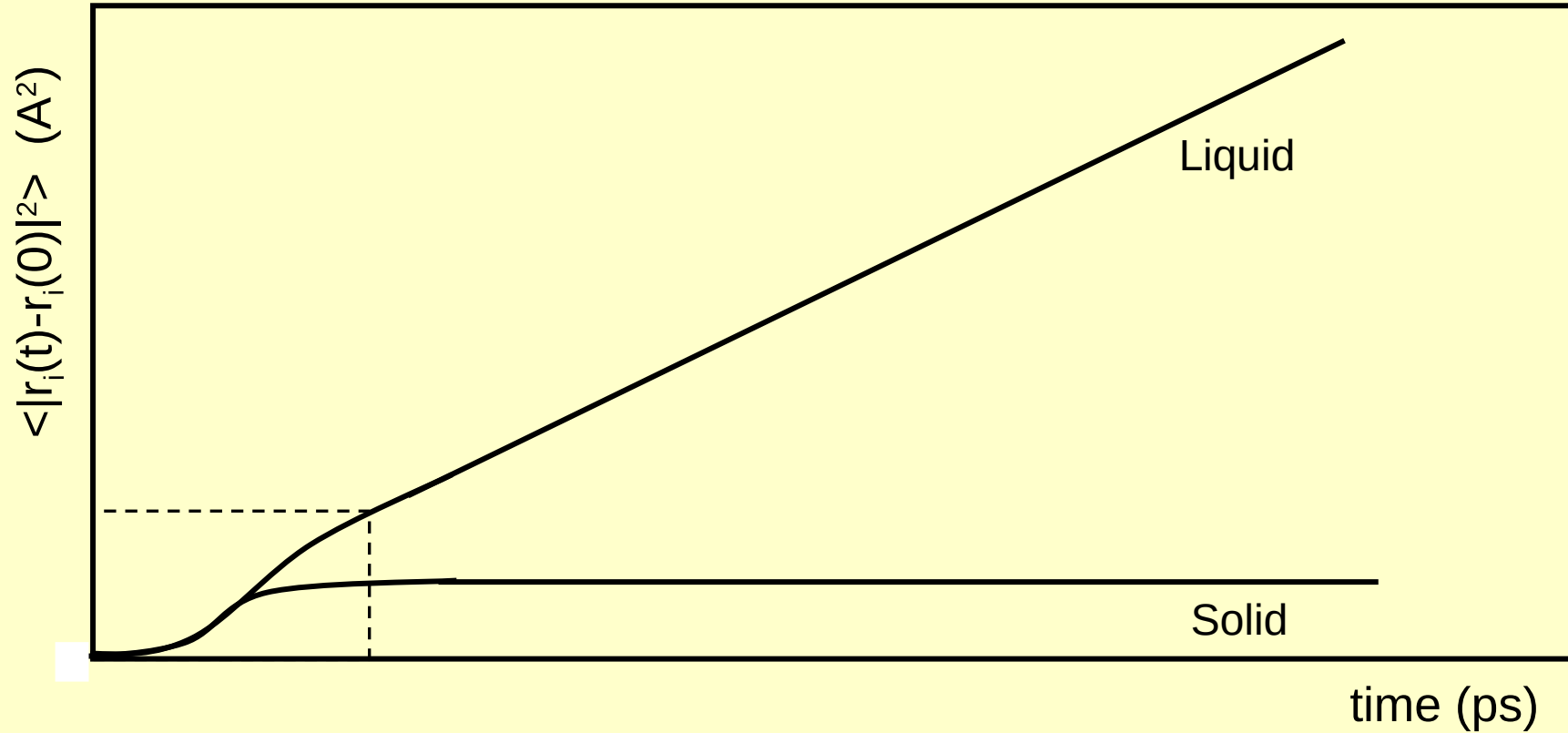
$$C(t) = \langle \vec{v}_i(t) \cdot \vec{v}_i(0) \rangle,$$

which may be integrated to give the diffusion coefficient:

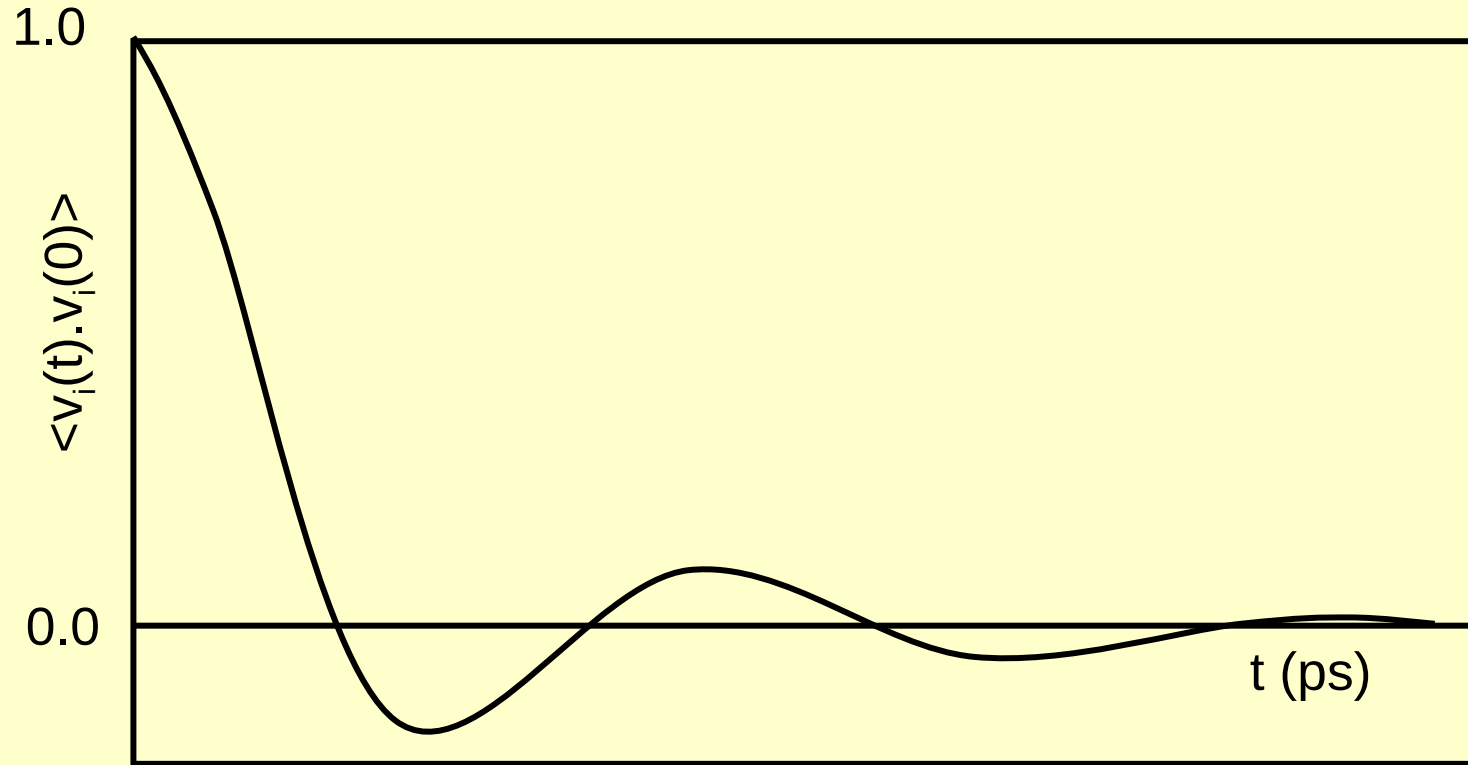
$$D = \frac{1}{3} \int_0^{\infty} \langle \vec{v}_i(t) \cdot \vec{v}_i(0) \rangle dt.$$

Such integrals are known as Green-Kubo integrals.

Solid and Liquid MSDs



A Typical VAF



Symplectic Integration Algorithms

- Important because they possess long time stability (No drift in system average properties).
- Trajectory, though approximate, remains close to the target energy surface. ('Exact' solution of a discretised Hamiltonian, which approximates true Hamiltonian.)
- Preserve volume of phase space of system.
- Time reversible forms available.

Tuckerman-Berne-Martyna Method

Liouville Equation :

$$\frac{\partial}{\partial t} f(\underline{r}^N, \underline{p}^N) = - \sum_{j=1}^{3N} \left\{ \dot{x}_j \frac{\partial}{\partial x_j} + \dot{p}_j \frac{\partial}{\partial p_j} \right\} f(\underline{r}^N, \underline{p}^N) = i\hat{\mathbf{L}}f(\underline{r}^N, \underline{p}^N)$$

Formally: $\frac{\partial}{\partial t} f = i\hat{\mathbf{L}}f \quad \therefore f = \exp(i\hat{\mathbf{L}}t)f_0$, so $\exp(i\hat{\mathbf{L}}t)$ is a *propagator*.

Now if $t = n\Delta t$ and $i\hat{\mathbf{L}} = i\hat{\mathbf{L}}_1 + i\hat{\mathbf{L}}_2$ then by Trotter's Theorem :

$$\begin{aligned} \exp(i\hat{\mathbf{L}}t) &= \left[\exp(i\hat{\mathbf{L}}\Delta t) \right]^n = \left[\exp(i(\hat{\mathbf{L}}_1 + \hat{\mathbf{L}}_2)\Delta t) \right]^n \\ &= \left[\exp(i\hat{\mathbf{L}}_2\Delta t / 2) \exp(i\hat{\mathbf{L}}_1\Delta t) \exp(i\hat{\mathbf{L}}_2\Delta t / 2) \right]^n + O(t^3 / n^2). \end{aligned}$$

Where for small Δt : $\exp(i\hat{\mathbf{L}}_j\Delta t) \approx 1 + i\hat{\mathbf{L}}_j\Delta t$. So setting

$$i\hat{\mathbf{L}}_1 = \sum_{j=1}^{3N} \left\{ \dot{x}_j \frac{\partial}{\partial x_j} \right\} \quad \text{and} \quad i\hat{\mathbf{L}}_2 = \sum_{j=1}^{3N} \left\{ \dot{p}_j \frac{\partial}{\partial p_j} \right\} \quad \text{leads to **Velocity Verlet**}$$

Extended Systems

Extended systems are required to handle different ensembles e.g.:

- The canonical ensemble (NVT)
- The isotropic isothermal-isobaric ensemble (NPT)
- The anisotropic isothermal-isobaric ensemble (N σ T)
- The Liouville equation with an extension variable χ is:

$$\frac{\partial}{\partial t} f(\vec{r}^N, \vec{p}^N, \chi) = - \left(\sum_{j=1}^{3N} \left\{ \dot{x}_j \frac{\partial}{\partial x_j} + \dot{p}_j \frac{\partial}{\partial p_j} \right\} + \dot{\chi} \frac{\partial}{\partial \chi} \right) f(\vec{r}^N, \vec{p}^N, \chi) = i \hat{L} f(\vec{r}^N, \vec{p}^N, \chi)$$

The Canonical (NVT) Ensemble

Thermostat variable: ξ

Equations of motion:

$$\begin{aligned}\dot{\vec{r}}_i &= \frac{\vec{p}_i}{m_i}, & \dot{\vec{p}}_i &= \vec{f}_i - \frac{p_\xi}{Q} \vec{p}_i, \\ \dot{\xi} &= \frac{p_\xi}{Q}, & \dot{p}_\xi &= \sum_{i=1}^N \frac{p_i^2}{m_i} - (N_f + 1) k_B T.\end{aligned}$$

Symplectic Canonical Algorithm

Liouvillian operators:

$$i\hat{L}_1 = \sum_{i=1}^N \frac{\vec{p}_i}{m_i} \frac{\partial}{\partial \vec{r}_i},$$

$$i\hat{L}_2 = \sum_{i=1}^N \vec{f}_i \frac{\partial}{\partial \vec{p}_i},$$

$$i\hat{L}_3 = -\frac{p_\xi}{Q} \sum_{i=1}^N \vec{p}_i \frac{\partial}{\partial \vec{p}_i},$$

$$i\hat{L}_4 = \frac{p_\xi}{Q} \frac{\partial}{\partial \xi},$$

$$i\hat{L}_5 = \left(\sum_{i=1}^N \frac{p_i^2}{m_i} - (N_f + 1)k_B T \right) \frac{\partial}{\partial p_\xi}.$$

Symplectic algorithm:

$$e^{i\Delta t \hat{L}} \simeq \left[e^{i\Delta t \hat{L}_5/2} e^{i\Delta t \hat{L}_4/2} e^{i\Delta t \hat{L}_3/2} \left[e^{i\Delta t \hat{L}_2/2} e^{i\Delta t \hat{L}_1} e^{i\Delta t \hat{L}_2/2} \right] e^{i\Delta t \hat{L}_3/2} e^{i\Delta t \hat{L}_4/2} e^{i\Delta t \hat{L}_5/2} \right]$$

The Isothermal-isobaric (NPT) Ensemble

Define volume scaling parameter: $\epsilon = \log(V^{1/3})$.

Equations of motion:

$$\begin{aligned}\dot{\vec{r}}_i &= \frac{\vec{p}_i}{m_i} + \frac{p_\epsilon}{W} \vec{r}_i, & \dot{\vec{p}}_i &= \vec{f}_i - \frac{p_\xi}{Q} \vec{p}_i - \left(1 + \frac{3}{N_f}\right) \frac{p_\epsilon}{W} \vec{p}_i, \\ \dot{p}_\xi &= \frac{p_\xi}{Q}, & \dot{p}_\epsilon &= \sum_{i=1}^N \frac{p_i^2}{m_i} + \frac{p_\epsilon^2}{W} - (N_f + 1) k_B T, \\ \dot{\epsilon} &= \frac{p_\epsilon}{W}, & \dot{p}_\epsilon &= 3V(P - P_0) + \frac{3}{N_f} \sum_{i=1}^N \frac{p_i^2}{m_i} - \frac{p_\xi}{Q} p_\epsilon.\end{aligned}$$

Symplectic Isothermal-isobaric Algorithm

Liouvillian operators:

$$i\hat{L}_1 = \sum_{i=1}^N \left(\frac{\vec{p}_i}{m_i} + \frac{p_\epsilon}{W} \vec{r}_i \right) \frac{\partial}{\partial \vec{r}_i}$$

$$i\hat{L}_2 = \sum_{i=1}^N \vec{f}_i \frac{\partial}{\partial \vec{p}_i},$$

$$i\hat{L}_3 = - \left(\left(1 + \frac{3}{N_f} \right) \frac{p_\epsilon}{W} + \frac{p_\xi}{Q} \right) \sum_{i=1}^N \vec{p}_i \frac{\partial}{\partial \vec{p}_i},$$

$$i\hat{L}_4 = \frac{p_\xi}{Q} \frac{\partial}{\partial \xi},$$

$$i\hat{L}_5 = \left(\sum_{i=1}^N \frac{p_i^2}{m_i} + \frac{p_\epsilon^2}{W} - (N_f + 1) k_B T \right) \frac{\partial}{\partial p_\xi}$$

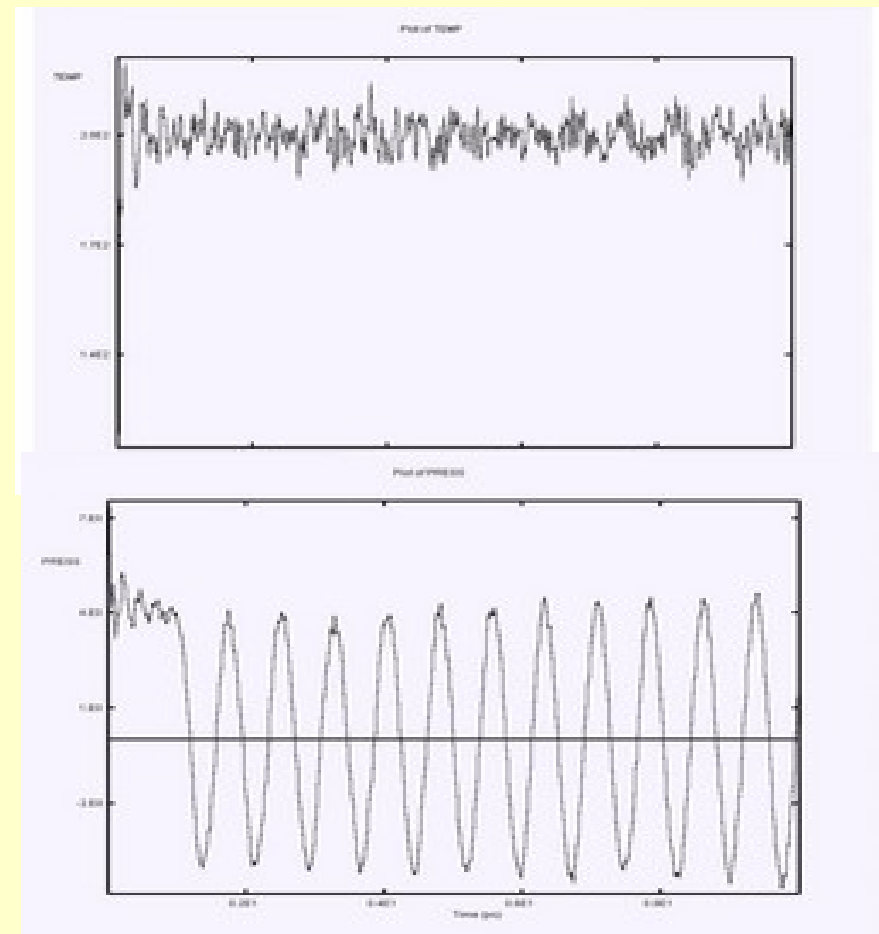
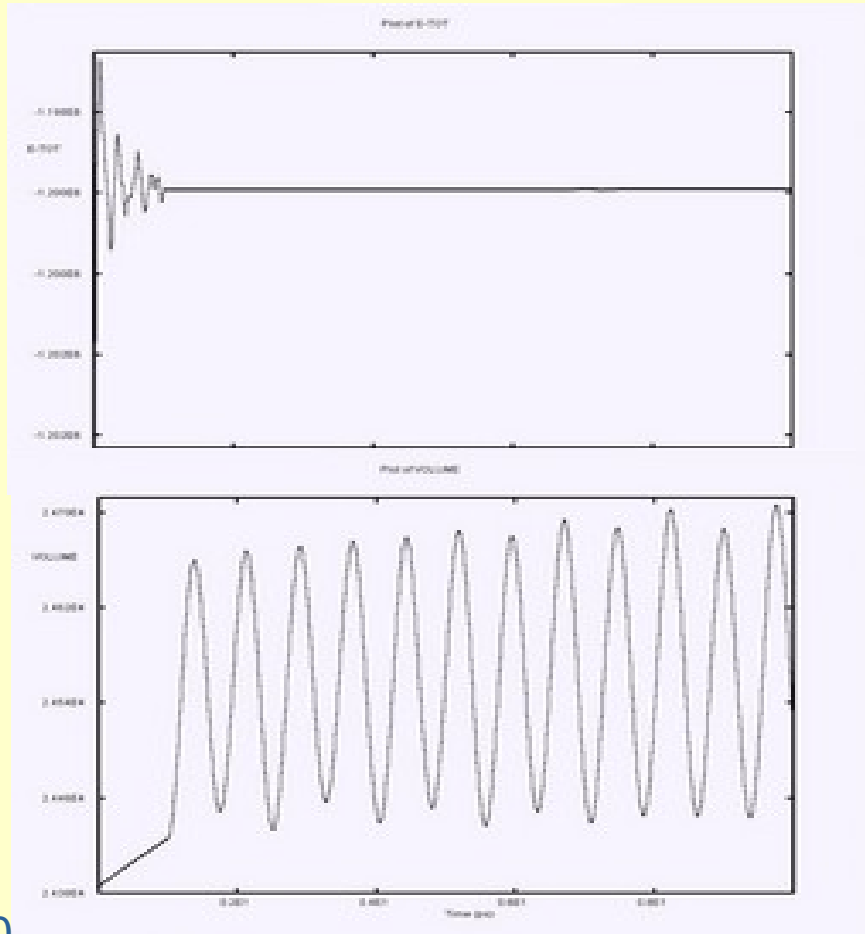
$$i\hat{L}_6 = \frac{p_\epsilon}{W} \frac{\partial}{\partial \epsilon},$$

$$i\hat{L}_7 = \left(3V(P - P_0) + \frac{3}{N_f} \sum_{i=1}^N \frac{p_i^2}{m_i} - \frac{p_\xi}{Q} p_\epsilon \right) \frac{\partial}{\partial p_\epsilon}.$$

Symplectic algorithm:

$$e^{i\Delta t \hat{L}} \simeq \left[e^{i\Delta t \hat{L}_7/2} e^{i\Delta t \hat{L}_6/2} e^{i\Delta t \hat{L}_5/2} e^{i\Delta t \hat{L}_4/2} e^{i\Delta t \hat{L}_3/2} \left[e^{i\Delta t \hat{L}_2/2} e^{i\Delta t \hat{L}_1} e^{i\Delta t \hat{L}_2/2} \right] e^{i\Delta t \hat{L}_3/2} e^{i\Delta t \hat{L}_4/2} e^{i\Delta t \hat{L}_5/2} e^{i\Delta t \hat{L}_6/2} e^{i\Delta t \hat{L}_7/2} \right]$$

Example: NPT Calcite



Long Ranged Forces

- Essential for realistic models of many systems where ions or charged atoms are present.
- Large repertoire of techniques available, but some are inadequate!
- Worst methods:
 - Anything involving a direct Coulomb sum!
- Best methods:
 - Ewald Sum
 - Smoothed Particle Mesh Ewald
 - Reaction Field
 - Multipole expansions

The Direct Coulomb Sum

$$E_{\text{coul}} = \frac{1}{4 \pi \epsilon_0} \sum_{\vec{L}=\vec{0}} \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N \frac{q_i q_j}{|\vec{L} + \vec{r}_{ij}|}$$

with: $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$.

Sum over lattice vectors \vec{L} and ion pairs (i, j) in MD cell.
Convergence is slow, conditional and possibly ambiguous.

The Ewald Sum

The safe way to sum the electrostatic energy in a periodic system.

$$U_{\text{coul}} = \frac{1}{2V \epsilon_0} \sum_{\vec{k} \neq \vec{0}}^{\infty} \frac{\exp(-k^2/4 \alpha^2)}{k^2} \sum_{j=1}^N |q_j \exp(-i \vec{k} \cdot \vec{r}_j)|^2$$

$$+ \frac{1}{4 \pi \epsilon_0} \sum_{\vec{L} = \vec{0}} \frac{1}{2} \sum_{j=1}^N \sum_{n \neq j}^N \frac{q_j q_n}{|\vec{L} + \vec{r}_{jn}|} \operatorname{erfc}(\alpha |\vec{L} + \vec{r}_{jn}|)$$

$$- \frac{\alpha}{4 \pi^{3/2} \epsilon_0} \sum_{j=1}^N q_j^2,$$

$$\text{with: } \vec{k} = \frac{2 \pi}{V^{1/3}} \begin{bmatrix} l \\ m \\ n \end{bmatrix}.$$

Reciprocal
space



Real space



Self interaction
correction



Smoothed Particle-Mesh Ewald

Ref: Essmann *et al.*, J. Chem. Phys. (1995) **103** 8577

The crucial part of the SPME method is the conversion of the reciprocal space component of the Ewald sum into a form suitable for Fast Fourier Transforms (FFT).

Thus:

$$U_{\text{recip}} = \frac{1}{2V \epsilon_0} \sum_{\vec{k} \neq \vec{0}}^{\infty} \frac{\exp(-k^2/4 \alpha^2)}{k^2} \sum_{j=1}^N |q_j \exp(-i \vec{k} \cdot \vec{r}_j)|^2,$$

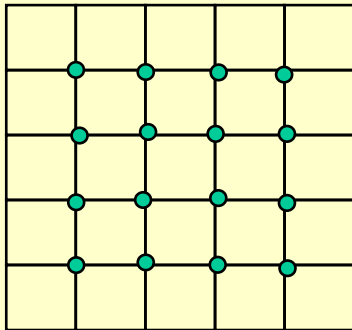
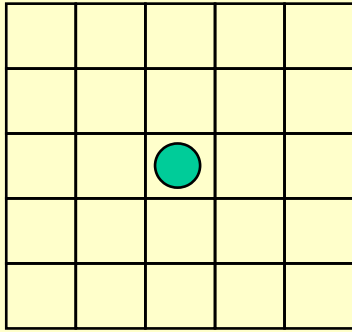
becomes

$$U_{\text{recip}} = \frac{1}{2V \epsilon_0} \sum_{k_1, k_2, k_3} G^T(k_1, k_2, k_3) Q(k_1, k_2, k_3).$$

where G and Q are 3D grid arrays (see later).

SPME: Spline Scheme

Central idea - share discrete charges on 3D grid:



Cardinal B-Splines $M_n(u)$ - in 1D:

$$\exp(2\pi i u_j k/L) \approx b(k) \sum_{l=-\infty}^{\infty} M_n(u_j - l) \exp(2\pi i k l/L)$$

$$b(k) = \exp(2\pi i(n-1)k/K) \left[\sum_{l=0}^{n-2} M_n(l+1) \exp(2\pi i k l/K) \right]^{-1}$$

$$M_n(u) = \frac{1}{(n-1)!} \sum_{k=0}^n (-1)^k \frac{n!}{k!(n-k)!} \max(u-k, 0)^{n-1}$$

$$M_n(u) = \frac{u}{(n-1)} M_{(n-1)}(u) + \frac{(n-u)}{(n-1)} M_{(n-1)}(u-1) \quad \text{Recursion relation}$$

SPME: Building the Arrays

The array:

$$Q(l_1, l_2, l_3) = \sum_{j=1}^N q_j \sum_{n_1, n_2, n_3} M_n(u_{1j} - l_1 - n_1 K_1) M_n(u_{2j} - l_2 - n_2 K_2) M_n(u_{3j} - l_3 - n_3 K_3)$$

Is the charge array and $Q^T(k_1, k_2, k_3)$ is its discrete Fourier transform.

$G^T(k_1, k_2, k_3)$ is the discrete Fourier transform of the function:

$$G(k_1, k_2, k_3) = \frac{\exp(-k^2/4\alpha^2)}{k^2} B(k_1, k_2, k_3) (Q^T(k_1, k_2, k_3))^*$$

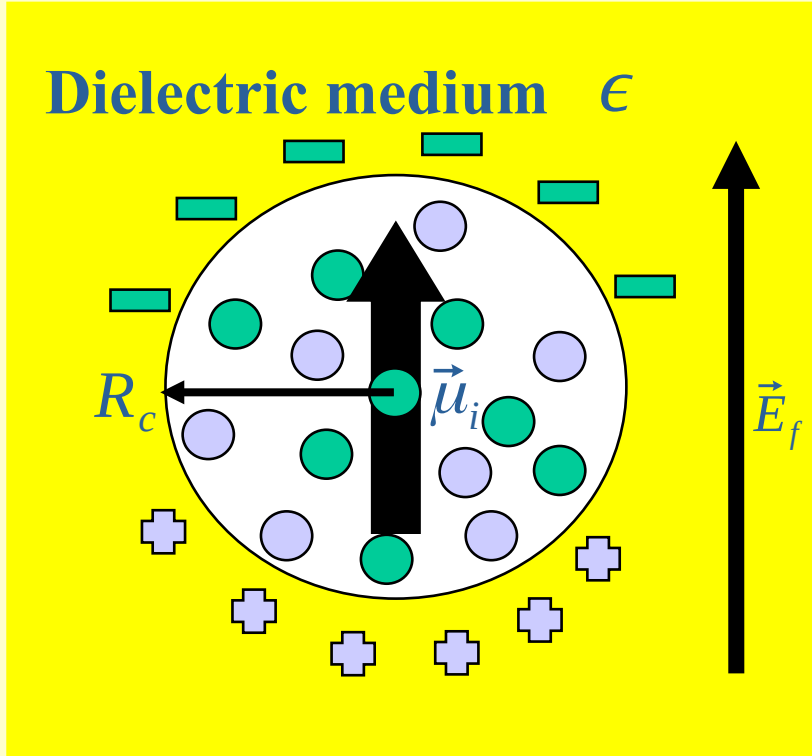
where

$$B(k_1, k_2, k_3) = |b_1(k_1)|^2 |b_2(k_2)|^2 |b_3(k_3)|^2 \dot{i}$$

SPME: Comments

- SPME is generally faster than conventional Ewald sum in most applications. Algorithm scales as $O(N \log(N))$.
- Always use a proprietary FFT (if available) for maximum performance.
- The 3D array can be large: complex arrays 256^3 are not untypical. This may be a problem on some systems.
- It is difficult to parallelise this scheme. Though parallel FFTs are available. Communications are a problem

Reaction Field Method



Physics of Reaction Field:

- Sphere of radius R_c around ion i possesses a *cavity dipole* $\vec{\mu}_i$
- Dipole $\vec{\mu}_i$ polarises the dielectric medium ϵ which induces the *Reaction Field*
- Field \vec{E}_f acts on ion i as a *long ranged force*.

Reaction Field: Formulae

Dipole:
$$\vec{\mu}_i = \sum_{j \neq i}^{R_c} q_j \vec{r}_{ij} = \sum_{j \neq i}^{R_c} q_j (\vec{r}_j - \vec{r}_i).$$

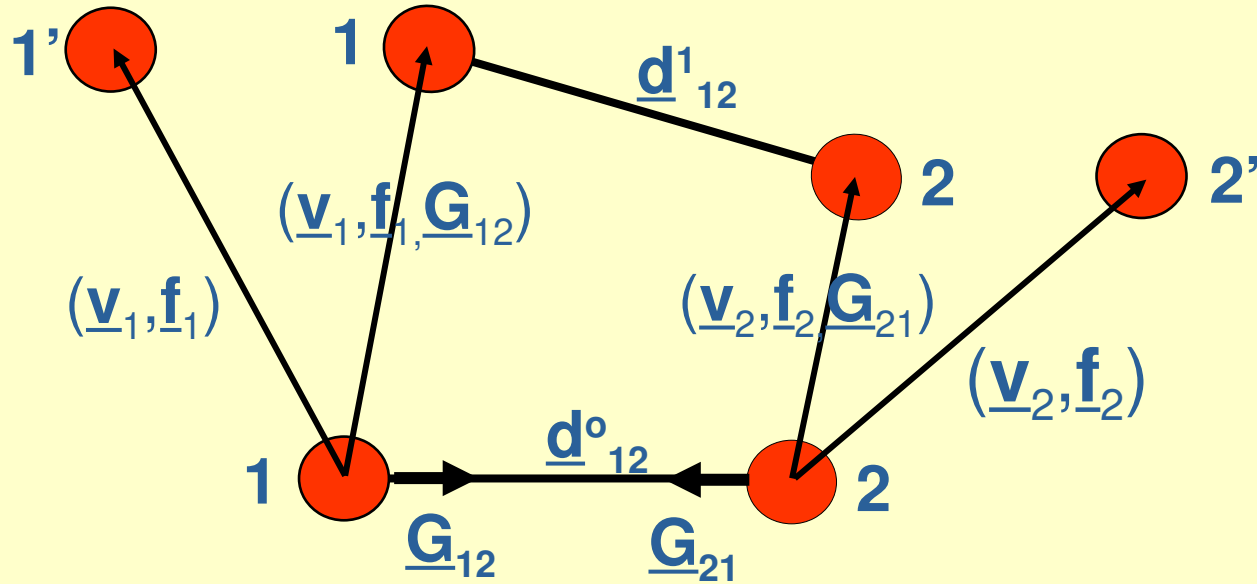
Field:
$$\vec{E}_i = \frac{B_0}{4 \pi \epsilon_0} \frac{\vec{\mu}_i}{R_c^3} \quad \text{with} \quad B_0 = \frac{2(\epsilon_s - 1)}{(2\epsilon_s + 1)}$$

Force:
$$\vec{f}_i = \vec{f}_i^C + \vec{f}_i^R,$$

and
$$\vec{f}_i^C = \frac{q_i}{4 \pi \epsilon_0} \sum_{j \neq i}^{R_c} \frac{q_j}{r_{ij}^3} \vec{r}_{ij},$$

and
$$\vec{f}_i^R = q_i \vec{E}_i = \frac{q_i}{4 \pi \epsilon_0} \frac{B_0}{R_c^3} \sum_{j \neq i}^{R_c} q_j \vec{r}_{ij}.$$

Bond Constraint Dynamics



Objective: calculate constraint force \underline{G}_{21}

Bond Constraints: SHAKE

SHAKE determination of the bond length correction:

$$\vec{r}_1 = \vec{r}'_1 + \frac{\Delta t^2}{m_1} \vec{G}_{12}, \quad \vec{r}_2 = \vec{r}'_2 + \frac{\Delta t^2}{m_2} \vec{G}_{21},$$

$$\therefore \vec{r}_1 - \vec{r}_2 = \vec{r}'_1 - \vec{r}'_2 + \Delta t^2 \vec{G}_{12} \left(\frac{1}{m_1} + \frac{1}{m_2} \right),$$

$$\therefore \vec{d}_{12} = \vec{d}'_{12} + \frac{\Delta t^2}{\mu_{12}} g_{12} \vec{d}_{12}^0, \quad \text{with} \quad \vec{G}_{12} = g_{12} \vec{d}_{12}^0,$$

$$\therefore d_{12}^2 = d_{12}'^2 + 2 \frac{\Delta t^2}{\mu_{12}} g_{12} \vec{d}_{12}^0 \cdot \vec{d}'_{12} + O(g_{12}^2),$$

$$\therefore g_{12} \approx \frac{\mu_{12} (d_{12}^2 - d_{12}'^2)}{2 \Delta t^2 \vec{d}_{12}^0 \cdot \vec{d}'_{12}}.$$

Iterate!

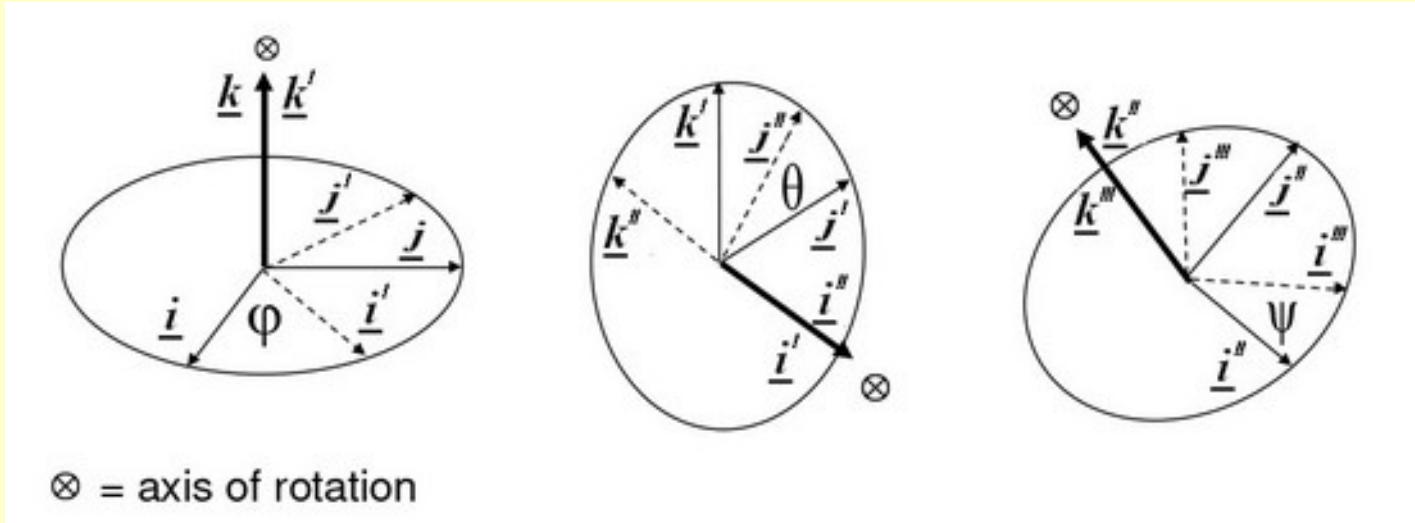
Rotational Motion

Required for treatment of rigid molecules.

- Euler's description of rigid body motion:
 - Euler angles: (ϕ, θ, ψ) .
 - Moment of inertia tensor: \mathbf{I} .
 - Laboratory and body-fixed (principal) frames.
 - Euler's equations of motion.
- Alternatively: Quaternion method.

Euler Angles (ϕ, θ, φ)

Using the x-convention (12 conventions exist!):



$$(\phi, \theta, \psi) \Rightarrow \mathbf{R} = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \cdot \begin{bmatrix} \cos \phi & \sin \phi & 0 \\ -\sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The Moment of Inertia Tensor

The moment of inertia tensor is a (3x3) matrix defined as:

$$I_{\alpha\beta} = \sum_{i=1}^{N_{\text{sites}}} m_i \left(|d_i|^2 \delta_{\alpha\beta} - d_i^\alpha d_i^\beta \right) \quad \text{with:} \quad \alpha, \beta \equiv x, y, z,$$

where $\vec{d}_i = \vec{r}_i - \vec{R}_0$ and $\vec{R}_0 = \frac{1}{M_0} \sum_{i=1}^{N_{\text{sites}}} m_i \vec{r}_i$.

This is not unique, it depends on molecular orientation.

There exists a orientation and associated rotation matrix \mathbf{R} for which \mathbf{I} is diagonal. This defines a unique orientational reference frame for the molecule called the body-fixed or **principal** frame.

Molecular Torques

Molecular torques are calculated in the laboratory frame, with respect to the molecule *centre of mass*.

$$\vec{\tau} = \sum_{j=1}^{N_{\text{sites}}} \vec{d}_j \times \vec{f}_j \quad \text{with} \quad \vec{d}_j = \vec{r}_j - \vec{R}_0,$$

and

$$\vec{R}_0 = \frac{1}{M_0} \sum_{j=1}^{N_{\text{sites}}} m_j \vec{r}_j.$$

N_{sites} Is the number of atomic sites in the *rigid* molecule, \vec{d}_j is the location of the site j with respect to the centre of mass \vec{R}_0 and \vec{f}_j is the force on site j .

Rigid Body Algorithms: Euler (1)

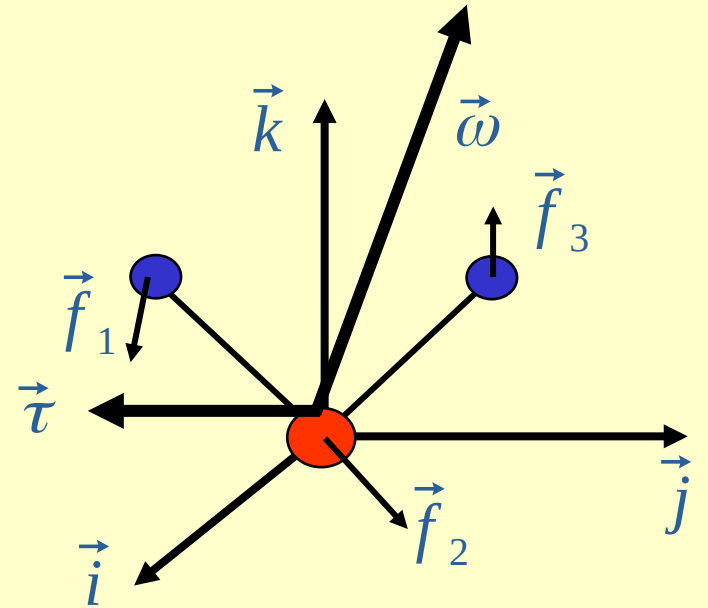
$$\text{Given: } \vec{\tau} = \sum_{j=1}^{N_{\text{sites}}} \vec{d}_j \times \vec{f}_j, \quad \vec{\tau}_p = \mathbf{R} \vec{\tau} \quad \vec{\omega}_p = \mathbf{R} \vec{\omega}$$

Euler's equations of motion (in an instantaneous frame of reference):

$$\dot{\omega}_x^p = \frac{\tau_x^p}{I_{xx}^p} + \frac{(I_{yy}^p - I_{zz}^p)}{I_{xx}^p} \omega_y^p \omega_z^p,$$

$$\dot{\omega}_y^p = \frac{\tau_y^p}{I_{yy}^p} + \frac{(I_{zz}^p - I_{xx}^p)}{I_{yy}^p} \omega_z^p \omega_x^p,$$

$$\dot{\omega}_z^p = \frac{\tau_z^p}{I_{zz}^p} + \frac{(I_{xx}^p - I_{yy}^p)}{I_{zz}^p} \omega_x^p \omega_y^p.$$



Rigid Body Algorithms: Euler (2)

Then solve the equations of motion for Euler's angles: (ϕ, θ, ψ)

$$\dot{\phi} = \frac{\sin \psi}{\sin \theta} \omega_x + \frac{\cos \psi}{\sin \theta} \omega_y$$

$$\dot{\theta} = \cos \psi \omega_x + \sin \psi \omega_y$$

$$\dot{\psi} = -\frac{\sin \psi}{\tan \theta} \omega_x - \frac{\cos \psi}{\tan \theta} \omega_y + \omega_z$$

However, these equations can become unstable because of $\sin \theta$ and $\tan \theta$ in the denominators of the first and third equations, when $\theta \rightarrow 0$.

Quaternions (1)

Quaternions provide singularity free equations of motion for molecular orientation. A quaternion \mathbf{q} is defined as a quartet of parameters derived from the Euler angles (ϕ, θ, ψ) :

$$\mathbf{q} = (q_0, q_1, q_2, q_3)^T,$$

where

$$q_0 = \cos(\theta/2) \cos((\phi + \psi)/2),$$

$$q_1 = \sin(\theta/2) \cos((\phi - \psi)/2),$$

$$q_2 = \sin(\theta/2) \sin((\phi - \psi)/2),$$

$$q_3 = \cos(\theta/2) \sin((\phi + \psi)/2),$$

and

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1.$$

Quaternions (2)

The quaternion equation of motion is:

$$\dot{\mathbf{q}} = \frac{1}{2} \mathbf{Q} \cdot (0, \omega_x^p, \omega_y^p, \omega_z^p),$$

where \mathbf{Q} is the (4x4) quaternion matrix:

$$\mathbf{Q} = \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix}.$$

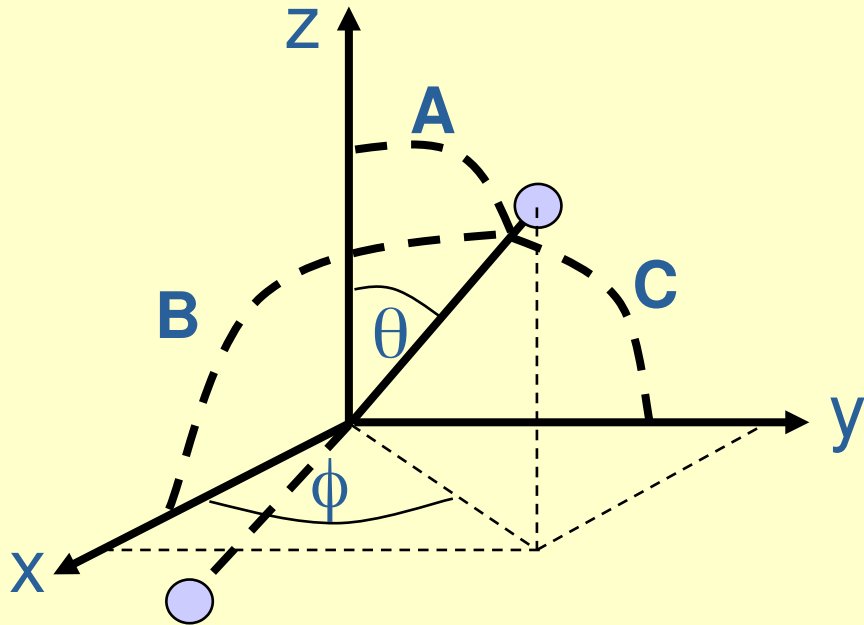
Quaternions (3)

The new rotation matrix is given by:

$$\mathbf{R} = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 - q_3^2 \end{pmatrix}.$$

The time evolution of $\vec{\omega}$ is obtained using Euler's equation.

Quaternions: A 2D Analogy



The orientation of a diatomic molecule is defined by the angles ϕ and ψ (in Figure). We can also use the direction cosines:

$$L = \cos(A) = \cos(\theta),$$

$$M = \cos(B) = \sin(\theta) \cos(\phi),$$

$$N = \cos(C) = \sin(\theta) \sin(\phi),$$

for which

$$L^2 + M^2 + N^2 = 1.$$

The analogy with quaternions is clear.

The End