

Chem 860. Lecture 5

Algorithms for MD-II: Constrained dynamics

February 5, 2009

1 Constrained dynamics

Recall the Verlet integration (propagation) scheme:

$$\mathbf{x}_i(t + \Delta t) = 2\mathbf{x}_i(t) - \mathbf{x}_i(t - \Delta t) + \frac{\mathbf{f}_i(t)}{m_i}\Delta t^2 + O(\Delta t^4)$$

Rule of thumb for Δt : 1/20th of the fastest vibration. For molecular systems, this corresponds to $\Delta t \sim 0.5$ fs.

Methods for more efficient MD calculations:

- Multiple time steps: different integration steps for different types of degrees of freedom (details later); e.g., short Δt for bond stretch, intermediate value for bending, and large value for torsion.
- Freeze high-frequency motions that are not of interest (e.g., X-H bonds), which allows to use larger time steps (~ 2 fs).

The general scheme for including constraints in studying dynamics: the Lagrangian multiplier approach. Given a set of constraints, $\{\sigma_\alpha\}$, e.g.,

$$\sigma_\alpha = r_{\alpha;ij}^2 - d_\alpha^2$$

one modifies the Lagrangian of the system,

$$\mathcal{L}' = \mathcal{L} + \sum_{\alpha} \lambda_{\alpha} \sigma_{\alpha} = \mathcal{T} - \mathcal{U} + \sum_{\alpha} \lambda_{\alpha} \sigma_{\alpha}$$

Substituting this into the Lagrangian Equation of motion,

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}'}{\partial \dot{\mathbf{x}}_i} \right) = \frac{\partial \mathcal{L}'}{\partial \mathbf{x}_i} \quad (1)$$

one obtains:

$$m_i \frac{d^2 \mathbf{x}_i(t)}{dt^2} = \mathbf{f}_i(t) + \sum_{\alpha} \lambda_{\alpha} \nabla_i \sigma_{\alpha}(t) = \mathbf{f}_i(t) + \mathbf{g}_i(t)$$

The physical meaning of this is very clear - a new set of “generalized force” (\mathbf{g}_i) is needed for maintaining the constraints during the evolution of the system.

The $\{\lambda_{\alpha}\}$ are the set of “Lagrangian multipliers” whose values need to be determined. This can be accomplished, theoretically, using the condition that the second time derivatives of the constraints are zero.

A simple example: free particle on the ring. The constraint is:

$$\sigma = \frac{1}{2}(r^2 - d^2)$$

Thus the equation of motion is (no other external force, and recall $\nabla r = \mathbf{r}/r$):

$$m \frac{d^2 \mathbf{r}(t)}{dt^2} = \lambda \nabla \sigma(t) = \lambda \mathbf{r}(t)$$

The condition that $d^2\sigma/dt^2 = 0$ gives (you can easily show that the first time derivative is $r\dot{r} = \mathbf{r} \cdot \dot{\mathbf{r}}$)

$$\mathbf{r} \cdot \frac{d^2 \mathbf{r}}{dt^2} + \left(\frac{d\mathbf{r}}{dt}\right)^2 = 0$$

Substituting this into the equation of motion (dot \mathbf{r} on both sides of equation), one obtains:

$$\lambda = -m\mathbf{v}^2/r^2$$

The generalized force is therefore

$$\mathbf{g} = \lambda \mathbf{r} = -m\mathbf{v}^2 \mathbf{r}/r^2 = -m\omega^2 \mathbf{r}$$

which is the expected centripetal force.

An **important** point is that this scheme of solving for λ does **NOT** work in realistic MD simulations because the integration schemes that we use are approximate. In fact, this scheme leads to exponential deviation from the constraint during MD simulations.

The practical approach is a predictor-corrector scheme based on the specific integration algorithm. In the presence of the generalized force, the Verlet integration scheme is in the form of,

$$\mathbf{x}_i(t+\Delta t) = 2\mathbf{x}_i(t) - \mathbf{x}_i(t-\Delta t) + \frac{\mathbf{f}_i(t)}{m_i} \Delta t^2 + \frac{\sum_{\alpha} \lambda_{\alpha}(t) \nabla_i \sigma_{\alpha}}{m_i} \Delta t^2 = \mathbf{x}_i^{uncons}(t+\Delta t) + \Delta \mathbf{x}_i^{corr}(\{\lambda_{\alpha}\})$$

We solve for $\{\lambda_{\alpha}\}$ by imposing that coordinates $\{\mathbf{x}_i(t + \Delta t)\}$ satisfy the constraints (σ_{α}) rigorously (i.e., we correct for the predicted unconstrained values $\mathbf{x}_i^{uncons}(t + \Delta t)$). In the presence of multiple constraints, the set of $\{\lambda_{\alpha}\}$ has to be solved iteratively.

2 Homework (Due. Feb. 9)

2.1 Pencil exercises: From Frenkel and Smit. Chap. 5

1. Show that a simple second-order Euler integrator does NOT satisfy time reversibility.
2. Derive the Leap-Frog Algorithm by using Taylor expansion for $\mathbf{v}(t + \frac{\Delta t}{2})$, $\mathbf{v}(t - \frac{\Delta t}{2})$, $\mathbf{x}(t + \Delta t)$ and $\mathbf{x}(t)$.
3. In explicit solvent simulations, most of the computer time is spend on dealing with water molecules (you need a lot of them to solvate a biomolecule and to set up a sufficiently large box with PBC). Therefore, it is useful to understand how to propagate their equations of motion efficiently with the SHAKE algorithm. The typical set up is to describe the internal structure of water by three bond vectors (2 OH and 1 HH). Write out the equations for propagating the trajectory using the Verlet scheme with these bonds constrained; i.e., how many Lagrangian multipliers do you need, and how do you solve for their values. Discuss briefly how can such procedures be applied to simulations that contain many water molecules. Useful discussions can be found in Allen and Tildesley, Chap. 3.

2.2 Computational exercises

We emphasized the importance of cut-off schemes in simulations. Here we explore this with a simple molecular complex that contains two NMA molecules (N-methyl acetamide, the standard model for peptide bond). Use the CHARMM script included in “hw2/” to compute the potential energy profile as a function of the distance between the two NMA molecules; explore how the results vary with different non-bond cut-off schemes (e.g., SHIFT, FSHIFT, SWITCH, FSWITCH for electrostatics, VSHIFT, VSWITCH for LJ interactions) and cut-off distances (refer to doc/nbond.doc for key word options). Also pick an atom and monitor its force components as a function of distance. Discuss your findings - a very useful reference is by B. Brooks et al. *J. Comput. Chem.* 15, 667 (1994)