Computational Biology HW4

Vince Baker

February 3, 2016

1 Problem 1

i) Following the derivation in Frankel and Smit, we calculate the potential through two partial integrations. The problem is spherically symmetric, so Φ will not depend on θ or ϕ. We first rewrite $\nabla^2\Phi(r)$ in a form with a second derivative:

$$\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right)\right)\Phi(r) = \frac{2}{r}\frac{\partial\Phi}{\partial r} + \frac{\partial^2\Phi}{\partial r^2} \tag{1.1}$$

$$\frac{1}{r}\frac{\partial^2}{\partial^2 r}(r\Phi) = \frac{2}{r}\frac{\partial\Phi}{\partial r} + \frac{\partial^2\Phi}{\partial r^2} \tag{1.2}$$

$$\nabla^2\Phi(r) = \frac{1}{r}\frac{\partial^2}{\partial^2 r}(r\Phi) \tag{1.3}$$

We now solve for Φ for a Gaussian charge distribution by two partial integrations.

$$\frac{1}{r}\frac{\partial^2}{\partial^2 r}(r\Phi) = 4\pi q_i\frac{a^3}{\epsilon_0\pi^{3/2}}e^{-a^2r^2} \tag{1.4}$$

$$\frac{\partial}{\partial r}(r\Phi) = 4\pi q_i\frac{a^3}{\epsilon_0\pi^{3/2}}\int_\infty^r dr\ re^{-a^2r^2} \tag{1.5}$$

$$\frac{\partial}{\partial r}(r\Phi) = -2q_i\frac{a}{\epsilon_0\pi^{1/2}}e^{-a^2r^2} \tag{1.6}$$

$$r\Phi = -2q_i\frac{a}{\epsilon_0\pi^{1/2}}\int_0^r dr\ e^{-a^2r^2} \tag{1.7}$$

$$\Phi = -\frac{q_i}{\epsilon_0 r}erf(ar) \tag{1.8}$$

ii) As $r \to 0$ the potential approaches the value $-\frac{2}{\epsilon_0\sqrt{\pi}}$. This makes Gaussian charge distributions (relatively) easy to handle computationally as there is no singularity as there is for a delta function charge distribution.
As $r \to \infty$ the Gaussian potential approaches 0. The error function derivative is $\propto e^{-x^2}$ so the potential rolls off faster than $\frac{1}{r^3}$, making it a short-range potential.

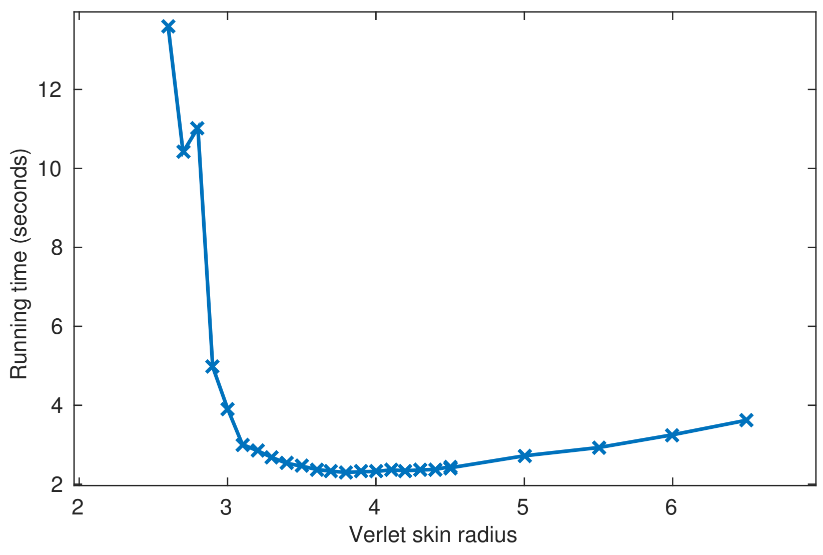
2 Computational experiment

1, 2) The full N^2 simulation took 63.64 seconds, the truncated potential with a cutoff range of $r_c = 2.5$ took 21.6 seconds. The faster execution time is expected, since the forces do not have to be calculated between every pair of molecules. This improvement is less than you might naively expect, as the cutoff method still must loop through all $N(N-1)/2$ molecule pairs and calculate minimum-image distances.

3) The computational time required for different Verlet skin radii is shown below, the value of rc was 2.5. When rl is very close to rc, the simulation is inefficient because the code will recompute the neighbor list whenever any particle moves farther than $(rl - rc)/2$. When rl is too large there are an excessive number of particles in the neighbor list, leading to increasing computational complexity.

We see that the Verlet list method is generally more efficient than the simple cutoff method. Even with a poor choice of rl the execution time is still reduced by a factor of 2. A proper choice of rl provides a full order of magnitude improvement.

Figure 1: Impact of Verlet skin radius on computational time


Running time (seconds)
Verlet skin radius