

## Topic 2: Molecular Dynamics of Lennard-Jones System (continued)

Many molecular dynamics programs use spatial decomposition, a finite range cutoff, and neighbor lists. Dr. Nakano's program pmd.c and pmd.h is a nice implementation of these important concepts. He has a writeup describing this program.

### Finite Range Force Cutoff

The evaluation of forces scales like  $\mathcal{O}(N^2)$  if *all pairs of particles* are taken into account. However, the Lennard-Jones potential energy function

$$U(r) = 4\varepsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right]$$

falls off fairly rapidly with increasing  $r$ , so the forces between particles that are very far apart may be negligible.

A reasonable approximation is to cut off the force for  $r > r_c$  where  $r_c$  is some suitably chosen cutoff radius. Unfortunately, doing this violates conservation of energy: because Newton's equations are discretized, the energy of a pair changes by a constant amount  $U(r_c)$  rather than a quantity proportional to the time step  $h$  when the separation crosses the cutoff length. This can be corrected by shifting the potential function

$$U_c(r) = \begin{cases} U(r) - U(r_c), & \text{for } r \leq r_c \\ 0, & \text{for } r > r_c \end{cases}$$

While this shifted potential is continuous at  $r = r_c$ , the force, which has strength  $-dU_c/dr$  is discontinuous at  $r = r_c$ . This discontinuity can cause instabilities in the integration algorithm. A simple remedy for this problem is to modify the potential further

$$U_c(r) = \begin{cases} U(r) - U(r_c) - \left. \frac{dU}{dr} \right|_{r=r_c} (r - r_c), & \text{for } r \leq r_c \\ 0, & \text{for } r > r_c \end{cases}$$

so that

$$F_c(r) = -\frac{dU_c}{dr} = \begin{cases} F(r) - F(r_c), & \text{for } r \leq r_c \\ 0, & \text{for } r > r_c \end{cases}$$

is continuous at  $r = r_c$ .

Since the potential and force have now been changed, observables such as the pressure must also be corrected. For example, it can be shown that

$$\langle U \rangle = \langle U_c \rangle + \frac{4\pi N}{V} \int_{r_c}^{\infty} r^2 dr U(r) g(r),$$

where  $g(r)$  is the radial distribution function, which can be approximated in the integral by  $g(r) = 1$ . Similarly, the virial formula for the pressure can be corrected using

$$PV = Nk_B T - \frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{r}_i \cdot \mathbf{F}_{ci} \right\rangle - \frac{N}{3V} \int_{r_c}^{\infty} r^3 dr \frac{dU(r)}{dr} g(r).$$

Note that the averages in these formulas involve pairs with  $r \leq r_c$  and calculating the averages scales like  $\mathcal{O}(N)$ .

## Neighbor Lists and Cell Structures

If only pairs with a separation  $r \leq r_c$  are taken into account, the total number of pairs scales like  $\mathcal{O}(N)$ . However there is still a problem! How do we decide *which* pairs have separation  $r \leq r_c$  without actually measuring  $r$  for *all* pairs? Two useful ways of solving this problem are:

**Verlet Neighbor List:** A list of all particle pairs with separation  $r_{\max} > r_c$  is maintained and updated every say 10 or 20 time steps.  $r_{\max}$  is chosen large enough that it is unlikely that a particle pair not in the list will come closer than  $r_c$  before the list is updated.

It is also possible to decide automatically when the neighbor list needs to be updated. When the list is created, a vector for each particle is set to zero. At each time step, the vector is incremented by the particle displacement. The list is updated when the sum of the magnitudes of the two largest vectors exceeds  $r_{\max} - r_c$ .

The neighbor list method becomes inefficient for very large  $N$  because each update requires  $\mathcal{O}(N^2)$  tests.

**Cell Structures and Linked Lists:** The system volume  $V$  is divided into cubic cells of side  $\ell > r_c$ . At each time step, the particles in each cube are stored in a *linked list*. Particles in a given cell interact only with particles in the 26 neighboring cells. For given  $r_c$ , the number of such pairs scales like  $\mathcal{O}(N)$ , and so does the construction of the linked lists.