Computational Physics Project 2: N-Body Physics

I. PROBLEM

The objective of this task is to solve the Newton's laws

$$m\ddot{\mathbf{x}}_i = -\sum_{i \neq j} \nabla V(\mathbf{x}_i, \mathbf{x}_j)$$

for N particles with positions $\mathbf{x}_i$. $V(\mathbf{x}_i, \mathbf{x}_j)$ is a pairwise potential function: Examples include electrostatic effects,

$$V(\mathbf{x}_i, \mathbf{x}_j) = \frac{q_i q_j}{4\pi\epsilon_0 \left|\mathbf{x}_i - \mathbf{x}_j\right|},$$

but also springs,

$$V(\mathbf{x}_i, \mathbf{x}_j) = \frac{K}{2} T_{ij} \left(\left|\mathbf{x}_i - \mathbf{x}_j\right| - L_{ij}\right)^2,$$

where $T_{ij} = 1$ if the particles are connected, or zero if not. Compact potentials could also be of interest whereby

$$V(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} f(\mathbf{x}_i, \mathbf{x}_j) & \left|\mathbf{x}_i - \mathbf{x}_j\right| < \sigma_{ij} \\ 0 & otherwise \end{cases}$$

The goal of this project is to create a code that can handle any of these scenarios, and apply them to some problems of interest.

II. DESIGN CONSIDERATIONS

1. Choice of integrator. SciPy provides scipy.integrate.odeint, a general purpose integrator that can handle many variables.

2. Program design. A modular design will enable you to tackle more problems and more interesting problems.

3. Data structures for efficient calculation of forces. You may want to think about using sparse matrices, kd-trees or quadtrees for distance exclusion etc.

4. Applications and goals for modelling. This kind of program has been used for simulating galaxy formation, elastic bodies including cellular mechanics; it's also the basis of Molecular Dynamics and other techniques. Discuss with the instructor ideas for things to model.