Class 19. *N*-body Techniques, Part 2

Time-integration Schemes

- Clearly, Newton's laws are IVP. Could use any method (Euler, RK4, etc.).
- But, issue is to balance accuracy *vs.* efficiency.
- Typically need many particles to capture dynamics correctly (e.g., in stellar system or galaxy). This consideration may be as important as accuracy of any one individual particle (exception: solar system—$N \sim 9$, $\tau \sim 10^9$–10^{10} orbits).
- Could use Euler scheme. But we have seen it is just as easy to design 2^{nd}-order scheme by centering derivatives $\Longrightarrow$ could use leapfrog (very stable).

Practical timestep control

- The stability criterion from the discussion of stiff systems also applies to the leapfrog integrator for the N-body problem.
- Can show need $\delta t < 2/\Omega$, where $\Omega^2 = |\nabla \boldsymbol{\mathcal{F}}|$ is a characteristic "interaction frequency" for a particle (in practice, need $\delta t \ll 2/\Omega$ to avoid problems).
- But Ω^2 is different for every particle; can be very large for particle undergoing close interaction.
- If have to take $\Omega_{\max}$, can be very restrictive. Two solutions:
  1. Use different timesteps for each particle (individual timesteps).
     E.g., $\delta t_i = \eta F_i / \dot{F}_i$ — effective, but complex implementation, and may break symplecticity of leapfrog (for example).
     - More complex expressions for δt_i can be formulated, e.g., involving higher-order derivatives of F. These are largely heuristics with convenient properties. It is difficult to prove analytically that one formulation is superior to another.
     - Sometimes δt_i is discretized, e.g., in factors of 2 (multistepping).
  2. Eliminate short-timescale phenomena by modifying gravity on small scales.
     E.g., Set $\delta t = \tau_D/30$ and/or use softening (see below).
- Always important to check whether simulation is giving physically meaningful results.
  - Handy technique: reduce timestep by factor of 2 to see if global behavior strongly affected. If so, may have to use smaller steps.
  - Beware of *chaos*: if state of system strongly dependent on initial conditions, change of timestep may give seemingly vastly different results. Need to monitor constants of motion to be sure.

Force evaluation

- Solving the IVP requires evaluation of the RHS of the ODEs, i.e., must compute interparticle forces.
- Will discuss PP, PM, P^3M, and tree methods.
- But first must consider another practical issue, related to timestep control...

Hard interactions

- Recall $\boldsymbol{\mathcal{F}}_{ij} = -Gm_j(\mathbf{r}_i - \mathbf{r}_j)/|\mathbf{r}_i - \mathbf{r}_j|^3$.
- Problem: if $|\mathbf{r}_i - \mathbf{r}_j|$ is small, $|\boldsymbol{\mathcal{F}}_{ij}|$ diverges, leading to timestep trouble as $|\mathbf{v}_i| \to \infty$.
- Physically, very close encounters occur on very short timescales, e.g., can form close binaries with very short periods.
- To alleviate problem, could use "softened" forces:

$$\boldsymbol{\mathcal{F}}_{ij} = -\frac{Gm_j(\mathbf{r}_i - \mathbf{r}_j)}{(|\mathbf{r}_i - \mathbf{r}_j|^2 + \varepsilon^2)^{3/2}},$$

  where ε = "softening parameter."

  - Maximum force now $\sim Gm^2/\varepsilon^2$.
  - Physically, this eliminates possibility of forming binaries with $r < \varepsilon$.
    - OK when particles represent collection of stars on similar orbits.
    - Not OK if studying small clusters, where each particle represents an individual star. In this case binaries *can* form and significantly affect evolution of entire cluster.
- Modern methods also sometimes use "regularization."
  - Binaries (or hierarchies) replaced by pseudo-particles until interaction with other particles becomes important.

Direct Summation (PP Method)

- Most straightforward way of evaluating $\mathbf{F}_{ij}$.
- But number of operations $= \frac{1}{2}N(N-1) \sim N^2$ for $N \gg 1$ (the $\frac{1}{2}$ comes from the fact that $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$.

  $\therefore 10\times$ more particles $\Longrightarrow 100\times$ more work.
- Severely limits number of particles that can be used (typically $\lesssim 10^{3\text{–}4}$).
- Motivates finding more efficient techniques.

Evaluating Forces on a Mesh (PM Method)

- Idea: compute forces (per unit mass) from gravitational potential:

$$\boldsymbol{\mathcal{F}} = -\nabla\Phi, \tag{1}$$

  where Φ = potential, a scalar function of $\mathbf{r}$.

- Potential given by solution of Poisson's equation:

$$\nabla^2\Phi = 4\pi G\rho. \tag{2}$$

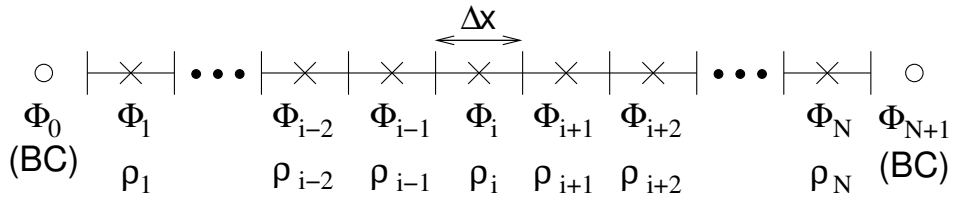
- Can FD (2) to compute Φ on a mesh, then FD (1) to compute $\boldsymbol{\mathcal{F}}$ from Φ.

- In 1-D, Poisson equation is:

$$\frac{\partial^2\Phi}{\partial x^2} = 4\pi G\rho.$$

  – This is an elliptic PDE.

- Consider discretizing Φ on a mesh, center Φ_i and ρ_i, at mesh centers $i = 1, ..., N$.



  Then, 2nd-order FDE for (2) is:

$$\frac{\frac{(\Phi_{i+1}-\Phi_i)}{\Delta x} - \frac{(\Phi_i-\Phi_{i-1})}{\Delta x}}{\Delta x} = \frac{\Phi_{i-1} - 2\Phi_i + \Phi_{i+1}}{(\Delta x)^2} = 4\pi G\rho_i \ (i = 1, ..., N). \tag{3}$$

$$\left.\begin{array}{rl} \text{For} \quad i = 1, & \text{need } \Phi_0 \\ i = N, & \text{need } \Phi_{N+1} \end{array}\right\} \text{(boundary conditions).}$$

- For the force, just FD (1) using the same mesh:

$$\mathcal{F}_{i+1/2} = -\left[\frac{\Phi_{i+1} - \Phi_i}{\Delta x}\right] \tag{4}$$

  (need to interpolate to cell centers).