Astrophysics 1 - N-body calculations

The N-body problem is a very classical computational problem with applications in many fields of astrophysics: bodies in planetary systems, stellar clusters, galaxies, clusters of galaxies, ... In this short computational hands-on problem, we use the N-body approach to study a few basis aspects of a self-gravitating system. First, we program the initial state of the system. Then, an integrator will be implemented to compute its dynamical evolution. In the third part, we will examine the variations of a few physical quantities to assess the quality of the calculation. Finally, the forth part will introduce the virial theorem as a tool to evaluate the gravitational boundedness of the system, and we will mimick an observational test of the boundedness of a system.

You are asked to keep your code in a good shape and available for next year. This problem will be used as a test case to learn high-performance computing starting from today's code. For this reason, it is requested that you work with the Fortran language.

1 Definition of the problem, the variables and the units

The system is composed of N objects (hereafter referred to as "bodies" or "particles") of equal mass m, totaling a mass $M_{\text{tot}} = Nm$. They are in the vacuum, devoid of any external forces, and they interact only through the gravitational force. We describe their time evolution in the frame of Newton's laws in a Galilean frame. Therefore, for the i-th body:

$$m\frac{d^2\vec{X}_i}{dt^2} = \sum_{j\neq i} \vec{F}_{j\to i}(t) = \sum_{j\neq i} \frac{Gm^2}{|\vec{X}_i - \vec{X}_j|^3}(\vec{X}_j - \vec{X}_i) \tag{1}$$

where $\vec{X}_i$ is the position vector of particle i, G is the universal gravitational constant, and $\vec{F}_{j\to i}(t)$ is the gravitational force from particle j onto particle i. Note that the order of the two position subtractions matters, as it defines the orientation of the forces.

To limit the risk of encountering the limits to the range of numerically represented floats and optimize the accuracy of the calculations, it is customary to choose a dimensionless system of units:

- $\tilde{X}_i = X_i/R$, where R is the initial size of the N-body system (and where we omitted the vector arrow for notation clarity),
- $\tilde{t} = t/t_0$, where t_0 is a timescale that needs to be determined.

To determine the timescale t_0, we replace X_i and t by their expressions as a function of $\tilde{X}_i$, R, $\tilde{t}$, and t_0 in eq. 1:

$$\frac{d^2\tilde{X}_i}{d\tilde{t}^2} = \sum_{j\neq i} \frac{1}{N|\tilde{X}_i - \tilde{X}_j|^3}(\tilde{X}_j - \tilde{X}_i) \times \frac{GMt_0^2}{R^3} \tag{2}$$

We choose to define t_0 so as to have the fraction GMt_0^2/R^3 equal to unity:

$$t_0 = \sqrt{\frac{R^3}{GM}} \tag{3}$$

Interestingly, this expression is very close to the well-known free-fall time ($t_{\text{ff}} = \sqrt{3\pi/(32G\rho)}$ where ρ is the mass volume density), which is indeed a meaningful timescale for this problem.

This whole procedure is sometimes summarized as "choosing $G = 1$". However this phrasing makes it difficult to link the physical units to the dimensionless units. Here, we propose to consider that we study the formation of a galaxy, hence the following values: $R = 50\,\mathrm{kpc}$ and $M = 10^{11}\,\mathrm{M}_\odot$. **Compute the corresponding (approximate) free-fall time** t_0. In the following we omit the tildes of the dimensionless quantities to lighten the notation.

2 Initial state of the system

At time $t = 0$, the bodies are distributed randomly and uniformly within the volume of a sphere of radius $R = 1$ in the aforementioned dimensionless unit system. The whole system is rotating as a solid body about the axis z, with an angular velocity Ω.

1. Create a Fortran source code where you define dynamically[1] two arrays, `pos` and `vel`, each of size $N \times 3$, representing respectively the position and velocity of all bodies.

2. Set a random seed. You can get help from https://fortranwiki.org/fortran/show/random_seed or another online resource. In normal usage, the seed should be set randomly. For debug purposes, it is useful to fix the seed to a given value so as to repeat exactly the same pseudo-random sequence of numbers when calling `random_number`.

3. Randomly set the position of N bodies within the volume of a sphere with radius R, centered on the origin of positions. A simple way to implement this is to randomly draw the x, y, z coordinates of points between -1 and 1, compute the distance of the current point to origin and keep the point if the distance is lesser than R.

4. Set the velocities of the bodies according to a solid rotation of the whole system at $t = 0$. We remind that in a solid rotation, each point rotates about the rotation axis, here the z-axis, along a circle of radius $r = \sqrt{x^2 + y^2}$, with a velocity proportional to r and Ω.

3 Integration of the trajectories

We will use here the leap-frog integrator. This choice is motivated by its good ratio between performance and simplicity. It is simple as it is programmed almost like the Euler integrator, with the difference that the position and velocity instants are shifted by half a time step. This simple trick makes it a second order and simplectic integrator. The scheme is built as follows:

$$m\frac{d\vec{v}_i}{dt}(t) = \sum_{j\neq i} \vec{F}_{j\to i}(t) \quad \Rightarrow \quad \vec{v}_i(t + dt/2) \approx \vec{v}_i(t - dt/2) + \frac{1}{m}\sum_{j\neq i} \vec{F}_{j\to i}(t) \cdot dt \tag{4}$$

$$\frac{d\vec{X}_i}{dt} = \vec{v}_i \quad \Rightarrow \quad \vec{X}_i(t + dt) \approx \vec{X}_i(t) + \vec{v}_i(t + dt/2) \cdot dt \tag{5}$$

where i is the current body, j designs the other bodies, and the shift by half a time step is intentional. The forces $\vec{F}_{j\to i}$ (and therefore the accelerations) are computed at the same instant as the positions. In the present problem we only consider the gravitational interaction between the pairs of bodies. According to Sect. 1, the force felt by body i due to its interaction with body j is expressed as:

$$\vec{F}_{j\to i} = \frac{1}{N|\vec{X}_i - \vec{X}_j|^3}(\vec{X}_j - \vec{X}_i) \tag{6}$$

1. Develop a subroutine that takes in input the positions of two bodies and returns the three components of the force.

2. Implement the leap-frog within a loop over time. Write the position and velocity of bodies in a file every few time steps.

[1] An array is said *dynamically*-defined when it is defined as "allocatable" in the variable declaration part of the code. When the dimensions are given directly in the variable declaration part, the array is said to be declared *statically*.

3. In a copy of your program, test your implementation in the case of a two-body system in circular orbits. Plot the trajectories, measure the period of the movement, and compare with the analytical calculation.

4. Once the two-body system works satisfyingly, run a larger simulation with $N = 100$ bodies. Plot a few trajectories. What do you observe?

5. Reduce the time step by a factor of 10. Does it change the previous observation ?

6. A classical and important technique consists in regularizing the gravitational potential:

$$\Phi \propto \frac{1}{\sqrt{r^2 + \epsilon^2}}$$

Change the force accordingly in your code. Run simulations for various values of ϵ, and choose a reasonable value for it.

4 Conservation of physical quantities

A good quality test of N-body simulations consists in examining the variations in physical quantities that are theoretically conserved. No dissipation and no interactions with external systems are included in our model, so that the energy and total angular momentum should be conserved.

1. Compute the total potential energy of the system at the current time step.

2. Compute the total kinetic energy of the system at the current time step. Do not forget to take into account the half time-step shift between velocity and position.

3. Plot the total mechanical energy as a function of time for several values of dt, but for the same total time range. Comments?

4. Compute the total angular momentum. Plot it as a function of time for several values of dt, but for the same total time range. Comments?

5 The Virial theorem

The so-called Virial theorem addresses the stability of a self-gravitating ensemble of particles based on considerations on the angular momentum. We follow here the derivation exposed by Ward-Thompson & Whitworth[2]. We consider the total angular moment of the ensemble of bodies:

$$\mathcal{I} = \sum_i m_i \vec{X}_i \cdot \vec{X}_i$$

At equilibrium, the angular moment is constant:

$$\dot{\mathcal{I}} = 2\sum_i m_i \vec{v}_i \cdot \vec{X}_i = 0 \quad \text{and} \quad \ddot{\mathcal{I}} = 2\sum_i m_i \dot{\vec{v}}_i \cdot \vec{X}_i + 2\sum_i m_i \vec{v}_i \cdot \vec{v}_i = 0$$

Denoting $\ddot{\mathcal{I}}_1$ and $\ddot{\mathcal{I}}_2$ the two terms in the right hand side above, we recognize that $\ddot{\mathcal{I}}_2 = 4\mathcal{K}$, where $\mathcal{K}$ is the total (translational) kinetic energy. The case of $\ddot{\mathcal{I}}_1$ requires more work. From Newton's law, we get $\ddot{\mathcal{I}}_1 = 2\sum_i \vec{F}_i \cdot \vec{X}_i$. Since $\vec{F}_i$ is the sum of the forces from all other bodies onto the i-th body, it can be written $\vec{F}_i = \sum_{j\neq i} \vec{F}_{j\to i}$. Hence:

$$\ddot{\mathcal{I}}_1 = 2\sum_i \sum_{j\neq i} \vec{F}_{j\to i} \cdot \vec{X}_i = \sum_i \sum_{j\neq i} (\vec{F}_{j\to i} \cdot \vec{X}_i + \vec{F}_{i\to j} \cdot \vec{X}_j) = \sum_i \sum_{j\neq i} \vec{F}_{j\to i} \cdot (\vec{X}_i - \vec{X}_j)$$

[2] *An introduction to star formation*, Ward-Thompson & Whitworth, Cambridge University Press, 2011

Replacing the force by its expression, we recognize in $\ddot{\mathcal{I}}_1$ twice the total (self-)gravitational potential energy of the system: $\ddot{\mathcal{I}}_1 = 2\mathcal{E}_\mathrm{G}$. Gathering the pieces, we obtain **the Virial theorem: a self-gravitating system at equilibrium satisfies the relation:**

$$\frac{1}{2}\ddot{\mathcal{I}} = \mathcal{E}_\mathrm{G} + 2\mathcal{K} = 0 \tag{7}$$

From this expression, it is customary to define the virial parameter $\alpha = 2|\mathcal{K}|/\mathcal{E}$ which takes the value of 1 for a system a equilibrium, and is < 1 for a gravitationally unstable (infalling) system.

5.1 Computing the virial parameter from the simulation

1. Compute α_sim, the virial parameter derived from the positions and velocities of the bodies in the simulation.

2. Run three simulations changing the initial rotation velocity in order to have one simulation for each case $\alpha_\mathrm{sim} \ll 1$, $= 1$, $\gg 1$. Observe whether the dynamics of the ensemble of bodies is infalling, stable, or expanding, as expected from the values of α_sim.

5.2 Computing the virial parameter from observations

From the observational point of view, in most cases one cannot access the complete position and velocity of bodies. Typically, one has access to the position on the plane-of-the-sky (POS) and the radial velocity (through the Doppler effect). A classical reasoning is the following:

- we assume that the third dimension (along the line-of-sight, LOS) is comparable to the extent in the POS;

- we assume that the distribution is approximately spherical; an effective radius R is derived, and the potential energy is derived from $\mathcal{E}_\mathrm{G} \sim -\mathcal{G}M/R$;

- we assume that the dispersion in radial velocity σ_v is comparable to the dispersion in velocity along transverse directions;

- we compute the kinetic energy from $\mathcal{K} = M\sigma_v^2/2$;

- we define the so-called virial mass by $M_\mathrm{vir} = \sigma_v R/\mathcal{G}$;

The analysis then consists in comparing the observationally measured mass M_obs with the virial mass. If $M_\mathrm{obs} \gg M_\mathrm{vir}$ the system is estimated to be gravitationally bound and infalling. If $M_\mathrm{obs} \ll M_\mathrm{vir}$ it is estimated to be gravitationally unbound and expanding. If $M_\mathrm{obs} \sim M_\mathrm{vir}$, the situation is unclear. Formally, the system could be considered gravitationnally bound and stable (at equilibrium), but due to uncertainties it is difficult to conclude.

1. Find a link between α_vir, M_vir, and M_obs, assuming that $M_\mathrm{obs} = M$. For what value of $M_\mathrm{obs}/M_\mathrm{vir}$ the system would in principle be at equilibrium ?

2. We consider that an observer is observing the simulated system from a location far along the z-axis: $(0, 0, -\infty)$. We neglect the cone effect due to the finite distance of the observer. Use the reasoning presented above to estimate the dynamical state of the observed system.