GANDALF school

Generating initial conditions in python

Giovanni Rosotti, David Hubber
Richard Booth, James Dale, Judith Ngoumou

1. **Uniform square**

   Generate an uniform square (i.e., in 2d) via Python using random placement of particles. Set all velocities to zero and give an arbitrary value to the temperature. Use periodic boundary conditions.

   Do a plot of the particle positions. How do the positions evolve in time if you now run the simulation? See what changes by changing the initial value of the temperature.

   Do also a rendered plot and see how that changes in time. What do you notice?

2. **A proto-planetary disc**

   Generate the initial conditions for a 3d proto-planetary disc via Python. To do this:

   - Assume that the vertical density profile is a Gaussian with standard deviation H, that the disc is azimuthally symmetric, and that the radial surface density profile is $\Sigma \propto r^{-1}$.
   - Assume that $H = c_s/\Omega$, where c_s is the sound speed given by $c_s = 0.05$ and Ω is the keplerian velocity.
   - Initialise the velocity to the keplerian value (only in the azimuthal direction).

   Suggestion: work in cylindrical coordinates. Use dimensionless units where $G = M_* = 1$. And don't forget to put a star in the center (what happens if you don't?). To start with, you can start in 2d and add the vertical dimension later.

   Do a rendered plot of the disc seen face-on. What do you notice in the initial conditions? What happens as time evolves (run for a few orbits)? Use an isothermal equation of state. Note: the temperature is set through the parameter temp0.