A Computational Cluster Cookbook

Head Empty, No Thoughts

(Rafael Bertolotto, Fiona Han, Logan White)*

*Names in alphabetical order.

I. A simple recipe to bake a stellar cluster

Baking initial conditions

- We need the following ingredients:
  - Masses
  - Positions
  - Velocities

Baking initial conditions… that don't go boom

- We need the following ingredients:
    - Masses
    - Positions
    - Velocities

- We want them to start in equilibrium

Baking a stellar cluster… that don't go boom

- We need the following ingredients:
    - Masses
    - Positions
    - Velocities

- We want them to start in equilibrium

- We want the cluster to actually look like a cluster

Baking a stellar cluster… that don’t go boom

IMF model

Baking a stellar cluster… that don't go boom

IMF model → Draw masses

Baking a stellar cluster… that don't go boom

IMF model → Draw masses

Density profile → Draw positions

Baking a stellar cluster… that don't go boom

IMF model → Draw masses

Density profile → Draw positions

Poisson's equation → $f(\vec{r}, \vec{v}) = f(E)$

$$\nabla \phi = -4\pi G \rho$$

Boltzmann's equation

$$\vec{v} \cdot \nabla_{\vec{r}} f = \nabla_{\vec{r}} \phi \cdot \nabla_{\vec{v}} f$$

Baking a stellar cluster… that don't go boom

IMF model → Draw masses

Density profile → Draw positions

Draw masses, Draw positions, $f(\vec{r}, \vec{v}) = f(E)$ → Draw velocities

Density profile, Poisson's equation, Boltzmann's equation → $f(\vec{r}, \vec{v}) = f(E)$

Poisson's equation

$$\nabla \phi = -4\pi G \rho$$

Boltzmann's equation

$$\vec{v} \cdot \nabla_{\vec{r}} f = \nabla_{\vec{r}} \phi \cdot \nabla_{\vec{v}} f$$

Our recipe for a perfectly baked stellar cluster

Piecewise power-law IMF: $$\Phi(m) \propto m^{-\alpha}$$

Our recipe for a perfectly baked stellar cluster

Piecewise power-law IMF: $$\Phi(m) \propto m^{-\alpha}$$

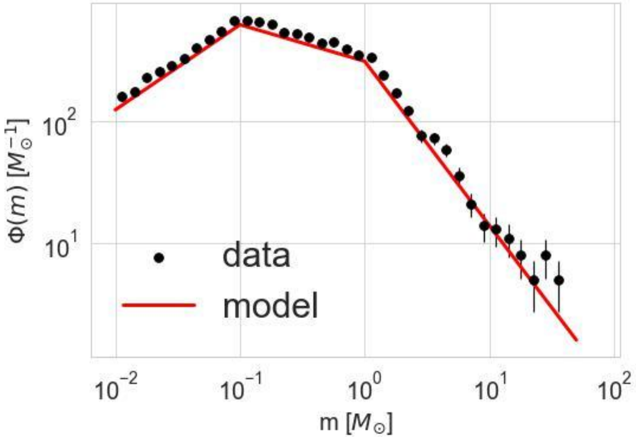
Plummer sphere cluster: $$\phi(\vec{r}) = \frac{GM}{\sqrt{r^2 - r_0^2}}$$

Our recipe for a perfectly baked stellar cluster

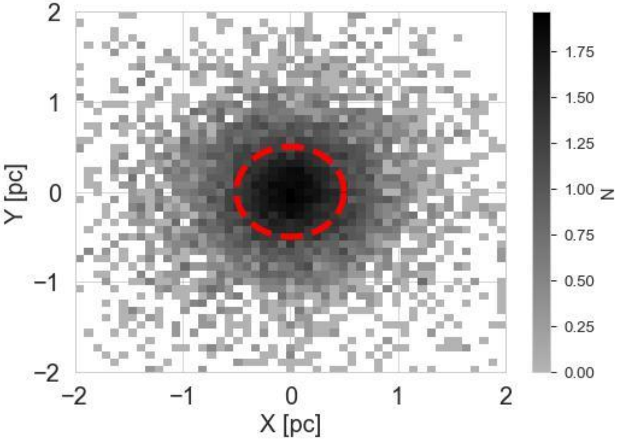
Piecewise power-law IMF: $$\Phi(m) \propto m^{-\alpha}$$

Plummer sphere cluster: $$\rho(\vec{r}) = \frac{3Mr_0^2}{4\pi(r^2+r_0^2)^{5/2}}$$

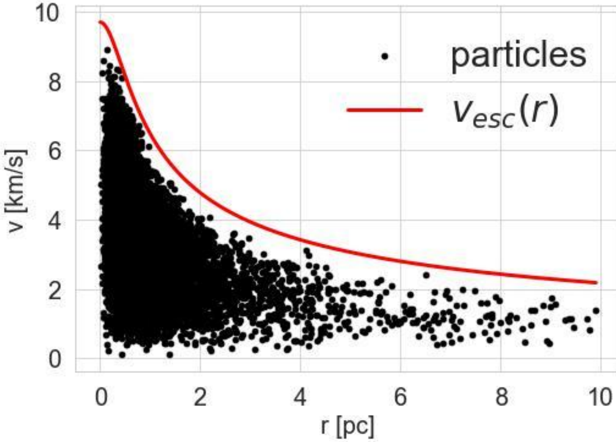
A (10,000 particles) freshly baked cluster!



Kroupa IMF



$$r_0 = 0.5\, pc$$



$$v < v_{\text{esc}}$$

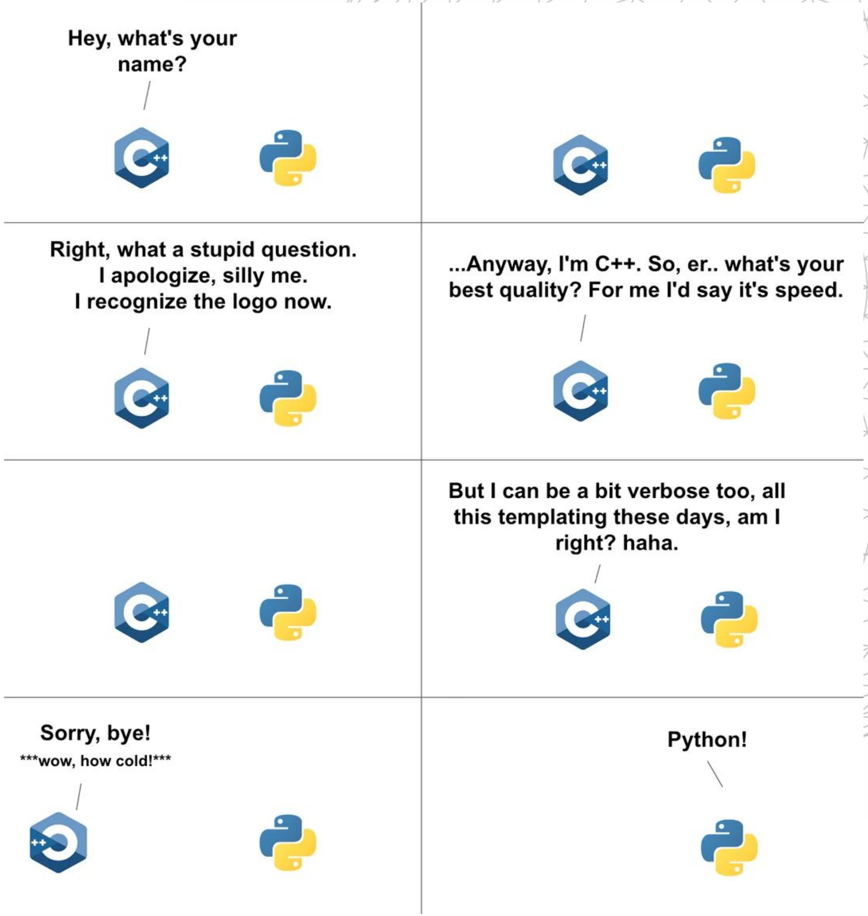
II. How to Make One Raisin on the Bread

N-body Problem

- Chaotic dynamics: almost guaranteed to not have a stable solution
- Used in everything from cluster to galactical to cosmological simulations!
  - Can only numerically solve these equations.

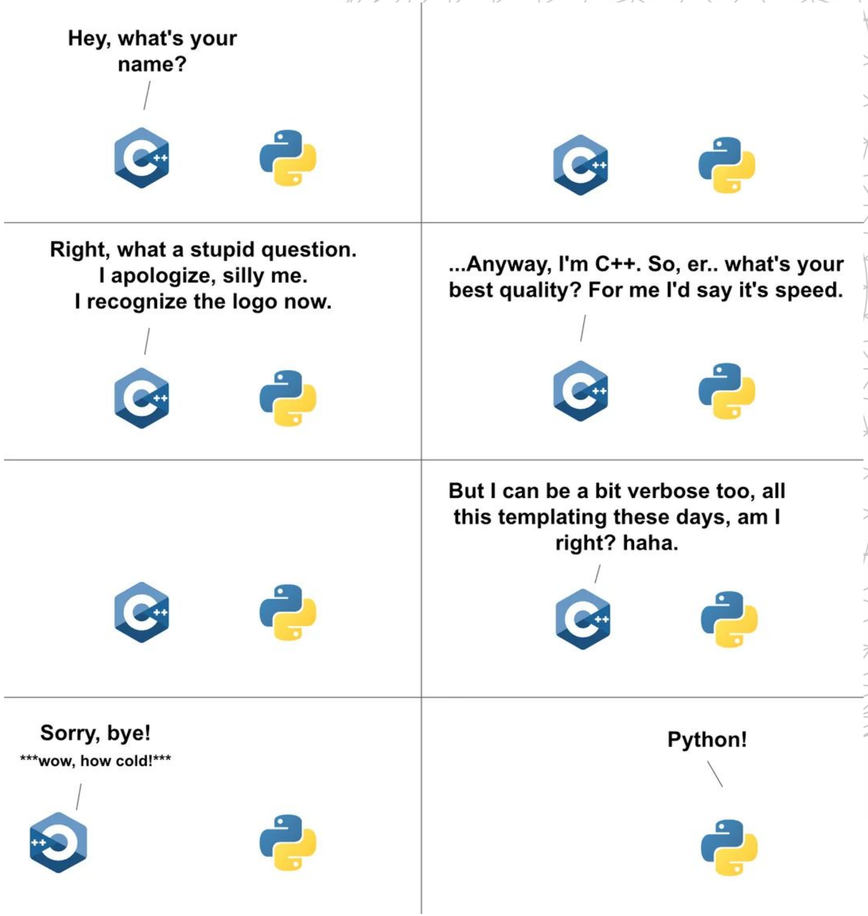
N-body Problem: traditional

- Solves the system fully;
- Expensive simulations: O(N^2)
- Python is usually slow.



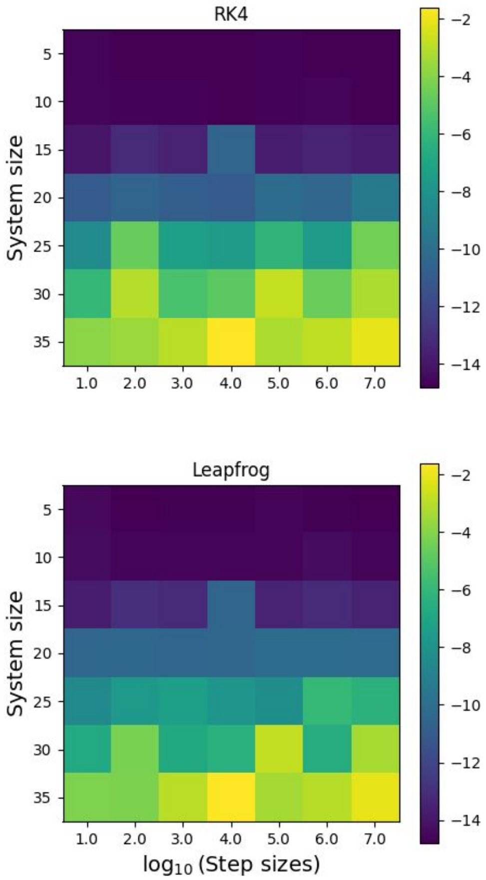
N-body Problem: traditional

- Solves the system fully;
- Expensive simulations: O(N^2)
- Python is usually slow.
- **Vectorization!**
    - Replace Python for loop with NumPy array operations based in C.
    - Fancy way of saying “np.einsum is fast”.



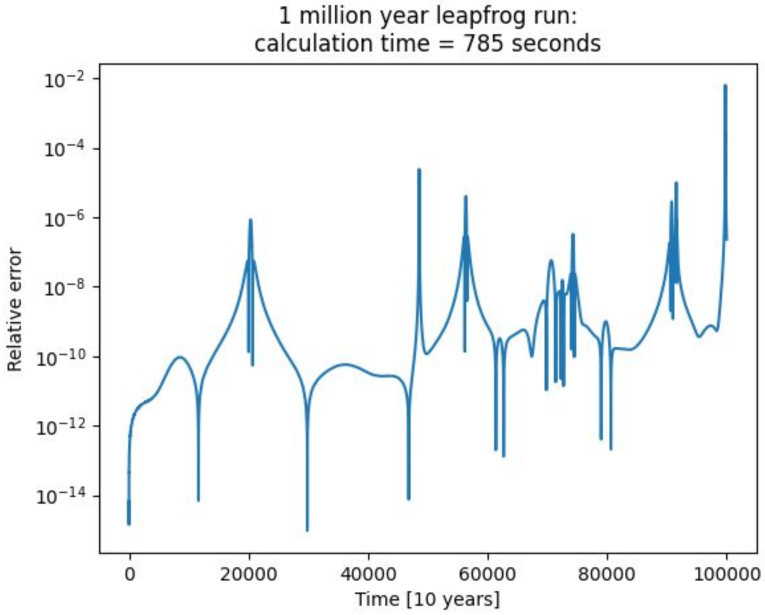
Performance & Error

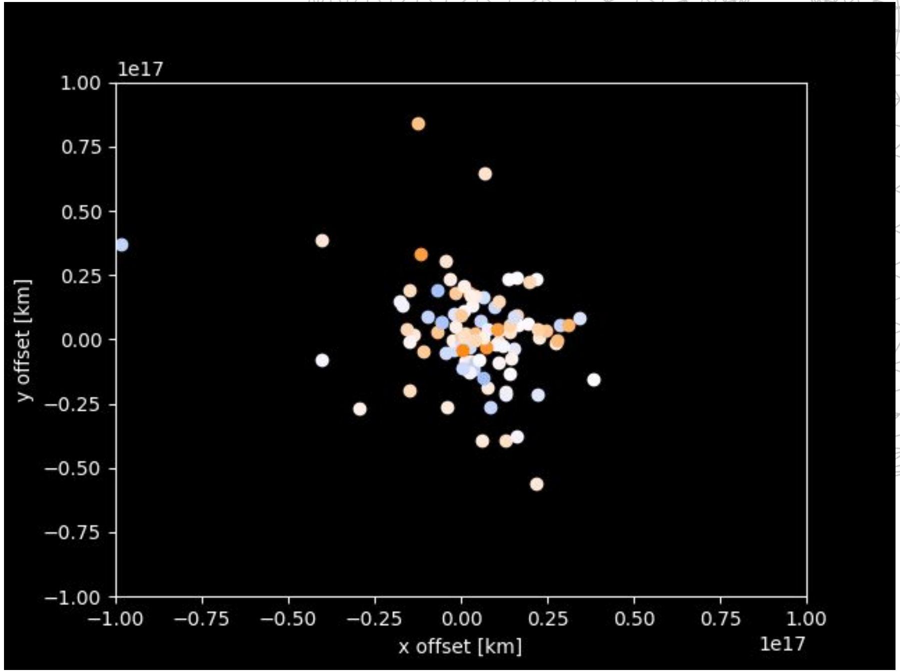
- Implemented RK4 and leapfrog.
    - RK4: higher precision
    - Leapfrog: bounded energy
- Leapfrog seems to have slightly better performance…
    - It is the “default” for longer runs.



Performance & error

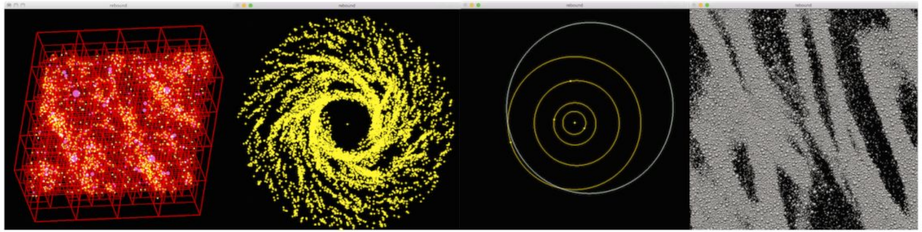
Longest run: 1 million years, dt = 10. Finished in 13 minutes.



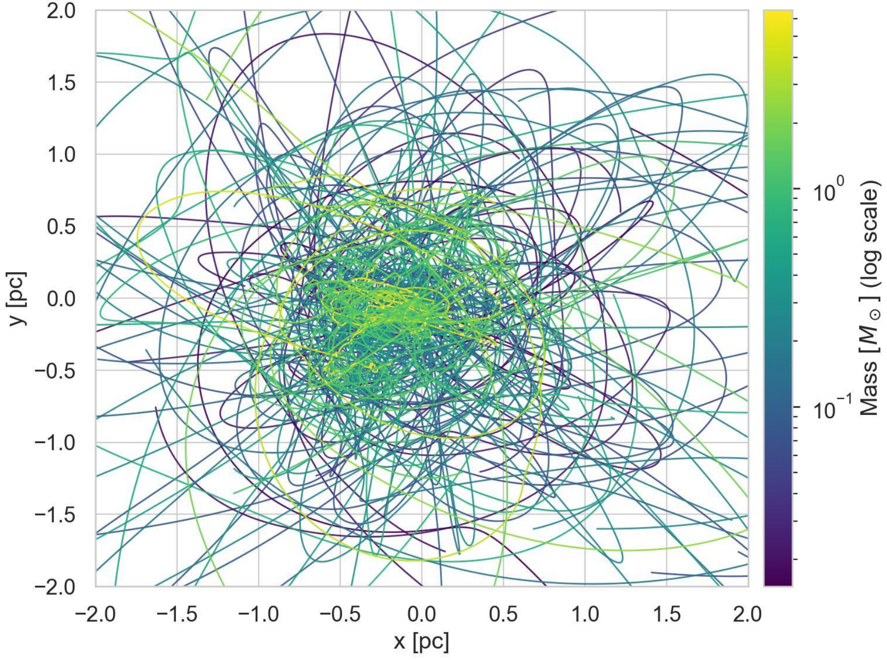


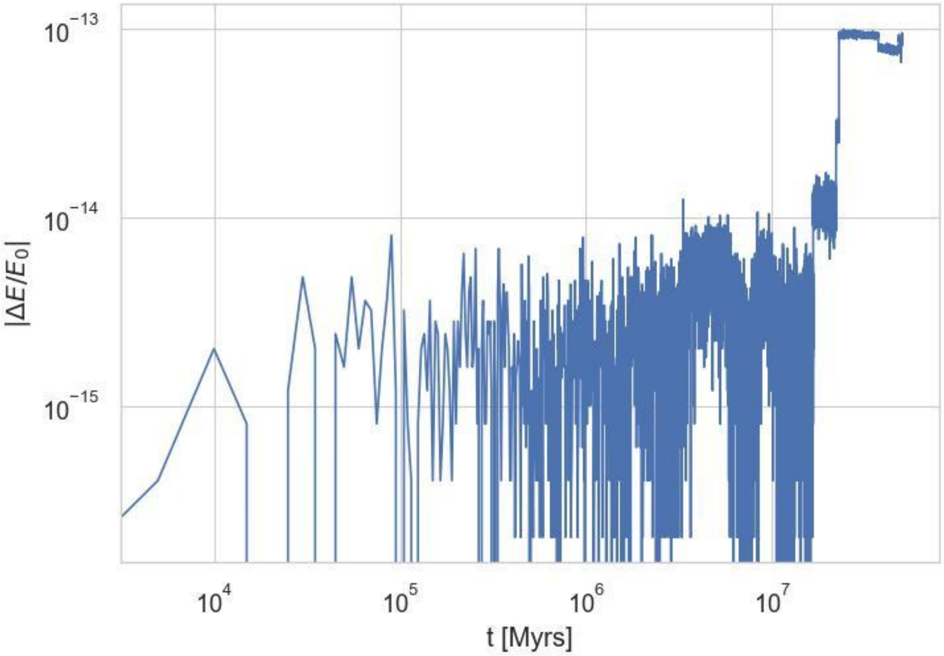
What about a 15th order IRK? (you read that right)

REBOUND
Open Source N-body Code



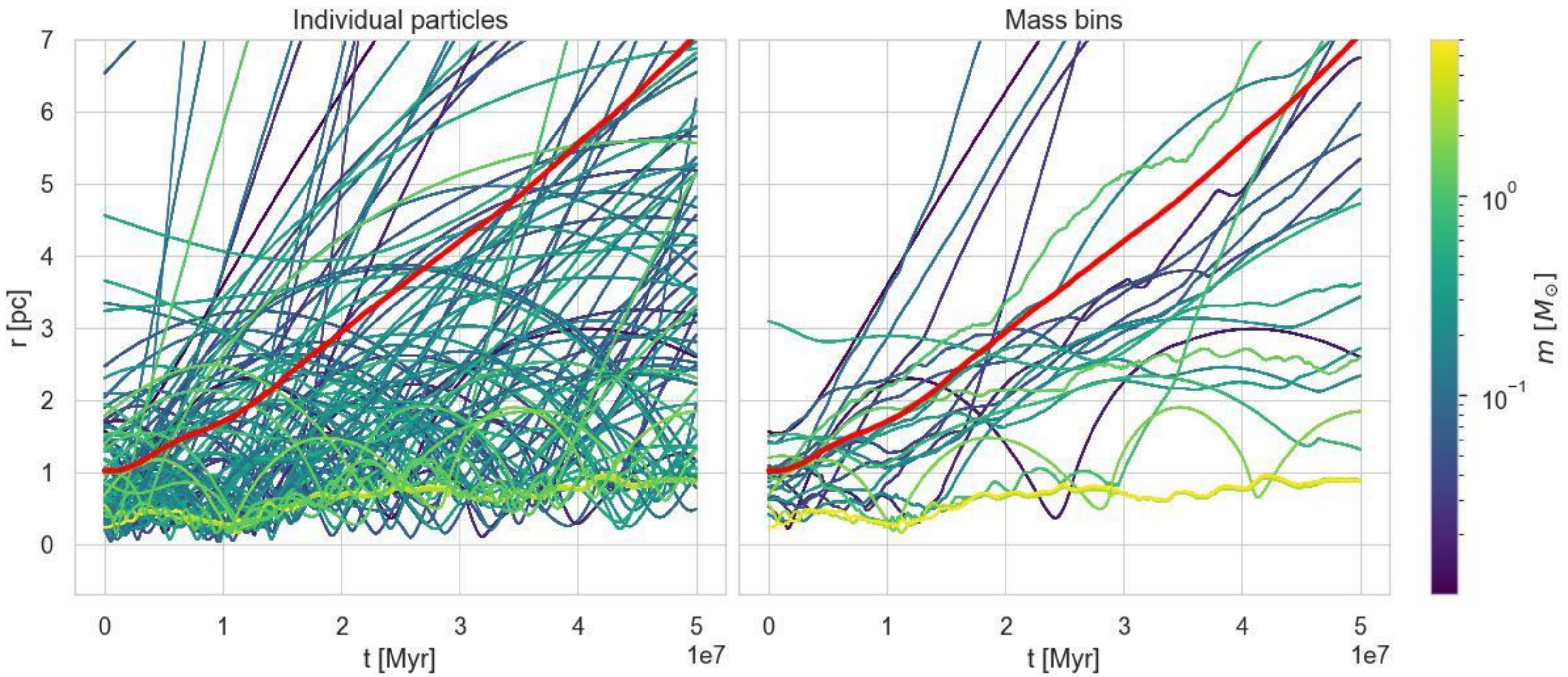
What about a 15th order IRK? (you read that right)





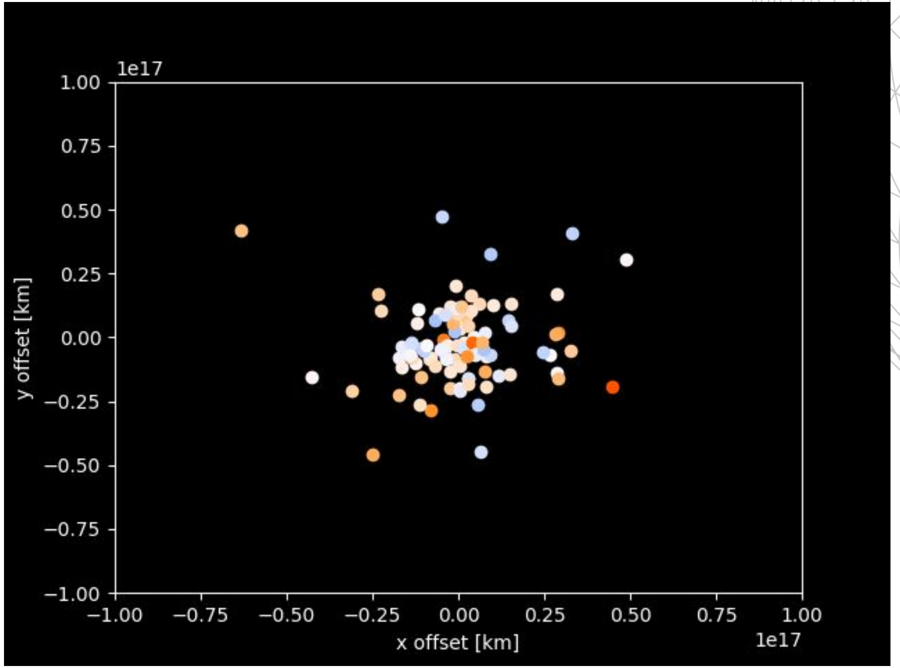
What about a 15th order IRK? (you read that right)

We see mass segregation!



Pretty simulation

- The stars in this cluster is color coded. (1 million years)



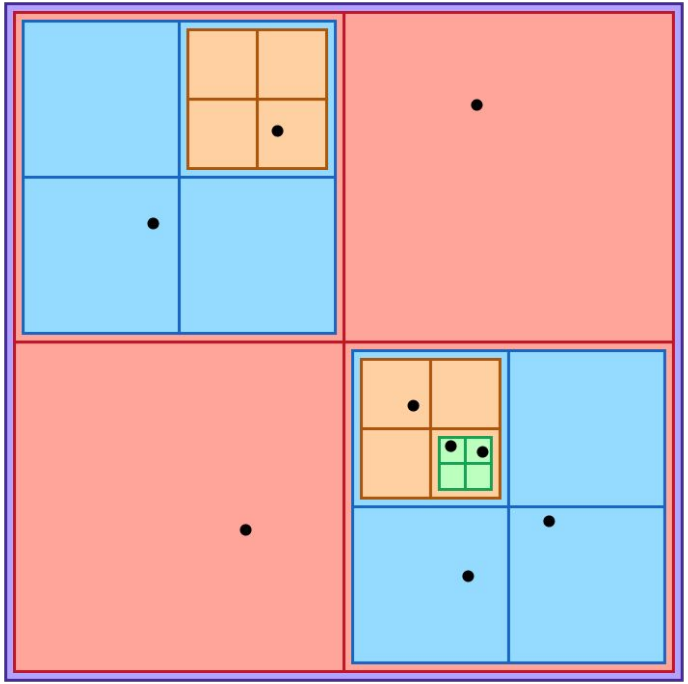
III. An N-body Sourdough Starter

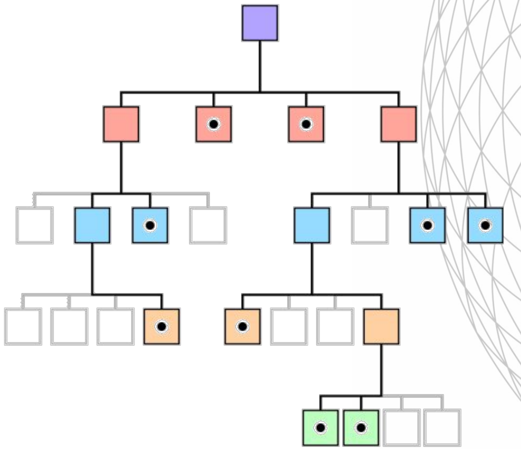
N-body Problem: Large scale?

- $O(N^2)$ grows very fast!

- Barnes–Hut algorithm.

How could we possibly get bigger?





Say hello to the **Barnes–Hut algorithm** (Barnes & Hut, 1986)! We're going to take a brief compsci interlude and talk about **trees**.

This is a **quadtree.** It has a **root**, with four **leaves**.

What do trees have to do with this?

Building our simulation as a tree puts each of our particles in its own box. This means we can perform **averaged force calculations** with our algorithm.

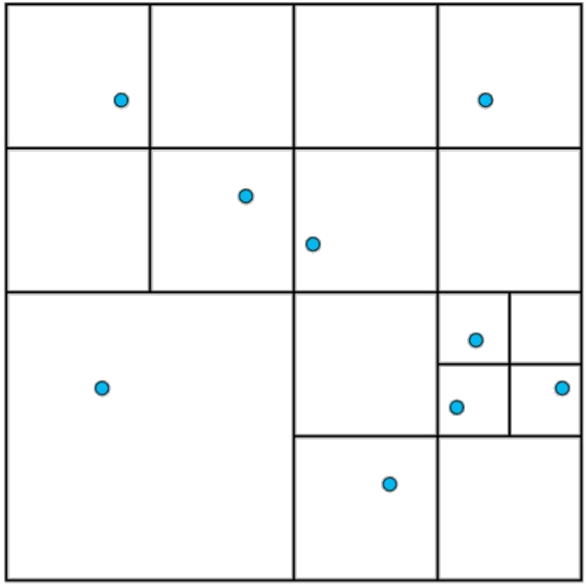
For a single body we don't want to calculate the force from all of the other particles in a large ($n > 10^4$) simulation– it's expensive.

Instead, we define a threshold **theta** ($\boldsymbol{\theta}$), the ratio between **node width** (s) and **distance to center of mass** (d). If:

1. ratio < θ, we do a direct force computation.
2. otherwise, we use the node's center of mass and total mass to get net force.

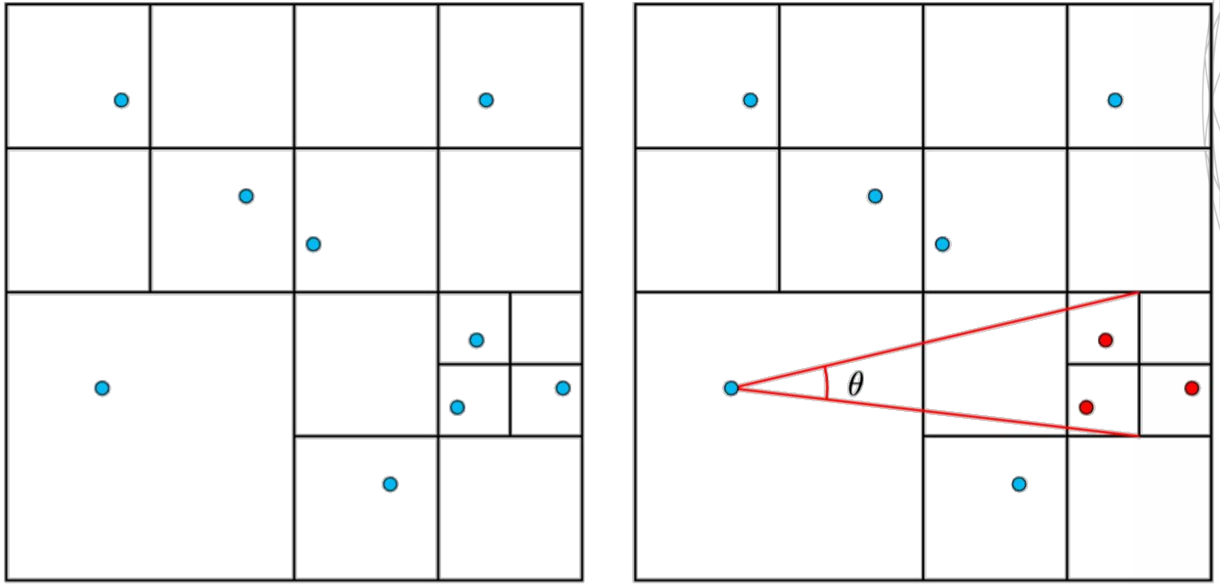
$$\text{def.} \quad \boldsymbol{\theta} = \mathbf{s} \,/\, \mathbf{d}$$

What do trees have to do with this?



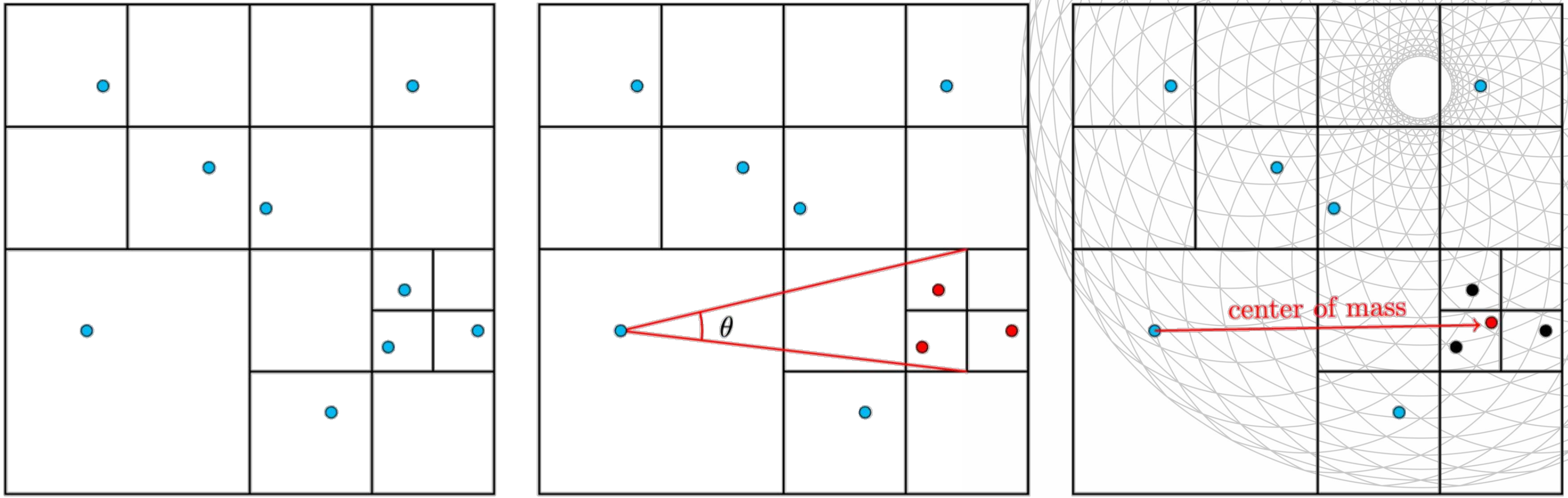
(a) Decomposing the spatial domain

What do trees have to do with this?



(a) Decomposing the spatial domain (b) Checking opening angle

What do trees have to do with this?



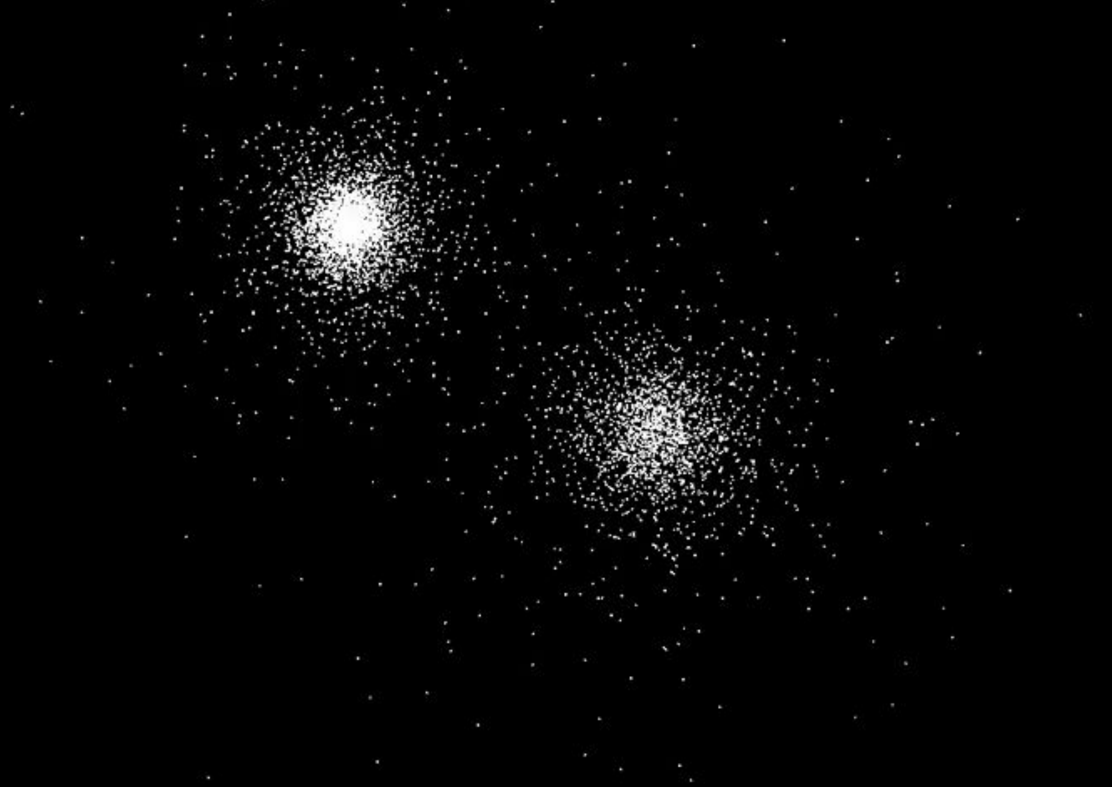
(a) Decomposing the spatial domain (b) Checking opening angle (c) Taking approximation

Hey, a home-cooked pair of clusters!

globr has the capability to model a cluster(s) of ~10^4 stars in three dimensions.

Because of Barnes–Hut, we can utilize a simple leapfrog integration scheme!

> **pros.** We can model medium to large systems with ease!
>
> **cons.** Because of the B–H approximation, we have higher error (no energy cons.)



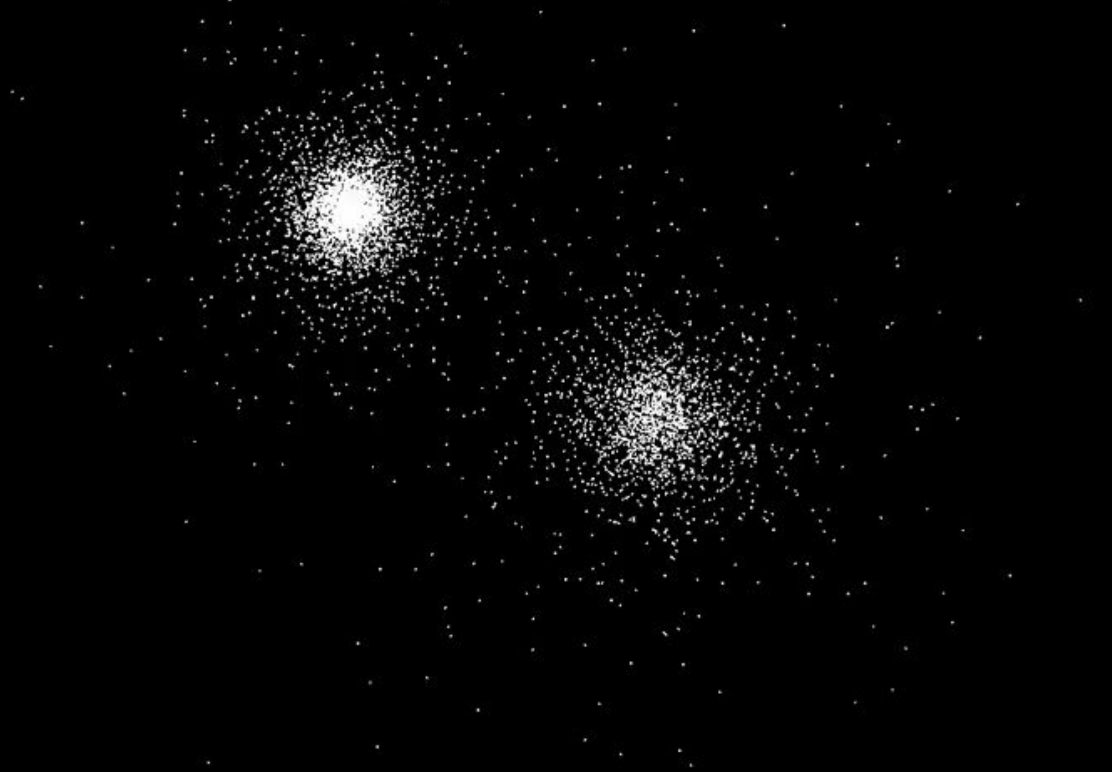
Two globular clusters intersecting!
$n_1 = 2000$, $R_0 = 0.5$ pc
$n_2 = 4000$, $R_0 = 0.3$ pc
Krouppa IMF

Hey, a home-cooked pair of clusters!

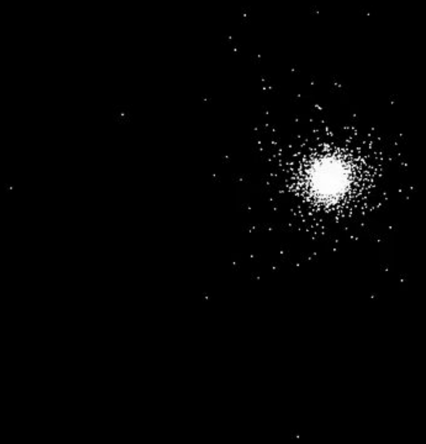
globr has the capability to model a cluster(s) of ~10^4 stars in three dimensions.

Output files provide star mass, positions, and simulation and physical diagnostics such as:

1. Physical time,
2. Kinetic and potential energy,
3. Simulation size (physical)

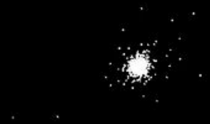


Two globular clusters intersecting!
n_1 = 2000, R_0= 0.5 pc
n_2 = 4000, R_0= 0.3 pc
Krouppa IMF



Krouppa IMF

$n_1 = 10^4$, $R_0 = 0.5$ pc

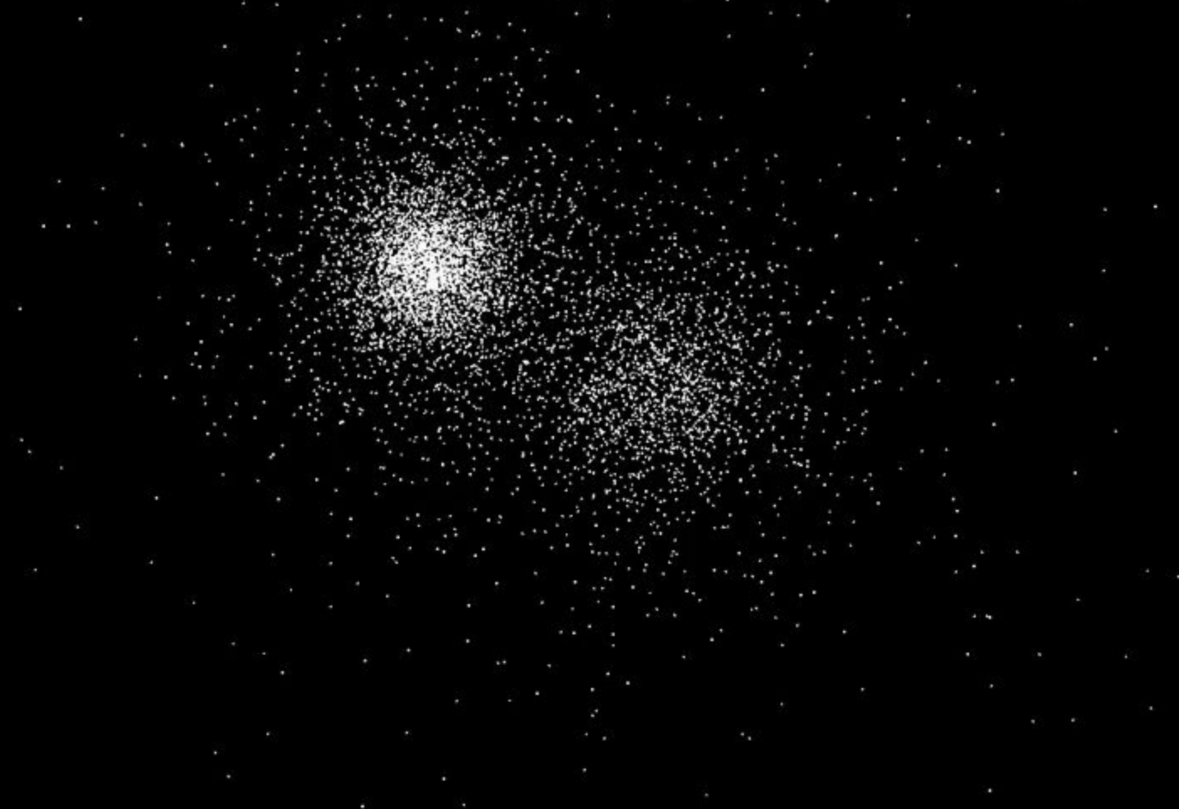


Salpeter IMF

$n_1 = 10^4$, $R_0 = 0.5$ pc

What if we change the initial conditions?

Two globular clusters intersecting (and merging)!
n_1 = 2000, R_0= 0.5 pc
n_2 = 4000, R_0= 0.3 pc
Krouppa IMF



Questions?

Thanks for listening!

N-Body Problem: Applied

- Simulate real cluster dynamics using an integrated IMF as the initial condition in our simulated cluster.

Cool plots to do

1. Performance plots:
   a. DeltaE/E0 vs t - done(python), vs dt might also be fun
   b. Final DeltaE/E0 vs N - also done
   c. Total t vs N
   d. Theta vs t (or N)
2. Mass segregation and core-collapse: r-r_cm vs t for different mass bins
3. Evolution of virial factor Q=T/U vs t
4. Previous plots for very steep IMF and not steep IMF