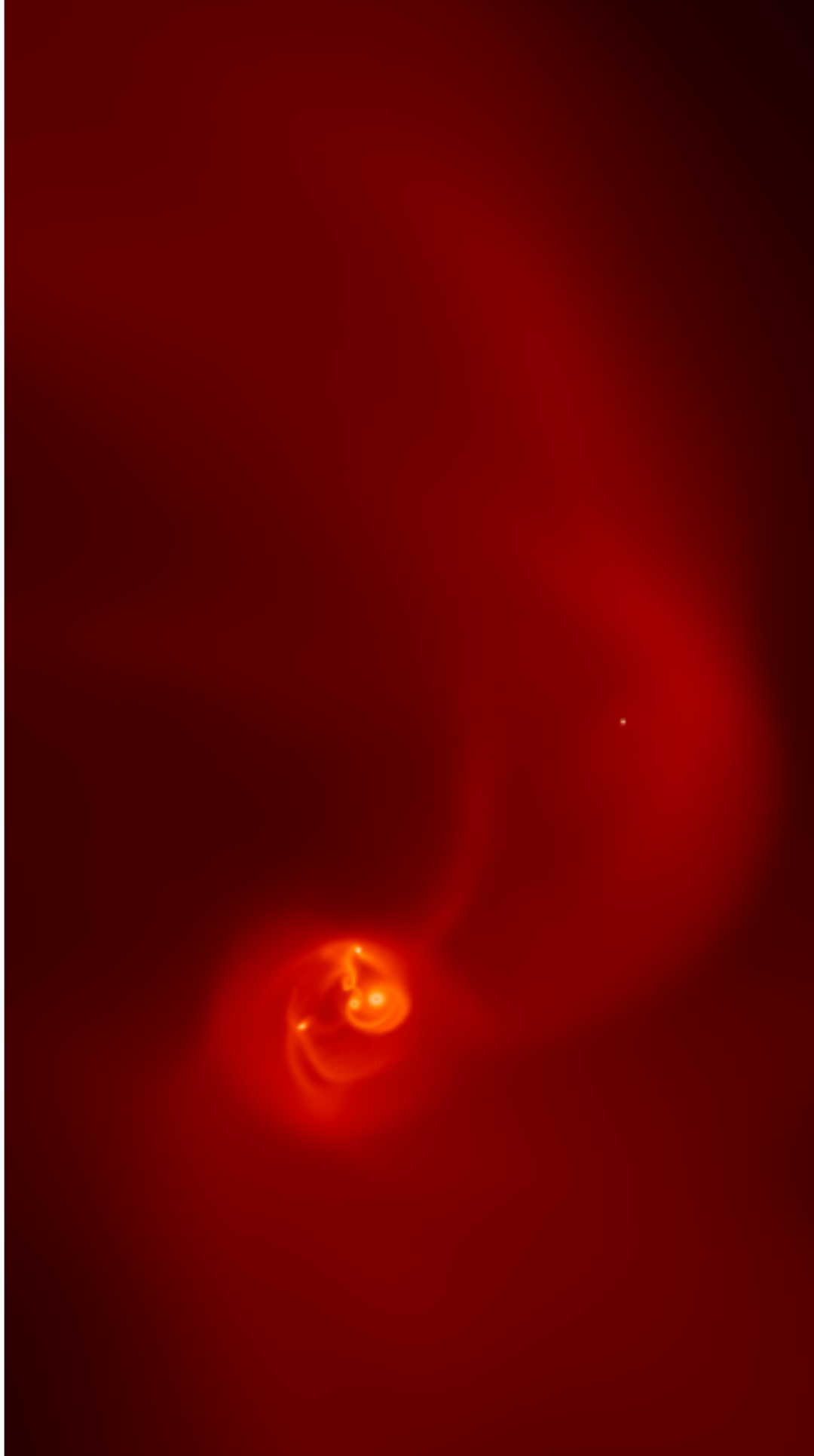
Smoothed Particle Hydrodynamics

David Hubber

USM, LMU, München
Excellence Cluster Universe,
Garching bei München

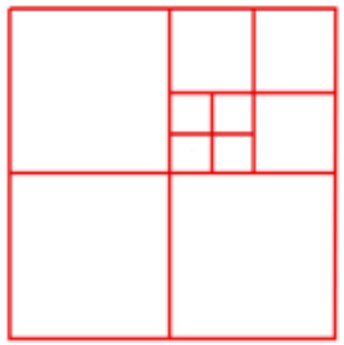
27th October 2015



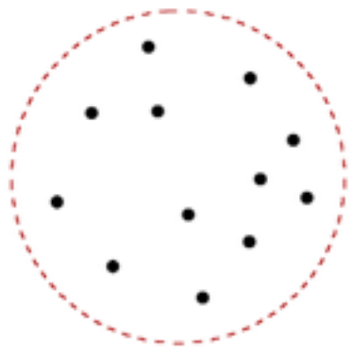
Multi-dimensional numerical hydrodynamics

- Many different paradigms have been developed over the last several decades

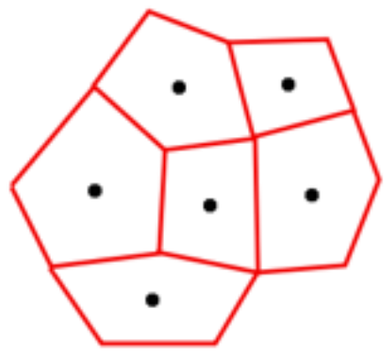
Grid



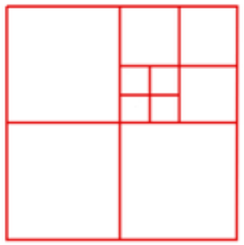
Particle



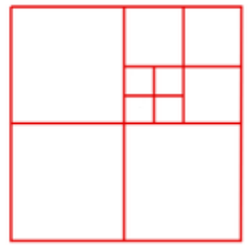
Tessellation



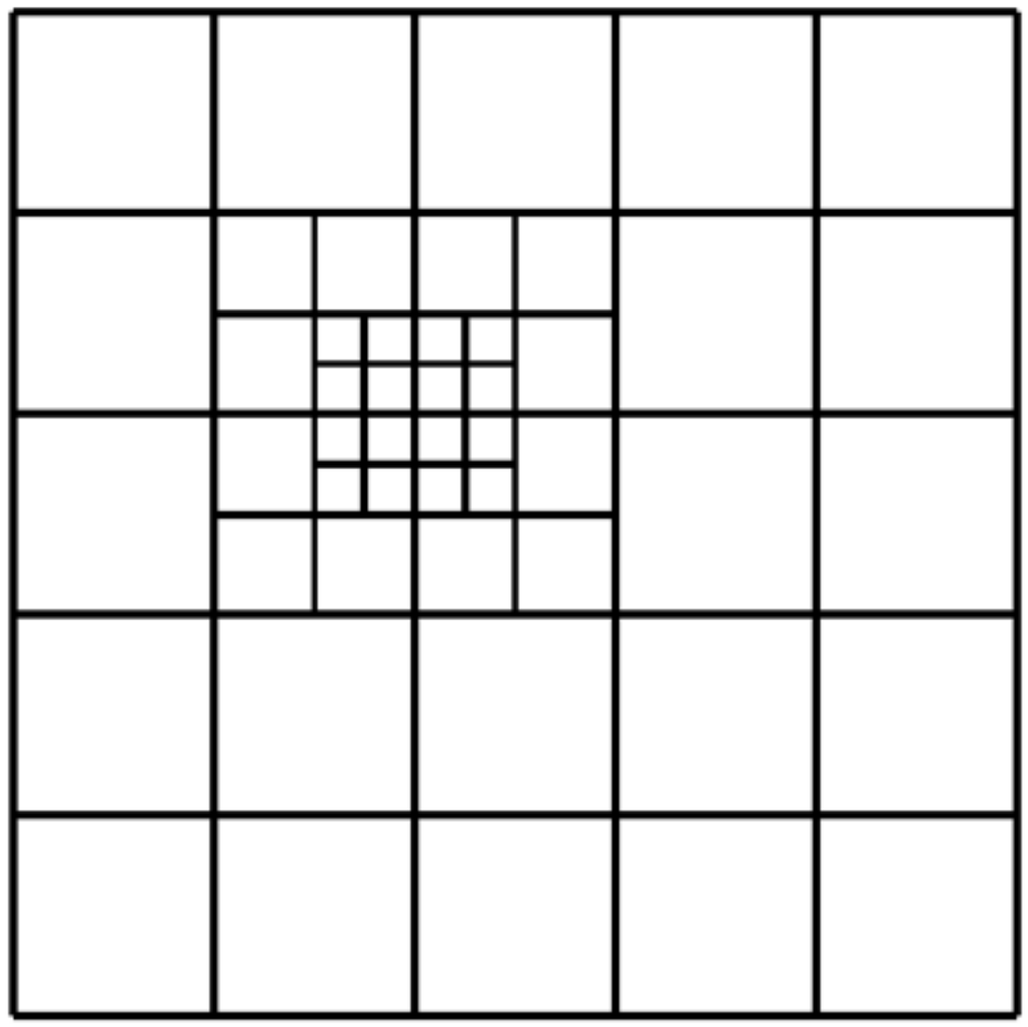
- Large spectrum of different algorithms and codes exist merging different aspects of the basic concepts.

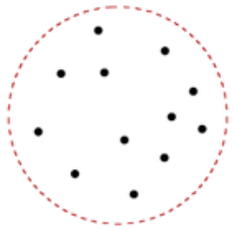


Grid methods

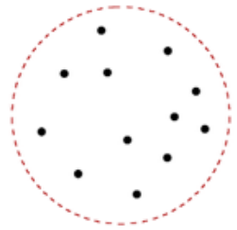


- Fluid is divided into a regular grid
- Grid cells are static
  - Eulerian representation
- Calculate rate of flow of mass, momentum and energy between neighbouring grid cells
  - Calculate net transfer for current timestep
- If higher resolution is needed in some regions, use Adaptive Mesh Refinement (AMR)
  - Split cells hierarchically into 4 smaller cells (8 for 3D)

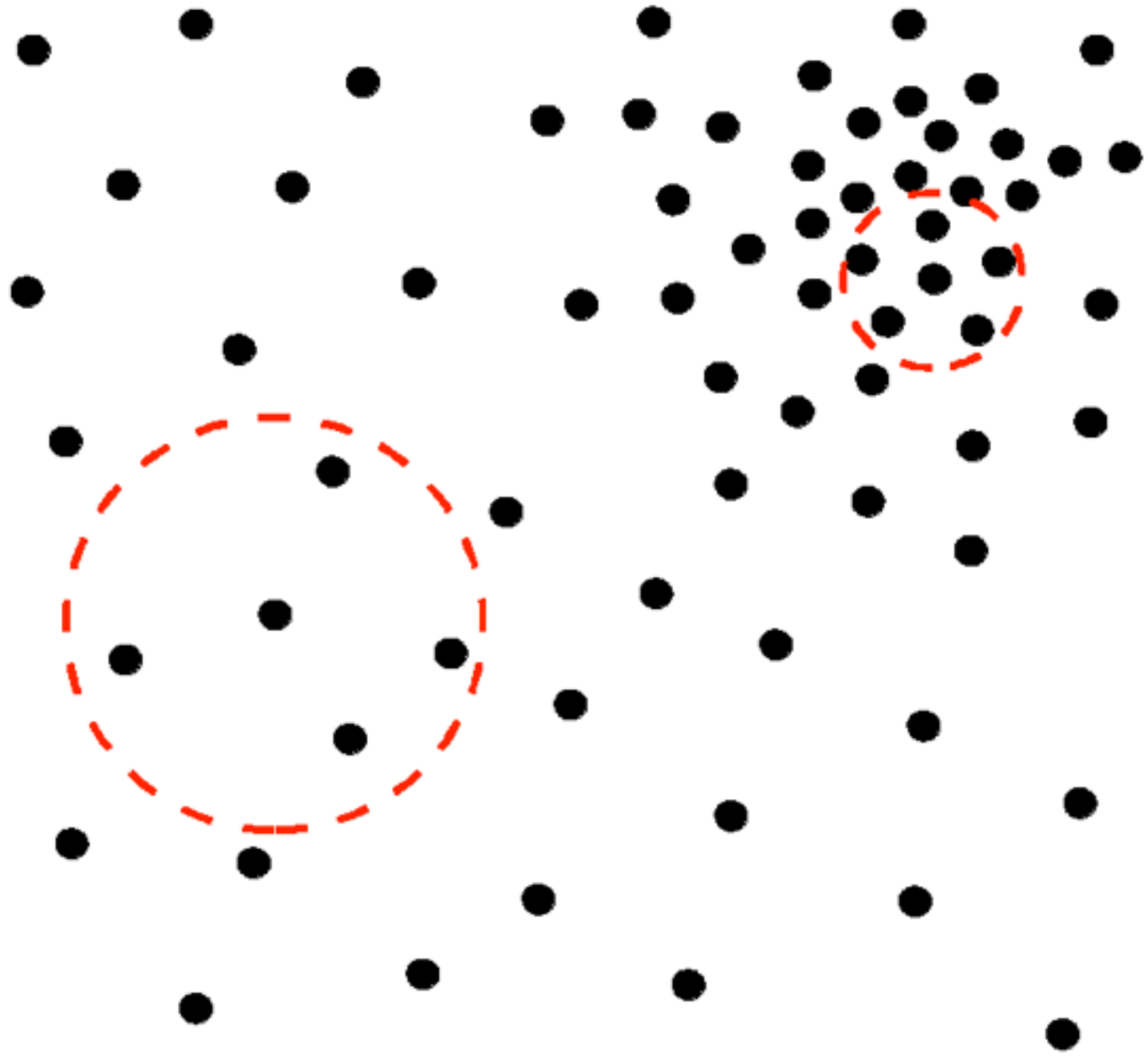


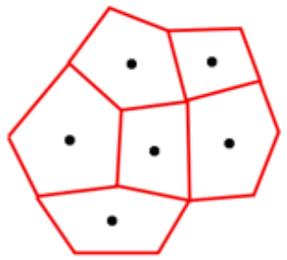


Smoothed Particle hydrodynamics (SPH)

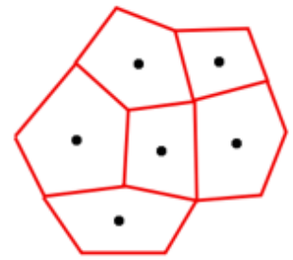


- Fluid is divided into discrete mass elements represented by particles
- Particles only interact hydrodynamically with other particles within a distance called the *smoothing length*
- The accuracy can be increased by increasing the number of neighbours inside the smoothing length
- Smoothing length adapts to keep same amount of mass inside interaction sphere, i.e. automatic spatial adaption

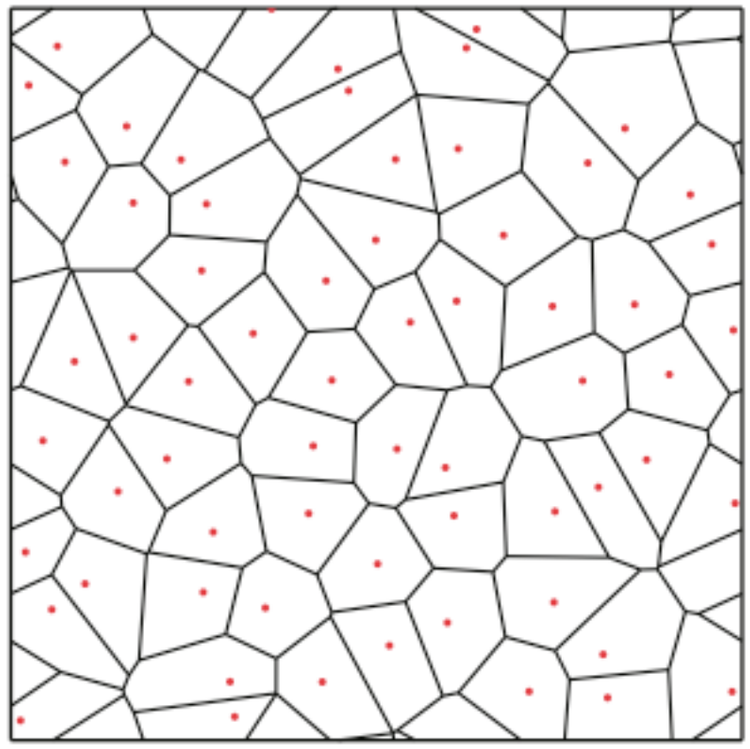




Tessellation methods



- Springel (2009) proposed a new unstructured finite-volume grid code.
  - Instead of grids locations being fixed in space, use particles as ‘mesh-generating points’
  - Use Voronoi tessellation to find cell volumes plus interaction surfaces
  - Solve hydrodynamic equations at each interacting surface using Finite-Volume method



Springel (2009)

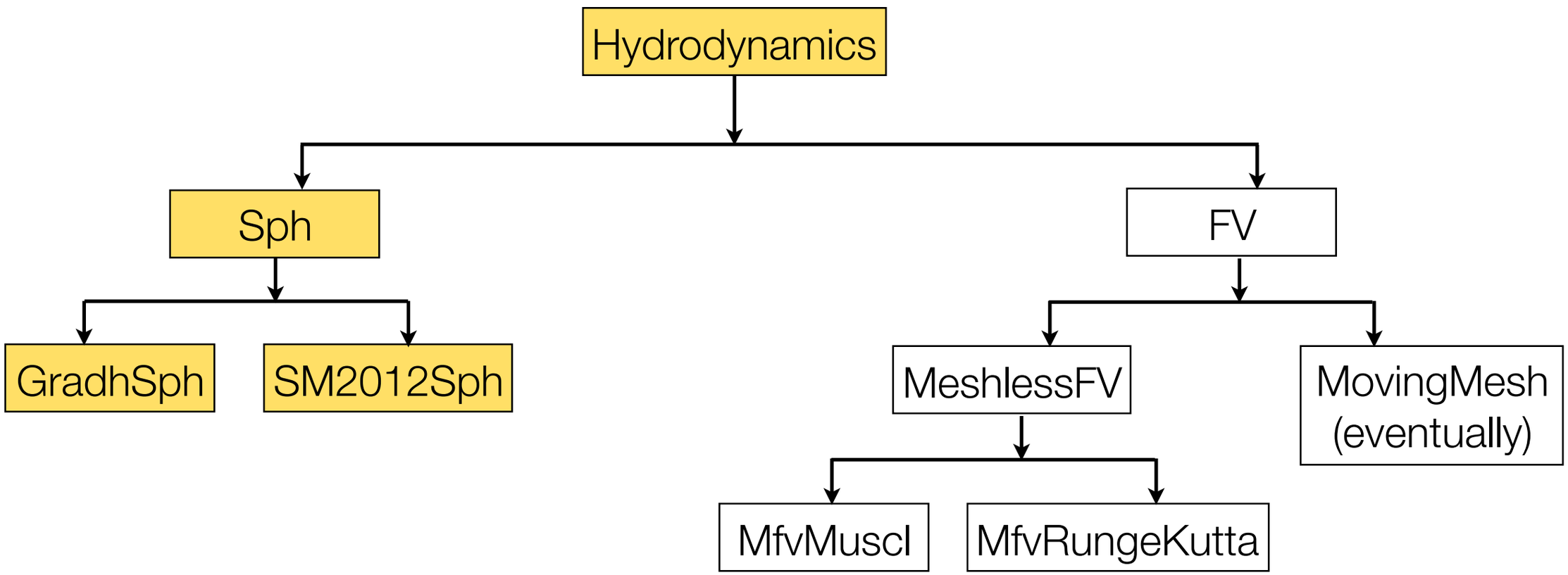
- Adopts many advantages from both SPH and Grid
  - Galilean invariant (not technically Lagrangian, but can be made so)
  - Adaptive, unstructured mesh, so no preferential directions or advection problems
  - Easy to increase/decrease resolution at any point in the simulation with any criteria (a la AMR)

- Some disadvantages though
  - Increased complexity on both SPH and Grid
  - Shock capturing not as good as uniform grid or AMR (although better than SPH)

Hydrodynamics class in GANDALF

- The main base class for defining the hydrodynamical algorithms in GANDALF is the **Hydrodynamics** class

- All Hydrodynamical algorithms must inherit from this class



Smoothing

- The smoothed value of any physical property *A* at a position **r** is calculated by convolving *A* over all space with a smoothing function, i.e.

$$\langle A(\mathbf{r})\rangle = \int_V A(\mathbf{r}')\, W(\mathbf{r}-\mathbf{r}', h)\, d^3\mathbf{r}'$$

W - *smoothing kernel*

h - *smoothing length* (determines the width of the smoothing function)

- The smoothing kernel can be any function provided it satisfies the following two properties

1. - Normalisation

$$\int_V W(\mathbf{r}-\mathbf{r}', h)\, d^3\mathbf{r}' = 1$$

2. - Dirac delta function

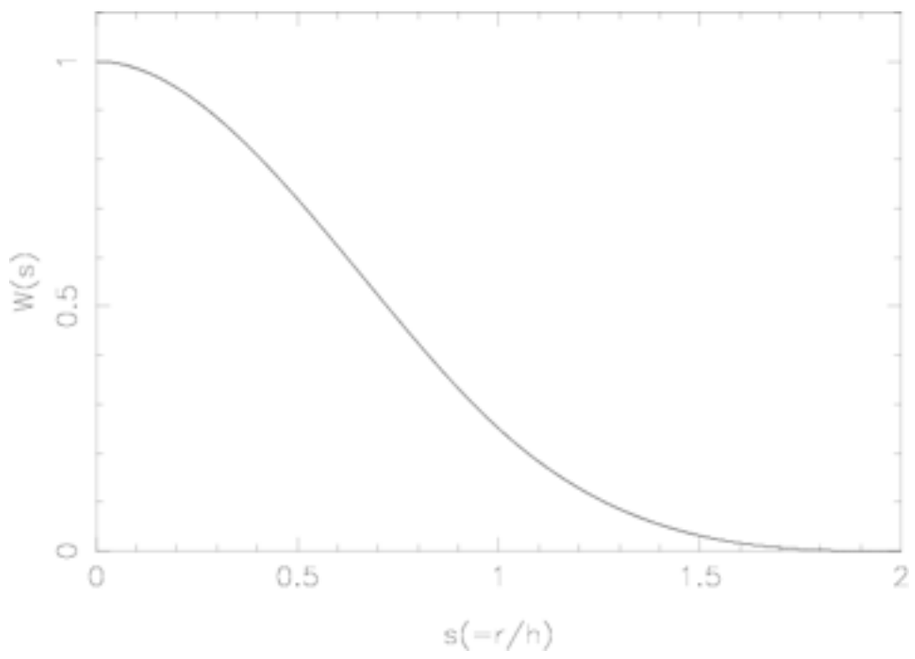
$$\lim_{h\to 0} W(\mathbf{r}-\mathbf{r}', h) = \delta(\mathbf{r}-\mathbf{r}')$$

The smoothing kernel

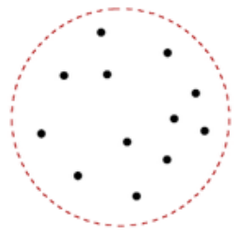
- The most employed smoothing kernel is the M4 cubic spline kernel (Monaghan & Lattanzio 1985)

$$W(r,h) = \frac{1}{\pi h^3} \left\{ \begin{array}{ll} 1 - \frac{3}{2}\left(\frac{r}{h}\right)^2 + \frac{3}{4}\left(\frac{r}{h}\right)^3 & \text{if } \; 0 \leq r \leq h\,; \\ \frac{1}{4}\left(2 - \frac{r}{h}\right)^3 & \text{if } \; h \leq r \leq 2\,h\,; \\ 0 & \text{if } \; r > 2h\,. \end{array} \right\}$$

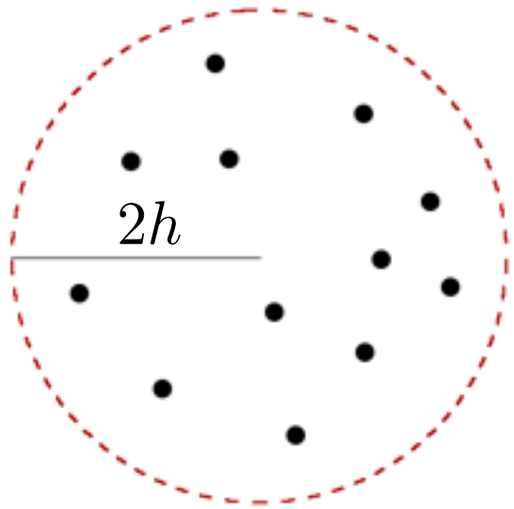
- The function itself, its first and second derivative are continuous (but not the third deriviative).
- Has compact support (i.e. has a finite range)



The summation approximation



- We now consider how to calculate the smoothed averages of an ensemble of discrete points instead of a continuous function.
- Assume we have an ensemble of masses m_j with positions $\mathbf{r}_j$, and with the property A_j. We can (approximately) equate the mass of the particle with $\rho_j \, d^3\mathbf{r}$.



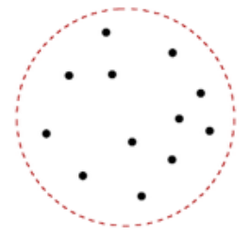
- The smoothed average of the property A at a position $\mathbf{r}$ is given by

$$\langle A(\mathbf{r})\rangle = \int_V \frac{A(\mathbf{r}')}{\rho(\mathbf{r}')}\, W(\mathbf{r}-\mathbf{r}', h)\, \rho(\mathbf{r}')\, d^3\mathbf{r}' = \sum_{j=1}^{N} m_j \frac{A_j}{\rho_j}\, W(\mathbf{r}-\mathbf{r}_j, h)$$

- Note that we cannot calculate the smoothed value of any property without knowing the density first. However, if we insert $A = \rho$, then we obtain the density

$$\langle \rho(\mathbf{r})\rangle = \sum_{j=1}^{N} m_j\, W(\mathbf{r}-\mathbf{r}_j, h)$$

Smoothed Particle Hydrodynamics



- We will now apply smoothing techniques to solve the equations of hydrodynamics using particles, i.e. Smoothed Particle Hydrodynamics
- The first step in SPH is always to calculate the density of all particles, i.e.

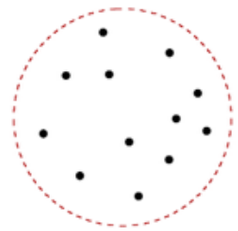
$$\rho_i = \langle \rho(\mathbf{r}_i) \rangle = \sum_{j=1}^{N} m_j \, W(\mathbf{r}_i - \mathbf{r}_j, h_i)$$

- Calculating the density directly by summation replaces the conservation of mass equation. Since the mass of a particle in an intrinsic property that cannot be changed, mass is automatically conserved in SPH.
- The smoothing length is calculated via the density by :

$$h_i = \eta \left(\frac{m_i}{\rho_i} \right)^{1/D}$$

- Note that since the smoothing length and the density depend on each other, the correct value must be obtained by iterating. This can be achieved using fixed point iteration (safe), or, for example, a Newton-Rhapson iterator (fast).

Using Lagrangian mechanics



- The SPH equations can also be derived using Lagrangian mechanics. This has the advantages of resulting in SPH equations which automatically obey the conservation laws.
- The Lagrangian for a hydrodynamic fluid is given by

$$L = \int \left(\frac{1}{2}\rho v^2 - \rho\, u(\rho, s) \right) d^3\mathbf{r}' \quad \Rightarrow \quad L = \sum_{j=1}^{N} \left(\frac{1}{2} m_j v_j^2 - m_j u_j(\rho_j, s_j) \right)$$

- The only other information that we must provide is

$$\rho_j = \sum_{k=1}^{N} m_k\, W(\mathbf{r}_j - \mathbf{r}_k, h)$$

(the SPH density equation)

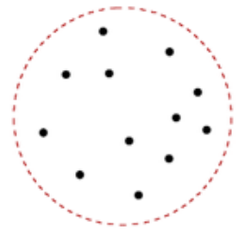
$$du_j = \frac{P_j}{\rho_j} d\rho_j$$

(from the 1st law of thermodynamics assuming no dissipation)

- We can then derive the equations of motion using the Euler-Lagrange equations

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \mathbf{v}_i} \right) - \frac{\partial L}{\partial \mathbf{r}_i} = 0$$

‘Grad-h’ SPH



- After some nasty maths, we obtain the **SPH momentum equation**

$$\frac{dv_i}{dt} = -\sum_{j=1}^{N} m_j \left(\frac{P_i}{\Omega_i \, \rho_i^2} \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h_i) + \frac{P_j}{\Omega_j \, \rho_j^2} \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h_j) \right)$$

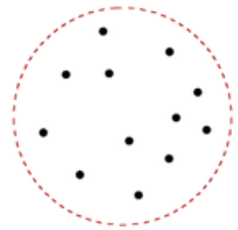
$$\Omega_i = 1 - \frac{\partial h_i}{\partial \rho_i} \sum_{k=1}^{N} m_k \frac{\partial}{\partial h_i} \left(W(\mathbf{r}_i - \mathbf{r}_j, h_i) \right)$$

- This is similar to the original SPH momentum equation, except for
  - The Ω term which takes account of gradients in h, and
  - The two kernel terms now use two different smoothing lengths (so there is no need to symmetrise the kernel as it is done automatically).
- The energy equation is similarly modified by the extra Ω term.

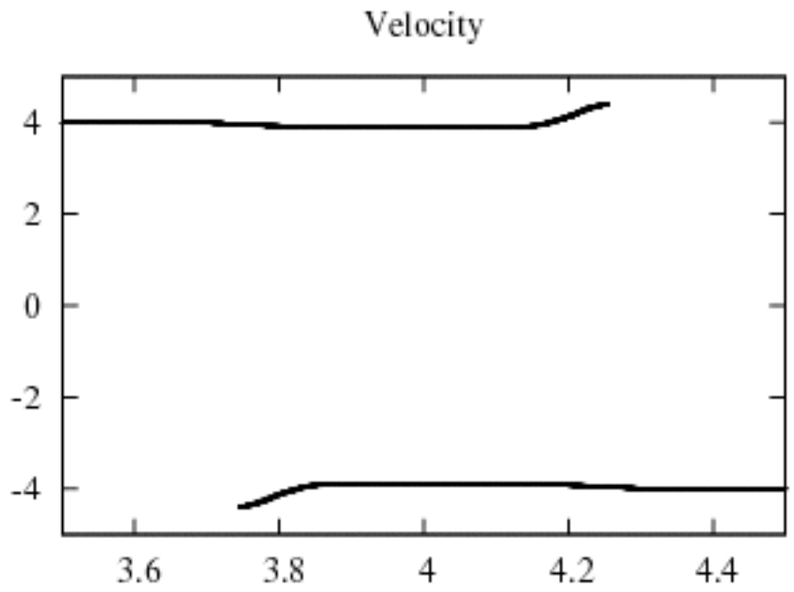
$$\frac{du_i}{dt} = \frac{1}{\Omega_i} \frac{P_i}{\rho_i^2} \sum_{j=1}^{N} m_j (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h_i)$$

- Price & Monaghan (2007) derived the equations of self-gravitating SPH using Lagrangian methods.

A simple shock-tube test

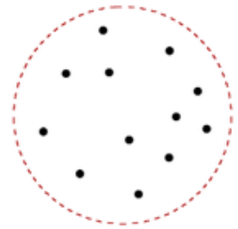


- Here we show a simple test of SPH where two colliding flows form a dense, shocked layer of gas.



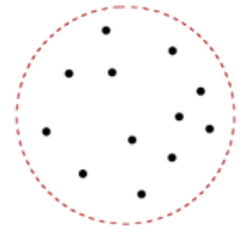
- Oops! We need to include artificial viscosity.

Why do we need artificial viscosity?



- Artificial viscosity is needed in SPH for several reasons
  - The length-scale over which gas molecules are accelerated/decelerated at the shock front is of order a few mean free paths. This is much smaller than the resolution length of hydrodynamical codes and therefore shocks are effectively discontinuities. Since the Euler equations (from which the SPH equations are based on) assume smoothly varying quantities which are differentiable, SPH can break down near shocks. Artificial viscosity works to 'smear out' the discontinuity over a finite range so they can be better modelled by numerical codes.
  - The Euler equations are essentially a low order approximation of the more general Boltzmann equation. The Euler equations assume that the gas is in Local Thermodynamic equilibrium (LTE), i.e. the distribution of velocities at any one point is isotropic and Maxwellian. This may not be the case for strong shocks where the velocity distribution is non-Maxwellian and therefore the Euler equations do not strictly apply.
  - In SPH, artificial viscosity is used to damp high frequency noise in the particle distribution which can build up if left unchecked (e.g. due to finite size of time-step).

Artificial viscosity in SPH



- Lattanzio & Monaghan (1983) proposed a form of artificial viscosity which mimics the properties of the von Neumann-Richtmyer artificial viscosity term.

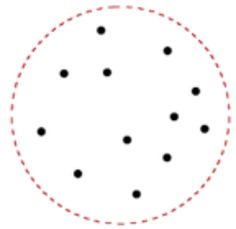
$$\left(\frac{d\mathbf{v}_i}{dt}\right)_{\mathrm{AV}} = -\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h)$$

$$\left(\frac{du_i}{dt}\right)_{\mathrm{AV}} = \frac{1}{2}\sum_{j=1}^{N} m_j \Pi_{ij} (\mathbf{v}_i - \mathbf{v}_j) \cdot \nabla_i W(\mathbf{r}_i - \mathbf{r}_j, h)$$

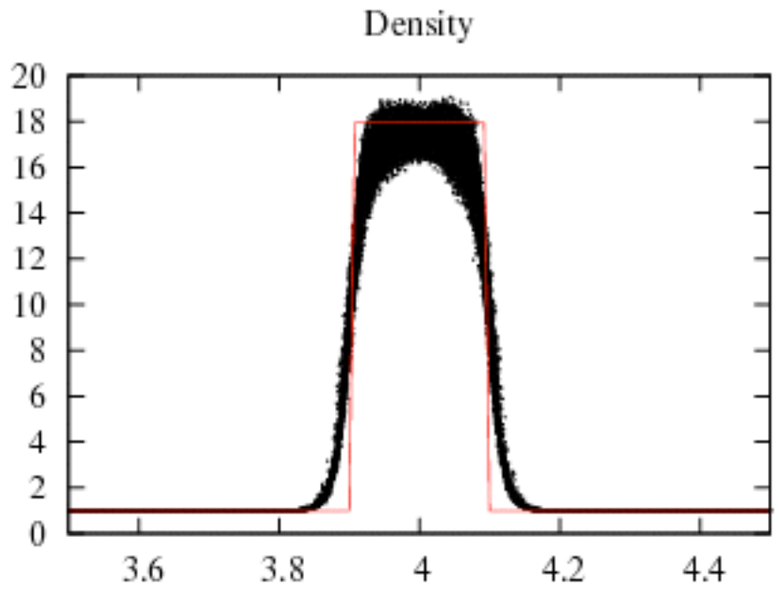
where $\Pi_{ij} = \dfrac{-\alpha c_i j \mu_{ij} + \beta \mu_{ij}^2}{\overline{\rho_{ij}}}$ and $\mu_{ij} = \dfrac{h\,(\mathbf{v}_i - \mathbf{v}_j)\cdot(\mathbf{r}_i - \mathbf{r}_j)}{\mathbf{r}_{ij}^2 + \eta^2}$

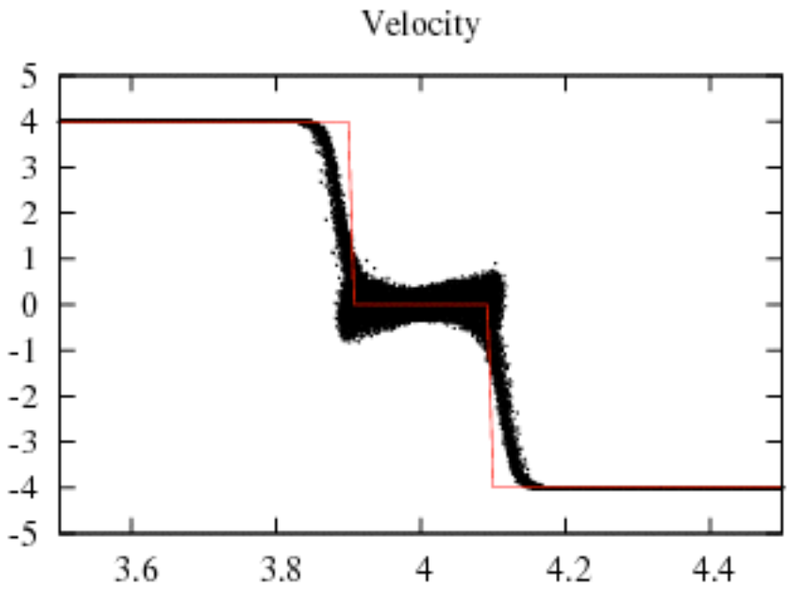
- The first α term is used for sub-sonic shocks and can dampen post-shock oscillations.
- The second β term is used to model high Mach number shocks where particle penetration may occur.
- (Note : Monaghan (1997) suggested a new form of artificial viscosity based on Riemann solvers which is now commonly used).

Artificial viscosity in SPH

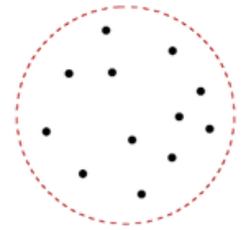


- The particles are now slowed down to rest at the shock front and reproduce the analytical results well with artificial viscosity.





Integration methods



- The motions of particles in SPH can be integrated in a similar manner to N-body codes.
  - 2nd-order Runge-Kutta (easy to implement, error control)
  - 2nd-order Leap-frog (symplectic, better conservation properties, only one force calc. per time-step)
  - 2nd-order predictor-corrector (only force calc. per time-step)

- The ideal time-step for an SPH particle is taken as the minimum of two separate criteria

$$\Delta t_1 = \gamma \frac{h}{c_s + h|\nabla \cdot \mathbf{v}|}$$

$$\Delta t_2 = \eta \sqrt{\frac{h}{|a|}}$$

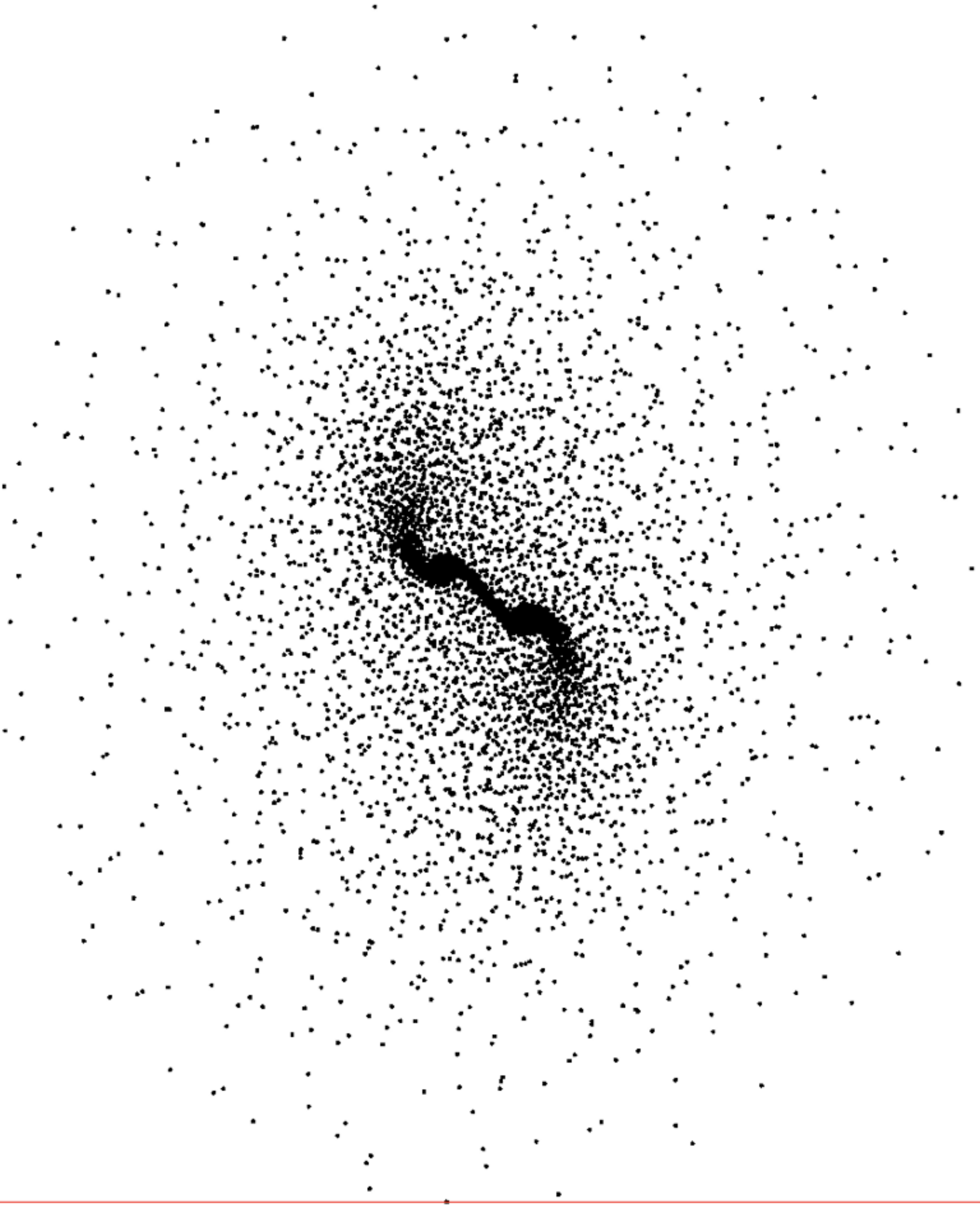
  - A *Courant-Friedrichs-Lewy*-like time-step condition to ensure the integration routine is stable
  - An N-body-like condition used as an extra safety measure, and also for scenarios where other forces (e.g. gravitational) dominate over the pressure force and thus require a shorter time-step

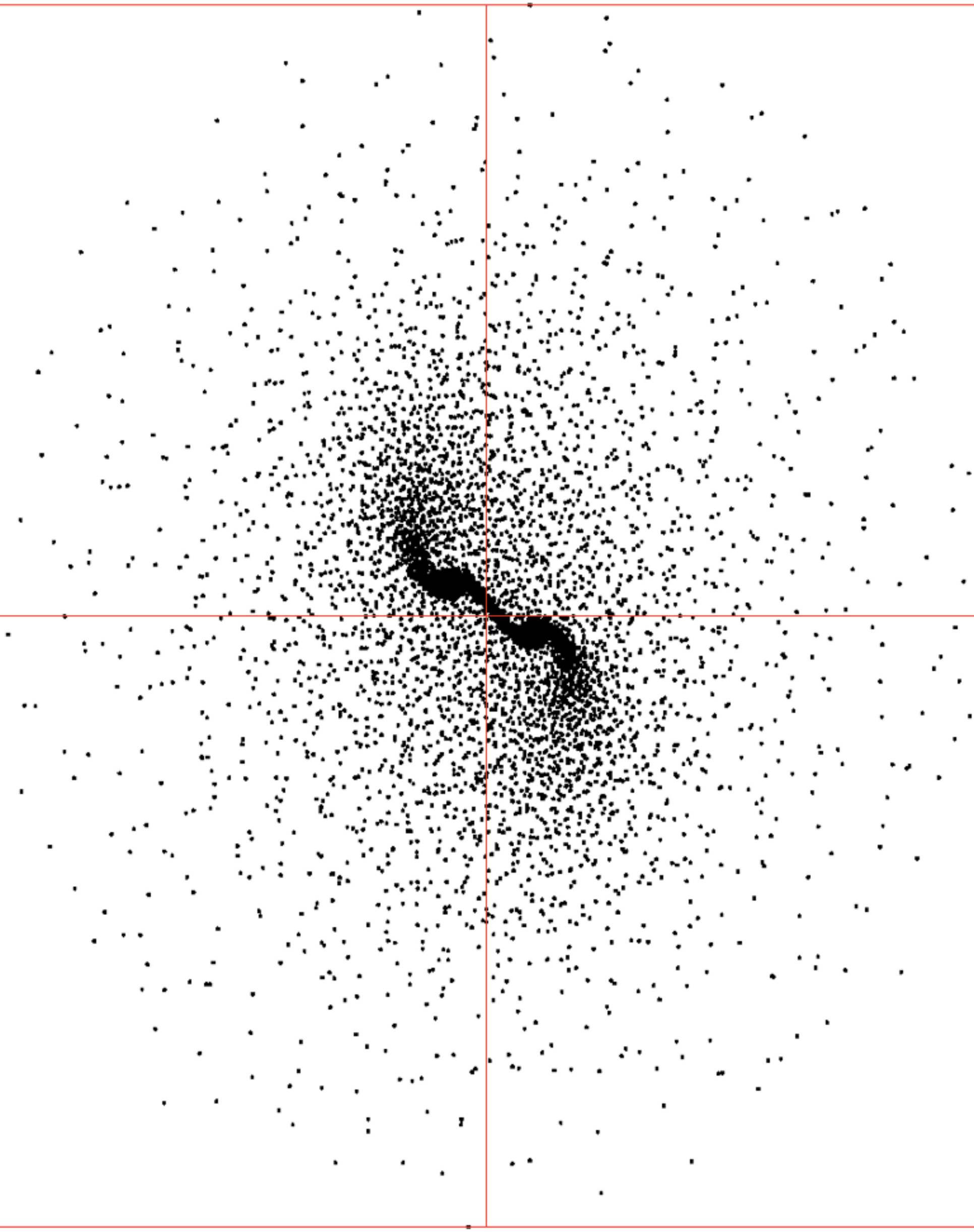
Trees

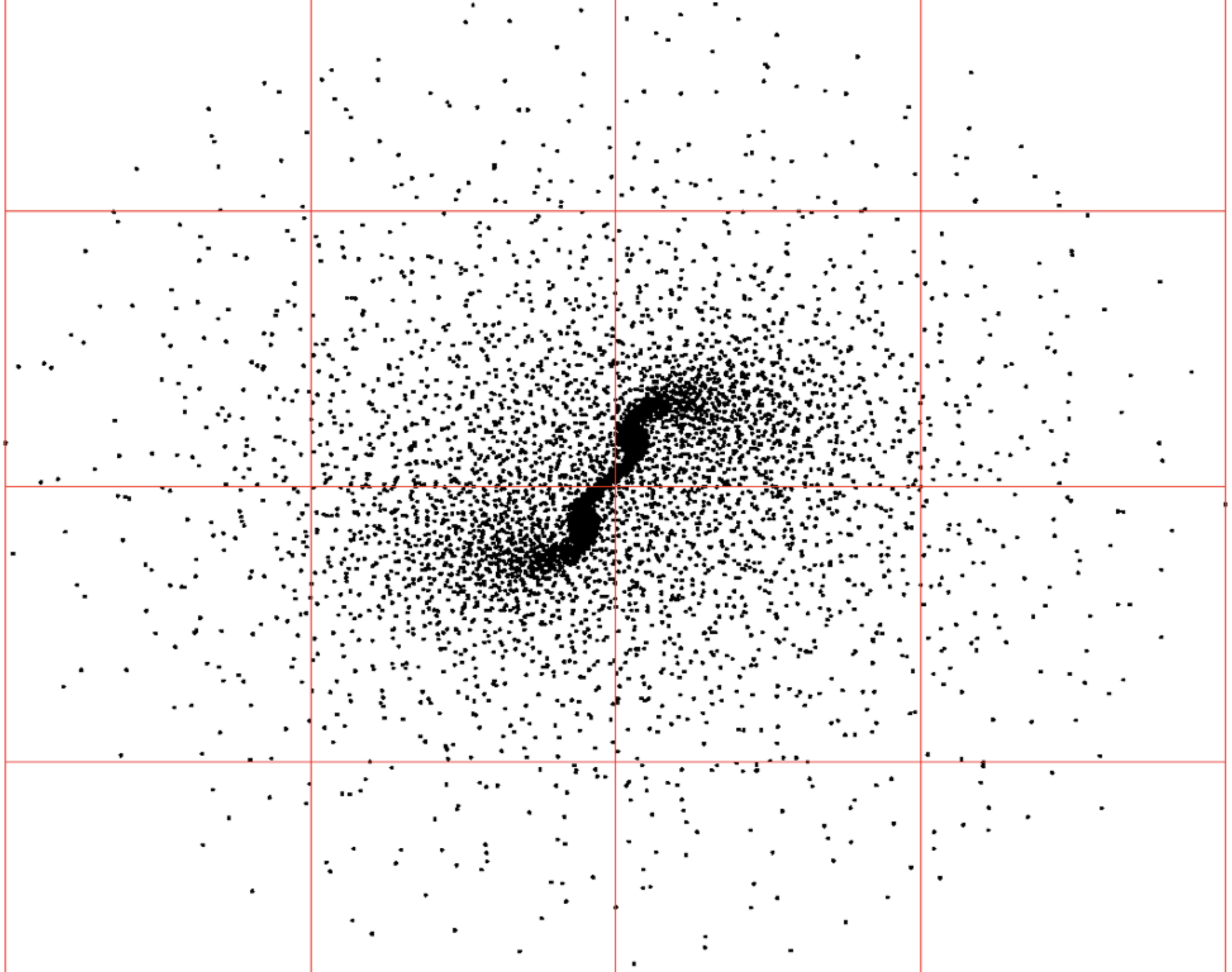
- Trees are data structures used to efficiently search a large data set for only certain entries by grouping the data in a systematic, hierarchical manner.

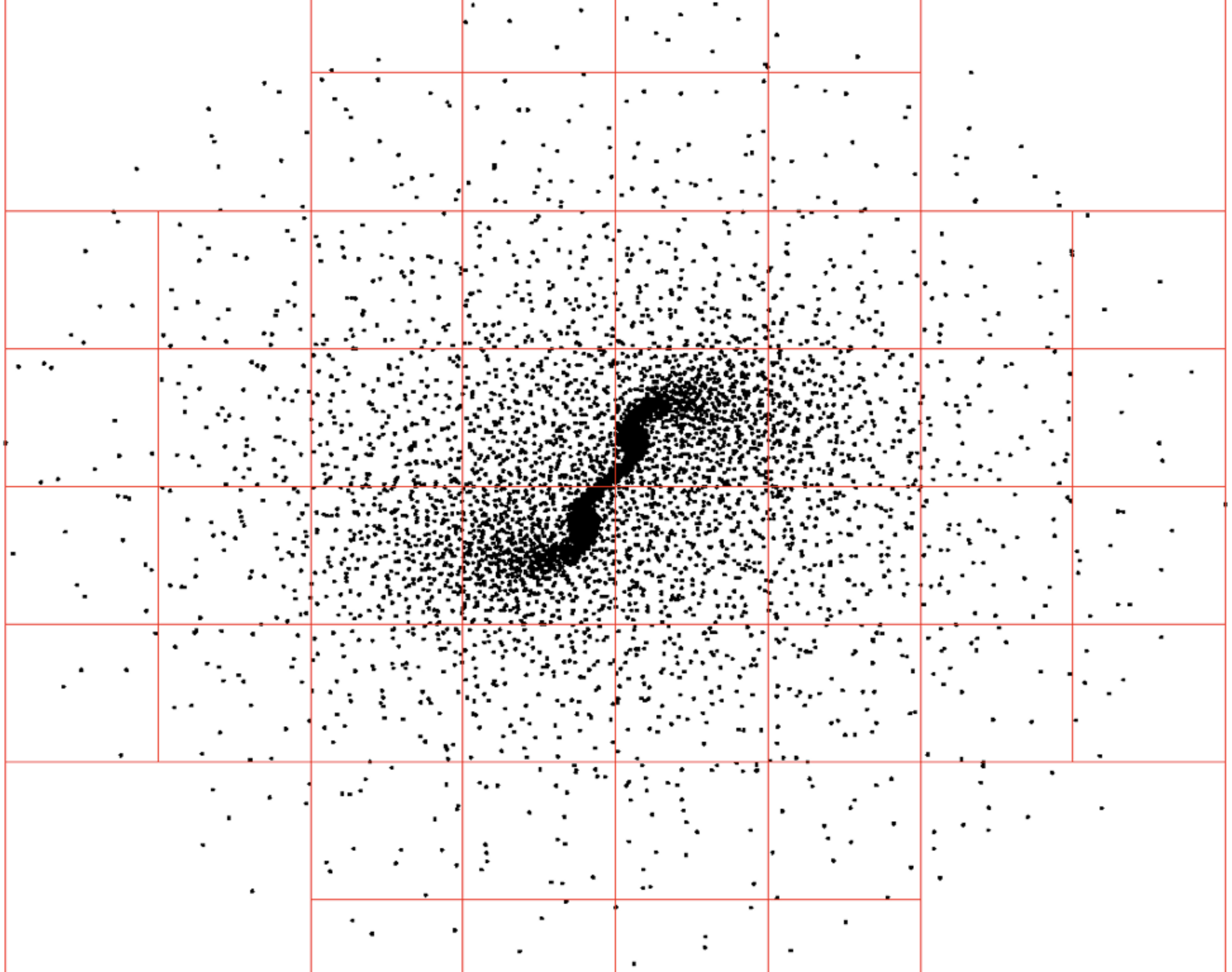
- In SPH, trees are used primarily to
    - Find neighbouring particles for SPH interactions
    - Efficiently calculate the gravitational force

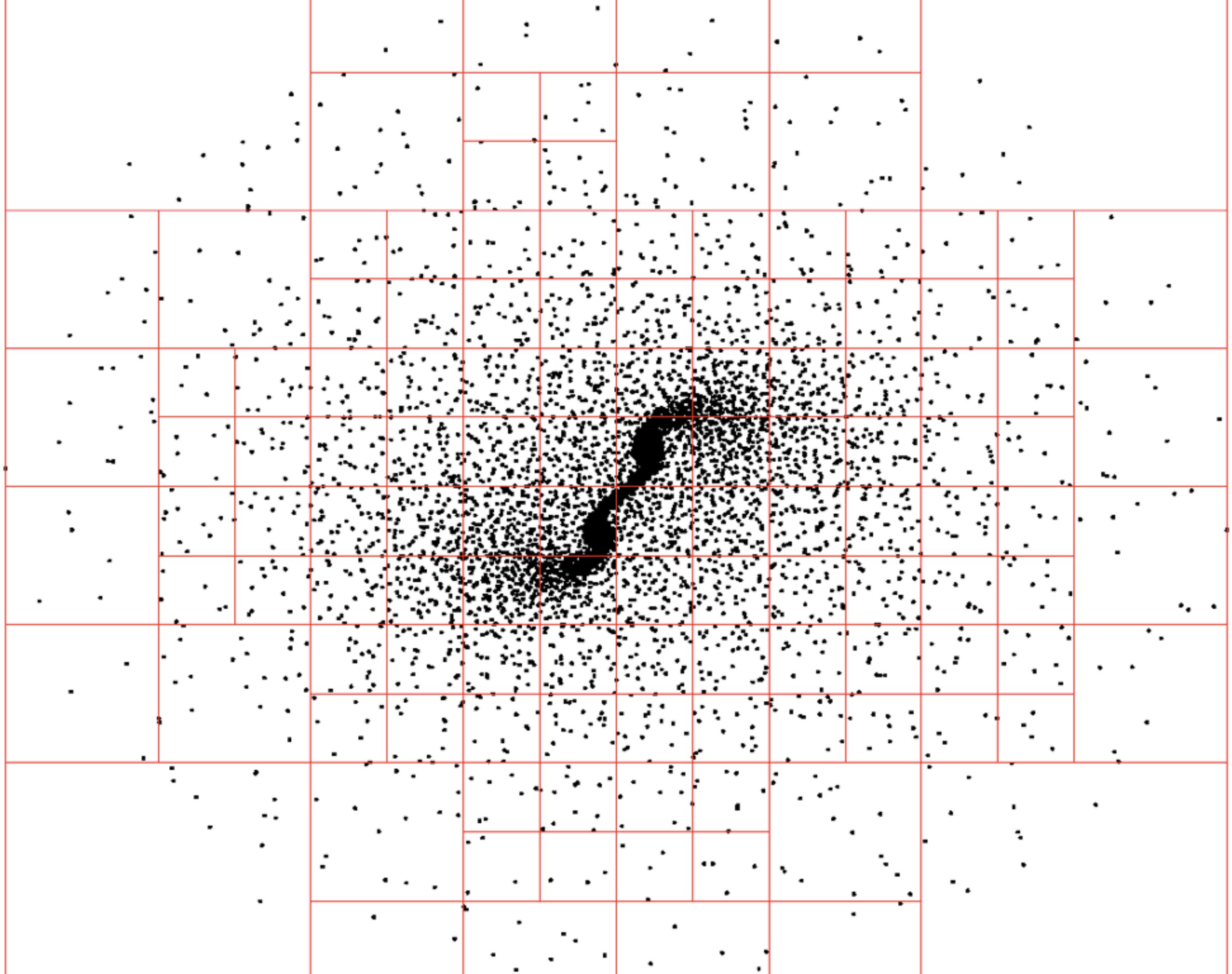
- There are a variety of different methods available when creating trees for SPH
    - Octal spatial decomposition tree (Barnes & Hut 1986)
    - Nearest-neighbour (bottom-up) binary tree
    - Balanced binary-number tree

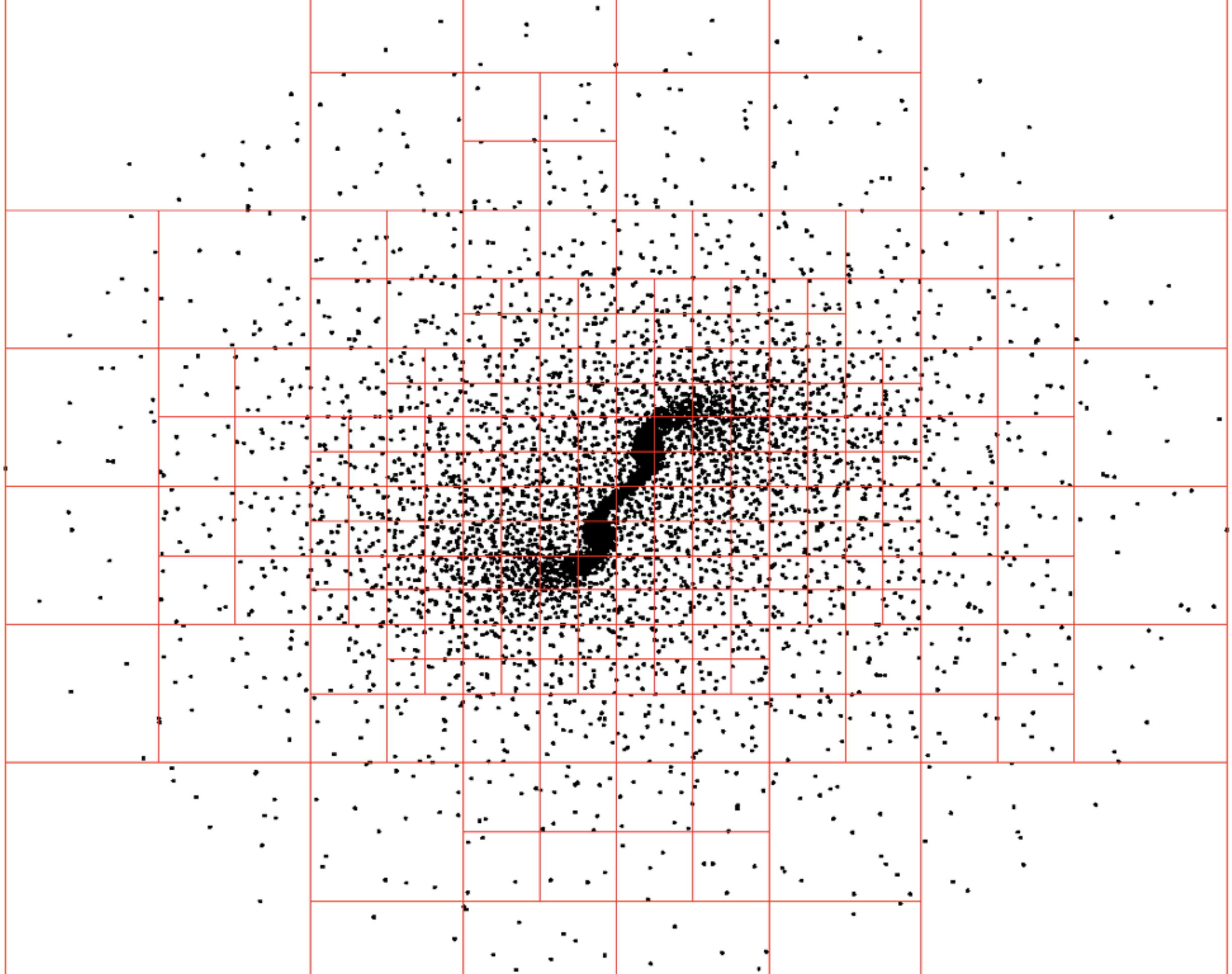


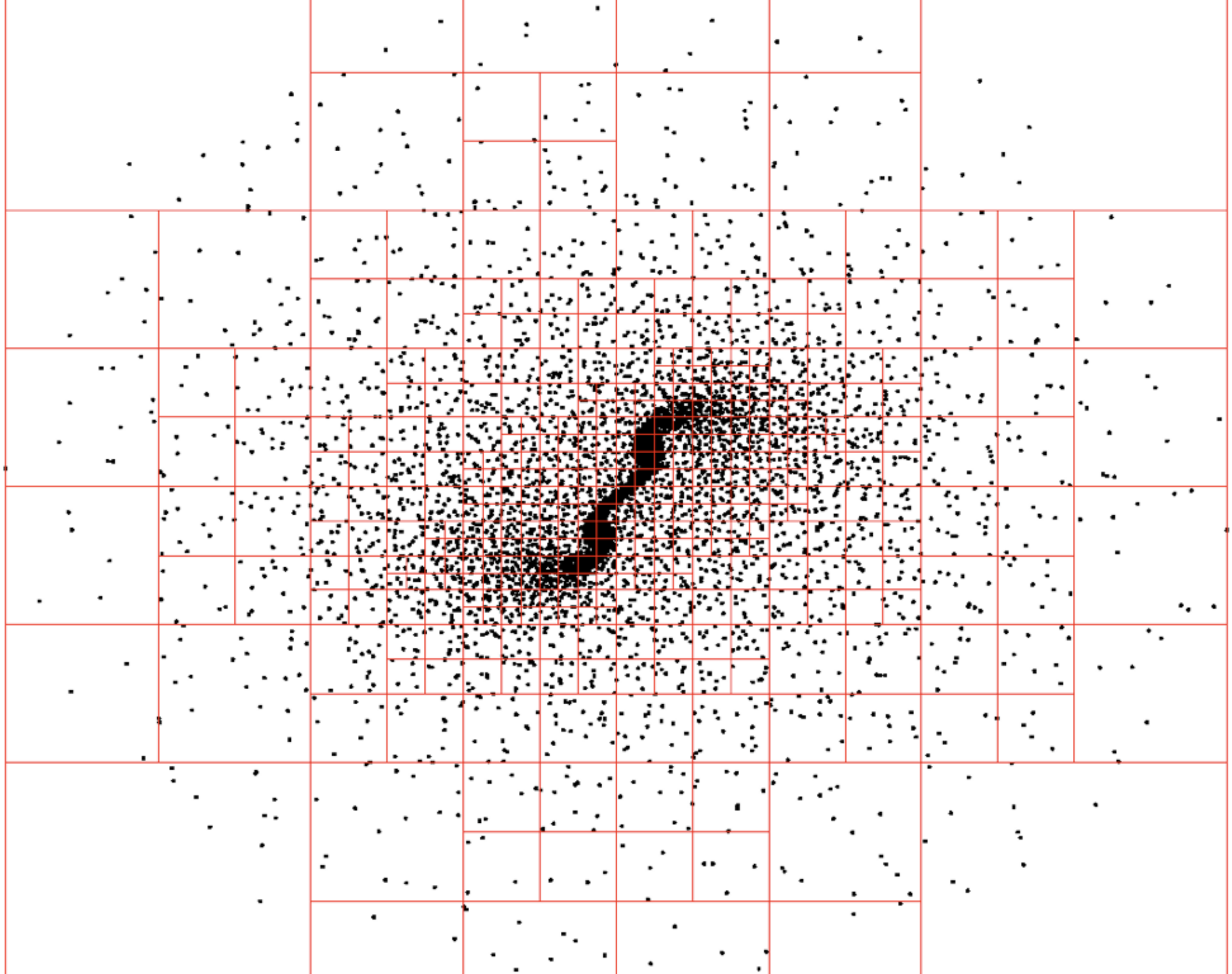


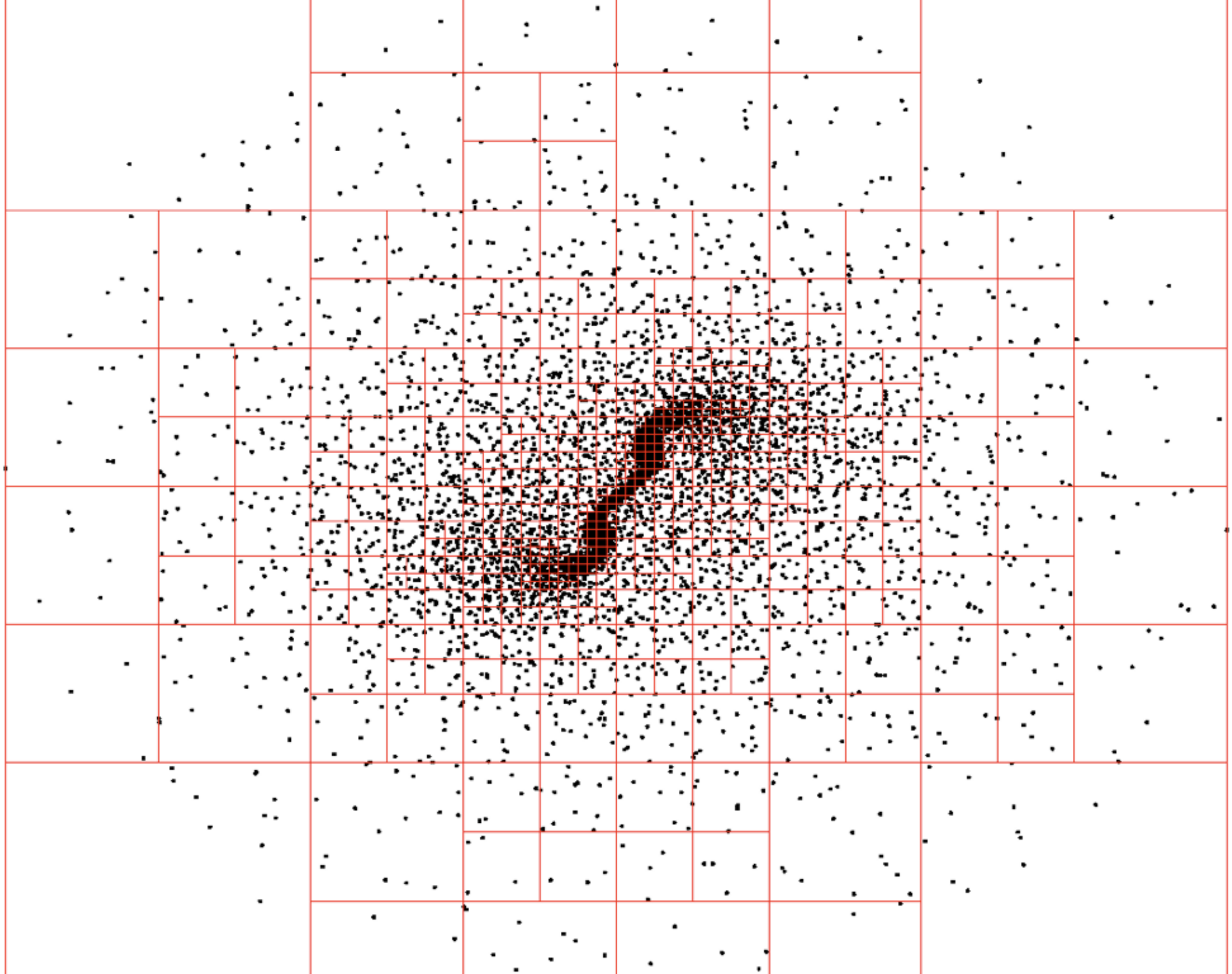


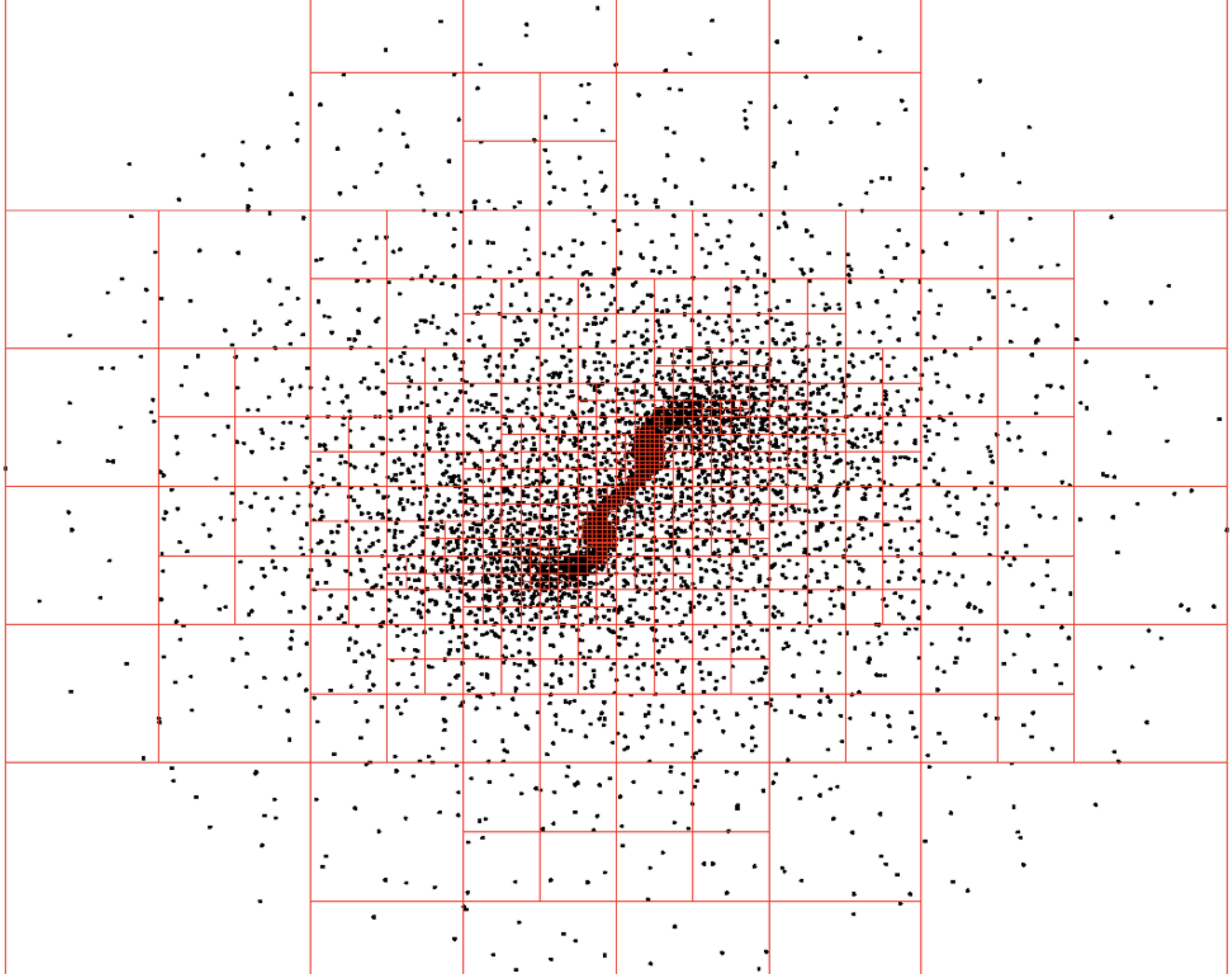






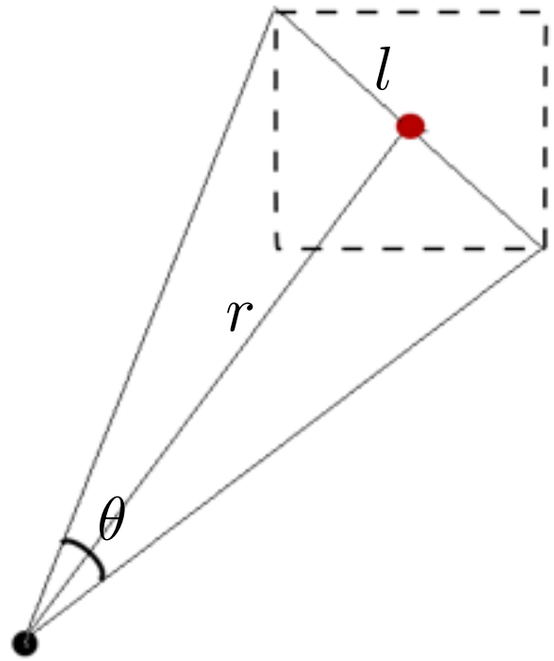






Tree gravity

- Calculating the gravitational force is typically the most expensive part (in CPU time) of an SPH simulation (without radiative transfer)
- Using a tree can transform the gravitational calculation from a $\mathcal{O}(N^2)$ operation to a $\mathcal{O}(N \log N)$ operation
- The gravitational force for a particle using the tree is calculated by walking the tree (starting the root cell) and then interrogating each cell to see if we can approximate its contribution to the gravitational force by the its ficticious centre-of-mass particle.

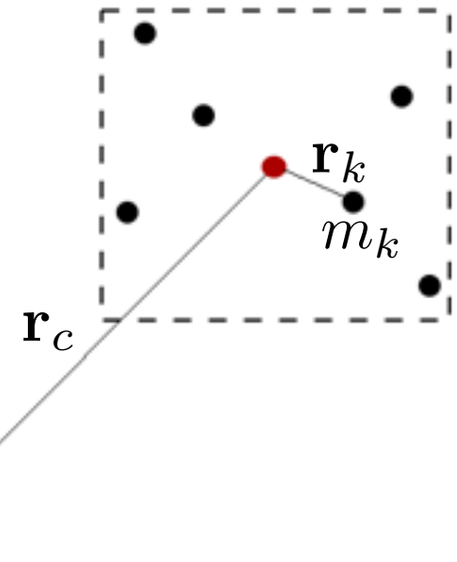


- The original criterion (Barnes & Hut 1986) was to accept the centre-of-mass approximation if the angle subtended by the cell (to the particle) is less than some user-defined maximum

$$\theta \equiv \frac{l}{r} < \theta_{\mathrm{MAX}}$$

Multipole approximation

- We can increase the performance of the BH tree by including higher-order multipole moments in the calculation of the gravitational force

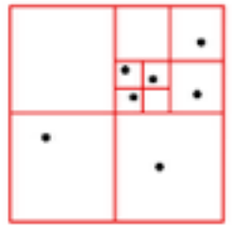


- The first four terms in the multipole expansion are

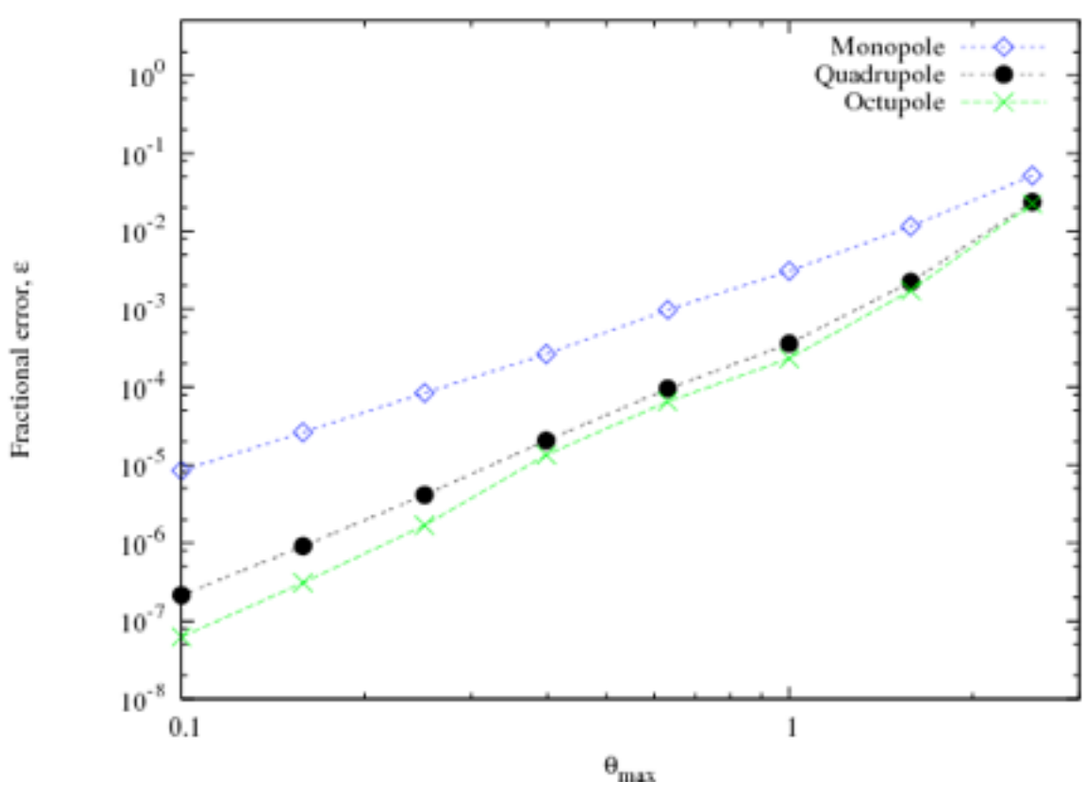
$$\mathbf{a}_{\text{GRAV}} = \underbrace{\mathbf{a}_{\text{MONOPOLE}}}_{\frac{\sum m_k}{r_c^2}} + \underbrace{\cancel{\mathbf{a}_{\text{DIPOLE}}}}_{\frac{\sum m_k r_k}{r_c^3}} + \underbrace{\mathbf{a}_{\text{QUADRUPOLE}}}_{\frac{\sum m_k r_k^2}{r_c^4}} + \underbrace{\mathbf{a}_{\text{OCTUPOLE}}}_{\frac{\sum m_k r_k^3}{r_c^5}} + \ldots$$

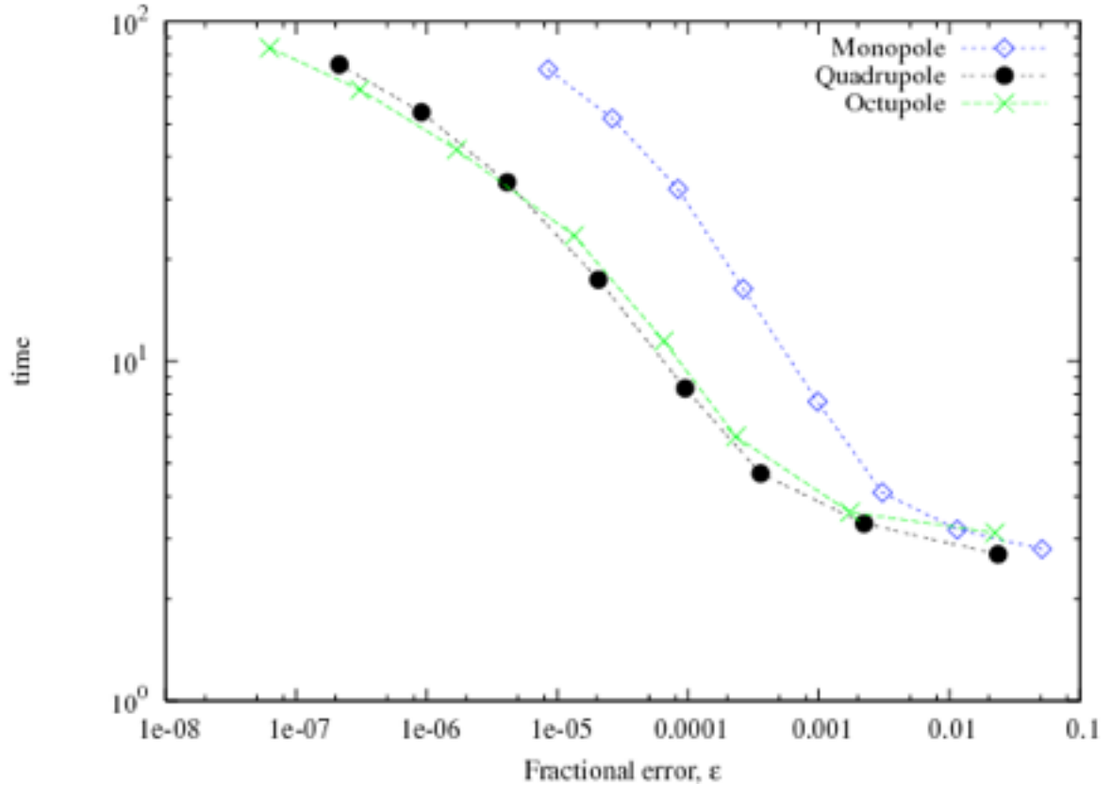
- The more multipole moment terms used, the more accurate the centre-of-mass approximation of a cell, but at the cost of extra cpu time and memory. The decision of how many terms to include becomes a compromise between accuracy, speed and memory.

Multipole approximation



- To investigate what is the optimal multipole expansion, we calculate the mean force errors for some particular geometry of particles for monopole, quadrupole and octupole expansions and compare the errors as a function of cpu time.





- Suggests that the optimal expansion is to quadrupole order.

Multipole acceptance criteria

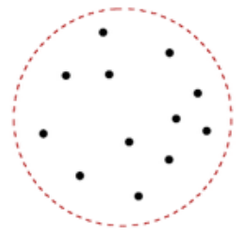
- Salmon & Warren (1994) reported that the simple geometric criterion used by Barnes & Hut (1986) can lead to catastrophic results for certain geometries if too high an opening angle is selected.
- They devised new opening criteria based on higher-order multipole terms : Multipole acceptance criteria (MAC)
- In their criterion, the multipole expansion can only be used for a cell if the leading error term in the expansion is smaller than some (user-defined) fraction of the total gravitational acceleration of the particle.

$$\frac{G \sum m_k r_k^M}{r_c^{M+2}} < \alpha_{\mathrm{TOL}} \, |\mathbf{a}_{\mathrm{GRAV}}|$$

where *M = 0* for monopoles, *M = 2* for quadrupoles and *M = 3* for octupole.

- Note : since we cannot possibly know the gravitational acceleration at the current time-step before we calculate it, we can usually use the previous time-steps acceleration as a good-enough estimate.

Modern improvements to SPH



- There exists a variety of formulations of SPH present in the literature
  - Pressure-Entropy SPH (Saitoh & Makino 2012/2013)
  - Improved artificial viscosity formulations (e.g. Cullen & Dehnen 2010)
- These improvements do indeed improve the accuracy of SPH
  - However, there is a cost in the longer CPU run times
  - Also the newer algorithms are more difficult to implement
- Whether you need these or not may depend largely on the astrophysics problem you are investigating!