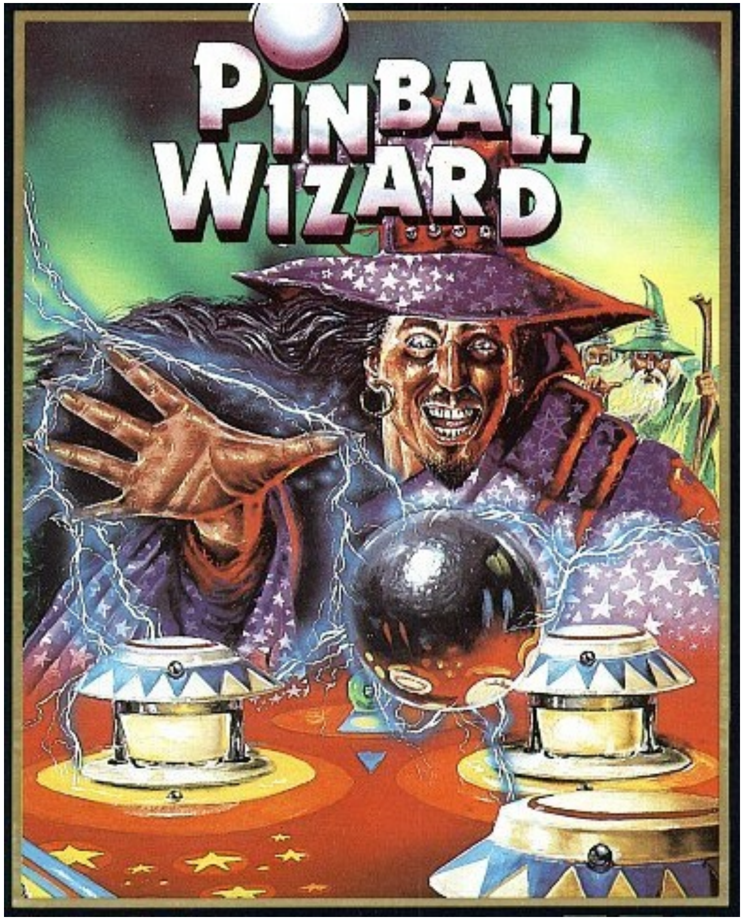
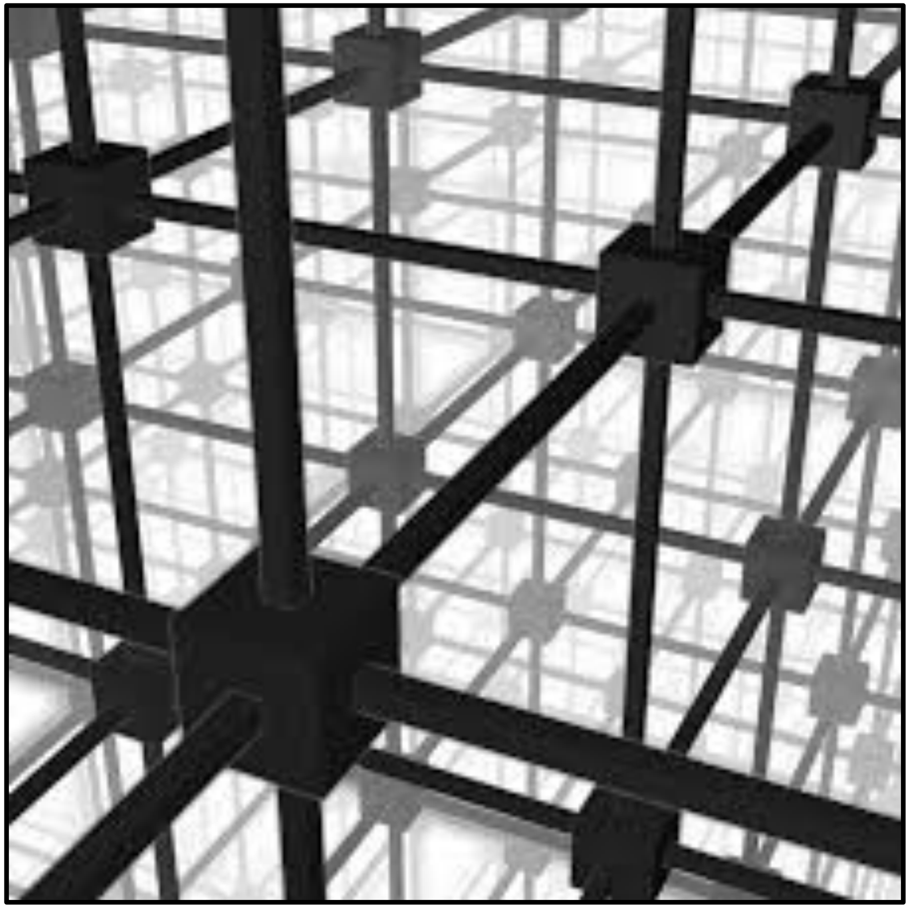
Alexander Knebe, *Universidad Autonoma de Madrid*



VS



Solving for Gravity

- full set of equations

• collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$

$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

• Poisson's equation

$$\Delta\phi = 4\pi G\rho_{tot}$$

• collisional matter (e.g. gas)

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) = 0$$

$$\frac{\partial\left(\rho\vec{v}\right)}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\overleftrightarrow{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ \left(-\nabla\phi\right)$$

$$\frac{\partial\left(\rho E\right)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right)$$

• ideal gas equations

$$p = \left(\gamma - 1\right)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

• Maxwell's equation

$$\frac{\partial\vec{B}}{\partial t} = -\nabla\times\left(\vec{v}\times\vec{B}\right)$$

- full set of equations

  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$
$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

leap-frog integration

  - Poisson's equation

$$\Delta\phi = 4\pi G\rho_{tot}$$

  - collisional matter (e.g. gas)

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) = 0$$

$$\frac{\partial\left(\rho\vec{v}\right)}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\overleftrightarrow{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ \left(-\nabla\phi\right)$$

$$\frac{\partial\left(\rho E\right)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right)$$

  - ideal gas equations

$$p = \left(\gamma - 1\right)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

  - Maxwell's equation

$$\frac{\partial\vec{B}}{\partial t} = -\nabla\times\left(\vec{v}\times\vec{B}\right)$$

Solving for Gravity

- full set of equations

- collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$

$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

- Poisson's equation

$$\Delta\phi = 4\pi G\rho_{tot}$$

- collisional matter (e.g. gas)

later...

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v}) = 0$$

$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\overleftrightarrow{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ (-\nabla\phi)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot(-\nabla\phi) + (\Gamma - L)$$

hyperbolic partial differential equations:
solutions are wave-like, i.e. perturbations need time to travel...

- ideal gas equations

$$p = (\gamma - 1)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

- Maxwell's equation

$$\frac{\partial\vec{B}}{\partial t} = -\nabla\times\left(\vec{v}\times\vec{B}\right)$$

Solving for Gravity

- full set of equations

  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$

$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

  - Poisson's equation

$$\Delta\phi = 4\pi G\rho_{tot}$$

  - collisional matter (e.g. gas)

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) = 0$$

$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\overleftrightarrow{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ \left(-\nabla\phi\right)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right)$$

  - ideal gas equations

$$p = \left(\gamma - 1\right)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

electrodynamics lecture...

  - Maxwell's equation

$$\frac{\partial\vec{B}}{\partial t} = -\nabla\times\left(\vec{v}\times\vec{B}\right)$$

- full set of equations

  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$
$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

  - Poisson's equation

$$\Delta\phi = 4\pi G\rho_{tot}$$

  - collisional matter (e.g. gas)

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v}) = 0$$
$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\overleftrightarrow{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ (-\nabla\phi)$$
$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot(-\nabla\phi) + (\Gamma - L)$$

physics lecture...

  - ideal gas equations

$$p = (\gamma - 1)\rho\varepsilon$$
$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

  - Maxwell's equation

$$\frac{\partial\vec{B}}{\partial t} = -\nabla\times\left(\vec{v}\times\vec{B}\right)$$

- full set of equations

  - collisionless matter (e.g. dark matter)

$$\frac{d\vec{x}_{DM}}{dt} = \vec{v}_{DM}$$

$$\frac{d\vec{v}_{DM}}{dt} = -\nabla\phi$$

now!

- Poisson's equation

$$\Delta\phi = 4\pi G\rho_{tot}$$

elliptical partial differential equation:
solution obtainable via FFT (for constant coefficients)

  - collisional matter (e.g. gas)

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\left(\rho\vec{v}\right) = 0$$

$$\frac{\partial(\rho\vec{v})}{\partial t} + \nabla\cdot\left(\rho\vec{v}\otimes\vec{v} + \left(p + \frac{1}{2\mu}B^2\right)\overleftrightarrow{1} - \frac{1}{\mu}\vec{B}\otimes\vec{B}\right) = \rho\ \left(-\nabla\phi\right)$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot\left(\left[\rho E + p + \frac{1}{2\mu}B^2\right]\vec{v} - \frac{1}{\mu}\left[\vec{v}\cdot\vec{B}\right]\vec{B}\right) = \rho\vec{v}\cdot\left(-\nabla\phi\right) + \left(\Gamma - L\right)$$

- ideal gas equations

$$p = \left(\gamma - 1\right)\rho\varepsilon$$

$$\rho\varepsilon = \rho E - \frac{1}{2}\rho v^2$$

- Maxwell's equation

$$\frac{\partial\vec{B}}{\partial t} = -\nabla\times\left(\vec{v}\times\vec{B}\right)$$

- Poisson's equation vs. Poisson's integral
- tree codes
- PM codes

- **Poisson's equation vs. Poisson's integral**
- tree codes
- PM codes

Solving for Gravity

- Poisson's equation

$$\Delta\Phi(\vec{x}) = 4\pi G\rho(\vec{x})$$

Solving for Gravity

- Poisson's equation

$$\Delta\Phi(\vec{x}) = 4\pi G\rho(\vec{x})$$

- Poisson's integral

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\,\rho(\vec{x}')d^3x' \quad ; \quad \mathcal{G}(\vec{x}) = \frac{1}{4\pi\,x}$$

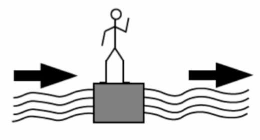
proof: see pages 18-35 of http://popia.ft.uam.es/aknebe/page3/files/ComputationalAstrophysics/Gravity01-TreeCodes.pdf

Solving for Gravity

- Poisson's equation

$$\Delta\Phi(\vec{x}) = 4\pi G\rho(\vec{x})$$

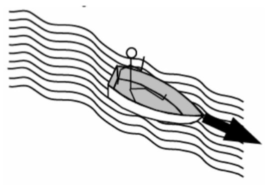
grid approach



- Poisson's integral

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\ \rho(\vec{x}')d^3x' \quad ; \quad \mathcal{G}(\vec{x}) = \frac{1}{4\pi\, x}$$

particle approach



- Poisson's equation

$$\Delta\Phi(\vec{x}) = 4\pi G\rho(\vec{x})$$

<u>grid approach</u>

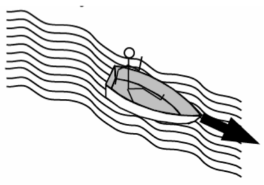
$$\Delta\Phi(\vec{r}_{i,j,k}) = 4\pi G\rho(\vec{r}_{i,j,k})$$

($\vec{r}_{i,j,k}$ =position of centre of grid cell (i,j,k))

- Poisson's integral

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\ \rho(\vec{x}')d^3x' \quad ; \quad \mathcal{G}(\vec{x}) = \frac{1}{4\pi\, x}$$

<u>particle approach</u>



Solving for Gravity

- Poisson's equation

$$\Delta\Phi(\vec{x}) = 4\pi G\rho(\vec{x})$$

<u>grid approach</u>

$$\Delta\Phi(\vec{r}_{i,j,k}) = 4\pi G\rho(\vec{r}_{i,j,k})$$

($\vec{r}_{i,j,k}$ =position of centre of grid cell (i,j,k))

- Poisson's integral

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\ \rho(\vec{x}')d^3x' \quad ; \quad \mathcal{G}(\vec{x}) = \frac{1}{4\pi\, x}$$

$$\rho(\vec{r}) = \sum_{i=1}^{N} m_i \delta_{\text{Dirac}}(\vec{r}-\vec{r}_i)$$

<u>particle approach</u>

$$\vec{F}(\vec{r}_i) = -\sum_{i\neq j} \frac{G m_i m_j}{(r_i - r_j)^3}(\vec{r}_i - \vec{r}_j)$$

Solving for Gravity

- Poisson's equation

$$\Delta\Phi(\vec{x}) = 4\pi G\rho(\vec{x})$$

grid approach

$$\Delta\Phi(\vec{r}_{i,j,k}) = 4\pi G\rho(\vec{r}_{i,j,k})$$

($\vec{r}_{i,j,k}$ =position of centre of grid cell (i,j,k))

weapon of choice: AMR codes

- Poisson's integral

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\,\rho(\vec{x}')d^3x' \quad ; \quad \mathcal{G}(\vec{x}) = \frac{1}{4\pi x}$$

weapon of choice: tree codes

particle approach

$$\vec{F}(\vec{r}_i) = -\sum_{i\neq j}\frac{Gm_i m_j}{(r_i - r_j)^3}(\vec{r}_i - \vec{r}_j)$$

- Poisson’s equation vs. Poisson’s integral
- **tree codes**
- PM codes

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

✓ advantage: easy to code

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

$$N \times N = N^2$$

✓ advantage: easy to code

↯ drawback: extremely time consuming (N^2 operations)

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

$$N \times N = N^2$$

overcoming the "N^2" issue?!

✓ advantage: easy to code

drawback: extremely time consuming (N^2 operations)

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

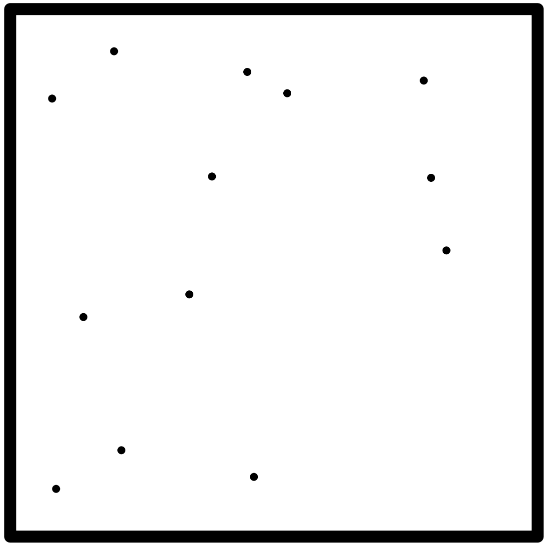
organizing particles into a "tree structure" will give $N \log(N)$ operations

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

  - generating the tree:

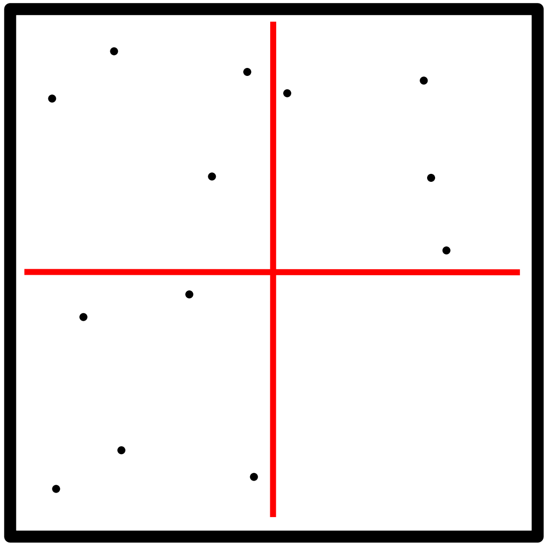


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

  - generating the tree:

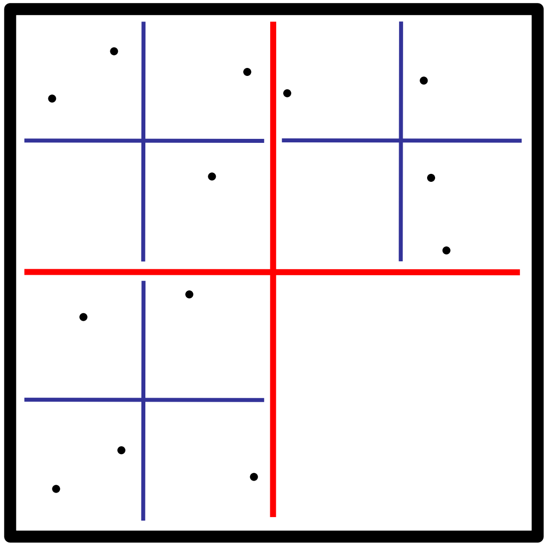


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{Gm_i m_j}{(r_i - r_j)^3}(\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

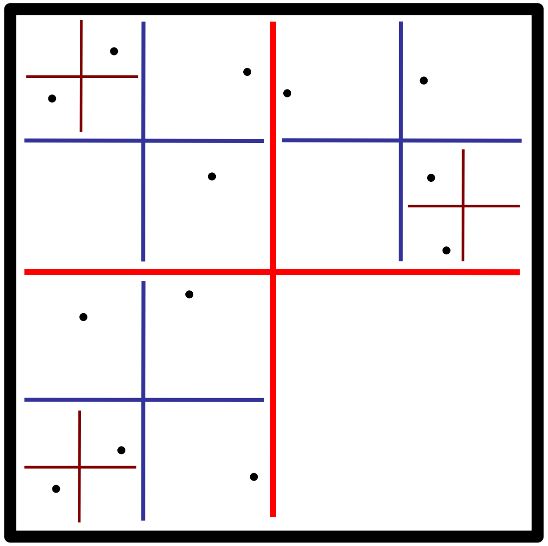


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

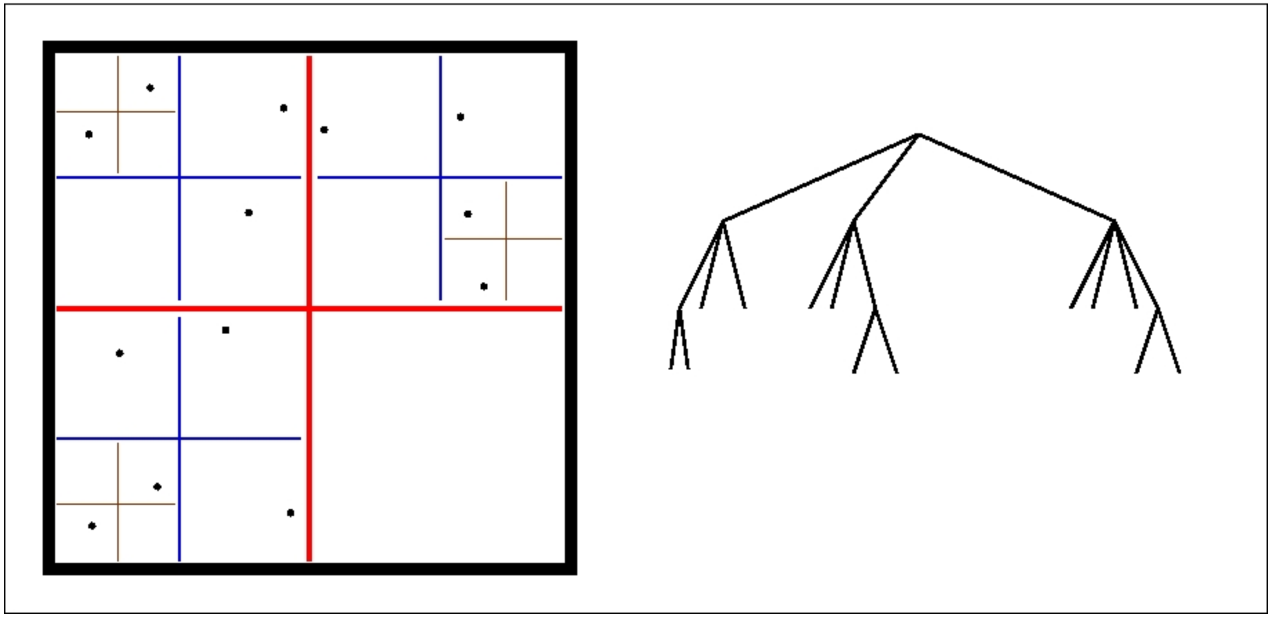


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

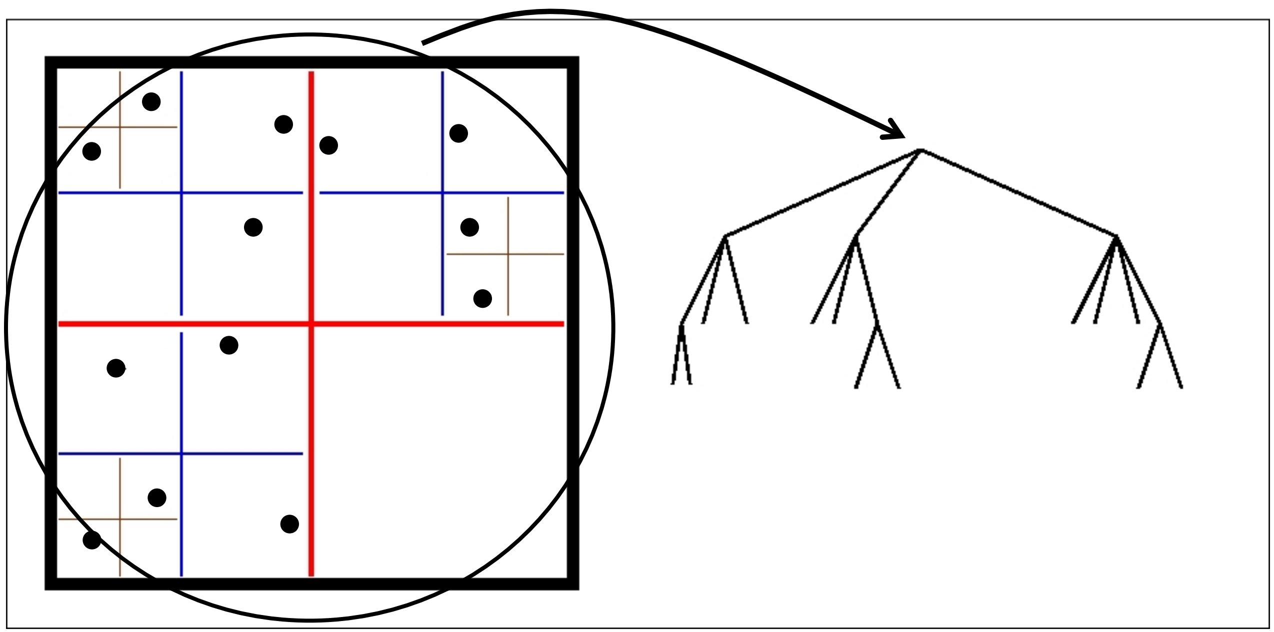


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

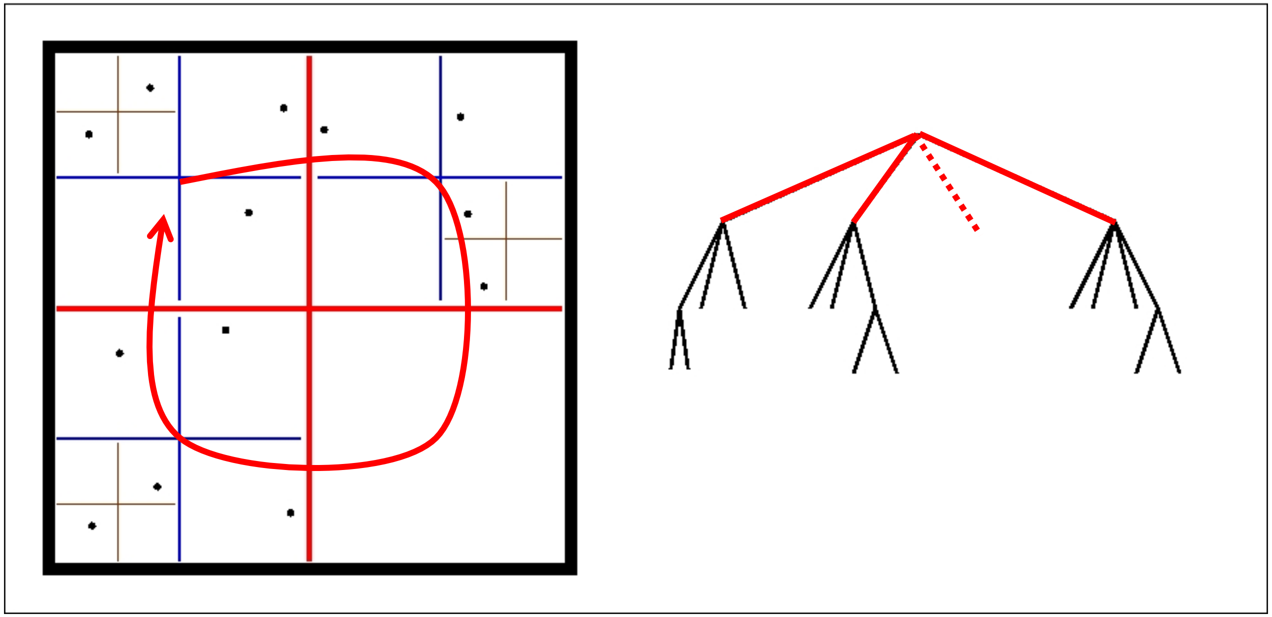


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

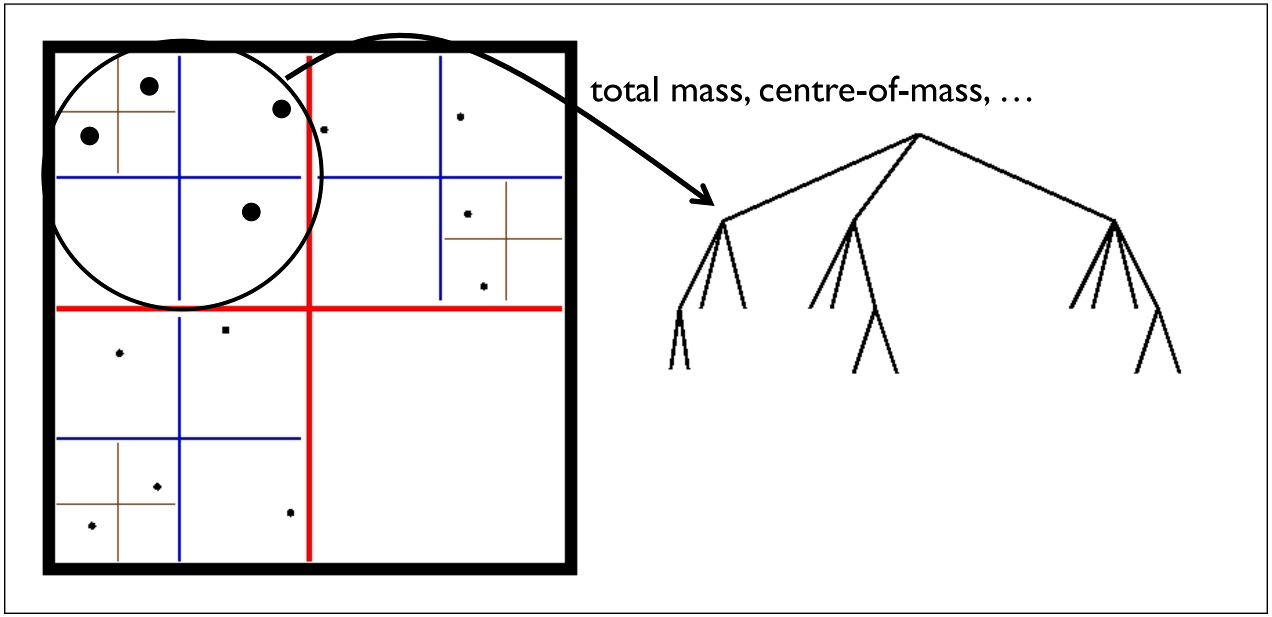


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

  - generating the tree:

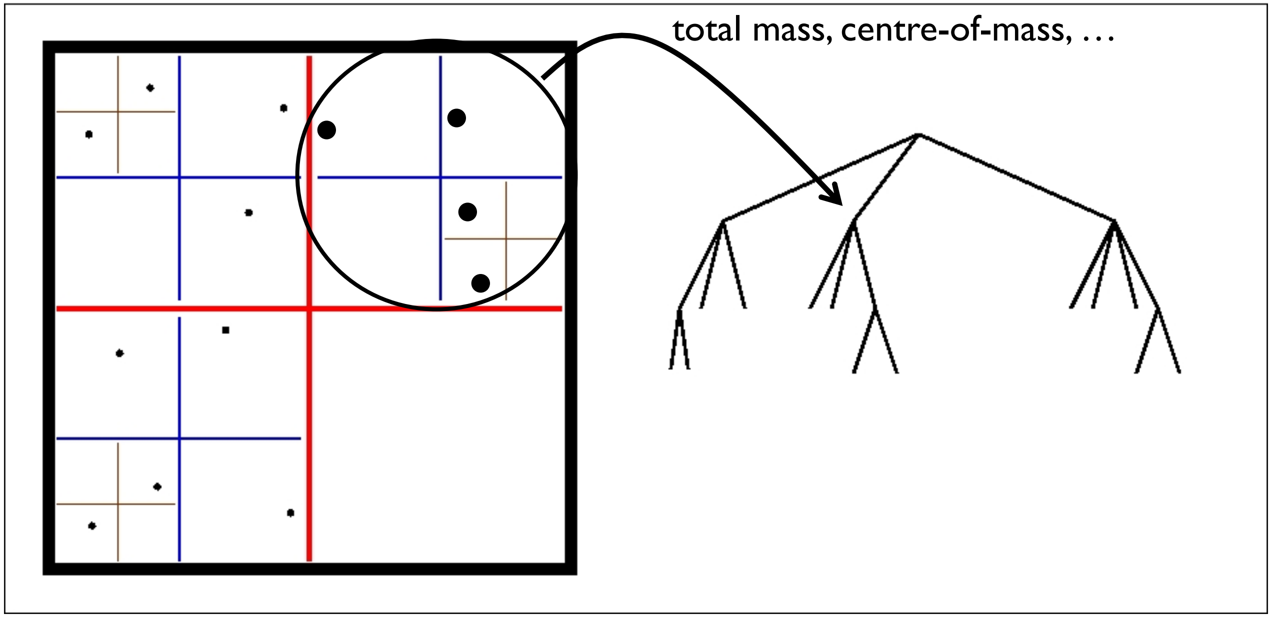


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

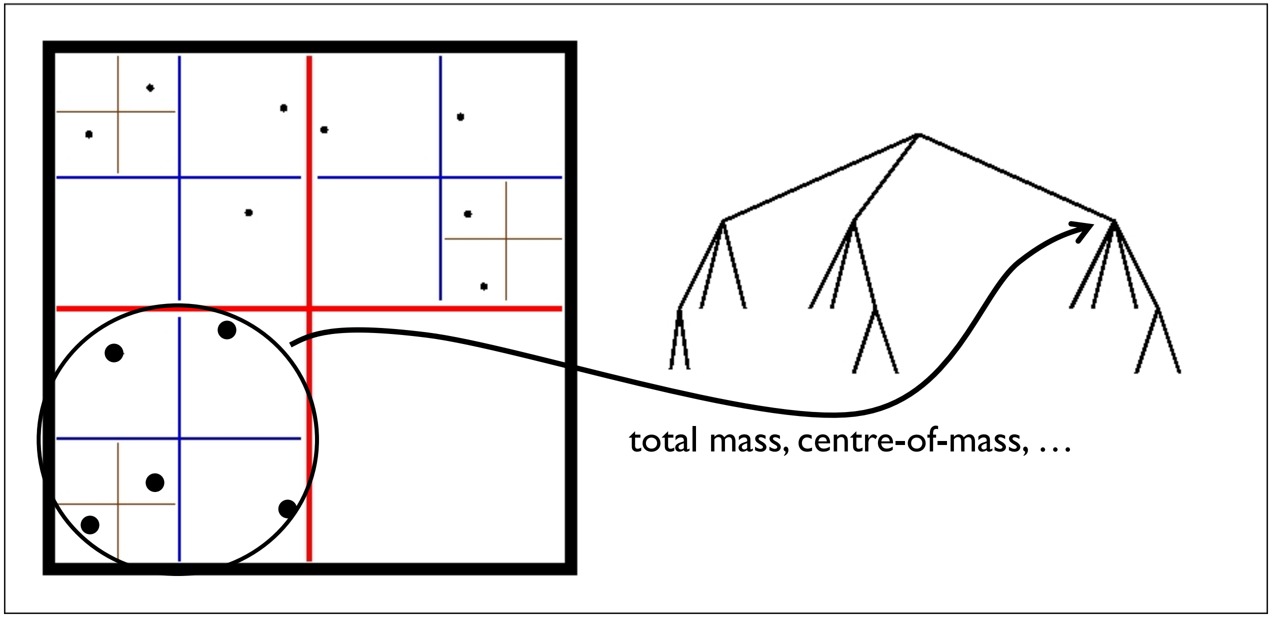


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

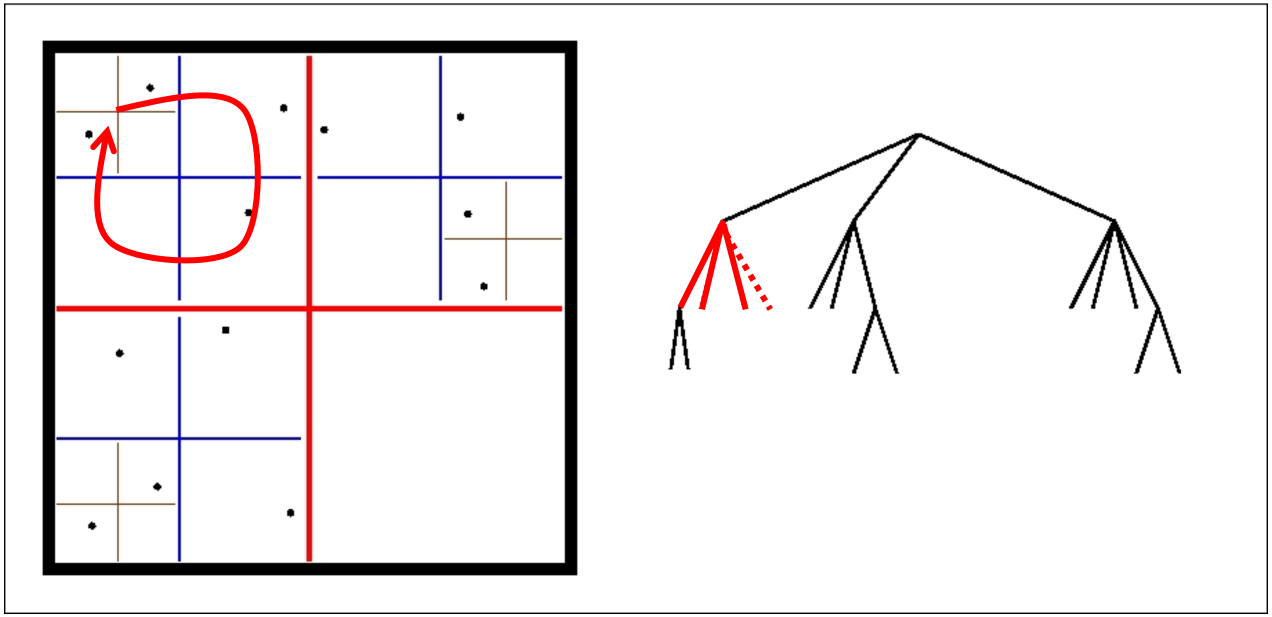


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

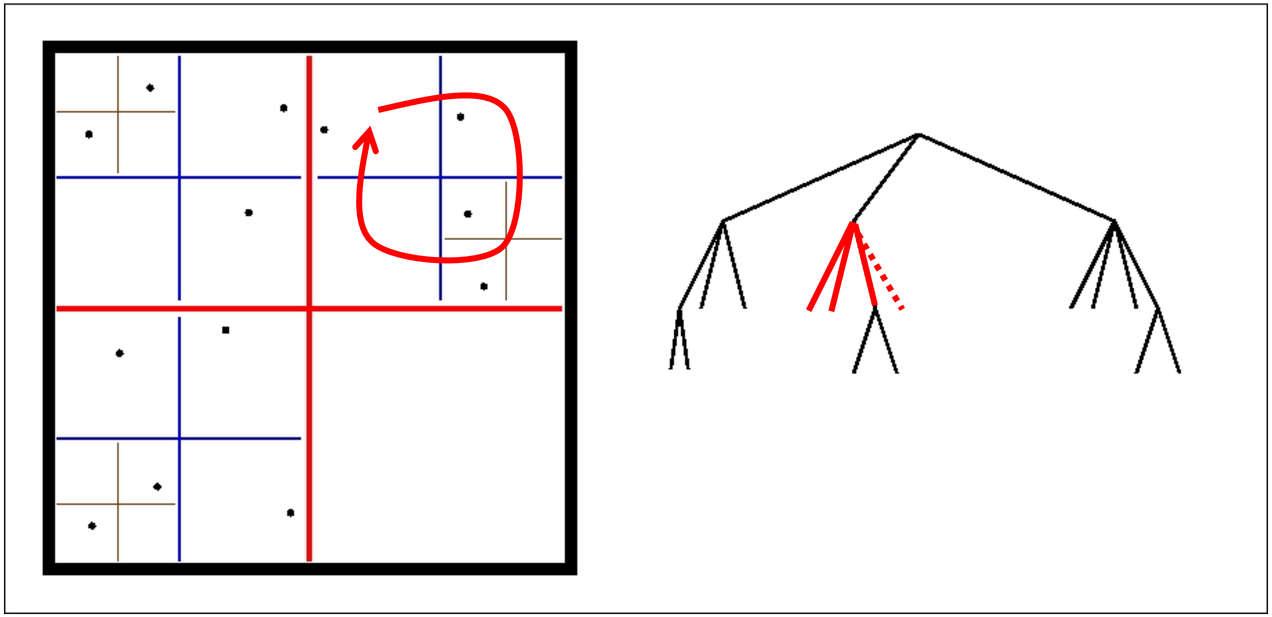


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

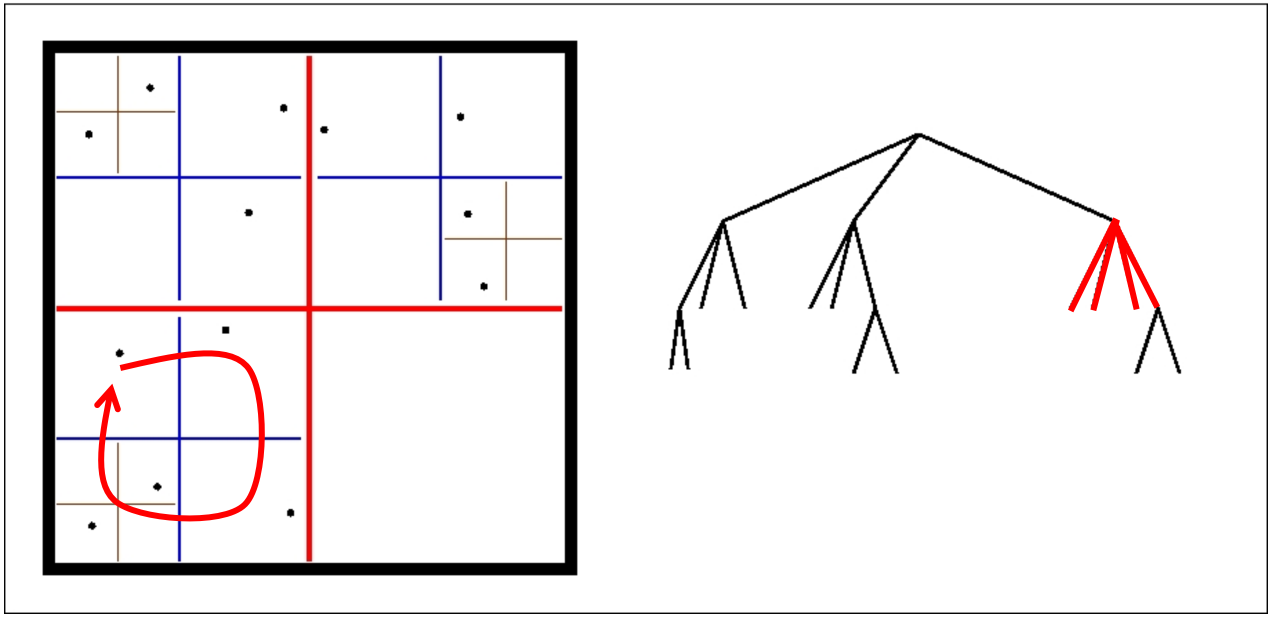


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

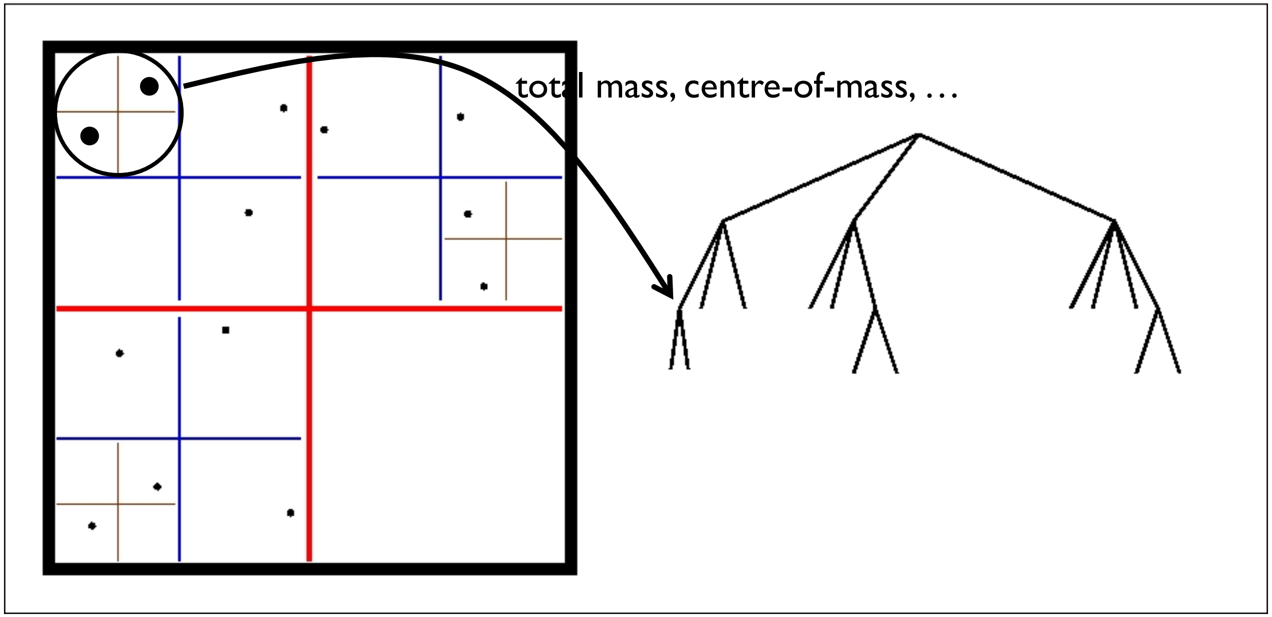


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{Gm_i m_j}{(r_i - r_j)^3}(\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

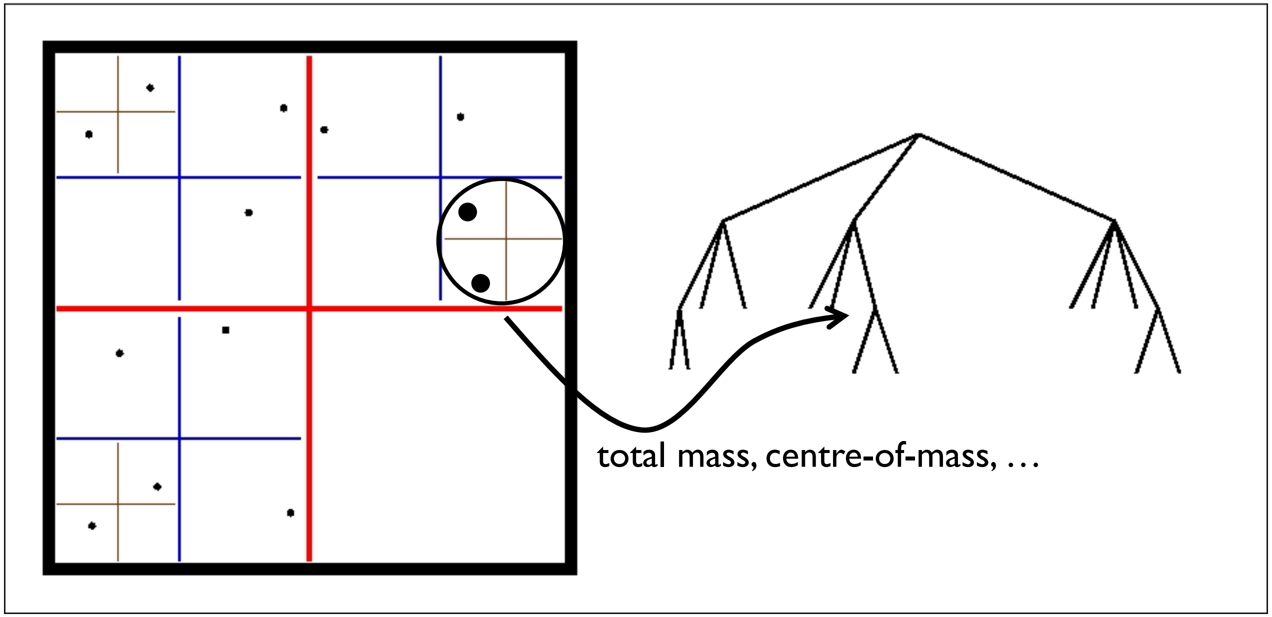


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

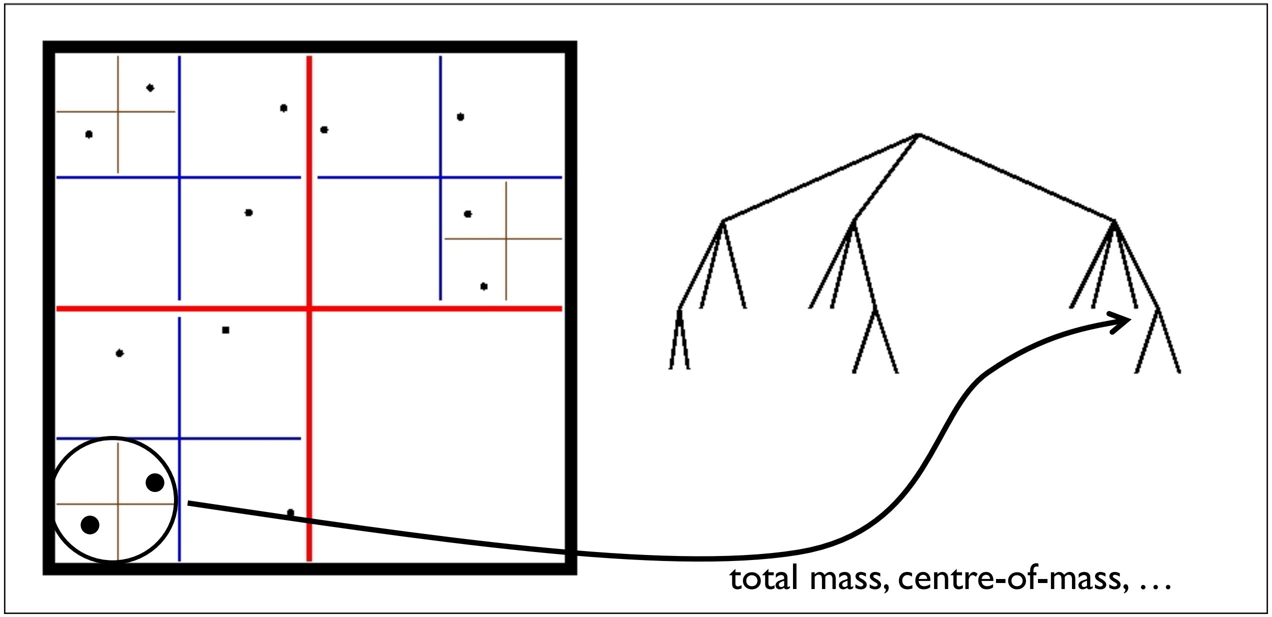


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

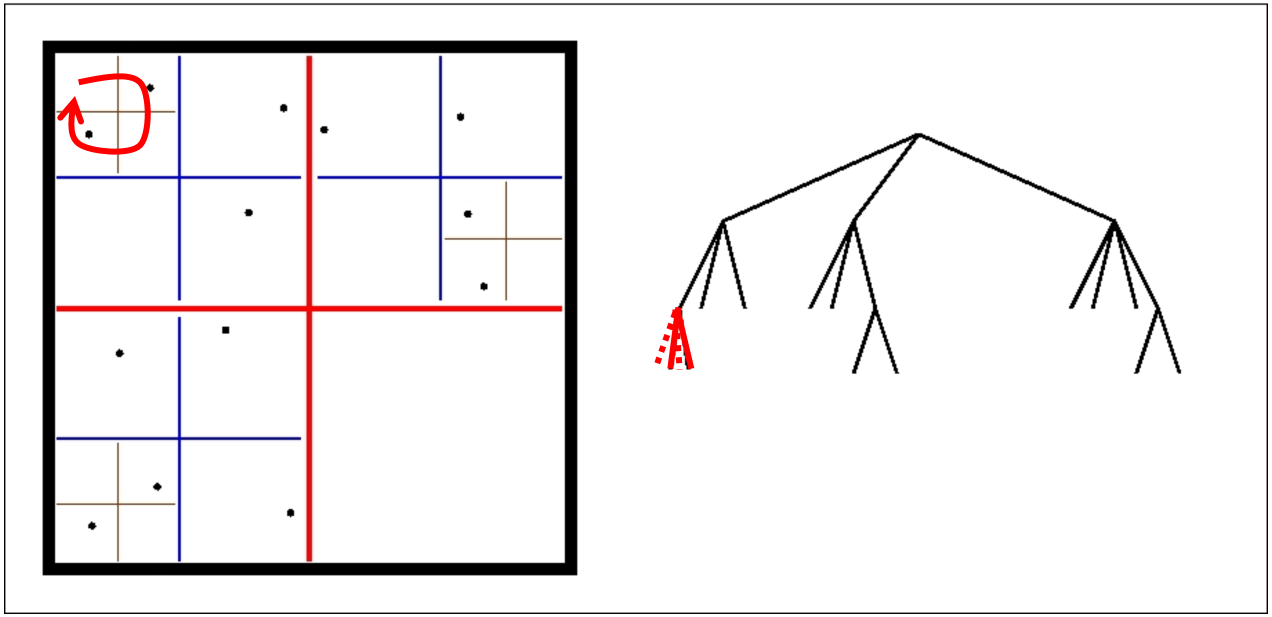


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

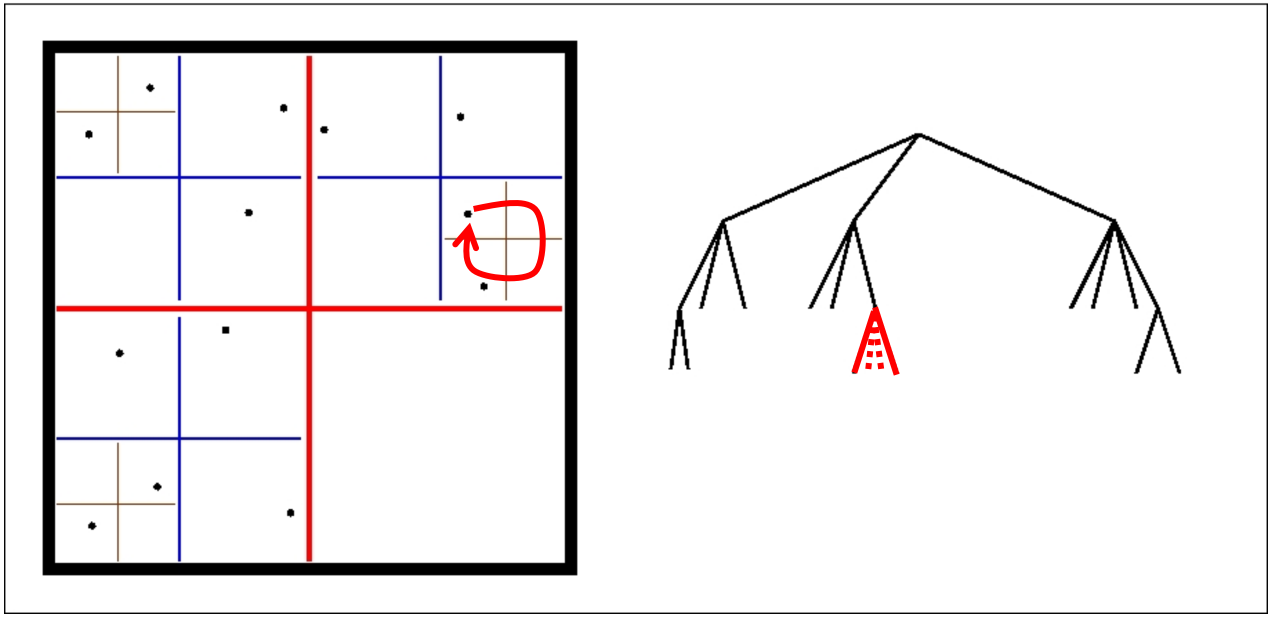


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- generating the tree:

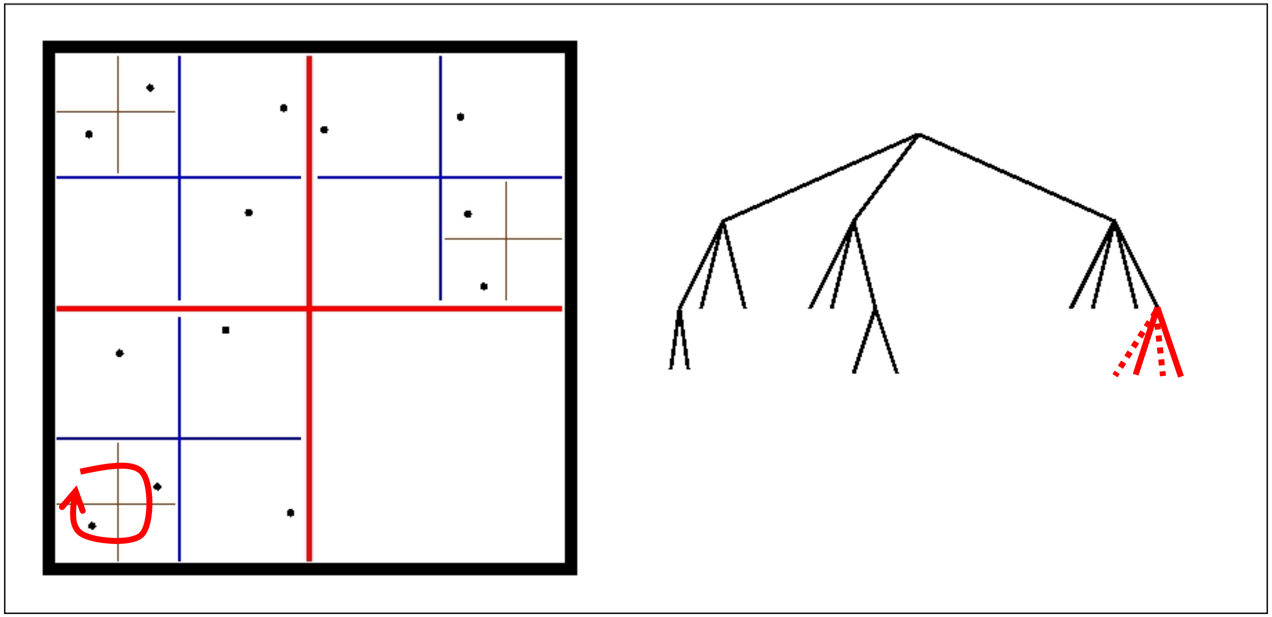


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

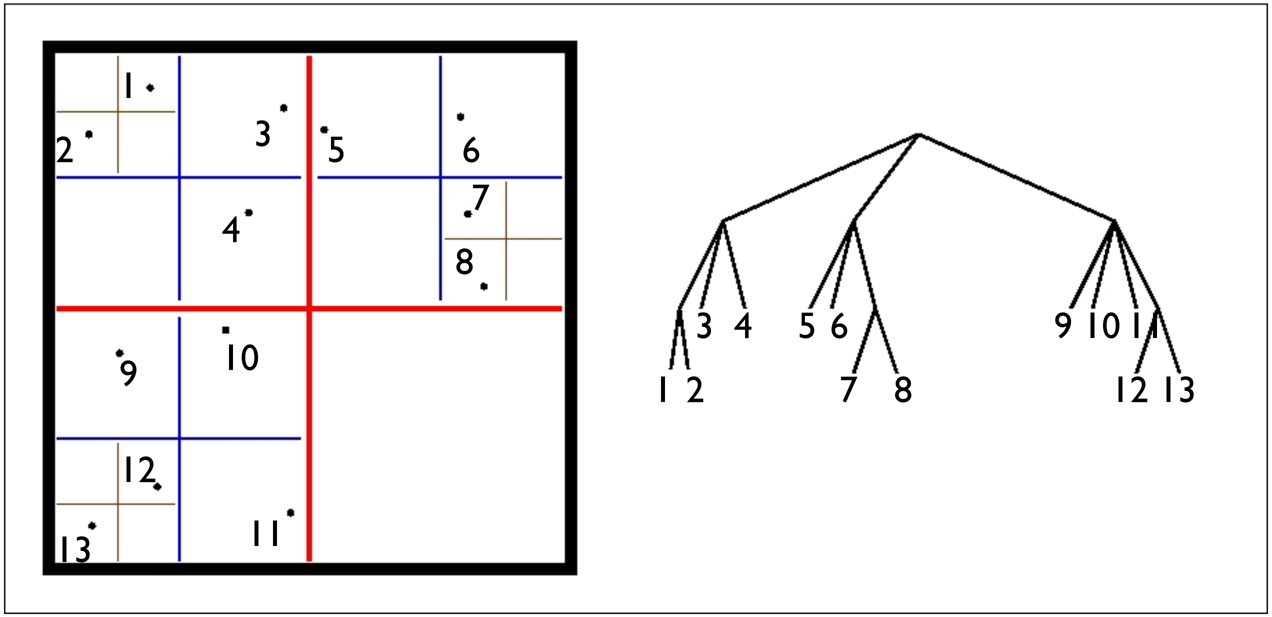
- generating the tree:



Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

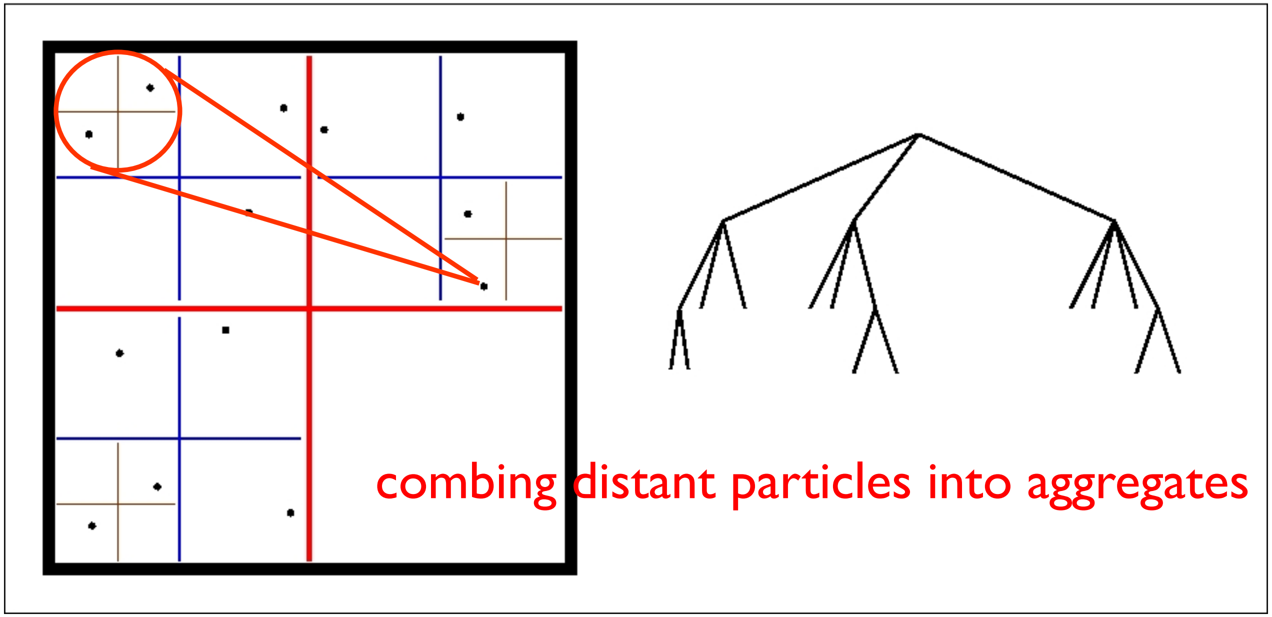


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):

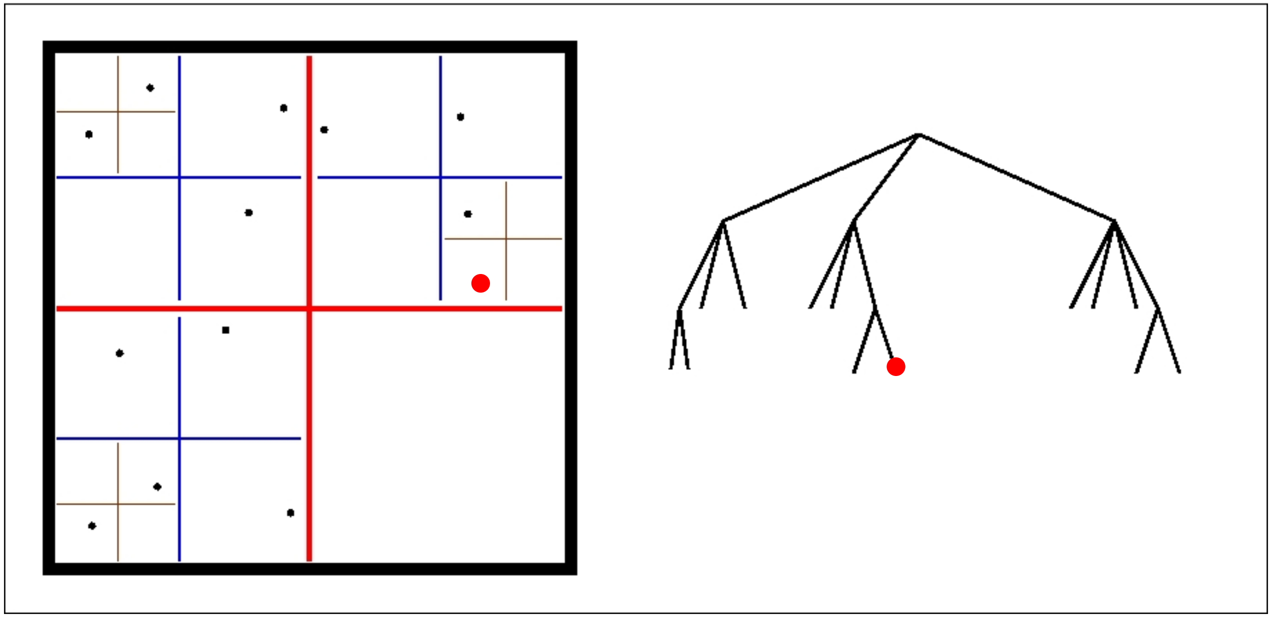


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):

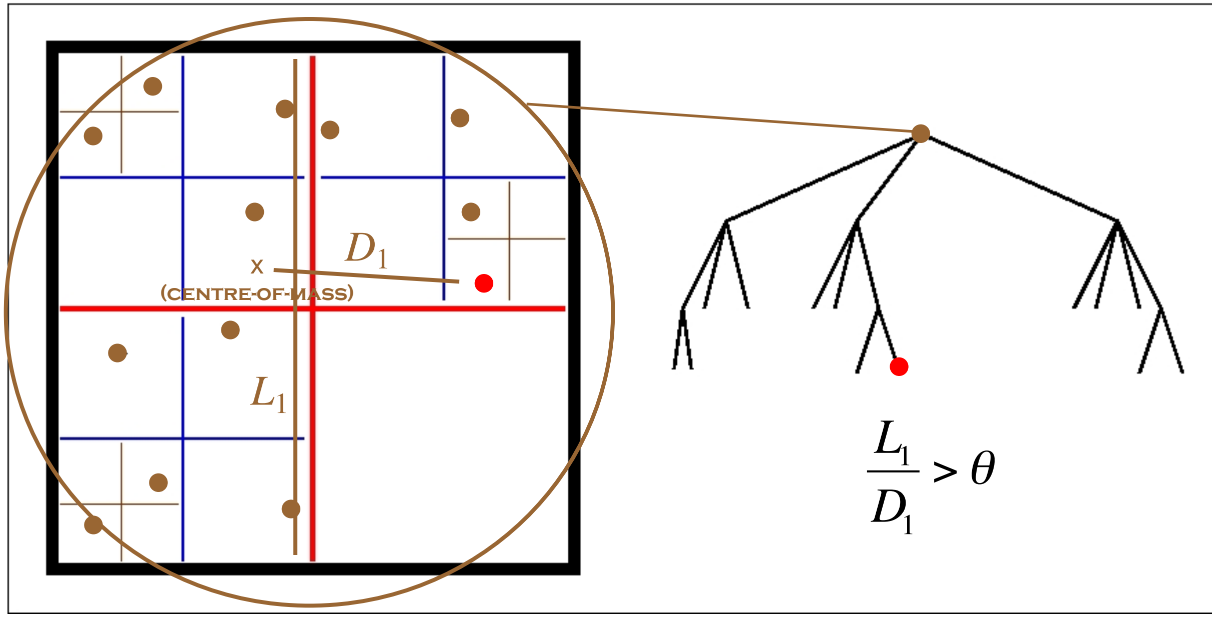


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{Gm_i m_j}{(r_i - r_j)^3}(\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):



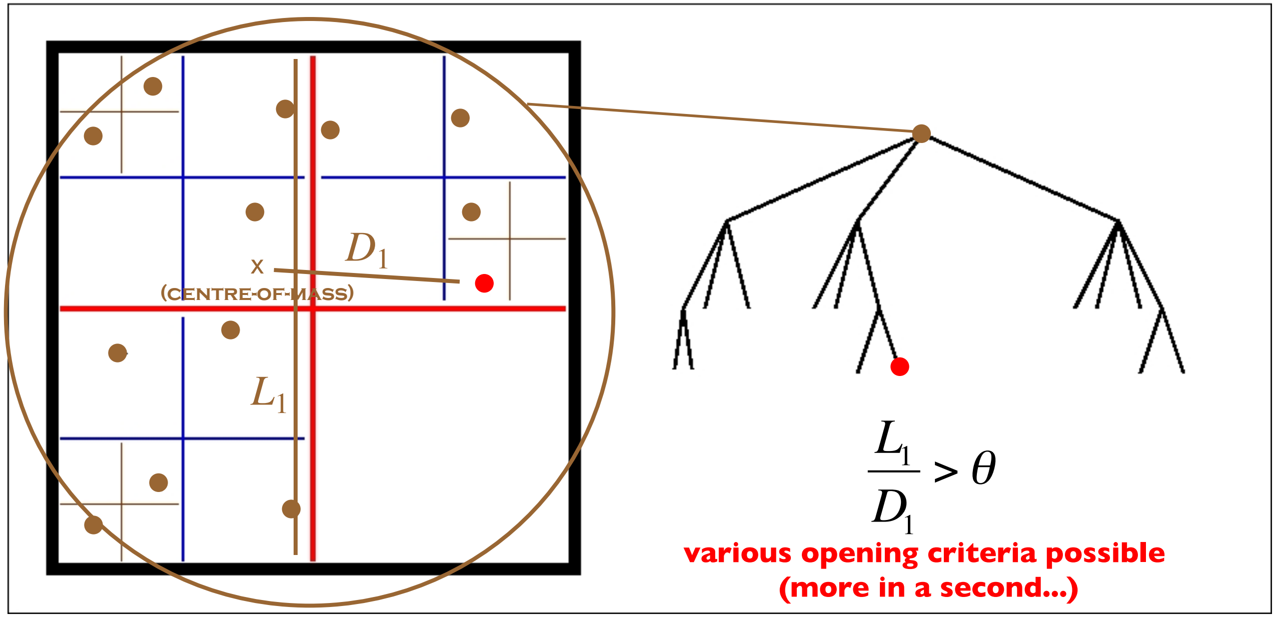
$$\frac{L_1}{D_1} > \theta$$

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):

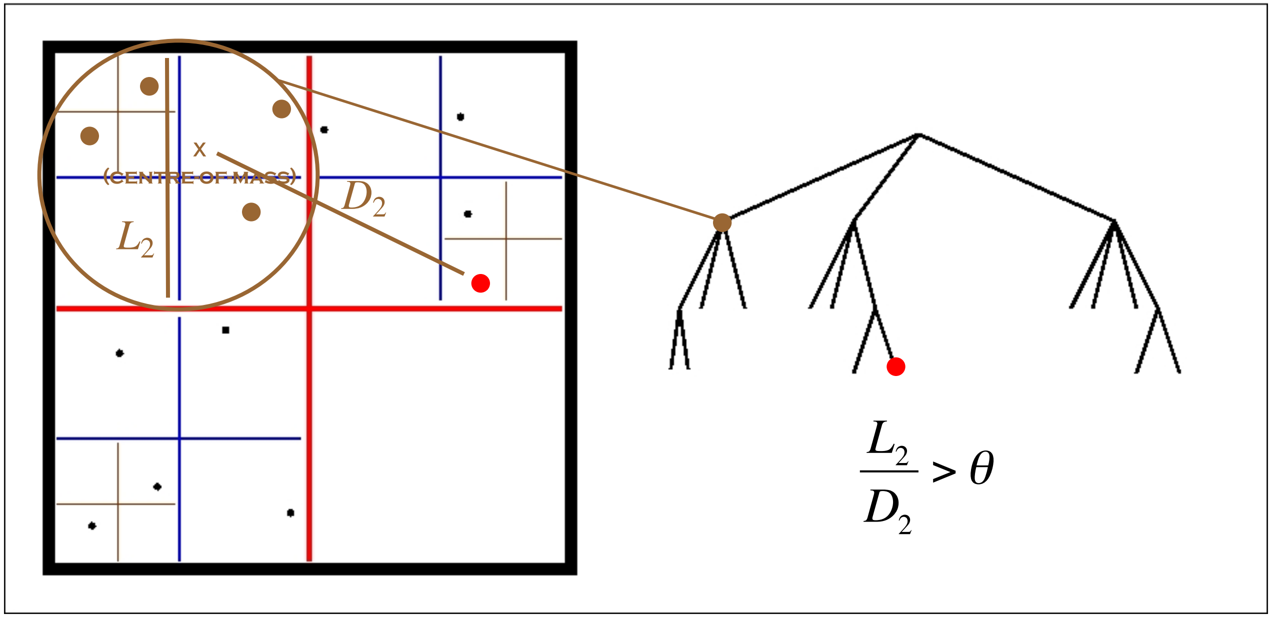


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):

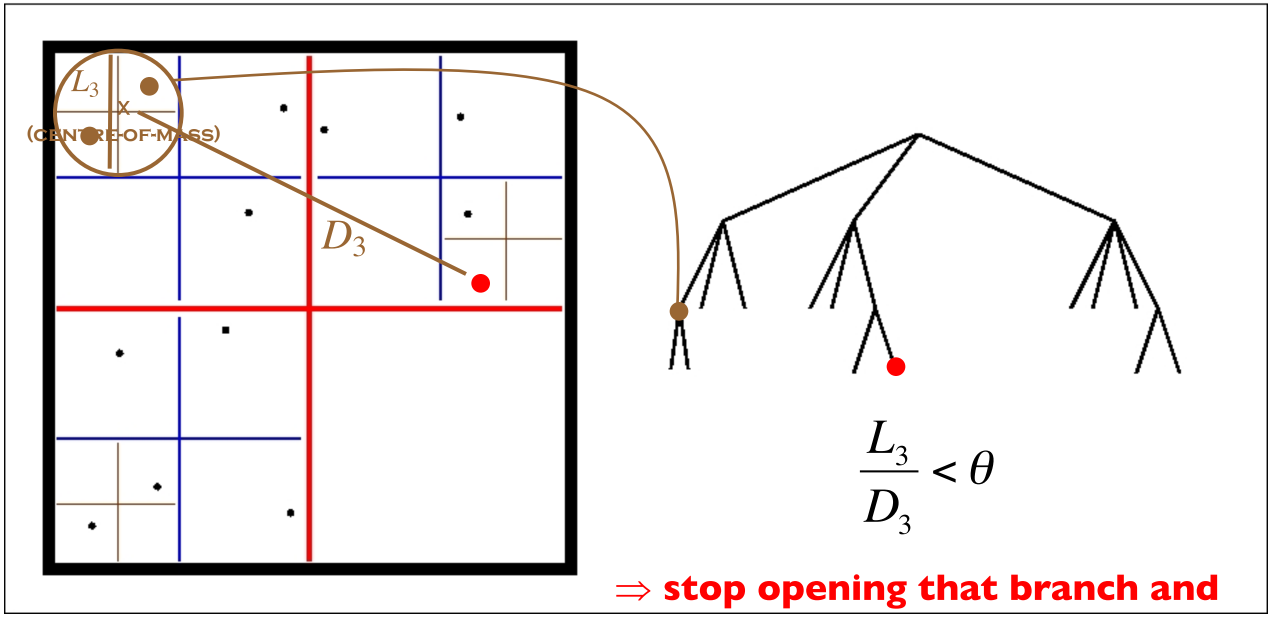


Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):



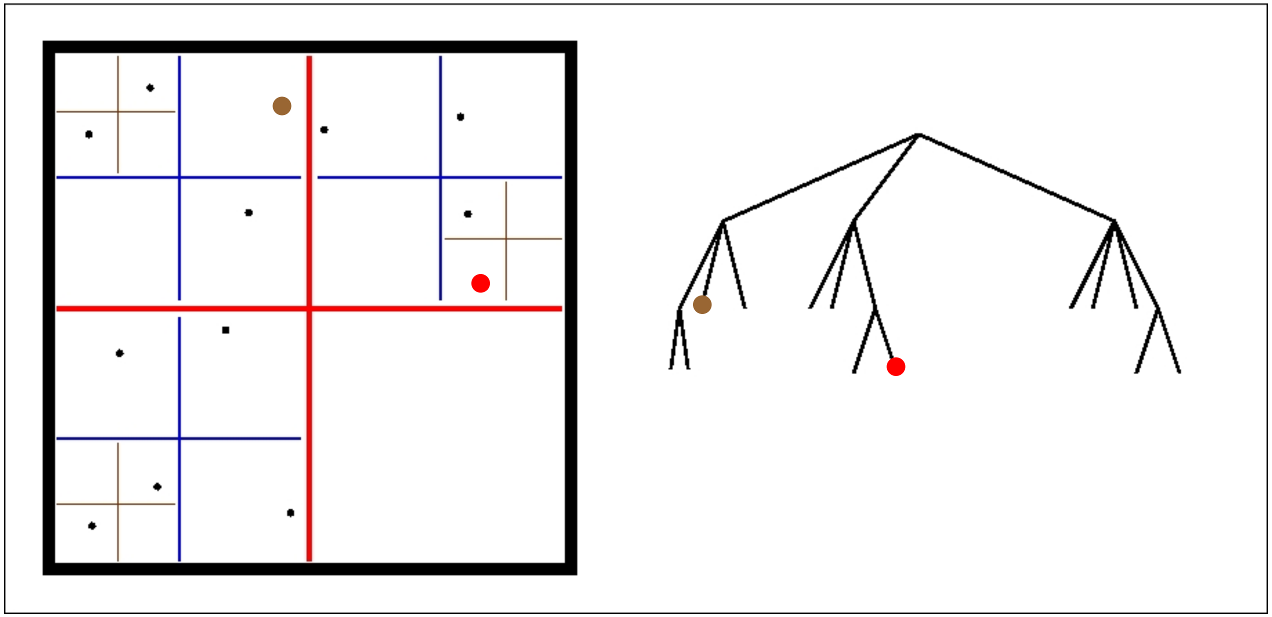
⇒ stop opening that branch and add force contribution from "super-particle"

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3}(\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):

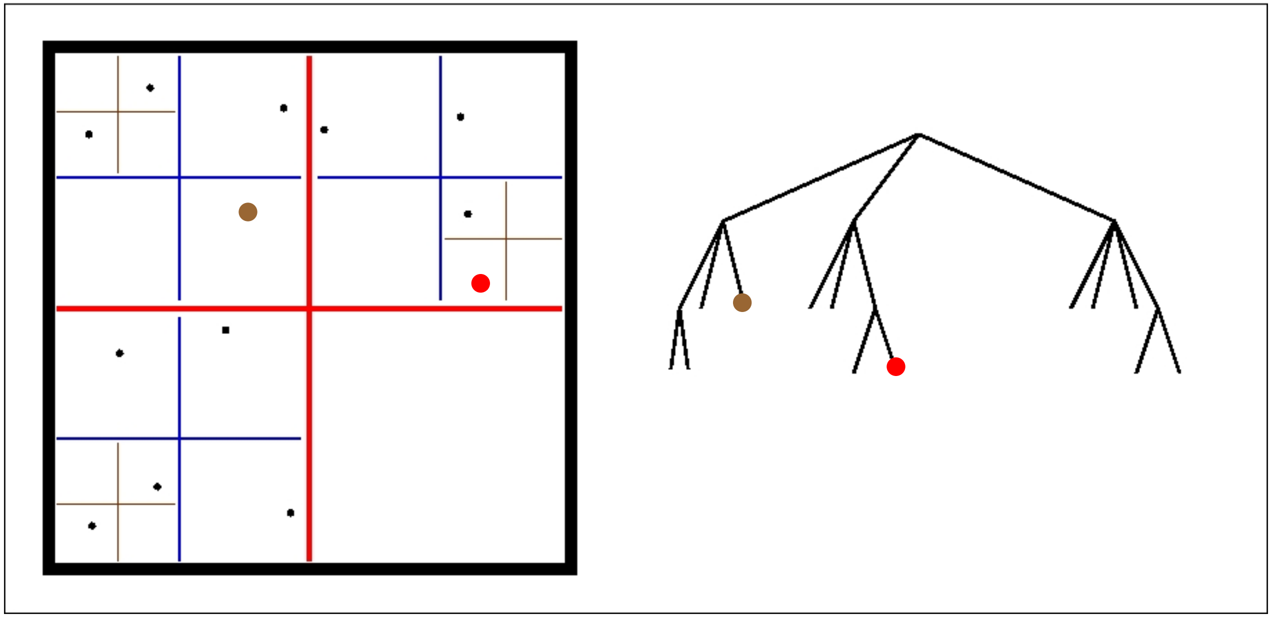


we still need to add the remaining contributions from that branch…

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):



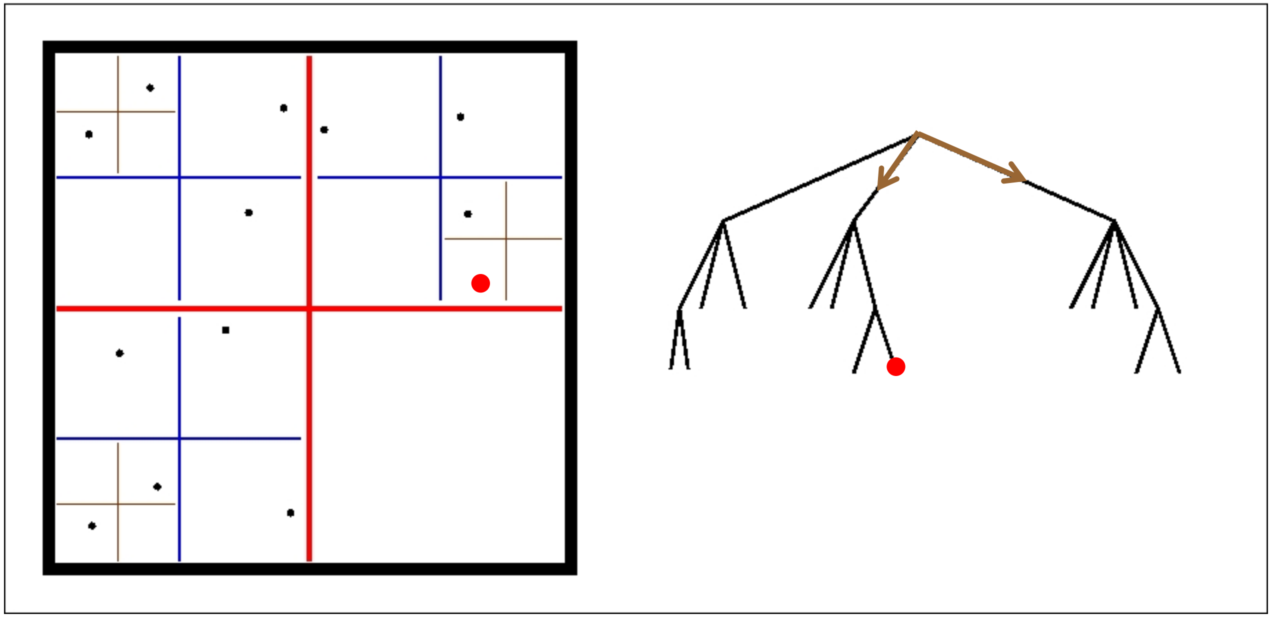
we still need to add the remaining contributions from that branch…

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- walking the tree ($\forall i \in N$):



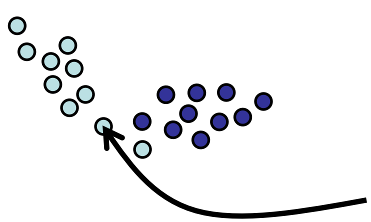
...as well as walking the other branches!

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

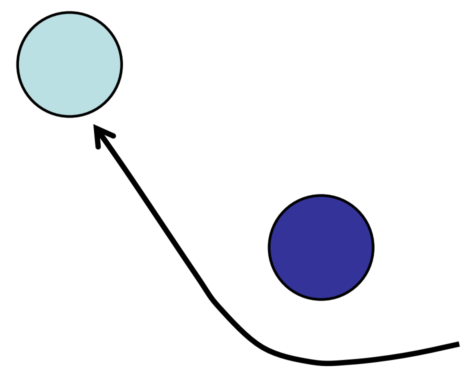
we use (collisionless) particles to sample $f(x,v,t)$



the particles sampling the field adjust

well sampled system

vs.



the particle sampling the field bounces off

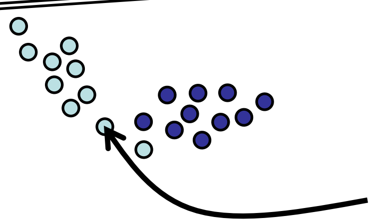
undersampled system

- direct particle-particle summation (PP)

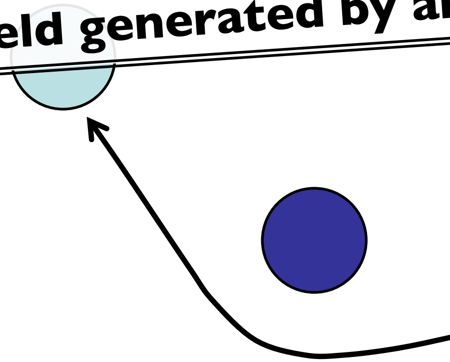
$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{G m_i m_j}{(r_i - r_j)^3} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

we use (collisionless) particles to sample $f(x,v,t)$

each particle should only feel the mean field generated by all particles!



vs.



the particles sampling the field adjust

the particle sampling the field bounces off

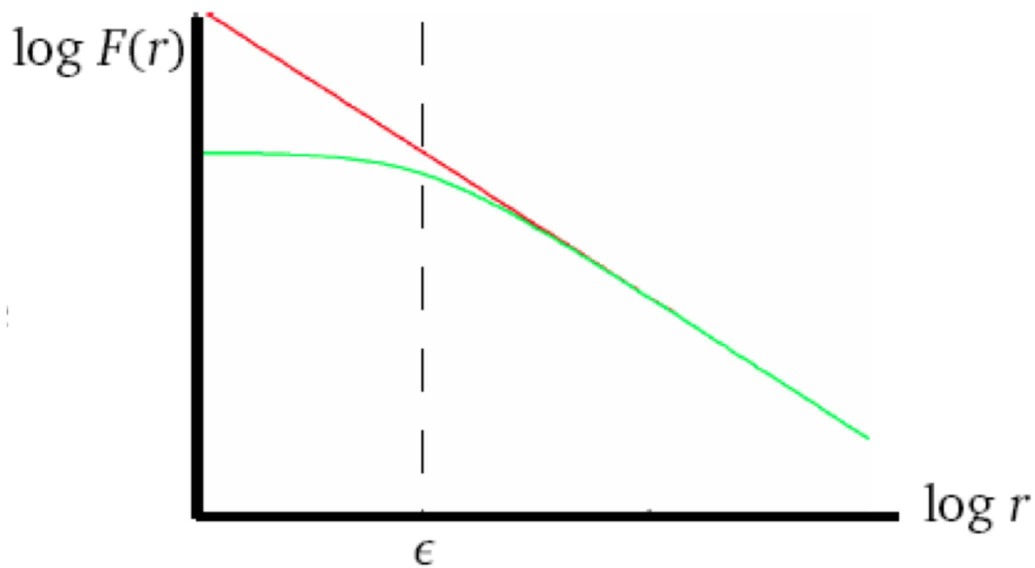
well sampled system

undersampled system

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$



Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

"soften" the force to…

1. avoid the singularity for $r_i = r_j$
2. smooth mass density on small scales

Solving for Gravity

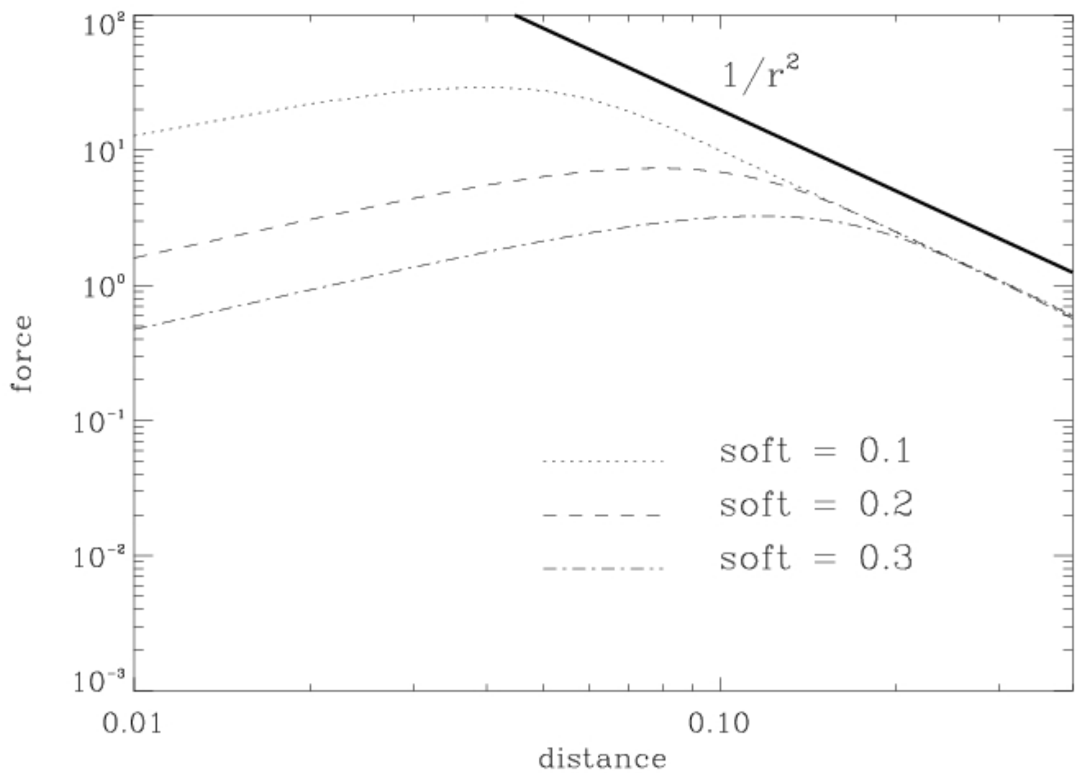
- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

ε determines the overall force resolution of the simulation

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$



Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

error budget?

Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i\neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

  - error estimate:

$$MISE = \left\langle \iiint \rho(\vec{x}) \left|\vec{F}(\vec{x}) - \vec{F}_{true}(\vec{x})\right|^2 d^3x \right\rangle$$

MISE = mean integrated square error

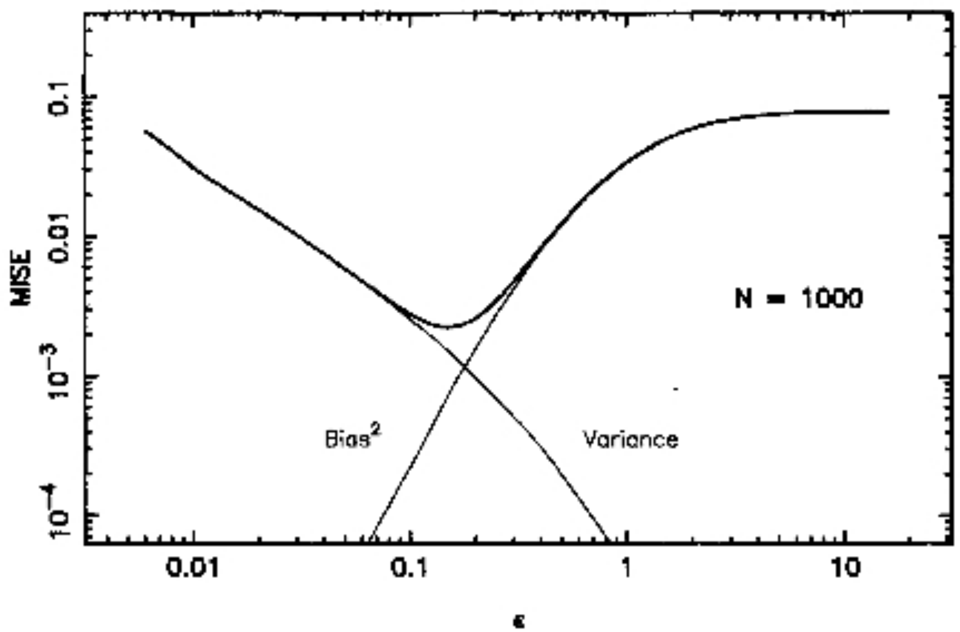
Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

- error estimate:

$$MISE = \left\langle \iiint \rho(\vec{x}) \left| \vec{F}(\vec{x}) - \vec{F}_{true}(\vec{x}) \right|^2 d^3x \right\rangle$$



Merritt, ApJ 111, 2462 (1996)

interplay between N and ε: $N\varepsilon^3$=const.

MISE = mean integrated square error

const.=$\left(\frac{B}{30}\right)^3$ for cosmological simulations (where B is the size of the cubical domain in 1D)

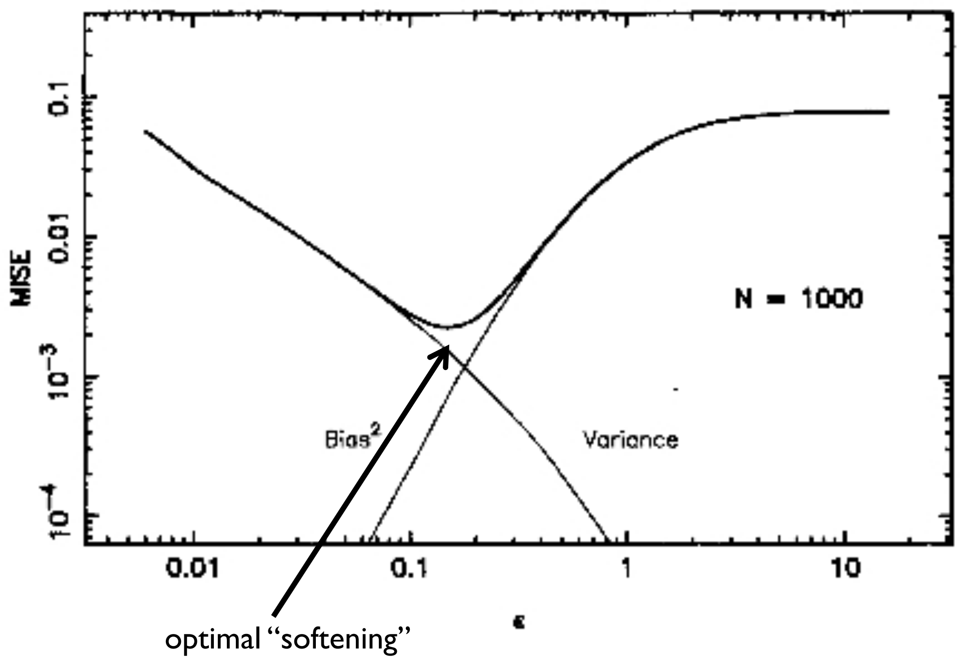
Solving for Gravity

- direct particle-particle summation (PP)

$$\vec{F}_i(\vec{r}_i) = -\sum_{i \neq j} \frac{G m_i m_j}{\left((r_i - r_j)^2 + \varepsilon^2\right)^{3/2}} (\vec{r}_i - \vec{r}_j) \quad \forall i \in N$$

  - error estimate:

$$MISE = \left\langle \iiint \rho(\vec{x}) \left| \vec{F}(\vec{x}) - \vec{F}_{true}(\vec{x}) \right|^2 d^3x \right\rangle$$



Merritt, ApJ 111, 2462 (1996)

MISE = mean integrated square error

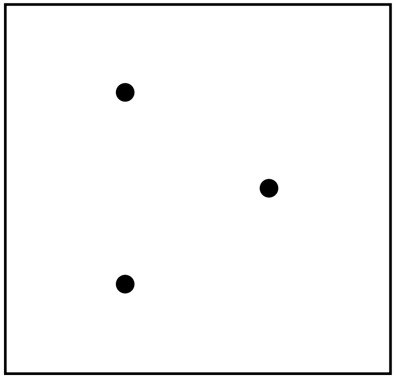
- Poisson’s equation vs. Poisson’s integral
- tree codes
- **PM codes**

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$

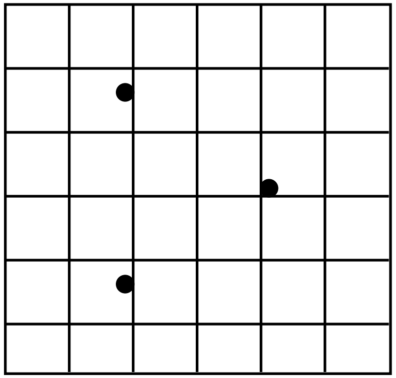


Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$

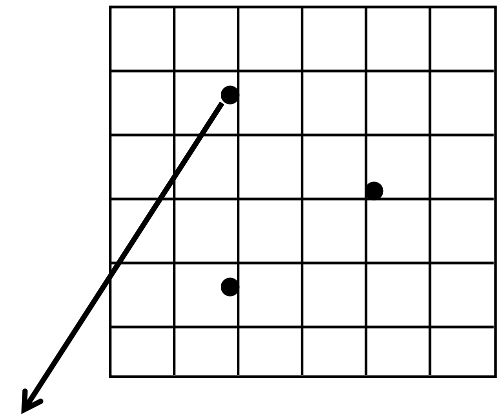


Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



1. calculate mass density on grid

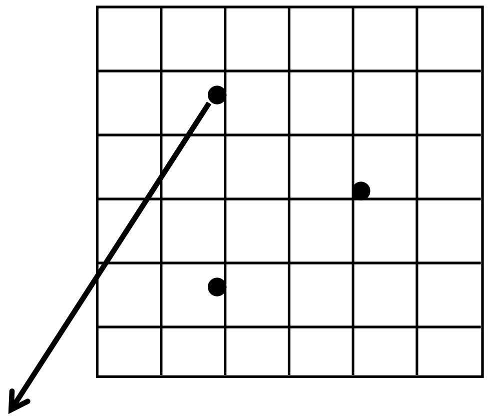
$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m}) \quad \forall i \in N$

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



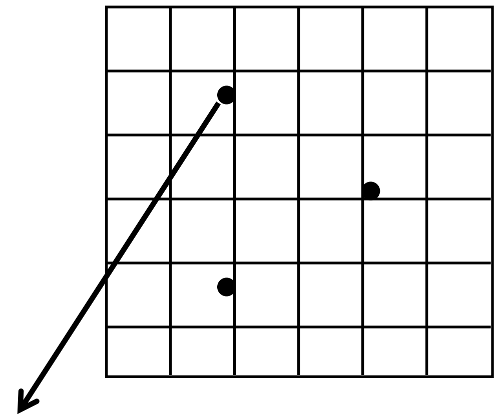
1. calculate mass density on grid $\vec{x}_i \to \rho(\vec{g}_{k,l,m}) \quad \forall i \in N$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



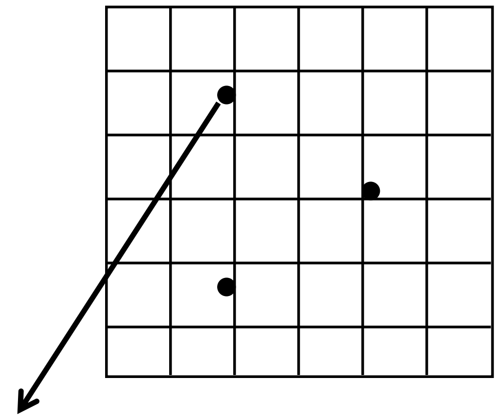
1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$ $\forall i \in N$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



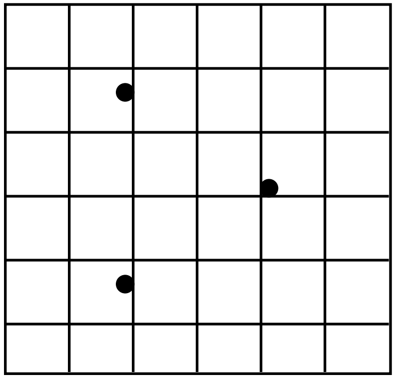
1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m}) \quad \forall i \in N$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i) \quad \forall i \in N$

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

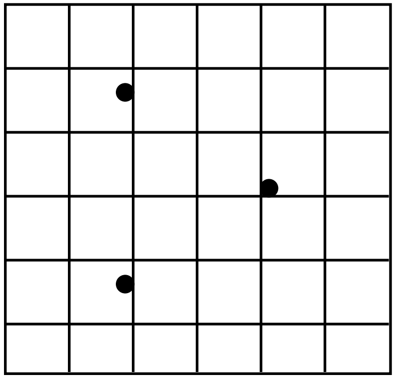
sounds like a waste of time and computer resources, but **exceptionally fast** in practice

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

Solving for Gravity

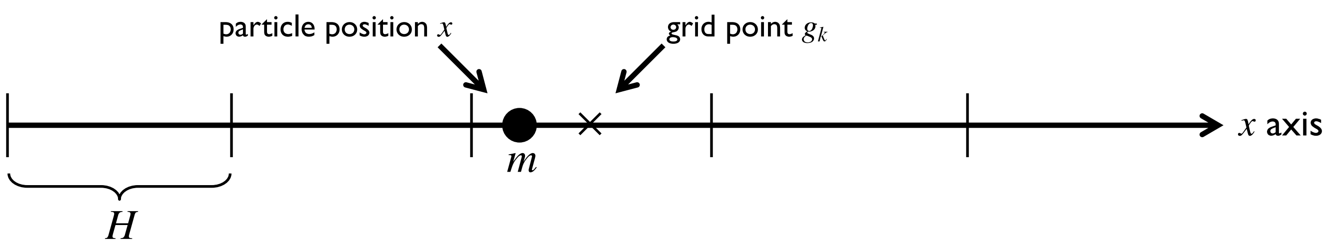
- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

<u>example:</u> 1 particle on 1 dimensional grid

$$M(g_k) = mW(d) \qquad d = |x - g_k|$$

$$\rho(g_k) = \frac{M(g_k)}{H}$$



Solving for Gravity

- density assignment schemes

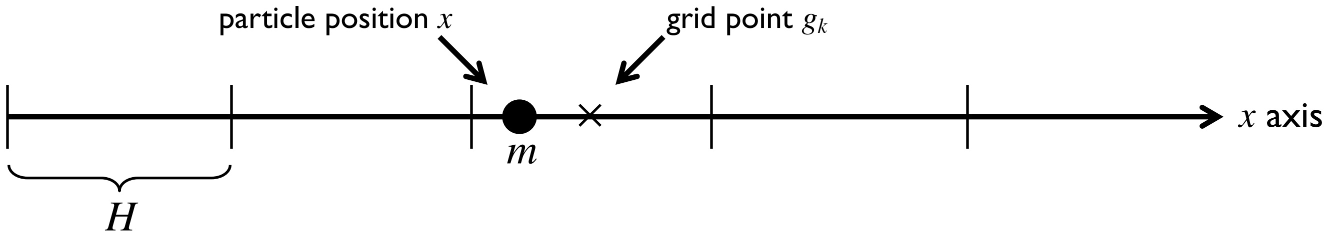
$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

example: 1 particle on 1 dimensional grid

$$M(g_k) = mW(d) \qquad d = |x - g_k|$$

mass assignment function

$$\rho(g_k) = \frac{M(g_k)}{H}$$



Solving for Gravity

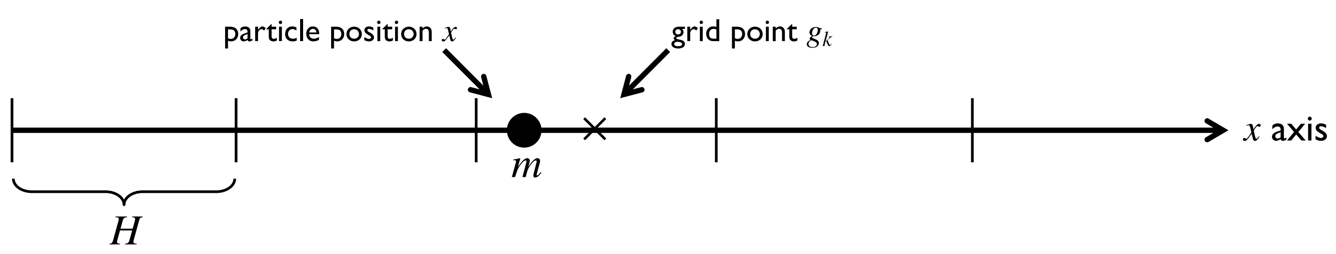
- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

<u>example:</u> 1 particle on 1 dimensional grid

• hierarchy of mass assignment schemes:

– Nearest-Grid-Point	NGP
– Could-In-Cell	CIC
– Triangular-Shaped Cloud	TSC
– …	



Solving for Gravity

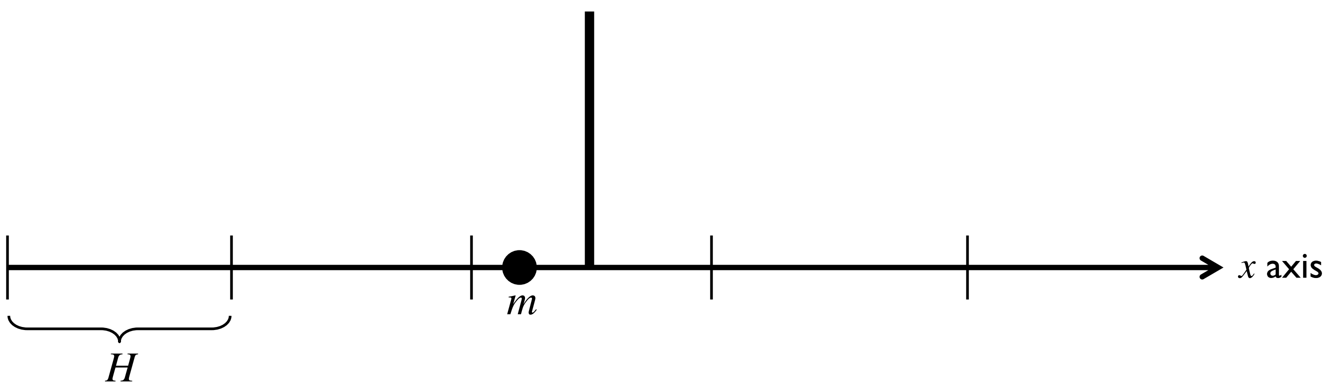
- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

<u>Nearest-Grid-Point (NGP):</u>

mass assignment function:

$$W(d) = \begin{cases} 1 & d \leq H/2 \\ 0 & \text{otherwise} \end{cases}$$



- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

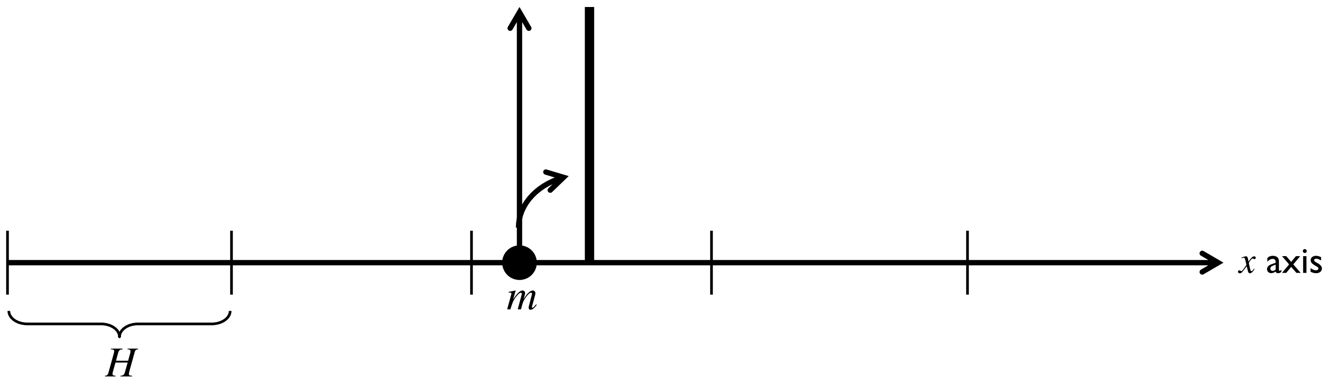
Nearest-Grid-Point (NGP):

particle shape:

$$S(x) = \delta(x)$$

mass assignment function:

$$W(d) = \begin{cases} 1 & d \leq H/2 \\ 0 & \text{otherwise} \end{cases}$$



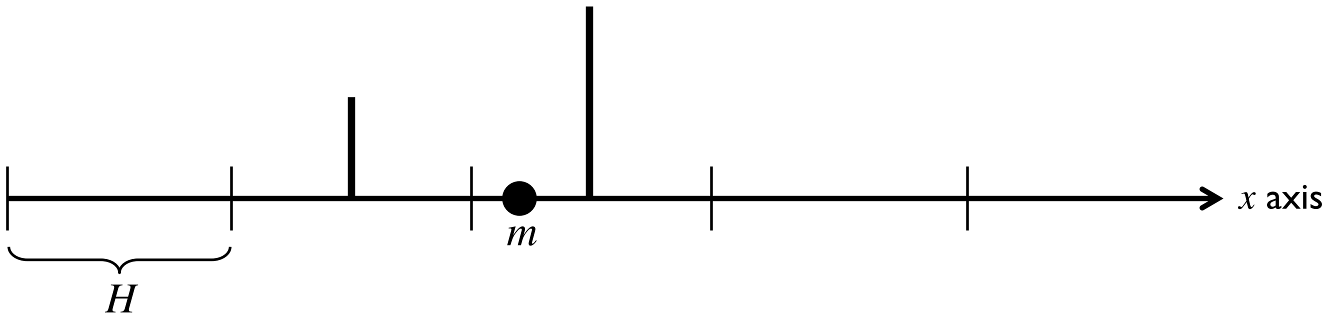
- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

<u>Cloud-In-Cell (CIC):</u>

mass assignment function:

$$W(d) = \begin{cases} 1 - \dfrac{d}{H} & d \leq H \\ 0 & \text{otherwise} \end{cases}$$



- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

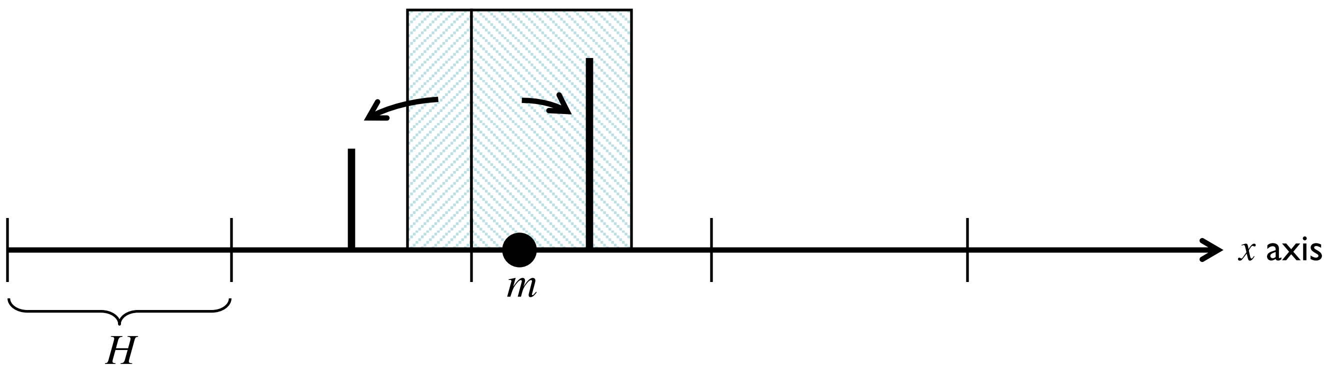
Cloud-In-Cell (CIC):

particle shape:

$$S(x) = \begin{cases} 1 & |x| \le H/2 \\ 0 & \text{otherwise} \end{cases}$$

mass assignment function:

$$W(d) = \begin{cases} 1 - \dfrac{d}{H} & d \le H \\ 0 & \text{otherwise} \end{cases}$$



Solving for Gravity

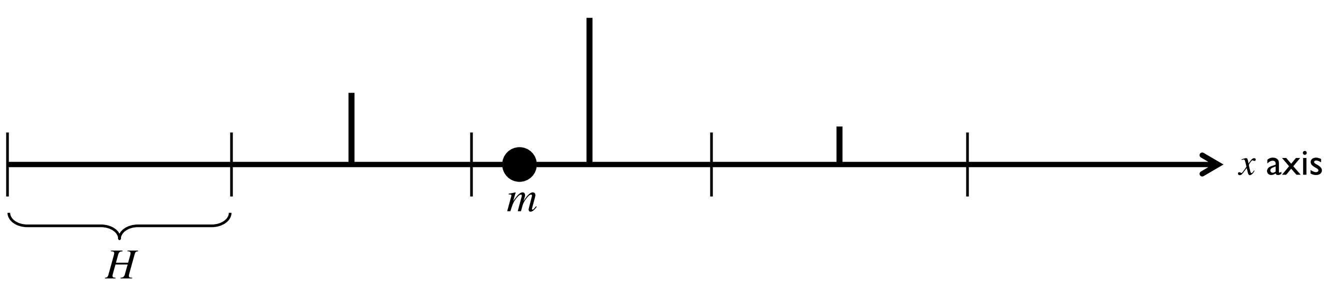
- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

Triangular-Shaped-Cloud (TSC):

mass assignment function:

$$W(d) = \begin{cases} \frac{3}{4} - \left(\frac{d}{H}\right)^2 & d \leq \frac{H}{2} \\ \frac{1}{2}\left(\frac{3}{2} - \frac{d}{H}\right)^2 & \frac{H}{2} \leq d \leq \frac{3H}{2} \\ 0 & \text{otherwise} \end{cases}$$



Solving for Gravity

- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

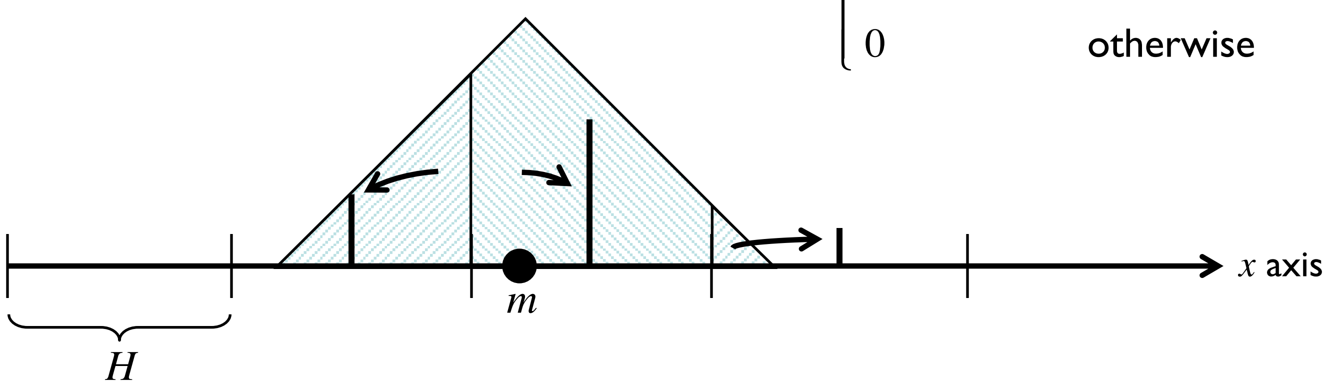
Triangular-Shaped-Cloud (TSC):

particle shape:

$$S(x) = \begin{cases} 1 - \frac{|x|}{H} & |x| \le H \\ 0 & \text{otherwise} \end{cases}$$

mass assignment function:

$$W(d) = \begin{cases} \frac{3}{4} - \left(\frac{d}{H}\right)^2 & d \le \frac{H}{2} \\ \frac{1}{2}\left(\frac{3}{2} - \frac{d}{H}\right)^2 & \frac{H}{2} \le d \le \frac{3H}{2} \\ 0 & \text{otherwise} \end{cases}$$



Solving for Gravity

- density assignment schemes

$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

<u>N particles on 3 dimensional grid</u>

$$\vec{d} = \vec{x}_i - \vec{g}_{k,l,m}$$

$$M(\vec{g}_{k,l,m}) = \sum_{i=1}^{N} m_i W(|d_x|) W(|d_y|) W(|d_z|)$$

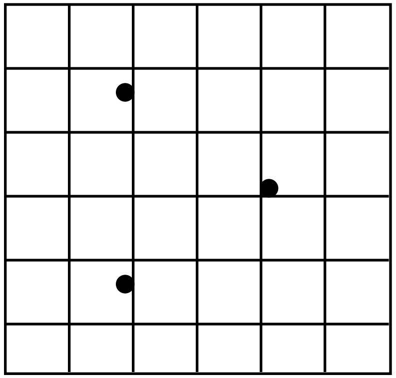
$$\rho(\vec{g}_{k,l,m}) = \frac{M(\vec{g}_{k,l,m})}{H^3}$$

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

- relaxation technique: applicable and usable for ***any*** differential equation
- FTT technique: only applicable and usable for ***linear*** differential equation

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

- relaxation technique: applicable and usable for ***any*** differential equation
- **FTT technique:** only applicable and usable for ***linear*** differential equation

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

- Green's function method:
  - solve differential equation by Fourier transformation
  - applicable and usable for ***linear*** differential equations

Solving for Gravity

- numerically integrate Poisson's equation *fast fourier transform method*
  - Green's function method

$$\Delta\Phi = \rho \qquad \Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\,\rho(\vec{x}')d^3x' \;\; ; \;\; \mathcal{G}(\vec{x}) = \frac{1}{4\pi\, x}$$

Solving for Gravity

- numerically integrate Poisson's equation *fast fourier transform method*
  - Green's function method

$$\Delta\Phi = \rho \qquad \Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x} - \vec{x}')\, \rho(\vec{x}')\, d^3x' \;\; ; \quad \mathcal{G}(\vec{x}) = \frac{1}{4\pi\, x}$$

$$\Phi = \rho \otimes \mathcal{G}$$

FFT ➔ convolution becomes multiplication

$$\hat{\Phi} = \hat{\rho} \quad \hat{\mathcal{G}}$$

(with $\hat{\mathcal{G}} = -\frac{1}{k^2}$ for Poisson's equation)

Solving for Gravity

- numerically integrate Poisson's equation *fast fourier transform method*
  - Green's function method

$$\Delta\Phi = \rho \qquad \Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x}-\vec{x}')\,\rho(\vec{x}')d^3x' \;\; ; \;\; \mathcal{G}(\vec{x}) = \frac{1}{4\pi x}$$

$$\Phi = \rho \otimes \mathcal{G}$$

FFT → convolution becomes multiplication

$$\hat{\Phi} = \hat{\rho} \quad \hat{\mathcal{G}}$$

(with $\hat{\mathcal{G}} = -\frac{1}{k^2}$ for Poisson's equation)

FFT^{-1}

$$\Phi$$

(FFT demands a *regular* grid though!)

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

- **relaxation technique**: applicable and usable for ***any*** differential equation
- FTT technique: only applicable and usable for ***linear*** differential equation

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

obtain iterative solver by discretizing differential equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

obtain iterative solver by discretizing differential equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

$$\begin{aligned}
\Delta\Phi_{k,l,m} &= \nabla\cdot\nabla\Phi_{k,l,m} \\
&= \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \cdot \begin{pmatrix} \frac{\partial\Phi_{k,l,m}}{\partial x} \\ \frac{\partial\Phi_{k,l,m}}{\partial y} \\ \frac{\partial\Phi_{k,l,m}}{\partial z} \end{pmatrix} \\
&= \frac{1}{H}\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \cdot \begin{pmatrix} \Phi_{k+1/2,l,m} - \Phi_{k-1/2,l,m} \\ \Phi_{k,l+1/2,m} - \Phi_{k,l-1/2,m} \\ \Phi_{k,l,m+1/2} - \Phi_{k,l,m-1/2} \end{pmatrix} \\
&= \frac{1}{H}\left(\frac{\partial\Phi_{k+1/2,l,m}}{\partial x} - \frac{\partial\Phi_{k-1/2,l,m}}{\partial x} + \frac{\partial\Phi_{k,l+1/2,m}}{\partial y} - \frac{\partial\Phi_{k,l-1/2,m}}{\partial y} + \frac{\partial\Phi_{k,l,m+1/2}}{\partial z} - \frac{\partial\Phi_{k,l,m-1/2}}{\partial z} \right) \\
&= \frac{1}{H^2}\left(\Phi_{k+1,l,m} - 2\Phi_{k,l,m} + \Phi_{k-1,l,m} + \Phi_{k,l+1,m} - 2\Phi_{k,l,m} + \Phi_{k,l-1,m} + \Phi_{k,l,m+1} - 2\Phi_{k,l,m} + \Phi_{k,l,m-1} \right)
\end{aligned}$$

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

obtain iterative solver by discretizing differential equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

discretized Poisson's equation

$$\Phi_{k,l,m} = \frac{1}{6}(\Phi_{k+1,l,m} + \Phi_{k-1,l,m} + \Phi_{k,l+1,m} + \Phi_{k,l-1,m} + \Phi_{k,l,m+1} + \Phi_{k,l,m-1} - \rho_{k,l,m}H^2)$$

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

obtain iterative solver by discretizing differential equation

$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

iterative solution: $\Phi^{i}_{k,l,m} \rightarrow \Phi^{i+1}_{k,l,m}$

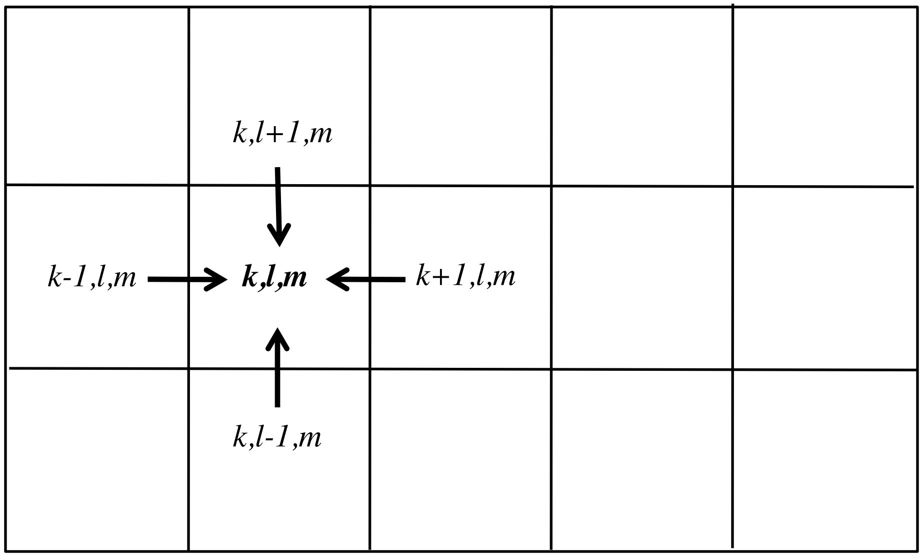
discretized Poisson's equation

$$\Phi^{i+1}_{k,l,m} = \frac{1}{6}(\Phi^{i}_{k+1,l,m} + \Phi^{i}_{k-1,l,m} + \Phi^{i}_{k,l+1,m} + \Phi^{i}_{k,l-1,m} + \Phi^{i}_{k,l,m+1} + \Phi^{i}_{k,l,m-1} - \rho_{k,l,m}H^2)$$

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

obtain iterative solver by discretizing differential equation

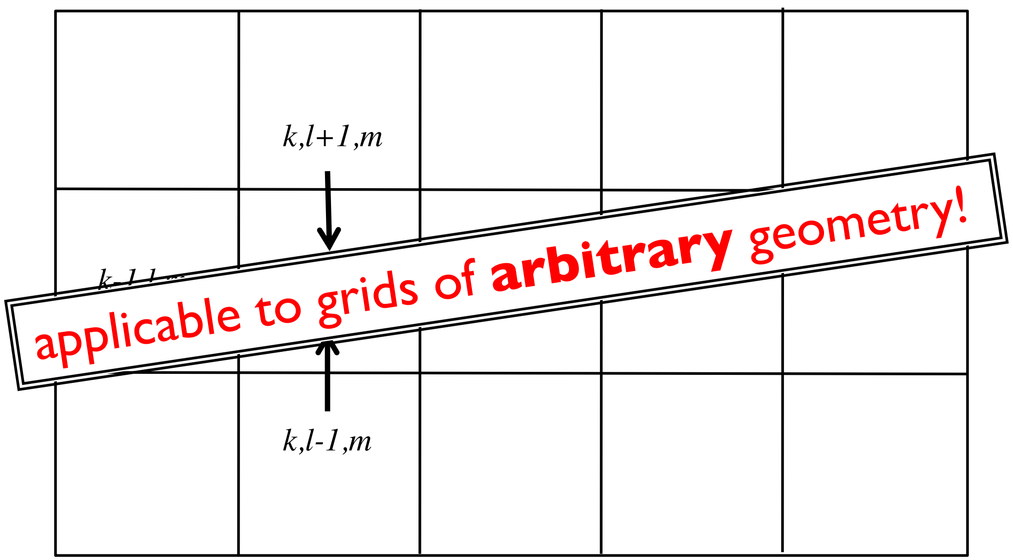


discretized Poisson's equation

$$\Phi^{i+1}_{k,l,m} = \frac{1}{6}(\Phi^{i}_{k+1,l,m} + \Phi^{i}_{k-1,l,m} + \Phi^{i}_{k,l+1,m} + \Phi^{i}_{k,l-1,m} + \Phi^{i}_{k,l,m+1} + \Phi^{i}_{k,l,m-1} - \rho_{k,l,m} H^2)$$

- numerically integrate Poisson's equation *relaxation technique*

obtain iterative solver by discretizing differential equation



discretized Poisson's equation

$$\Phi^{i+1}_{k,l,m} = \frac{1}{6}(\Phi^{i}_{k+1,l,m} + \Phi^{i}_{k-1,l,m} + \Phi^{i}_{k,l+1,m} + \Phi^{i}_{k,l-1,m} + \Phi^{i}_{k,l,m+1} + \Phi^{i}_{k,l,m-1} - \rho_{k,l,m} H^2)$$

Solving for Gravity

- numerically integrate Poisson's equation

accuracy of either

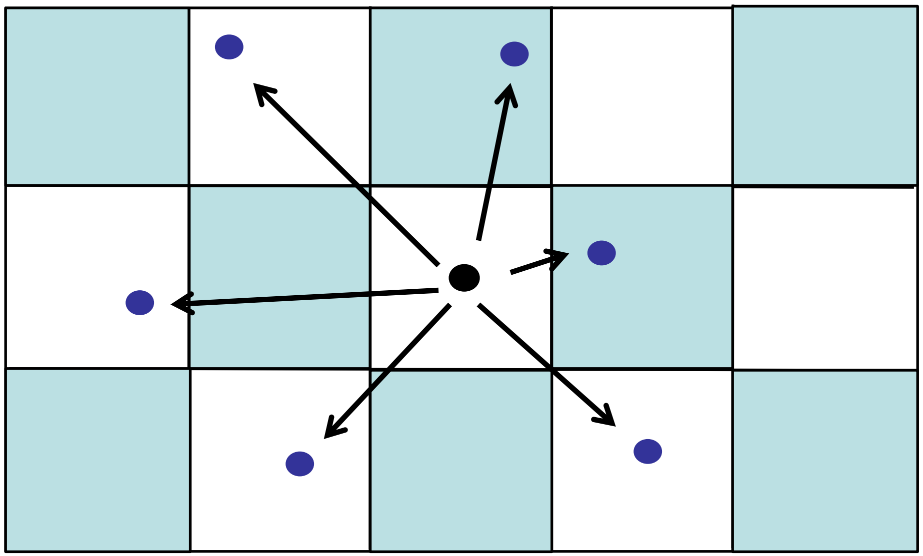
relaxation
or
FFT method

to solve Poisson's equation?

Solving for Gravity

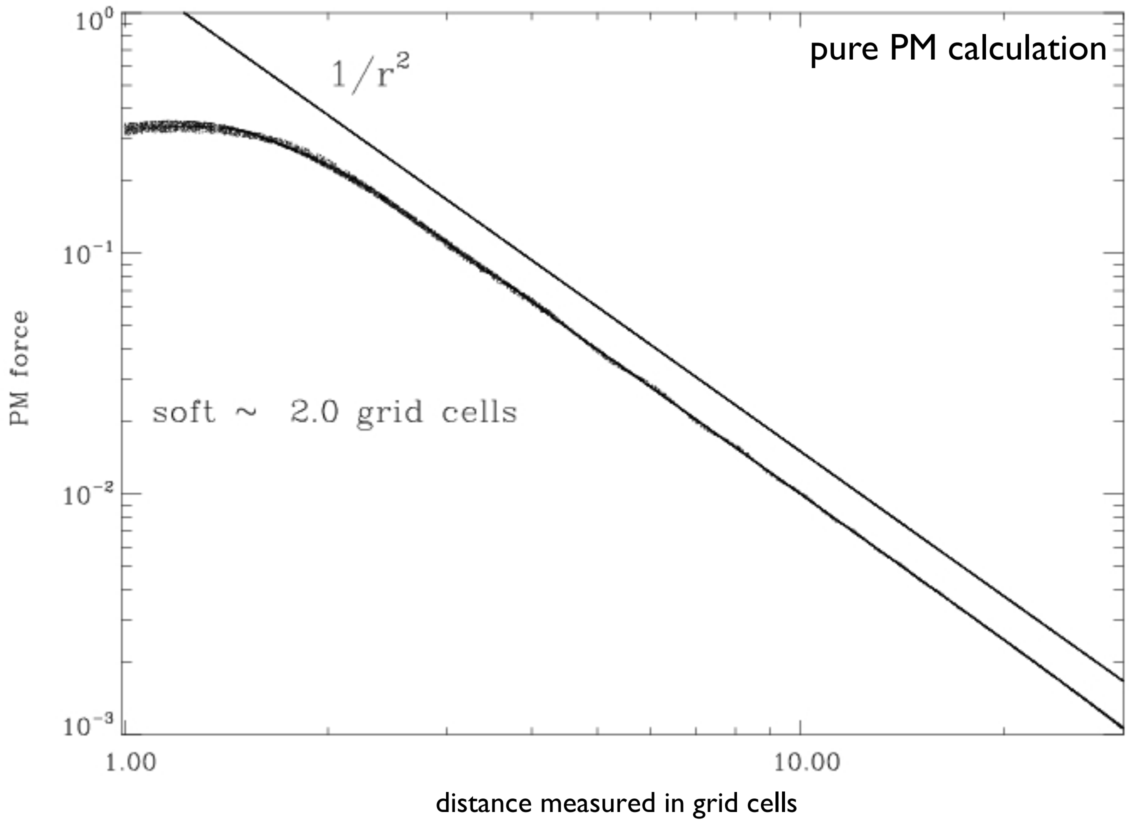
- numerically integrate Poisson's equation

pure PM calculation



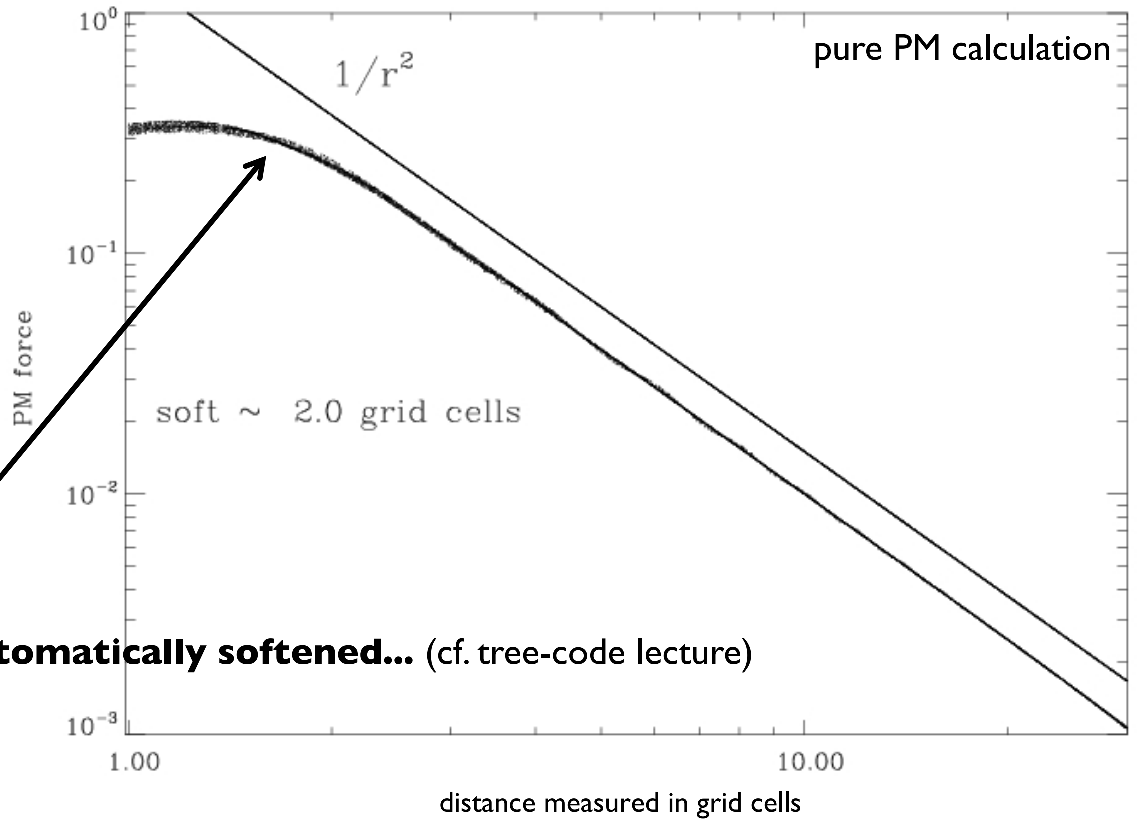
Solving for Gravity

- numerically integrate Poisson's equation



Solving for Gravity

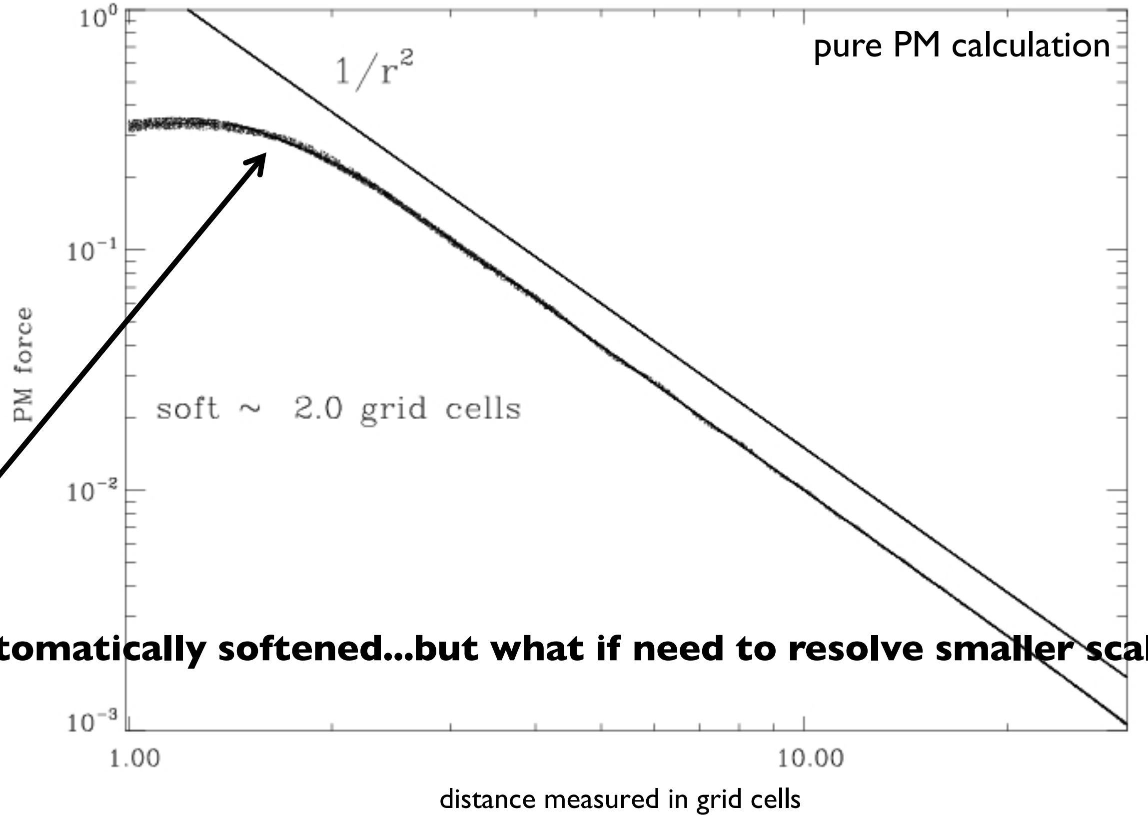
- numerically integrate Poisson's equation



the force is automatically softened... (cf. tree-code lecture)

Solving for Gravity

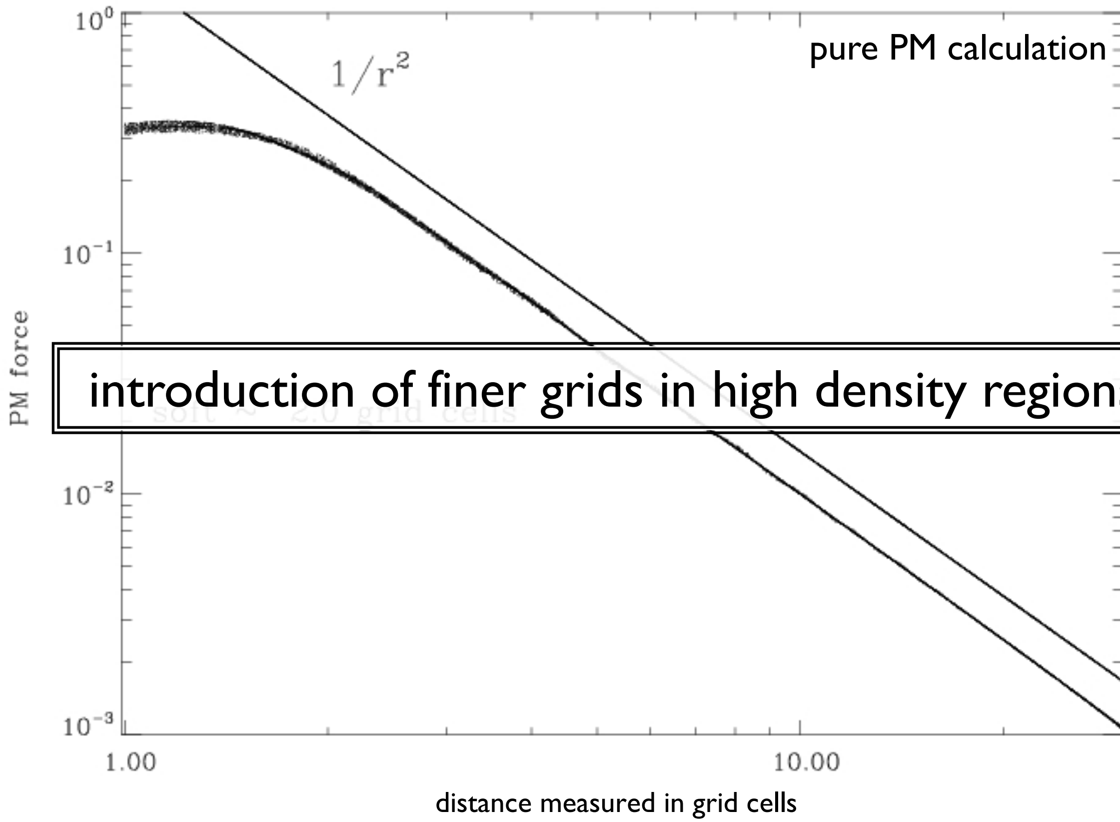
- numerically integrate Poisson's equation



the force is automatically softened...but what if need to resolve smaller scales?

Solving for Gravity

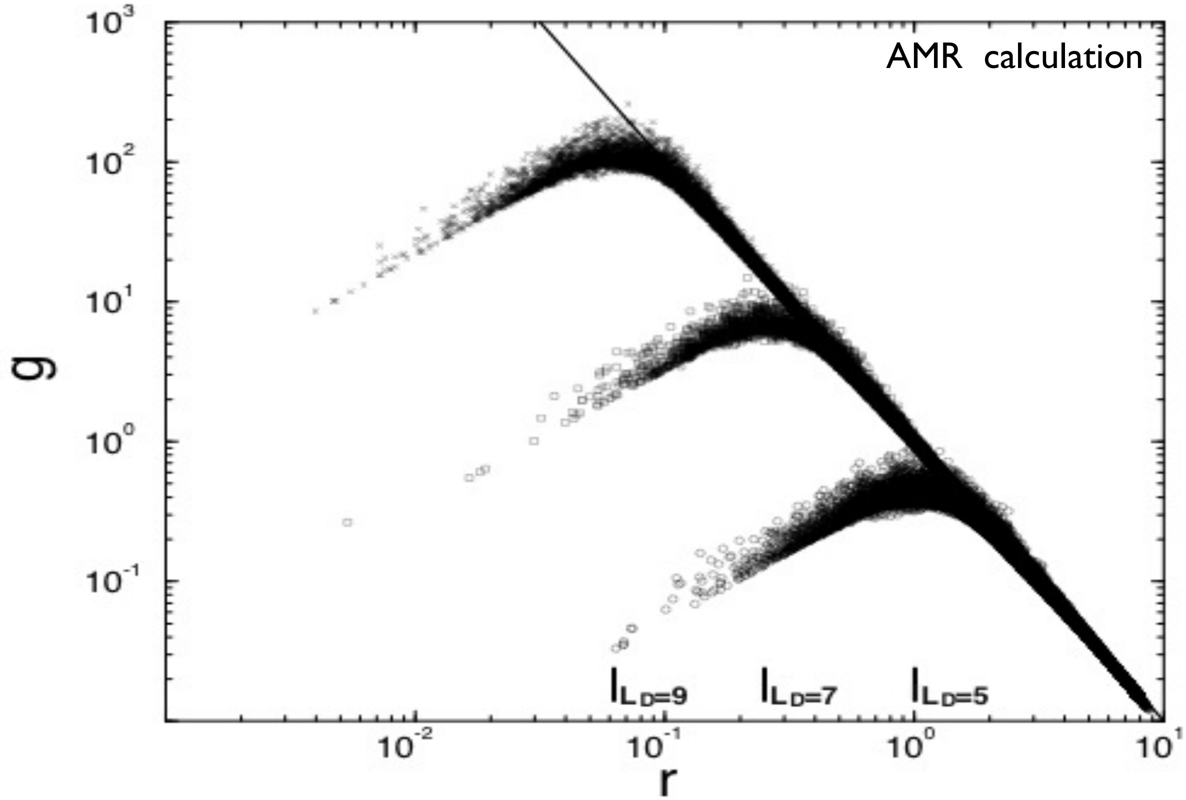
- numerically integrate Poisson's equation



introduction of finer grids in high density regions

Solving for Gravity

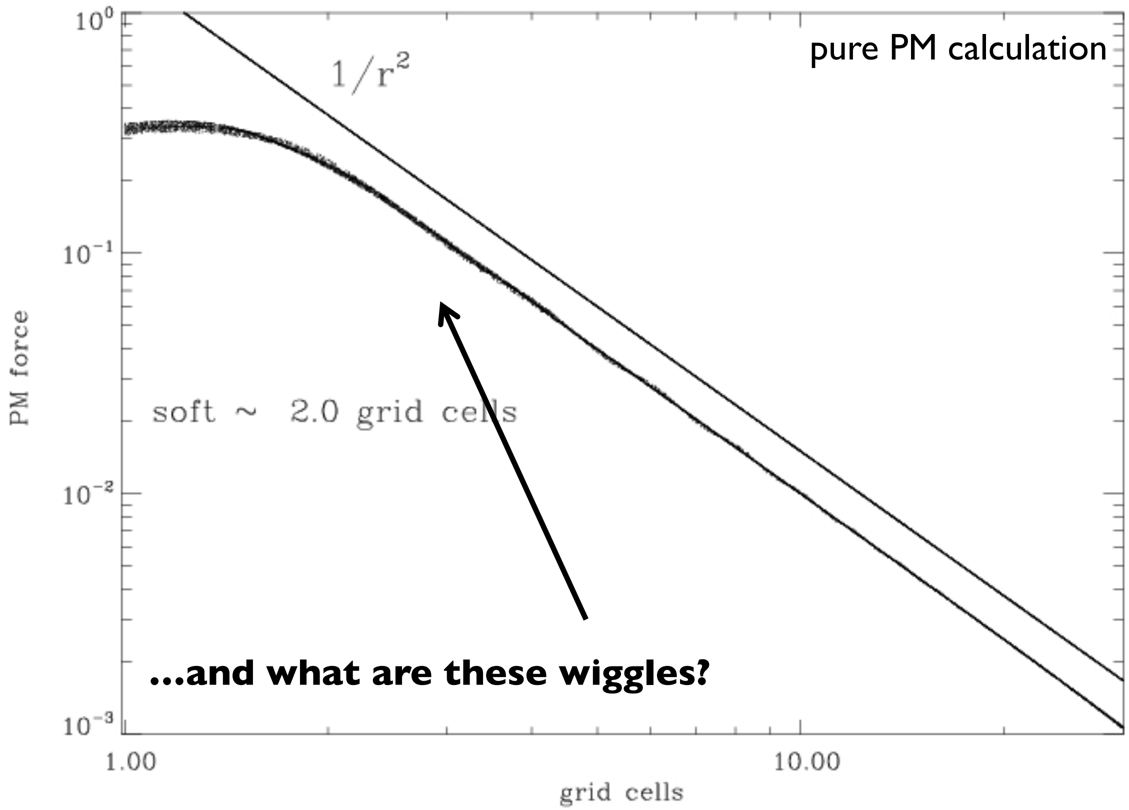
- numerically integrate Poisson's equation



Yahagi & Yoshi (2001)

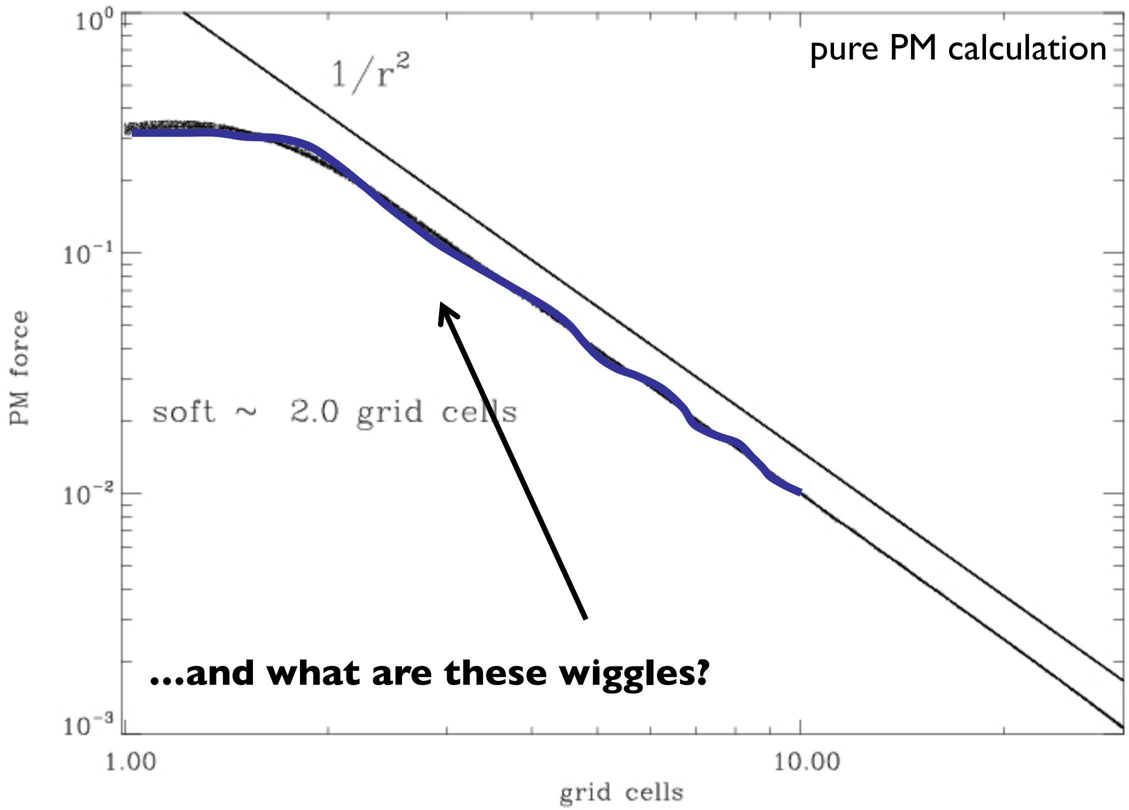
Solving for Gravity

- numerically integrate Poisson's equation



Solving for Gravity

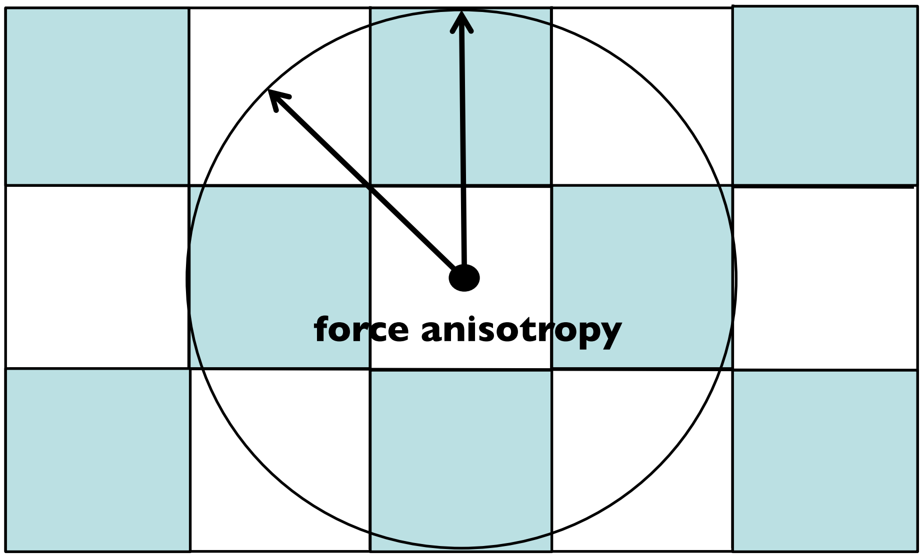
- numerically integrate Poisson's equation



Solving for Gravity

- numerically integrate Poisson's equation

pure PM calculation



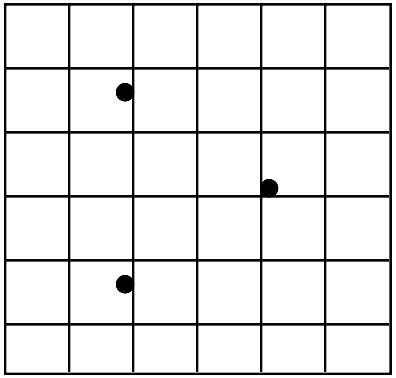
...and what are these wiggles?

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

Solving for Gravity

- obtaining the forces

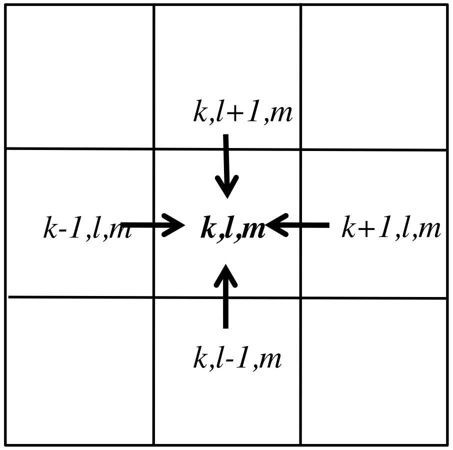
$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$

$$\downarrow$$

$$F_x(\vec{g}_{k,l,m}) = -m\frac{\Phi(\vec{g}_{k+1,l,m}) - \Phi(\vec{g}_{k-1,l,m})}{2H}$$

$$F_y(\vec{g}_{k,l,m}) = -m\frac{\Phi(\vec{g}_{k,l+1,m}) - \Phi(\vec{g}_{k,l-1,m})}{2H}$$

$$F_z(\vec{g}_{k,l,m}) = -m\frac{\Phi(\vec{g}_{k,l,m+1}) - \Phi(\vec{g}_{k,l,m-1})}{2H}$$



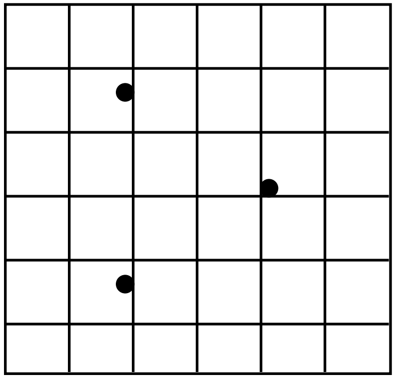
H = (current) grid spacing

Solving for Gravity

- numerically integrate Poisson's equation

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$

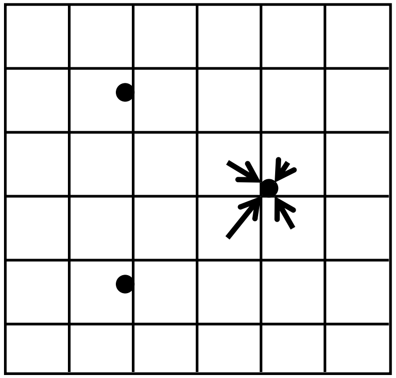


1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

Solving for Gravity

- interpolating the forces

$$\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{r}_i)$$

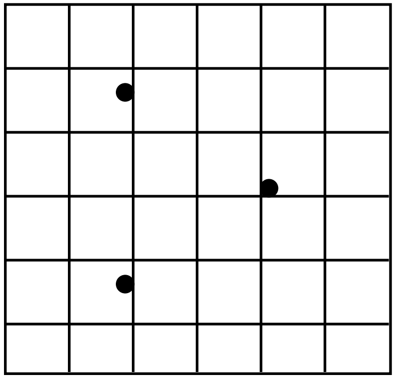


Solving for Gravity

- Particle-Mesh (PM) method

$$\Delta\Phi(\vec{g}_{k,l,m}) = 4\pi G\rho(\vec{g}_{k,l,m})$$

$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

anyone fancies to write a PM code as the project?