Alexander Knebe, *Universidad Autonoma de Madrid*

- 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)

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

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

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

• 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)

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

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

…and the force

• Poisson's equation

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

• collisional matter (e.g. gas)

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

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

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

• 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

- Poisson's equation

$$\vec{F}(\vec{x}) = -m\nabla\Phi(\vec{x})$$

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

Solving for Gravity

- Poisson's equation

$$\vec{F}(\vec{x}) = -m\nabla\Phi(\vec{x})$$

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

particle approach

$$\vec{F}(\vec{x}_i) = -\sum_{i\neq j}\frac{Gm_i m_j}{(x_i - x_j)^3}(\vec{x}_i - \vec{x}_j)$$

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

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

$$\vec{F}(\vec{x}_{i,j,k}) = -m\nabla\Phi(\vec{x}_{i,j,k})$$

Solving for Gravity

- Poisson's equation

$$\vec{F}(\vec{x}) = -m\nabla\Phi(\vec{x})$$

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

**weapon of choice: tree codes**

<u>particle approach</u>

$$\vec{F}(\vec{x}_i) = -\sum_{i\neq j}\frac{Gm_i m_j}{(x_i - x_j)^3}(\vec{x}_i - \vec{x}_j)$$

<u>grid approach</u> ($\vec{x}_{i,j,k}$ =position of centre of grid cell $(i,j,k)$)

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

$$\vec{F}(\vec{x}_{i,j,k}) = -m\nabla\Phi(\vec{x}_{i,j,k})$$

Solving for Gravity

- Poisson's equation

$$\vec{F}(\vec{x}) = -m\nabla\Phi(\vec{x})$$

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

particle approach

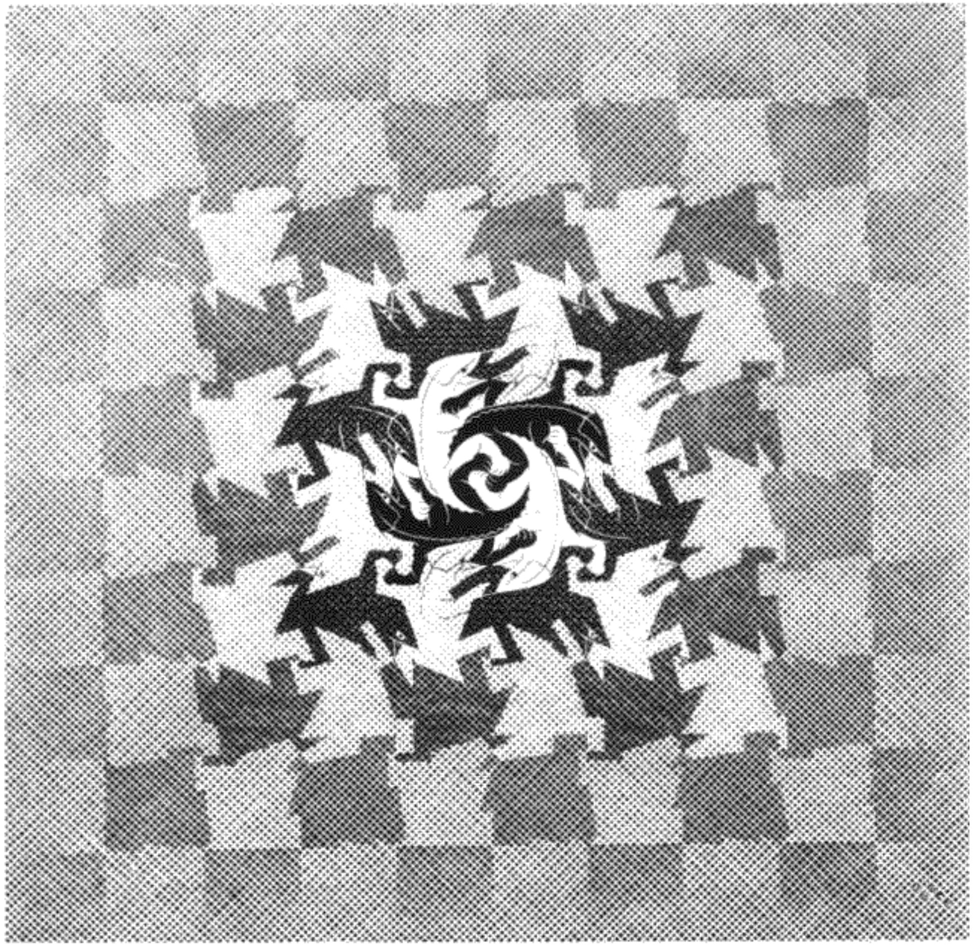
$$\vec{F}(\vec{x}_i) = -\sum_{i\neq j}\frac{Gm_i m_j}{(x_i - x_j)^3}(\vec{x}_i - \vec{x}_j)$$

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

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

$$\vec{F}(\vec{x}_{i,j,k}) = -m\nabla\Phi(\vec{x}_{i,j,k})$$

**weapon of choice: AMR codes**



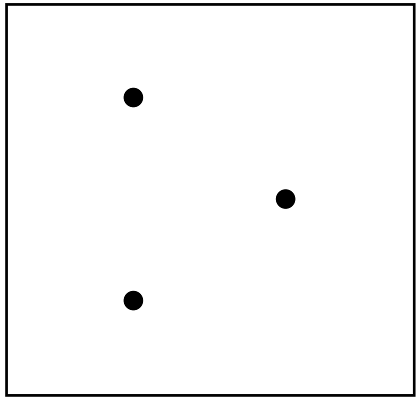
the particle-mesh (PM) method

Solving for Gravity

- numerically integrate Poisson's equation

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

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

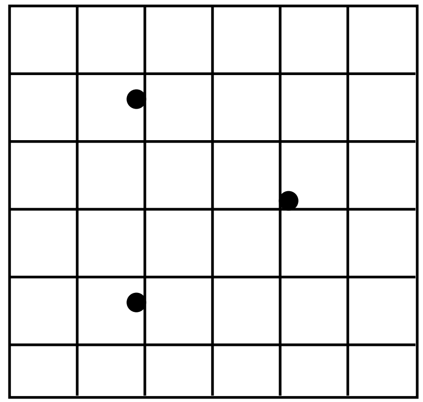


Solving for Gravity

- numerically integrate Poisson's equation

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

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

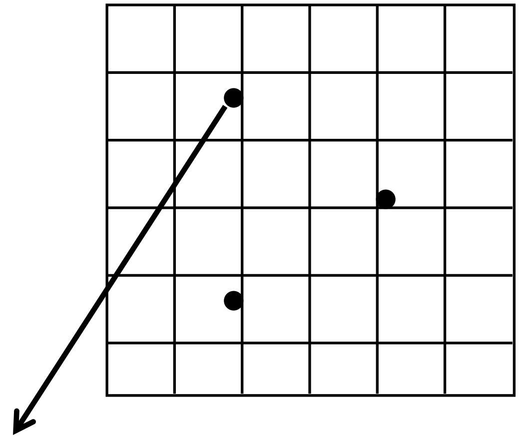


Solving for Gravity

- numerically integrate Poisson's equation

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

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



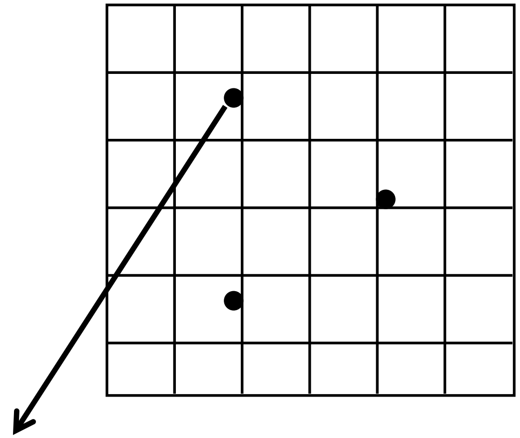
1. calculate mass density on grid $\quad \vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m}) \quad \forall i \in N$

Solving for Gravity

- numerically integrate Poisson's equation

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

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



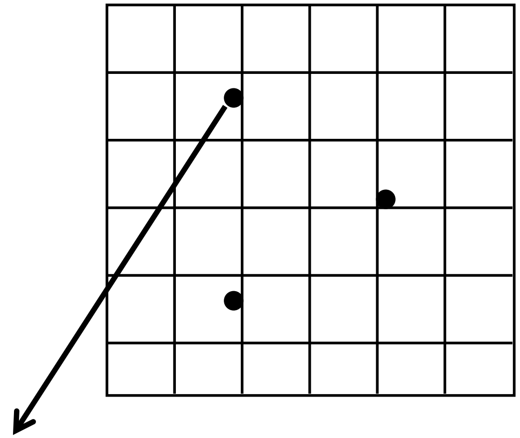
1. calculate mass density on grid $\quad \vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m}) \quad \forall i \in N$
2. solve Poisson's equation on grid $\quad \Phi(\vec{g}_{k,l,m})$

Solving for Gravity

- numerically integrate Poisson's equation

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

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



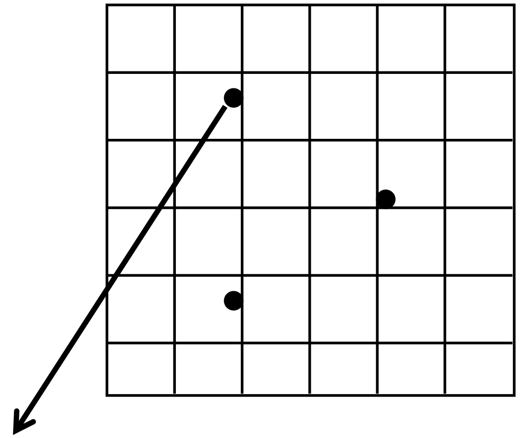
1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$ $\forall i \in N$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$

Solving for Gravity

- numerically integrate Poisson's equation

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

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



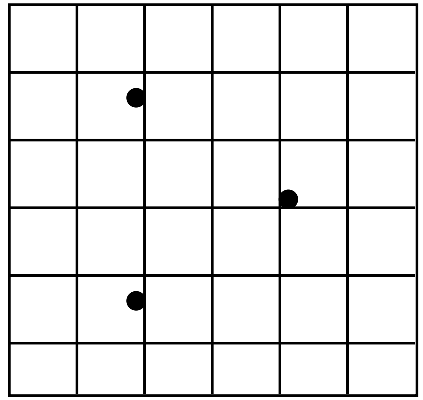
1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m}) \quad \forall i \in N$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow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

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

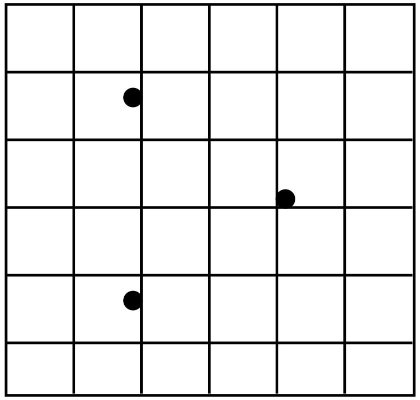
sounds like a waste of time and computer resources,
but **exceptionally fast** in practice

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

Solving for Gravity

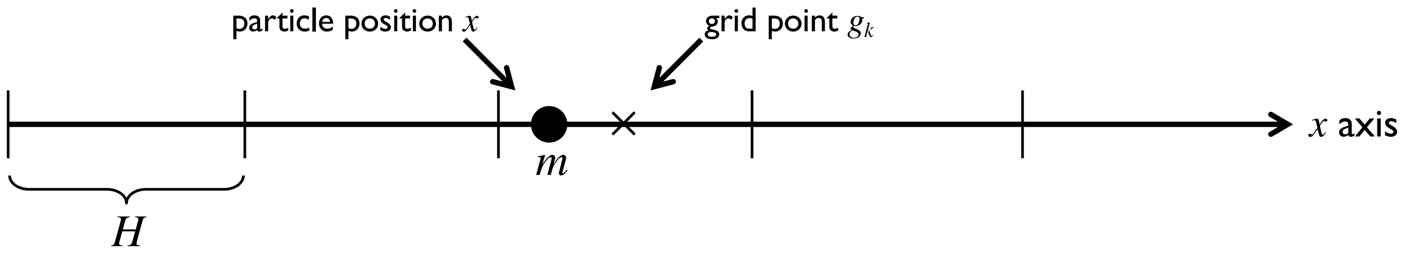
- density assignment schemes

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

<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

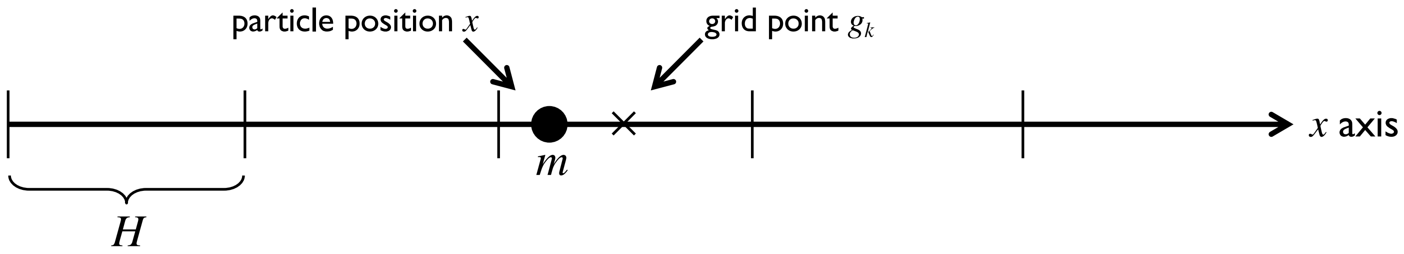
$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

<u>example:</u> 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

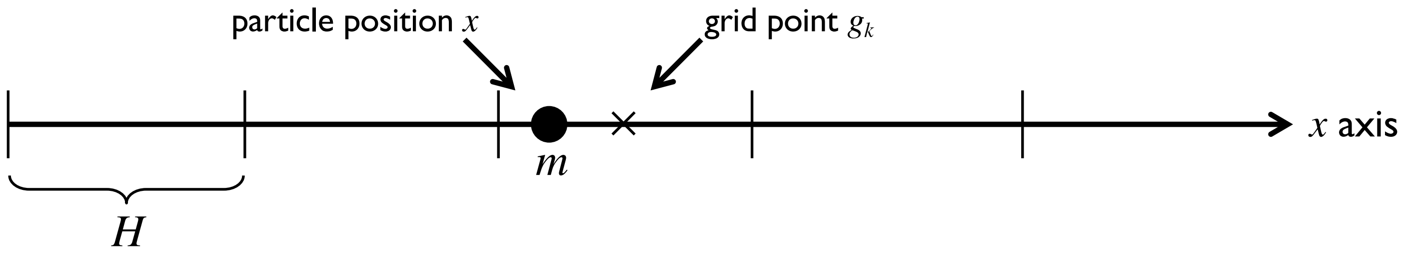
- density assignment schemes

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

<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

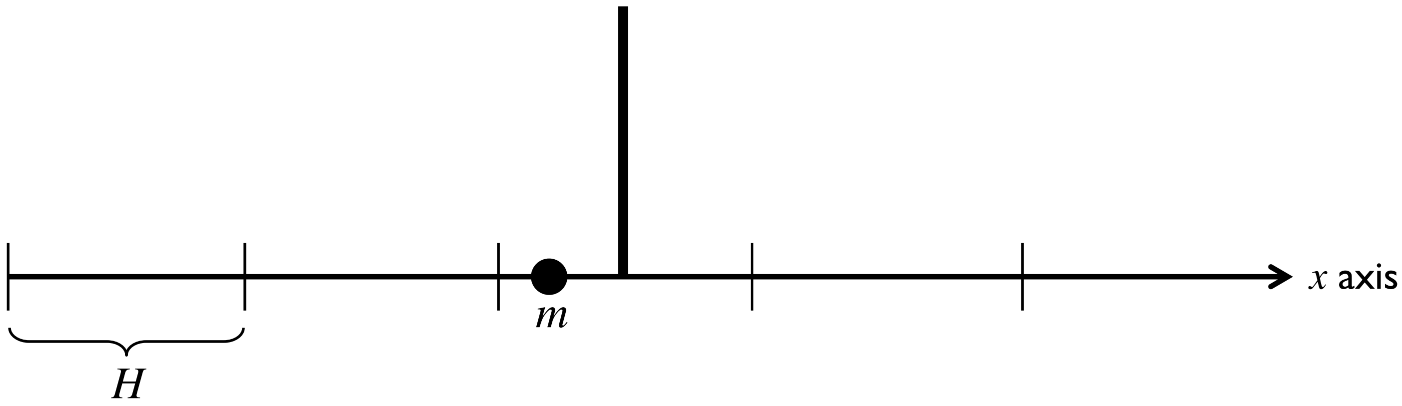
- density assignment schemes

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

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

mass assignment function:

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



Solving for Gravity

- density assignment schemes

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

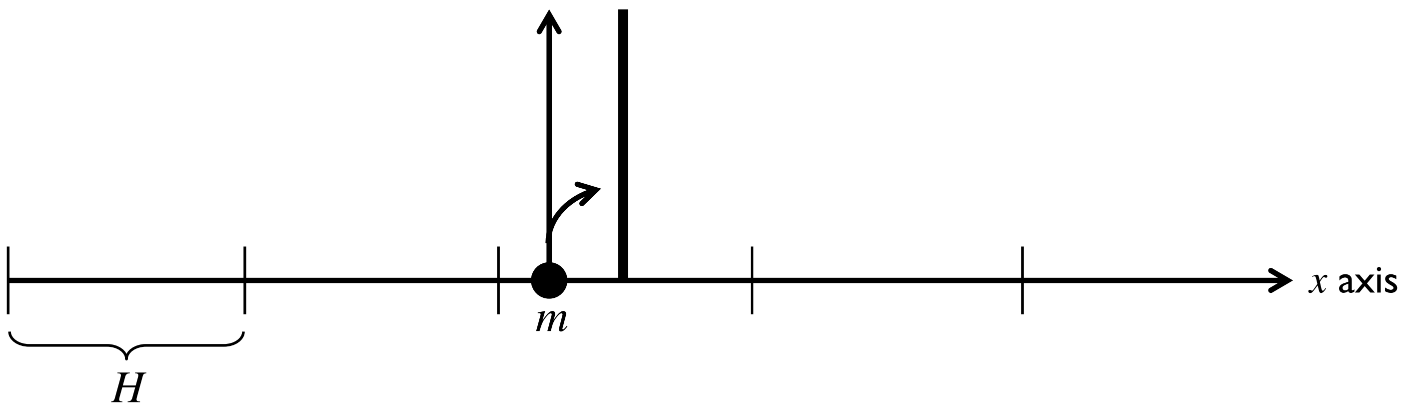
Nearest-Grid-Point (NGP):

particle shape:

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

mass assignment function:

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



Solving for Gravity

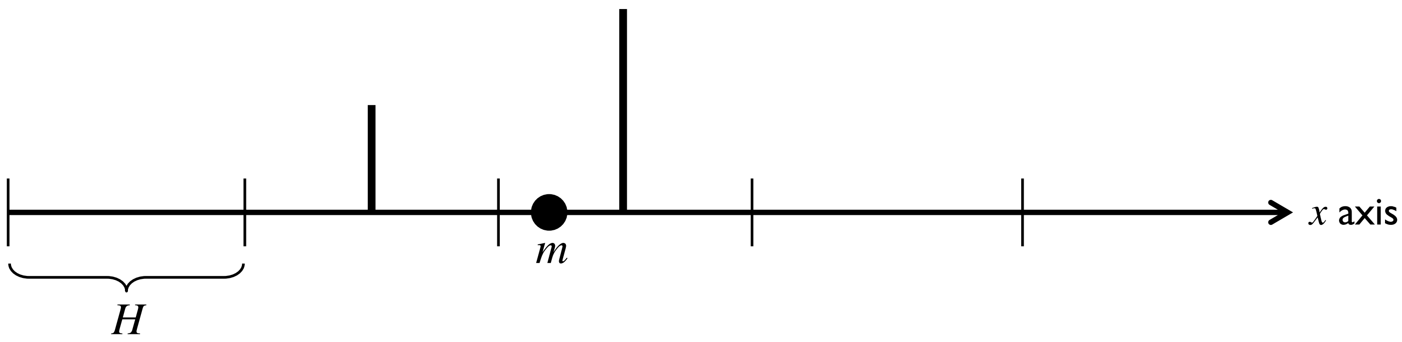
- density assignment schemes

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

Cloud-In-Cell (CIC):

mass assignment function:

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



Solving for Gravity

- density assignment schemes

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

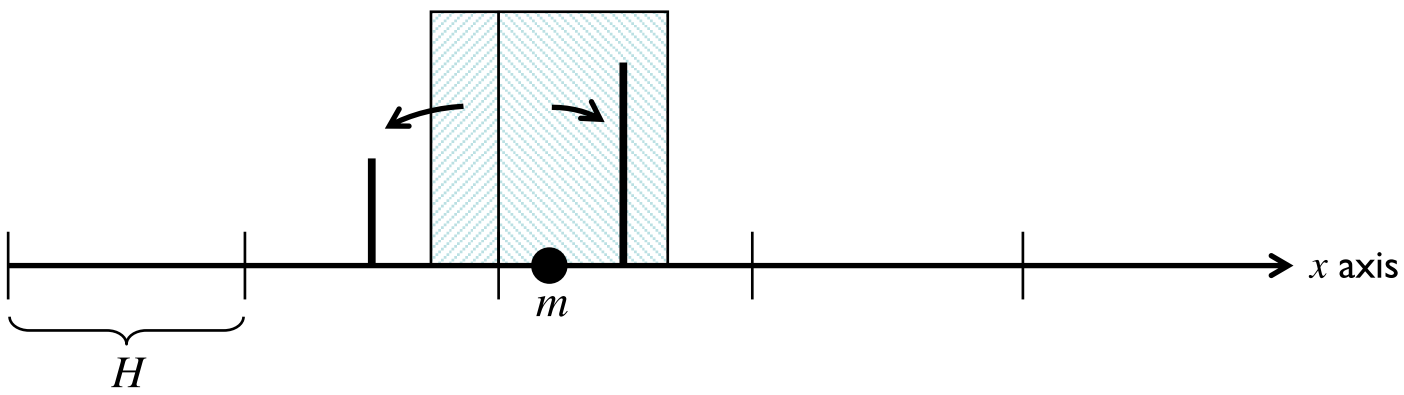
<u>Cloud-In-Cell (CIC):</u>

particle shape:

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

mass assignment function:

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



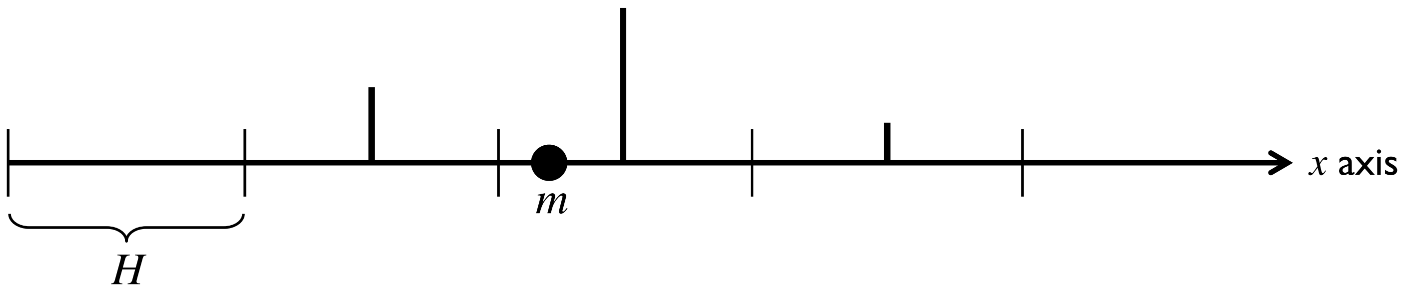
- density assignment schemes

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

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

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

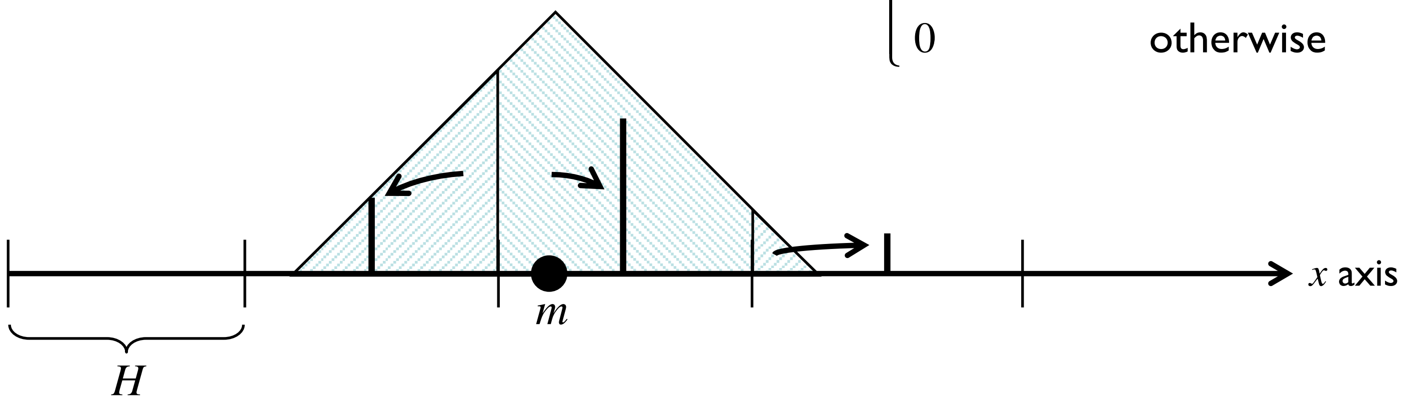
Triangular-Shaped-Cloud (TSC):

particle shape:

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

mass assignment function:

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



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

- density assignment schemes

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

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

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

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

for every grid point we need to loop over all $N$ particles…

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

Solving for Gravity

- density assignment schemes - in practice

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

…rather loop over all particles
and
assign them to the appropriate grid points,
because the mapping $x_i \rightarrow g_k$ is rather easy

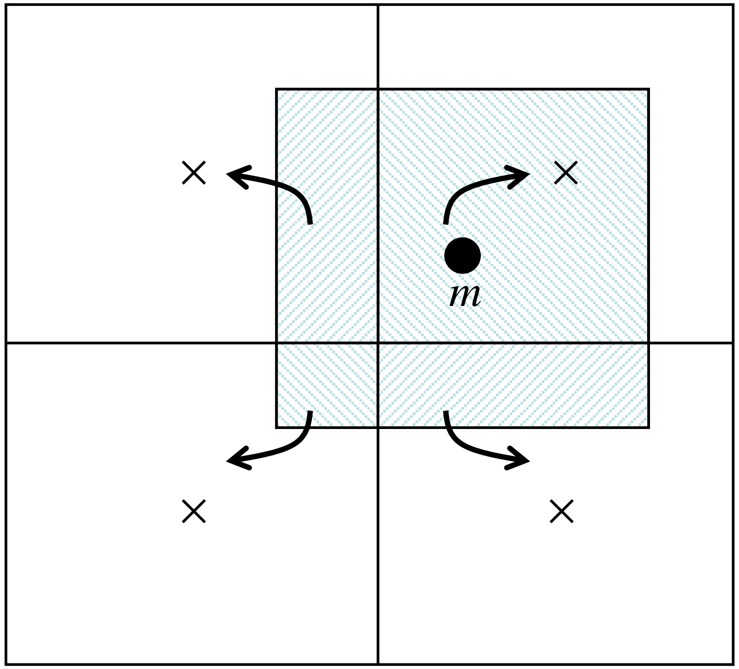
Solving for Gravity

- density assignment schemes - in practice

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

  - example for CIC assignment in 2D:

$\vec{x}_i$ contributes its mass $m_i$ to the 4 closest grid points :

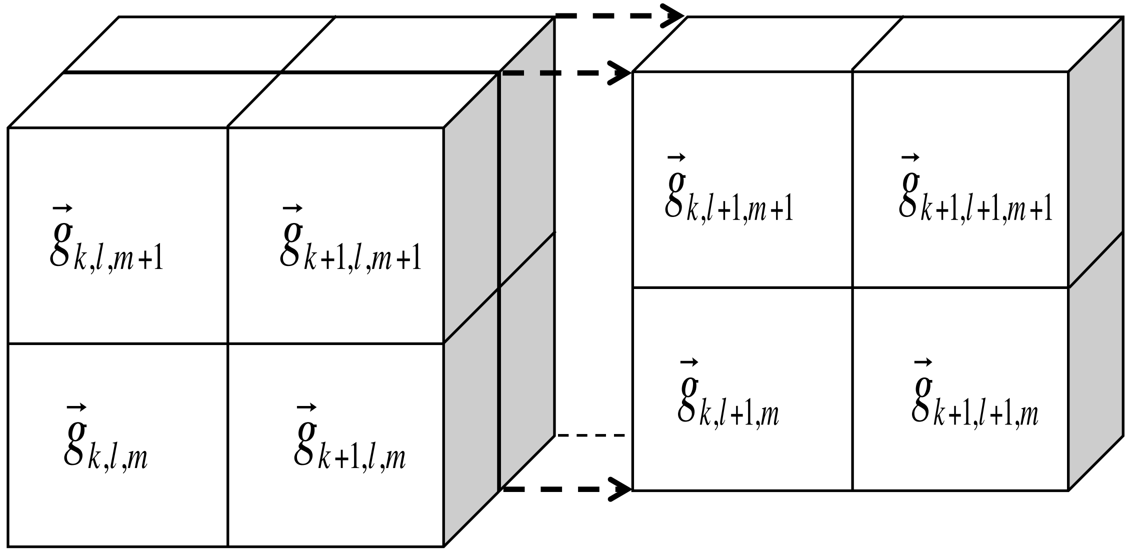


- density assignment schemes - in practice

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

- example for CIC assignment in 3D:

$\vec{x}_i$ contributes its mass $m_i$ to the 8 closest grid points :



Solving for Gravity

- density assignment schemes
  - which scheme to choose?

| | |
|---|---|
| NGP = stepwise force | (1 grid point) |
| CIC = continuous piecewise linear force | (8 grid points) |
| TSC = continuous force and first derivative | (27 grid points) |

Solving for Gravity

- density assignment schemes
  - which scheme to choose?

NGP = too crude

CIC = common choice

TSC = pretty smooth

increased smoothing of density field

Solving for Gravity

- density assignment schemes

    - which scheme to choose?

NGP = too crude

CIC = common choice

TSC = pretty smooth

increased smoothing of density field

smoothing the density field will lead to a "bias" in the forces
but at the same time decrease the "variance"!

$$\text{bias} = \left( \left\langle \vec{F}(\vec{x}) \right\rangle - \vec{F}_{true}(\vec{x}) \right) \propto \varepsilon^{\alpha}$$

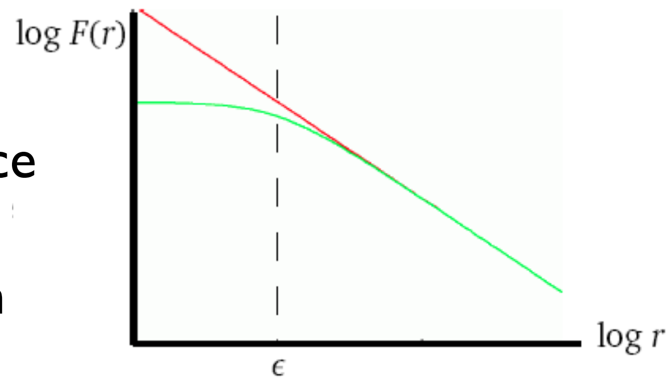
$$\text{var} = \left\langle \vec{F}^2(\vec{x}) \right\rangle - \left\langle \vec{F}(\vec{x}) \right\rangle^2 \propto N^{-\beta}$$

- density assignment schemes
    - which scheme to choose?

NGP = too crude

CIC = common choice

TSC = pretty smooth




smoothing the density field will lead to a "bias" in the forces but at the same time decrease the "variance"!

$$\text{bias} = \left(\left\langle \vec{F}(\vec{x}) \right\rangle - \vec{F}_{true}(\vec{x})\right) \propto \varepsilon^{\alpha}$$

$$\text{var} = \left\langle \vec{F}^2(\vec{x}) \right\rangle - \left\langle \vec{F}(\vec{x}) \right\rangle^2 \propto N^{-\beta}$$

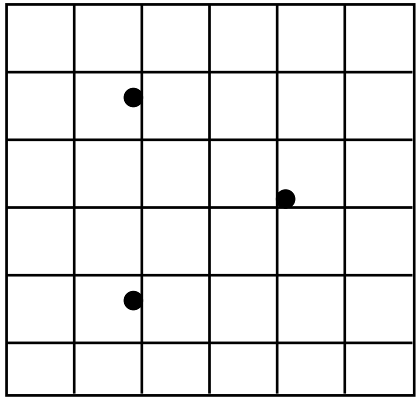
(interplay between $N$ and $\varepsilon$ : $N\varepsilon^3$=const.)

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

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

$$\Delta\Phi = \rho$$

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

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

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 *fast fourier transform method*
  - discretized Green's function

$$\hat{G}(\vec{k}) = -\frac{1}{k^2} \longrightarrow \hat{G}(\vec{g}_{k,l,m}) = -\frac{1}{\sin^2\left(\frac{k_x}{2}\right) + \sin^2\left(\frac{k_y}{2}\right) + \sin^2\left(\frac{k_z}{2}\right)}$$

*compensates for grid anisotropies...*

$$\hat{G}(\vec{g}_{0,0,0}) = 0, \quad k_x = \frac{2\pi\, k}{L}, \quad k_y = \frac{2\pi\, l}{L}, \quad k_y = \frac{2\pi\, m}{L}$$

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

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

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

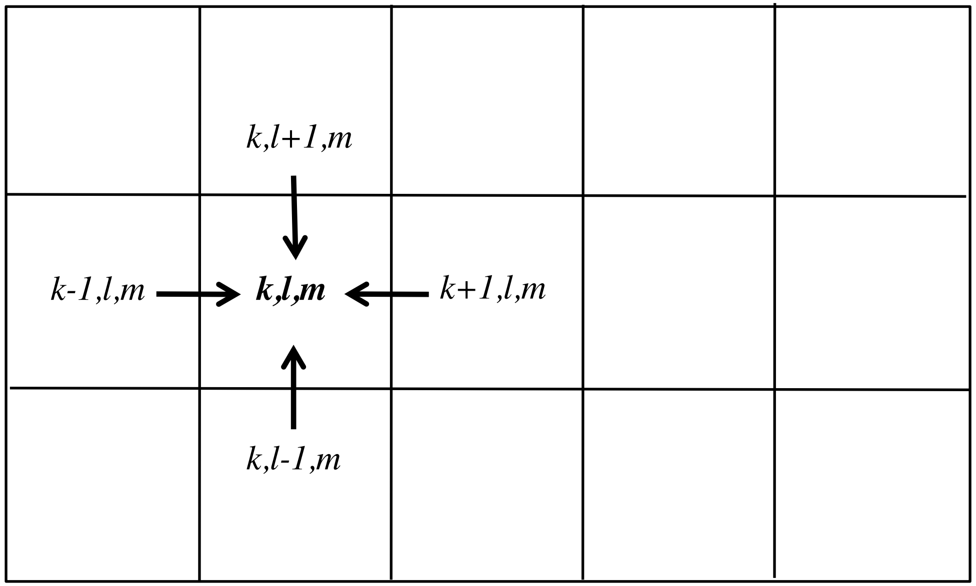
discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

*obtain iterative solver by discretizing differential equation*



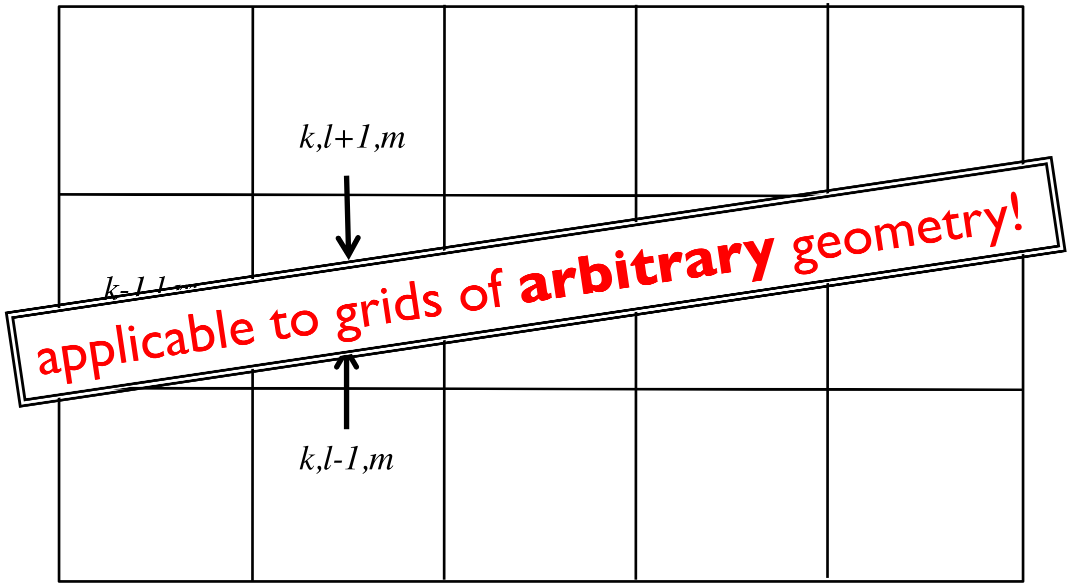

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*

*obtain iterative solver by discretizing differential equation*

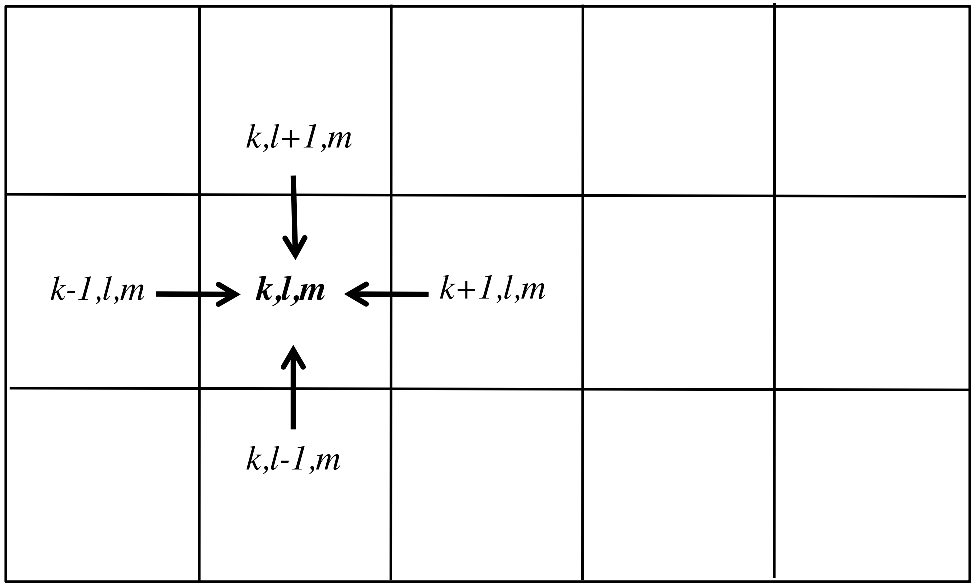



discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
    - how to sweep through the grid?

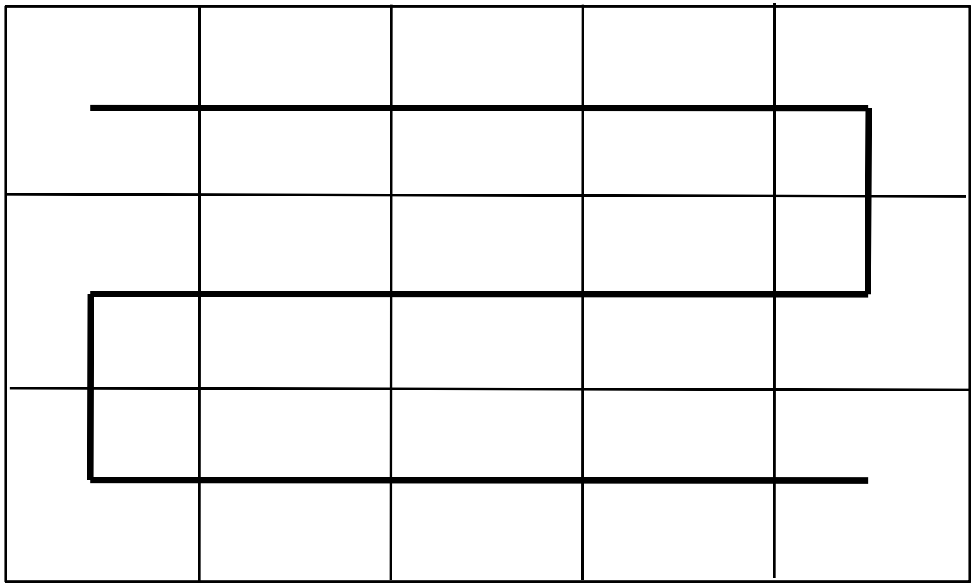



discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - how to sweep through the grid?



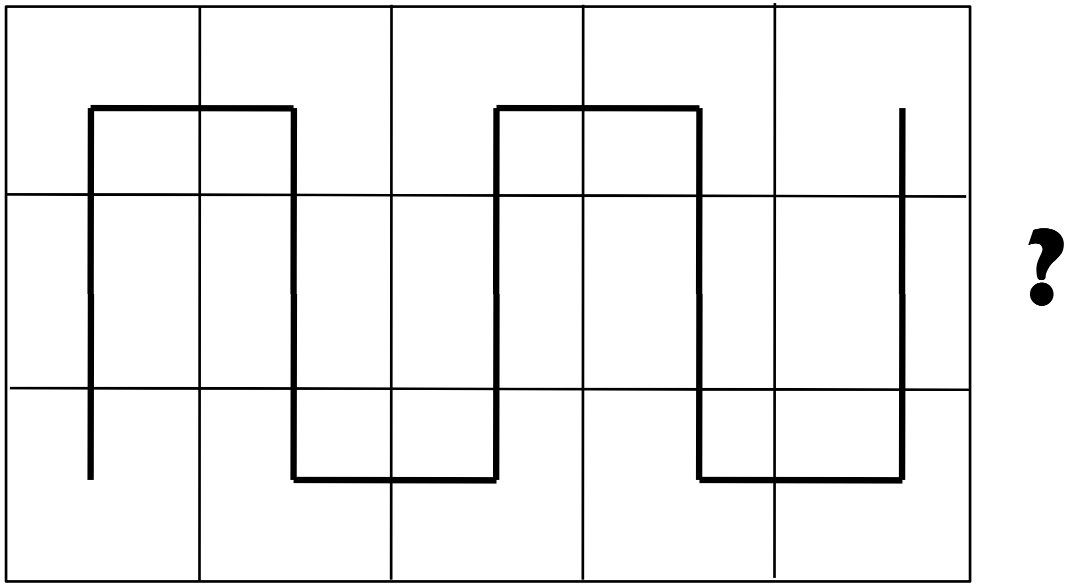
?

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
    - how to sweep through the grid?

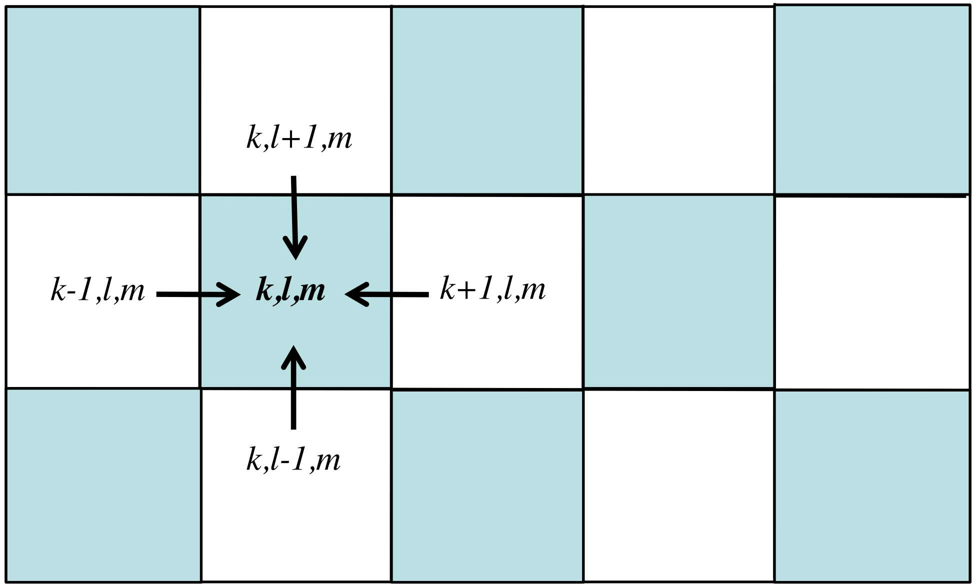



discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - Gauss-Seidel sweeps:




discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - Gauss-Seidel sweeps:
    - loop over all "black" cells
    - loop over all "red" cells

one iteration of the potential

$$\Phi^i_{k,l,m} \rightarrow \Phi^{i+1}_{k,l,m}$$

how many iterations $i$ are necessary?

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$\Delta\Phi^{i}_{k,l,m} = \rho_{k,l,m}$$

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
    - stopping criterion:

$$\Delta\Phi^{i}_{k,l,m} - \rho_{k,l,m} \xrightarrow{?} 0$$

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$\Delta\Phi^{i}_{k,l,m} - \rho_{k,l,m} \xrightarrow{?} 0$$

**density as given by currently best guess for $\Phi^i$!**

**density as given by mass assignment scheme!**

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$\Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \xrightarrow{?} 0$$

residual: $R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\|$ $\quad \|\bullet\|$ = suitable norm

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$\Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \xrightarrow{?} 0$$

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \leq \varepsilon \mathrm{T}$$

$\|\bullet\|$ = suitable norm

tolerance

error estimate

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \leq \varepsilon \mathrm{T}$$

– truncation error:

error due to discreteness of grid

discretized Poisson's equation

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

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \leq \varepsilon \mathrm{T}$$

– truncation error:

error due to discreteness of grid

$\xrightarrow{\text{estimation}}$ compare solution on actual grid to solution on coarser grid

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \le \varepsilon \mathrm{T} = \varepsilon \left\| T_{k,l,m} \right\|$$

– truncation error: $\mathrm{T}_{k,l,m} = \underbrace{\mathcal{P}\left[\Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right)\right]} - \left(\Delta\Phi^i_{k,l,m}\right)$

$$\mathcal{R}\Phi^i_{k,l,m} = \boldsymbol{\Phi}^i_{j,n,p}$$ restriction to coarser grid

$$\Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right) = \rho^{i-1}_{j,n,p}$$

$$\mathcal{P}\left[\Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right)\right] = \rho^i_{k,l,m}$$ prolongation to finer grid

discretized Poisson's equation

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

- numerically integrate Poisson's equation *relaxation technique*

  - stopping criterion:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \leq \varepsilon \mathrm{T} = \varepsilon \left\| T_{k,l,m} \right\|$$

– truncation error: $\mathrm{T}_{k,l,m} = \mathcal{P}\left[\Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right)\right] - \underbrace{\left(\Delta\Phi^i_{k,l,m}\right)}$

$$\begin{aligned} \mathcal{R}\Phi^i_{k,l,m} &= \boldsymbol{\Phi}^i_{j,n,p} \\ \Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right) &= \rho^{i-1}_{j,n,p} \\ \mathcal{P}\left[\Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right)\right] &= \rho^i_{k,l,m} \stackrel{?}{=} \Delta\Phi^i_{k,l,m} \end{aligned}$$

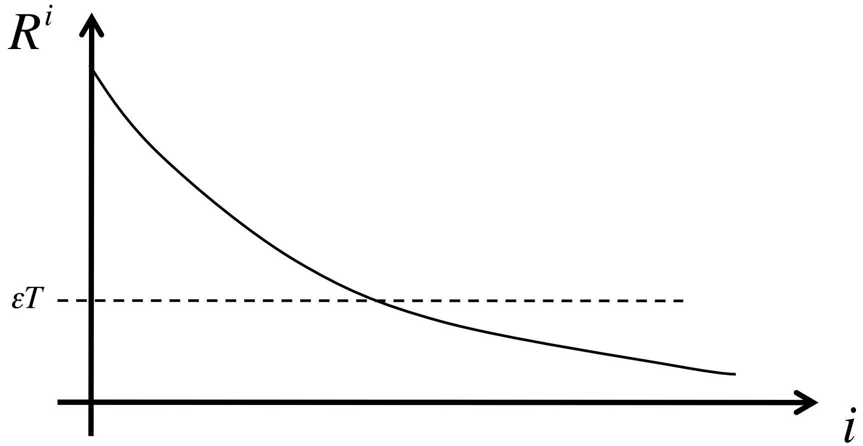
discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
    - stopping criterion:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \le \varepsilon \mathrm{T} = \varepsilon \left\| T_{k,l,m} \right\|$$



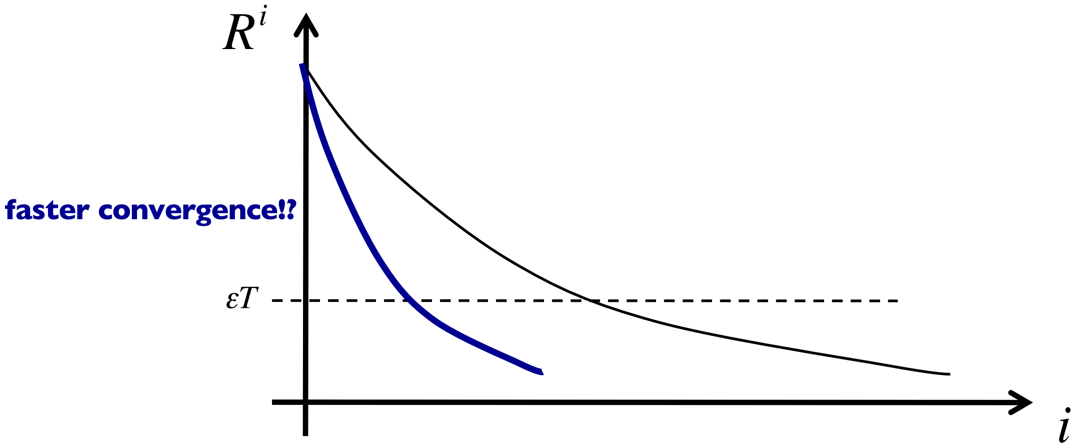

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - stopping criterion:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\| \leq \varepsilon \mathrm{T} = \varepsilon \left\| T_{k,l,m} \right\|$$




discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - convergence:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\|$$

– slow convergence: $R^{i+1} \approx R^i$

large-scale errors in $\Phi$ cannot be "relaxed"
sufficiently fast on the actual grid

=> use coarser grids to speed up convergence…

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation *relaxation technique*
  - convergence:

$$R^i = \left\| \Delta\Phi^i_{k,l,m} - \rho_{k,l,m} \right\|$$

– slow convergence: $R^{i+1} \approx R^i$

**multi-grid relaxation techniques**

=> beyond the scope of this lecture though…

discretized Poisson's equation

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

Solving for Gravity

- numerically integrate Poisson's equation

accuracy of either

*relaxation*

or

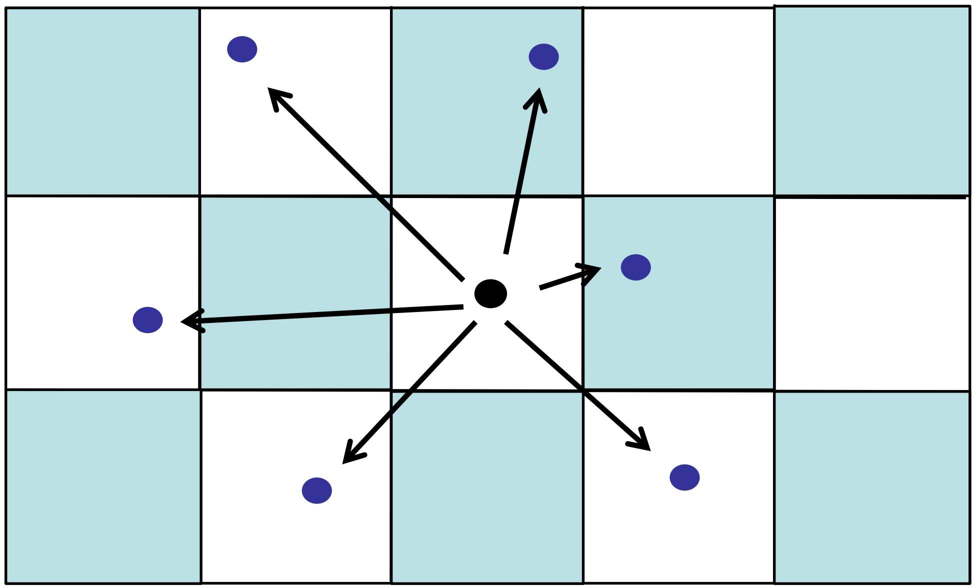
*FFT method*

to solve Poisson's equation?

Solving for Gravity

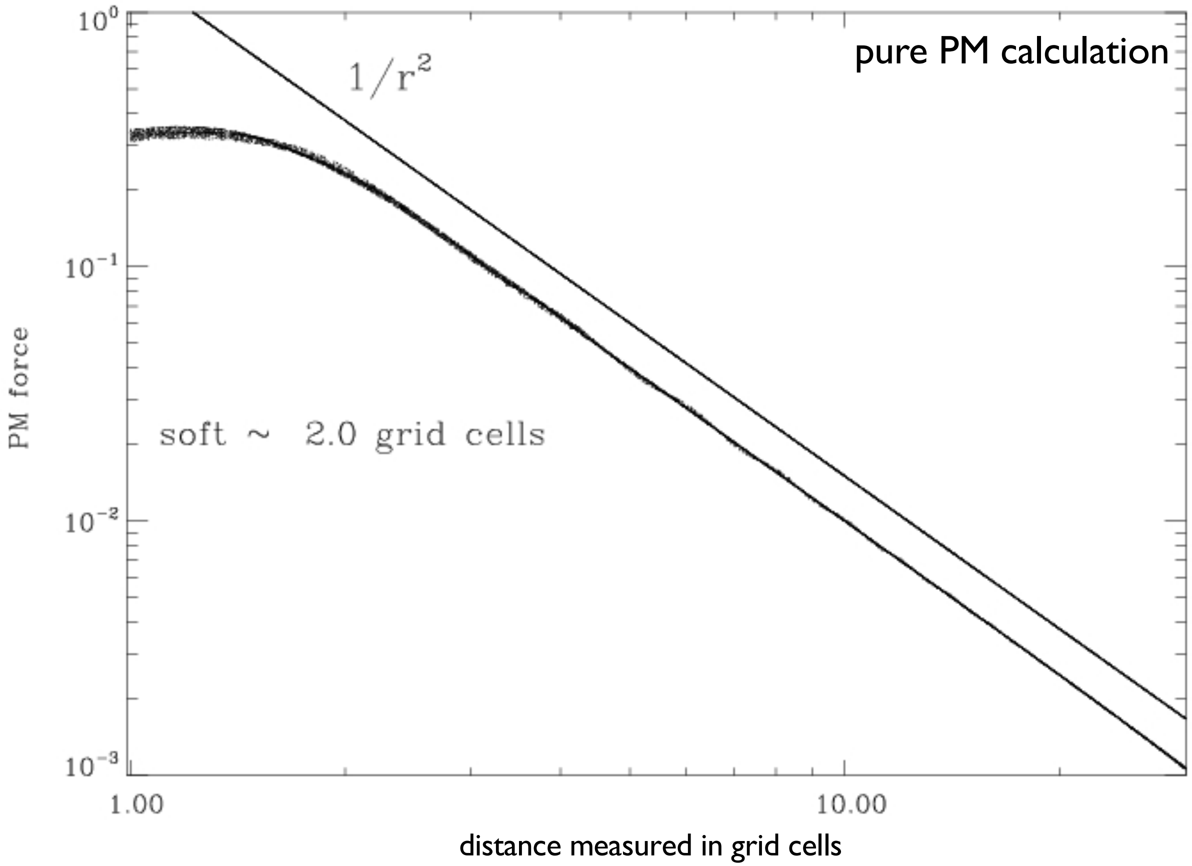
- numerically integrate Poisson's equation

pure PM calculation



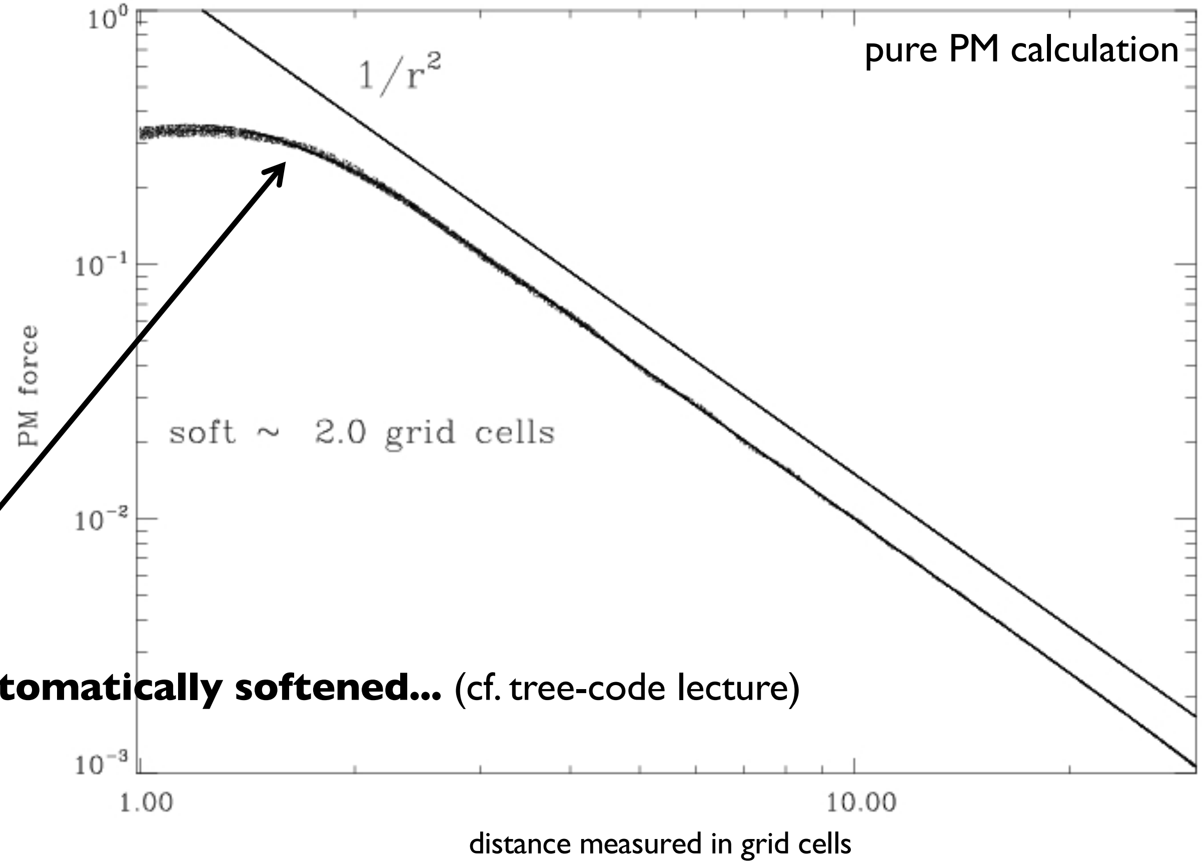
Solving for Gravity

- numerically integrate Poisson's equation



Solving for Gravity

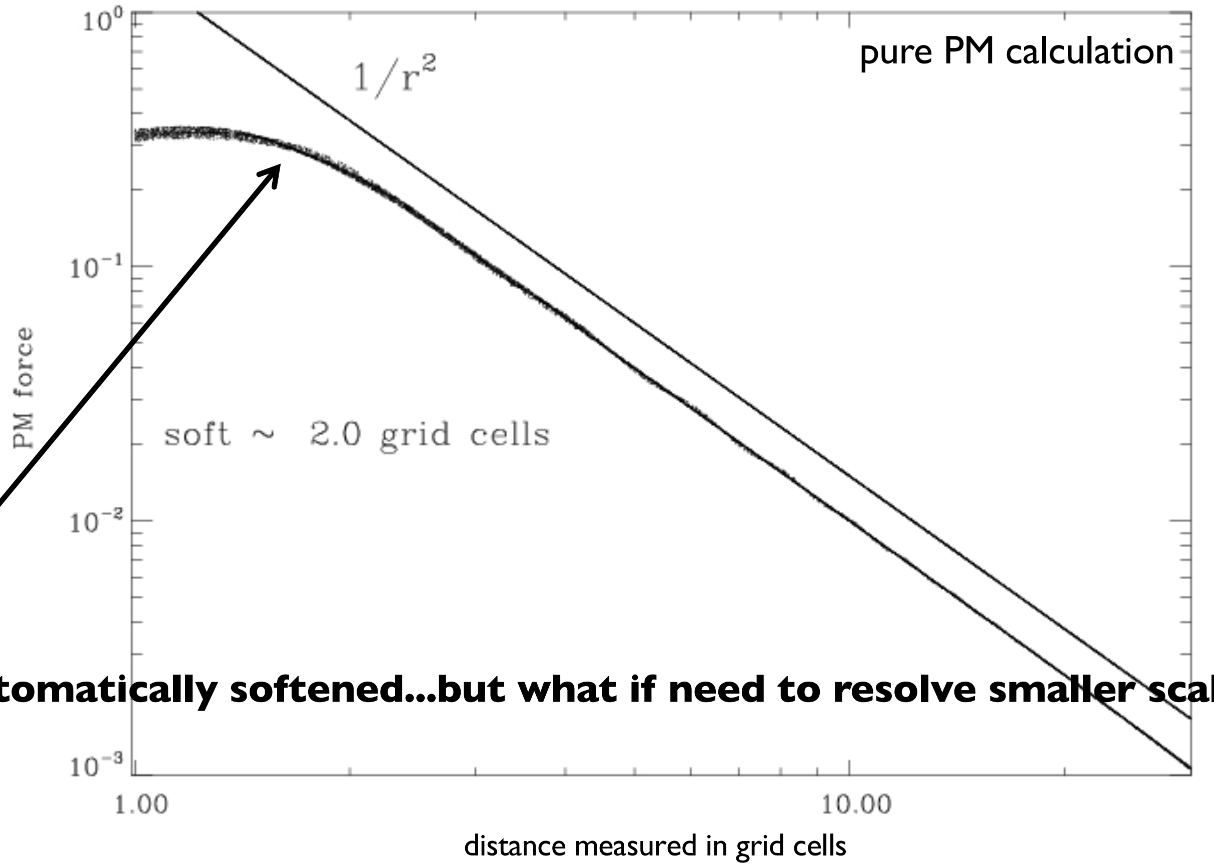
- numerically integrate Poisson's equation




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

Solving for Gravity

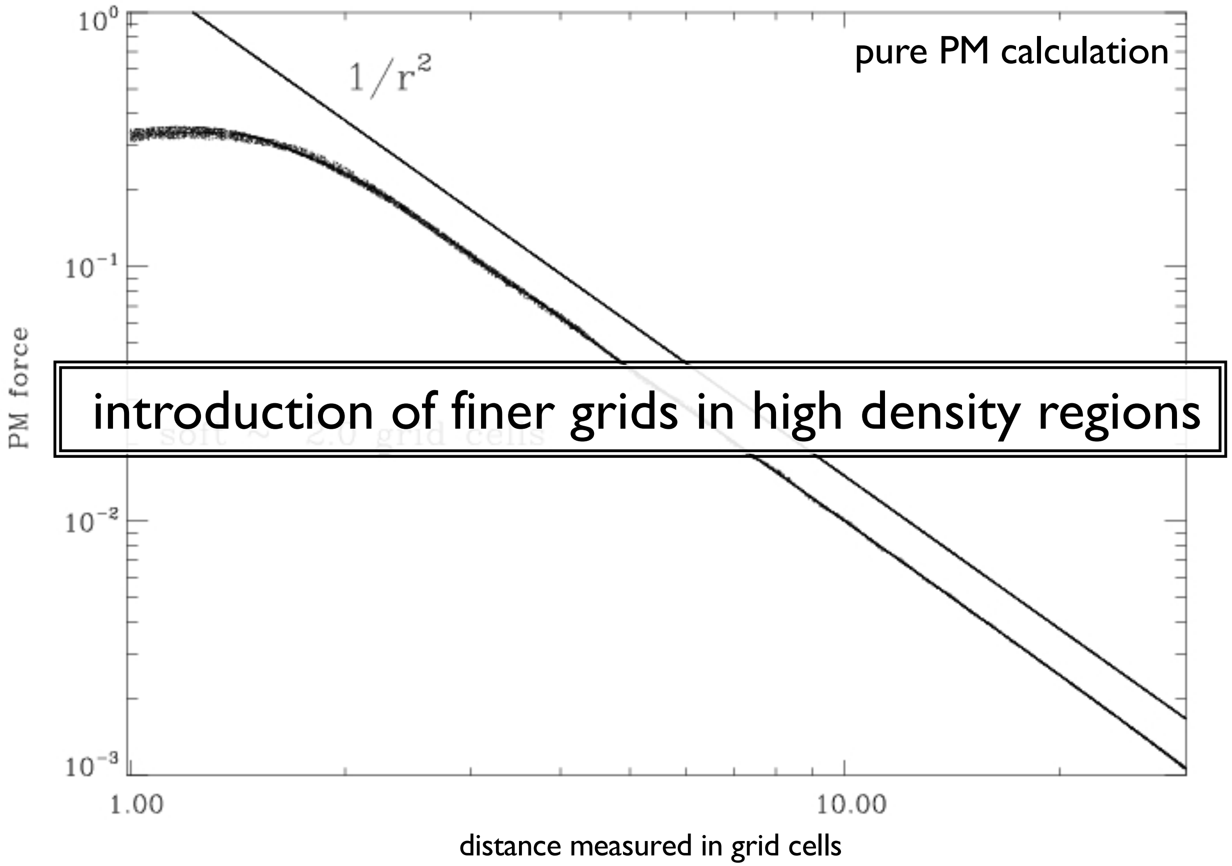
- numerically integrate Poisson's equation




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

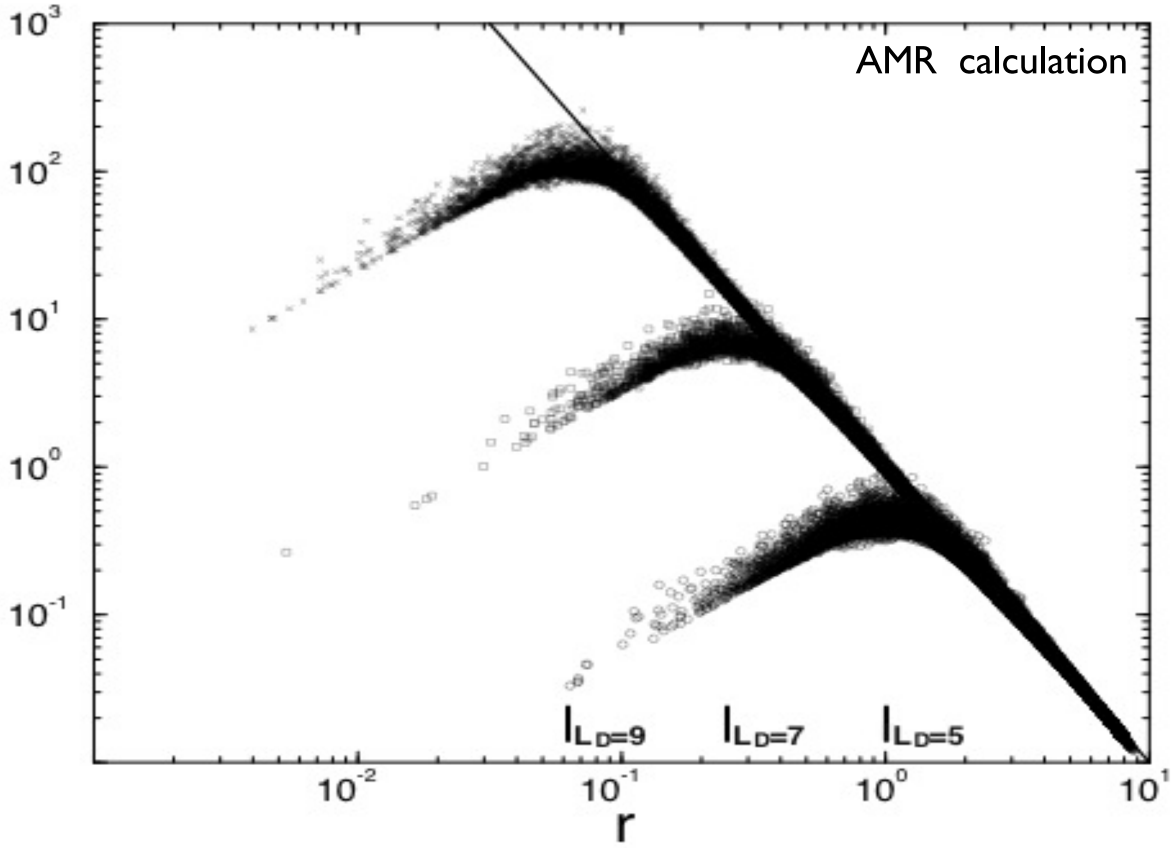
Solving for Gravity

- numerically integrate Poisson's equation



Solving for Gravity

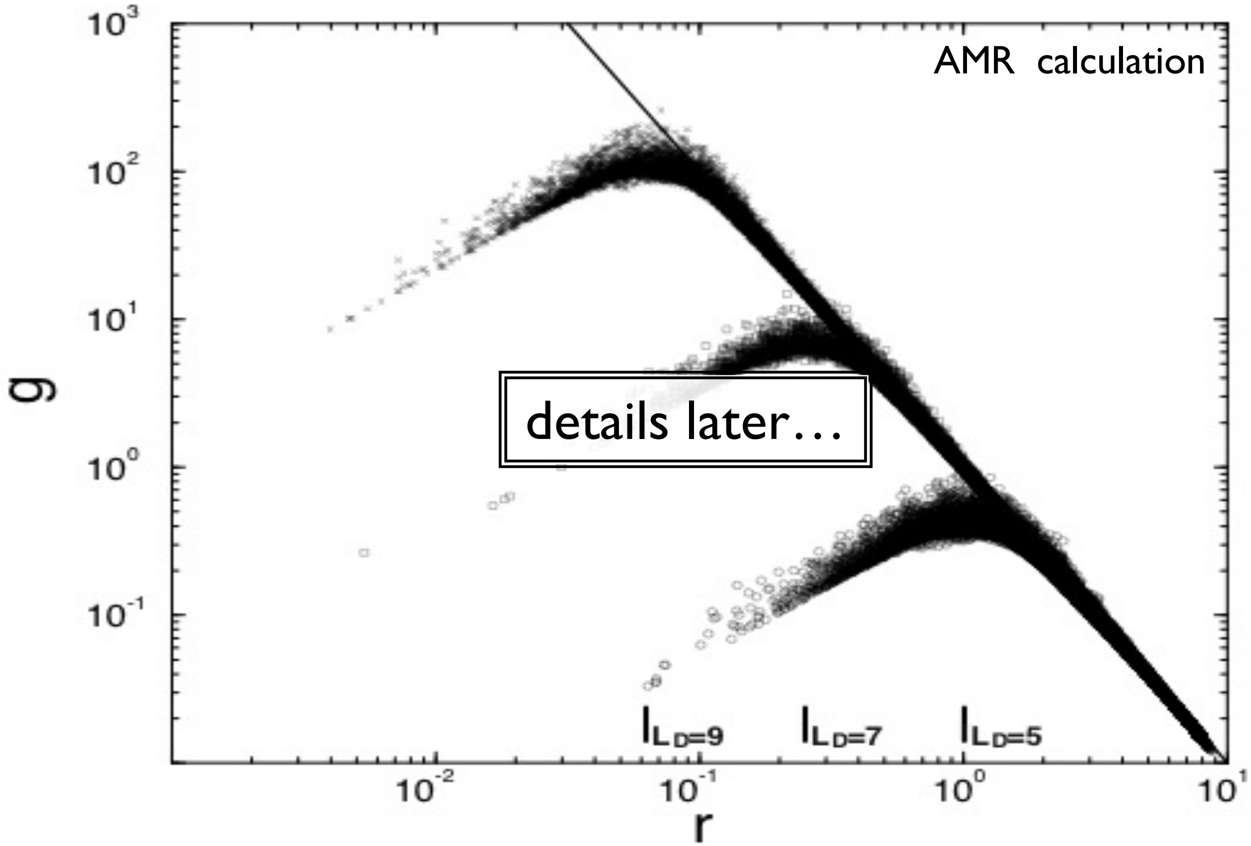
- numerically integrate Poisson's equation




Yahagi & Yoshi (2001)

Solving for Gravity

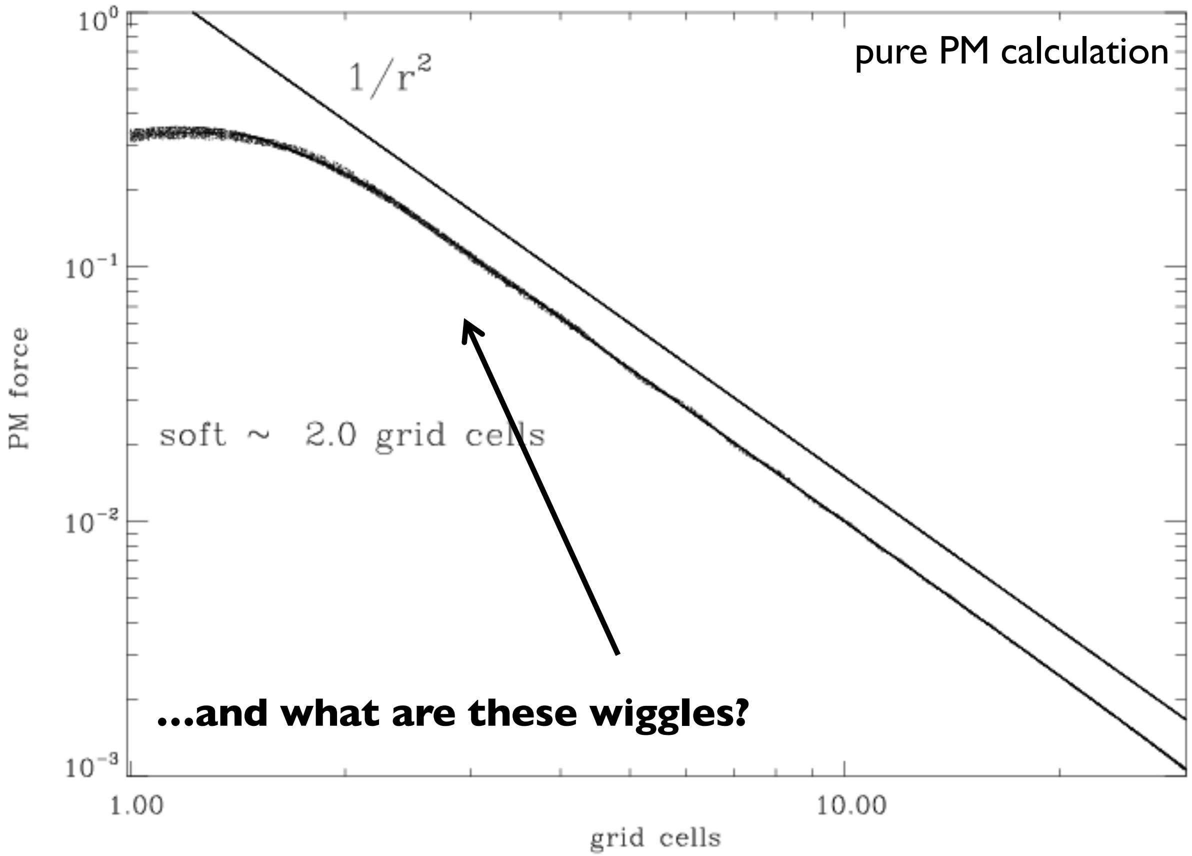
- numerically integrate Poisson's equation




Yahagi & Yoshi (2001)

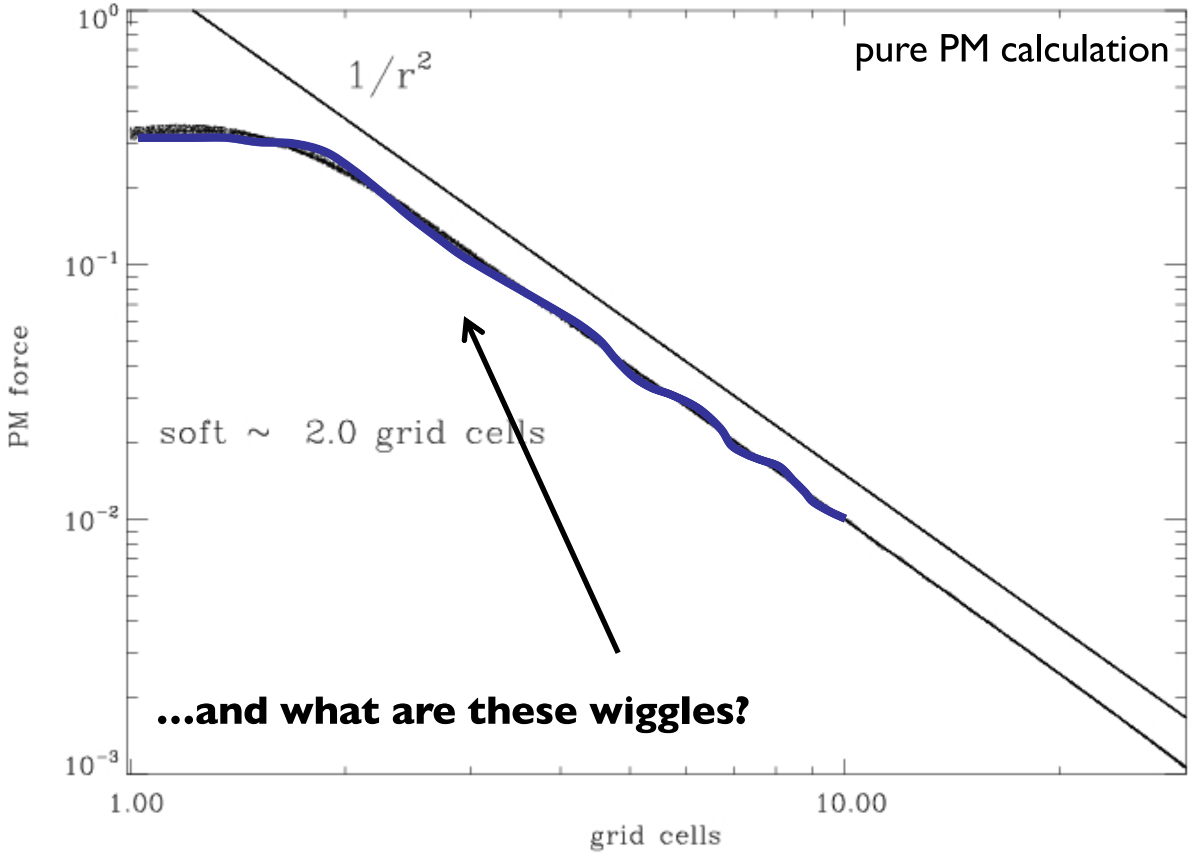
Solving for Gravity

- numerically integrate Poisson's equation



Solving for Gravity

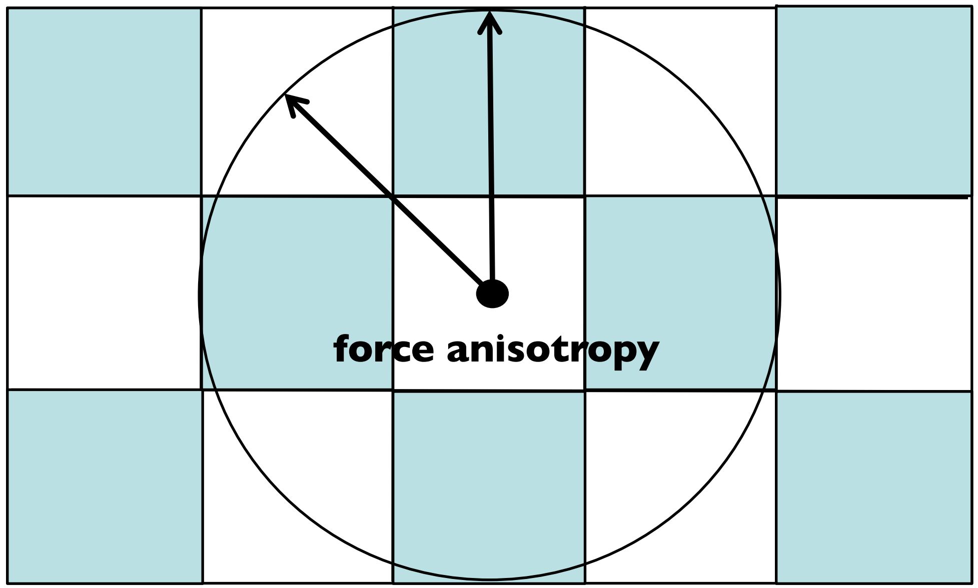
- numerically integrate Poisson's equation



Solving for Gravity

- numerically integrate Poisson's equation

pure PM calculation



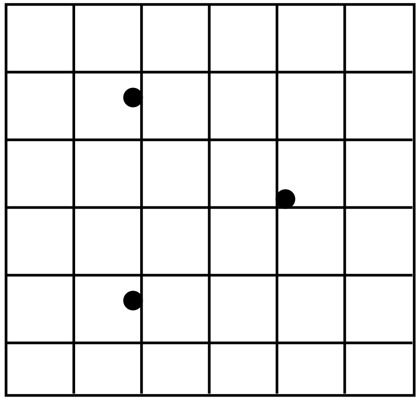

**...and what are these wiggles?**

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

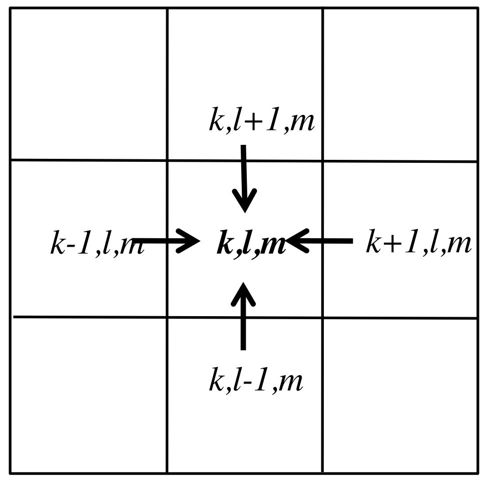
$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$

$$\downarrow$$

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

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

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



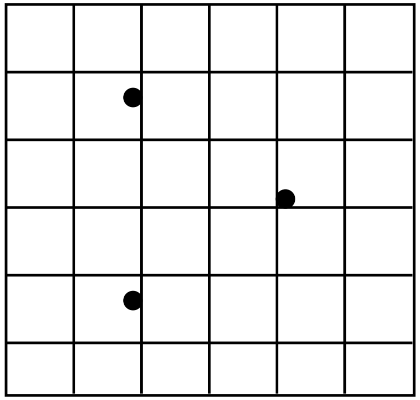

$H$ = (current) grid spacing

Solving for Gravity

- numerically integrate Poisson's equation

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

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

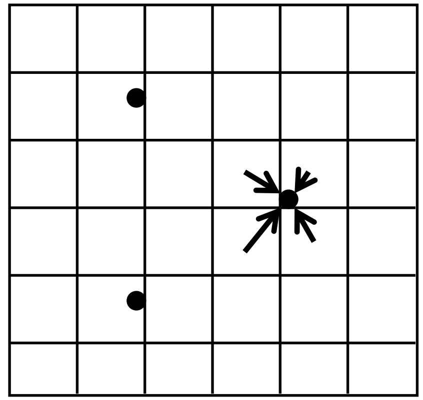


1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

Solving for Gravity

- interpolating the forces

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



Solving for Gravity

- interpolating the forces

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

use the inverse of the mass assignment scheme
to insure momentum conservation and minimize force anisotropies

$$\vec{F}(\vec{r}_i) = \sum_k \sum_l \sum_m \vec{F}(\vec{g}_{k,l,m}) W(|\vec{r}_i - \vec{g}_{k,l,m}|)$$

in practice the triple sum is "only" over 8 (CIC) or 27 (TSC) cells…

Solving for Gravity

- interpolating the forces

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

use the inverse of the mass assignment scheme
to insure momentum conservation and minimize force anisotropies

$$\vec{F}(\vec{r}_i) = \sum_k \sum_l \sum_m \vec{F}(\vec{g}_{k,l,m}) W(|\vec{r}_i - \vec{g}_{k,l,m}|)$$

---

*check by calculating the total (periodic) force:

$$\begin{aligned}
\vec{F}_{tot} &= \sum_{i=1}^{N} \vec{F}(\vec{r}_i) = \sum_{i=1}^{N} \sum_k \sum_l \sum_m \vec{F}(\vec{g}_{k,l,m}) W(\vec{r}_i - \vec{g}_{k,l,m}) \\
&= \ldots \\
&= \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{k,l,m} \sum_{k',l',m'} \frac{m_i m_j}{H^3} W(|\vec{r}_i - \vec{g}_{k,l,m}|) \mathcal{G}(\vec{g}_{k,l,m} - \vec{g}_{k',l',m'}) W(|\vec{r}_j - \vec{g}_{k',l',m'}|) \\
&\qquad \text{anti-symmetric!} \\
&= \ldots \\
&= \vec{0} \text{ (because of invariance under coordinate inversion)}
\end{aligned}$$

PM scheme:

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

$$\Phi(\vec{g}_{k,l,m}) = \sum_{k'} \sum_{l'} \sum_{m'} \mathcal{G}(\vec{g}_{k,l,m} - \vec{g}_{k',l',m'}) \rho(\vec{g}_{k',l',m'})$$

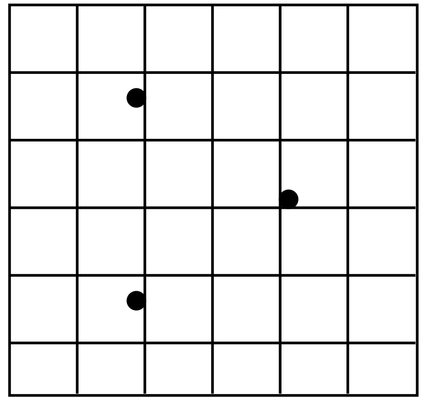
$$\rho(\vec{g}_{k,l,m}) = \frac{M(\vec{g}_{k,l,m})}{H^3}$$

$$M(\vec{g}_{k,l,m}) = \sum_{i=1}^{N} m_i W(|\vec{r}_i - \vec{g}_{k,l,m}|)$$

- 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?***

Alexander Knebe, *Universidad Autonoma de Madrid*