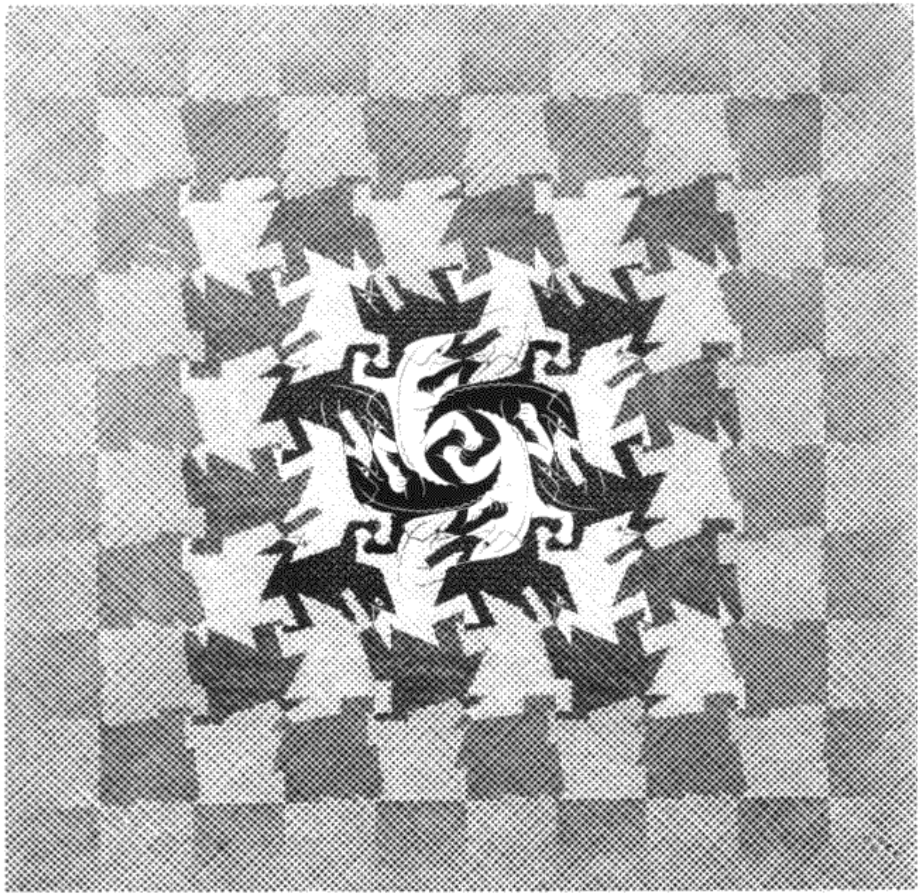
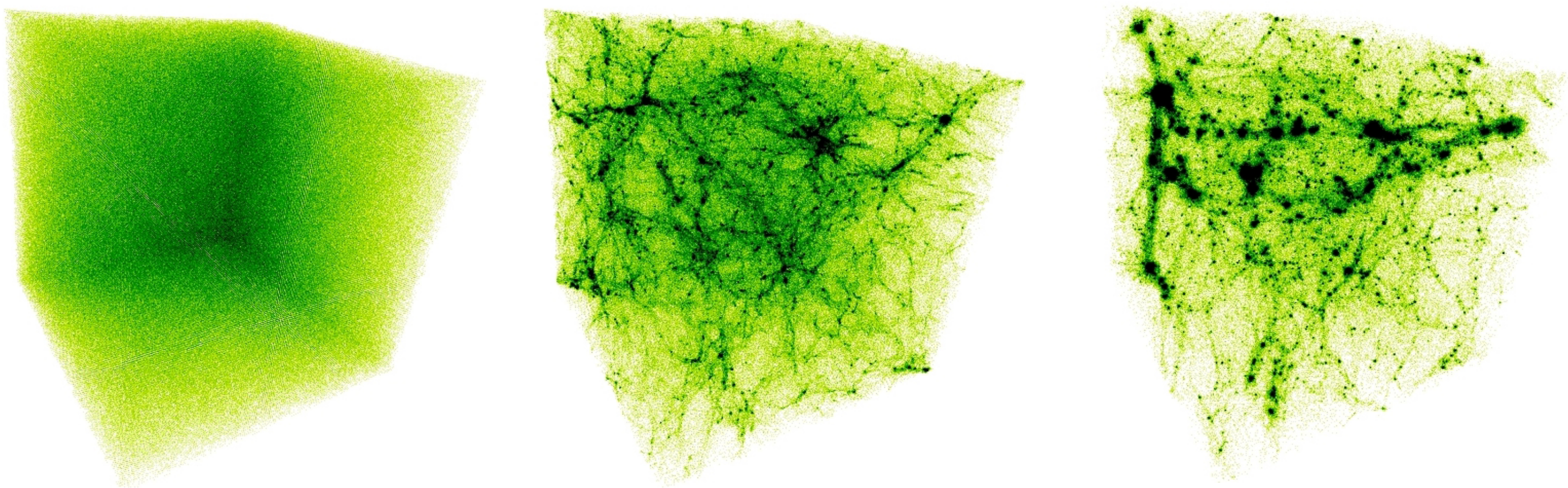
Alexander Knebe, *Universidad Autonoma de Madrid*



Platynota-Mesh code?

PM Codes



particle mesh (PM) codes

- Poisson's equation

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

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

- 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))

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

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

- 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

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))

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

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

- 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))

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

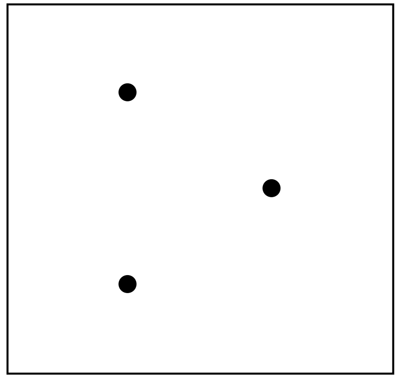
$$\vec{F}(\vec{x}_{i,j,k}) = -m\nabla\Phi(\vec{x}_{i,j,k})$$

weapon of choice: AMR codes – based upon PM!

- numerically integrate Poisson's equation

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

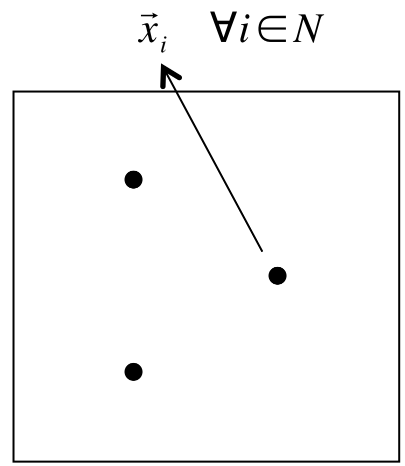
$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



- numerically integrate Poisson's equation

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

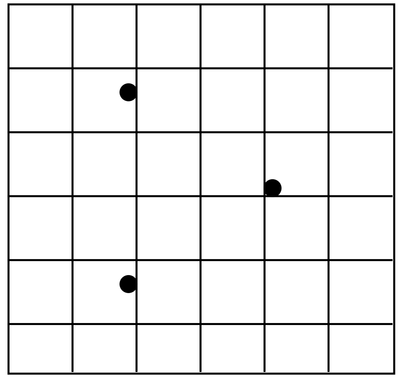
$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



- numerically integrate Poisson's equation

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

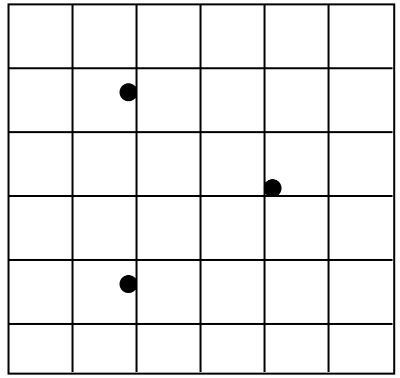
$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\Phi(\vec{g}_{k,l,m})$$



- numerically integrate Poisson's equation

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

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

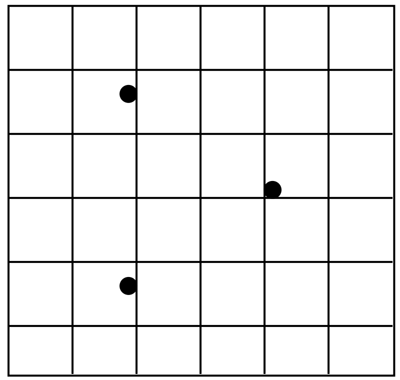


1. calculate mass density on grid $\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\vec{F}(\vec{g}_{k,l,m}) \rightarrow \vec{F}(\vec{x}_i)$

- numerically integrate Poisson's equation

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

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



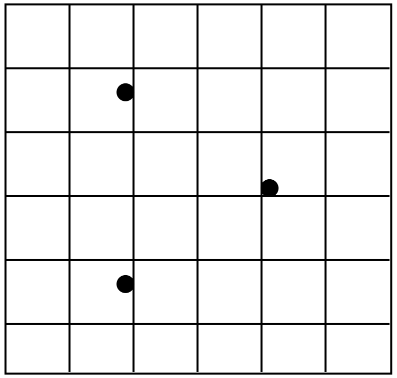
1. calculate mass density on grid $\quad \vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$
2. solve Poisson's equation on grid $\quad \Phi(\vec{g}_{k,l,m})$
3. differentiate potential to get forces $\quad \vec{F}(\vec{g}_{k,l,m})$
4. interpolate forces back to particles $\quad \vec{F}(\vec{g}_{k,l,m}) \rightarrow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

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

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

- density assignment schemes

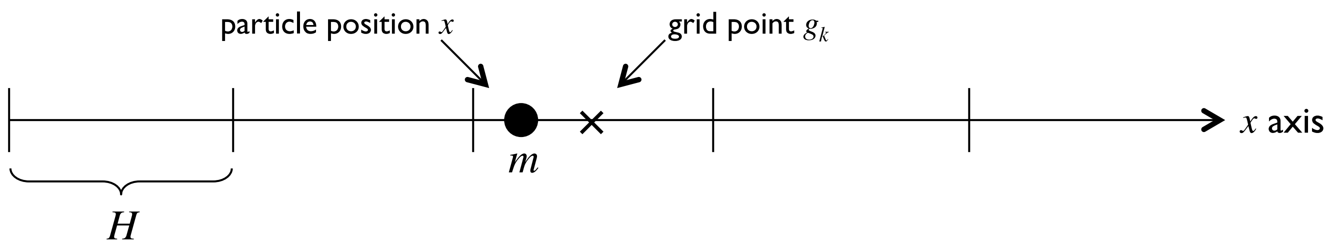
$$\vec{x}_i \rightarrow \rho(\vec{g}_{k,l,m})$$

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

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

mass assignment function

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



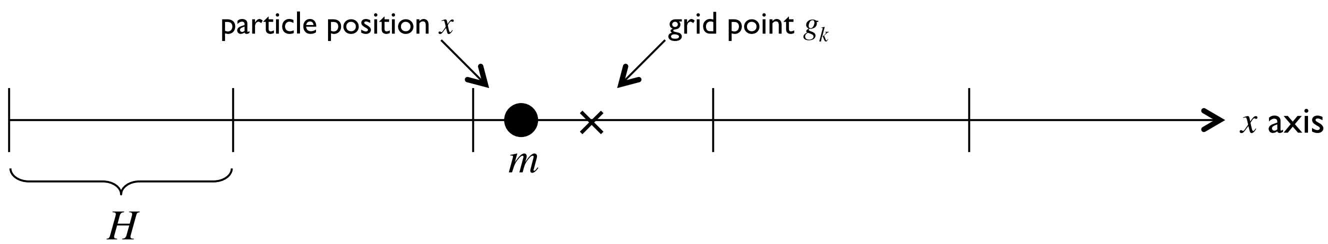
- density assignment schemes

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

<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
– …	



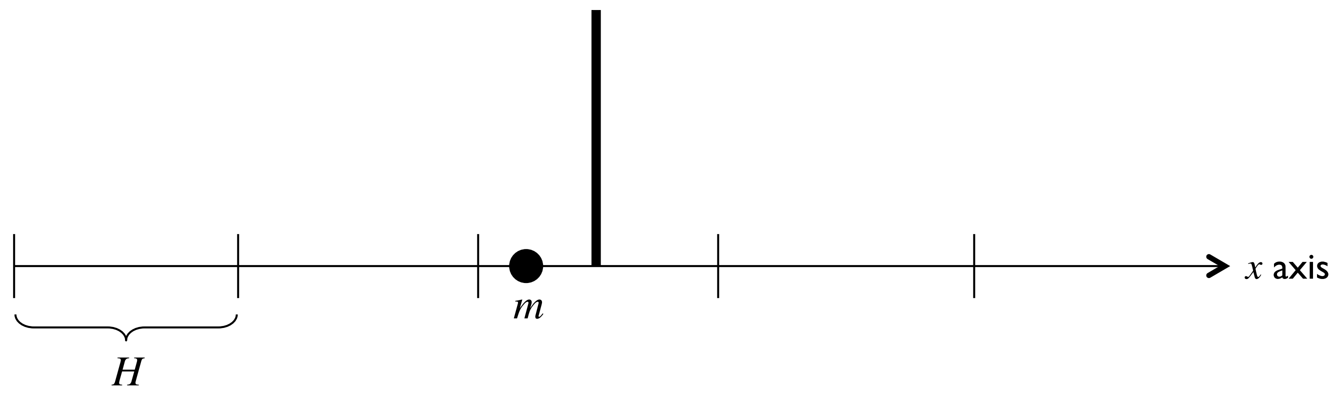
- density assignment schemes

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

Nearest-Grid-Point (NGP):

mass assignment function:

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



- density assignment schemes

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

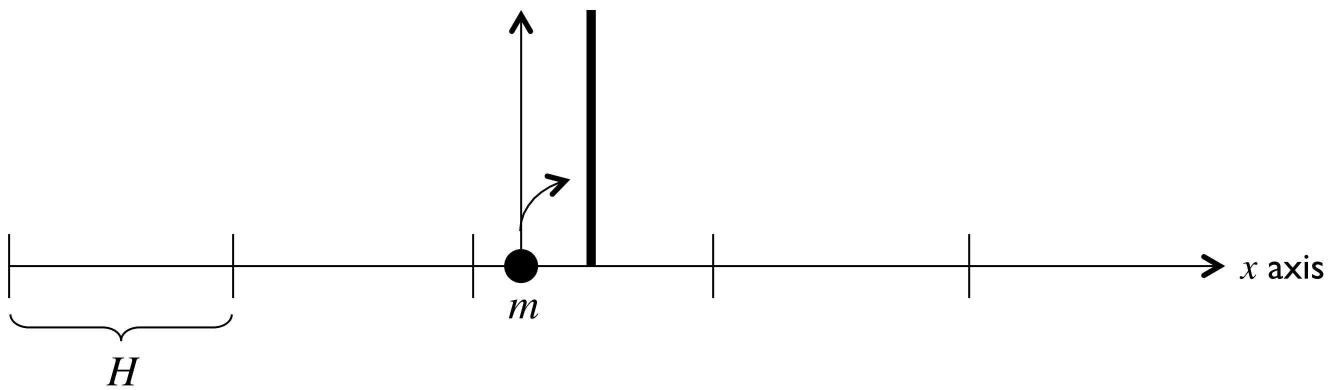
Nearest-Grid-Point (NGP):

particle shape:

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

mass assignment function:

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



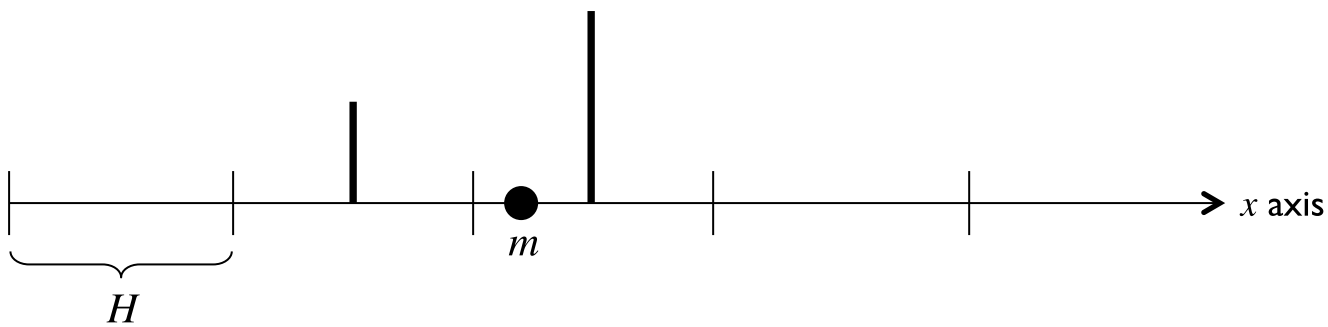
- density assignment schemes

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

Cloud-In-Cell (CIC):

mass assignment function:

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



- density assignment schemes

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

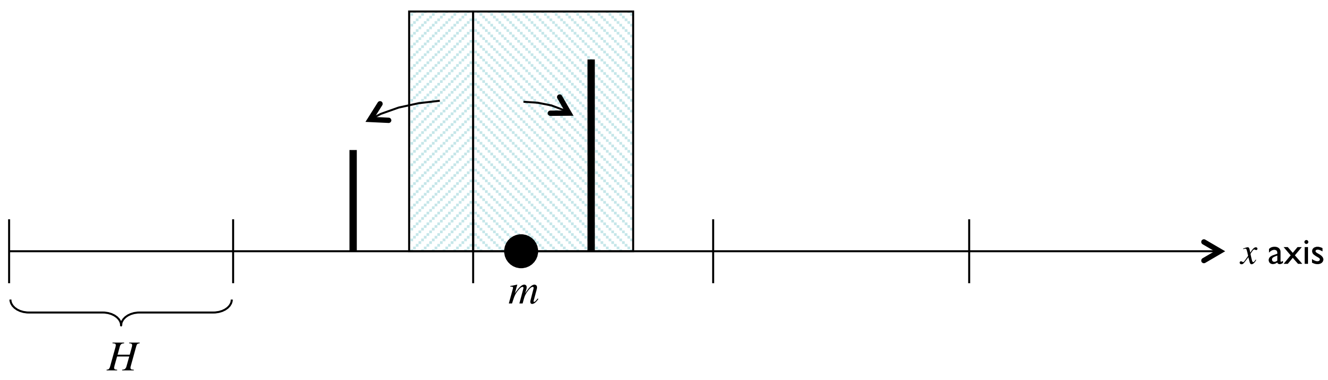
Cloud-In-Cell (CIC):

particle shape:

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

mass assignment function:

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



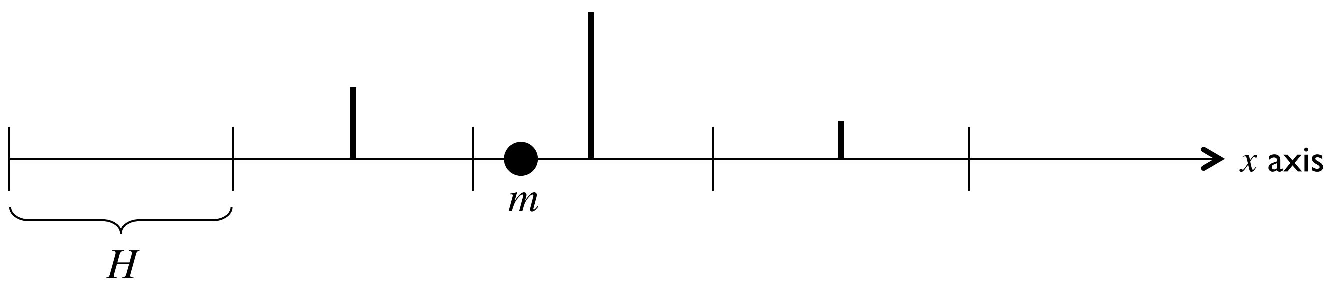
- density assignment schemes

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

<u>Triangular-Shaped-Cloud (TSC):</u>

mass assignment function:

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



- density assignment schemes

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

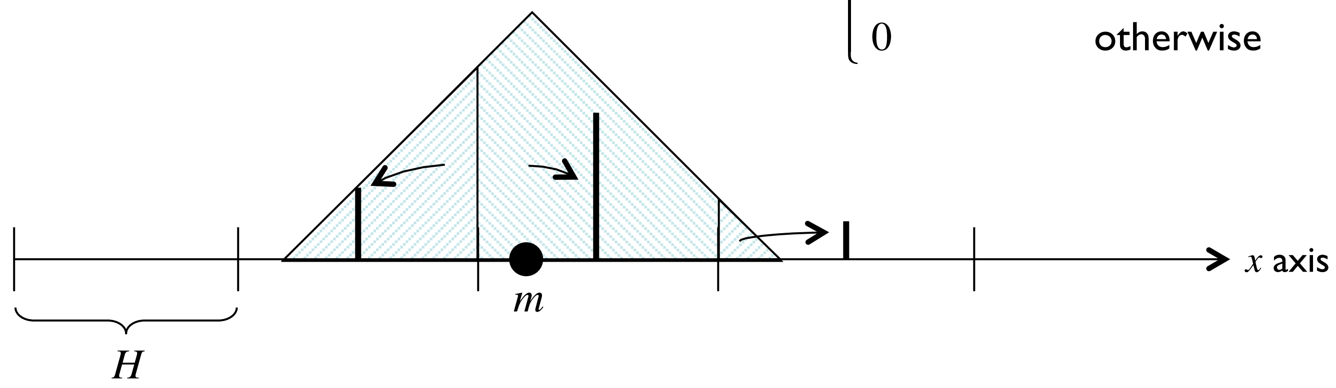
Triangular-Shaped-Cloud (TSC):

particle shape:

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

mass assignment function:

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



- density assignment schemes

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

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

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

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

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

- 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…

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

- density assignment schemes - in practice

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

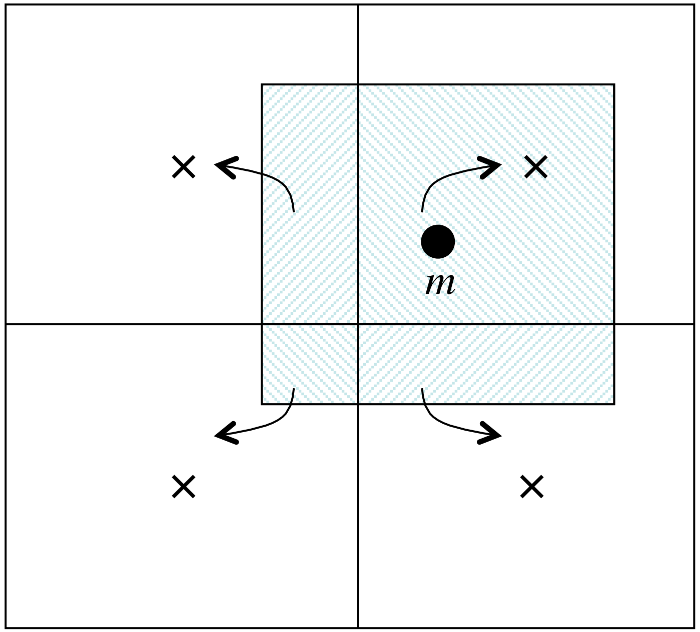
…rather loop over all particles
and
assign them to the appropriate grid points,
because the mapping x_i -> g_k is rather easy

- 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 :

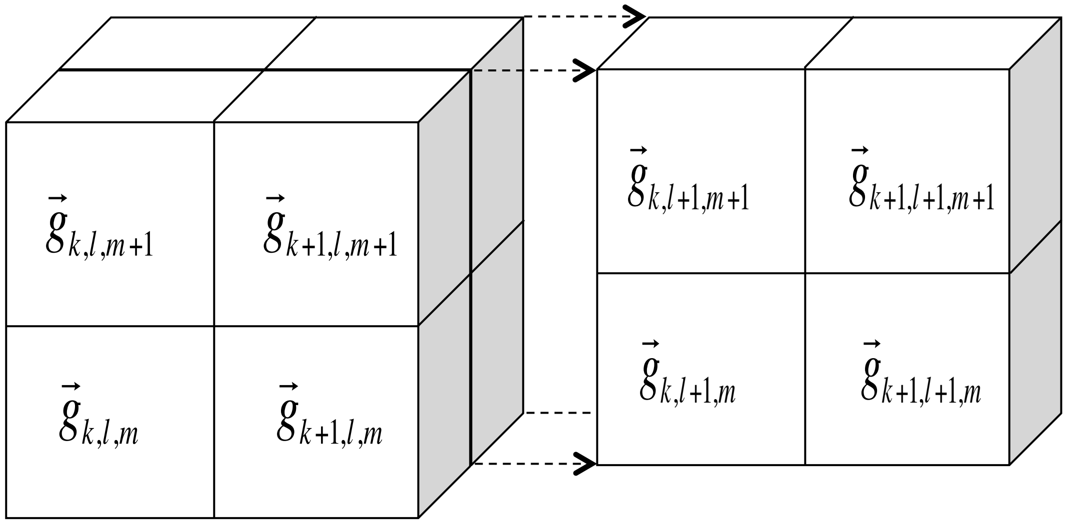


- density assignment schemes - in practice

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

  - example for CIC assignment in 3D:

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



- 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)

- 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"!

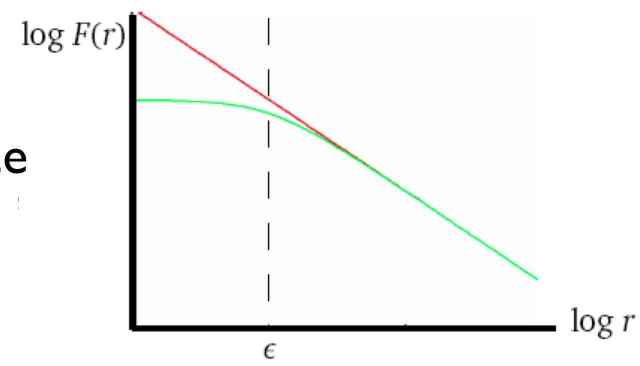
- density assignment schemes
  - which scheme to choose?

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

NGP = too crude

CIC = common choice

TSC = pretty smooth



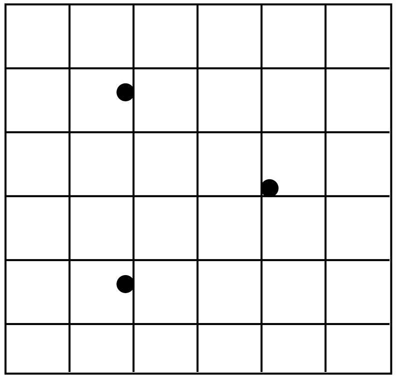
smoothing the density field will lead to a "bias" in the forces
but at the same time decrease the "variance"!

(more later when studying the accuracy...)

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

- numerically integrate Poisson's equation

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

  - relaxation technique:
    - obtain iterative solver by discretizing differential equation
    - applicable and usable for ***any*** differential equation

- numerically integrate Poisson's equation

relaxation technique

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

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

- numerically integrate Poisson's equation

relaxation technique

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

discretized Poisson's equation

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

- numerically integrate Poisson's equation

relaxation technique

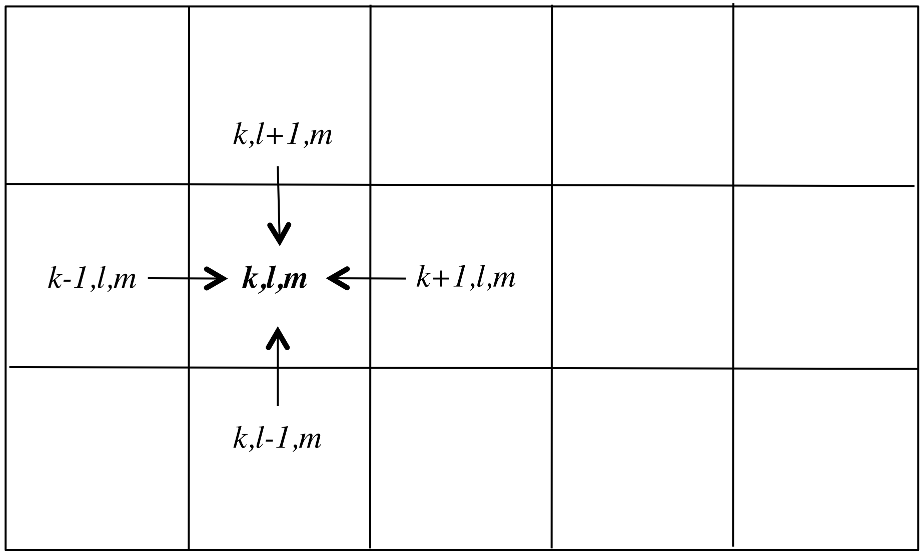
$$\Delta\Phi_{k,l,m} = \rho_{k,l,m}$$

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

discretized Poisson's equation

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

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

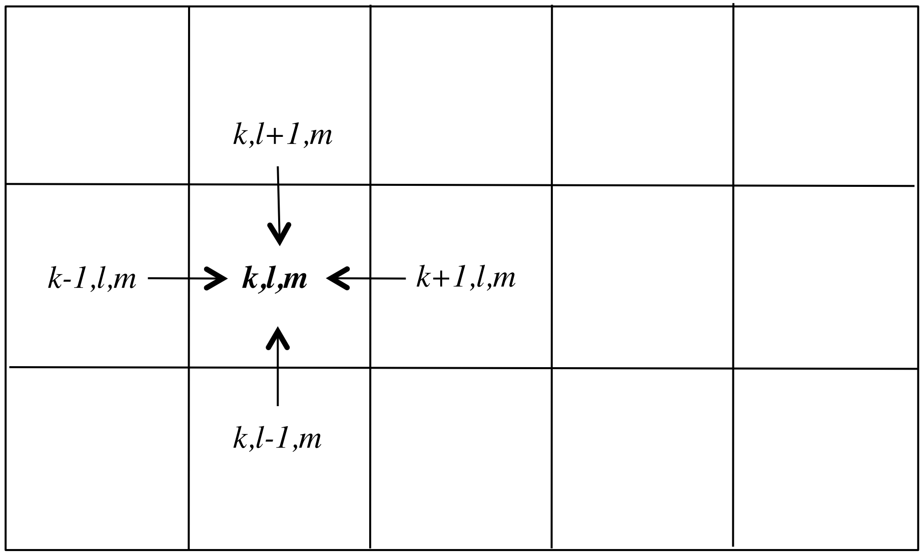


discretized Poisson's equation

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

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

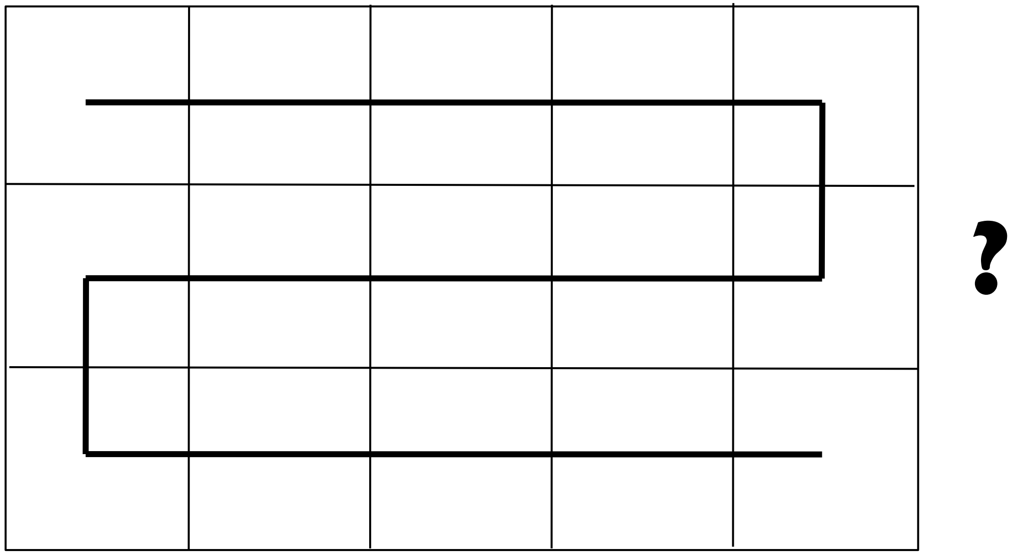
applicable to grids of *arbitrary* geometry!



discretized Poisson's equation

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

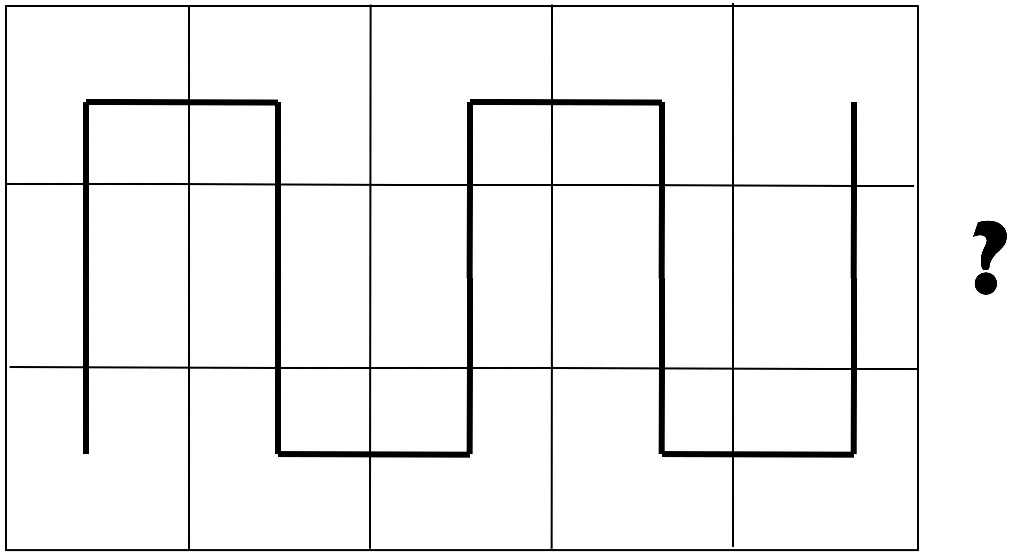
- numerically integrate Poisson's equation *relaxation technique*
    - how to sweep through the grid?



discretized Poisson's equation

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

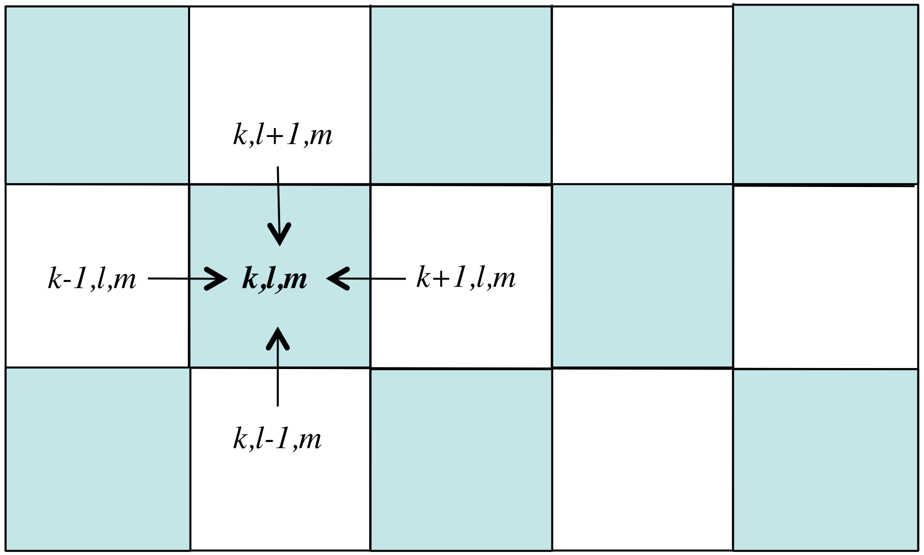
- numerically integrate Poisson's equation *relaxation technique*
  - how to sweep through the grid?



discretized Poisson's equation

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

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



discretized Poisson's equation

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

- 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

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

- 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

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

- 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 mass assignment scheme!

density as given by currently best guess for Φ^i !

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:

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

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

discretized Poisson's equation

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

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

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

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

$\|\bullet\|$ = suitable norm

tolerance error estimate

discretized Poisson's equation

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

- 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

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

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

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

– 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

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

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

discretized Poisson's equation

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

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

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

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

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

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

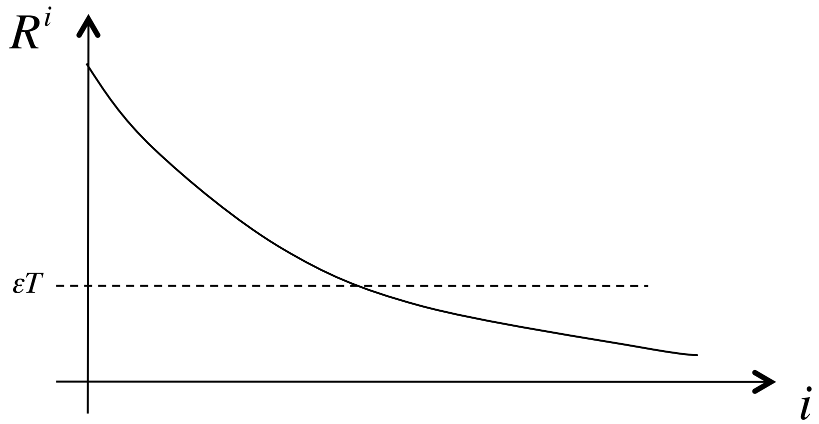
$$\mathcal{P}\left[\Delta\left(\mathcal{R}\Phi^i_{k,l,m}\right)\right] = \rho^i_{k,l,m} \overset{?}{=} \Delta\Phi^i_{k,l,m}$$

discretized Poisson's equation

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

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

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

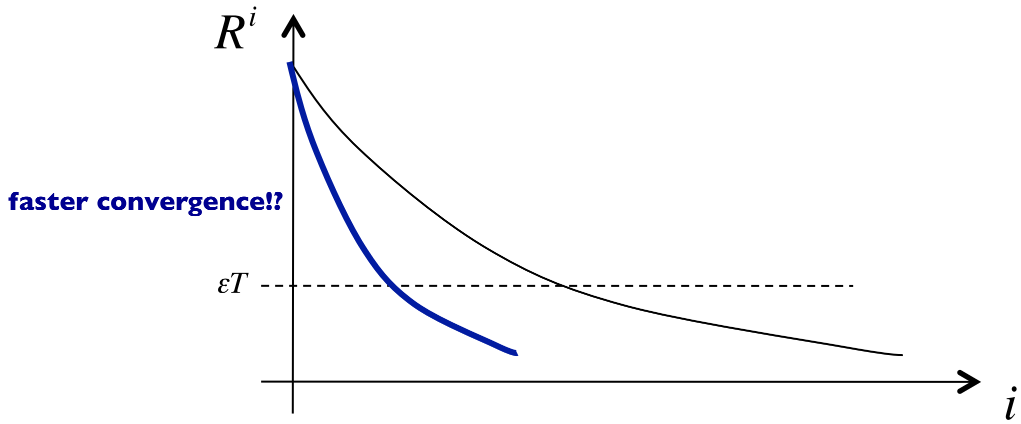


discretized Poisson's equation

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

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

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



discretized Poisson's equation

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

- 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 Φ cannot be "relaxed"
sufficiently fast on the actual grid

=> use coarser grids to speed up convergence…

discretized Poisson's equation

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

- 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

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

- 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

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

$$\Delta\Phi = \rho \qquad \rightarrow \text{ equation we wish to solve}$$

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

$$\Delta\Phi = \rho$$ → equation we wish to solve

$$\Delta\mathcal{G} = \delta$$ → equation way easier to solve…

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

$$\Delta\Phi = \rho$$

?

$$\Delta\mathcal{G} = \delta$$

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

$$\Delta\Phi = \rho$$

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x} - \vec{x}')\, \rho(\vec{x}') d^3 x'$$

$$\Delta\mathcal{G} = \delta$$

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

$$\Delta\Phi = \rho$$

$$\Phi(\vec{x}) = \iiint \mathcal{G}(\vec{x} - \vec{x}')\, \rho(\vec{x}')d^3x'$$

$$\Delta\mathcal{G} = \delta$$

$$\begin{aligned}\Delta\Phi(\vec{x}) &= \Delta \iiint \mathcal{G}(\vec{x} - \vec{x}')\rho(\vec{x}')d^3x' \\ &= \iiint \Delta\mathcal{G}(\vec{x} - \vec{x}')\rho(\vec{x}')d^3x' \\ &= \iiint \delta(\vec{x} - \vec{x}')\rho(\vec{x}')d^3x' \\ &= \rho(\vec{x})\end{aligned}$$

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

$$\Delta \mathcal{G} = \delta$$

Ansatz : $\mathcal{G} = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k$ (spectral decomposition of $\mathcal{G}$)

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

$$\Delta \mathcal{G} = \delta$$

Ansatz : $\mathcal{G} = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k$

$$\begin{aligned}
\delta &= \Delta \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k \\
&= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, \Delta e^{i\vec{k}\cdot\vec{x}} d^3k \\
&= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i\vec{k}\cdot\vec{x}} d^3k
\end{aligned}$$

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

$$\Delta \mathcal{G} = \delta$$

Ansatz : $\mathcal{G} = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k$

$$\delta = \Delta \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, \Delta e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i\vec{k}\cdot\vec{x}} d^3k \xrightarrow{e^{-i(\vec{k}'\cdot\vec{x})}} \delta\, e^{-i(\vec{k}'\cdot\vec{x})} = e^{-i(\vec{k}'\cdot\vec{x})} \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3k$$

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

$$\Delta \mathcal{G} = \delta$$

$$\textit{Ansatz :} \quad \mathcal{G} = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$\delta = \Delta \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, \Delta e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i\vec{k}\cdot\vec{x}} d^3k \quad \xrightarrow{e^{-i(\vec{k}'\cdot\vec{x})}} \quad \delta\, e^{-i(\vec{k}'\cdot\vec{x})} = e^{-i(\vec{k}'\cdot\vec{x})} \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i\vec{k}\cdot\vec{x}} d^3k$$

$$= \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\, (-k^2) e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3k$$

$$\xrightarrow{\iiint d^3x}$$

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

$$\Delta \mathcal{G} = \delta$$

$$\iiint \delta \, e^{-i(\vec{k}'\cdot\vec{x})} d^3x = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k}) \, (-k^2) \iiint e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3x d^3k$$

$$1 = \iiint \hat{\mathcal{G}}(\vec{k}) \, (-k^2) \frac{1}{(2\pi)^3} \iiint e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3x d^3k$$

$$= \iiint \hat{\mathcal{G}}(\vec{k}) \, (-k^2) \delta(\vec{k}-\vec{k}') d^3k$$

$$= -k^2 \hat{\mathcal{G}}(\vec{k})$$

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

$$\Delta \mathcal{G} = \delta$$

$$\iiint \delta\, e^{-i(\vec{k}'\cdot\vec{x})} d^3x = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k})\,(-k^2) \iiint e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3x d^3k$$

$$\boxed{1} = \iiint \hat{\mathcal{G}}(\vec{k})\,(-k^2) \frac{1}{(2\pi)^3} \iiint e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3x d^3k$$

$$= \iiint \hat{\mathcal{G}}(\vec{k})\,(-k^2)\delta(\vec{k}-\vec{k}')d^3k$$

$$= \boxed{-k^2\hat{\mathcal{G}}(\vec{k})}$$

$$\hat{\mathcal{G}}(\vec{k}) = -\frac{1}{k^2}$$

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

$$\Delta \mathcal{G} = \delta$$

$$\iiint \delta \, e^{-i(\vec{k}'\cdot\vec{x})} d^3x = \frac{1}{(2\pi)^3} \iiint \hat{\mathcal{G}}(\vec{k}) \, (-k^2) \iiint e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3x d^3k$$

$$\boxed{1} = \iiint \hat{\mathcal{G}}(\vec{k}) \, (-k^2) \frac{1}{(2\pi)^3} \iiint e^{i(\vec{k}-\vec{k}')\cdot\vec{x}} d^3x d^3k$$

$$= \iiint \hat{\mathcal{G}}(\vec{k}) \, (-k^2) \delta(\vec{k}-\vec{k}') d^3k$$

$$= \boxed{-k^2 \hat{\mathcal{G}}(\vec{k})}$$

$$\hat{\mathcal{G}}(\vec{k}) = -\frac{1}{k^2}$$

$$\xrightarrow{FFT^{-1}(\hat{\mathcal{G}}(\vec{k}))}$$

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

$$\Delta \mathcal{G} = \delta$$

$$\hat{\mathcal{G}}(\vec{k}) = -\frac{1}{k^2} \qquad \mathcal{G}(\vec{x}) = \frac{1}{4\pi x}$$

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

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

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

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

FFT ➔ convolution becomes multiplication

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

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

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

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

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}

$$\Phi$$

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

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

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!

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

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

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

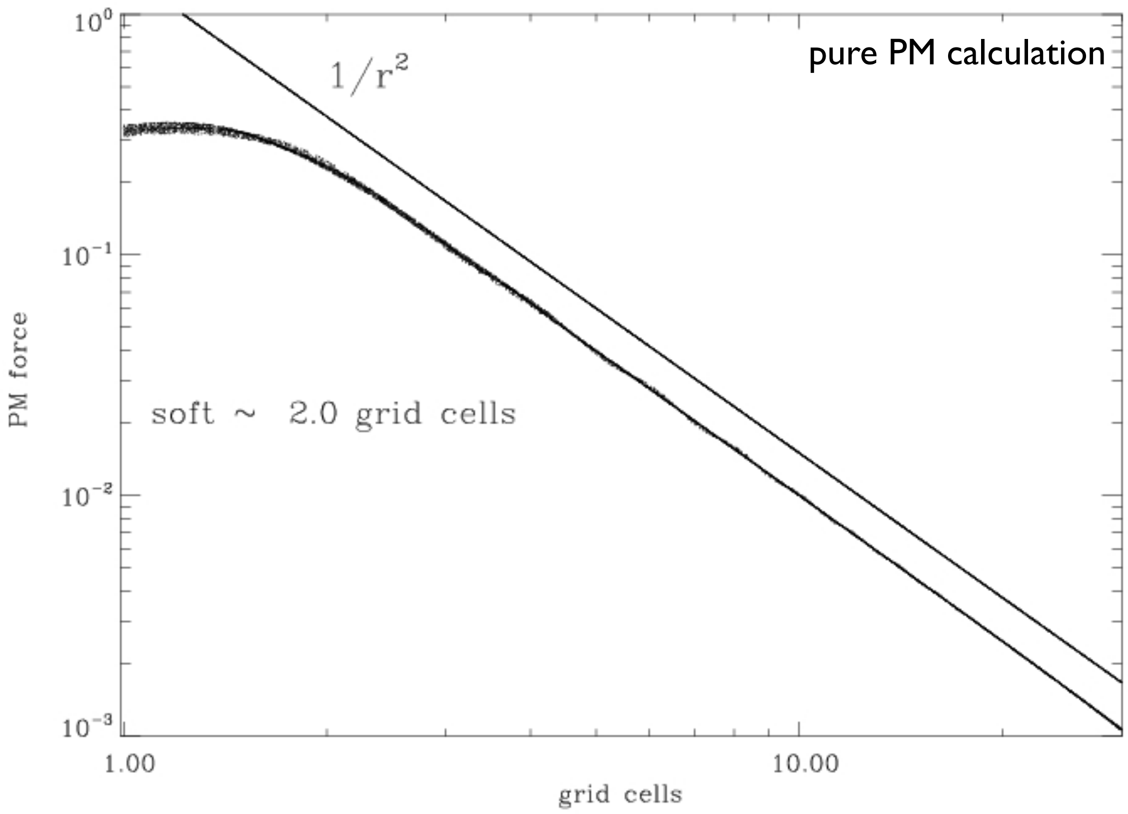
- numerically integrate Poisson's equation

accuracy of either

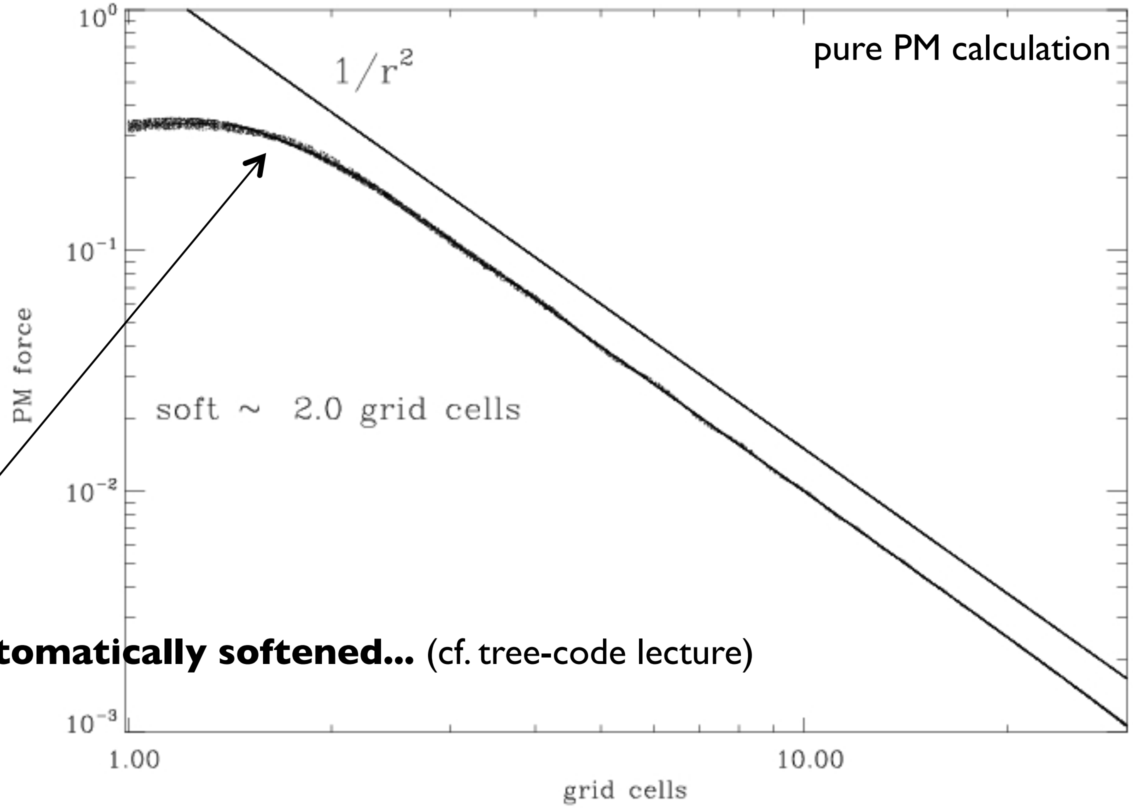
relaxation
or
FFT method

to solve Poisson's equation?

- numerically integrate Poisson's equation

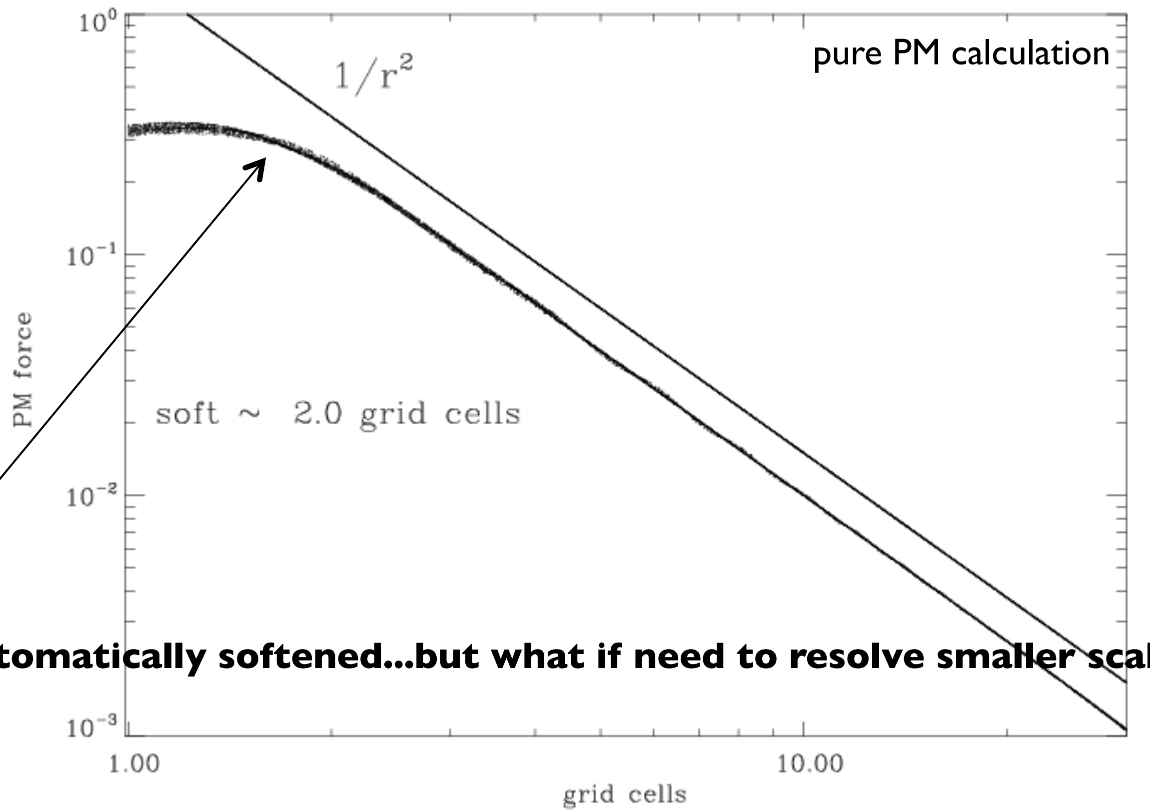


- numerically integrate Poisson's equation



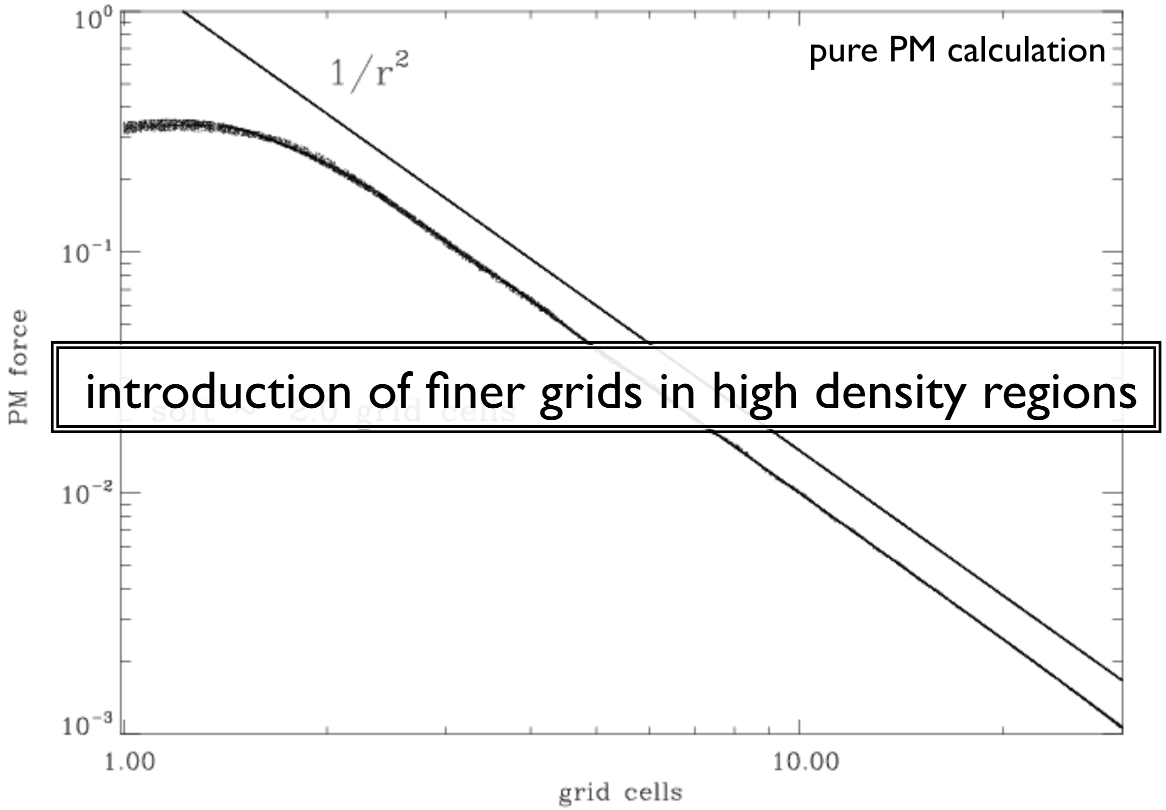
the force is automatically softened... (cf. tree-code lecture)

- numerically integrate Poisson's equation



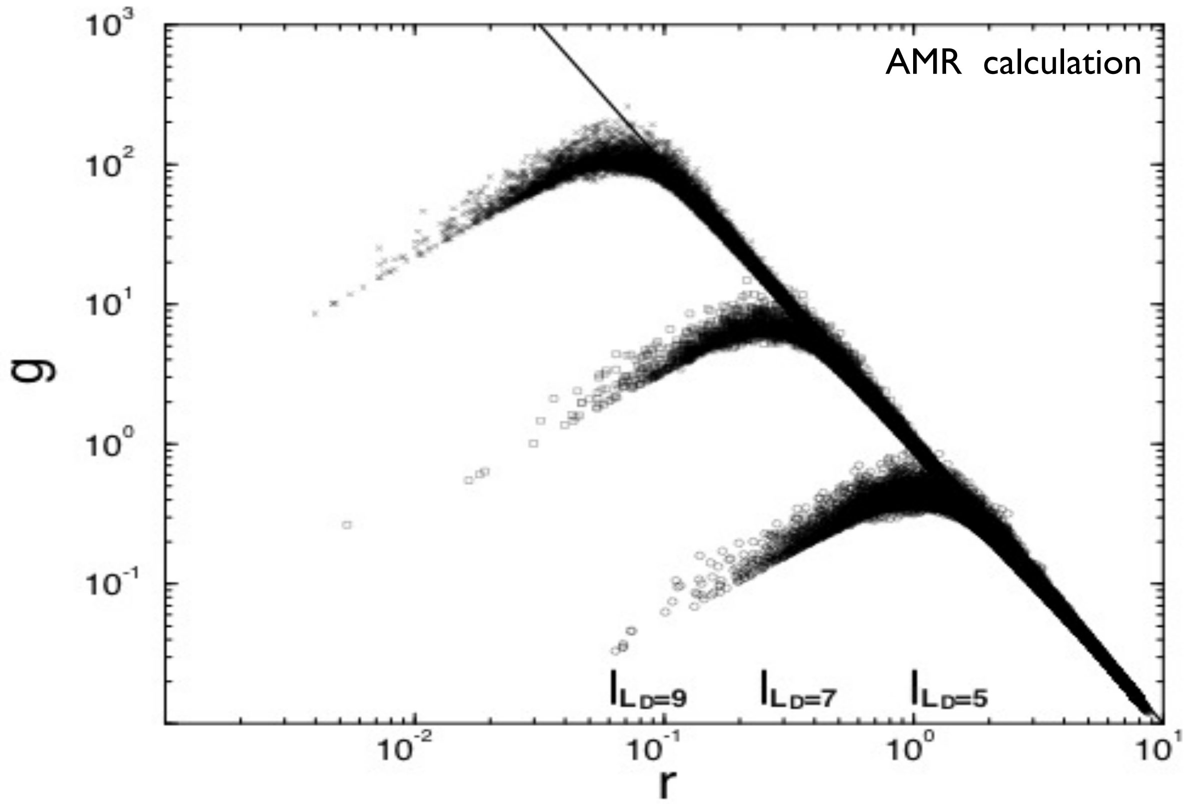
the force is automatically softened...but what if need to resolve smaller scales?

- numerically integrate Poisson's equation



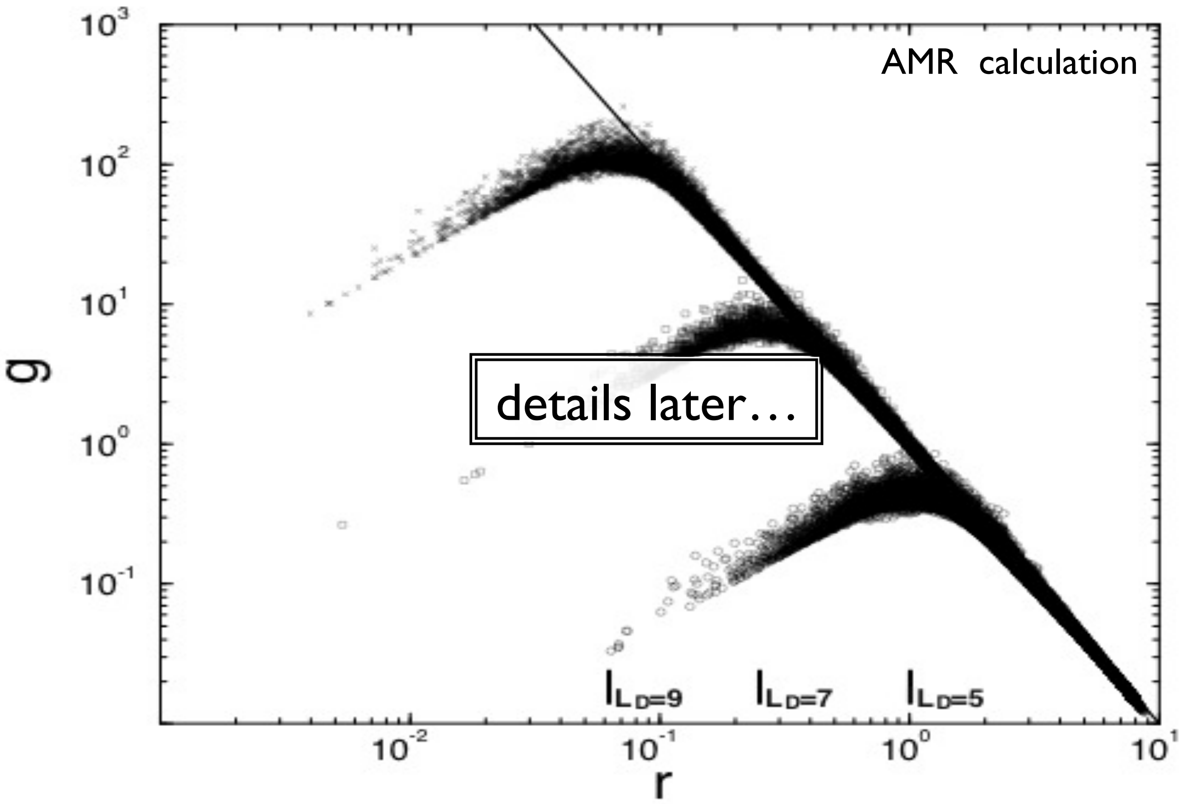
introduction of finer grids in high density regions

- numerically integrate Poisson's equation



Yahagi & Yoshi (2001)

- numerically integrate Poisson's equation

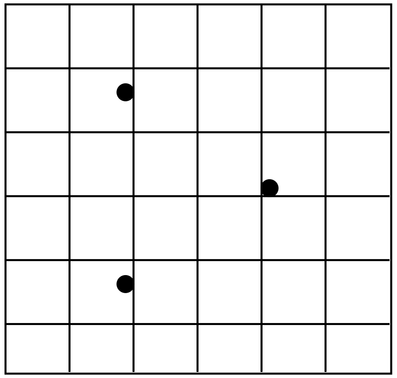


Yahagi & Yoshi (2001)

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

- obtaining the forces

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

$$\downarrow$$

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

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

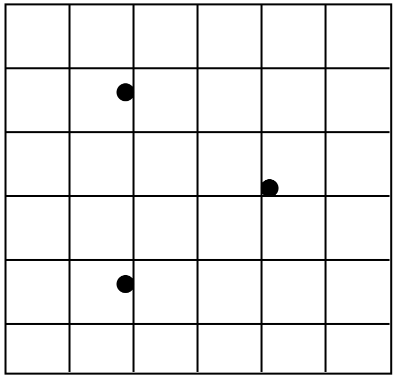
$$F_z(\vec{g}_{k,l,m}) = -m\frac{\Phi(\vec{g}_{k,l,m+1}) - \Phi(\vec{g}_{k,l,m-1})}{2H}$$

H = (current) grid spacing

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

- interpolating the forces

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

- 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

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

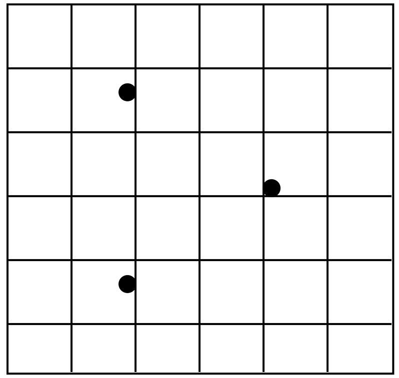
exercise…

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

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

Alexander Knebe, *Universidad Autonoma de Madrid*