Particle-Mesh (PM) Algorithm

Numerical method widely used in

- cosmology } to model { gravitational } force
- plasma physics } long-range { electromagnetic }

Uses FFT to compute forces on particles.
Breaks down at scales $\lesssim$ mesh cellsize.
Commonly combined with other methods that work on small scales:

- P^3M Particle-Particle Particle-Mesh
- Tree PM
- Adaptive PM
- can adjoin hydro, other physics.

Insert → N, L, h here. from 19.2,3

Motivation?

Poisson equation

$$\nabla^2 \phi = 4\pi G \rho \quad , \quad \vec{g} = -\vec{\nabla}\phi$$

becomes algebraic in Fourier space

$$\vec{\nabla} \rightarrow i\vec{k} \quad (\text{mathematica}: \rightarrow -i\vec{k})$$

$$-k^2 \phi_k = 4\pi G \rho_k \quad , \quad \vec{g}_k = -i\vec{k}\phi_k$$

$$\text{ie } \phi_k = -\frac{4\pi G \rho_k}{k^2} \quad , \quad = \frac{4\pi G\, i\vec{k} \rho_k}{k^2} .$$

Algorithm for force calculation:

1. Assign particles to mesh using smoothing window W.
2. FFT density on mesh
3. Deduce potential ϕ, acceleration g
4. FFT back to get acceleration on mesh
5. Interpolate accel on particles using same window W.

Using same window W in steps 1 & 5 ensures that

$$\underbrace{\text{force}(x_a, x_b)} = -\text{force}(x_b, x_a)$$

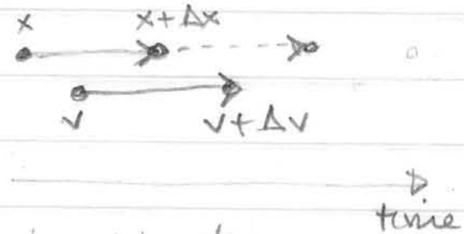
force on particle at x_a
by particle at x_b

ie pairwise momentum conservation.

Leap-frog integration

Most common method for updating particle positions and velocities is leap-frog:

1. $\Delta x = v \Delta t$
2. $\Delta v = g \Delta t$



Positions and velocities are 'staggered', evaluated at midpoint of other's timestep. 2nd order accurate with two evaluations. Works because acceleration is function

$g(x)$

only of position.
Change in positions depends only on velocities;
——— velocities ——— positions.

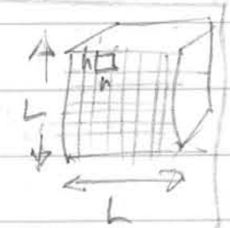
Box, mesh, particles

Typically take # particles = # cells, though not necessary. Invariably choose # cells = N^d; $N = 2^n$.
d: dimensions, eg. 3

19.3.

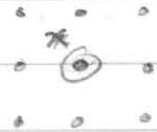
Closer look at Force calculation

Periodic box
Box size : L
Cell size : $h \equiv \frac{L}{N}$

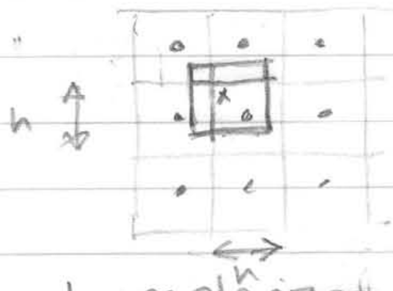


1. Assign particles to mesh.

Simplest : Nearest Grid Point (NGP)



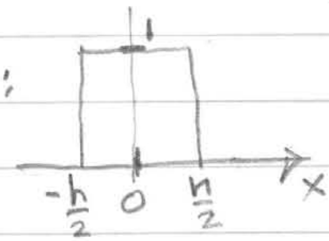
Commonest : Cloud in Cell (CIC)



Assignment is characterized by a function

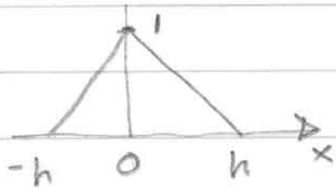
$W(x_n - x_a)$ = fraction of particle at x_a assigned to mesh point x_n

NGP in 1D:



$$W(x) = \begin{cases} 1 , & |x| < h/2 \\ 0 , & |x| > h/2 \end{cases}$$

CIC in 1D:



$$W(x) = \begin{cases} 1 - \frac{|x|}{h} , & |x| < h \\ 0 , & |x| > h \end{cases}$$

Density at mesh point x_n is

$$\rho_n = \sum_{\text{particles } a} W(x_n - x_a)$$

2. FFT ρ_n to get $\quad x_n = \frac{L}{N} n$

$$\hat{\rho}_m = \frac{1}{N} \sum_n{}' \rho_n e^{-2\pi i mn/N} \qquad k_m = \frac{2\pi}{L} m$$

3. Obtain potential $\tilde{\phi}$ and acceleration $\tilde{g}$.
"Poor man's Poisson solver" is just the continuum version:

$$\hat{\phi}_m = -\frac{4\pi G \hat{\rho}_m}{k_m^2} = -\frac{4\pi G}{(2\pi/L)^2} \frac{\hat{\rho}_m}{m^2}, \quad \hat{\phi}_0 = 0$$

$m \neq 0$

$$\tilde{g}_m = -ik\tilde{\phi}_m = \frac{4\pi G}{(2\pi/L)} \frac{i m \hat{\rho}_m}{m^2} \qquad m \neq 0 \text{ and } [\tfrac{N}{2}] \text{ if } N \text{ even}$$

$\tilde{g}_0 = 0$ and $\tilde{g}_{[\frac{N}{2}]} = 0$ if N even, to ensure g is real.

In > 1D, m is a vector of integers eg $\{m_x, m_y, m_z\}$ in 3D

There are more sophisticated versions of $\tilde{\phi}$ and $\tilde{g}$ that
(a) solve discretized ∇^2 not continuum ∇^2;
and/or
(b) are crafted to fit smoothly to small-scale treatment.

Whatever the choice, relation between $\hat{\rho}_m$, $\tilde{\phi}_m$, $\tilde{g}_m$ is always algebraic in Fourier space.

Q: Why?
A: Spatial translation invariance of physics.

Green's function of Laplacian

In general
$$\tilde{\phi}_m = \tilde{G}_m \hat{\rho}_m \qquad \tilde{G}_{-m} = G_m^* = G_m$$
$$\tilde{g}_m = \tilde{D}_m \tilde{\phi}_m \qquad \tilde{D}_{-m} = D_m^* = -D_m$$

where $\hat{G}_m$, $\hat{D}_m$ are algebraic functions of m.

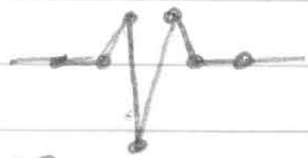
postpone

Discretized Laplacian ∇^2 ??

$$\frac{\partial \phi}{\partial x} \overset{\text{discretize}}{\rightsquigarrow} \frac{\phi(x+h) - \phi(x)}{h}$$

$$\frac{\partial^2 \phi}{\partial x^2} \overset{\text{discretize}}{\rightsquigarrow} \frac{1}{h}\left[\frac{\phi(x+h)-\phi(x)}{h} - \frac{\phi(x)-\phi(x-h)}{h}\right]$$

$$= \frac{1}{h^2}\left[\phi(x-h) - 2\phi(x) + \phi(x+h)\right]$$

$= \phi$ convolved with 

In Fourier space this becomes
$\tilde{\phi}$ multiplied by FT of 

4. FFT back

$$\vec{g}_n = \sum_m{}' \tilde{\vec{g}}_m e^{2\pi i mn/N}$$

5. Interpolate acceleration g_n on mesh to get acceleration $g(x_p)$ at positions of particles:

$$\vec{g}(x_p) = \sum_{\text{grid points } n} W(x_n - x_p)\, \vec{g}_n$$

Put 1–5 together:

$$\vec{g}(x_b) = \sum_{\text{particles } a} \underbrace{g(x_b, x_a)}$$

accel of particle at x_b
due to particle at x_a

accel of particle at x_b

where

$$\vec{g}(x_b, x_a) = \frac{1}{N} \sum_m{}' \, \vec{\tilde{D}}_m \, \tilde{G}_m \sum_n{}' \sum_{n'}{}' \, W(x_n - x_b) \, W(x_{n'} - x_a) \, e^{2\pi i m (n - n')/N}$$

Important that $g(x_b, x_a)$ changes sign under particle exchange $x_b \leftrightarrow x_a$

$$\vec{g}(x_a, x_b) = -\vec{g}(x_b, x_a)$$

ensuring pairwise momentum conservation.

19.5 → here.