Particle Mesh Code

José Ortuño Macías,
Universidad Autónoma de Madrid

May 31, 2017

UNIVERSIDAD AUTONOMA
DE MADRID

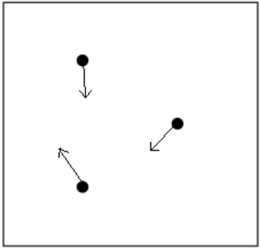
Contents

Introduction

The N-body problem

Definition

Given known orbital properties (instantaneous position, velocity and time) of a group of bodies, predict their interactive gravitational forces; and consequently, predict their true orbital motions for all future times.



The N-body problem

- We are dealing with **collisionless matter**:

$$\begin{aligned} \frac{d\vec{x}}{dt} &= \vec{v} \\ \frac{d\vec{v}}{dt} &= -\nabla\phi \end{aligned} \qquad \textit{equations of motion}$$

- We need to solve the **Poisson equation**:

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

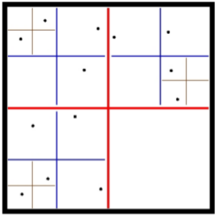
- There are two main methods:

The N-body problem

Particle approach

$$\vec{F}(\vec{x_j}) = \sum_{i \neq j} \frac{G m_j m_i}{(x_i - x_j)^3} (\vec{x_i} - \vec{x_j})$$

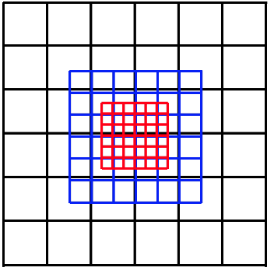
- *Tree codes*



Grid approach

$$\Delta\phi(\vec{g}_{k,l,m}) = 4\pi G \rho(\vec{g}_{k,l,m})$$
$$\vec{F}(\vec{g}_{k,l,m}) = -m\nabla\phi(\vec{g}_{k,l,m})$$

- Particle Mesh codes
- Adaptive Mesh refinement

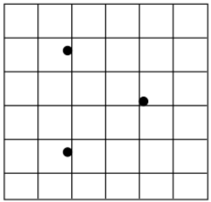


PM algorithm and code

PM algorithm

1. Divide the space into cells of a grid
2. Compute the mass density on grid
3. Solve Poisson equation on grid
4. Differenciate potential to get forces
5. Interpolate forces back to particles

Discretization of space



PM code:

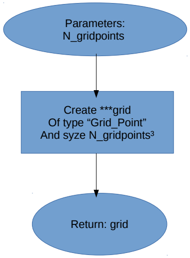
Firstly create two new structures:

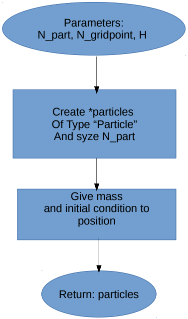
- Particle $\begin{cases} mass \\ x,\ y,\ z \\ v_x,\ v_y,\ v_z \\ f_x,\ f_y,\ f_z \end{cases}$
- Grid_Point $\begin{cases} density \\ potential \\ a_x,\ a_y,\ a_z \end{cases}$

Discretization of space

PM code:
Function declaration:
Grid_Point ***grid_creator(parameters)

PM code:
Function declaration:
Particle *particles_creator(parameters)

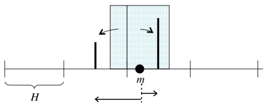




Density on grid

PM algorithm:

- Cloud-In-Cell (CIC)



Weight function:

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

Mass of each cell:

$$M\left(\vec{g}_{k,l,m}\right) = \sum_i m_i W(|d_x|) W(|d_y|) W(|d_z|)$$

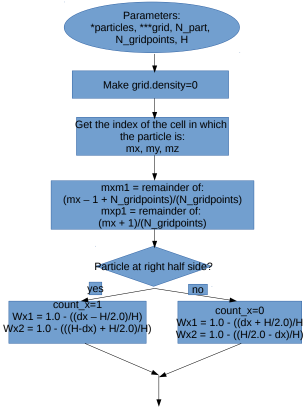
Density of each cell:

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

PM code:

Function declaration:

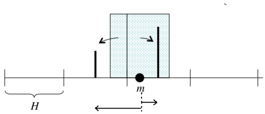
void Mass_GridPoint(parameters)



Density on grid

PM algorithm:

- Cloud-In-Cell (CIC)



Weight function:

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

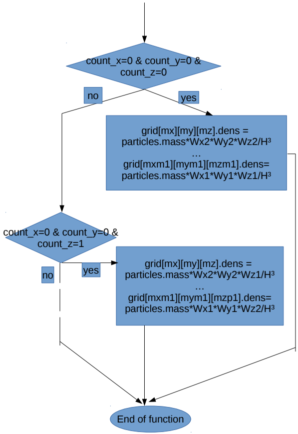
Mass of each cell:

$$M\left(\vec{g}_{k,l,m}\right) = \sum_i m_i W(|d_x|) W(|d_y|) W(|d_z|)$$

Density of each cell:

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

PM code:

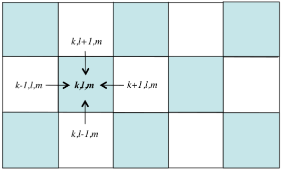


Poisson equation on grid

PM algorithm:

- relaxation technique:

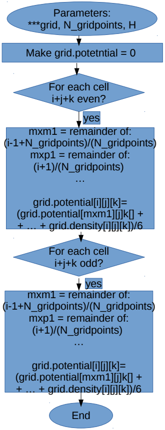
$$\phi^{i+1}_{k,l,m} = \frac{1}{6}\left(\phi^{i}_{k+1,l,m} + \phi^{i}_{k-1,l,m} + \phi^{i}_{k,l+1,m} + \phi^{i}_{k,l-1,m} + \phi^{i}_{k,l,m+1} + \phi^{i}_{k+1,l,m-1} - 4\pi G \rho_{k,l,m} H^2\right)$$



PM code:

Function declaration:

void Potential_GridPoint(parameters)



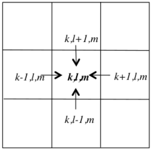
Differenciate potential

PM algorithm:

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

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

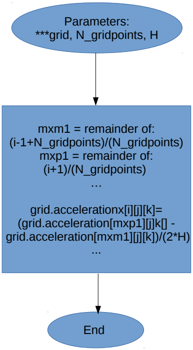
$$F_z\left(\vec{g}_{k,l,m}\right) = -m\frac{\phi\left(\vec{g}_{k,l,m+1}\right)-\phi\left(\vec{g}_{k,l,m-1}\right)}{2H}$$



PM code:

Function declaration:

void Acceleration_GridPoint(parameters)

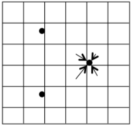


Interpolate forces

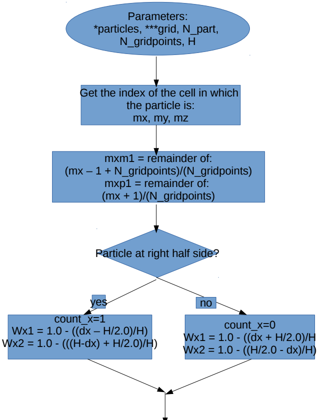
PM code:
Function declaration:
void Force_particles(parameters)

PM algorithm:

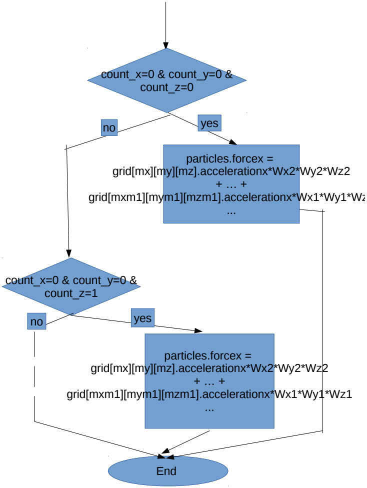
$$\vec{F}\left(\vec{g}_{k,l,m}\right) \longrightarrow \vec{F}\left(\vec{r}_j\right)$$



$$\vec{F}\left(\vec{r}_j\right) = \sum_k \sum_l \sum_m \vec{F}\left(\vec{g}_{k,l,m}\right) W\left(|d_x|\right) W\left(|d_y|\right) W\left(|d_z|\right)$$



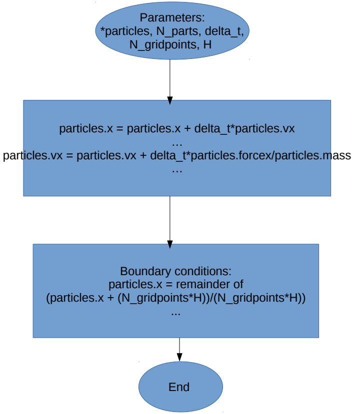
Interpolate forces



N-body code

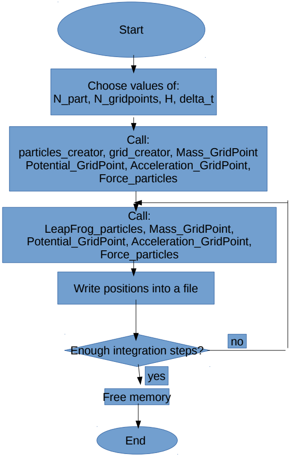
Leap-frog scheme

Function declaration:
void LeapFrog_particles(parameters)



Main function

Function declaration: void Main



Some results

Sun-Earth

Randomly distributed particles

Pros and Cons

Pros and Cons

Cons

- The forces depend on the number of iterations of the potential calculator.
  It needs so many iterations to reach the limits of the grid.
- For particles in the same cell or neighbour cells, the force is "soften".
- Time consuming for great number of cells.
- Some particles star to increase their velocity because of the boundary condition.
- Dependence on the time step of the integration.

Pros

- Still fast for a great number of particles.

Sun, Earth, Mars and Jupiter

Collapse of a sphere