Lecture 8

Particle in cell methods

D. Dickinson
d.dickinson@york.ac.uk

York Plasma Institute, Department of Physics, University of York

Autumn 2016

UNIVERSITY of York

Overview of course

This course provides a brief overview of concepts relating to numerical methods for solving differential equations.

Topics to be covered include:

- ~~Integrating ODEs~~
    - ~~Explicit techniques~~
    - ~~Implicit techniques~~
    - ~~Using Scipy to integrate ODEs~~
- ~~Spatial discretisation~~
    - ~~Finite differencing~~
    - ~~Spectral methods~~
    - ~~Finite elements~~
- Particle In Cell (PIC) approaches
- Continuum techniques

Notes available on the VLE.

Overview this lecture

This lecture will look at

- An introduction to particle in cell methods

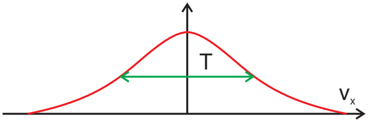
Particle in cell methods

Now that we've looked at how we can integrate equations in time and solve discretised spatial problems we're in a position to combine these and solve more complicated systems of equations similar to the ones we looked at in the first lecture.

Today we'll be looking at an approach known as particle in cell (PIC) in the context of a system of charged particles. PIC methods are quite common in plasma physics as they are relatively simple to implement and parallelise well. Unfortunately as we'll see, noise can be a problem in such simulations.

In a real system we typically have $\sim 10^{20-21}$ particles. This isn't practical however, with a supercomputer we may be able to treat $\sim 10^{9-10}$ and on a standard laptop something like 10^4 particles is more practical.

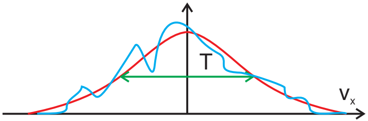
As such, we are effectively simulating a random subset of the real particles. This can have consequences for how well we can capture certain physics. Consider calculating the temperature at each location. We can plot the PDF as a function of velocity, first consider the case where we have a large number of particles,



Particle in cell methods : Particle noise

In a real system we typically have $\sim 10^{20-21}$ particles. This isn't practical however, with a supercomputer we may be able to treat $\sim 10^{9-10}$ and on a standard laptop something like 10^4 particles is more practical.

As such, we are effectively simulating a random subset of the real particles. This can have consequences for how well we can capture certain physics. Consider calculating the temperature at each location. We can plot the PDF as a function of velocity, first consider the case where we have a large number of particles, now if we only have a few particles.



When we don't have many particles we end up with a very noisy approximation of a smooth function. As we have a statistical sample the noise is Poisson like and signal to noise $\propto \sqrt{N}$, where N is particle count.

Noise is one of the main concerns for PIC simulations.

Particle in cell methods : Charged particle problem

The problem of charged particle motion is one of the systems we looked at in the first SciPy practical lecture and we'll revisit it here to illustrate the PIC approach. The motion of a single charged particle is determined by the equations

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}, \quad \frac{d\boldsymbol{v}}{dt} = \frac{q}{m}\boldsymbol{E}$$

where the electric field comes from Poisson's equation

$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}$$

Now we can use the equations of motion

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}, \quad \frac{d\boldsymbol{v}}{dt} = \frac{q}{m}\boldsymbol{E}$$

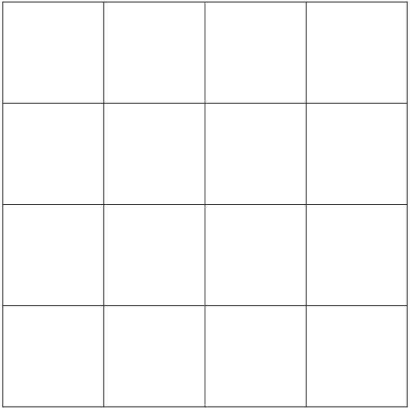
with our time integrator (e.g. odeint) to evolve the system in time.

To do this we need to calculate $\boldsymbol{E}$. A simple approach to this would be to use Coulomb's law to calculate the force on each particle from all the others, as we did in the odeint demonstration lecture. Unfortunately this is quite inefficient; for each of the N particles there are $N-1$ other particles acting on it. So we have to do $N(N-1)$ calculations to find $\boldsymbol{E}$ for each particle. As we have to repeat this calculation each time we want to know the time derivatives[1] this approach is not feasible even for the case with 10^4 particles.

To solve this problem we'll adopt a PIC approach where we treat the charged particles as markers and solve for $\boldsymbol{E}$ on a discrete grid.

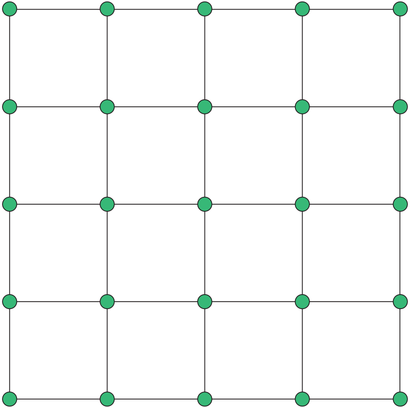
[1] Remember that some techniques like RK4 require multiple evaluations per time step.

Particle in cell methods : Domain



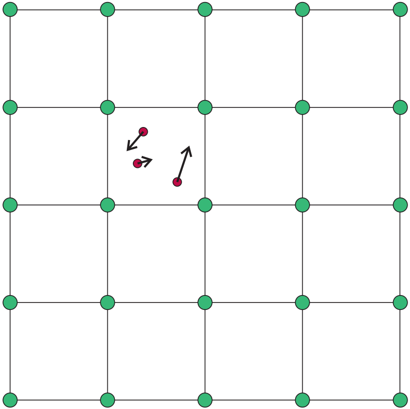
To implement a PIC algorithm to simulate the charged particle system the first step is to split our domain into discrete cells,

Particle in cell methods : Domain



To implement a PIC algorithm to simulate the charged particle system the first step is to split our domain into discrete cells, we'll solve the fields on cell corners

Particle in cell methods : Domain



To implement a PIC algorithm to simulate the charged particle system the first step is to split our domain into discrete cells, we'll solve the fields on cell corners and add particles with specific position and velocity within the cells.

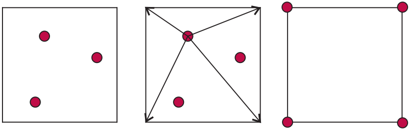
Particle in cell methods : Gather/Scatter approach

To avoid the problems inherent in the simple Coulomb approach to calculating $\boldsymbol{E}$ the PIC approach makes an approximation. We assume that the force on a given particle due to a bunch of particles in a distant cell can be represented by a single charge, which we place on a cell corner. This gives arise to what is referred to as the gather/scatter approach to calculating the field.

This consists of three stages:

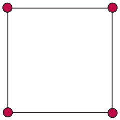
- Gather : Collect charges onto cell corners.
- Calculate : Solve Poisson's equation on grid.
- Scatter : Use field on grid to find field at particle position.

Particle in cell methods : Gather/Scatter approach

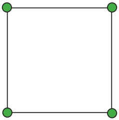


Gather Collect charges onto cell corners. For each particle work out distance to its four enclosing corners and contribute to the charge on these corners weighted by the distance. For example a singly charge particle at the exact centre of a cell will contribute 1/4 to each corner, whilst one practically on a corner will contribute most of it's charge to this corner and very little to the rest.

Particle in cell methods : Gather/Scatter approach



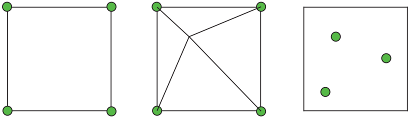
$$\nabla^2\phi = -\frac{\rho}{\epsilon_0}$$



Gather Collect charges onto cell corners.

Calculate Solve Poisson's equation on grid. We've seen previously how to solve $\epsilon_0\nabla^2\phi = -\rho$ and we simply need to apply these techniques here, where ρ comes from the charge collected on cell corners.

Particle in cell methods : Gather/Scatter approach



Gather Collect charges onto cell corners.

Calculate Solve Poisson's equation on grid.

Scatter Interpolate field from cell corners to particle position. Once we know the field at the cell corner we need to estimate its value at the particle location.

Particle in cell methods : Gather/Scatter approach

Gather Collect charges onto cell corners.
Calculate Solve Poisson's equation on grid.
Scatter Interpolate field from cell corners to particle position.

This approach is typically much more efficient than a direct Coulomb approach and scales much more efficiently.

We can summarise the whole PIC algorithm using a simple flow chart.

- We use our time integration techniques to update $\boldsymbol{x}$ and $\boldsymbol{v}$.
- We use our two point boundary value techniques to calculate the field on the grid.