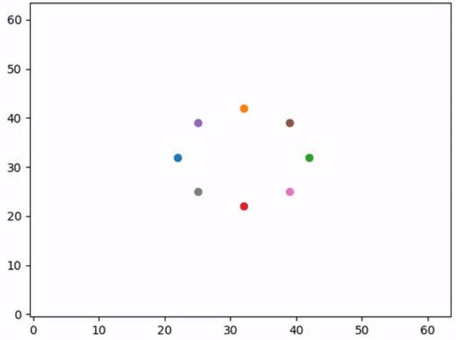
Parallelized Particle-in-Cell Method for Plasma Simulation

CS 205

Kevin Howarth, Aditi Memani, Hari Raval, Taro Spirig

Motivation

Simulating Particles: Naive Approach

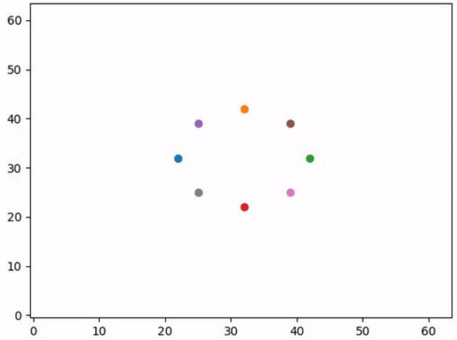


- **Goal:**
  - Simulate Particle Trajectories
- **Initial Approach:**
  - Solve this problem as an N-body problem
  - Coulomb Force tells us the exact force on 2 charged particles given their position
- **Problem:**
  - **$O(N^2)$** runtime due to consideration of all binary interactions
  - Fusion problems can have up to 10^{18} particles / m^3 !

How do we scale well with large numbers of particles?

Simulating Particles: Particle-in-Cell Approach

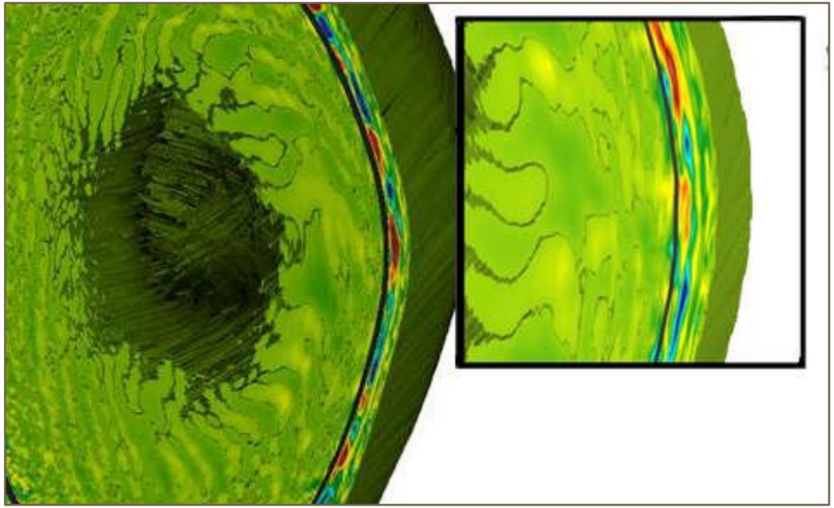
- **(Same) Goal:**
  - Simulate Particle Trajectories
- **Robust Approach:**
  - Avoid considering binary interactions among all particles
  - Perform work only once on each particle
  - Particle-in-Cell algorithm
- **Analysis:**
  - **O(N)** runtime that still simulates motion of particles
  - Scales with the number of particles in interesting plasmas



Applications of Particle-in-Cell (PIC)

- PIC can describe complicated physics in fusion plasmas
- XGC is the Princeton Plasma Physics Laboratory's PIC code

PIC is used for cutting-edge research!

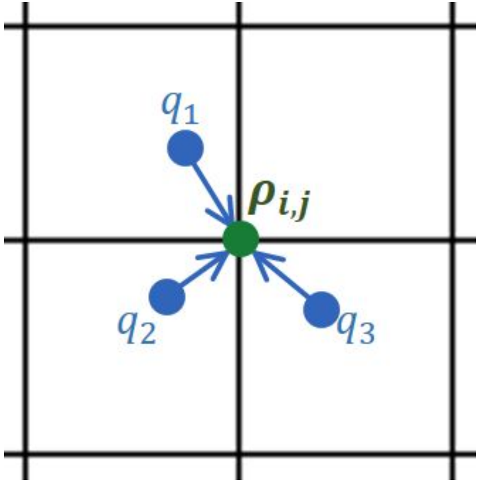


A property heatmap inside a tokamak simulated by XGC
Source: insidehpc.com

Mathematical Model

Mathematical Model

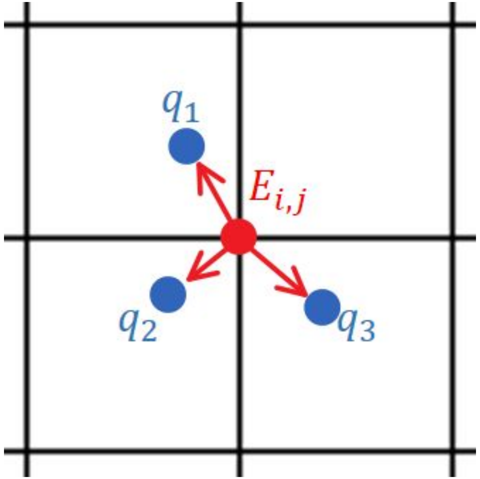
1. Interpolate from Particles to Mesh



2. Solve Discrete Poisson Equation on Mesh

$$\nabla^2 \phi = -\rho, \qquad \nabla\phi = E$$

3. Interpolate from Mesh back to Particles



4. Time-step Particle Locations

$$\frac{d\overline{v}}{dt} = qE(\overline{x}), \qquad \frac{d\overline{x}}{dt} = \overline{v}$$

Parallelization

Parallelization

- **Why?**
  - In real applications, consider a lot of particles
  - Increase in grid points → increase in accuracy
  - PIC algorithm is a good candidate for parallelization
- **What?**
  - PIC algorithm considers each particle separately when interpolating and time-stepping
  - Limiting factor on number of grid points: solution of the discrete Poisson equation
- **How?**
  - First focus: shared memory model using openMP
    - Limitations to using shared memory
    - Future considerations: distributed memory using MPI

Analysis

Analysis

- **Compare runtime scaling across 3 key factors:**
  - Dimensions of the mesh grid
  - Maximum number of particles analyzed at once
  - Optimal thread count for the parallel regions
- **Explore parallelized interpolation methods**
  - Distance Based Interpolation
  - Bilinear Interpolation
  - Piecewise Quadratic Interpolation (M4)
- **Speedup**
  - Compare Sequential and Parallelized Algorithms

Thank You