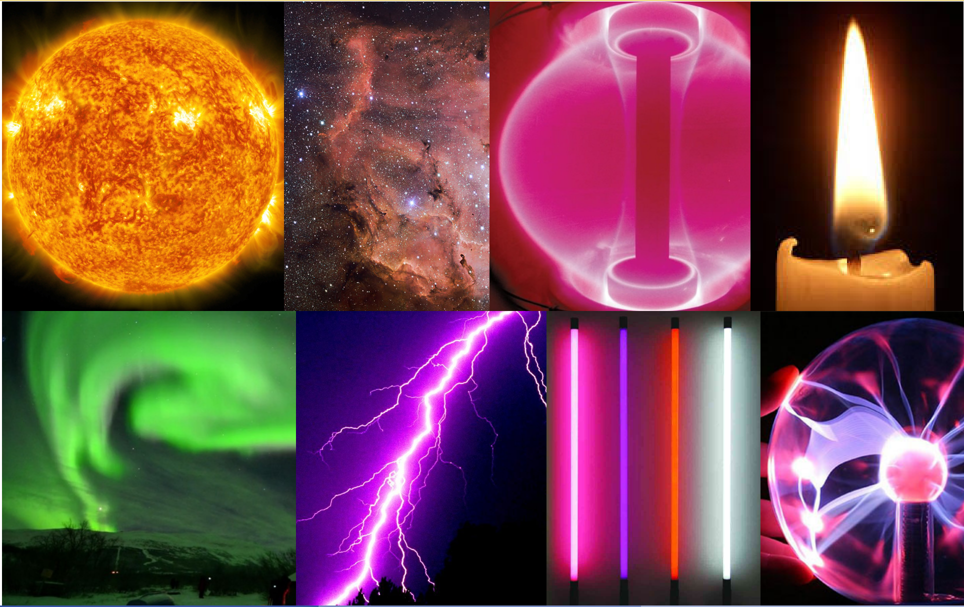
Monte-Carlo/PIC methods to solve Vlasov-Boltzmann equation

David Pfefferlé[1]

[1]The University of Western Australia, 35 Stirling Highway, Crawley WA 6009, Australia

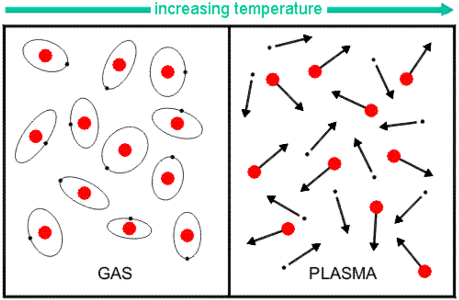
Mini-course/workshop on the application of computational mathematics to plasma physics
June 24-27, 2019 - Canberra, Australia

How to model a plasma?



A plasma is a globally neutral soup

of charged ions and electrons



separation of electrons from nucleus

Brute-force 2D model

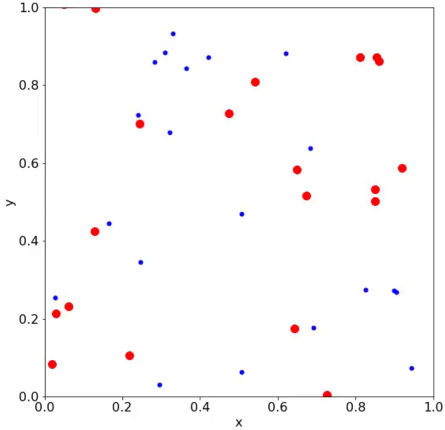
- take N electrons and N ions
- uniform distribution in space
- Gaussian distribution in velocity with variance $v_{th} = \sqrt{T/m}$
- Coulomb force over all pairs

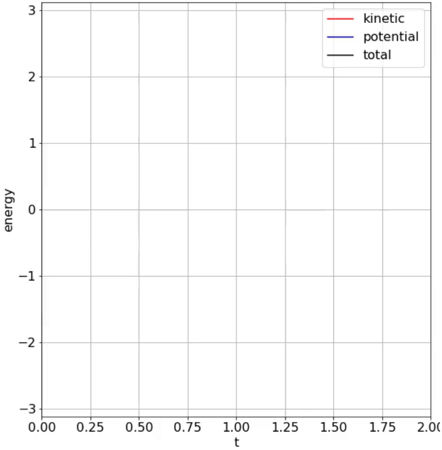
$$\dot{\boldsymbol{x}}_i = \boldsymbol{v}_i$$

$$\dot{\boldsymbol{v}}_i = \sum_{j \neq i} \frac{e_i e_j (\boldsymbol{x}_i - \boldsymbol{x}_j)}{m_i |\boldsymbol{x}_i - \boldsymbol{x}_j|^3}$$

- $2N \times 4D$ system of ODEs

Brute-force model of plasma in a box





plasma box $[0,1]^2$ *with* 20 *ions and* 20 *electrons,* $m_i/m_e = 5$, $T = 20$, *mirror boundary conditions.*

Issues with brute-force model are serious

THE UNIVERSITY OF WESTERN AUSTRALIA

- stiffness due to singular Coulomb force
- numerical accuracy and stability (poor energy conservation, crashes)
- improvement at larger temperature, lower Coulomb coupling
- computational time grows like N^2 at best, more like N^3 (accuracy)
- real plasma has $N \sim 10^{16} \Rightarrow$ intractable even on largest clusters
- boundary conditions ? realistic mass ratios ? ...

We need a reduced model $\Rightarrow$ Vlasov equation [Balescu, 1988]

- most of the time, particles are free-flying (especially at high temperature)
- Coulomb force yields small deflections (except rare face-on events)
- with $N \sim 10^{16} \Rightarrow$ mean field theory should work

Liouville equation for phase-space density

- system of ODEs is a flow on $6 \times N \sim 10^{16}$ dimensional manifold
- as a Hamiltonian system (reversible dynamics), phase-space volume $\prod_{i=1}^{N} d\boldsymbol{x}_i d\boldsymbol{v}_i$ is conserved
- Liouville equation holds for PDF $F(\boldsymbol{x}_1, \boldsymbol{v}_1, \ldots, \boldsymbol{x}_N, \boldsymbol{v}_N, t)$

$$\frac{dF}{dt} = \partial_t F + \sum_{i=1}^{N} (\dot{\boldsymbol{x}}_i \cdot \partial_{\boldsymbol{x}_i} F + \dot{\boldsymbol{v}}_i \cdot \partial_{\boldsymbol{v}_i} F) = 0$$

where $\dot{\boldsymbol{x}}_i = \boldsymbol{v}_i$ and $\dot{\boldsymbol{v}}_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} \boldsymbol{a}_{ij} = \frac{1}{m_i} \partial_{\boldsymbol{x}_i} \sum_{\substack{j=1 \\ j\neq i}}^{N} \underbrace{\frac{e_i e_j}{|\boldsymbol{x}_i - \boldsymbol{x}_j|}}_{V_{ij},\ \text{Coulomb}}$

- notice $\boldsymbol{a}_{ij} = -\boldsymbol{a}_{ji}$

Statistical average of microscopic events

- averaging over $1, 2, \ldots, N-2, N-1$ particles: collection of reduced distribution functions

$$\vdots$$

$$f^{\alpha\beta}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{x}_2, \boldsymbol{v}_2, t) = N_\alpha N_\beta \int F(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{x}_2, \boldsymbol{v}_2, \ldots, \boldsymbol{x}_N, \boldsymbol{v}_N, t) d\boldsymbol{x}_3 d\boldsymbol{v}_3 \ldots$$

$$f^{\alpha}(\boldsymbol{x}, \boldsymbol{v}, t) = N_\alpha \int F(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{x}_2, \boldsymbol{v}_2, \ldots, \boldsymbol{x}_N, \boldsymbol{v}_N, t) d\boldsymbol{x}_2 d\boldsymbol{v}_2 \ldots d\boldsymbol{x}$$

- averaging Liouville equation yields BBGKY hierarchy

$$\vdots$$

$$\partial_t f^\alpha + \boldsymbol{v} \cdot \partial_{\boldsymbol{x}} f^\alpha + \sum_\beta \int d\boldsymbol{x}_2 d\boldsymbol{v}_2 \; \boldsymbol{a}_{12}^{\alpha\beta} \cdot \partial_{\boldsymbol{v}} f^{\alpha\beta}(\boldsymbol{x}, \boldsymbol{v}, \boldsymbol{x}_2, \boldsymbol{v}_2, t) = 0$$

Vlasov via Markovian assumption

$$f^{\alpha\beta}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{x}_2,\boldsymbol{v}_2,t) = f^{\alpha}(\boldsymbol{x},\boldsymbol{v},t) f^{\beta}(\boldsymbol{x}_2,\boldsymbol{v}_2,t) + g^{\alpha\beta}(\boldsymbol{x},\boldsymbol{v},\boldsymbol{x}_2,\boldsymbol{v}_2,t)$$

product (independence) and "binary correlation function" $g^{\alpha\beta}$

similarly, $g^{\alpha\beta\gamma}$ "three-body correlations", $g^{\alpha\beta\gamma\delta}$ "four-body", ...

Closure

1. retain only "binary correlations" (truncate hierarchy)
2. Markovian assumption
   $f^{\alpha\beta} \sim f^{\alpha} f^{\beta}$ and $g^{\alpha\beta} = C[f^{\alpha}, f^{\beta}] \Rightarrow$ irreversibility

Vlasov-Boltzmann equation

$$\frac{df^{\alpha}}{dt} = \partial_t f^{\alpha}(\boldsymbol{x},\boldsymbol{v},t) + \boldsymbol{v}\cdot\partial_{\boldsymbol{x}} f^{\alpha} + \frac{e_{\alpha}}{m_{\alpha}}\boldsymbol{E}\cdot\partial_{\boldsymbol{v}} f^{\alpha} = \sum_{\beta} C[f^{\alpha}, f^{\beta}]$$

where $\boldsymbol{E}$ is the mean electric field

What is the mean electric field ?

Recall $f^{\alpha\beta} = f^\alpha f^\beta$, $\boldsymbol{a}_{12}^{\alpha\beta} = \frac{1}{m_\alpha}\partial_{\boldsymbol{x}} \frac{e_\alpha e_\beta}{|\boldsymbol{x}-\boldsymbol{x}_\beta|}$,

$$\sum_\beta \int d\boldsymbol{x}_2 d\boldsymbol{v}_2 \; \boldsymbol{a}_{12}^{\alpha\beta} \cdot \partial_{\boldsymbol{v}} f^\alpha(\boldsymbol{x},\boldsymbol{v},t) f^\beta(\boldsymbol{x}_2,\boldsymbol{v}_2,t)$$
$$= \frac{e_\alpha}{m_\alpha}\left(\partial_{\boldsymbol{x}} \underbrace{\int d\boldsymbol{x}_2 \frac{\rho(\boldsymbol{x}_2,t)}{|\boldsymbol{x}-\boldsymbol{x}_2|}}_{\Phi(\boldsymbol{x},t)}\right) \cdot \partial_{\boldsymbol{v}} f^\alpha$$

where $\rho(\boldsymbol{x},t) = \sum_\beta e_\beta n_\beta(\boldsymbol{x},t)$ is the charge density
and $n_\beta(\boldsymbol{x},t) = \int d\boldsymbol{v} f^\beta(\boldsymbol{x},\boldsymbol{v},t)$ is the particle density of species β

- $\frac{-1}{4\pi|\boldsymbol{x}-\boldsymbol{x}_2|}$ is the Green's function of the Laplacian

$$\Phi(\boldsymbol{x},t) = \int d\boldsymbol{x}_2 \frac{\rho(\boldsymbol{x}_2,t)}{|\boldsymbol{x}-\boldsymbol{x}_2|} \iff \nabla^2\Phi = -4\pi\rho$$

- mean electric field $\boldsymbol{E} = -\nabla\Phi$, $\nabla\cdot\boldsymbol{E} = 4\pi\rho$
  (is curl-free $\nabla\times\boldsymbol{E} = \boldsymbol{0}$)

Vlasov-Poisson model

$$\partial_t f_i(\boldsymbol{x}, \boldsymbol{v}, t) + \boldsymbol{v} \cdot \partial_{\boldsymbol{x}} f_i + \frac{e}{m_i} \boldsymbol{E} \cdot \partial_{\boldsymbol{v}} f_i = C[f_i, f_i] + C[f_i, f_e]$$

$$\partial_t f_e(\boldsymbol{x}, \boldsymbol{v}, t) + \boldsymbol{v} \cdot \partial_{\boldsymbol{x}} f_e - \frac{e}{m_e} \boldsymbol{E} \cdot \partial_{\boldsymbol{v}} f_e = C[f_e, f_e] - C[f_i, f_e]$$

$$\boldsymbol{E} = -\nabla \Phi, \qquad \nabla^2 \Phi = -4\pi e (n_i - n_e)$$

$$n_{i,e}(\boldsymbol{x}, t) = \int d\boldsymbol{v}\, f_{i,e}(\boldsymbol{x}, \boldsymbol{v}, t), \qquad \int d\boldsymbol{x} (n_i - n_e) = 0$$

Particle-in-Cell/Monte-Carlo method

Approximate reduced (single) particle distribution function via "macro-particles"

$$f(\boldsymbol{x}, \boldsymbol{v}, t) = \sum_{l=1}^{M} w_l \delta(\boldsymbol{x} - \boldsymbol{x}_l(t)) \delta(\boldsymbol{v} - \boldsymbol{v}_l(t))$$

where $\dot{\boldsymbol{x}}_l = \boldsymbol{v}_l$, $\dot{\boldsymbol{v}}_l = \frac{e_l}{m_l} \boldsymbol{E}(\boldsymbol{x}_l(t), t)$ and w_l is the "weight" (phase-space volume)

1. "particle push": advect $f(\boldsymbol{x}, \boldsymbol{v}, t)$ via Lagrangian method
   - Hamiltonian flow to ensure conservation laws
   - no mesh in velocity
   - drawback: noise $\sim 1/\sqrt{N}$ requires lots of markers
2. "field solve": deposit "charge" on Eulerian mesh and solve $\boldsymbol{E}$
   - standard methods (finite difference, FEM, spectral)

Particle push using leap-frog scheme

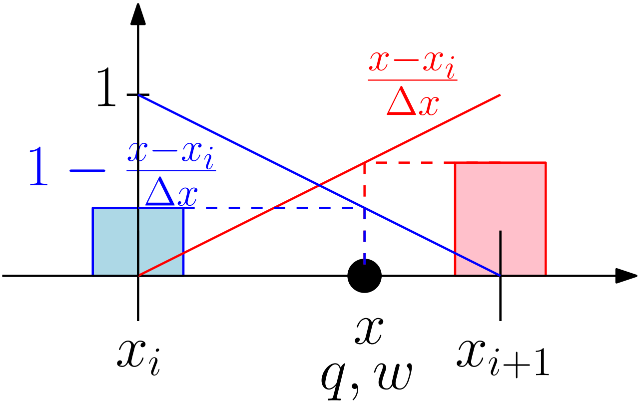
$$\begin{aligned} \dot{\boldsymbol{x}} &= \boldsymbol{v} \\ \dot{\boldsymbol{v}} &= \tfrac{e}{m}\boldsymbol{E}(\boldsymbol{x},t) \end{aligned} \qquad \Rightarrow \qquad \begin{aligned} \boldsymbol{v}_{n+1/2} &= \boldsymbol{v}_n + \tfrac{dt}{2}\tfrac{q}{m}\boldsymbol{E}(\boldsymbol{x}_n,t_n) \\ \boldsymbol{x}_{n+1} &= \boldsymbol{x}_n + dt\boldsymbol{v}_{n+1/2} \\ \boldsymbol{v}_{n+1} &= \boldsymbol{v}_{n+1/2} + \tfrac{dt}{2}\tfrac{q}{m}\boldsymbol{E}(\boldsymbol{x}_{n+1},t_n) \end{aligned}$$

- 2nd order explicit symplectic integrator
- simple, fast, robust

Python code:

```
y[:,1]+=0.5*dt*q/m*efield.eval_field(y[:,0])
y[:,0]+=dt*y[:,1]
y[:,1]+=0.5*dt*q/m*efield.eval_field(y[:,0])
```

Particle-in-Cell charge deposition



In 1D

$$\nabla^2 \Phi = \rho \Rightarrow \Phi''(x) = \rho(x)$$

Finite difference discretisation

$$\frac{\Phi_{i+1} - 2\Phi_i + \Phi_{i-1}}{(\Delta x)^2} = \rho_i$$

Dirichlet boundary conditions

$$\begin{pmatrix} -2 & 1 & 0 & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -2 & 1 \\ 0 & \cdots & 0 & 1 & -2 \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \vdots \\ \vdots \\ \vdots \\ \Phi_{N-1} \end{pmatrix} = \begin{pmatrix} (\Delta x)^2 \rho_1 - \Phi_0 \\ (\Delta x)^2 \rho_2 \\ \vdots \\ \vdots \\ (\Delta x)^2 \rho_{N-2} \\ (\Delta x)^2 \rho_{N-1} - \Phi_N \end{pmatrix}$$

Thomas algorithm for tridiagonal matrix

Our system is $\begin{pmatrix} \boldsymbol{g} & \boldsymbol{u} & 0 \\ \boldsymbol{l} & \ddots & \ddots \\ 0 & \ddots & \end{pmatrix} \boldsymbol{\Phi} = \boldsymbol{\rho}$,

where diagonal $N-1$-vector $\boldsymbol{g} = -2$, upper and lower diagonal $N-2$-vectors $\boldsymbol{u} = \boldsymbol{l} = 1$

Thomas algorithm is a clever Gauss-Jordan elimination:

1. forward upper sweep with $u'_1 = u_1/g_1$ and
$$u'_i = \frac{u_i}{g_i - l_i u'_{i-1}}, \qquad i = 2, 3, \dots, N-2$$
2. forward sweep on rhs with $\rho'_1 = \rho_1/g_1$ and
$$\rho'_i = \frac{\rho_i - l_i \rho'_{i-1}}{g_i - l_i u'_{i-1}}, \qquad i = 2, 3, \dots, N-2$$
3. backward propagation $\Phi_{N-1} = \rho'_{N-1}$
$$\Phi_i = \rho'_i - u'_i \Phi_{i+1}, \qquad i = N-2, N-3, \dots, 1$$

Bibliography I

R. Balescu, *Transport Processes in Plasmas: Classical transport*, Transport Processes in Plasmas (North-Holland, 1988), ISBN 9780444870919.