Basics of kinetic plasma simulation

- Vlasov-Poisson Equations
- Particle codes
  - Klimontovich-Dupree representation
- Vlasov codes
  - Semi-Lagrangian method and others
- PIC simulation
  - NGP & SUDS, form factors
- Computing Considerations
- Remarks

Discrete Phase Space Representation

- Vlasov-Poisson system

$$\frac{dF}{dt} \equiv \frac{\partial F}{\partial t} + v\frac{\partial F}{\partial x} + \frac{q}{m}E\frac{\partial F}{\partial v} = 0\,, \qquad E = -\frac{\partial \phi}{\partial x}$$

$$\frac{\partial^2 \phi}{\partial x^2} = -4\pi e \int (F_i - F_e)dv\,,$$

- Klimontovich-Dupree discrete representation:

$$\boxed{F(x,v,t) = \sum_{j=1}^{N} \delta(x - x_j)\delta(v - v_j)}$$

$$n(x) = \int F dv = \sum_{j=1}^{N} \delta(x - x_j)$$

$$J(x) = q\int vF dv = q\sum_{j=1}^{N} v_j \delta(x - x_j)$$

$$p(x) = m\int v^2 F dv = m\sum_{j=1}^{N} v_j^2 \delta(x - x_j)$$

$$\boxed{\frac{\partial}{\partial t} = \frac{d\mathbf{x}_j}{dt}\cdot\frac{\partial}{\partial \mathbf{x}_j} + \frac{d\mathbf{v}_j}{dt}\cdot\frac{\partial}{\partial \mathbf{v}_j}}$$

- Equations of Motion:

$$\boxed{\frac{dx_j}{dt} = v_j\,, \qquad \frac{dv_j}{dt} = \frac{q}{m}E(x_j)}$$

- K-D discrete representation + Equations of Motion: $\longrightarrow$ $\displaystyle \frac{dF}{dt} \equiv \frac{\partial F}{\partial t} + \frac{dx}{dt}\frac{\partial F}{\partial x} + \frac{dv}{dt}\frac{\partial F}{\partial v} = 0$

Two Different Approaches for Plasma Simulations

- **Vlasov Equation**

$$\frac{dF}{dt} \equiv \frac{\partial F}{\partial t} + \mathbf{v} \cdot \frac{\partial F}{\partial \mathbf{x}} + \frac{q}{m}\left(\mathbf{E} + \frac{1}{c}\mathbf{v} \times \mathbf{B}\right) \cdot \frac{\partial F}{\partial \mathbf{v}} = 0$$

$$\boxed{\nabla_{\mathbf{v}} \cdot [\mathbf{v} \times \mathbf{B}(\mathbf{x})] = 0 \ ?}$$

-- Particle codes: j - particle

$$\frac{d\mathbf{x}_j}{dt} = \mathbf{v_j} \ , \qquad \frac{d\mathbf{v}_j}{dt} = \frac{q}{m}\left(\mathbf{E} + \frac{1}{c}\mathbf{v}_j \times \mathbf{B}\right)_{\mathbf{x}_j}$$

representation

$$F = \sum_{j=1}^{N} \delta(\mathbf{x} - \mathbf{x}_j)\delta(\mathbf{v} - \mathbf{v}_j)$$

-- Klimontovich and Dupree representation

-- Continuum codes: many ways to solve the equation, -- e.g.,

$$\frac{d\mathbf{x}_g}{dt} = \mathbf{v}_g, \qquad \frac{d\mathbf{v}_g}{dt} = \frac{q}{m}\left(\mathbf{E} + \frac{1}{c}\mathbf{v}_g \times \mathbf{B}\right)_{\mathbf{x}_g}$$

$$F(\mathbf{x}_g + d\mathbf{x}_g, \mathbf{v}_g + d\mathbf{v}_g, t + dt) = F(\mathbf{x}_g, \mathbf{v}_g, t)$$

1. Semi-Lagrangian method [Cheng and Knorr, JCP '76],
2. Fourier Transform method [Denavit and Kruer, Phys. Fluids'72],
3. Finite-Difference method.

- **Poisson's Equation: same for both**

$$\nabla^2\phi = -4\pi e \int (F_i - F_e) d\mathbf{v}, \quad \mathbf{E} = -\nabla\phi$$

-- Poisson's equation can be solved on a spatial grid

-- It can also be solved through direct calculations using particles.

Coulomb's Law:

$$F_j = q_j q_i \frac{(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^3}$$

$$\mathbf{E}_j = \sum_{i=1}^{N} q_i \frac{(\mathbf{x}_j - \mathbf{x}_i)}{|\mathbf{x}_j - \mathbf{x}_i|^3}$$

Self-consistent N-body simulation: N^2 calculations
Tree Codes: NlogN calculations
Field Methods: N calculations

Hierarchical force calculation algorithms (e.g. Greengard 1990) provide fast, general, and reasonably accurate approximations for gravity and other inverse-square forces. They fill the gap between direct sum methods, which are accurate and general but require O(N2) operations for a complete force calculation, and field methods, which have limited generality and accuracy but require only O(N) operations. All hierarchical methods partition the mass distribution into a tree structure, where each node of the tree provides a concise description of the matter within some spatial volume. This tree structure is used to introduce explicit approximations into the force calculation. Hierarchical methods require either O(N) or O(N log N) operations per force calculation, depending on the representation employed. The algorithm described here improves on an earlier hierarchical O(N log N) method (Barnes & Hut 1986, hereafter BH86) which has been widely employed in astrophysical simulations.[http://www.ifa.hawaii.edu/faculty/barnes/treecode/treeguide.html]

Particle-In-Cell (PIC) Simulations --

The simulation particles can be regarded as Lagrangian markers embedded randomly in the Vlasov fluid moving with it through phase space [see, Morse and Nielson, 1969]. This Monti-Carlo viewpoint explicitly recognizes the random samplings of the particle-in-cell models in the phase space through collective effects, i.e., the particles influence each other through the self-consistent fields, rather than remaining independent markers in the phase-fluid. Therefore, we can use analytical methods analogous to those of normal plasma kinetic theory to understand their behavior[see, Birdsall & Langdon, 1991].

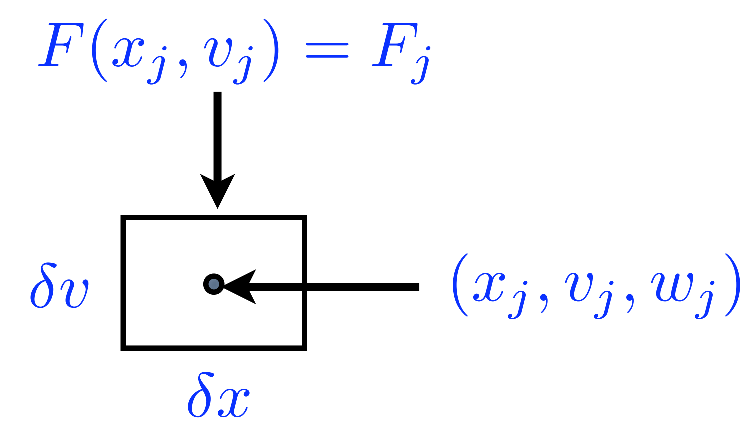
Vlasov Continuum (VCON) Simulation --

Direct numerical integration of the Vlasov equation continuously generates filamentation of the distribution function in the phase space, the ripples. The challenge here is to develop affordable computational methods to dispose of these ripples without affecting the physics at hand [see, Cheng and Knorr, 1976]. Again, we need normal kinetic theory to understand the consequence of this ripple reduction on entropy production and, in turn, its effect on relevant (lower) velocity moments.

Kinetic Simulation of Plasmas via Lagrangian Method

$$\frac{d}{dt}F(x,v,t)=0$$

- Particle Simulation [Dawson et al., 1968; Birdsall et al., 1968]



$$F(x,v,t)=\sum_{j=1}^{N} w_j[\delta(x-x_j(t)]\delta[v-v_j(t)],$$

$$F(x,v,t+\Delta t)=\sum_{j=1}^{N} w_j[\delta(x-x_j(t+\Delta t)]\delta[v-v_j(t+\Delta t)]$$

$$w_j=1 \qquad x_j(t+\Delta t)=x_j(t)+v_j(t)\Delta t \qquad v_j(t+\Delta t)=v_j(t)+\frac{q}{m}E[x_j(t)]\Delta t$$

- No re-normalization of the particle weight, w_j, in the simulation, i.e., it remains constant.

- Vlasov Simulation: aka Continuum Simulation [Cheng and Knorr, 1976]

$$F(x,v,t)=\sum_{j=1}^{N} F_j(x,v,t)$$

$$F_j(x+\Delta x,v+\Delta v,t+\Delta t)=F_j(x,v,t)$$

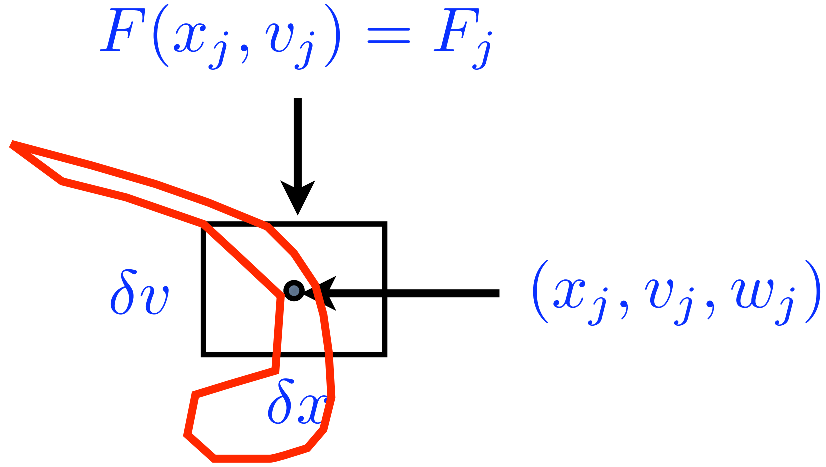
$$\Delta x=v\Delta t, \qquad \Delta v=\frac{q}{m}E\Delta t$$

- By re-normalizing F_j on a grid at every time step, the scheme is called the Vlasov simulation

- Without re-normalizing F_j, the scheme becomes similar to particle simulation, since $F_j=const.$

Kinetic Simulation of Plasmas via Lagrangian Method (cont.)

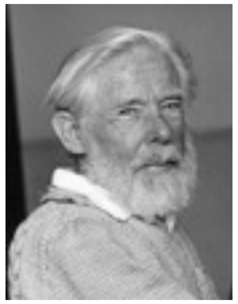
- The phase space element, δxδv, will contort and filament in time:
  - particle simulation, without re-normalization, can't describe this phenomena, so it has to use many particles so as to approximate the phase space dynamics in long time simulations. Eventually, one would run of the particles and the simulation can't describe the correct physics.
  - Vlasov simulation needs to use a finer grid in phase space so as not to introduce unwanted physics, such as artificial dissipation through coarse graining of the velocity space. Artificial dissipation would result in artificial increase in entropy and, in turn, gives rise to artificial particle and energy transport.



- Another difference between the two:
  - particle simulation uses only 3D spatial grids for charge calculation, which is easier for massively parallel computation, but has noise problem
  - Vlasov simulation uses 3D spatial grids + 3D velocity space grids, which is harder for massively parallel computation, but has no noise problem.
- One more difference:
  - without re- normalization, particle simulation samples the phase space continuously
  - with re-normalization, Vlasov simulation sees a discrete phase space

Particle Simulation of Plasmas: contributed by many

- Dawson's sheet model ('62): 1000 sheets in one dimensional simulation; Birdsall and Buneman were even earlier.



- Finite-size particles and Particle-in-Cell (PIC) simulation : [Dawson et al. '68; Birdsall et al. '68]
  -- Based on Klimontovich-Dupree representation
  -- Close interactions are modified
  -- Debye shielding w/o $n\lambda_D^3 >> 1$
  -- Long range interactions are intact
  -- Coulomb interactions become collisionless [Okuda et al. '72]
  -- Collisions can be re-introduced as subgrid phenomena.
  -- N^2 calculations reduce to $N\log N$

- Short wavelength and high frequency particle noise is minimized through the charge sharing and charge smoothing schemes and particle noise can be studied by Fluctuation-Dissipation Theorem [see, e.g., Klimontovich '67, Langdon '79]

- Landau Damping damps away long wavelength numerical noise due to finite number of particles and most of the numerical inaccuracies.

Debye Shielding & Debye Cloud

$$\frac{dF_\alpha}{dt} \equiv \frac{\partial F_\alpha}{\partial t} + v\frac{\partial F_\alpha}{\partial x} - \frac{q_\alpha}{m_\alpha}\frac{\partial \phi}{\partial x}\frac{\partial F_\alpha}{\partial v} = 0$$

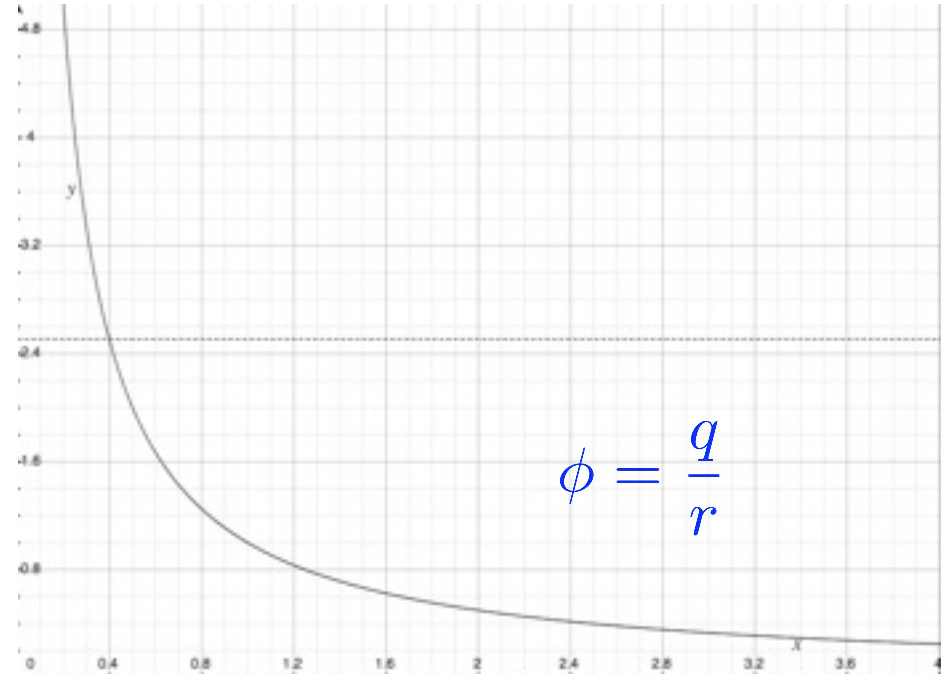
$$F_\alpha = F_{0\alpha} + \delta f_\alpha \qquad F_\alpha = \frac{n_0}{\sqrt{2\pi}v_{t\alpha}} e^{-v^2/2v_{t\alpha}^2}$$

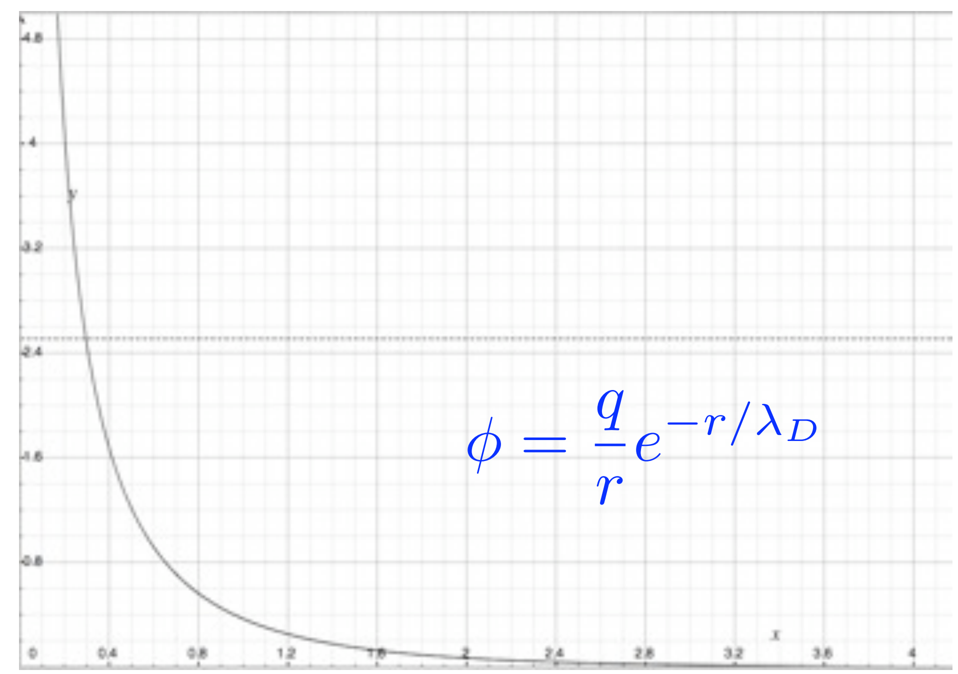
$$\delta f_\alpha = -\frac{q_\alpha}{T_\alpha}\phi F_\alpha$$

$$\frac{\delta n_\alpha}{n_0} = -\frac{q_\alpha}{T_\alpha}\phi$$

$$\nabla^2\phi - \frac{\phi}{\lambda_D^2} = 0 \qquad \frac{1}{\lambda_D^2} = \frac{1}{\lambda_{Di}^2} + \frac{1}{\lambda_{De}^2}$$

$$\nabla^2 \rightarrow \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial}{\partial r})$$ -- spherical coord.





Shielded particle:

$$\phi = \frac{q_\alpha}{r}e^{-r/\lambda_D}$$

Bare particle:

$$\phi = \frac{q_\alpha}{r}$$

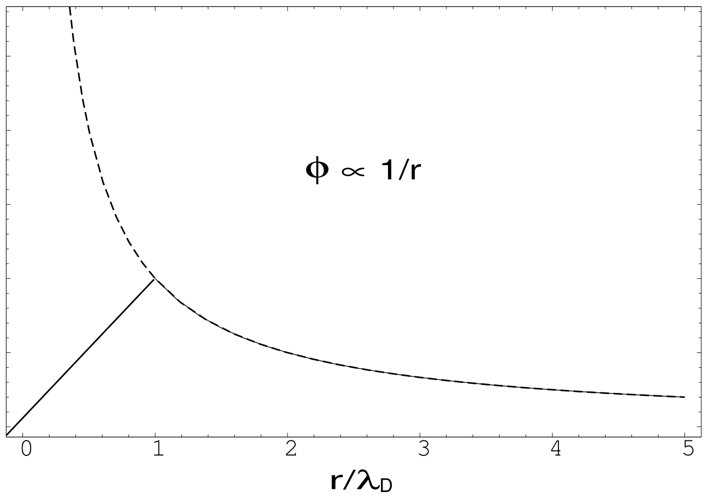
Shielding cloud consisting of electrons and ions: quasineutral

“Simulation of Microscopic processes in Plasma,” Viktor Dycek, Proceedings of the Invited Papers for International Conference on Plasma Physics, Kiev, USSR, Vol. 2, World Scientific (April, 1987)
“Theory and Simulation of the Test Particle Debye Cloud,” Huang, Hsin-Chien, PH.D. Thesis, UCLA (1988).

Some Basic Principles of PIC

- Particle simulation actually solves the following equation: identical to Vlasov equation for a conservative system.

$$\frac{\partial F}{\partial t} + \frac{\partial}{\partial \mathbf{x}} \cdot (F\frac{d\mathbf{x}}{dt}) + \frac{\partial}{\partial \mathbf{v}} \cdot (F\frac{d\mathbf{v}}{dt}) = 0$$



- Shape function, S, for finite-size particles

$$n_\alpha(\mathbf{x}) = \int F_\alpha d\mathbf{v} = \sum_{j=1}^{N} \delta(\mathbf{x} - \mathbf{x}_{\alpha j}) \rightarrow \sum_{j=1}^{N} S(\mathbf{x} - \mathbf{x}_{\alpha j})$$

- Force modification for finite-size particles mimicking the Debye cloud:

$$\mathbf{F}_i = q_i \mathbf{E}(\mathbf{x}_i) = \sum_{j \neq i} q_i q_j (\mathbf{x}_i - \mathbf{x}_j)/\lambda_D^3$$

- But, the long range force is intact

$$\mathbf{F}_i = q_i \mathbf{E}(\mathbf{x}_i) = \sum_{j \neq i} q_i q_j (\mathbf{x}_i - \mathbf{x}_j)/|\mathbf{x}_i - \mathbf{x}_j|^3$$

- Thus, PIC achieves Debye shielding without using millions of particles to satisfy $n\lambda_D^3 >> 1$

- Landau damping plays a crucial part for numerical stability.

One-Dimensional Particle Simulation

- Charge density given by each particle -- α is the species and *j* is the particle

$$\rho(x) = \sum_\alpha q_\alpha \sum_j \delta(x - x_{\alpha j})$$

- Poisson's equation

$$\frac{\partial E}{\partial x} = 4\pi\rho$$

- Force on the j-th particle

$$F_{\alpha j} = q_\alpha E(x_{\alpha j}) = q_\alpha \int_\alpha E(x)\delta(x - x_{\alpha j})dx$$

- Particle pushing

$$\frac{dx_{\alpha j}}{dt} = v_{\alpha j}$$

$$\frac{dv_{\alpha j}}{dt} = \frac{F_{\alpha j}}{m_\alpha} = \frac{q_\alpha}{m_\alpha}E(x_{\alpha j})$$

- In order to solve Poisson's equation on the grid, we need to put charge density on the grid.

One-Dimensional PIC (cont.)

Nearest Grid Point (NGP) -- noisy for short wavelength modes

- Put the *j*-th particle on the nearest grid g

$$\rho(x) = \sum_{\alpha} q_\alpha \sum_{j} \delta(x - x_{\alpha j}) \longrightarrow \rho(x) = \sum_{g} \sum_{\alpha} (\rho^{NGP})_{\alpha,g} \delta(x - x_g) \quad j \in g$$

$$(\rho^{NGP})_{\alpha,g} = q_\alpha \sum_{j \in g} 1$$

Charge density is now given on a grid rather than a collection of of point particles.

- Force on the *j*-th particle

$$F_{\alpha j} = q_\alpha \int E(x)\delta(x - x_g)dx \qquad j \in g$$

Subtracted Dipole Scheme (SUDS)

$$\rho(x) = \sum_{\alpha} q_\alpha \sum_{j} S(x - x_{\alpha j})$$

$$F_{\alpha j} = q_\alpha E(x_{\alpha j}) = q_\alpha \int E(x) S(x - x_{\alpha j})dx$$

smoothing out noise for short wavelength modes

$$S(x) = exp(-x^2/2a^2)/(\sqrt{2\pi}a)$$ -- shape factor and a is the particle size

$$\int S(x)dx = 1$$

$$S(k) = exp(-k^2a^2/2)$$

One-Dimensional PIC - SUDS

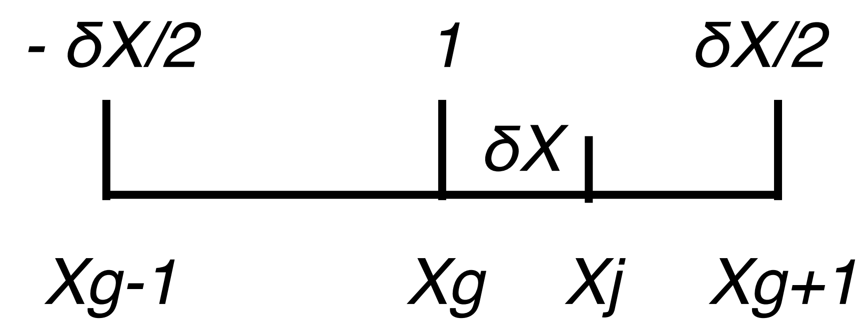
$$\rho(x) \approx \sum_{\alpha} q_\alpha \sum_j S(x - x_g) + (x_g - x_{\alpha j}) S'(x - x_g)$$

$$\approx \sum_{\alpha} q_\alpha \sum_j S(x - x_g) + \frac{x_g - x_{\alpha j}}{2\Delta x} \times \left[S(x - x_{g-1}) - S(x - x_{g+1})\right]$$

$$\downarrow$$

$$\rho(x) \approx \sum_{\alpha,g} \left[(\rho^{NGP})_{\alpha,g} + (\rho^D)_{\alpha,g-1} - (\rho^D)_{\alpha,g+1}\right] S(x - x_g)$$

$$= \sum_{\alpha,g} (\rho^P)_{\alpha,g} S(x - x_g)$$



$$(\rho^D)_{\alpha,g} = q_\alpha \sum_{j \in g} (x_{\alpha j} - x_g)/2\Delta x$$

- Density is now given on a grid with smoothing for short wavelength noise.
- Higher order expansions can be used to further reduce noise, or just simply use more particles.

One-Dimensional PIC - SUDS (cont.)

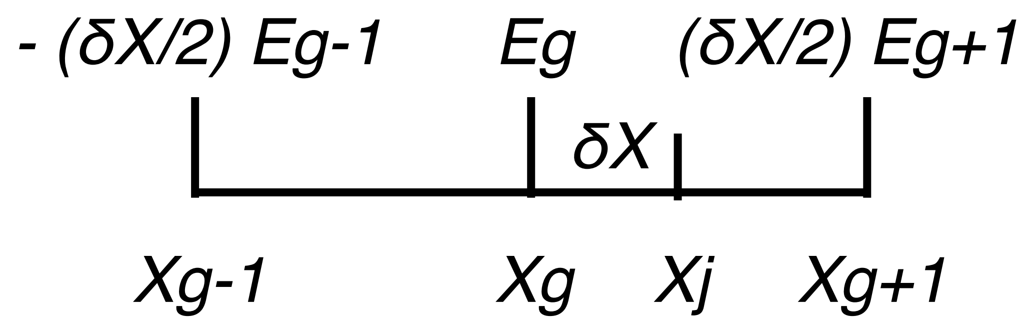
- Force on the *j*-th particle now becomes

$$F_{\alpha j} \approx q_\alpha \int E(x) S(x - x_g) dx + q_\alpha \frac{x_g - x_{\alpha j}}{2\Delta x} \int E(x) \left[S(x - x_{g-1}) - S(x - x_{g+1}) \right] dx$$

- Applying Fourier transforms in *k*-space

$$S(x) = \sum_k S^*(k) exp(-ikx)$$

$$E(k) = (1/L) \int E(x) exp(-ikx) dx$$



- We obtain $F_{\alpha j} \approx q_\alpha \tilde{E}(x_g) + q_\alpha \frac{x_{\alpha j} - x_g}{2\Delta x} \left[\tilde{E}(x_{g+1}) - \tilde{E}(x_{g-1}) \right]$

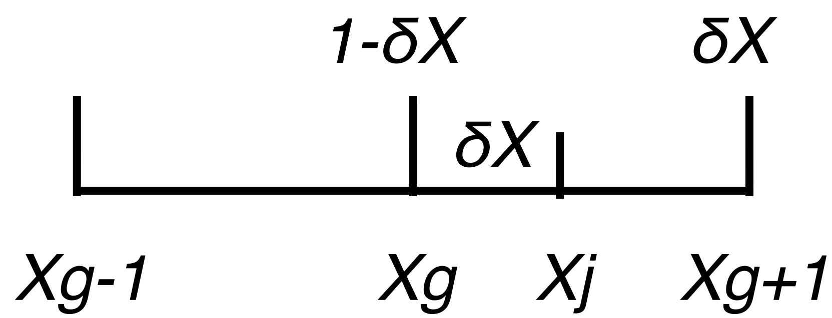
where $\tilde{E}(x) = \sum_k \tilde{E}(k) exp(ikx) = \sum_k S^*(k) E(k) exp(ikx)$

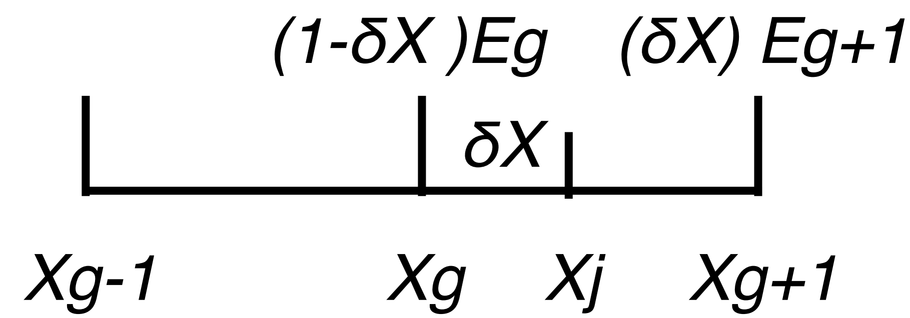
- Poisson's equation becomes $\tilde{E}(k) = (-4\pi i/k) S^*(k) \rho(k)$

$$= \left[-\frac{4\pi i}{k} |S(k)|^2 \right] \sum_g \left[exp(-ikx_g) \sum_\alpha (\rho^P)_{\alpha,g} \right]$$

- Form factor to reduce short wavelength noise : $|\tilde{S}(k)|^2 = e^{-k^2 a^2}$ a - particle size ≈ grid size

One-Dimensional PIC - Linear Interpolation



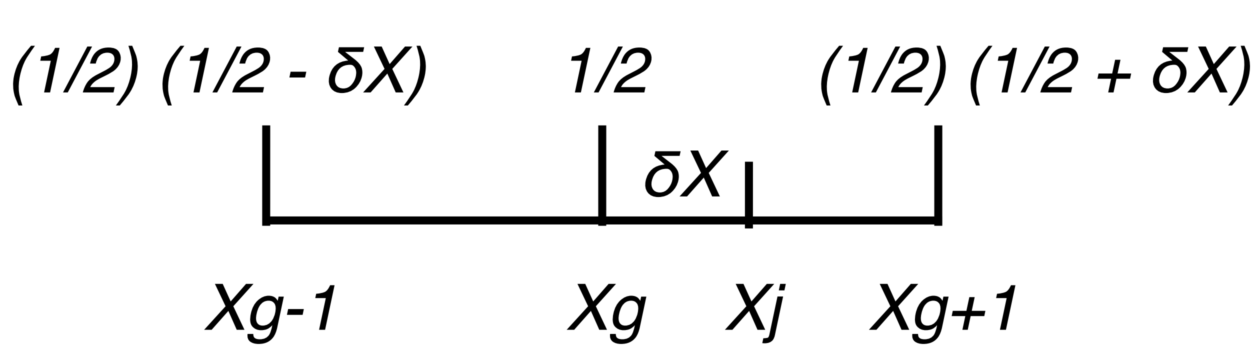


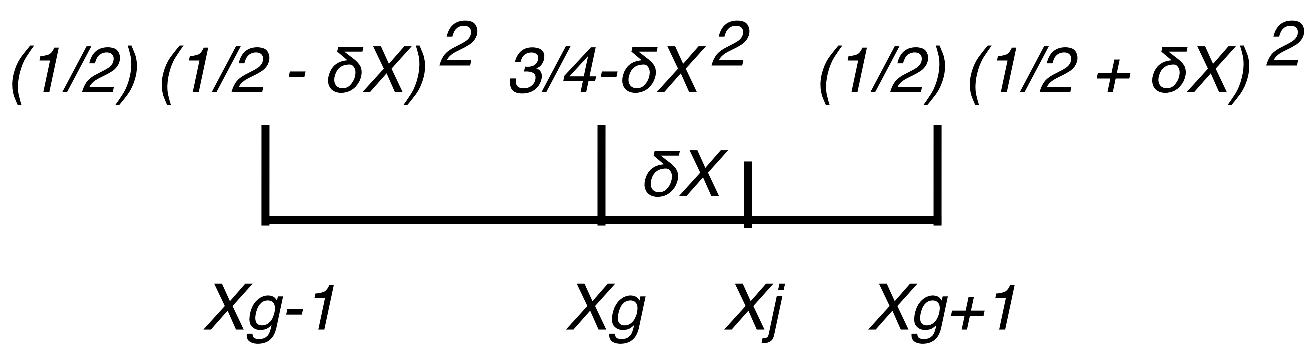
-- SUDS involves three grid points for each particle and is the least noisy.

-- LI involves only two grid points and is a bit noisier, but is most popular.

-- NGP involves only one grid point and is most noisy, but is computational most efficient.

One-Dimensional PIC - MSUDS & Quadratic Spline



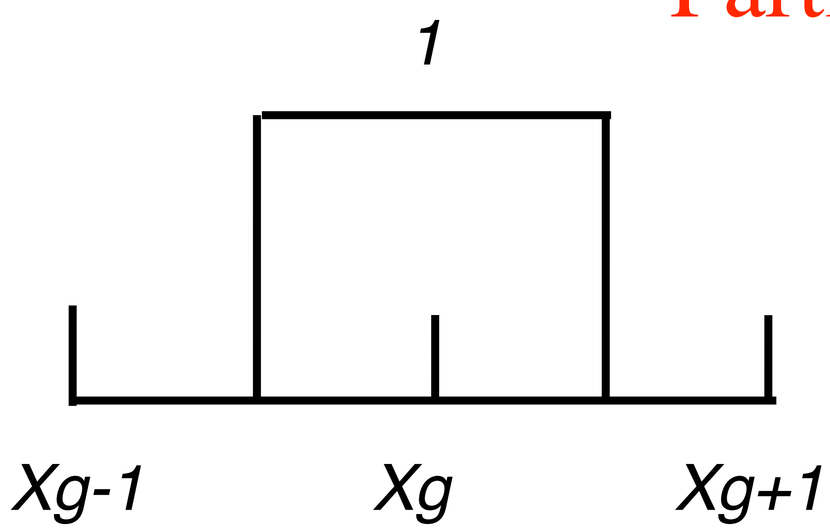


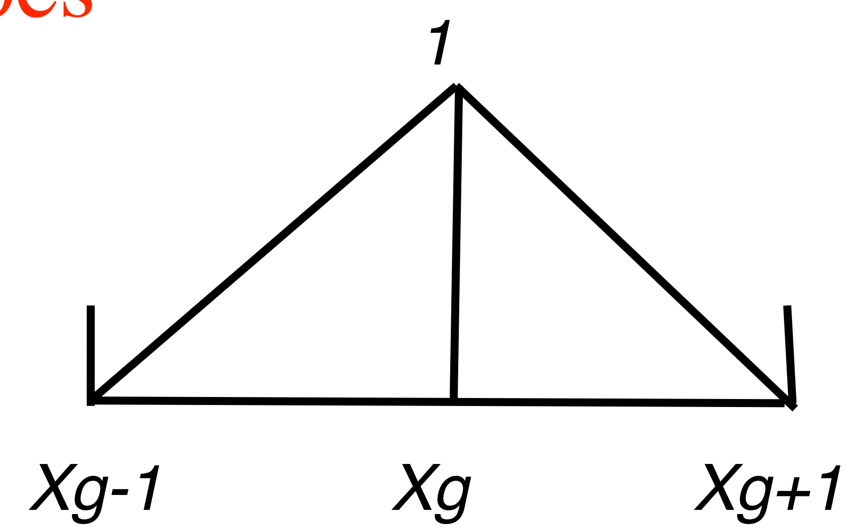
-- QS is the best, but is also computational much more expensive.

-- Therefore, it is better off to use more particles with simpler interpolation schemes.

What are the shape functions, S(x), for NGP, LI, SUDS, MSUDS and QS? Hint: a step function, a triangle, a hat, and ... ?

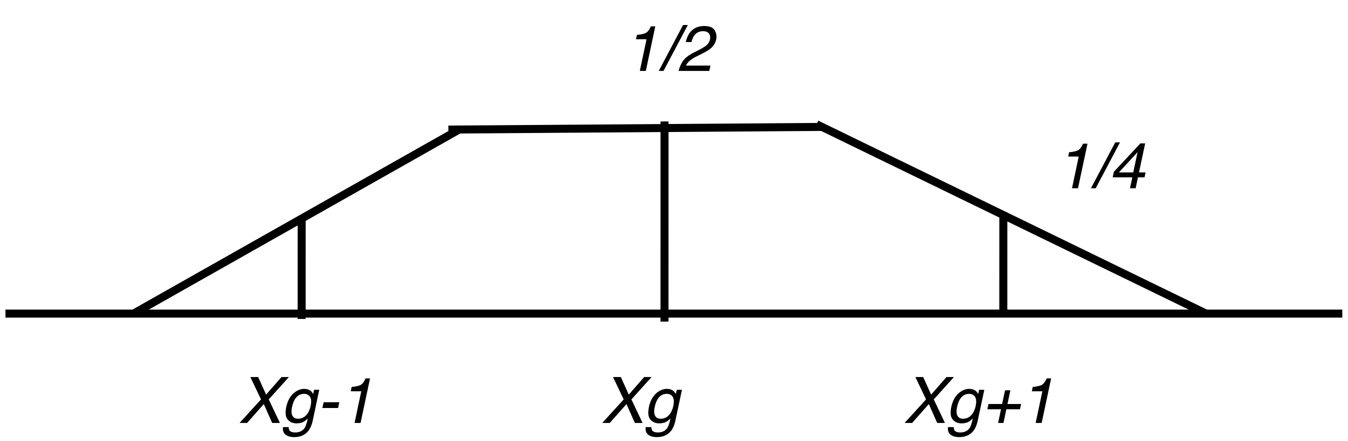
Particle Shapes

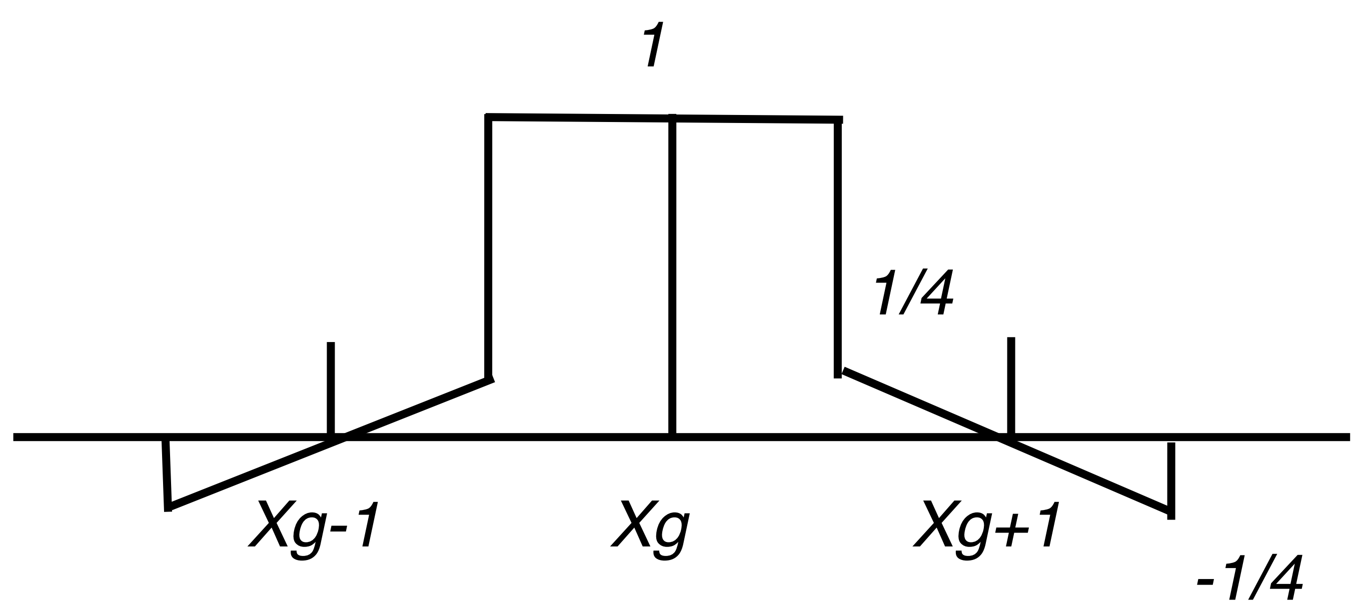




NGP

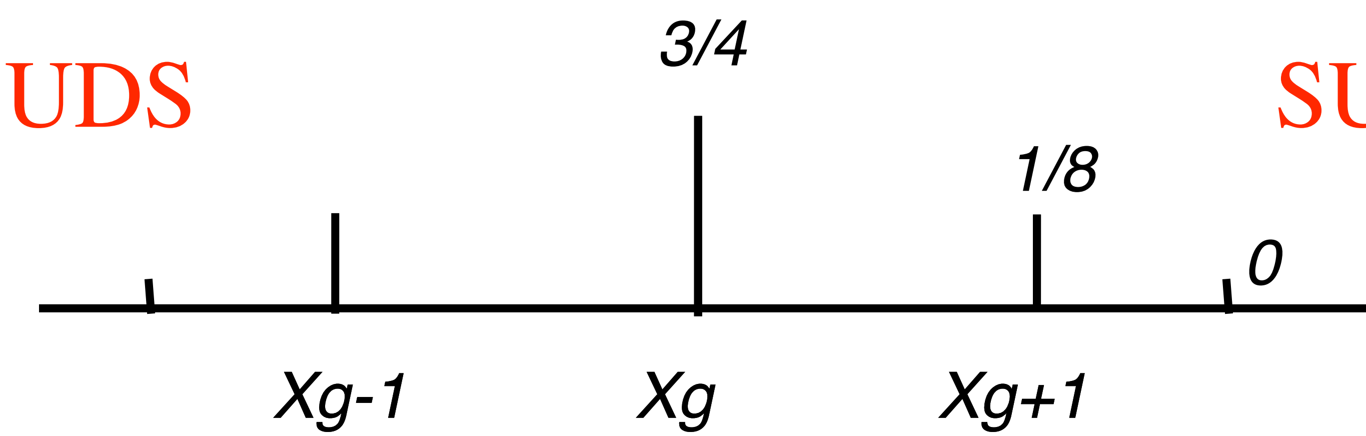
Linear Interpolation





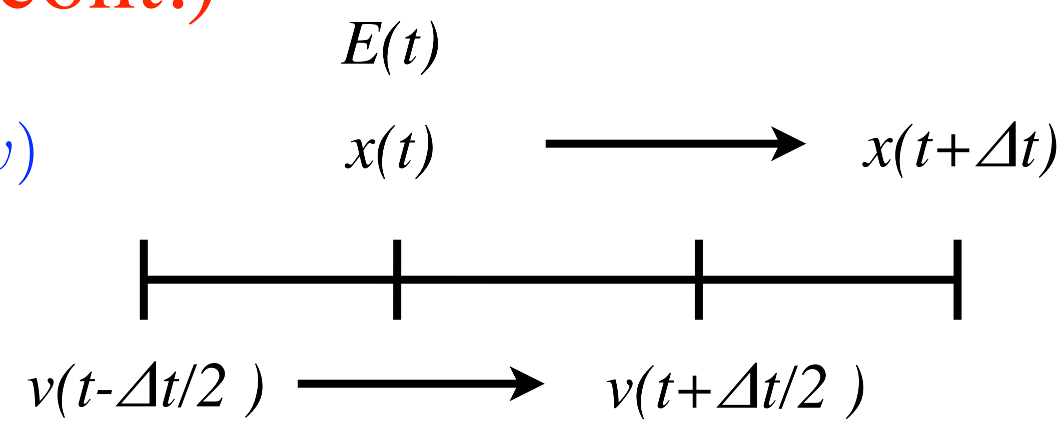
MSUDS

SUDS



QS is nearly Gaussian

One-dimensional PIC (cont.)



- Leap-frog Particle Pusher: $\dfrac{dx}{dt} \neq g(x), \quad \dfrac{dv}{dt} \neq g(v)$

$$x(t+\Delta t) = x(t) + v(t+\Delta t/2)\Delta t$$

$$v(t+\Delta t/2) = v(t-\Delta t/2) + (q/m)E(t)\Delta t$$

Predictor-Corrector for $\dfrac{dz}{dt} = g(z,t)$

$$z^*(t+\Delta t) = z(t-\Delta t) + g[z(t),t]\Delta t$$

$$z(t+\Delta t) = z(t) + \frac{1}{2}g[z(t),t]\Delta t + \frac{1}{2}g[z^*(t+\Delta t), t+\Delta t]\Delta t$$

- Energy Conservation

From $\dfrac{\partial F}{\partial t} + v\dfrac{\partial F}{\partial x} + \dfrac{q}{m}E\dfrac{\partial F}{\partial v} = 0$

Using Klimontovich-Dupree Representation to obtain

$$\sum_\alpha \sum_j m_\alpha v_{\alpha j}^2/2 - \int EJdx = 0\,, \qquad J = \sum_\alpha q_\alpha \sum_j v_{\alpha j}\delta(x - x_{\alpha j})$$

Charge conservation: $\dfrac{\partial \rho}{\partial t} + \dfrac{\partial J}{\partial x} = 0$

From Poisson's equation to obtain $\dfrac{\partial E}{\partial t} + 4\pi J = 0$

We then obtain $\displaystyle\sum_\alpha \sum_j m_\alpha v_{\alpha j}^2/2 + L\sum_k |E(k)|^2/8\pi = 0\,, \quad E(k) = \tilde{E}(k)/S_k^*$

Basic Differences in PIC and Continuum Codes

- **Memory requirements:**
  - -- Particle codes/ Continuum codes
    - = NxNyNz (1 + 5n) / NxNyNzNuNv
    - = (1 + 5n) / NuNv [= (1 + 5x20)/(10 x10) ≈ 1]
  - -- Convergence for PIC: n ≈ 20 per cell OK for 3D
  - -- Convergence for continuum: Nu > 10 , Nv > 10 what about filamentation???

- **Resolution in velocity space for one cell:**
  - -- PIC: n (20)
  - -- Continuum: NuNv (100), better

- **Resolution in velocity space when summing over NxNyNz**
  - -- PIC: n NxNyNz (increase in resolution with simulation domain), much better
  - -- Continuum: NuNv (no change in resolution by increasing simulation domain)

- **Numerical properties:**
  - -- PIC: long term memory, t = ∞ ⟶ numerical noise
  - -- Continuum: short term memory, t = time step ⟶ numerical dissipation

Special Considerations for Particle Pushing

Mixed representations:

x_j, v_j, F_j and w_j -- data for j-th particle

ρ_g and Φ_g - data for the g-th grid

- Charge: scatter and add operation -- particle to grid

$\rho_g(x_j) = \rho_g(x_j) + w_j$, j --- particle

- Solve: grid only

$\nabla^2 \Phi_g = \rho_g$,

- Force: gather operation -- grid to particle

$F_j = \Phi_g(x_j)$

- Push: particle only

$x_j = x_j + v_j$
$v_j = v_j + F_j$

- Sensitive to memory bandwidth and memory latency

Simple Models for Collisions

$$\boxed{C(F_e) = \frac{\nu_{ei}}{2sin\theta}\frac{\partial}{\partial\theta}\left[sin\theta(\frac{\partial F_e}{\partial\theta})\right]}$$

Lorentz Model
- conserve number density and energy

Let $cos\theta = v_{\|}/v$, $sin\theta = v_{\perp}/v$

$$\frac{\partial}{\partial\theta} = -v_{\perp}\frac{\partial}{\partial v_{\|}} + v_{\|}\frac{\partial}{\partial v_{\perp}}$$

Let $F_e = f_{\|e}f_{\perp e}^{M}$

$v_{\|} \rightarrow v$

$$\boxed{C(f_e) = \nu_{ei}\frac{\partial}{\partial v}\left(v_{te}^2\frac{\partial f_e}{\partial v} + v f_e\right)}$$

Lenard-Bernstein Model
- conserve number density only

$$\boxed{C(\delta f_e) = -\nu_{ei}\left(\delta f_e - F_{Me}\int \delta f_e d\mu dv_{\|}\right)}$$

Krook (BGK) Model

- Collisions are strong short range interactions within the Debye sphere with

$$g = \frac{1}{n\lambda_D^3} \ll 1,$$

whereas Poissons's equation describes weak long range interactions in scales much longer than the Debye length.

Lorentz Model for PIC

$$\frac{\partial F}{\partial t} = \frac{\nu}{2}\frac{\partial}{\partial \lambda}\left[(1-\lambda^2)\frac{\partial F}{\partial \lambda}\right]$$

$\lambda \equiv v_{\parallel}/v = cos\theta$

$v_{\perp}/v = sin\theta$

Let $\langle\lambda\rangle = \int_{-1}^{1} \lambda F d\lambda$ $\quad\quad \langle\lambda^2\rangle = \int_{-1}^{1} \lambda^2 F d\lambda$

$$\frac{\partial\langle\lambda\rangle}{\partial t} = -\nu\langle\lambda\rangle$$

$$\frac{\partial\langle\lambda^2\rangle}{\partial t} = \nu(1-3\langle\lambda^2\rangle)$$

$\sigma^2 \equiv \langle\lambda^2\rangle - \langle\lambda\rangle^2$ standard deviation

$$\frac{\partial\sigma^2}{\partial t} = \nu\left[1-3\langle\lambda^2\rangle + 2\langle\lambda\rangle^2\right]$$

$\sigma^2 \to 0$ $\quad\quad \dfrac{\partial\sigma^2}{\partial t} = \nu\left[1-\langle\lambda\rangle^2\right]$

Monte Carlo Collisions: $\boxed{v_{\parallel}(1-\nu\Delta t) \pm v_{\perp}(\nu\Delta t)^{1/2} \to v_{\parallel}}$ $\quad \boxed{v_{\parallel}^2 + v_{\perp}^2 = cons.}$

$\pm$ randomly generated

[Boozer and Kuo-Petravic, PF ‘81]

Lenard-Bernstein Model for VCON

Fourier Transforms:

$$F(x) = \sum_k \bar{F}(k) e^{ikx} \qquad \bar{F}(k) = \frac{1}{N_x} \sum_k F(x) e^{-ikx}$$

$$F(v) = \sum_q \bar{F}(q) e^{-iqv} \qquad \bar{F}(q) = \frac{1}{N_v} \sum_v F(v) e^{iqv}$$

One Dimensional Vlasov-Poisson system:

$$\frac{\partial F}{\partial t} + v\frac{\partial F}{\partial x} - \frac{q_\alpha}{m}\frac{\partial \phi}{\partial x}\frac{\partial F}{\partial v} = \nu \frac{\partial}{\partial v}\left(v_t^2 \frac{\partial \delta F}{\partial v} + vF \right)$$

$$\frac{\partial \bar{F}}{\partial t} - ik\frac{\partial \bar{F}}{\partial q} + \cdots = -\nu \left(v_t^2 q^2 \bar{F} + q\frac{\partial \bar{F}}{\partial q} \right)$$

$$k^2 \phi = 4\pi e \int (F_i - F_e)\, dv$$

Comparisons between Particle-In-Cell and Vlasov Continuum Methods

Since the two methods differ fundamentally in their approach, the agreement found confirms their validity. However, the problems considered have shown limitations in both methods, which must be taken into account in the physical interpretation of numerical simulation results. Discrete particle effects in particle simulations, which are particularly evident in regions of low density in phase space, yield beaming instabilities which must be minimized or accounted for in the physical interpretation of the results. Similarly, solutions of the Vlasov equation tend to develop increasingly fine structures with increasing time. The fine structures are suppressed by truncation of the Fourier expansions to a finite number of modes, but enough modes must be retained to make the half-widths $\Delta x = 1/2m_{\max}$ and $\Delta v = \pi/q_{\max}$ small compared with the characteristic lengths and velocities of the phenomena being considered. As indicated in the dis-

[Denavit and Kruer, Phys. Fluids 14, 1782 (1971)]

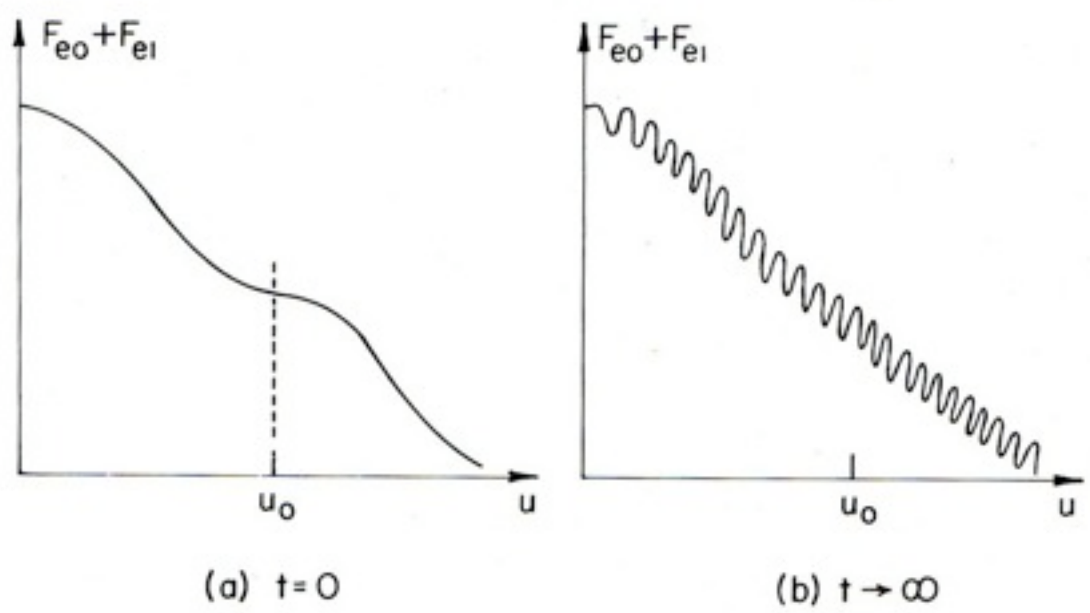
Comparisons between Particle-In-Cell and Vlasov Continuum Methods (cont.)

$$\frac{dF}{dt} \equiv \frac{\partial F}{\partial t} + \mathbf{v} \cdot \frac{\partial F}{\partial \mathbf{x}} + \frac{q}{m}\left(\mathbf{E} + \frac{1}{c}\mathbf{v} \times \mathbf{B}\right) \cdot \frac{\partial F}{\partial \mathbf{v}} = 0$$

Let $F \propto exp(i\mathbf{k} \cdot \mathbf{x})$

For ***E = 0*** and ***B= 0,*** $F \propto exp(-i\mathbf{k} \cdot \mathbf{v}t)$ -- free streaming

It arises because a particle perturbed at t = 0, carries the memory of its perturbation with it for all times. wherever it goes. This memory can be erased by collisions or coarse graining in VCON codes. For PIC codes, this memory is carried by the particle for all times until it runs out of the resolution prescribed at t = 0.



Krall & Trievelpiece, ‘73

Remarks

-- PIC accuracy improves with wavelengths, but it is inaccurate for short wavelength modes. Computational time is proportional *N log N*, but gather/scatter operations involve random access.
-- N^2 calculations are highly accurate for both long and short wavelength modes, but they need more computational time. Need algorithms for Debye shielding effects.
-- For PIC, the phase space volume associated with each particle is fixed for the during of the simulation. Thus, for accurate description of the filamentation in phase space, ones need more and more particles for long time simulation.
-- Vlasov continuum (VCON) methods in phase space can be easily parallelizable, but ripple (wrinkle) removal has to be handled carefully without sacrificing the nonlinear physics. Specifically, the ripple removal, acting like collisions, introduces numerical dissipation. We need to know the consequences.
-- PIC is similar to Lattice Boltzmann Methods (LBM) -- Boltzmann equation + BGK operator.

-- Some of the codes for kinetic simulation of core transport for fusion plasmas:

GTS - Global General Geometry PIC (PPPL)
GTC - Global PIC (UCI &PPPL)
GEM - Wedge PIC (Colorado)
GT3D - Wedge PIC (JAEA, Japan)
ORB5 - Global General Geometry PIC (CRPP, Switzerland)

GS2 - Local VCON (Maryland)
GENE - Local VCON (IPP, Germany)
GYRO - Wedge VCON (GA)
GKV - Local VCON (NIFS, Japan)
GYSELA - Global General Geometry VCON (CEA, France)