The δf PIC-MHD model in NIMROD

Charlson C. Kim

PSI Center
U. Washington

NIMROD Team Meeting
Madison, WI
July 29, 2009

Outline

- hybrid kinetic-MHD model
- PIC and δf and the δf PIC-MHD model
- the details
  - drift kinetic equation and slowing down distribution
  - the implementation - `some code`
    - ...
- example to illustrate diagnostics
- status and future directions

Hybrid Kinetic-MHD Model

The Hybrid Kinetic-MHD Equations[a]

- in the limit $n_h \ll n_0$, $\beta_h \sim \beta_0$, quasi neutrality, only modification of MHD equations is addition of the hot particle pressure tensor in the momentum equation:

$$\rho\left(\frac{\partial \mathbf{U}}{\partial t} + \mathbf{U}\cdot\nabla\mathbf{U}\right) = \mathbf{J}\times\mathbf{B} - \nabla p_b - \nabla\cdot\underline{\mathbf{p}}_h$$

  the subscripts b, h denote the bulk plasma and hot particles

  - ρ, $\mathbf{U}$ for the entire plasma, both bulk and hot particle

- the steady state equation

$$\mathbf{J}_0\times\mathbf{B}_0 = \nabla p_0 = \nabla p_{b0} + \nabla p_{h0}$$

  - p_{b0} is scaled to accomodate hot particles
  - assumes equilibrium hot particle pressure is isotropic

[a] C.Z.Cheng,'A Kinetic MHD Model for Low Frequency Phenomena',*J. Geophys. Rev* **96**, 1991

Linearized Momentum equation and $\delta \underline{\mathbf{p}}_h$

- evolved momentum equation is $(\mathbf{U}_s = 0)$

$$\rho_s \frac{\partial \delta \mathbf{U}}{\partial t} = \mathbf{J}_s \times \delta \mathbf{B} + \delta \mathbf{J} \times \mathbf{B}_s - \nabla \cdot \delta \underline{\mathbf{p}}_b - \nabla \cdot \delta \underline{\mathbf{p}}_h$$

- assume CGL-like form $\delta \underline{\mathbf{p}}_h = \begin{pmatrix} \delta p_\perp & 0 & 0 \\ 0 & \delta p_\perp & 0 \\ 0 & 0 & \delta p_\parallel \end{pmatrix}$

- evaluate pressure moment at a position $\mathbf{x}$ is

$$\begin{aligned} \delta p(\mathbf{x}) &= \int m(v - V_h)^2 \delta f(\mathbf{x}, \mathbf{v}) d^3 v \\ &= \sum_{i=1}^{N} \delta f_i m(v_i - V_h)^2 \delta^3(\mathbf{x} - \mathbf{x}_i) \end{aligned}$$

where sum is over the particles, δf_i is the perturbed phase space volume, m mass of the particle, and V_h is the velocity moment of the particles

PIC and δf and the δf PIC-MHD model

Overview of the PIC method

- in principle, $f(\mathbf{x}(t), \mathbf{v}(t))$ contains everything **but** intractable
- PIC is a Lagrangian simulation of phase space $f(\mathbf{x}, \mathbf{v}, t)$
- PIC approximation

$$f(\mathbf{x}, \mathbf{v}, t) \simeq \sum_{i=1}^{N} g_i(t) S(\mathbf{x} - \mathbf{x}_i(t)) \delta(\mathbf{v} - \mathbf{v}_i(t))$$

  where the sum is over the number of particles N, i denotes the particle index, g_i is the phase space volume represented by the particle

- PIC evolves $f(\mathbf{x}_i(t), \mathbf{v}_i(t))$ by
  - deposit moment of $f(\mathbf{x}_i(t), \mathbf{v}_i(t))$ on grid using $S(\mathbf{x} - \mathbf{x}_i)$
  - field solve using moment
  - advance $[\mathbf{x}_i, \mathbf{v}_i]$ using solved fields
- all dynamics is in particle motion
- typically PIC is noisy, can't beat $1/\sqrt{N}$

The δf PIC method[a]

- δf PIC reduces the discrete particle noise associated with conventional PIC
- Vlasov Equation

$$\frac{\partial f(\mathbf{z},t)}{\partial t} + \dot{\mathbf{z}} \cdot \frac{\partial f(\mathbf{z},t)}{\partial \mathbf{z}} = 0$$

- split phase space distribution into steady state and evolving perturbation

$$f = f_{eq}(\mathbf{z}) + \delta f(\mathbf{z},t)$$

- apply PIC to $\delta f(\mathbf{z},t)$
- particles advance along characteristics $\dot{\mathbf{z}}$
- substitute f in Vlasov Equation to get δf evolution equations

$$\dot{\delta f} = -\dot{\tilde{\mathbf{z}}} \cdot \frac{\partial f_{eq}}{\partial \mathbf{z}}$$

using $\dot{\mathbf{z}} = \dot{\mathbf{z}}_{eq} + \dot{\tilde{\mathbf{z}}}$ and $\dot{\mathbf{z}}_{eq} \cdot \frac{\partial f_{eq}}{\partial \mathbf{z}} = 0$

[a] A. Y. Aydemir, "A unified MC interpretation of particle simulations...", *Physics of Plasmas*, **1**, 1994

The δf PIC-MHD model

- get fields from NIMROD
- advance particles and δf

$$\mathbf{z}_i^{n+1} = \mathbf{z}_i^n + \dot{\mathbf{z}}(\mathbf{z}_i)\Delta t$$
$$\delta f_i^{n+1} = \delta f_i^n + \dot{\delta f}(\mathbf{z}_i)\Delta t$$

- deposit moment

$$\delta p(\mathbf{x}) = \sum_{i=1}^{N} \delta f_i m (v_i - V_h)^2 S(\mathbf{x} - \mathbf{x}_i)$$

- advance NIMROD equations with the hybrid Kinetic-MHD momentum equation

$$\rho_s \frac{\partial \delta \mathbf{U}}{\partial t} = \mathbf{J}_s \times \delta \mathbf{B} + \delta \mathbf{J} \times \mathbf{B}_s - \nabla \cdot \delta \underline{\mathbf{p}}_b - \nabla \cdot \delta \underline{\mathbf{p}}_h$$

The Details

Drift Kinetic Equation of Motion

- the drift kinetic equations of motion are used as the particle characteristics

$$\dot{\mathbf{x}} = v_\parallel \hat{\mathbf{b}} + \frac{m}{eB^4}\left(v_\parallel^2 + \frac{v_\perp^2}{2}\right)\left(\mathbf{B} \times \nabla \frac{B^2}{2}\right) + \frac{\mathbf{E} \times \mathbf{B}}{B^2} + \frac{\mu_0 m v_\parallel^2}{eB^2}\mathbf{J}_\perp$$

$$m\dot{v}_\parallel = -\hat{\mathbf{b}} \cdot (\mu \nabla B - e\mathbf{E})$$

- reduces $6D$ phase space variables to $4D+1$ $\left[\mathbf{x}(t), v_\parallel(t), \mu = \frac{\frac{1}{2}mv_\perp^2}{\|\mathbf{B}\|}\right]$
- follows gyrocenter in the limit of zero Larmour radius

Slowing Down Distribution for Hot Particles

- the slowing down distribution function

$$f_{eq} = \frac{P_0 \exp(\frac{P_\zeta}{\psi_0})}{\varepsilon^{3/2} + \varepsilon_c^{3/2}}$$

$P_\zeta = g\rho_\| - \psi$ is the canonical toroidal momentum and ε is the energy, ψ_0 is the gradient scale length, and ε_c is the critical slowing down energy

- the evolution equation for δf

$$\begin{aligned}
\dot{\delta f} &= f_{eq}\left\{\frac{mg}{e\psi_0 B^3}\left[\left(v_\|^2 + \frac{v_\perp^2}{2}\right)\delta\mathbf{B}\cdot\nabla B - \mu_0 v_\| \mathbf{J}\cdot\mathbf{E}\right]\right. \\
&\quad \left. + \frac{\delta\mathbf{v}\cdot\nabla\psi_p}{\psi_0} + \frac{3}{2}\frac{e\varepsilon^{1/2}}{\varepsilon^{3/2} + \varepsilon_c^{3/2}}\mathbf{v}_D\cdot\mathbf{E}\right\} \\
\mathbf{v}_D &= \frac{m}{eB^3}\left(v_\|^2 + \frac{v_\perp^2}{2}\right)(\mathbf{B}\times\nabla B) + \frac{\mu_0 m v_\|^2}{eB^2}\mathbf{J}_\perp \\
\delta\mathbf{v} &= \frac{\mathbf{E}\times\mathbf{B}}{B^2} + \mathbf{v}_\| \cdot \frac{\delta\mathbf{B}}{B}
\end{aligned}$$

Summary of Equations

- kinetic-MHD momentum equation

$$\rho_s \frac{\partial \delta \mathbf{U}}{\partial t} = \mathbf{J}_s \times \delta \mathbf{B} + \delta \mathbf{J} \times \mathbf{B}_s - \nabla \cdot \delta \underline{\mathbf{p}}_b - \nabla \cdot \delta \underline{\mathbf{p}}_h \quad (1)$$

- pressure moment

$$\begin{pmatrix} \delta p_\perp \\ \delta p_\parallel \end{pmatrix} = \sum_{i=1}^{N} \delta f_i \begin{pmatrix} \mu \|\mathbf{B}\| \\ \frac{1}{2} m v_\parallel^2 \end{pmatrix} S(\mathbf{x} - \mathbf{x}_i) \quad (2)$$

- drift kinetic characteristics

$$\dot{\mathbf{x}} = v_\parallel \hat{\mathbf{b}} + \frac{m}{eB^4} \left(v_\parallel^2 + \frac{v_\perp^2}{2} \right) \left(\mathbf{B} \times \nabla \frac{B^2}{2} \right) + \frac{\mathbf{E} \times \mathbf{B}}{B^2} + \frac{\mu_0 m v_\parallel^2}{eB^2} \mathbf{J}_\perp \quad (3)$$

$$m \dot{v}_\parallel = -\hat{\mathbf{b}} \cdot (\mu \nabla B - e\mathbf{E}) \quad (4)$$

- δf evolution equation

$$\dot{\delta f} = f_{eq}\left\{\frac{mg}{e\psi_0 B^3}\left[\left(v_\parallel^2+\frac{v_\perp^2}{2}\right)\delta\mathbf{B}\cdot\nabla B-\mu_0 v_\parallel \mathbf{J}\cdot\mathbf{E}\right]\right.$$
$$\left.+\frac{\delta\mathbf{v}\cdot\nabla\psi_p}{\psi_0}+\frac{3}{2}\frac{e\varepsilon^{1/2}}{\varepsilon^{3/2}+\varepsilon_c^{3/2}}\mathbf{v}_D\cdot\mathbf{E}\right\} \tag{5}$$

$$\mathbf{v}_D = \frac{m}{eB^3}\left(v_\parallel^2+\frac{v_\perp^2}{2}\right)(\mathbf{B}\times\nabla B)+\frac{\mu_0 m v_\parallel^2}{eB^2}\mathbf{J}_\perp \tag{6}$$

$$\delta\mathbf{v} = \frac{\mathbf{E}\times\mathbf{B}}{B^2}+\mathbf{v}_\parallel\cdot\frac{\delta\mathbf{B}}{B} \tag{7}$$

Implementation Details

- particle data structure
- initialization - importance sampling
  - spatial load
  - velocity load
- PIC in FEM
  - inverse mapping and the search/sort and parallelization
  - particle field evaluation
  - particle deposition

Implementation Details

- particle data structure
- initialization - importance sampling
  - spatial load
  - velocity load
- PIC in FEM
  - inverse mapping and the search/sort and parallelization
  - particle field evaluation
  - particle deposition

Particle Data Structure `part_type_mod.f`

- particle data type

```
TYPE :: particle
  REAL(r8), DIMENSION(0:2) :: R,Z,P,v1,v2,v3,w
  REAL(r8) :: g0(0:1),NS(0:3),GNS(1:2,0:3),eta,xi
  INTEGER(i4) :: ix,iy,id,pass
END TYPE particle
```

- species data type composed of array of types

```
TYPE :: species
  CHARACTER(10) :: part_type,hot_distro,pnorm_field
  LOGICAL :: add_phot,phdump,pdbg,trace
  INTEGER(i4) :: number,ntot,MPI_PART,packsz
  REAL(r8) :: mass,charge,vhmx,ecrit,avu0,avw0,orbits
  REAL(r8) :: P0,R0,nh0,rh0,hwdth,pres0,dmphi,anisop
  REAL(r8) :: eparallel,wwgt(0:2),passwgt(0:1)
  TYPE(particle), DIMENSION(:), POINTER :: part
END TYPE species
```

Particle Type

- $6D$ phase space variables and weight

  `REAL(r8), DIMENSION(0:2) :: R,Z,P,v1,v2,v3,w`

  - for `part_type='drift'`
  - `v1` is $v_{||}$
  - `v2(1:2)` is $(\mu, \|\mathbf{B}\|^{-1})$
- `REAL(r8) ::  g0(0:1),NS(0:3),GNS(1:2,0:3),eta,xi`
  - `g0` phase space volume
  - `NS,GNS` shape function value and derivatives (for linear basis)
  - `eta,xi` logical coordinate
- `INTEGER(i4) ::  ix,iy,id,pass`
  - `ix,iy` logical index
  - `id` unique particle I.D.
  - `pass` passed or trapped status

Species Type

- species descriptors and array of particles

```
TYPE :: species
  CHARACTER(10) :: part_type,hot_distro,pnorm_field
  LOGICAL :: add_phot,phdump,pdbg,trace
  INTEGER(i4) :: number,ntot,MPI_PART,packsz
  REAL(r8) :: mass,charge,vhmx,ecrit,avu0,avw0,orbits
  REAL(r8) :: P0,R0,nh0,rh0,hwdth,pres0,dmphi,anisop
  REAL(r8) :: eparallel,wwgt(0:2),passwgt(0:1)
  TYPE(particle), DIMENSION(:), POINTER :: part
END TYPE species
```

- `CHARACTER(10) :: part_type,hot_distro,pnorm_field`
  - `part_type=[drift, boris]` equations of motion
  - `hot_distro=[monoenergetic_perp, monoenergetic, maxwellian, slowing down]` velocity distribution function
  - `pnorm_field=[pres_eq, uniformp, uniformn, gauss]` spatial distribution

- `LOGICAL :: add_phot,phdump,pdbg,trace`
- `INTEGER(i4) :: number,ntot,MPI_PART,packsz`
  - `number` local particle count
  - `ntot` global particle count
  - `MPI_PART,packsz` mpi parameters
- `REAL(r8) :: mass,charge,vhmx,ecrit,avu0,avw0,orbits`
  - `vhmx` parameter for velocity distribution
  - `ecrit` ε_c slowing down critical energy in KeV
  - `avu0,avw0` equilibrium offset (*unused*)
  - `orbits` number of subcycled particle steps
- `REAL(r8) :: P0,R0,nh0,rh0,hwdth,pres0,dmphi,anisop`
  - `P0` ψ_0 in slowing down distribution `Ppsi0` in nimrod.in
  - `R0` major radius (*unused*)
  - `nh0,rh0,hwdth` $=\mathtt{nh0}\exp\left(-\frac{(r-\mathtt{rh0})^2}{(L*\mathtt{hwdth})^2}\right)$ for `pnorm_field=gauss`
  - `pres0`(*unused*)

- `dmphi` length in periodic direction
- `anisop` multiplier for anisotropic pressure, sometimes set to zero

- `REAL(r8) ::  eparallel,wwgt(0:2),passwgt(0:1)`
  - `eparallel` multiplier for eparallel nonlinearity
  - `wwgt(0:2)` multiplier for different portions of weight equation controlled by `weight_use=[all, velocity, spatial, gradn]`
  - `passwgt(0:1)` multiplier for moment calculation for passed and trapped particles controlled by `phase_use=[all, pass, trap]`

Other parameters in &particle_input

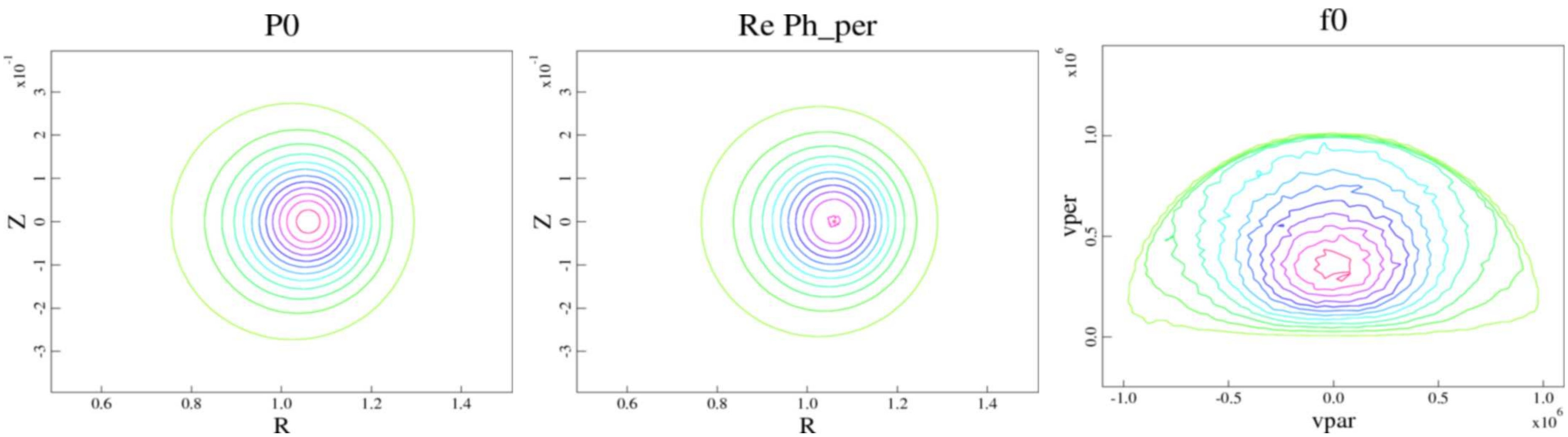
- `nm` total number of requested particles
- `restart` logical to read in `part_dump*.bin` file
- `phqty`=2,6 number of components of pressure tensor
- `betafrac`=[0,1] fraction of pressure carried by hot particles
- `ph_pd` polynomial order for particles and phot
- `tpartdr` directory for `part_dump*.bin` file
- `bmxs,emxs` offset in `mx` to load annulus of particles
- `pdbg` turn on verbose output

Implementation Details

- particle data structure
- initialization - importance sampling
  - spatial load
  - velocity load
- PIC in FEM
  - inverse mapping and the search/sort and parallelization
  - particle field evaluation
  - particle deposition

Initialization - mimic $f_e q(\mathbf{z})$

- particle loading mimics $f_e q(\mathbf{z})$ in space and velocity - importance sampling



- example uses `pnorm_field=pres_eq`, `hot_distro=slowing down`
- phot in dumpfile if `phdump=.true.`
- diagnostic equilibrium phot is in `dump.99998`,`dump.99999`
- volume average $f(v_{\|}, v_{\perp}), \delta f(v_{\|}, v_{\perp})$ from `$>xd phase`

Algorithmic Details

- nonuniform loading in physical and velocity space
- match numeric load with analytic distribution $f_{eq}(\mathbf{x}, \mathbf{v})$
- physical space loaded proportional to `pnorm_field`
  - area weighted norm field is calculated on FE grid
  - particles are loaded cell by cell $\propto \frac{\Delta \mathcal{P}_a}{\mathcal{P}_a}$
- velocity space loaded to match `hot_distro`
  - ensure isotropy in $\mathbf{v}$
  - first initialize energy
  - compute uniform mapping function

$$D(\varepsilon) = \frac{\int_0^{\varepsilon} f(\varepsilon')d\varepsilon'}{\int_0^{\varepsilon_{max}} f(\varepsilon')d\varepsilon'}$$

  - $\varepsilon = D^{-1}(\texttt{UniformRandomNumber})$
  - then pick direction and project energy to velocity

** $g\rho_{\|}$ not accounted for in slowing down distribution

Implementation Details

- particle data structure
- initialization - importance sampling
  - spatial load
  - velocity load
- PIC in FEM
  - inverse mapping and the search/sort and parallelization
  - particle field evaluation
  - particle deposition

PIC in FEM - nontrivial

- particles pushed in (R, Z) **but** field quantities (including (R, Z)) in (η, ξ)
- FE representation for (R, Z) needs to be inverted

$$R = \sum_j R_j N_j(\eta, \xi), \quad Z = \sum_j Z_j N_j(\eta, \xi),$$

- Use Newton method to solve for (η, ξ) given (R, Z)

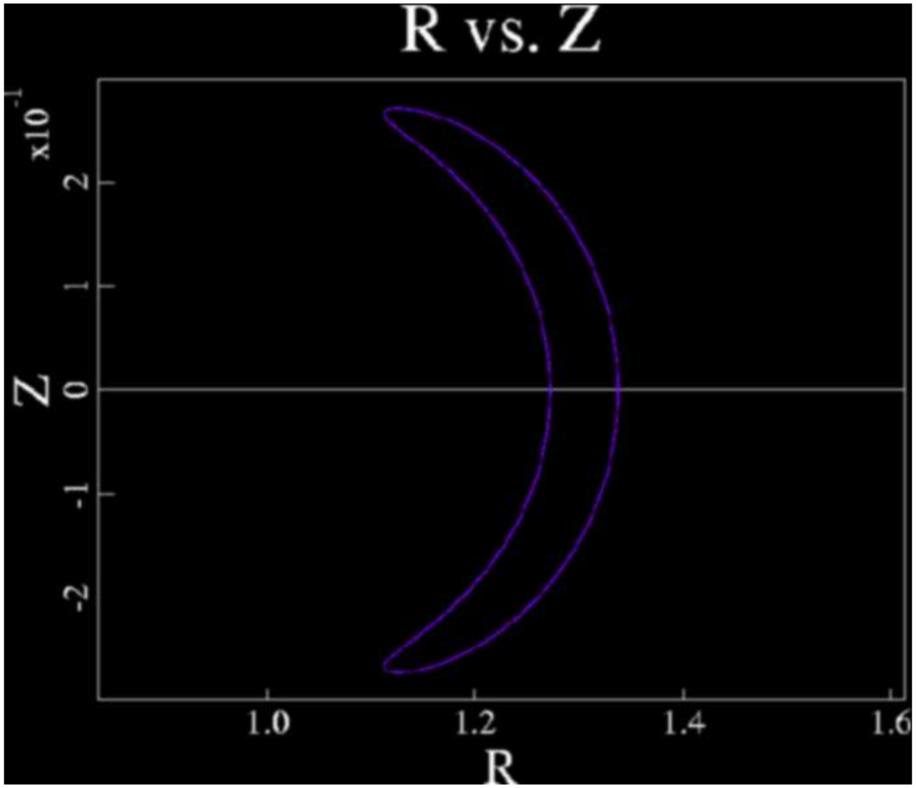
$$\left\{ \begin{array}{c} \eta^{k+1} \\ \xi^{k+1} \end{array} \right\}_j = \left\{ \begin{array}{c} \eta^{k} \\ \xi^{k} \end{array} \right\}_j + \frac{1}{\Delta_j^k} \left(\begin{array}{cc} \frac{\partial Z}{\partial \xi} & -\frac{\partial R}{\partial \xi} \\ -\frac{\partial Z}{\partial \eta} & \frac{\partial R}{\partial \eta} \end{array} \right)_j^k \left\{ \begin{array}{c} R - R^k \\ Z - Z^k \end{array} \right\}_j$$

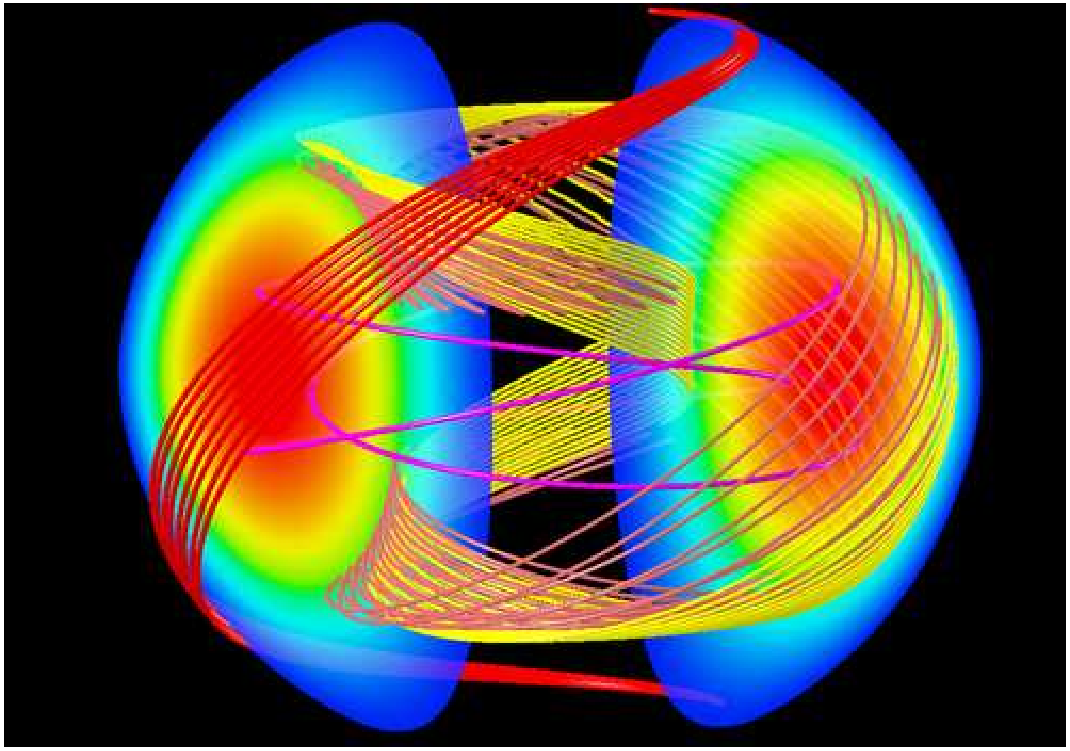
where Δ_j^k is the determinant and k denotes iteration, j denotes FE

- iterative process in both k and j
- sorting and parallelization done at the same time
- slowest part!

Examples

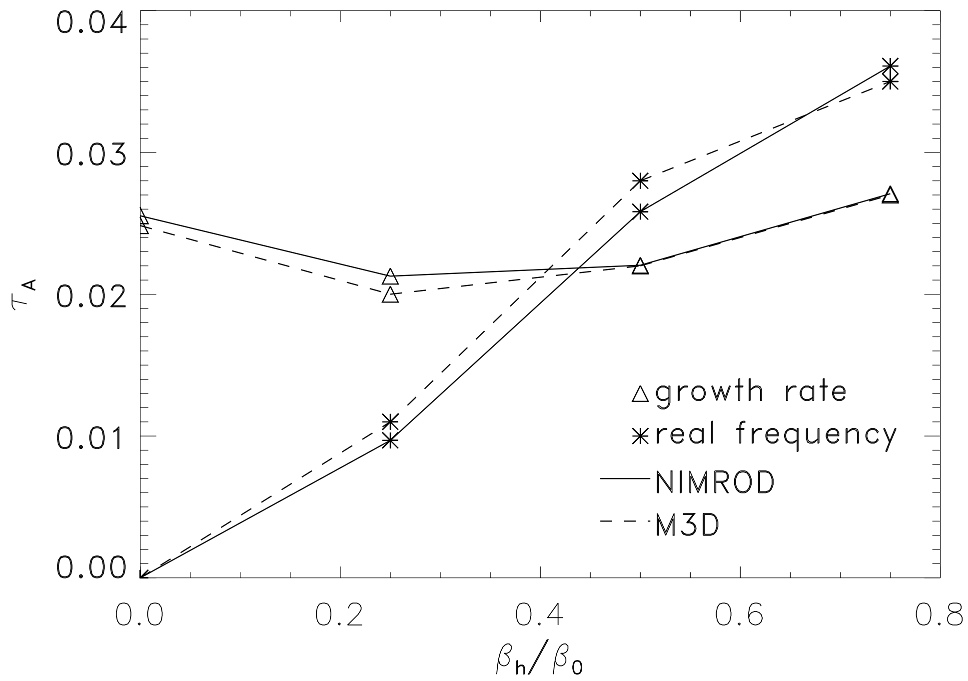
Particle Traces





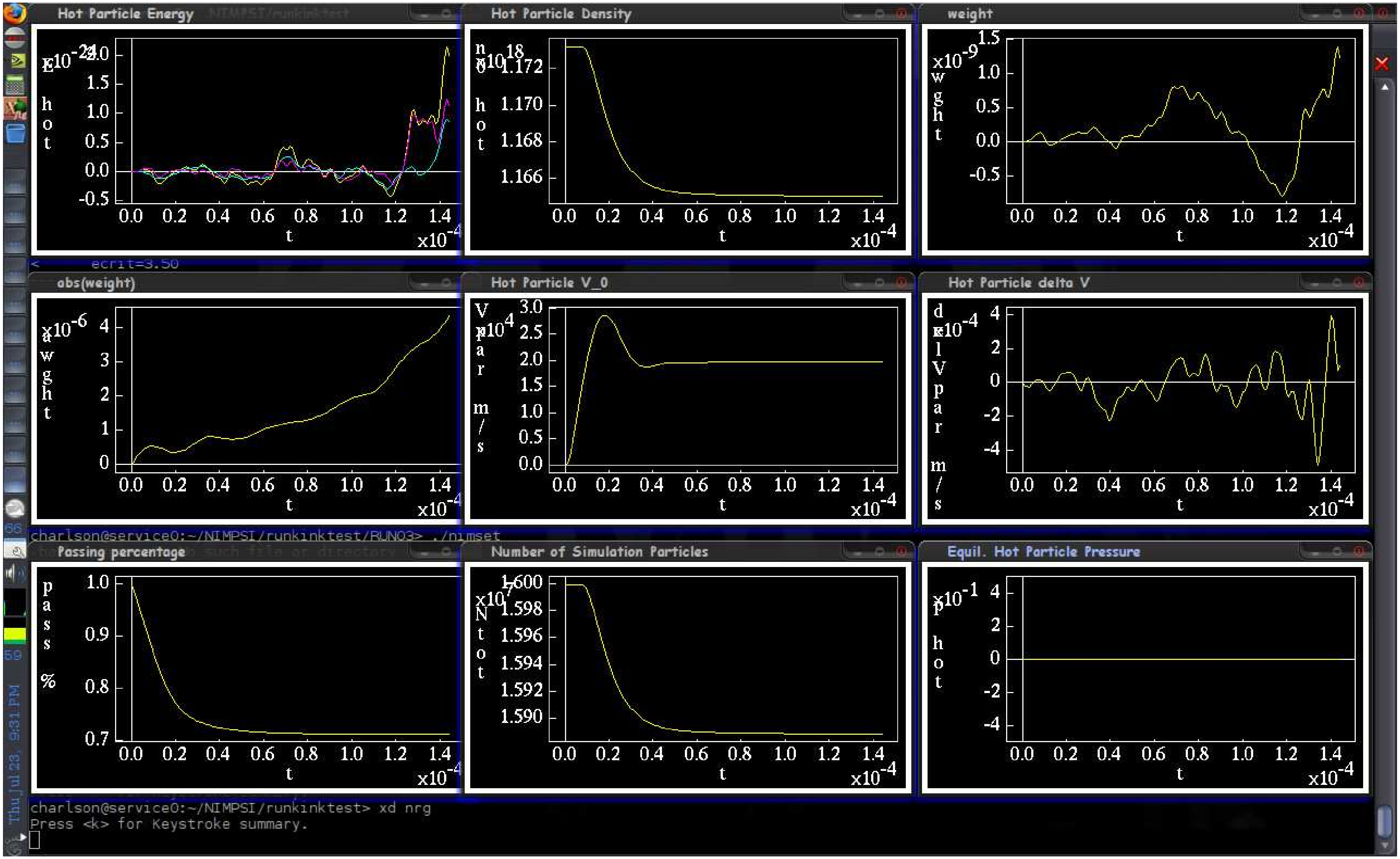
$(1, 1)$ **kink benchmark**

- monotonic q, $q_0 = .6$, $q_a = 2.5$, $\beta_0 = .08$, shifted circular tokamak R/a=2.76
- dt=1e-7, $\tau_A = 1.e6$



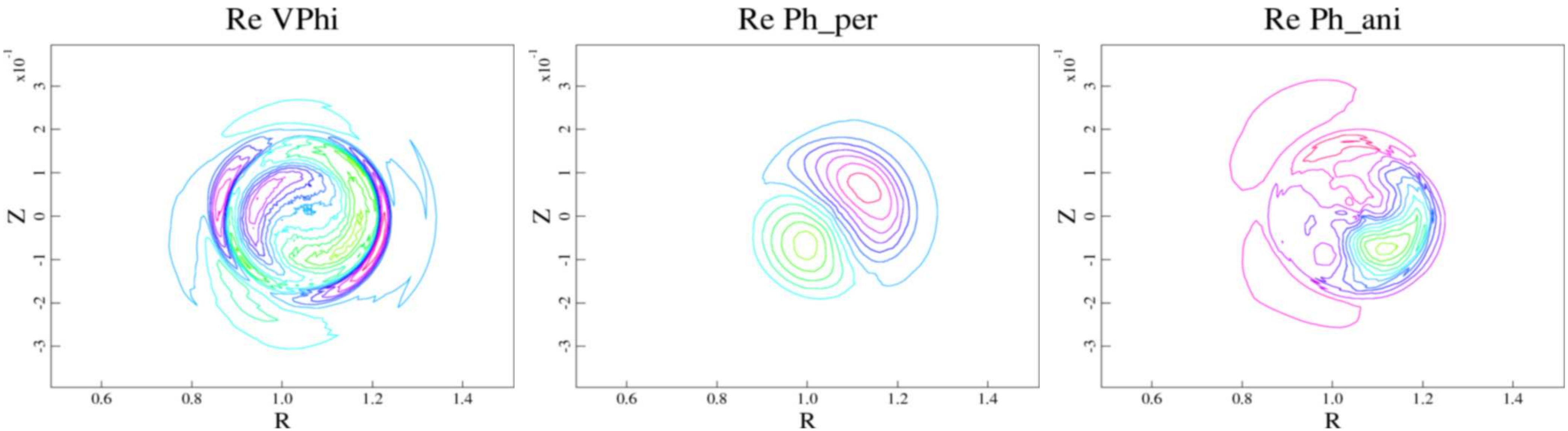
Diagnostics indicate trajectories are resolved

$>xd nrg



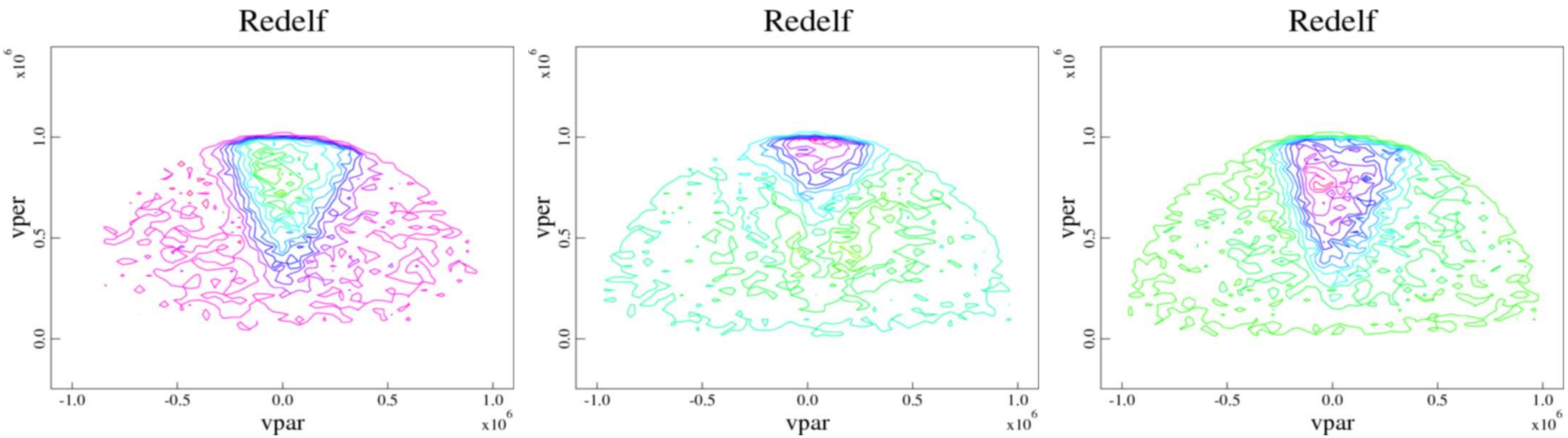
Contours suggest sufficient particle number

```
$>xd conph
```



$\delta f(v_{\|}, v_{\perp})$ shows activity concentrated in trapped cone

```
$>xd phase
```



at $t = [4.0, 4.5, 5.0] \times 10^{-4} s$

Status of Drift Kinetic Particles

- drift kinetic particles are moderately robust in linear NIMROD
- general parallelization in rblocks
- run on ~ 100 procs with $\sim 10M$'s particles
- up to $\varepsilon_{max} \simeq 250KeV$
- minor bug in restart
- more diagnostics in the works

Future Directions

- better field gather routine $\mathbf{E}, \mathbf{B}$
- domain decomposition with `nlayer`
- high order finite elements
- improved deposition
- improved parallelization - domain cloning, asynchronous
- utilize sorted particle list
- nonlinear - center equations
- try type of arrays data structure
- full f option
- TAE's and EPM's

Deposition of $\delta \underline{\mathbf{p}}_h$ onto Finite Element grid

- recall pressure moment at a position $\mathbf{x}$ is

$$\delta p(\mathbf{x}) = \int m(v - V_h)^2 \delta f(\mathbf{x}, \mathbf{v}) d^3 v$$

$$\delta p_n(R, Z) = \sum_{i=1}^{N} m(v_i - V_h)^2 g_0 w_i \delta(R - R_i)\delta(Z - Z_i) e^{-in\phi_i}$$

where sum is over the particles, m mass of the particle, $g_0 w_i$ is the perturbed phase density, and V_h is the hot flow velocity, deposition is done in Fourier space[a]

- express lhs $\delta p_n(R, Z)$ in the finite element basis,

$$\delta p_n(R, Z) = \sum_k \delta p_n^k N^k(\eta, \xi)$$

where sum k is over the basis functions and (η, ξ) are functions of (R, Z)

[a] for large n runs, a conventional ϕ deposition with a **FFT** is performed

- project onto the finite element space by casting in weak form:

$$\int N^l \sum_k \delta p_n^k N^k d^3x = \int N^l \sum_{i=1}^{N} m(v_i - V_h)^2 g_0 w_i \delta(R - R_i)\delta(Z - Z_i) e^{-in\phi_i} d^3x$$

$$\underline{\mathbf{M}}\delta p_n^k = \sum_{l \in k} \sum_{i=1}^{N} m(v_i - V_h)^2 g_0 w_i e^{-in\phi_i} N^l(\eta_i, \xi_i)$$

$\underline{\mathbf{M}}$ is the finite element mass matrix

- Petrov-Galerkin method

- gather is done using the same shape functions

$$A(\mathbf{x}_i) = \sum_l A^l N^l(\eta_i, \xi_i)$$

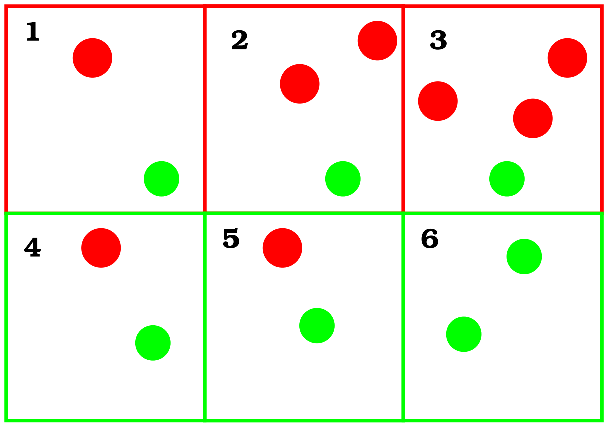
for some field quantity A

- should use shape function $S(R - R_i)$ but requires deposition in real space
  - need ghost cells or global grid with `MPI_ALLGATHER`
  - consistent with gather???

Particle Sorting

- Sorting is important because:
  - essential to domain decomposition
  - cache thrashing is minimized
- use "bucket" sort
- particles are sorted on global FE grid
- particles outside of the processor sub-domain (rblock) are passed to their appropriate processor

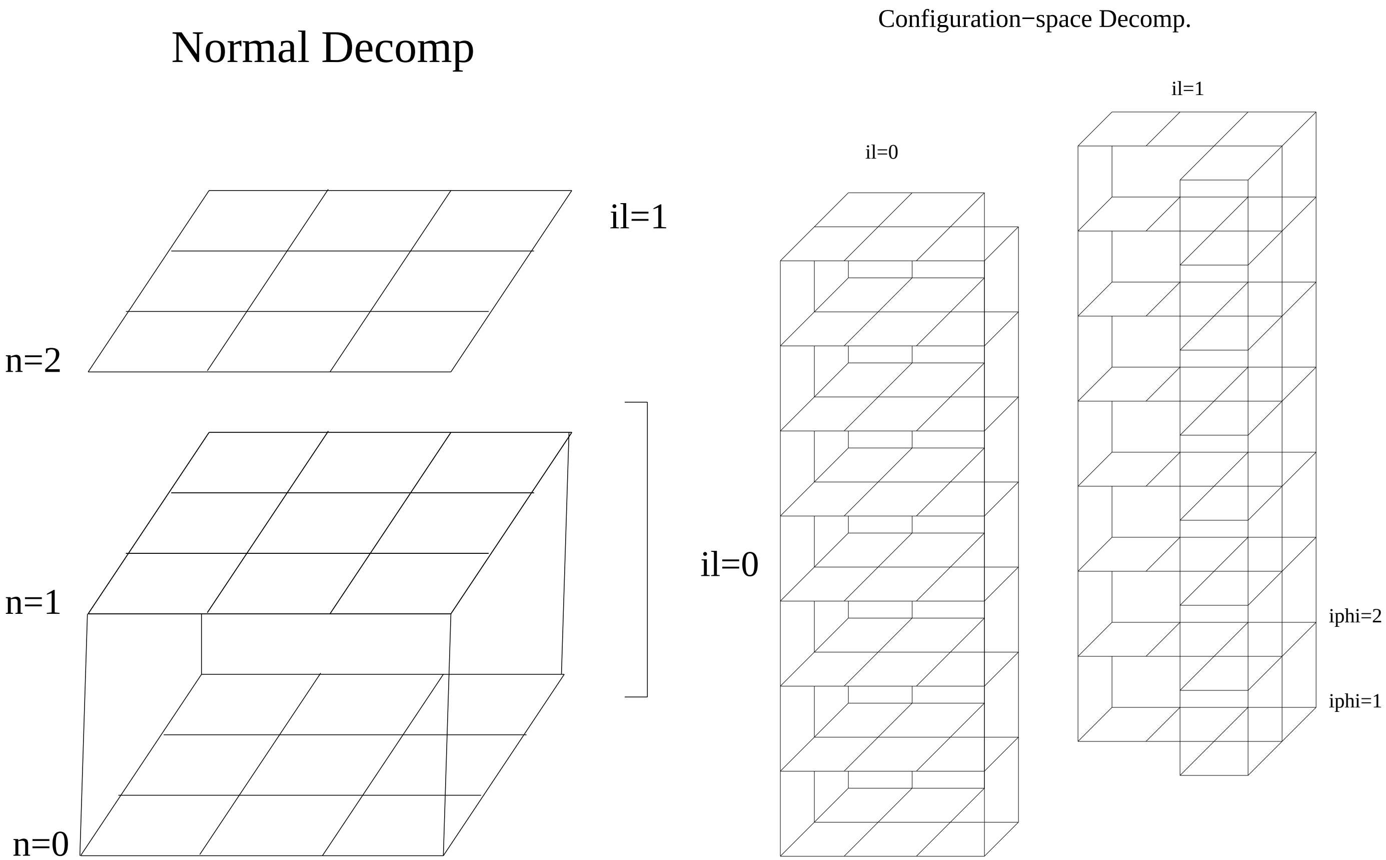
Bucket Sort



cell number	1	2	3	4	5	6
red count	1	2	3	1	1	0
green count	1	1	1	1	1	2
total count	2	3	4	2	2	2
local displacement	2	5	9	2	4	6

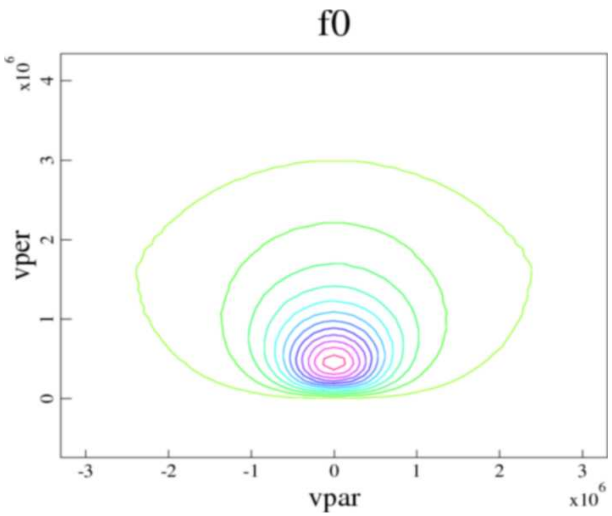
- number of particles in each element is tabulated (red count, green count)
- **total count** gives the number of particles in each element
- displacement array is calculated from the **total count** - bucket size is determined
- pass particles outside of subdomain (red particle to green cells and green particles to red cells)
- using the displacement array, the sorted particle list is filled

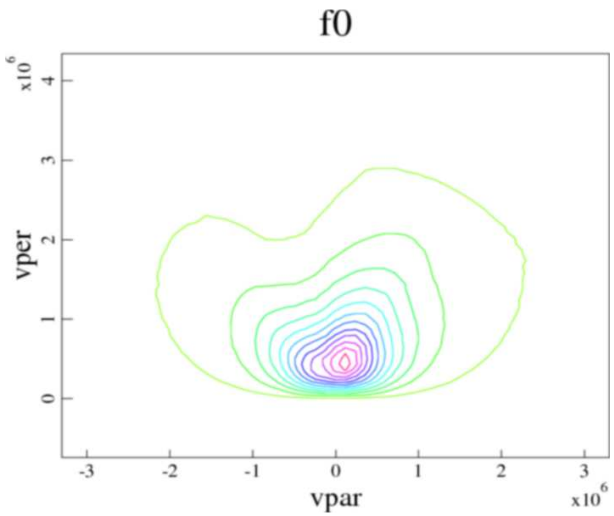
Domain Decomposition with `nlayer`> 1

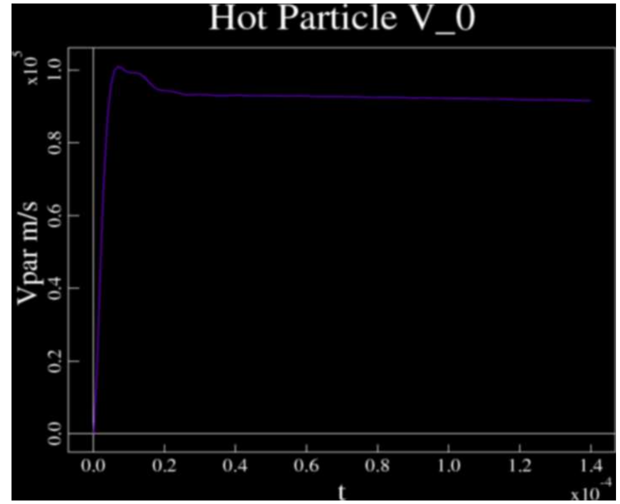


- should not be too bad
- works with NIMROD's domain decomposition
- forces deposition to be done in real space
- particles will reside in nlayer subdomain of rblocks

Effect of ignoring $g\rho_{\parallel}$







- tricky to take into account due to mixing of $\mathbf{x}$ and $\mathbf{v}$
  - inclusion results in anisotropy
  - equilibrium balance is no longer with scalar pressure
- mostly a sampling issue
- maybe a consistency issue
- $V_{hot} \neq 0$! in few transit times
- $f0$ NOT isotropic

TAE's

- D. Spong for ITPA Meeting
- toroidicity induce Alfven eigenmodes (TAE) are marginally stable global modes that exist in the toroidicity induced gap of the Alfven continuum
- energetic particles can destabilize
- have equilibrium specifications
  - shifted circular equilibrium
  - zero β, aspect ratio .1
  - $q(\rho) \simeq 1.1 + 0.8\rho^{2.4}$
  - energetic particles are maxwellian $T_h \simeq 10\text{KeV}$
- may impact heating methods and confinement of α's
- TAE's also of interest to RFP's
- possible evidence of TAE's in extensions of benchmark case