



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

ESL Startup Workshop

R.H. Cohen

December 5, 2005

ESL Startup Workshop
Berkeley, CA, United States
November 30, 2005 through November 30, 2005

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

ESL STATUS AND MEETING OBJECTIVES

**R. Cohen Presentation to
ESL Workshop
Nov. 30, 2005**

STATUS AND PROJECT GOALS

- ESL = Edge Simulation Laboratory
- Goal: develop continuum gyrokinetic edge code for ...
 - Discuss near-term application focus this afternoon
- Status: OFES base-program project, outgrowth of edge FSP proposal
 - OFES and OASCR recognized the quality of the proposal but viewed it as more appropriate for base program support.
- Funding
 - OFES has committed \$550K for FY06.
 - OASCR funding is “pending” -- primarily, the size of the budget reserve after taxes for hurricanes, etc.
 - Possibilities for future-year growth
- LLNL LDRD project (in 3rd and final year):
 - Funding level also uncertain, due to ongoing overall review of Lab’s LDRD portfolio and strategy

Workshop goals

- Original plan: 2-day workshop to review status of contributing efforts and initiate activities for coming year
 - Was predicated on clear budget situation
- Goals given current situation: Discuss elements that will be part of project “no matter what” ...
 - Review status of LLNL LDRD code
 - Hear input/reactions from collaborators
 - Begin discussion next steps, options
- Deferred to a teleconference following more complete picture of funding:
 - Introduction to CHOMBO
 - Design requirements
 - Team formation
 - Setting of more detailed project goals
 - Process issues

ORGANIZATION OF TODAY'S MEETING

- 9:45 - 10:15 Welcome; Workshop objectives and status of ESL
- 10:15 - 12:30, Status of LLNL LDRD code (TEMPEST):
 - Physics content
 - Numerics (algorithms)
- 12:30 - 1:30, Lunch
- 1:30 - 2:30, Status/Plans for LLNL LDRD code (cont):
 - Software
 - V&V
 - Planned approach to 5D
- 2:30 - 3:30 Comments from ESL collaborators
- 3:30-3:45 Coffee break
- 3:45 - 4:15 Comments from ESL collaborators
- 4:15 - 5:30 Moving forward: suggested options at various funding levels; open discussion

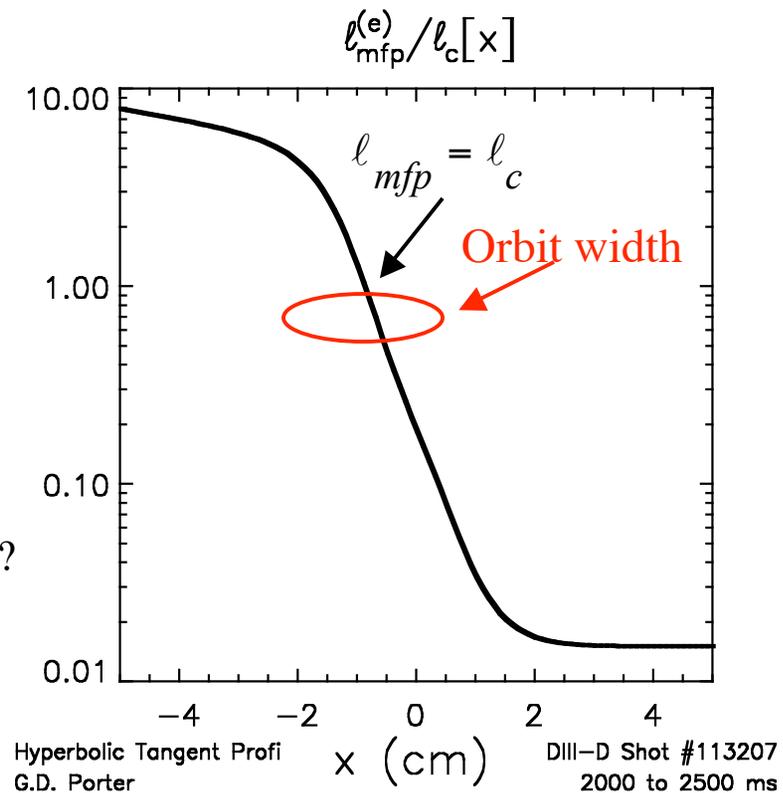
Status of Tempest

W.M. Nevins

Tempest: Code Description

- kinetic code
 - Mean-free-path \sim connection length
- “full” f (not \bar{f})
 - neo-classical perturbation $\sim O(1)$
- Continuum integration scheme
 - Control of discrete particle noise \square
more than 10^3 particles/cell
- Field solve:
 - Electrostatic (walk before you run)
 - Couple to fluid species on same grid?
 - Impurities
(important to power balance)
 - Electrons?
(to solve time-scale problems ...)

DIII-D Edge Barrier



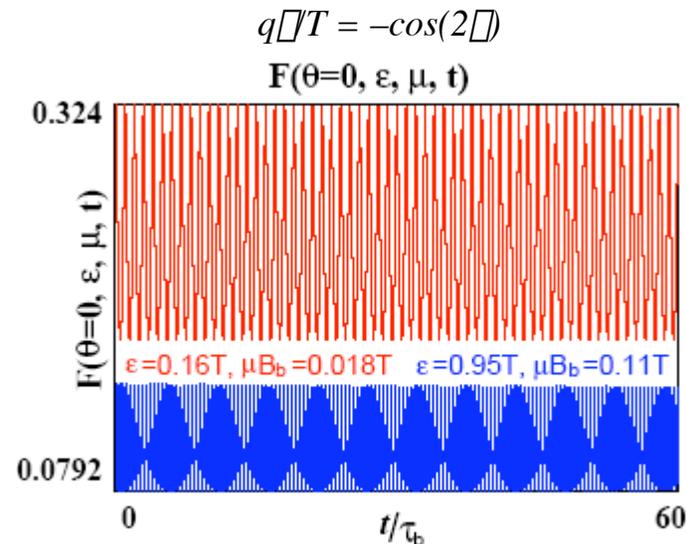
Tempest Coordinate System

Issue: Achieving high accuracy when ϵ varies by order 1 along \mathbf{B} -field

Our solution:

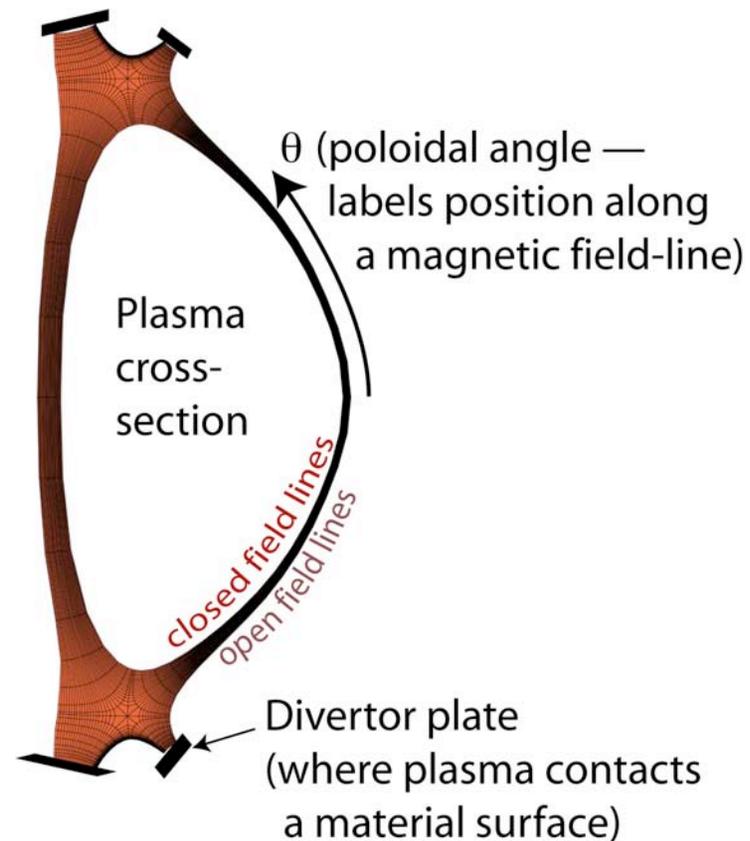
- Choose co-ordinate system to be (nearly) aligned to phase-space flow (a lesson from GYRO ... but modified to allow for potential variation)
 - Energy, $\epsilon_0 = 1/2 mv^2 + q\phi_0$
 - Magnetic moment, $\mu = (1/2 mv_{\perp}^2)/B$
- Use poloidal angle, θ , to measure position along \mathbf{B}
 - Unlike GYRO, θ is the same for each (ϵ, μ)
 - This allows a collision operator to be local in θ

□ Collisionless integration of over 100's transits/bounces with almost no distortion in waveform



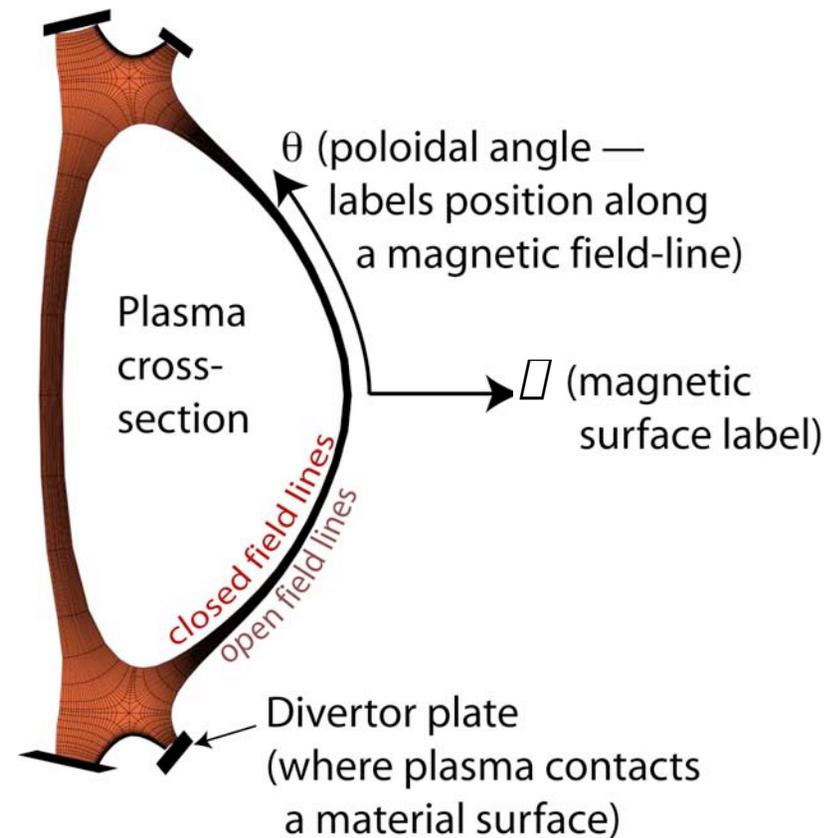
First Milestone: 3-D code \square (\square , \square , \square)

- 2 velocities
 - Energy, $\square = 1/2 mv^2 + q\square$
 - Magnetic moment, $\square = (1/2 mv_{\perp}^2)/B$
- One space
 - Position along B (poloidal angle, \square)
- Code development issues
 - Parallel advection (streaming)
 - Collisions (linear and non-linear)
 - Integrating physics algorithms into evolving code framework
- Physics which can be addressed:
 - flux-limited heat transport, q_{\parallel}
 - Pastukhov end-losses
- \square Milestone largely met
(still fussing with collision operator)



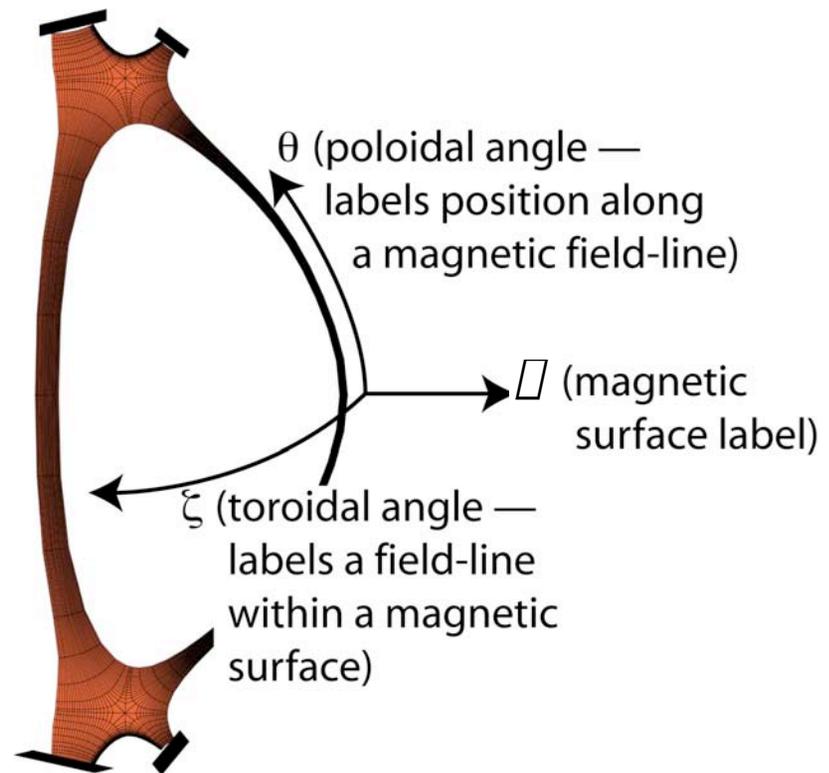
Second Milestone: A Kinetic Analogue to UEDGE

- Edge gyrokinetic code in 4-D
 - Two velocity coordinates, $(v_{\parallel}, v_{\perp})$
 - Two spatial coordinates, (ψ, θ)
- Essential issues will be addressed
 - $L_p \sim \lambda_D$ (orbit width)
 - Data structures to accommodate magnetic geometry with separatrix
 - Interface to the impurity model
- Delivers interesting physics
 - “Orbit Squeezing” and orbit loss
 - Neo-classical transport with $\lambda_p/L_{plasma} \sim O(1)$
- Issues remain w.r.t. with boundary conditions, conservation of f



Sept. '06 Milestone: A Kinetic Analogue to BOUT

- Straight-forward extension from 3- & 4-D codes (one hopes ...)
- Edge gyrokinetic code in 5-D
 - Two velocity coordinates, $(v_{\parallel}, v_{\perp})$
 - Three spatial coordinates, (ρ, θ, ζ)
- Includes physics important to edge turbulence
 - Ion orbit loss (for radial ρ -well)
 - Trapped and passing particles
 - Radial transition from collisional to collisionless
- Models turbulent transport (and its suppression)

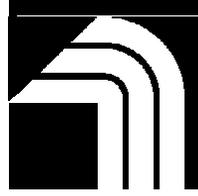


Ion Gyrokinetic Equation

*X. Q. Xu, Z. Xiong, B. I. Cohen, and R. H. Cohen,
M. R. Dorr, J. A. Hittinger, G. D. Kerbel, W. M. Nevins, and H. Qin[†]*

*Lawrence Livermore National Laboratory
University of California, Livermore, CA 94550*

*[†]Princeton Plasma Physics Laboratory
Princeton University, Princeton, New Jersey 08543, USA*



Presented at

ESL kickoff Meeting
November 30, 2005; Berkeley, CA

Acknowledgment to

Drs. Alain J. Brizard, J. Candy, P. Collela, L. Chen, A. Dimits, W. D. Dorland,
T. S. Hahm, T. D. Rognlien, P. Snyder, M. V. Umansky, S. J. Wang, and W. X. Wang

*Performed for US DOE by LLNL under Contract W-7405-ENG-48 and is supported as LLNL LDRD project 04-SI-003.

Electrostatic Ion Gyrokinetic Equation



Ion gyro-kinetic equations for the time-dependent five-dimensional (5D) distribution functions are simplified from H. Qiu and et. al. (submitted to Contrib. Plasma Phys.; T. S. Hahm, Phys. Plasmas, Vol. 3, 4658 (1996)). The gyrocenter distribution function $F_\alpha(\mathbf{x}, \mu, \bar{E}_0, t)$ in gyrocenter coordinates: $\mathbf{x} = \mathbf{x} - \rho, \rho = \mathbf{b} \times \mathbf{v}/\Omega_{c\alpha}$,

$$\frac{\partial F_\alpha}{\partial t} + \mathbf{v}_\perp \cdot \frac{\partial \mathbf{x}_\perp}{\partial \mathbf{x}_\perp} + (\bar{v}_{\parallel\alpha} + v_{B\alpha\text{nos}}) \mathbf{b} \cdot \frac{\partial \mathbf{x}}{\partial F_\alpha} + \left[q \frac{\partial \langle \Phi_0 \rangle}{\partial B} + \mu \frac{\partial t}{\partial B} - \frac{B}{B^*} v_{\parallel} q \frac{\partial \langle \delta \phi \rangle}{\partial s} - \mathbf{v}_0^{\perp} \cdot (q \bar{\Delta} \langle \delta \phi \rangle) \right] \frac{\partial F_\alpha}{\partial E_0} = C(F_\alpha, F_\alpha), \quad (1)$$

$$\mathbf{v}_\perp = \frac{c\mathbf{b}}{B} \times \frac{qB^*_{\parallel}}{M^{\alpha c}} \langle q \bar{\Delta} \langle \Phi \rangle + \mu \bar{\Delta} B \rangle + v_{\parallel}^{\parallel} \frac{qB^*_{\parallel}}{M^{\alpha c}} (\bar{\Delta} \times \mathbf{b}). \quad (2)$$

$$\mathbf{v}_0^{\perp} = \frac{c\mathbf{b}}{B} \times \frac{qB^*_{\parallel}}{M^{\alpha c}} \langle q \bar{\Delta} \langle \Phi_0 \rangle + \mu \bar{\Delta} B \rangle + v_{\parallel}^{\parallel} \frac{qB^*_{\parallel}}{M^{\alpha c}} (\bar{\Delta} \times \mathbf{b}). \quad (3)$$

$$\bar{v}_{\parallel} = \pm \sqrt{\frac{M^{\alpha}}{2} (E_0 - \mu B - q \langle \Phi_0 \rangle)}, \quad (4)$$

$$v_{B\alpha\text{nos}} = \frac{q}{\mu c} (\mathbf{b} \cdot \bar{\Delta} \times \mathbf{b}), \quad (5)$$

$$B^*_{\parallel\alpha} \equiv B \left[1 + \frac{\mathbf{b}}{B} \cdot \frac{\Omega_{c\alpha}}{v_{\parallel}} \bar{\Delta} \times \mathbf{b} \right], \Omega_{c\alpha} = \frac{qB}{M^{\alpha c}}, \mu = \frac{M^{\alpha} v_{\perp}^2}{2B}, \quad (6)$$

Here $Z^{\alpha e}, M^{\alpha}$ are the electric charge and mass of electrons ($\alpha = e$), ions ($\alpha = i$). μ is the guiding center magnetic moment. The left-hand side of Eq. (1) describes the particle motion in the electric field and magnetic field. C_α is the Coulomb collision operator. The over-bar is used for the gyrocenter variables and $\langle \rangle$ denotes the gyroangle averaging.

• The field Φ is split into two parts: Φ is the large amplitude and the slow variation part; $\delta\phi$ is the small amplitude and the rapid variation part. E_0 is almost energy, a coordinate aligned with the direction of propagation.

• The $\mathbf{E}_0 \times B$ flow terms due to the large amplitude and the slow variation Φ_0 will be added.



I. Radial boundary conditions

The radial Robin boundary condition at the inner core surface $\psi = \psi_c$ and the outer wall surface $\psi = \psi_w$ is

$$C_{br} F_{b\alpha} |_{\psi_c, \psi_w} + (1 - C_{br}) \frac{\partial F_{b\alpha}}{\partial r} |_{\psi_c, \psi_w} = \left[C_{br} - (1 - C_{br}) \frac{D_{b\alpha} n_{b\alpha}}{\Gamma_{b\alpha}} \right] F_{mb\alpha}. \quad (7)$$

$$F_{mb\alpha} = \frac{n_{b\alpha}}{n_{b\alpha}^0} \exp \left[-\frac{\mu B}{T_{b\alpha}} - \frac{(v_{\parallel} - u_{b\alpha})^2}{v_{thb\alpha}^2} \right], \quad (8)$$

$$\frac{B}{u_{b\alpha}} = \frac{u_{b\alpha} \bar{x}(\psi)}{u_{b\alpha}^0 \bar{x}(\psi)} - \frac{B_p}{Z^{\alpha} e B^2} \left(\frac{\partial \ln F_{b\alpha}}{\partial \ln F_{b\alpha}} \frac{\partial \psi}{\partial \psi} + \frac{T_{b\alpha}}{Z^{\alpha} e \Phi} \frac{\partial \psi}{\partial \psi} \right), \quad (9)$$

$$\mathcal{F}_{0,\alpha} = \text{erf} \left(\frac{v_{\parallel, \alpha}^{max} - u_{b\alpha}}{v_{thb\alpha}} \right) \left[1 - \exp \left(-\frac{\Gamma_{b\alpha}^{\alpha}(x)}{-\mu_{max} B} \right) \right]. \quad (10)$$

where we assume the drifting Maxwellian distribution in velocity space at the radial boundary surfaces, and $v_{\parallel, \alpha}^{max} = \sqrt{2} \langle E_{max}^0 - Z^{\alpha} e \langle \Phi \rangle \rangle / M_{\alpha}$. E_{max}^0 and μ_{max} are the maximum energy E_0 and magnetic moment μ in the velocity space mesh. This is a generalization of the Dirichlet ($C_{br} = 1$) and Neumann ($C_{br} = 0$) boundary conditions. Here $n_{b\alpha}$, $T_{b\alpha}$, $u_{b\alpha}$, $\Gamma_{b\alpha}$ and $D_{b\alpha}$ are the density, temperature, parallel drifting velocity, particle flux and diffusion coefficient of the particle species α at boundary surfaces.

- There is no boundary condition for particles convecting out of the simulation domain.

II. Poloidal boundary conditions

The boundary conditions for F_{α} is the sheath boundary conditions in θ in the SOL and the private flux regions at the divertor plates, periodic in θ in “core” (inside of separatrix).

- Sheath boundary conditions for the case of normal intersection of the field lines with the walls



Gyrokinetic Poisson equation

In the long wavelength limit $k_{\perp} \rho_a \ll 1$, the self-consistent electric field is typically computed from the gyrokinetic Poisson equation for the multiple species

$$(11) \quad \left(\sum_a \frac{\rho_a^2}{2\lambda_{D_a}^2} \Delta_{\perp}^2 \ln N_a \right) \cdot \Delta_{\perp} \Phi + \Delta_{\perp}^2 \Phi = -4\pi e \left[\sum_a Z_a N_a(\mathbf{x}, t) - n_e(\mathbf{x}, t) \right] - \sum_a \left[\frac{\rho_a^2}{2\lambda_{D_a}^2} N_a Z_a e \frac{\Delta_{\perp}^2 p_{\perp a}}{1} \right]$$

- This equation is an extension of typical neoclassical electric field calculation to have poloidal variation.

- A Boltzmann electron model will be used for the neoclassical electric field calculation.
- If the first-order Padé approximation to $\Gamma_0 = 1/(1+b)$ for the modified Bessel function is used, then the same field solve will be used in the arbitrary wavelength regime.

- We will have to explore the boundary for the separation of the long wavelength and large amplitude component N_0 of $N_a(x, t)$ from the short wavelength and small amplitude component δN , which leads to the separation of Φ_0 from $\delta\phi$ in Φ .

• Boundary conditions

- Radial boundary conditions

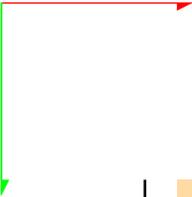
The neoclassical analytical ambipolar value E_r is set at core boundary surface and a given potential is set at wall.

$$(12) \quad E_{\psi}|_{\psi_c} = E_{neo}^{\psi} = -\frac{\partial\Phi}{\partial\psi} = \frac{cRB_t}{B} \langle U_{\alpha\parallel} \rangle - \frac{T_{\alpha}}{Z_{\alpha}e} \left\{ k \frac{\partial\psi}{\partial \ln T_{\alpha}} - \frac{\partial\psi}{\partial \ln P_{\alpha}} \right\}$$

(13)

- Poloidal boundary conditions

The boundary conditions for Φ is the sheath boundary conditions in θ in the SOL and the private flux regions at the divertor plates, periodic in θ in “core” (inside of separatrix).



General Gyrokinetic Equations for Edge Plasmas

Hong Qin

Princeton Plasma Physics Laboratory, Princeton University

Ronald H. Cohen, William M. Nevins, and Xue Qiao Xu

Lawrence Livermore National Laboratory

Supported by U.S. DOE #AC02-76CH03073 and LLNL LDRD #04-SI-03

<http://www.pppl.gov/~hongqin/Gyrokinetics.php>

A general gyrokinetic system developed for edge physics

- ❑ Existing gyrokinetic systems do not apply to edge plasmas.
 - No nonlinear dynamics of the background electromagnetic field.
 - No full FLR effects.

- ❑ A general gyrokinetic system developed for edge physics. New features:
 - Time-dependent background for pedestal dynamics.
 - Nonlinear perturbations for microturbulence and ELMs.
 - Gyro-orbit squeezing effect due to the large E_r shearing.
 - Full FLR effect for short wavelength fluctuations.
 - "Polarization drift density" replaced by a more general expression.

- ❑ Geometric method adopted.
 - Gyrokinetic theory is about gyro-symmetry.
 - Decouple the gyro-phase, not "averaging out".
 - Pullback transformation is indispensable.
 - Coordinate-independent properties automatically satisfied.
 - Energy, momentum, and phase space volume conservation.

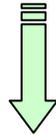
General gyrokinetic equations

$$\frac{dZ_j}{dt} \frac{\partial F}{\partial Z_j} = 0, (0 \leq j \leq 6).$$

$$F = \langle F \rangle + \tilde{F}.$$

Gyrocenter coordinates are good!

$$\frac{\partial}{\partial \theta} \left(\frac{dZ}{dt} \right) = 0,$$



Gyrokinetic equation

$$\frac{\partial \langle F \rangle}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_{\mathbf{x}} \langle F \rangle + \frac{du}{dt} \frac{\partial \langle F \rangle}{\partial u} = 0,$$

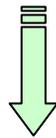
$$\frac{\partial \tilde{F}}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_{\mathbf{x}} \tilde{F} + \frac{du}{dt} \frac{\partial \tilde{F}}{\partial u} + \frac{d\theta}{dt} \frac{\partial \tilde{F}}{\partial \theta} = 0,$$

$\tilde{F} \neq 0$ with collision

Gyrokinetic Maxwell's equations

Pullback

\tilde{F} decouples, set $\tilde{F} = 0$.



$$\nabla^2 \phi = -4\pi \sum_s q_s \int \left[F(\bar{\mathbf{Z}}) + G_1 \cdot \nabla F(\bar{\mathbf{Z}}) + \frac{1}{2} (G_1 \cdot \nabla)^2 F(\bar{\mathbf{Z}}) + G_2 \cdot \nabla F(\bar{\mathbf{Z}}) \right]_{\bar{\mathbf{Z}} \rightarrow g_0(z)} d^3 \mathbf{v}.$$

$$\nabla^2 \mathbf{A} = 4\pi \sum_s q_s \int \left[F(\bar{\mathbf{Z}}) + G_1 \cdot \nabla F(\bar{\mathbf{Z}}) + \frac{1}{2} (G_1 \cdot \nabla)^2 F(\bar{\mathbf{Z}}) + G_2 \cdot \nabla F(\bar{\mathbf{Z}}) \right]_{\bar{\mathbf{Z}} \rightarrow g_0(z)} \mathbf{v} d^3 \mathbf{v},$$

Gyrocenter dynamics

$$i_\tau d\gamma = 0.$$

Curvature drift

$\mathbf{E} \times \mathbf{B}$, ∇B drift

∇D & curvature \mathbf{D} drift

$$\frac{d\mathbf{X}}{dt} = \frac{\mathbf{B}^\dagger}{\mathbf{b} \cdot \mathbf{B}^\dagger} \left(u + \frac{\mu}{2} \mathbf{b} \cdot \nabla \times \mathbf{b} \right) - \frac{\mathbf{b} \times \mathbf{E}^\dagger}{\mathbf{b} \cdot \mathbf{B}^\dagger},$$

$$\frac{du}{dt} = \frac{\mathbf{B}^\dagger \cdot \mathbf{E}^\dagger}{\mathbf{B}^\dagger \cdot \mathbf{b}},$$

$$\frac{d\theta}{dt} = B_0 + \mathbf{R} \cdot \frac{d\mathbf{X}}{dt} - R_0 + \frac{\mathbf{E}_0 \cdot \nabla B_0}{B_0^2} + \frac{u}{2} \mathbf{b} \cdot \nabla \times \mathbf{b}$$

$$+ \frac{\partial}{\partial \mu} \langle \psi_1 + \psi_2 \rangle - \frac{1}{2B_0} [\nabla \cdot \mathbf{E}_0 - \mathbf{b}\mathbf{b} : \nabla \mathbf{E}_0]$$

$$\frac{d\mu}{dt} = 0, \mu \equiv \frac{w^2}{2B_0},$$

drift by spacetime inhomogeneities of \mathbf{E}_0

Banos drift

$$\mathbf{B}^\dagger \equiv \nabla \times (\mathbf{A}_0 + u\mathbf{b} + \mathbf{D}),$$

$$\mathbf{E}^\dagger \equiv \mathbf{E}_0 - \nabla \left[\mu B_0 + \frac{D^2}{2} + \langle \psi_1 + \psi_2 \rangle \right] - u \frac{\partial \mathbf{b}}{\partial t} - \frac{\partial \mathbf{D}}{\partial t}.$$

$$\phi^\dagger \equiv \phi_0 + \mu B_0 + \frac{D^2}{2} + \langle \psi_1 + \psi_2 \rangle,$$

$$\mathbf{A}^\dagger \equiv \mathbf{A}_0 + u\mathbf{b} + \mathbf{D},$$

$$\mathbf{B}^\dagger = \nabla \times \mathbf{A}^\dagger, \mathbf{E}^\dagger = -\nabla \phi^\dagger - \frac{\partial \mathbf{A}^\dagger}{\partial t}.$$

Effective potentials

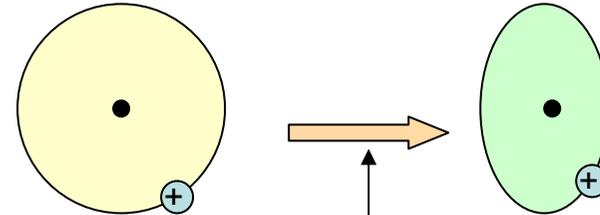
Gyrokinetic Poisson equation

$$\nabla^2 \phi(\mathbf{x}) \equiv -4\pi \sum_s q_s \left[N + N_{\phi_0} + N_{\phi_1} \right],$$

$$S_1 = \frac{w}{B_0^3} \mathbf{E} \cdot (\nabla \mathbf{D} \cdot \mathbf{c}) \times \mathbf{b} - \frac{wu}{B_0^2} \mathbf{b} \cdot \nabla \mathbf{D} \cdot \mathbf{c} + \int \tilde{\phi}_1 d\theta + \frac{w^2}{4B_0^2} \nabla \mathbf{D} \cdot \mathbf{ac}$$

$$N(\mathbf{x}) \equiv \int 2\pi w dw du I_0(\rho \nabla_{\perp}) F(\mathbf{x}, w, u),$$

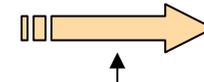
$$N_{\phi_0}(\mathbf{x}) \equiv \frac{1}{B_0^2} (\mathbf{e}_1 \mathbf{e}_1 + \mathbf{e}_2 \mathbf{e}_2) \nabla \left[n(\mathbf{x}) (\mathbf{D} + V_{\parallel}(\mathbf{x}) \mathbf{b}) \cdot \nabla \mathbf{D} \right],$$



Orbit squeezing
by large Er shear

$$N_{\phi_1}(\mathbf{x}) \equiv -\phi_1(\mathbf{x}) \sum_{i=1}^{\infty} \frac{2i}{(i!)^2} \left(\frac{\nabla_{\perp}^2}{\sqrt{2\Omega_0}} \right)^i M_{2i-2}(\mathbf{x}) + \sum_{\substack{i,j=0 \\ i+j \neq 0}}^{\infty} \frac{2(i+j)}{(i!j!)^2} \left(\frac{\nabla_{\perp}^2}{\sqrt{2\Omega_0}} \right)^i \left[M_{2(i+j)-2}(\mathbf{x}) \left(\frac{\nabla_{\perp}^2}{\sqrt{2\Omega_0}} \right)^j \phi_1(\mathbf{x}) \right],$$

Full FLR effect



$$\frac{1}{B_0^2} \nabla_{\perp} \cdot (n \nabla_{\perp} \phi_1) + O(\rho^4 \nabla_{\perp}^4)$$

$$\rho^2 \nabla_{\perp}^2 \ll 1$$

Polarization density

$$n(\mathbf{x}) \equiv \int 2\pi w dw du F(\mathbf{x}, w, u),$$

$$M_i(\mathbf{x}) \equiv \int 2\pi w dw du w^i F(\mathbf{x}, w, u).$$

$$V_{\parallel}(\mathbf{x}) \equiv \frac{1}{n(\mathbf{x})} \int 2\pi w dw du F(\mathbf{x}, w, u),$$

$$I_0(\rho \nabla_{\perp}) \equiv \sum_{i=0}^{\infty} \frac{1}{(i!)^2} \left(\frac{\nabla_{\perp}^2}{\sqrt{2\Omega_0}} \right)^i$$

Future work

- Gyrokinetic Ampere's law with full FLR effects.
 - Expressed in terms of moments of gyrocenter distribution function and differential operators.
- Gyrokinetic collision operator.

Assume FP collision operator is correct for our current purpose.

$$F = \langle F \rangle + \tilde{F},$$

$$\frac{\partial \langle F \rangle}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_{\mathbf{x}} \langle F \rangle + \frac{du}{dt} \frac{\partial \langle F \rangle}{\partial u} = \langle C[g^* \circ \langle F \rangle + g^* \circ \tilde{F}] \rangle,$$

$$\frac{\partial \tilde{F}}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_{\mathbf{x}} \tilde{F} + \frac{du}{dt} \frac{\partial \tilde{F}}{\partial u} + \frac{d\theta}{dt} \frac{\partial \tilde{F}}{\partial \theta} = \overline{C[g^* \circ \langle F \rangle + g^* \circ \tilde{F}]}$$

Treat collision as a perturbation $C \ll 1$:

$O(1)$: \tilde{F} and F are decoupled, take $\tilde{F} = 0$.

$O(C)$:

$$\frac{\partial \langle F \rangle}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_{\mathbf{x}} \langle F \rangle + \frac{du}{dt} \frac{\partial \langle F \rangle}{\partial u} = \langle C[g^* \circ \langle F \rangle] \rangle,$$

$$\frac{\partial \tilde{F}}{\partial t} + \frac{d\mathbf{X}}{dt} \cdot \nabla_{\mathbf{x}} \tilde{F} + \frac{du}{dt} \frac{\partial \tilde{F}}{\partial u} + \frac{d\theta}{dt} \frac{\partial \tilde{F}}{\partial \theta} = \overline{C[g^* \circ \langle F \rangle]} \xrightarrow{???} \frac{d\theta}{dt} \frac{\partial \tilde{F}}{\partial \theta} = \overline{C[g^* \circ \langle F \rangle]}.$$



Preliminary Tests of TEMPEST

Andy Z. Xiong, X.Q. Xu, R. H. Cohen,

Lawrence Livermore National Laboratory

This work was performed for US DOE by LLNL under Contract W-7405-ENG-48 and is supported as LLNL LDRD Project 04-SI-003.



Equations

➤ Normalized ion gyro-kinetic equations in (ψ, θ, E, μ) coordinates.

$$\frac{\partial}{\partial \hat{t}} \hat{F}_\alpha + \hat{v}_{d\psi} \frac{\partial \hat{F}_\alpha}{\partial \hat{\psi}} + \left[\hat{v}_{d\theta} + (\hat{v}_{\parallel\alpha} + \hat{v}_{Banos}) \left(\frac{\hat{B}_p}{\hat{h}\hat{B}} \right) \right] \frac{\partial \hat{F}_\alpha}{\partial \theta} = \frac{\tau_n}{F_n} \sum_{\alpha'} C(F_{\alpha'}, F_\alpha),$$

$$\hat{v}_{\parallel\alpha} = \pm \sqrt{(\hat{E}_0 - \hat{\mu}\hat{B} - Z_\alpha \langle \hat{\Phi}_0 \rangle) / m_\alpha},$$

$$\hat{v}_{Banos} = \frac{\hat{\rho}_s}{2Z_\alpha} \hat{\mu} \hat{\kappa}_{\parallel},$$

$$\hat{v}_{d\psi} = -\frac{\hat{\rho}_s}{2Z_\alpha} \frac{\hat{I}(\hat{\psi})}{\hat{J}\hat{B}\hat{B}_{\parallel}^*} \left[\left(\hat{\mu} + \frac{2m_\alpha \hat{v}_{\parallel}^2}{\hat{B}} \right) \frac{\partial \hat{B}}{\partial \theta} + Z_\alpha \left(\frac{\partial \langle \hat{\Phi} \rangle}{\partial \theta} \right) \right]$$

$$\hat{v}_{d\theta} = \frac{\hat{\rho}_s}{2Z_\alpha} \frac{\hat{I}(\hat{\psi})}{\hat{J}\hat{B}\hat{B}_{\parallel}^*} \left[\left(\hat{\mu} + \frac{2m_\alpha \hat{v}_{\parallel}^2}{\hat{B}} \right) \frac{\partial \hat{B}}{\partial \hat{\psi}} + Z_\alpha \left(\frac{\partial \langle \hat{\Phi} \rangle}{\partial \hat{\psi}} \right) \right]$$

$$\hat{I}(\hat{\psi}) = \hat{R}\hat{B}_t, \quad \hat{J} = (\hat{\nabla}\hat{\psi} \times \hat{\nabla}\theta \cdot \hat{\nabla}\zeta)^{-1} = \frac{\hat{h}}{\hat{B}_p},$$

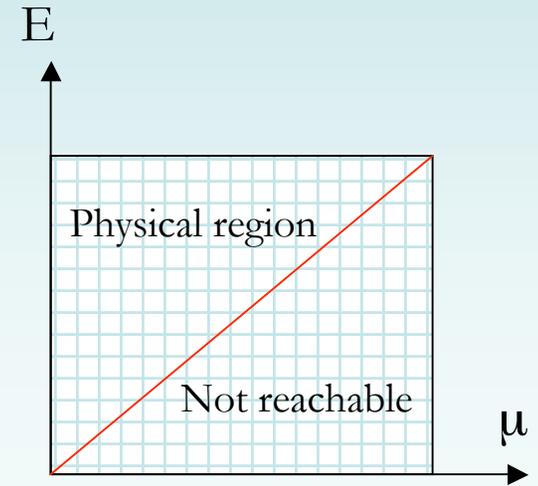
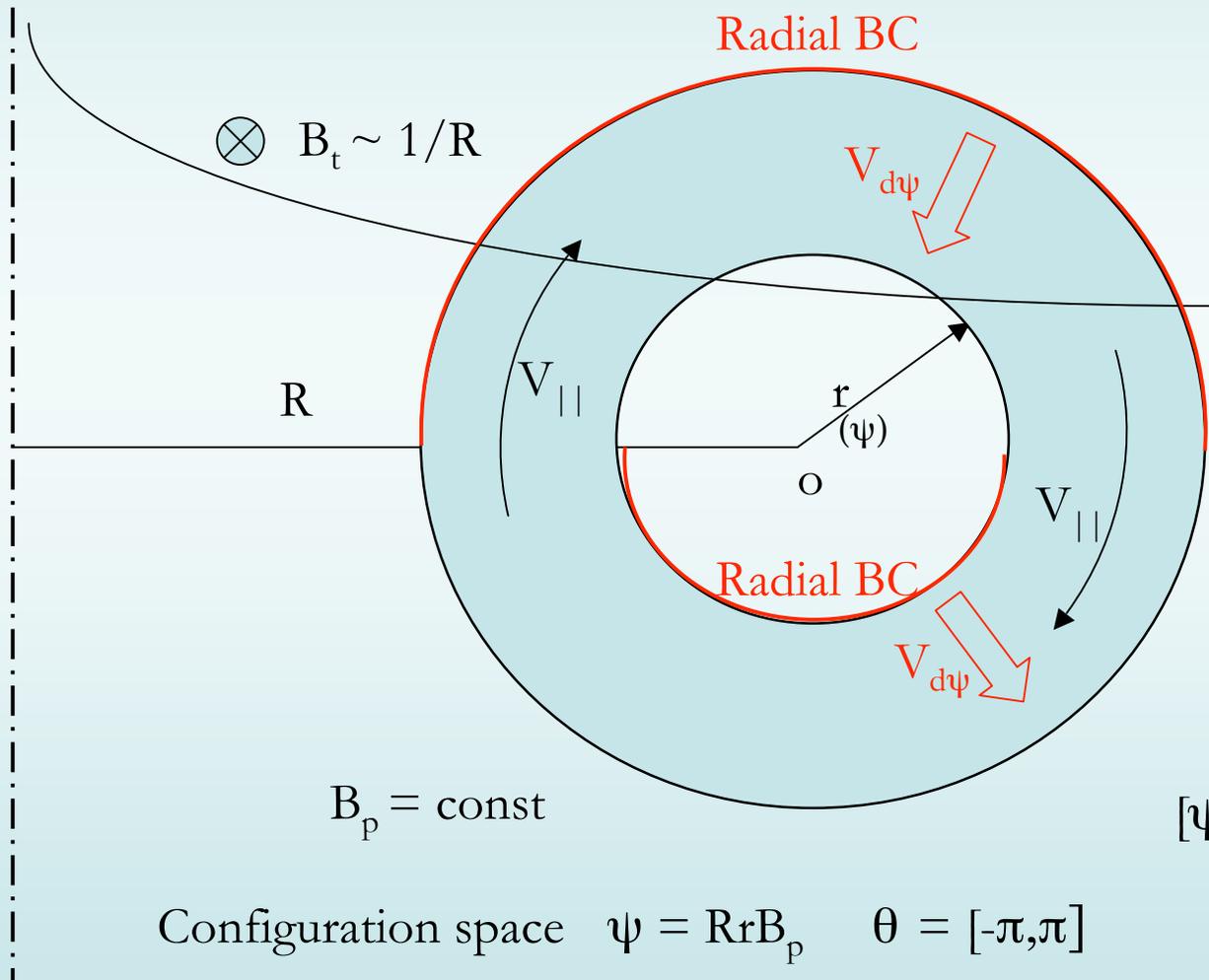
$$\hat{\kappa}_{\parallel} = -\frac{\hat{J}_{11}\hat{J}_{33}}{\hat{B}^2} \frac{\partial \hat{I}(\hat{\psi})}{\partial \hat{\psi}} + \frac{\hat{I}}{\hat{J}\hat{B}^2} \left\{ \frac{\partial}{\partial \hat{\psi}} (\hat{J}\hat{J}_{11}\hat{J}_{33}) + \frac{\partial}{\partial \theta} [\hat{J}\hat{J}_{12}\hat{J}_{22}] \right\}.$$

In the tests so far, $\hat{v}_{d\theta} = \hat{v}_{Banos} = C(F_{\alpha'}, F_\alpha) = 0$



Phase Space

Toroidal annulus of large aspect ratio.



Velocity space

Typical size of the grid

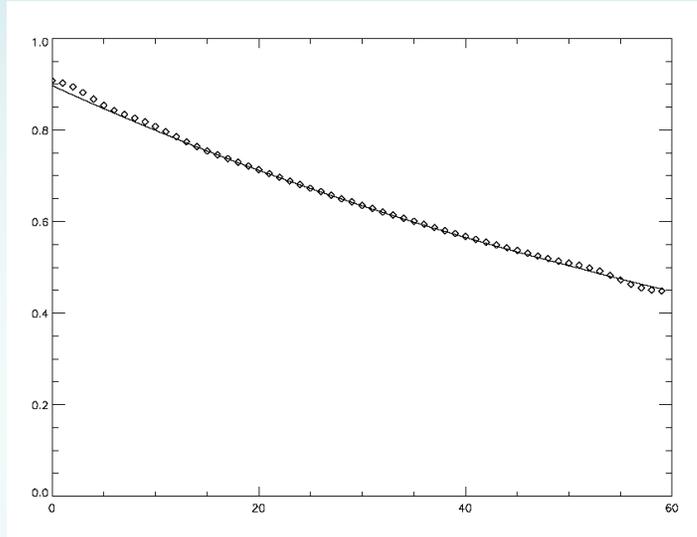
$$[\psi, \theta, E, \mu] : [50, 60, 20, 15]$$



Flow Velocity

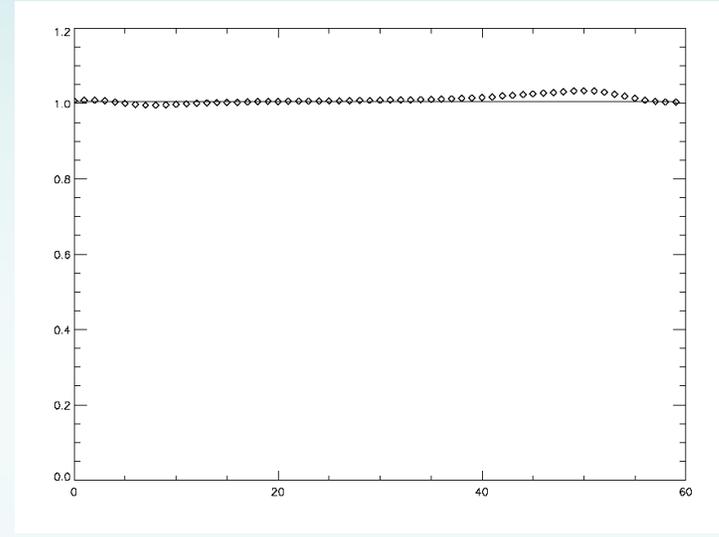
Generation of parallel flow velocity in the absence of collision and temperature gradient.

Ni



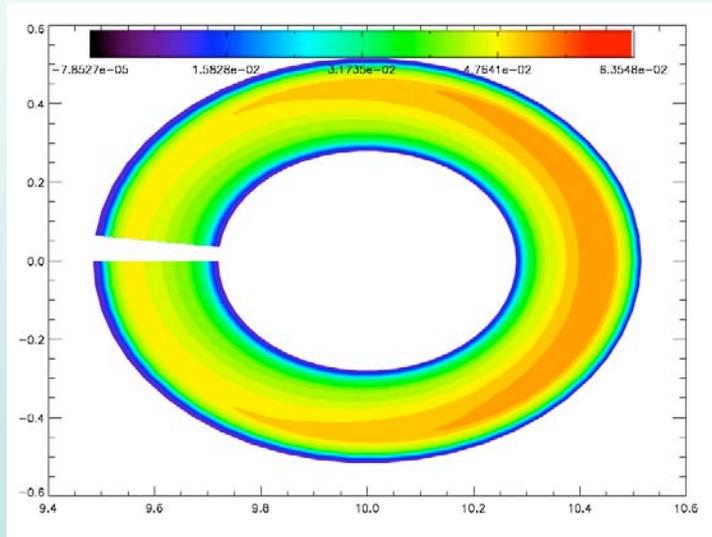
r

Ti

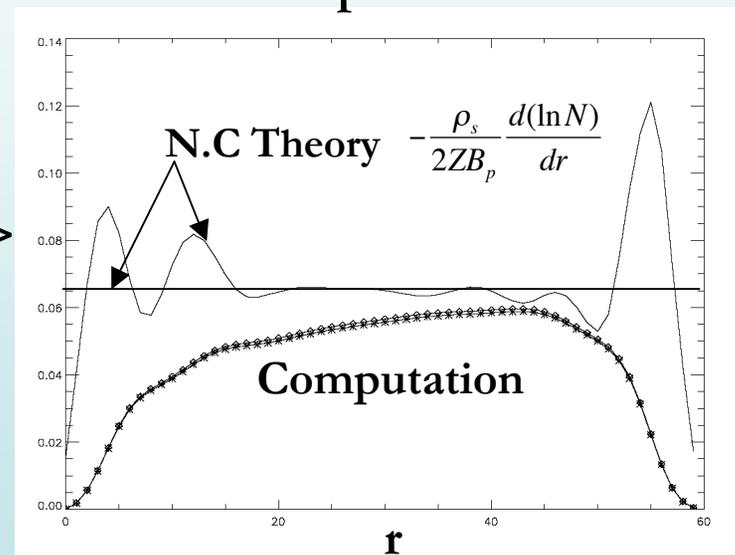


r

Up



<Up>

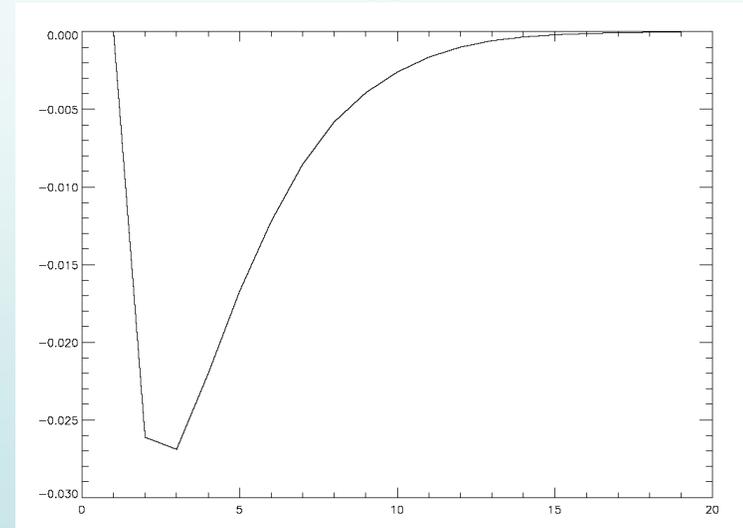
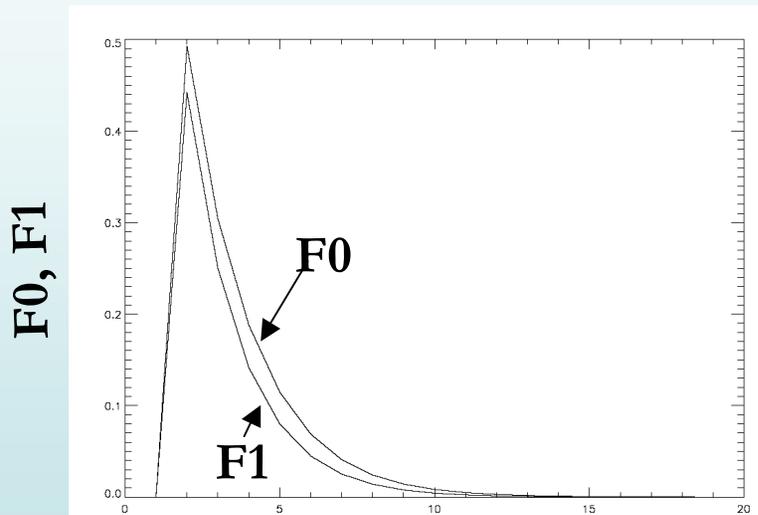
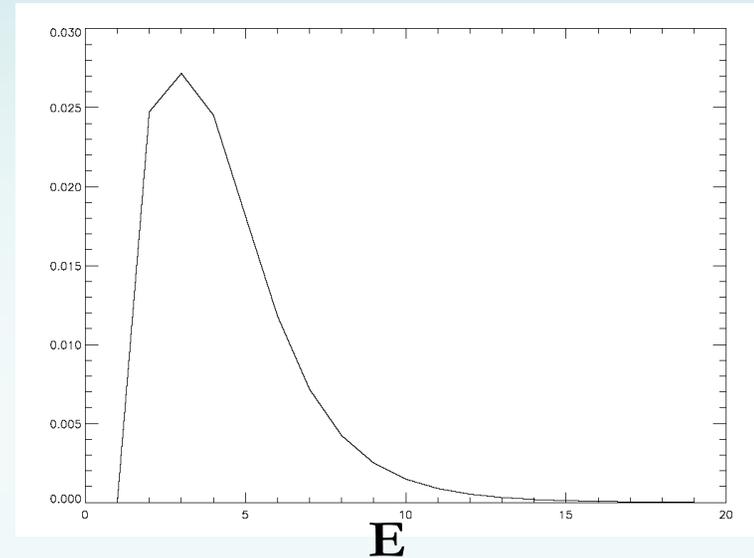
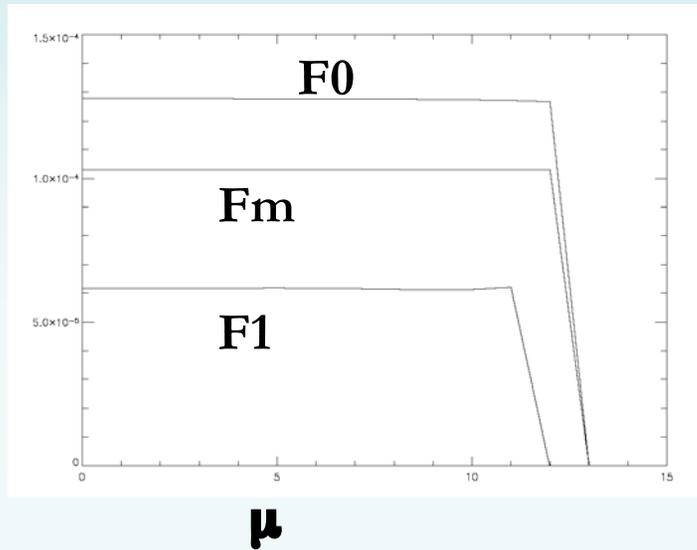


r



Distr. Func.

Under streaming and radial drift, both F0, and F1 deviate from their initial Maxwellian distribution, but with opposite different signs, thus producing Up.





Summary

Numerical algorithms for parallel streaming and radial drift are capable to capture different orbits accurately with poloidal turning points and radial discontinuity.

Parallel flow velocity is produced by streaming and radial drift in the absence of collision, and the magnitude is close to the prediction of the neoclassic theory.

Further tests are needed with different collisionality once the collision package is in place.

Providing a model of edge plasma impurities and neutrals from plasma/wall interactions

G. H. Gilmer, L. A. Zepeda-Ruiz, J. Marian, C. J. Mundy, E. Bringa, C&MS

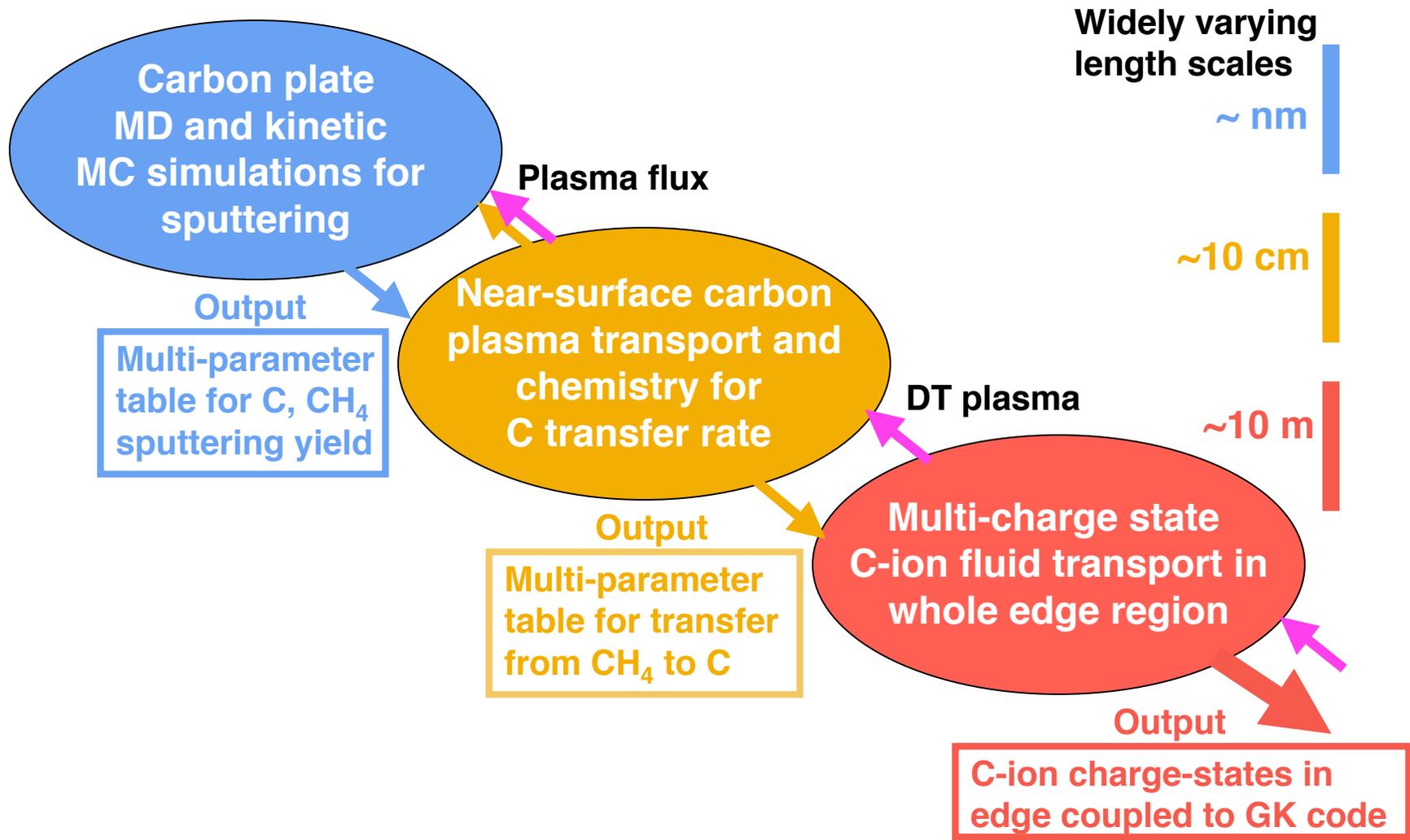
T. D. Rognlien, FEP - LLNL

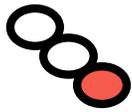
J. P. Verboncoeur and J. Hammel , UCB

Initially, multi-charge-state impurity and neutral species will be provided via the existing 2D UEDGE PYTHON link in TEMPEST

Joint work between C&MS and FEP is providing a quantitative model linking 3 regions for impurities

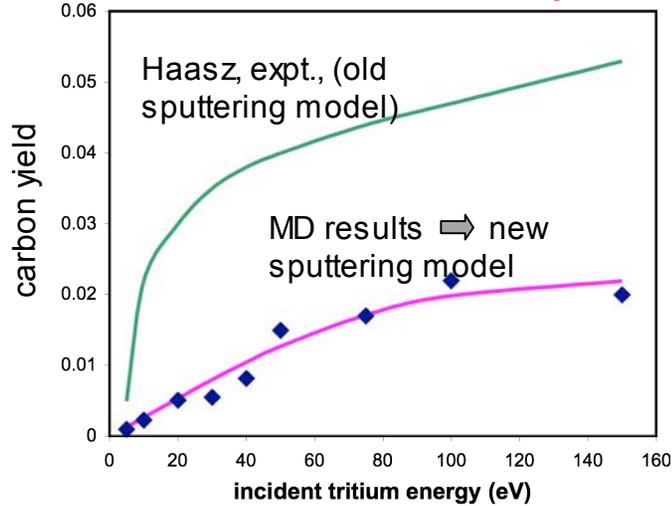
Impurities in the edge plasma are important for power balance



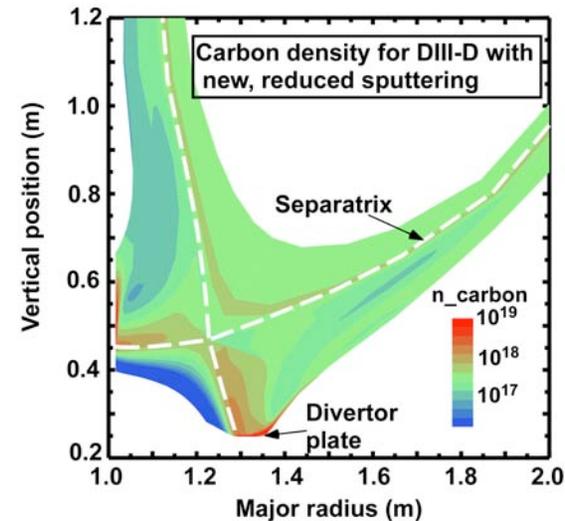
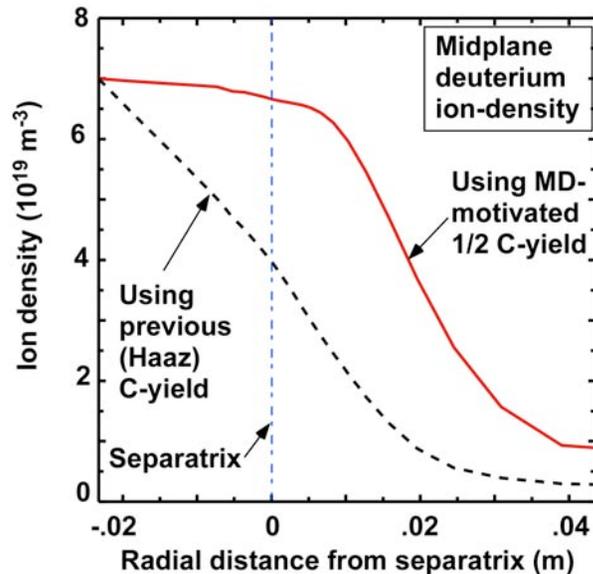
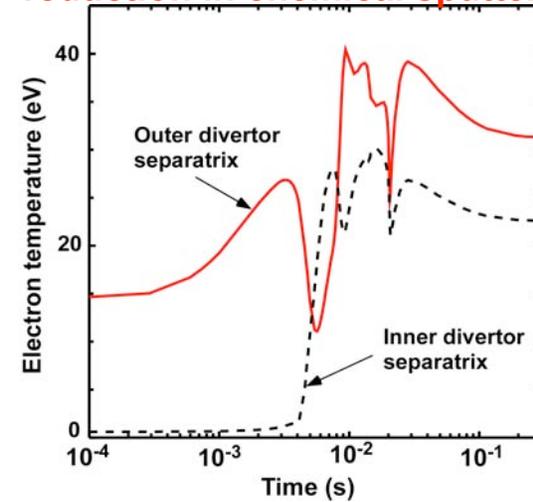


Full edge profile of carbon ions (charges Z= 1-6) from fluid UEDGE provides impurities for FSL code

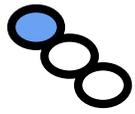
Why do we care about modeling impurities; new MD results show lower C sputtering



UEDGE divertor T_e for a factor of 2 reduction in chemical sputtering



Hydrogen plasma sensitive to C-yield

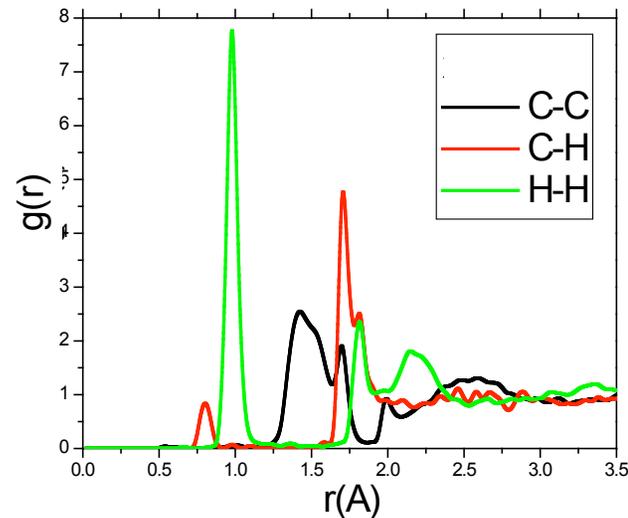
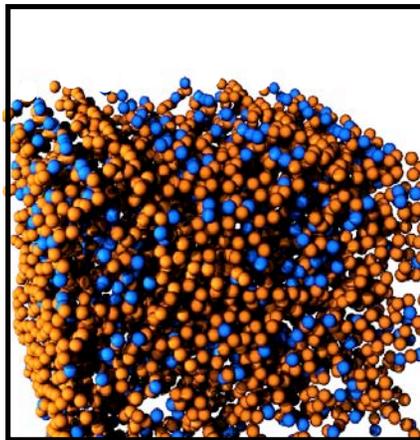


A crucial step for realistic MD simulations is proper construction of the target material

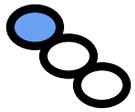
We have succeeded in developing targets corresponding to long-term exposure to reactor plasma as follows:

- amorphous graphite formed by pressurized melting and quenching.
- include 25% H (deuterium/tritium) in target to account for plasma exposure.
- anneal C/H target to stabilize structure.
- bombard with tritium/deuterium to include the effect of steady-state exposure.

Resulting carbon MD target showing H (blue)



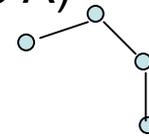
Normalized spatial distribution of lattice ions



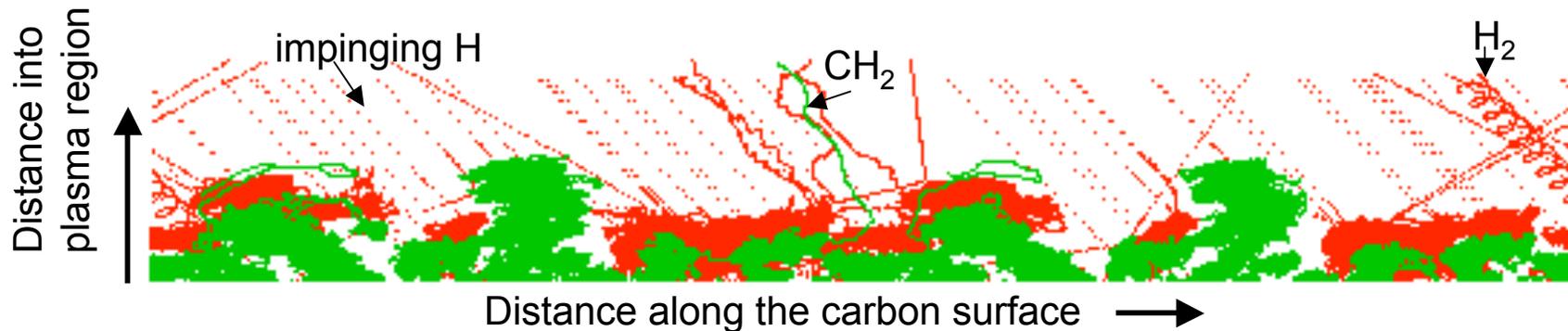
Sputtering MD simulations utilize state-of-the-art AIREBO & Brenner inter-atomic potentials

AIREBO (Adaptive Intermolecular Reactive Empirical Bond Order) potential is an extension of REBO that includes:

- Short-range, bonding interactions from Brenner ($<3 \text{ \AA}$)
- Long-range, non-bonding interactions ($<6 \text{ \AA}$)
- Torsional interactions (4-body) \longrightarrow



Sputtered-particle trajectories (solid colors) just above the surface

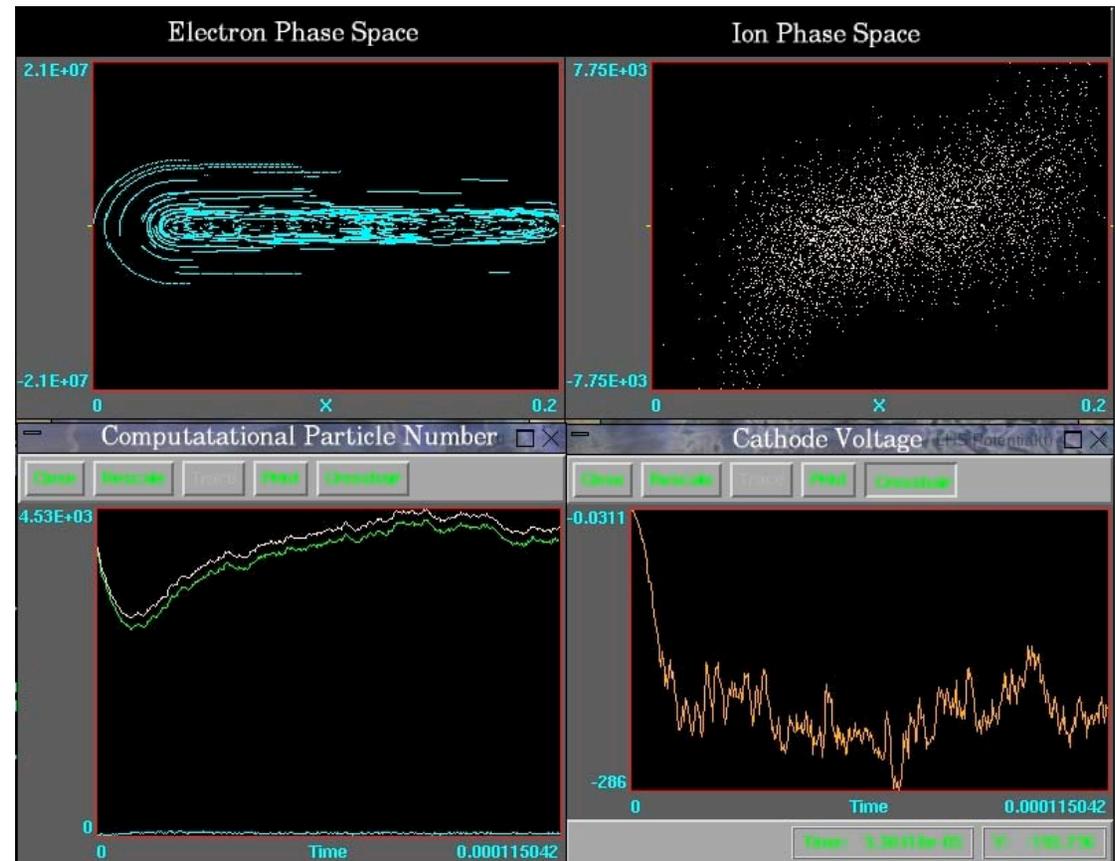


Conditions: $T_{\text{surf}} = 500\text{K}$, $E_{\text{inc}} = 20\text{eV}$, $\Theta_{\text{inc}} = 30^\circ$

After 40 D/T impacts: 6H, 2H₂, CH₂

Near-surface H/C plasma modeled by XOOPIC; developing more efficient Boltzmann electron model

- PIC ions, Boltzmann-PIC hybrid electrons
- Electrons above specified threshold treated as particles – retains kinetic effects, Monte Carlo collision model
- Electron bulk modeled as inertialess Maxwell-Boltzmann distribution:
$$n(\mathbf{x}) = n_0 \exp(-q\phi(\mathbf{x})/T)$$
- Can choose arbitrary Boltzmann electron distribution function, $f(E)$, e.g. with cutoff tails.
- Boltzmann species collisions based on $f(E)$



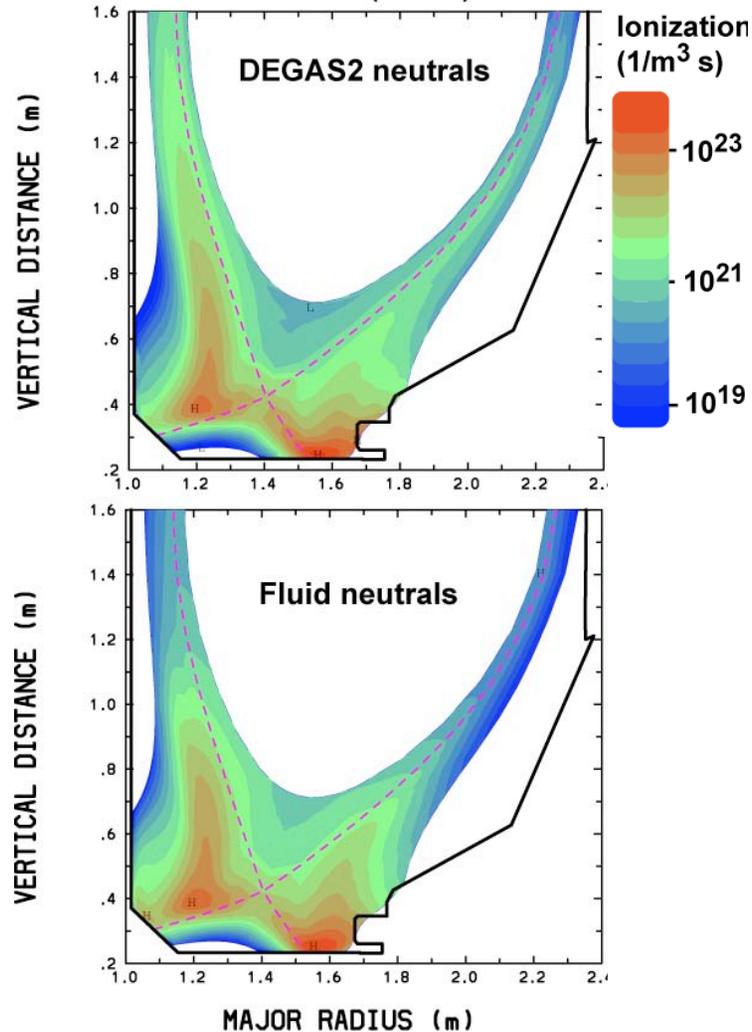
Current-driven 1D DC discharge runs up to 100 times faster than full PIC electron model.

Based on Cartwright et al., *Phys. Plasmas* 7, 3252 (2000).

Initial neutral model will be fluid provided by UEDGE; reasonable comparison to DEGAS; gap effects matter

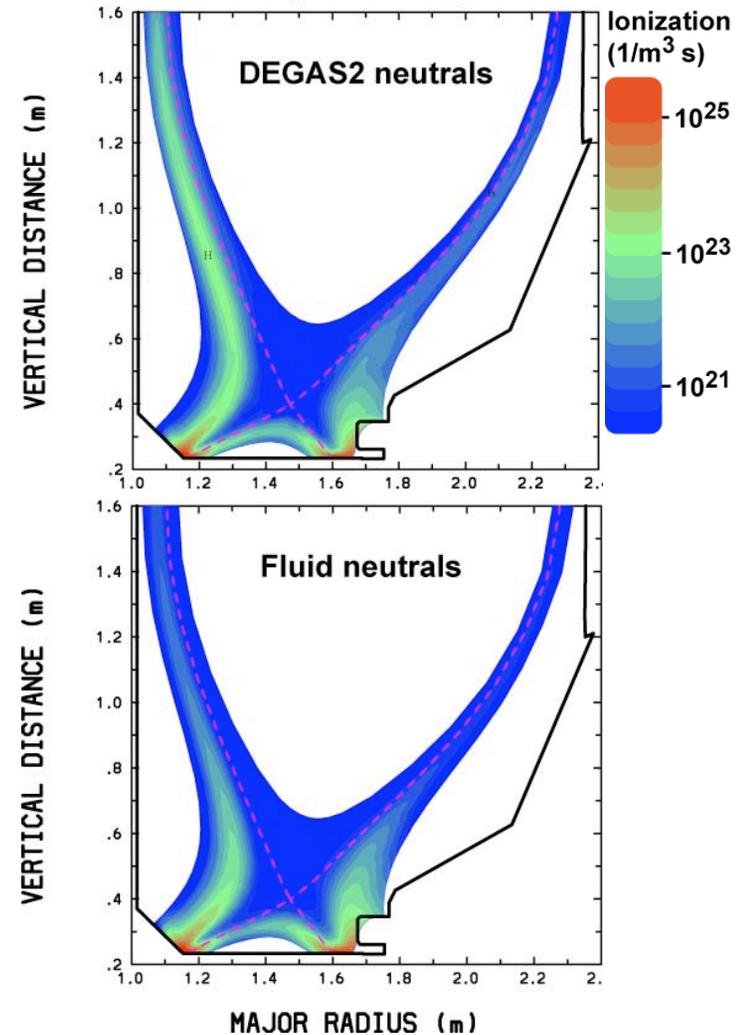
a) $D_{\perp} = 0.65 \text{ m}^2/\text{s}$; L-mode

Low-Density -2
L-mode (SAPP)



b) $D_{\perp} = 0.13 \text{ m}^2/\text{s}$; H-mode

ELMy H-mode,
higher density



Time Integration

- Method of Lines



Time Integration

- **Method of Lines**
 - ▶ Discretize phase space operators



Time Integration

- **Method of Lines**

- ▶ Discretize phase space operators
- ▶ Result is large, coupled nonlinear system of
 - ❖ temporal ordinary differential equations (ODEs)
 - ❖ algebraic constraints



Time Integration

- **Method of Lines**

- ▶ Discretize phase space operators
- ▶ Result is large, coupled nonlinear system of
 - ❖ temporal ordinary differential equations (ODEs)
 - ❖ algebraic constraints
- ▶ That is, a system of Differential-Algebraic Equations (DAEs)



Time Integration

- **Method of Lines**

- ▶ Discretize phase space operators
- ▶ Result is large, coupled nonlinear system of
 - ❖ temporal ordinary differential equations (ODEs)
 - ❖ algebraic constraints
- ▶ That is, a system of Differential-Algebraic Equations (DAEs)

- **Implicit Differential-Algebraic Solver (IDA)**

- ▶ IDA is the DAE solver in the SUNDIALS suite



Time Integration

- **Method of Lines**

- ▶ Discretize phase space operators
- ▶ Result is large, coupled nonlinear system of
 - ❖ temporal ordinary differential equations (ODEs)
 - ❖ algebraic constraints
- ▶ That is, a system of Differential-Algebraic Equations (DAEs)

- **Implicit Differential-Algebraic Solver (IDA)**

- ▶ IDA is the DAE solver in the SUNDIALS suite
 - ❖ SUNDIALS is part of DOE's SciDAC TOPS infrastructure



Time Integration

- **Method of Lines**

- ▶ Discretize phase space operators
- ▶ Result is large, coupled nonlinear system of
 - ❖ temporal ordinary differential equations (ODEs)
 - ❖ algebraic constraints
- ▶ That is, a system of Differential-Algebraic Equations (DAEs)

- **Implicit Differential-Algebraic Solver (IDA)**

- ▶ IDA is the DAE solver in the SUNDIALS suite
 - ❖ SUNDIALS is part of DOE's SciDAC TOPS infrastructure
- ▶ Variable-order Backward Difference Formulas (BDFs) approximate time derivatives



Time Integration

- **Method of Lines**

- ▶ Discretize phase space operators
- ▶ Result is large, coupled nonlinear system of
 - ❖ temporal ordinary differential equations (ODEs)
 - ❖ algebraic constraints
- ▶ That is, a system of Differential-Algebraic Equations (DAEs)

- **Implicit Differential-Algebraic Solver (IDA)**

- ▶ IDA is the DAE solver in the SUNDIALS suite
 - ❖ SUNDIALS is part of DOE's SciDAC TOPS infrastructure
- ▶ Variable-order Backward Difference Formulas (BDFs) approximate time derivatives
- ▶ Pre-conditioned approximate Newton iteration solves system



IDA DAE Formulation

- **Semi-discrete gyro-kinetic system:**

$$\dot{F}_\alpha + H_0 (F_\alpha, \Phi) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

$$H_1 (F_\alpha, \Phi) = 0,$$

where H_0 and H_1 are discrete spatial operators



IDA DAE Formulation

- **Semi-discrete gyro-kinetic system:**

$$\begin{aligned}\dot{F}_\alpha + H_0(F_\alpha, \Phi) &= 0, & \alpha = 1, 2, \dots, N_{\text{species}} \\ H_1(F_\alpha, \Phi) &= 0,\end{aligned}$$

where H_0 and H_1 are discrete spatial operators

- More succinctly,

$$R(\dot{F}_\alpha, F_\alpha, \Phi) = 0,$$

where R is the user-supplied *residual function*



IDA DAE Formulation

- **Semi-discrete gyro-kinetic system:**

$$\begin{aligned}\dot{F}_\alpha + H_0 (F_\alpha, \Phi) &= 0, & \alpha = 1, 2, \dots, N_{\text{species}} \\ H_1 (F_\alpha, \Phi) &= 0,\end{aligned}$$

where H_0 and H_1 are discrete spatial operators

- More succinctly,

$$R(\dot{F}_\alpha, F_\alpha, \Phi) = 0,$$

where R is the user-supplied *residual function*

- **Fully discrete, nonlinear system**, using k -th order BDF, at time n :

$$G(y^n) \equiv G(F_\alpha^n, \Phi^n) \equiv R\left(\frac{1}{h^n} \sum_{i=0}^k \sigma_i^n F_\alpha^{n-i}, F_\alpha^n, \Phi^n\right) = 0$$



IDA Numerical Solution

- **Preconditioned, Inexact Newton iteration on $R = 0$:**

$$P^{-1} J \delta y_m^n = -P^{-1} G(y_m^n), \quad (*)$$

where $\delta y_m^n = y_{m+1}^n - y_m^n$



IDA Numerical Solution

- **Preconditioned, Inexact Newton iteration on $R = 0$:**

$$P^{-1} J \delta y_m^n = -P^{-1} G(y_m^n), \quad (*)$$

where $\delta y_m^n = y_{m+1}^n - y_m^n$

- **Approximate Jacobian:**

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \approx \frac{\partial G}{\partial y} = \frac{\partial R}{\partial y} + \frac{\sigma_0^n}{h^n} \frac{\partial R}{\partial \dot{y}}$$



IDA Numerical Solution

- **Preconditioned, Inexact Newton iteration on $R = 0$:**

$$P^{-1} J \delta y_m^n = -P^{-1} G(y_m^n), \quad (*)$$

where $\delta y_m^n = y_{m+1}^n - y_m^n$

- **Approximate Jacobian:**

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \approx \frac{\partial G}{\partial y} = \frac{\partial R}{\partial y} + \frac{\sigma_0^n}{h^n} \frac{\partial R}{\partial \dot{y}}$$

- **Preconditioner:**

$$P = \text{diag} \left(\frac{\sigma_0^n}{h^n} I, \tilde{J}_{22} \right)$$



IDA Numerical Solution

- **Preconditioned, Inexact Newton iteration on $R = 0$:**

$$P^{-1} J \delta y_m^n = -P^{-1} G(y_m^n), \quad (*)$$

where $\delta y_m^n = y_{m+1}^n - y_m^n$

- **Approximate Jacobian:**

$$J = \begin{pmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{pmatrix} \approx \frac{\partial G}{\partial y} = \frac{\partial R}{\partial y} + \frac{\sigma_0^n}{h^n} \frac{\partial R}{\partial \dot{y}}$$

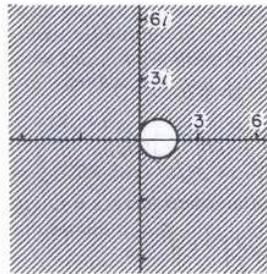
- **Preconditioner:**

$$P = \text{diag} \left(\frac{\sigma_0^n}{h^n} I, \tilde{J}_{22} \right)$$

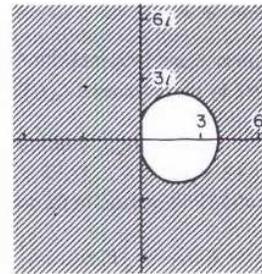
- Linear system (*) solved using GMRES



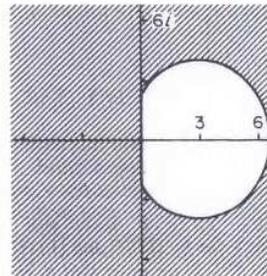
BDF Stability Domain



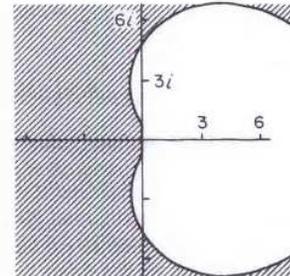
$k = 1$



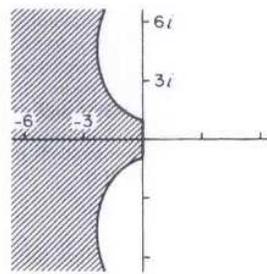
$k = 2$



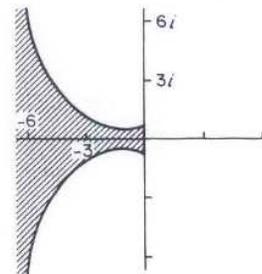
$k = 3$



$k = 4$



$k = 5$



$k = 6$

- From J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*, Wiley (1991), pg. 100



Parallel Streaming: Issues

- **Solve** in *phase space* (x, ε) , where $x = (\psi, \theta, \zeta)$ and $\varepsilon = (\epsilon_0, \mu)$:

$$\left(\frac{\partial}{\partial t} + \frac{v_{\parallel}}{h} \frac{\partial}{\partial \theta} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

where

$$\frac{1}{2} m_{\alpha} v_{\parallel}^2 \equiv \epsilon_0 - \mu B(x) - q_{\alpha} e \Phi_0(x) \geq 0$$



Parallel Streaming: Issues

- **Solve** in *phase space* (x, ε) , where $x = (\psi, \theta, \zeta)$ and $\varepsilon = (\epsilon_0, \mu)$:

$$\left(\frac{\partial}{\partial t} + \frac{v_{\parallel}}{h} \frac{\partial}{\partial \theta} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

where

$$\frac{1}{2} m_{\alpha} v_{\parallel}^2 \equiv \epsilon_0 - \mu B(x) - q_{\alpha} e \Phi_0(x) \geq 0$$

- **Requires:**
 - ▶ Low *numerical dissipation* \Rightarrow high fidelity through dozens of bounces
 - ▶ Methods to deal with *arbitrary trapping regions*
 - ▶ Efficient algorithms compatible with *parallel decomposition* in θ



Parallel Streaming: Approach

- **Low Numerical Dissipation:**
 - ▶ Use total energy (ϵ_0) as an independent variable



Parallel Streaming: Approach

- **Low Numerical Dissipation:**

- ▶ Use total energy (ϵ_0) as an independent variable \Rightarrow

- ❖ coordinate aligns with direction of motion

- ❖ two “sheets” of F_α connect through the *turning point boundary*



Parallel Streaming: Approach

- **Low Numerical Dissipation:**

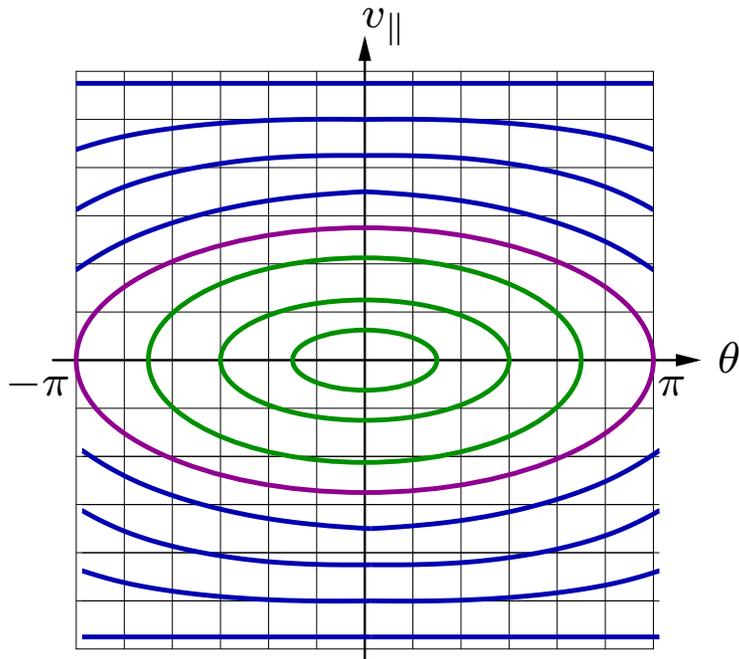
- ▶ Use total energy (ϵ_0) as an independent variable \Rightarrow
 - ❖ coordinate aligns with direction of motion
 - ❖ two “sheets” of F_α connect through the *turning point boundary*
- ▶ Use standard *fourth-order, upwind* finite differences along orbit



Parallel Streaming: Approach

- **Low Numerical Dissipation:**

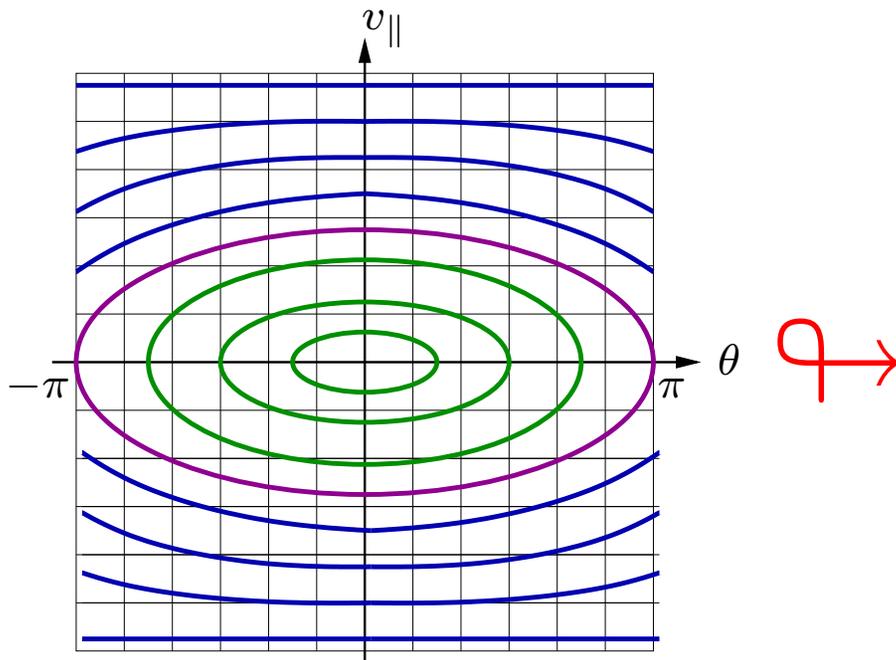
- ▶ Use total energy (ϵ_0) as an independent variable \Rightarrow
 - ❖ coordinate aligns with direction of motion
 - ❖ two “sheets” of F_α connect through the *turning point boundary*
- ▶ Use standard *fourth-order, upwind* finite differences along orbit



Parallel Streaming: Approach

- **Low Numerical Dissipation:**

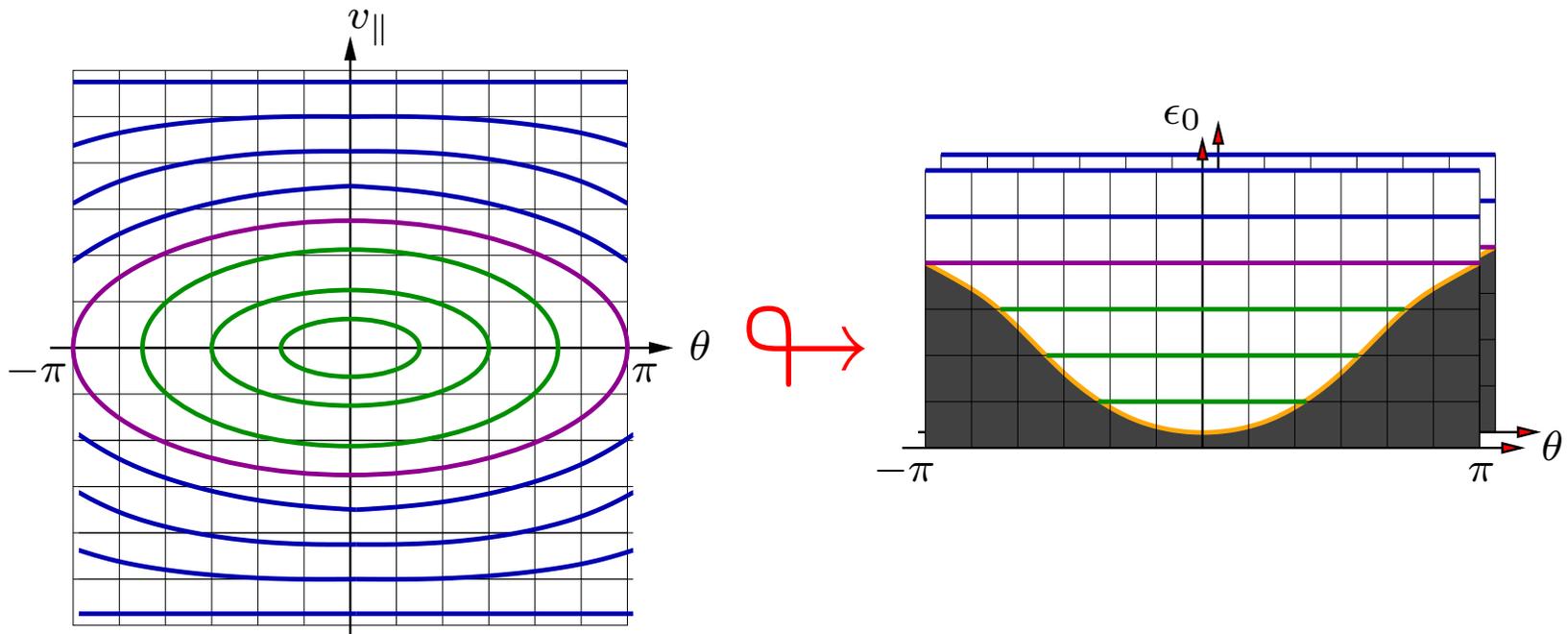
- ▶ Use total energy (ϵ_0) as an independent variable \Rightarrow
 - ❖ coordinate aligns with direction of motion
 - ❖ two “sheets” of F_α connect through the *turning point boundary*
- ▶ Use standard *fourth-order, upwind* finite differences along orbit



Parallel Streaming: Approach

- **Low Numerical Dissipation:**

- ▶ Use total energy (ϵ_0) as an independent variable \Rightarrow
 - ❖ coordinate aligns with direction of motion
 - ❖ two “sheets” of F_α connect through the *turning point boundary*
- ▶ Use standard *fourth-order, upwind* finite differences along orbit



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**
 - ▶ Construct *orbit lists* that describe the index space connectivity



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**

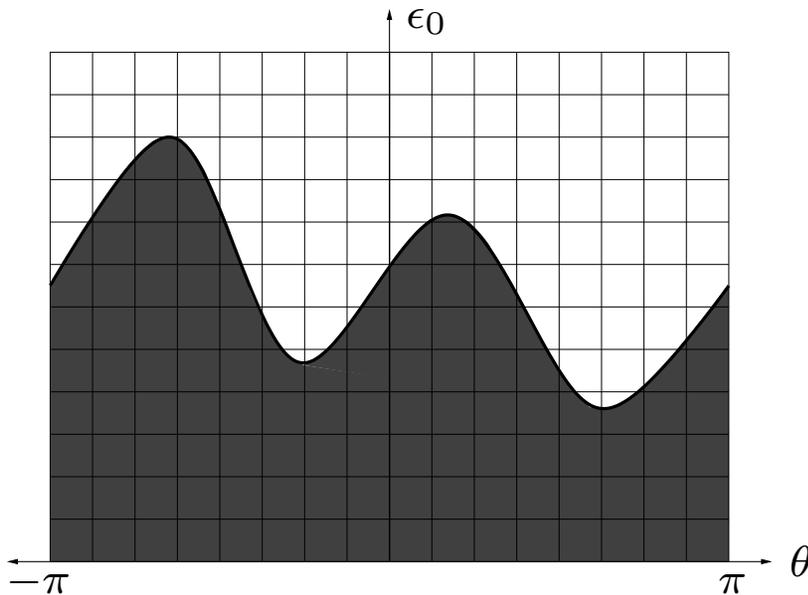
- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**

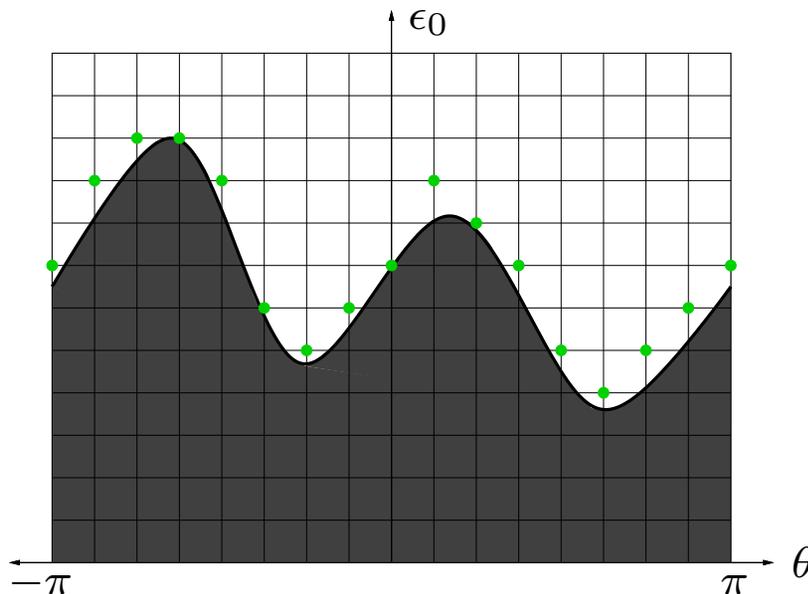
- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



1. Find turning points θ -sweep

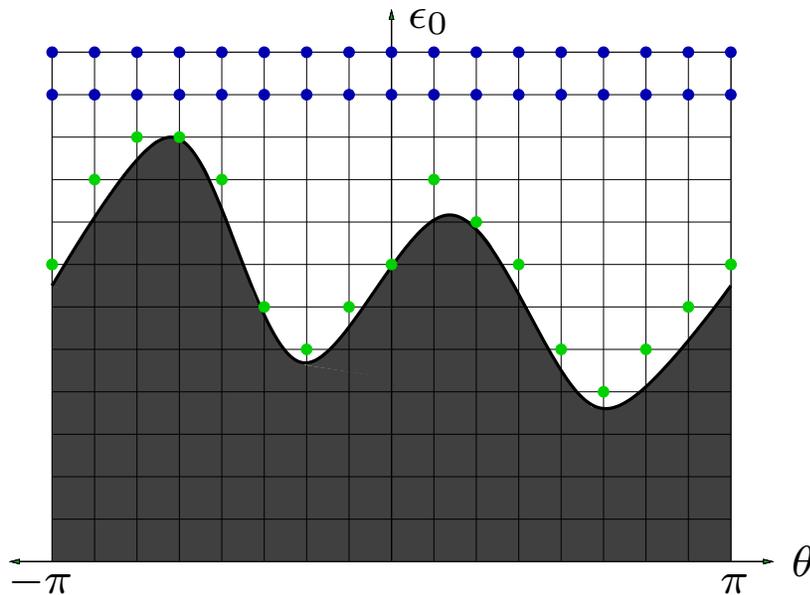
$$m_{ijn}^* = \left\lceil \frac{\mu_n B_{ij} + q\Phi_{0,ij}}{\Delta\epsilon} \right\rceil$$



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



1. Find turning points θ -sweep
2. $\forall m > m_{\max}$, full interval

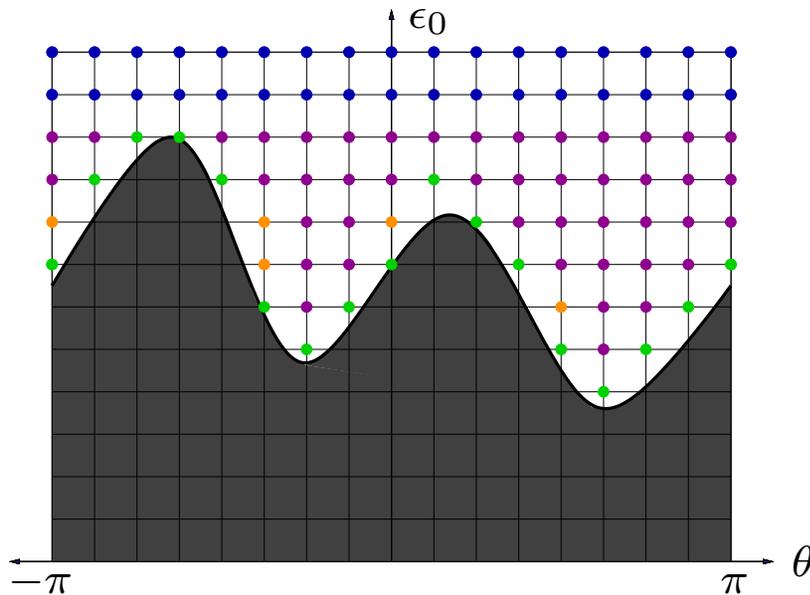
$$m_{ijn}^* = \left\lceil \frac{\mu_n B_{ij} + q\Phi_{0,ij}}{\Delta\epsilon} \right\rceil$$



Parallel Streaming: Approach (cont'd)

- **Multiple trapping regions:**

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



1. Find turning points θ -sweep
2. $\forall m > m_{\max}$, full interval
3. Sweep in θ and build intervals

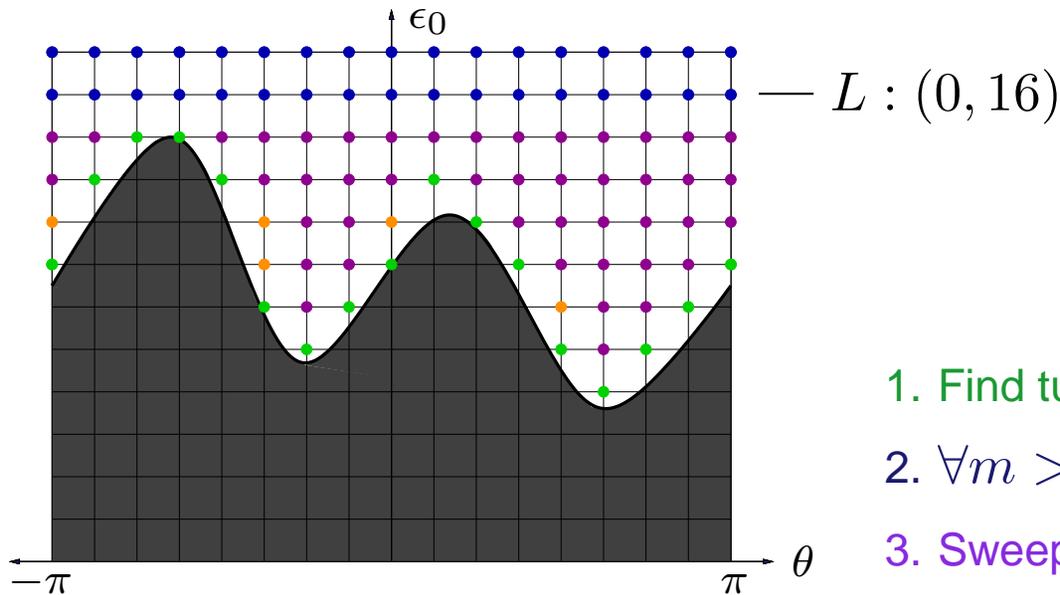
$$m_{ijn}^* = \left\lceil \frac{\mu_n B_{ij} + q\Phi_{0,ij}}{\Delta\epsilon} \right\rceil$$



Parallel Streaming: Approach (cont'd)

- Multiple trapping regions:

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



1. Find turning points θ -sweep
2. $\forall m > m_{\max}$, full interval
3. Sweep in θ and build intervals

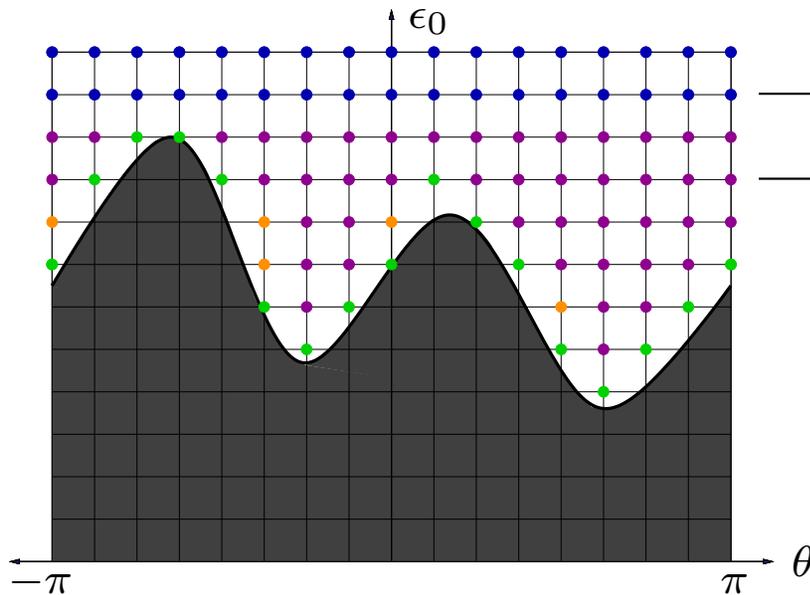
$$m_{ijn}^* = \left\lceil \frac{\mu_n B_{ij} + q\Phi_{0,ij}}{\Delta\epsilon} \right\rceil$$



Parallel Streaming: Approach (cont'd)

- Multiple trapping regions:

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



— $L : (0, 16)$
 — $L : (0, 1] \rightarrow [4, 16)$

1. Find turning points θ -sweep
2. $\forall m > m_{\max}$, full interval
3. Sweep in θ and build intervals

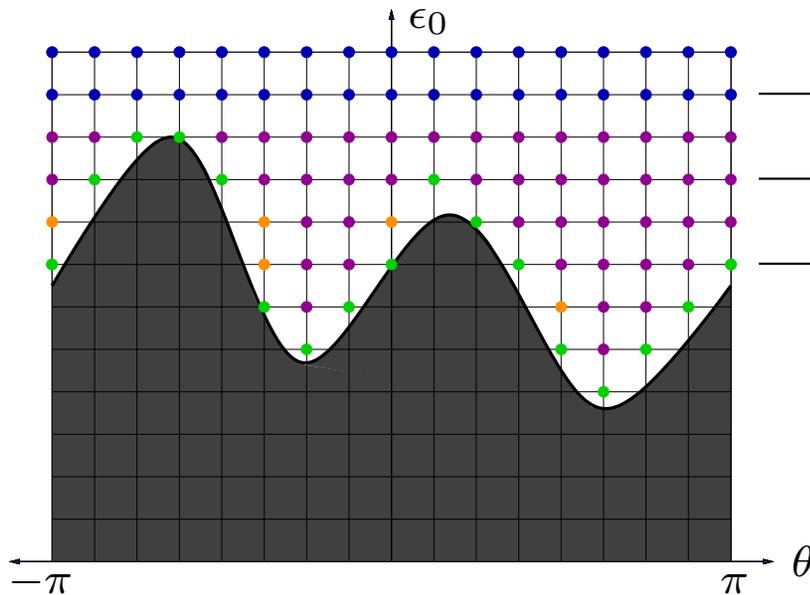
$$m_{ijn}^* = \left\lceil \frac{\mu_n B_{ij} + q\Phi_{0,ij}}{\Delta\epsilon} \right\rceil$$



Parallel Streaming: Approach (cont'd)

- Multiple trapping regions:

- ▶ Construct *orbit lists* that describe the index space connectivity
 - ❖ Each populated interval \mathcal{I} defined by *poloidal index pairs* $[j_a, j_b]$
 - ❖ Each populated interval end point is “*passing*” or “*turning point*”
- ▶ Populated intervals for each $(\psi_i, \epsilon_{0,m}, \mu_n)$ stored in *linked list* L_{imn}
 - ❖ Allows for multiple trapping regions



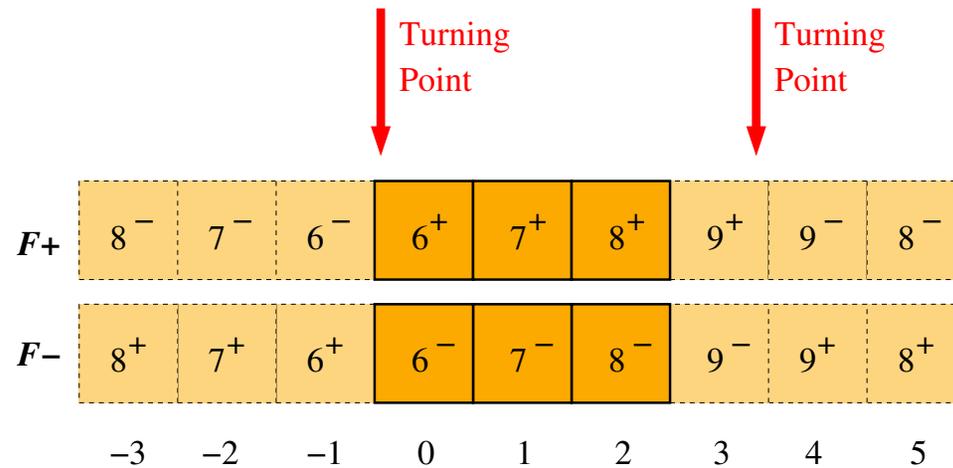
- $L : (0, 16)$
- $L : (0, 1] \rightarrow [4, 16)$
- $L : (0, 0] \rightarrow [5, 8] \rightarrow [11, 16)$

1. Find turning points θ -sweep
2. $\forall m > m_{\max}$, full interval
3. Sweep in θ and build intervals

$$m_{ijn}^* = \left\lceil \frac{\mu_n B_{ij} + q\Phi_{0,ij}}{\Delta\epsilon} \right\rceil$$



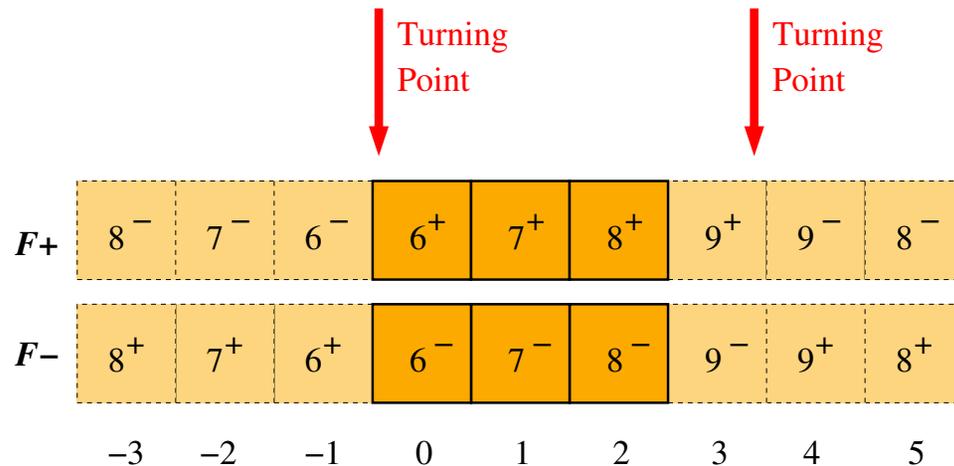
Parallel Streaming: Evaluation



1. **For each** ψ_i , ϵ_m , and μ_n :
 - (a) **For each** interval \mathcal{I} in the list L_{imn} :



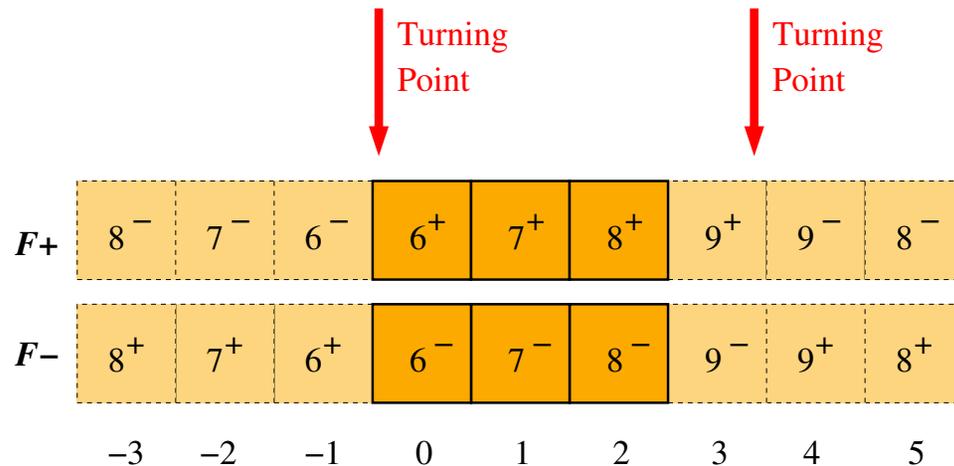
Parallel Streaming: Evaluation



1. **For each** ψ_i , ϵ_m , and μ_n :
 - (a) **For each** interval \mathcal{I} in the list L_{imn} :
 - i. **Copy** local intersections of both sheets into temporary arrays



Parallel Streaming: Evaluation



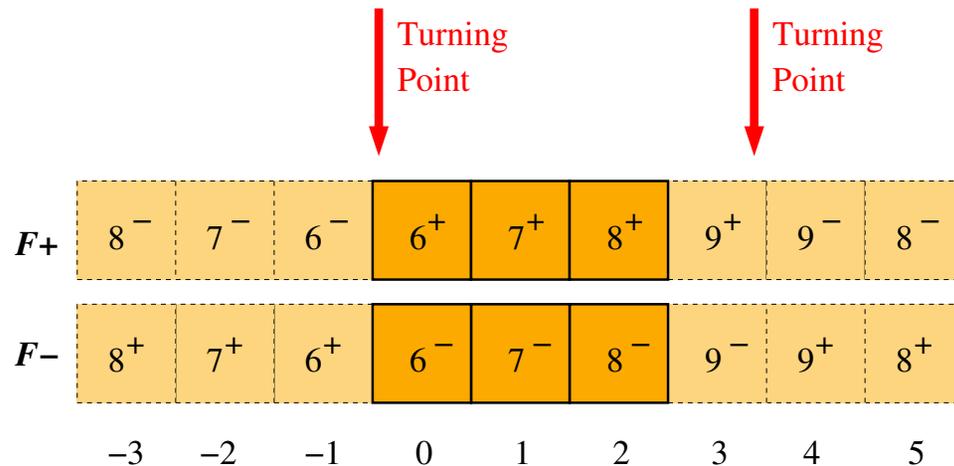
1. **For each** ψ_i , ϵ_m , and μ_n :

(a) **For each** interval \mathcal{I} in the list L_{imn} :

- i. **Copy** local intersections of both sheets into temporary arrays
- ii. **If** an end is a turning point, **copy** from *other sheet* into ghost cells



Parallel Streaming: Evaluation



1. **For each** ψ_i , ϵ_m , and μ_n :
 - (a) **For each** interval \mathcal{I} in the list L_{imn} :
 - i. **Copy** local intersections of both sheets into temporary arrays
 - ii. **If** an end is a turning point, **copy** from *other sheet* into ghost cells
 - iii. **Difference** and **add** results into the residual vector



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend
- Either fourth-order upwind or *fifth-order upwind WENO* discretization
 - ▶ Weighted Essentially Non-Oscillatory



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend
- Either fourth-order upwind or *fifth-order upwind WENO* discretization
 - ▶ Weighted Essentially Non-Oscillatory
 - ▶ Takes *convex combination* of three third-order approximations



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend
- Either fourth-order upwind or *fifth-order upwind WENO* discretization
 - ▶ Weighted Essentially Non-Oscillatory
 - ▶ Takes *convex combination* of three third-order approximations
 - ▶ *For smooth data*, combination gives fifth-order approximation



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend
- Either fourth-order upwind or *fifth-order upwind WENO* discretization
 - ▶ Weighted Essentially Non-Oscillatory
 - ▶ Takes *convex combination* of three third-order approximations
 - ▶ *For smooth data*, combination gives fifth-order approximation
 - ▶ *For discontinuous data*, weights bias to smoothest approximation



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend
- Either fourth-order upwind or *fifth-order upwind WENO* discretization
 - ▶ Weighted Essentially Non-Oscillatory
 - ▶ Takes *convex combination* of three third-order approximations
 - ▶ *For smooth data*, combination gives fifth-order approximation
 - ▶ *For discontinuous data*, weights bias to smoothest approximation
- WENO automatically handles the low-energy boundary



Radial Drift

- Currently only radial drifts implemented

$$\left(\frac{\partial}{\partial t} + v_{d\psi} \frac{\partial}{\partial \psi} \right) F_{\alpha}(x, \varepsilon, t) = 0, \quad \alpha = 1, 2, \dots, N_{\text{species}}$$

- Only a single interval with which to contend
- Either fourth-order upwind or *fifth-order upwind WENO* discretization
 - ▶ Weighted Essentially Non-Oscillatory
 - ▶ Takes *convex combination* of three third-order approximations
 - ▶ *For smooth data*, combination gives fifth-order approximation
 - ▶ *For discontinuous data*, weights bias to smoothest approximation
- WENO automatically handles the low-energy boundary
- A *pseudo-flux* formulation is used (nonconservative form)



Collision Operator for Gyrokinetic Tokamak Edge Model

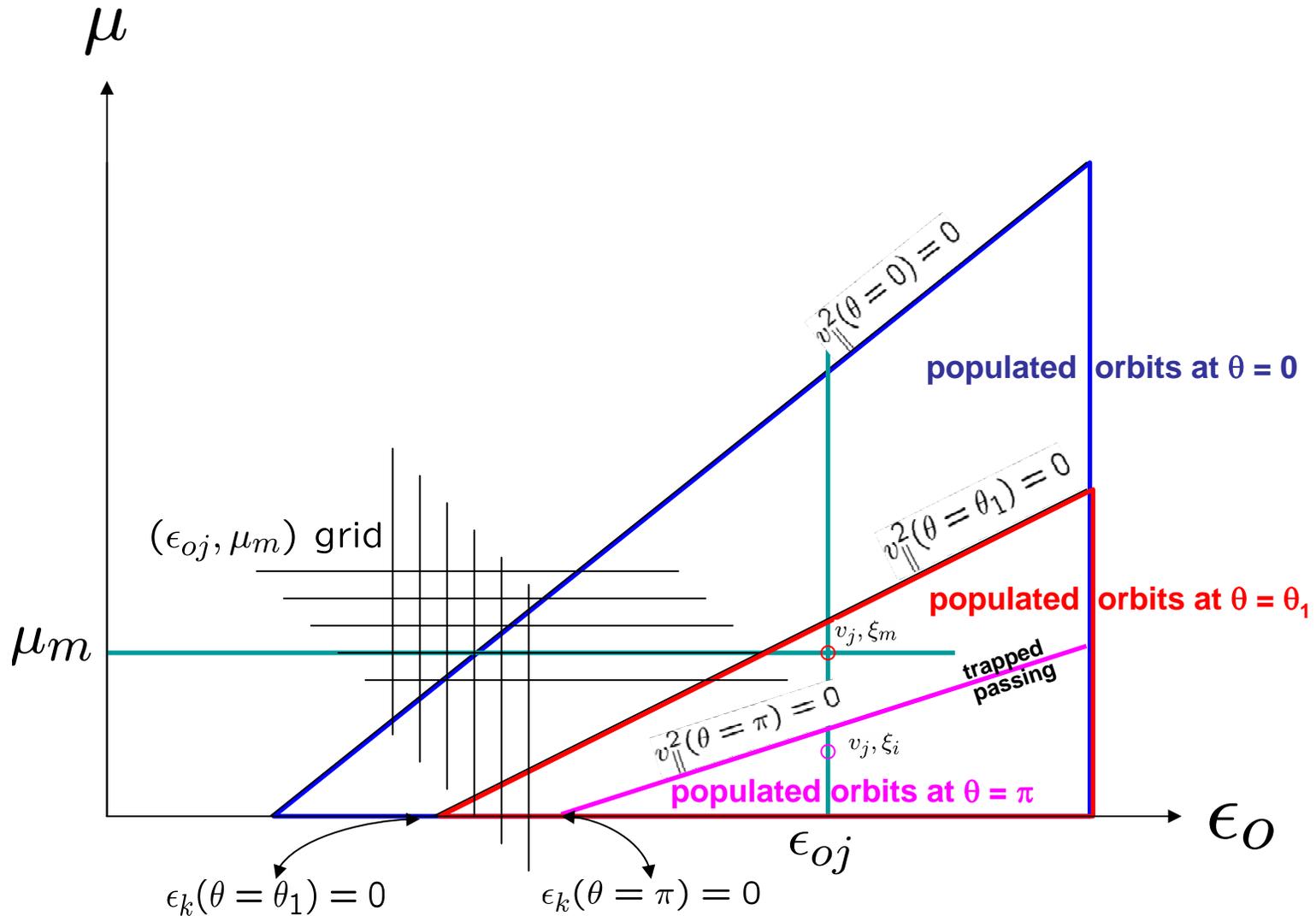
Contributions from Bob Harvey,
Ken Kupfer and Mike McCoy

Vlasov Fokker-Planck Equation in (v, ξ)

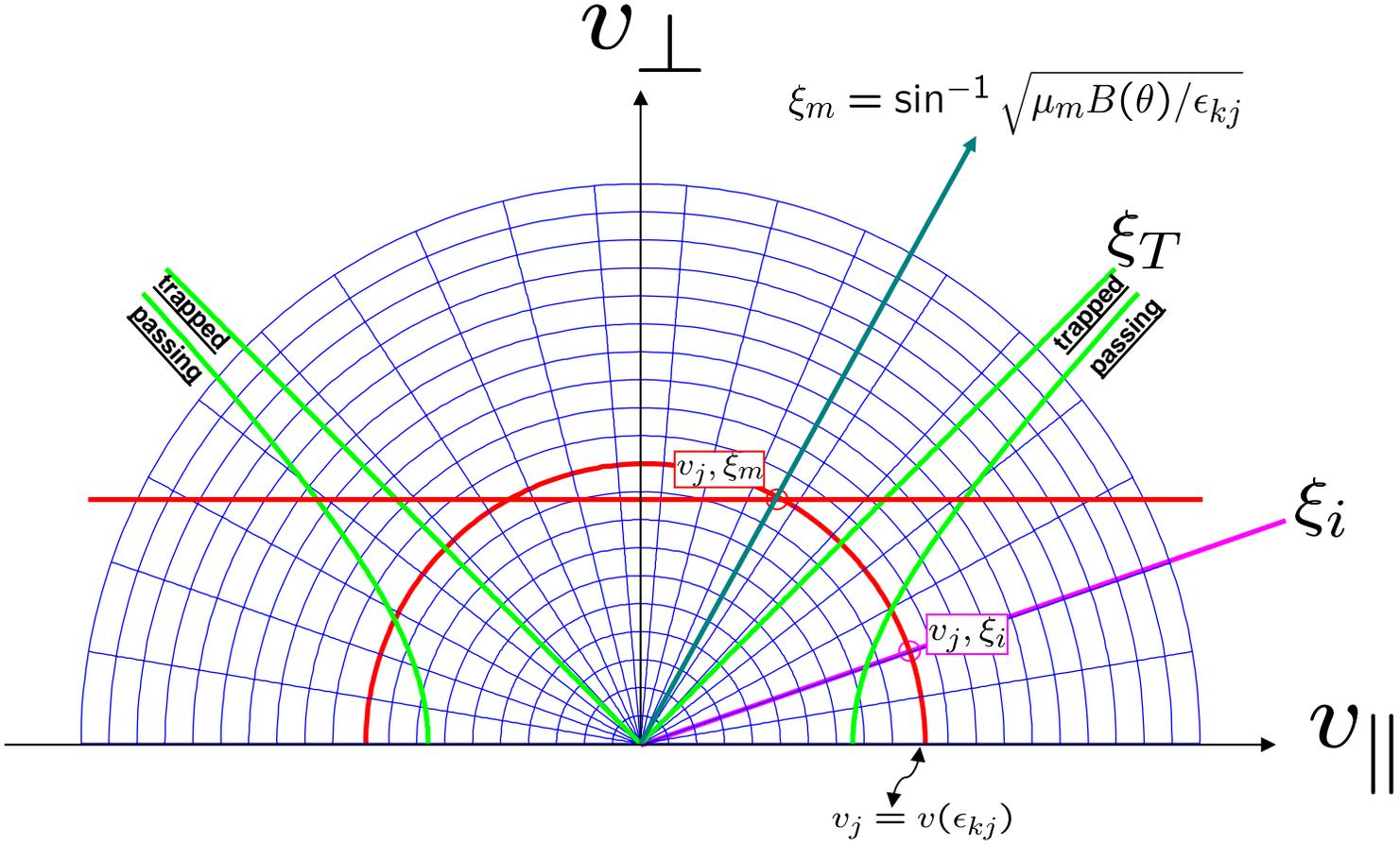
$$\frac{\partial f_k}{\partial t} + \mathbf{v} \cdot \nabla f_k + \nabla_{\mathbf{v}} \cdot \left(\frac{q_k}{m_k} (\mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B}) f_k + \Gamma_{ck} \right) = 0$$

$$\begin{aligned} \left(\frac{\delta f_k}{\delta t} \right)_c &= - \langle \nabla_{\mathbf{v}} \cdot \Gamma_{ck} \rangle_{\phi} \\ &= \gamma_k \left(\frac{1}{v^2} \frac{\partial \mathcal{G}_k}{\partial v} + \frac{1}{v^2 \sin \xi} \frac{\partial \mathcal{H}_k}{\partial \xi} \right) \\ &= \gamma_k \left(\frac{1}{v^2} \frac{\partial}{\partial v} (A_k + B_k \frac{\partial}{\partial v} + C_k \frac{\partial}{\partial \xi}) \right. \\ &\quad \left. + \frac{1}{v^2 \sin \xi} \frac{\partial}{\partial \xi} (D_k + E_k \frac{\partial}{\partial v} + F_k \frac{\partial}{\partial \xi}) \right) f_k \end{aligned}$$

(ϵ_0, μ) COM Space

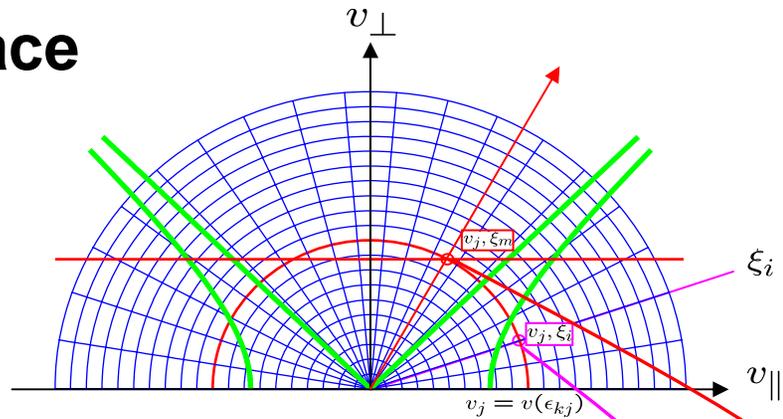


Collision Operator v - ξ Space

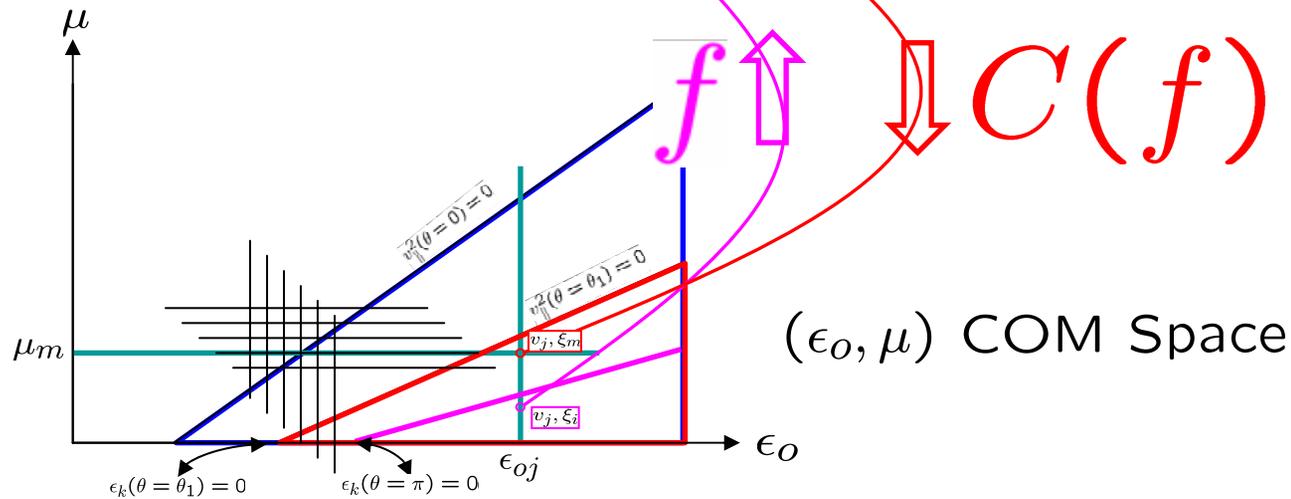


Interpolation Scheme

v- ξ Space

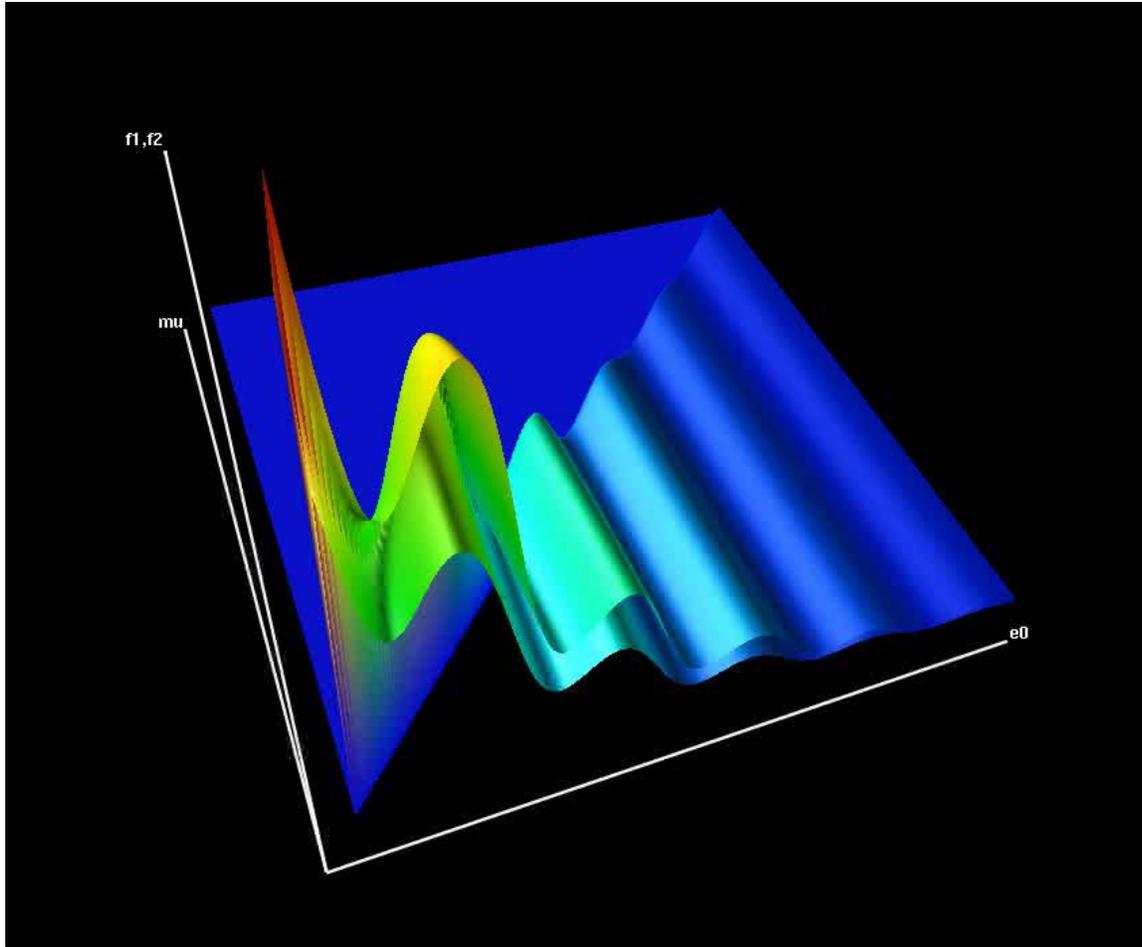


**Donor
interpolants map
to grid points**

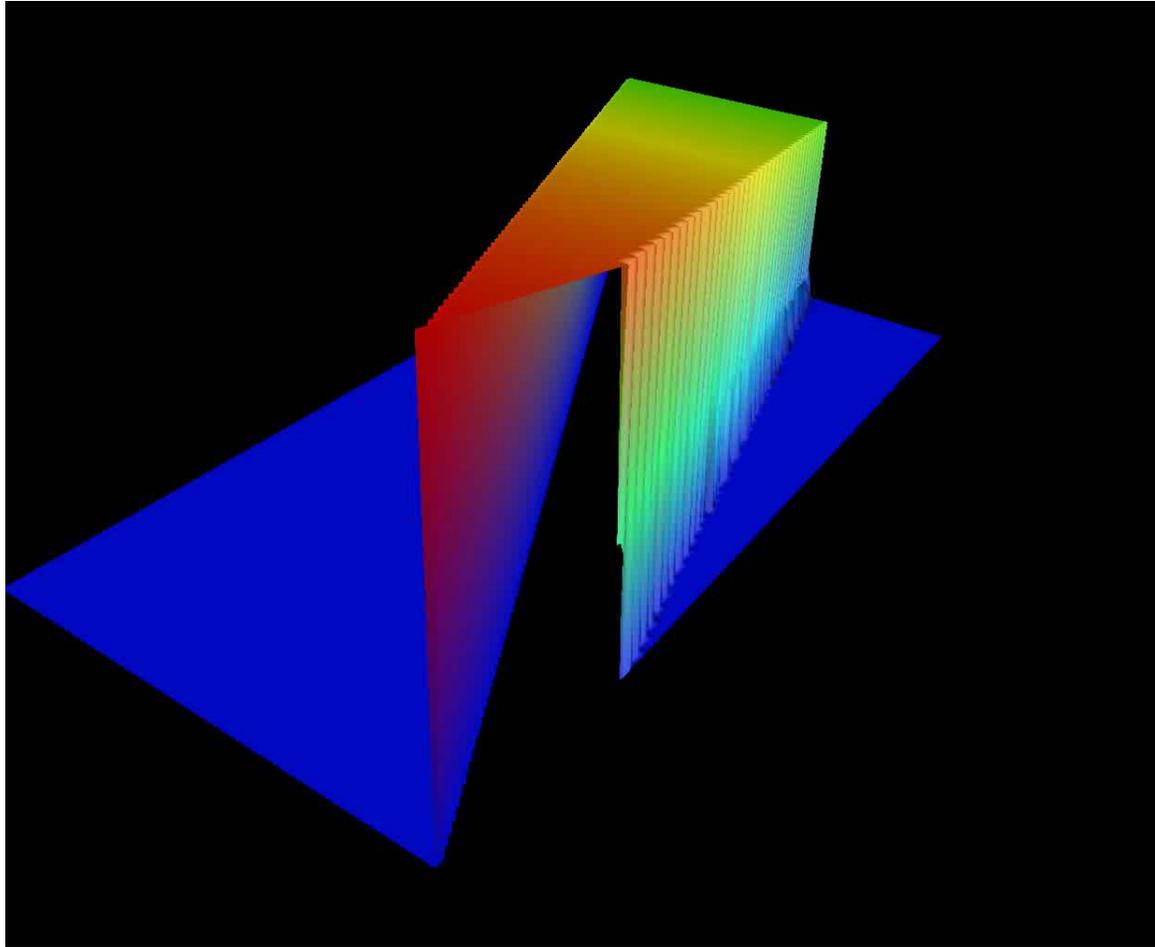


(ϵ_0, μ) COM Space

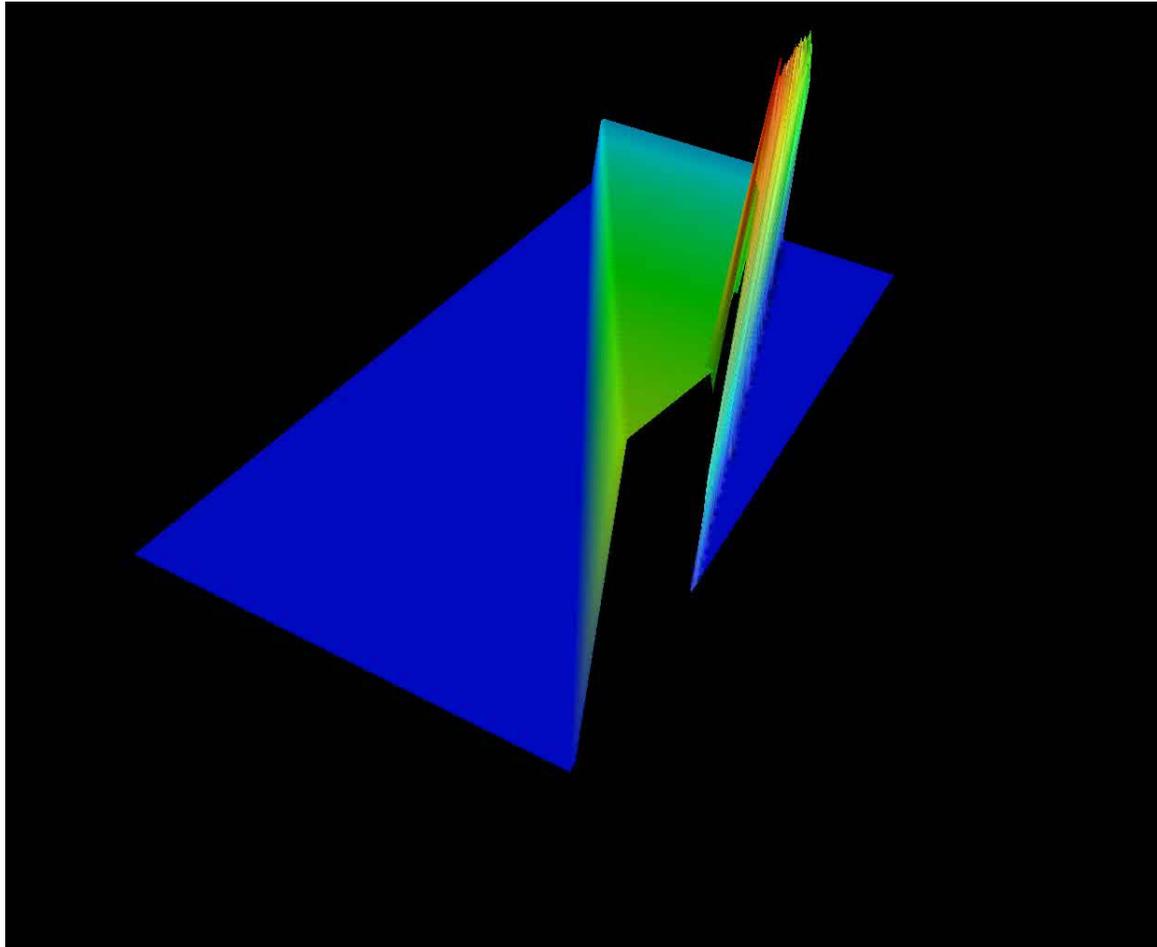
Smooth f OK



**f with fine structure
is more difficult**



log(f) with loss cone



$$\log\left(\left|\left(\frac{\partial f}{\partial t}\right)_c\right|\right)$$

**Better approach:
transform the collision operator
from (v, θ) to (ϵ_0, μ)**

Difference in (ϵ_0, μ)

$$\left(\frac{\delta f}{\delta t}\right)_c = v_{\parallel} \frac{\partial}{\partial \epsilon_0} \Gamma_{\epsilon_0} + v_{\parallel} \frac{\partial}{\partial \mu} \Gamma_{\mu}, \quad (1)$$

$$\Gamma_{\epsilon_0} = D_{\epsilon_0} f + D_{\epsilon_0 \epsilon_0} \frac{\partial f}{\partial \epsilon_0} + D_{\epsilon_0 \mu} \frac{\partial f}{\partial \mu}, \quad (2)$$

$$\Gamma_{\mu} = D_{\mu} f + D_{\mu \epsilon_0} \frac{\partial f}{\partial \epsilon_0} + D_{\mu \mu} \frac{\partial f}{\partial \mu} \quad (3)$$

Compute nonlinear FP coefficients in (v, ξ) and interpolate as before

The relation of coefficients between (v, ξ) to (ϵ_0, μ) space are:

$$D_{\epsilon_0} = \frac{m}{vv_{\parallel}} A_k, \quad (1)$$

$$D_{\epsilon_0 \epsilon_0} = \frac{m^2}{v_{\parallel}} B_k, \quad (2)$$

$$D_{\epsilon_0 \mu} = \frac{m^2 v_{\perp}}{v^2 v_{\parallel} B} (v_{\perp} B_k + vv_{\parallel} C_k), \quad (3)$$

$$D_{\mu} = \frac{2\mu}{v^3 v_{\parallel}} A_k + \frac{m}{vB} D_k, \quad (4)$$

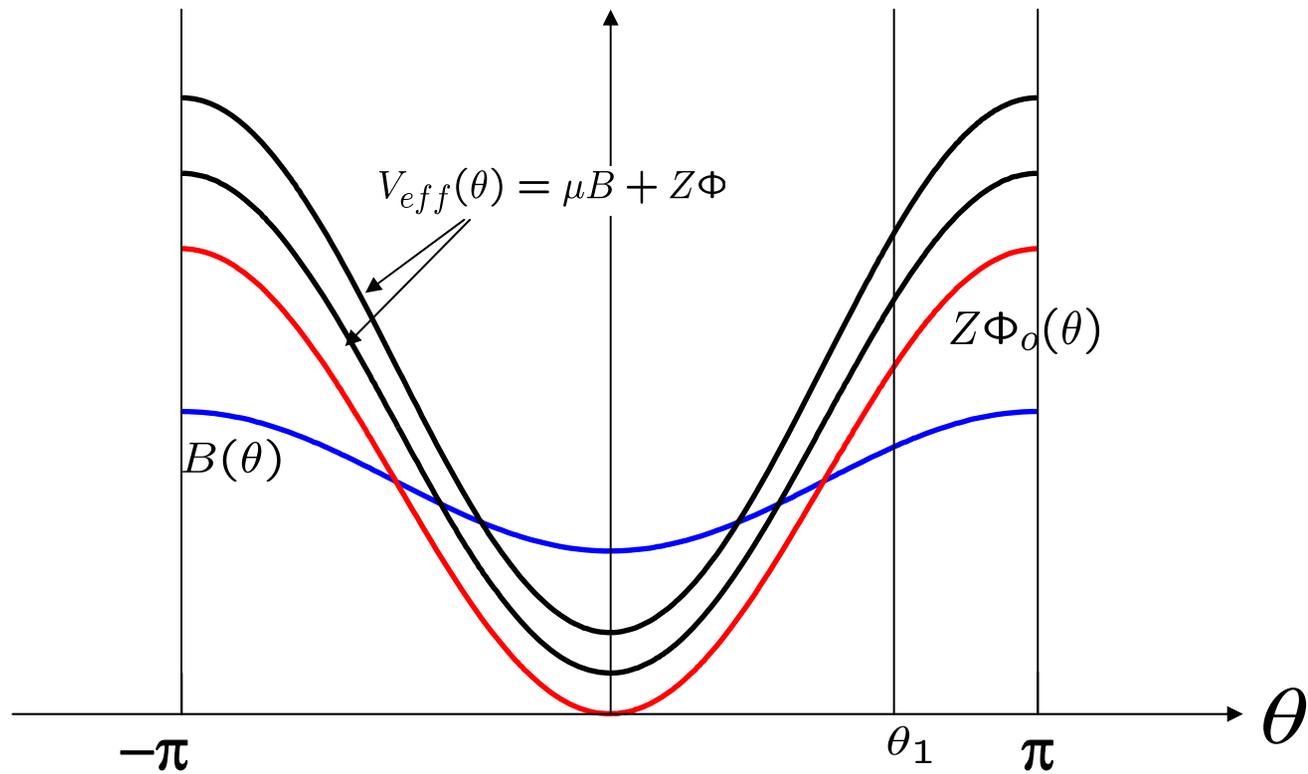
$$D_{\mu \epsilon_0} = \frac{2m\mu}{v^2 v_{\parallel}} B_k + \frac{m^2}{B} E_k, \quad (5)$$

$$D_{\mu \mu} = \frac{2m\mu v_{\perp}}{v^4 v_{\parallel} B} (v_{\perp} B_k + vv_{\parallel} C_k) \quad (6)$$

$$+ \frac{m^2 v_{\perp}}{v^2 B^2} (v_{\perp} E_k + vv_{\parallel} F_k). \quad (7)$$

where $v_{\perp} = \sqrt{2B\mu/m}$ and $v_{\parallel} = \pm \sqrt{\frac{2}{m} (\epsilon_0 - \mu B - q\Phi_0)}$.

Effective Potential Along Magnetic Field



GK Poisson Equation



$$\left(\sum_{\alpha} \frac{\rho_{\alpha}^2}{2\lambda_{D\alpha}^2} \right) \nabla_{\perp}^2 \Phi + \left(\sum_{\alpha} \frac{\rho_{\alpha}^2}{2\lambda_{D\alpha}^2} \nabla_{\perp} \ln N_{\alpha} \right) \cdot \nabla_{\perp} \Phi + \nabla^2 \Phi = -4\pi e \left(\sum_{\alpha} Z_{\alpha} N_{\alpha} - n_e \right) - \sum_{\alpha} \frac{\rho_{\alpha}^2}{2\lambda_{D\alpha}^2} \frac{1}{N_{\alpha} Z_{\alpha} e} \nabla_{\perp}^2 p_{\perp\alpha}$$

- Discretized in ψ - θ coordinates using standard finite differencing
- Field solver uses *Hypre* library of parallel solvers and preconditioners
- Solvers:
 - Conjugate Gradient (CG)
 - Generalized Minimum Residual (GMRES)
 - Stabilized BiConjugate Gradient (BiCGSTAB)
- Preconditioners
 - Diagonal scaling
 - Block Gauss-Seidel with PFMG or SMG in each block
 - BoomerAMG
- Others available

Problems with GK Poisson Equation Solution



B. Cohen and R. Cohen, Sept. 2005

- There is a possible difficulty in solving the fully nonlinear GK Poisson equation in the quasi-neutral limit, *i.e.*, when the ion and electron charge densities are equal to high degree:

$$-\nabla^2 \phi - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi = 4\pi(q\bar{N}_i - en_e)$$

- If the difference between the charge densities is smaller than the errors in N_i and n_e , solving for the potential will be problematic. Can we inhibit the near cancellation?
- A simple iterative approach such as the following does *not* converge:

$$-\nabla^2 \phi_n^j - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi_n^j + 4\pi e^2 n_{n-1} \phi_n^j / T_{e,n} = 4\pi(q\bar{N}_i - en_e)_n + 4\pi e^2 n_{n-1} \phi_n^{j-1} / T_{e,n}$$

Vlasov Equation Implicit Integration Algorithm



- We need to inhibit the $qN_i - en_e$ near cancellation. A good way to do this is to replace by n_e with the following implicit prescription:

$$\text{Predictor } \frac{f^{n+1,p} - f^{n-1}}{2\Delta t} + \vec{v}_{\mathbf{E}\times\mathbf{B}}^n \cdot \nabla_{\perp} f^n + \vec{v}_{\parallel} \cdot \nabla \frac{f^{n+1,p} + f^{n-1}}{2} + \frac{e}{m_e} (\nabla_{\parallel} \frac{\phi^{n+1,p} + \phi^{n-1}}{2}) \frac{\partial f^n}{\partial v_{\parallel}} = 0$$

$$-\nabla^2 \phi^{n+1,p} - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi^{n+1,p} = 4\pi(q\bar{N}_i^{n+1} - en_e^{n+1,p}), n_e^{n+1,p} = \int d^3 v f_e^{n+1,p}(\phi^{n+1,p})$$

Collecting terms in $\phi^{n+1,p}$ and expanding in $k_{\parallel} v_{\parallel} \Delta t < 1$,

$$-\nabla^2 \phi^{n+1,p} - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi^{n+1,p} - 2\omega_{pe}^2 \Delta t^2 \nabla_{\parallel}^2 \phi^{n+1,p} = 4\pi(q\bar{N}^{n+1} - e\tilde{n}_e^{n+1,p})$$

where $\tilde{n}_e^{n+1,p} = \int d^3 v f_e^{n+1,p}(\phi^{n+1,p} \equiv 0)$

- This is the Vlasov equation analog of the direct implicit algorithm for particle codes (Friedman, A., Langdon, A.B., and Cohen, B.I., Comments Plasma Phys. Controlled Fusion, Vol. 6, No. 6 pp. 225-236 (1981))

Vlasov Equation Implicit Integration Algorithm (cont'd)



- Corrector

$$\frac{f^{n+1} - f^n}{\Delta t} + \frac{1}{4} (\vec{v}_{\mathbf{E}\times\mathbf{B}}^{n+1,p} + \vec{v}_{\mathbf{E}\times\mathbf{B}}^n) \cdot \nabla_{\perp} (f^{n+1,p} + f^n) + \vec{v}_{\parallel} \cdot \nabla \frac{f^{n+1} + f^n}{2} + \frac{e}{2m_e} (\nabla_{\parallel} \frac{\phi^{n+1} + \phi^n}{2}) \frac{\partial (f^{n+1,p} + f^n)}{\partial v_{\parallel}} = 0$$

$$-\nabla^2 \phi^{n+1} - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi^{n+1} = 4\pi (q\bar{N}_i^{n+1} - en_e^{n+1}), n_e^{n+1} = \int d^3 v f_e^{n+1}(\phi^{n+1})$$

Collecting terms in $\phi^{n+1,p}$ and expanding in $k_{\parallel} v_{\parallel} \Delta t < 1$,

$$-\nabla^2 \phi^{n+1} - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi^{n+1} - \frac{1}{2} \omega_{pe}^2 \Delta t^2 \nabla_{\parallel}^2 \phi^{n+1} = 4\pi (q\bar{N}_i^{n+1} - e\tilde{n}_e^{n+1})$$

where $\tilde{n}_e^{n+1} = \int d^3 v f_e^{n+1}(\phi^{n+1} \equiv 0)$

- In the 1D limit, parallel to \mathbf{B} , there are ion acoustic waves and electron plasma waves. To resolve the acoustic waves, $\omega_{pe}^2 \Delta t^2 \gg 1$ is desirable.

Vlasov Equation Implicit Integration Algorithm (cont'd)



Simplification -- Boltzmann Electrons

$$-\nabla^2 \phi - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi = 4\pi [q\bar{N}_i - en_o \exp(e\phi/T_e)]$$

Iterative solution:

$$-\nabla^2 \phi^j - (\omega_{pi}^2 / \Omega_i^2) \nabla_{\perp}^2 \phi^j + K^{j-1} \phi^j = 4\pi [q\bar{N}_i - en_o \exp(e\phi^{j-1}/T_e)] + K^{j-1} \phi^{j-1}$$

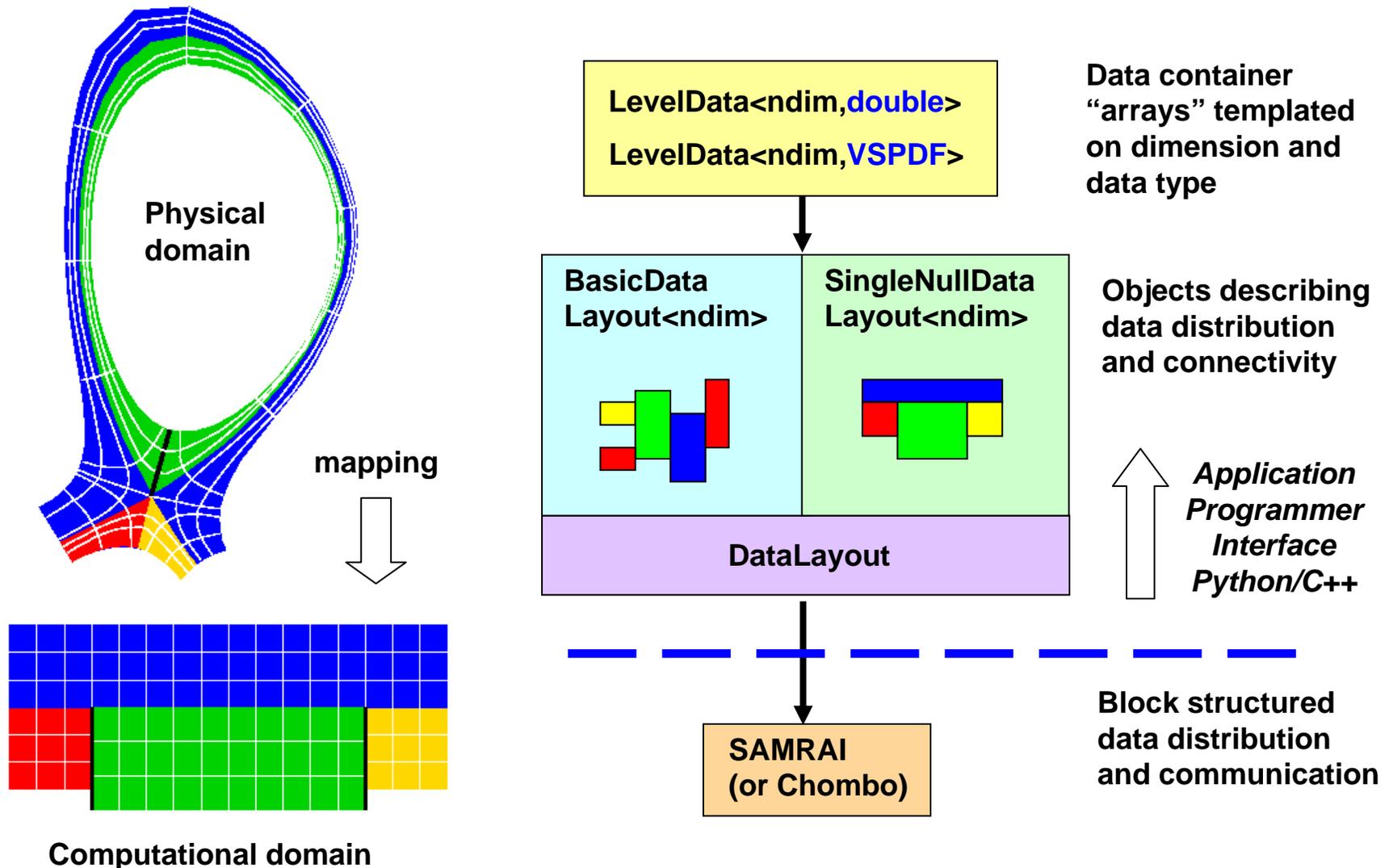
where $K(\mathbf{x}) = 1/\lambda_e^2(\mathbf{x}) = (4\pi n_o e^2 / T_e) \exp(e\phi^{j-1}/T_e)$

and $K = 1/\lambda_e^2 = (1/2)[\max(K(\mathbf{x})) + \min(K(\mathbf{x}))]$

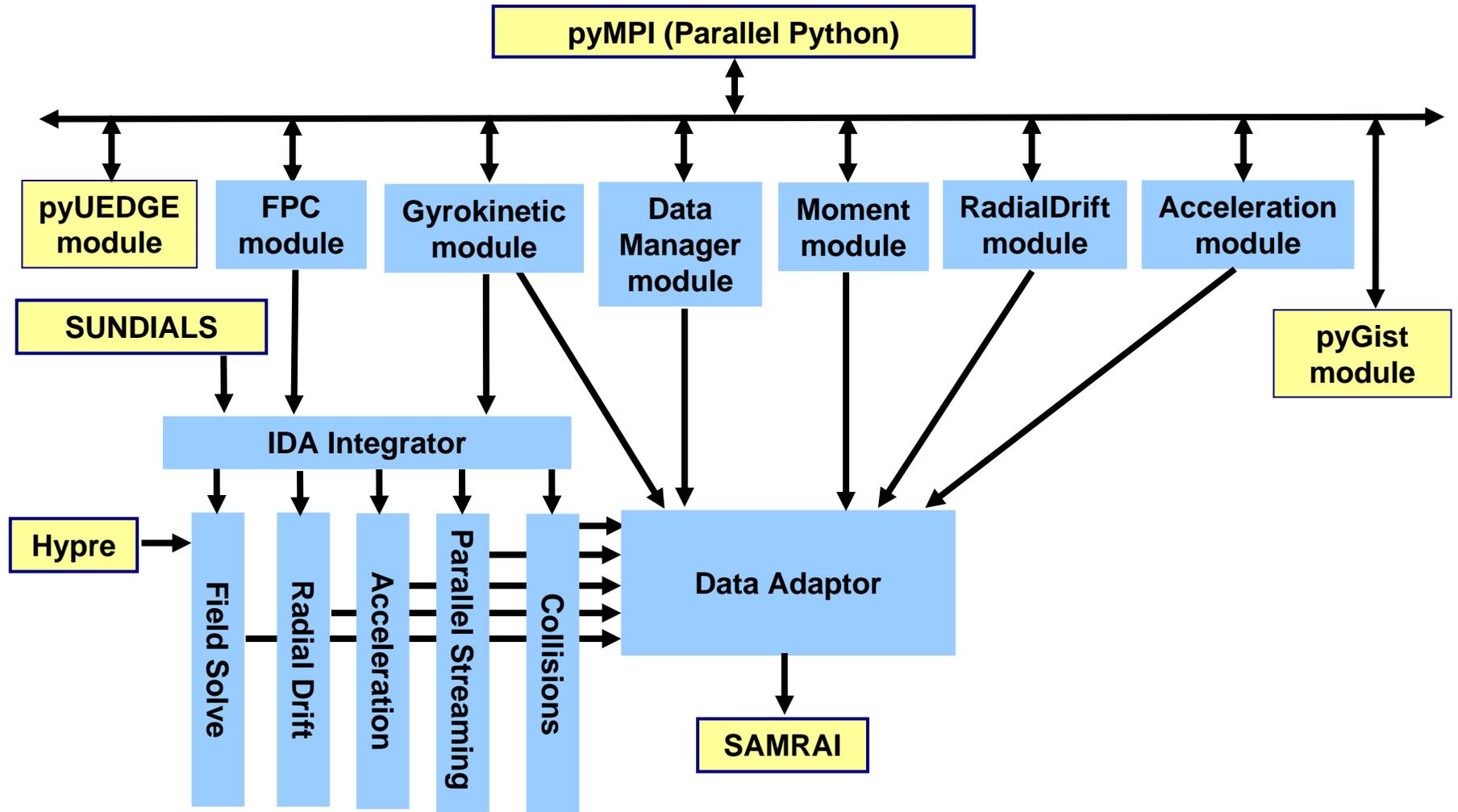
refs.: Cohen, et al., Phys. Plasmas **4**, 956 (1997); Concus and Golub, SIAM J. Num. Anal. **10**, 1103 (1973).

A Newton iteration will also work.

Flexible structures have been developed for mapped, multiblock, locally rectangular data



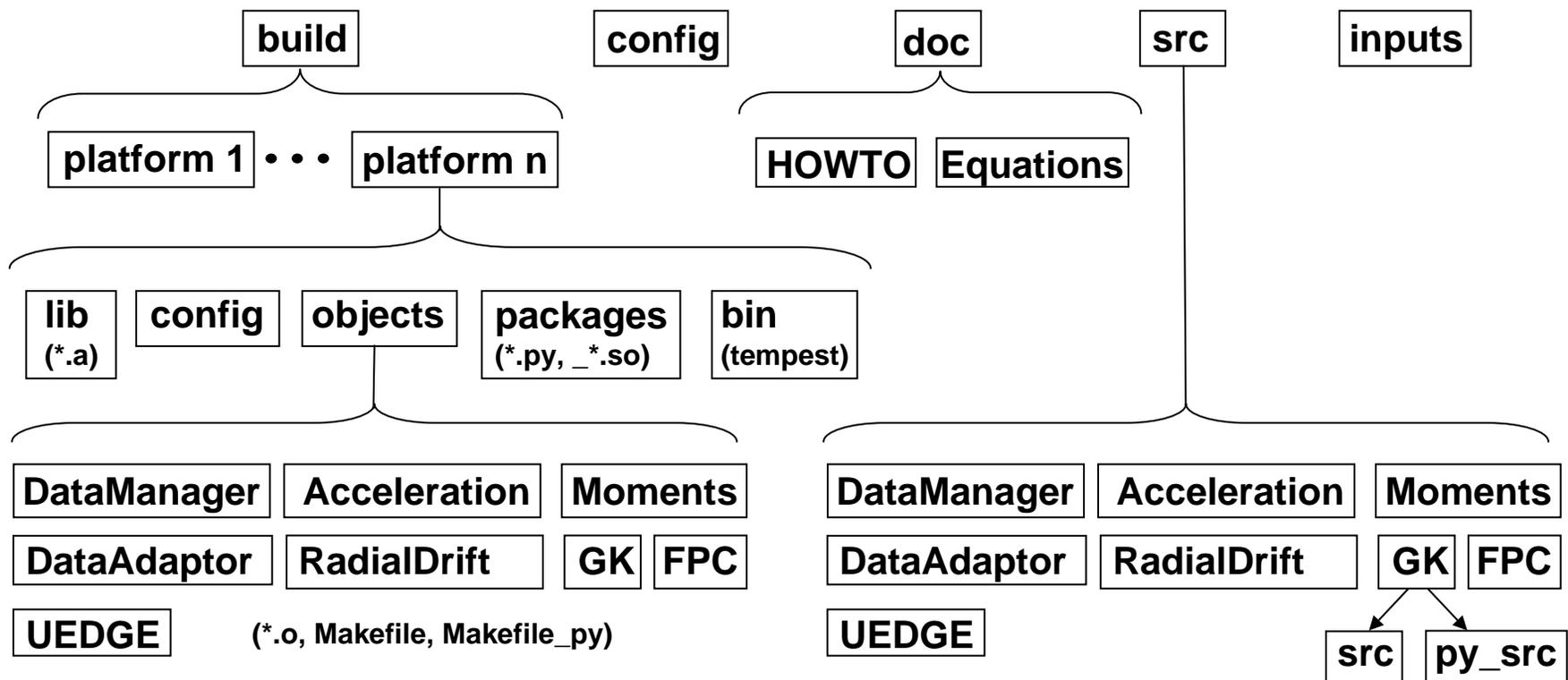
Code schematic



Source code directory structure

base: Third party software (pyMPI, Python, Sundials, SAMRAI, Hypre, etc.)
Each package has its own build system

dev: Obtained by performing cvs checkout pyedge_dev
Build performed by autoconf



A hybrid C++ / F77 approach exploits the strengths of each language

```
BasicDataLayout dl(boxes,mapping);
IntVector<NDIM> nghosts(1);
LevelData<NDIM,double> phi(dl,nghosts);

/* Loop over local patches */
for (DataLayoutIterator<NDIM> p(dl); p; p++)
{
  const Box b(dl.getBoxForPatch(p));
  const Box gb(phi.getGhostBox(p));
  double * d_ptr = phi(p());

  setphi_(b.lower(0), b.lower(1),
          b.upper(0), b.upper(1),
          d_ptr,
          gb.lower(0), gb.lower(1),
          gb.upper(0), gb.upper(1));
}

/* Update ghost cells */
phi.exchange();
```

```
subroutine setphi(lo0, lo1, hi0, hi1,
& phi, glo0, glo1, ghi0, ghi1)
integer lo0, lo1, hi0, hi1,
& glo0, glo1, ghi0, ghi1
double precision phi(glo0:ghi0,glo1:ghi1)
c local variables
integer i, j

do j = lo1, hi1
  do i = lo0, hi0
    phi(i,j) = 3.14d0
  enddo
enddo

return
end
```

Current and near term development activities

- **Redesign of inputs**
- **Restart capability**
- **Conversion to Nd SAMRAI**
 - Update to dimensionally templated classes
 - Replacement of VSPDF class
- **More documentation**

Running Tempest

- From the command line:

```
jafh@grendel> cd inputs
jafh@grendel> mpirun ../build/grendel-icc-ifort-dbg/bin/tempest
*****
* LLNL Edge Plasma Framework *
*****

Type "help", "copyright", "credits" or "license" for more information.
>>> execfile('input.py')
```



Running Tempest

- From the command line:

```
jafh@grendel> cd inputs
jafh@grendel> mpirun ../build/grendel-icc-ifort-dbg/bin/tempest
*****
* LLNL Edge Plasma Framework *
*****

Type "help", "copyright", "credits" or "license" for more information.
>>> execfile('input.py')
```

- `tempest` executes `pyMPI` in a suitably configured environment



Running Tempest

- From the command line:

```
jafh@grendel> cd inputs
jafh@grendel> mpirun ../build/grendel-icc-ifort-dbg/bin/tempest
*****
* LLNL Edge Plasma Framework *
*****

Type "help", "copyright", "credits" or "license" for more information.
>>> execfile('input.py')
```

- `tempest` executes `pyMPI` in a suitably configured environment
- Python interpreter then *executes* the input script



Running Tempest

- From the command line:

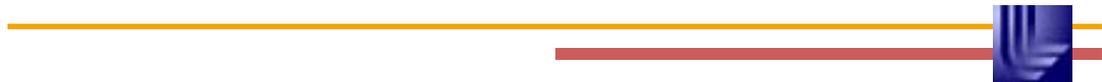
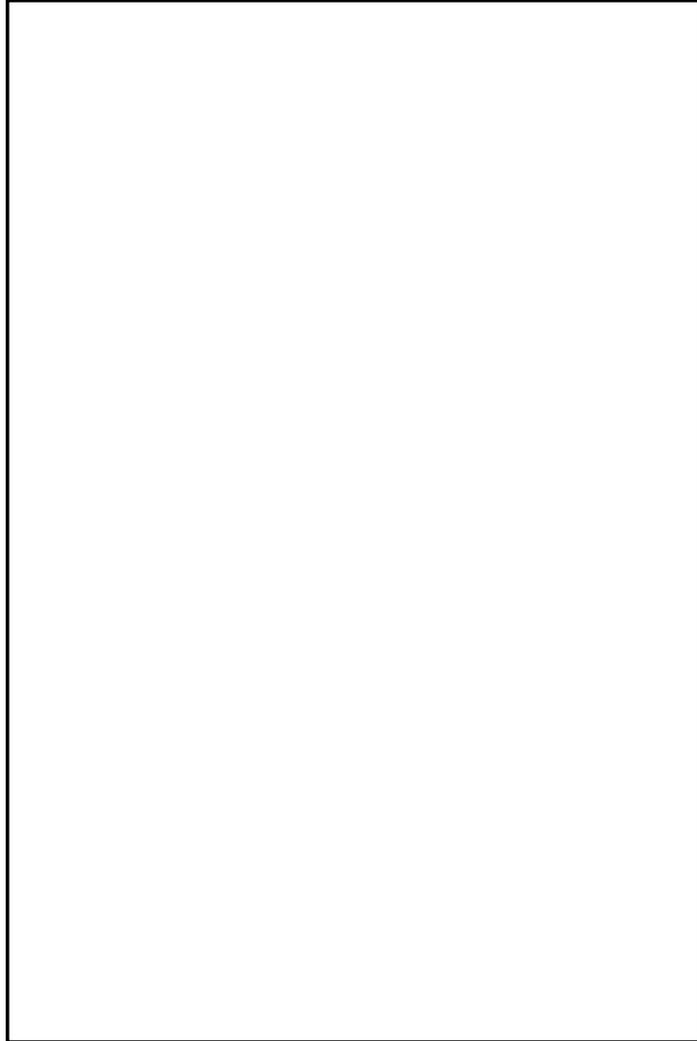
```
jafh@grendel> cd inputs
jafh@grendel> mpirun ../build/grendel-icc-ifort-dbg/bin/tempest
*****
* LLNL Edge Plasma Framework *
*****

Type "help", "copyright", "credits" or "license" for more information.
>>> execfile('input.py')
```

- `tempest` executes `pyMPI` in a suitably configured environment
- Python interpreter then *executes* the input script
- Scripted interface allows for *rapid development* of pre- and post-processing code, e.g., initial conditions



Input File Structure



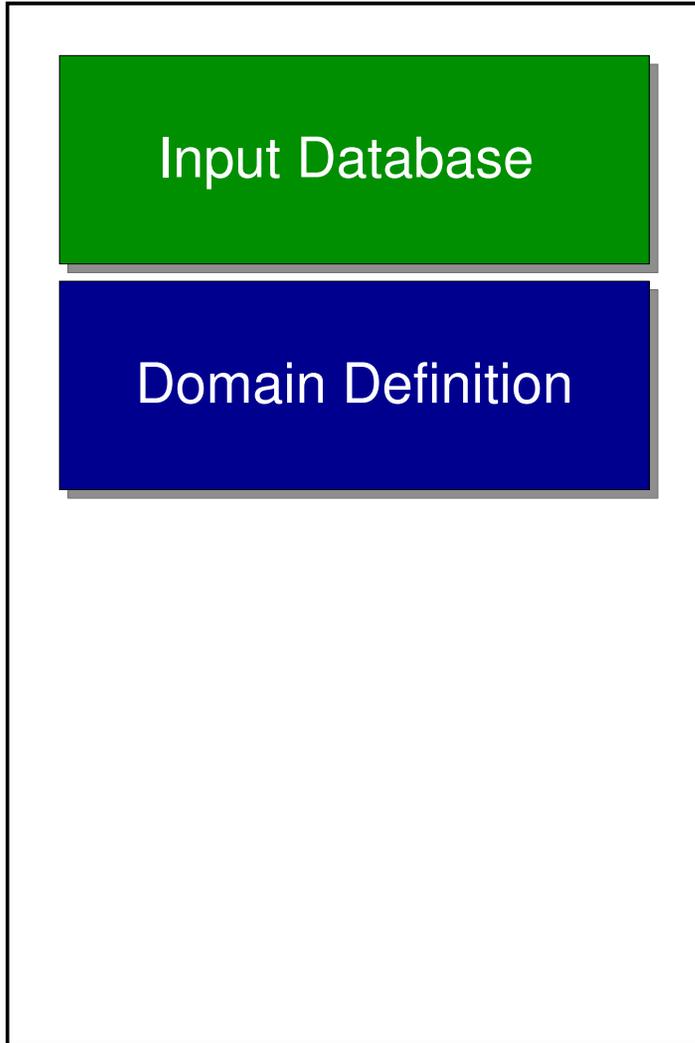
Input File Structure



Define input parameters for modules



Input File Structure

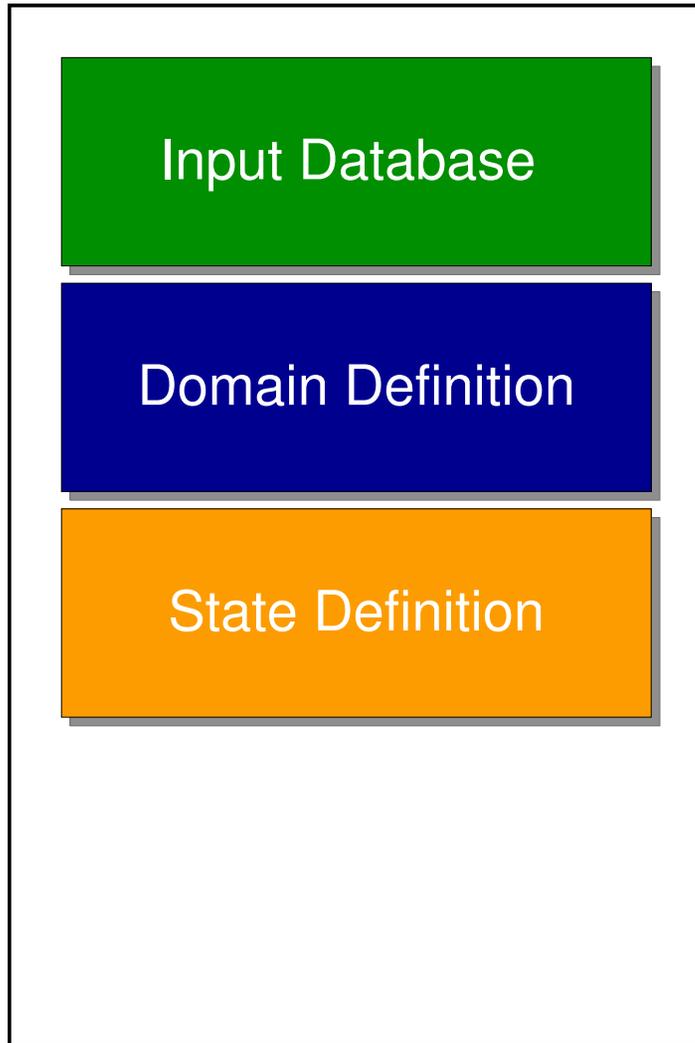


Define input parameters for modules

Define physical geometry & index spaces



Input File Structure



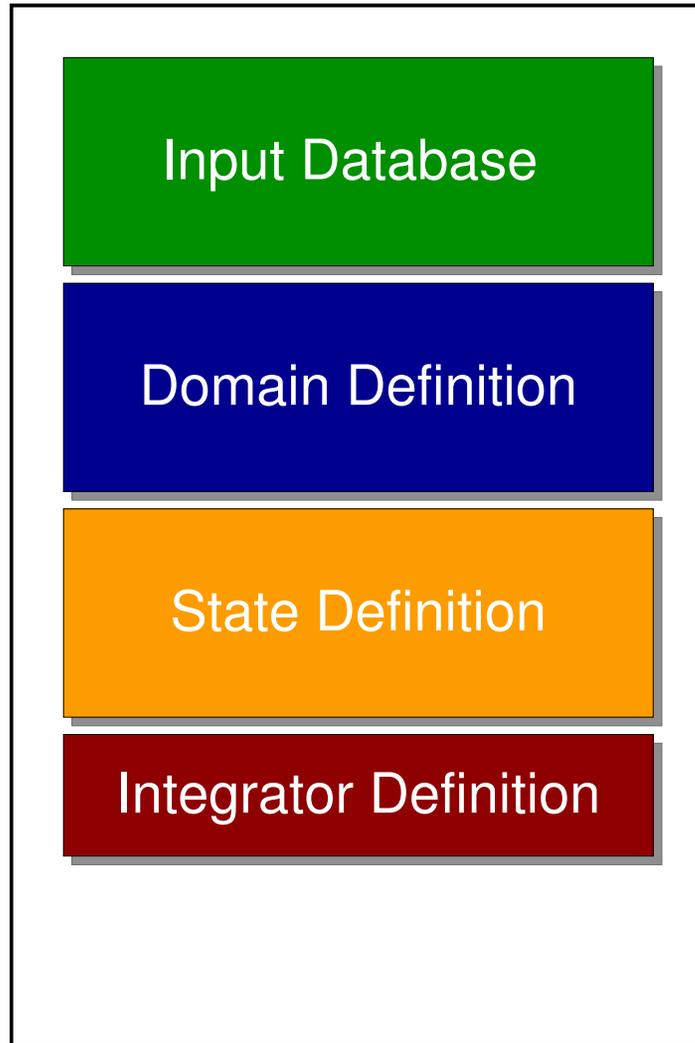
Define input parameters for modules

Define physical geometry & index spaces

Define and initialize variables



Input File Structure



Define input parameters for modules

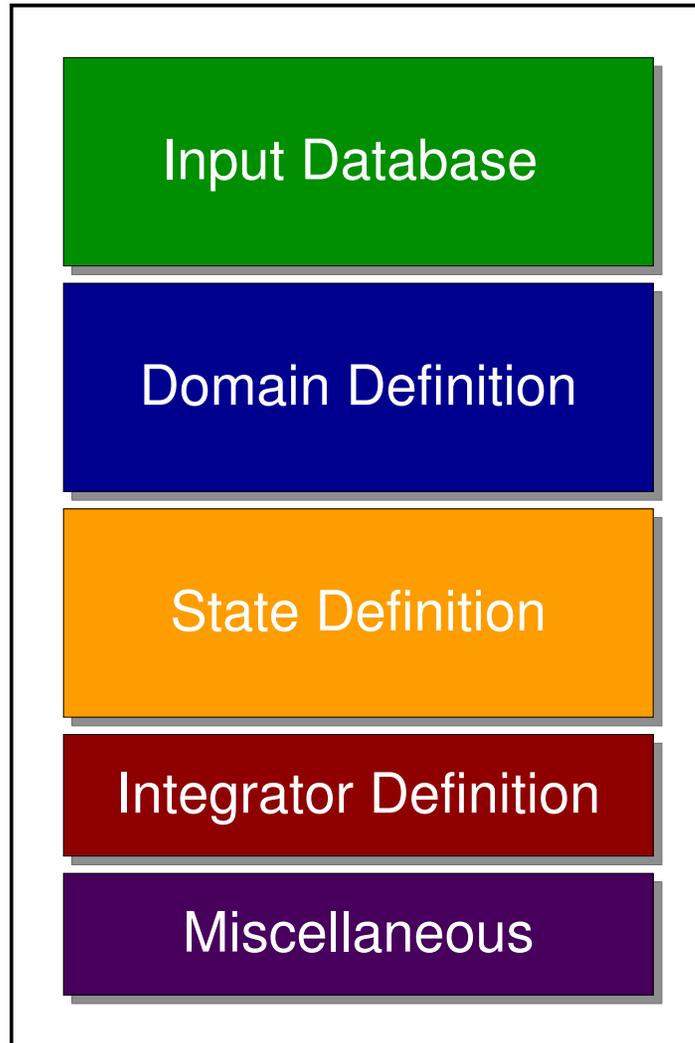
Define physical geometry & index spaces

Define and initialize variables

Define and initialize integrator



Input File Structure



Define input parameters for modules

Define physical geometry & index spaces

Define and initialize variables

Define and initialize integrator

Define auxiliary functions, such as `run ()` function and plotting routines



TEMPEST V&V: ENDLOSS TESTS

Presented by: R.H. Cohen



ESL WORKSHOP

Nov. 30, 2005

OUTLINE

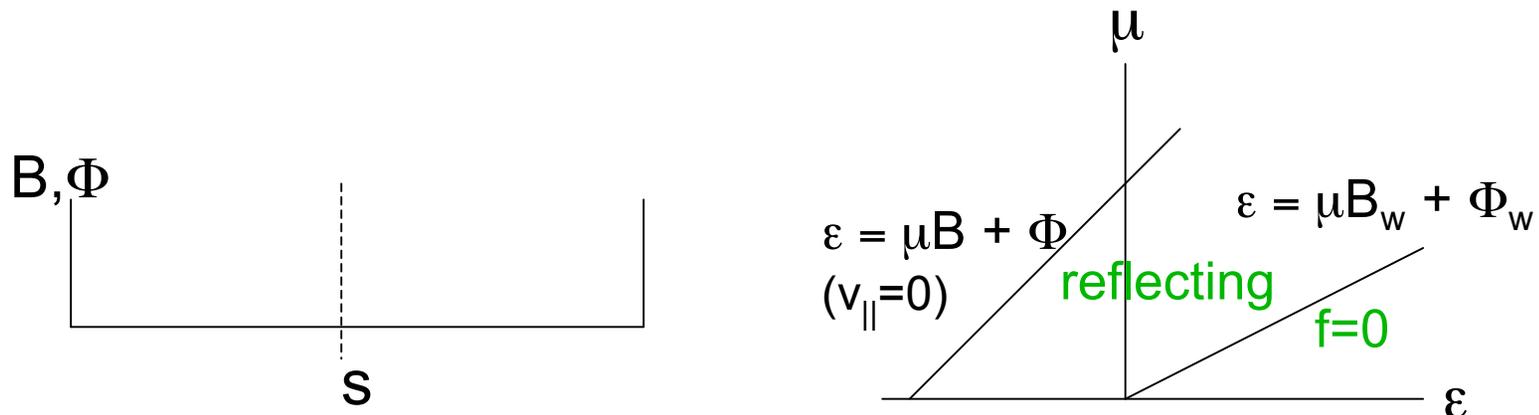


- Problem definition
- Basic stuff: collisionless tests
- Collisional results with the FPC package in code
 - Conclusions: results dominated by non-conservation of package
- Experiments with “quick fixes”
- Summary of solution (details in Kerbel’s presentation)

Problem definition



- Uniform B and Φ with confining mirror ratio and potential jump at wall
- Directly relevant to electrons in SOL; indirectly to ion “x-point loss problem”.
- Ingredients: collisions, streaming moments



- Analytic theory exists for loss rates in two limits:
 - Weak collisions (“Pastukhov” -- empty loss cone; bounce average)
 - Flow-confinement regime: $\lambda_{mfp} < L$; $\lambda_{mfp} R e^{e\Phi/T} > L$ (loss cone fills; parallel diffusion; but loss rate small enough that density flat.) (For higher density, solve diffusion eq.)

Testing/benchmarking (cont)



- Pastukhov limit: many papers; various approximations. All are formally expansions in $\exp(-e\Phi/T)$. All are of form:

$$\tau_p \sim \tau_0 \varphi \ln RH(R, \varphi) \exp(\varphi)$$

with $\varphi = e\Phi/T$, $\tau_0 = m^{1/2}(2T)^{3/2}/(8\pi e^4 n \ln \Lambda)$.

- Results here: Najmabadi et al, NF24, 75 (1984).
- Flow limit: loss cone fills in faster than transit

$$\tau_{fl} = 2\pi^{1/2} R L \exp(e\Phi/T) / v_{th}$$

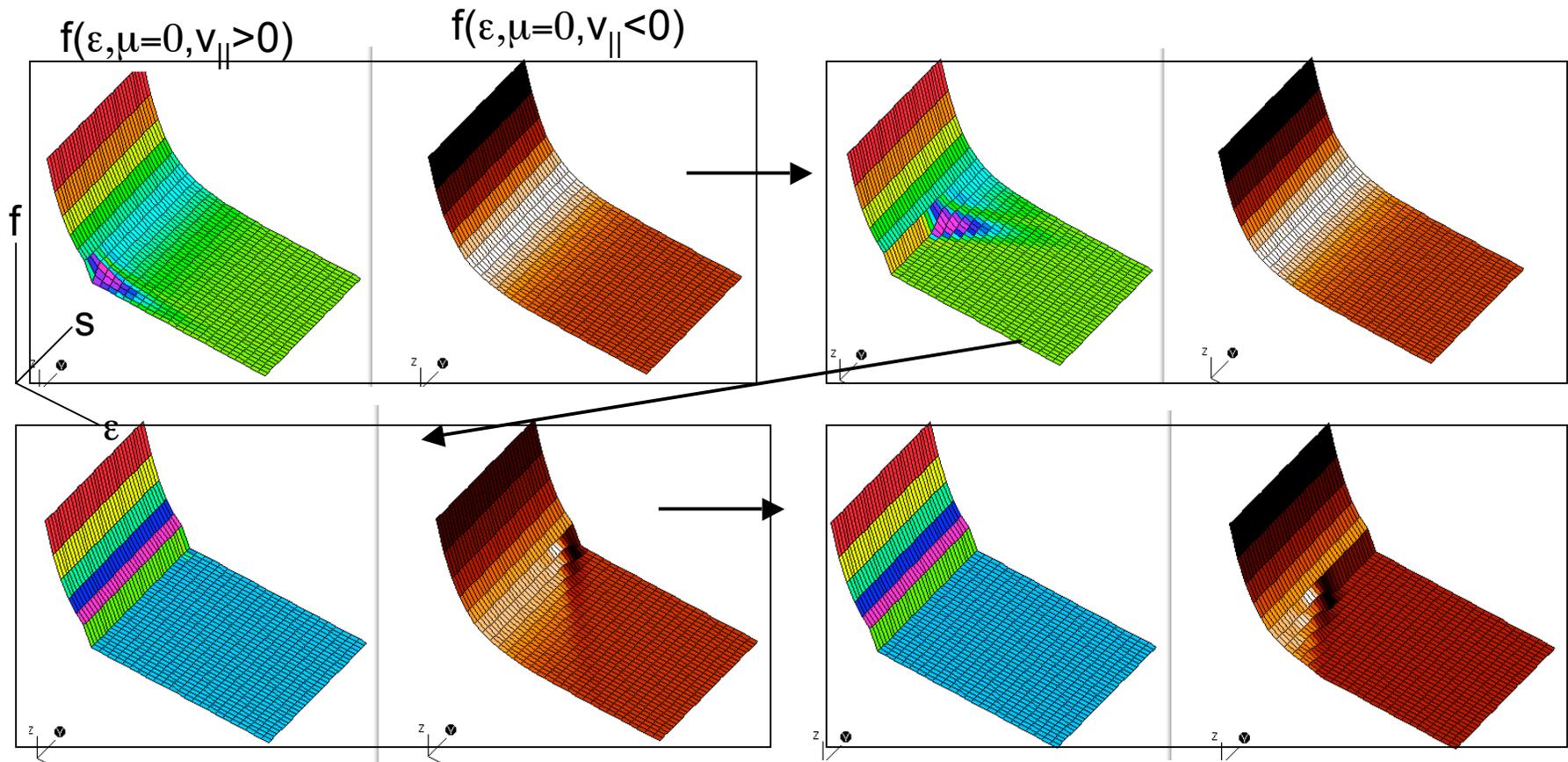
- Interpolation [Rognlien & Cutler NF 20, 1003 (1980), Cohen NF 19, 1295 (1979)]:

$$\tau = \tau_p + \tau_{fl}$$

Code passes basic collisionless test



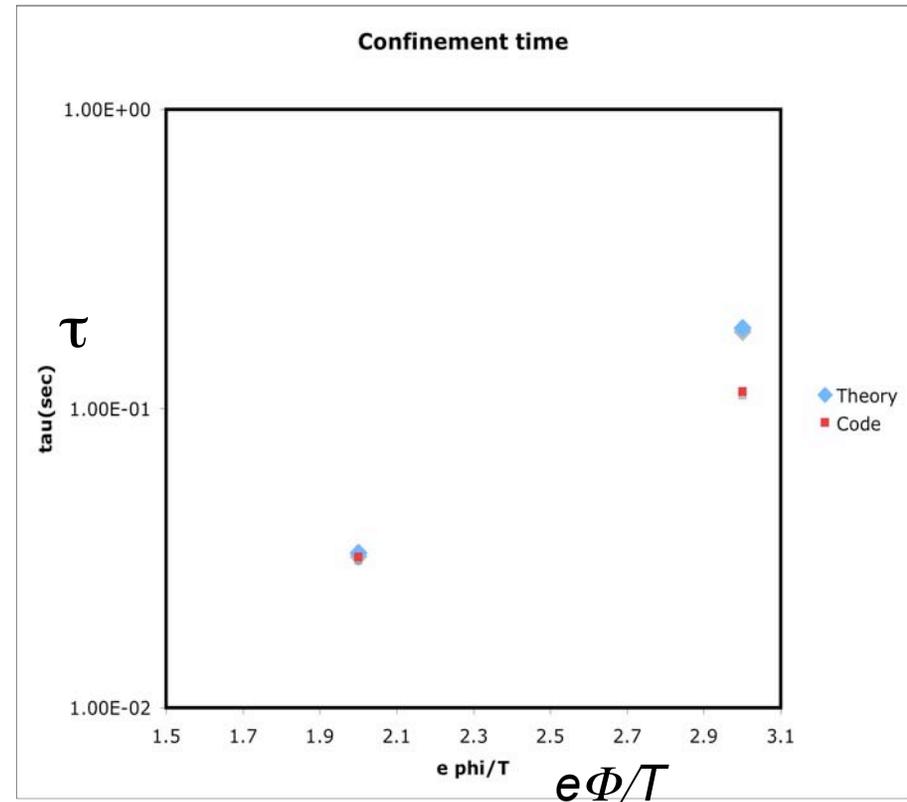
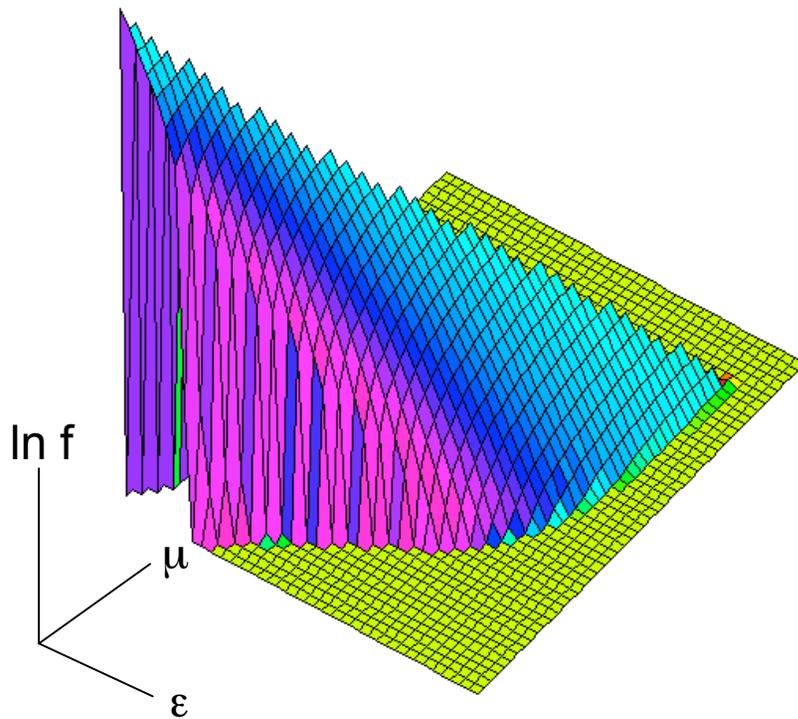
- Start with full Maxwellian, track propagation of emptied loss cone across sytem and accompanying density decrease.
 - Results: quantitatively correct propagation speed, density decrease.
 - This checks: streaming, moments, open field line b.c.'s



Tests with FPC package: good and bad news



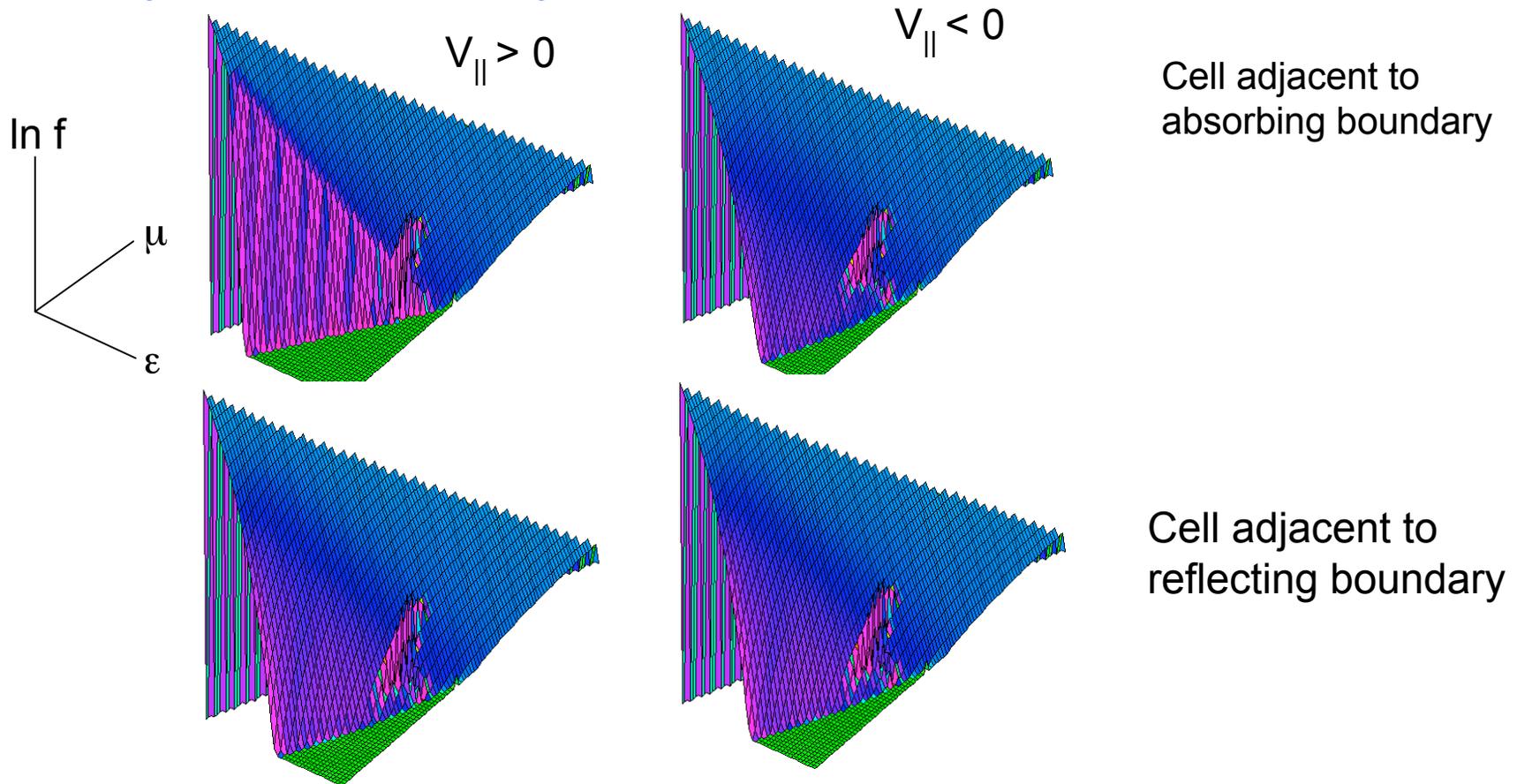
- Pastukhov limit:
 - density scaling nearly perfect but potential scaling not



Endloss: flow limit



- Confinement time almost independent of n as in theory; good Φ scaling. Distinctive qualitative features of f correct... loss cone emptied out adjacent to wall; $f^{+,-}$ symmetric far away.



Bad news, and explanation



- Confinement times are too short, by a factor varying from $O(1)$ to > 10 depending on Φ/T , density, FPC interpolation scheme (but only weakly on resolution)
- Significant numerical particle non-conservation associated with collision-operator implementation.
- Difference between theoretical and observed loss rate is closely matched by $\int (df/dt)_c$.

Experiments with “quick fixes”



- Crook correction to collision operator to cancel out numerical loss

$$\left(\frac{df}{dt}\right)_c \rightarrow \left(\frac{df}{dt}\right)_c - \frac{1}{n} \left(\frac{dn}{dt}\right)_c f$$

- Results in confinement times too *long* compared to theory (by factors of 2-10)
- Hybrid collision operator: use FPC package for energy scatter only; take pitch-angle scattering from new ϵ, μ coding
 - Results unsubstancially different from full FPC package
 - Suggests that problems are related to the energy scattering part of FPC implementation

Summary of solution



- Chosen option: Write fluxes directly in ε, μ variables.
 - Can easily deploy low-order explicitly conservative or high-order non-explicitly conservative differencing schemes.
 - Express ε, μ fluxes in terms of coefficients times $\partial f / \partial \varepsilon, \partial f / \partial \mu$
 - Interpolate coefficients from FPC (v, θ) or analytic forms
- Other possible solutions discussed
 - Change GK coordinates -- rely on advanced numerics instead of E, μ to preserve f under streaming
 - Basis-function versus finite-difference representation of E, μ
 - (Find a bug in current coding)

Gyroaveraging in TEMPEST



- The guiding center distribution function must be mapped to the particle distribution around the gyro-orbit for purposes of calculating moments, *e.g.*, charge and current densities, fluid velocities, pressures. We need a pullback transformation:

$$f_{gc}(\vec{X}_{gc}; E, \mu, \sigma_{\pm}) \rightarrow f(\vec{X}_{gc} + \rho \hat{e}_i, \vec{V}_{gc} + d\vec{\rho}_i/dt)$$

- For the pullback we can take advantage of the time integration of the advection operator already developed (suggested by G. Kerbel):

$$(\partial/\partial t + (\rho/\Delta t)\hat{e}_i \cdot \nabla)f_i = 0$$

- A four-point gyro-average would consist of

$$\langle f \rangle = (f_1 + f_2 + f_3 + f_4)/4$$

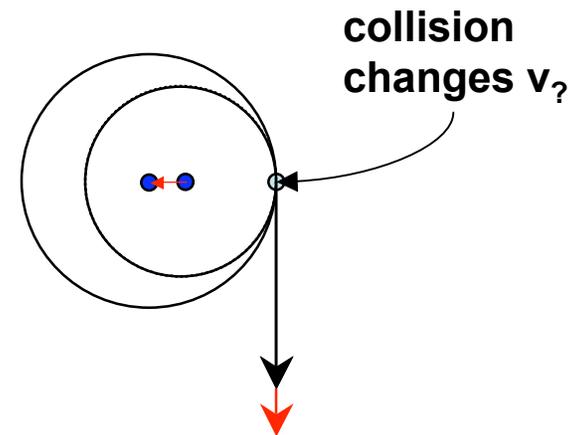
$$\hat{e}_{1,2} = \pm \hat{b} \times \hat{e}_{\nabla\psi}, \hat{e}_{3,4} = \pm \hat{e}_{\nabla\psi}$$

- 3D Fokker-Planck collisions including classical guiding-center diffusion can be calculated by applying the F-P operator to f_{particle} at the particle gyro positions and averaging back to f_{gc} if the particles *locally* share a common guiding center polarization displacement and drift.



Implementing 3D Gyrokinetic, Nonlinear, Fokker-Planck Collisions - G. Kerbel

Finite gyro-orbit size:
collisional changes in v_z
at fixed gyro-position
shift the gyro-center



- To get **3D gyro-kinetic collisions**, transform gyro-center f to gyro-position f on the *same* spatial mesh, calculate collisions, then transform back and gyro-average.
- This has **never been done before** in a gyrokinetic code and is well underway.

Extending TEMPEST to 5D



- The toroidal angle ξ labels the field line (but not distance along the field line, which is labeled by θ , except at an X-point) and is an ignorable coordinate in an axisymmetric equilibrium (but not for the fluctuations or for stellarators).
- We have options on how we accommodate ξ
 - Use Fourier decomposition and a spectral approach (as in GYRO)
 - Use a mix of Fourier decomposition and finite-differencing or pseudo-spectral methods on a grid (as in BOUT or PG3EQ)
- Xu is adding the toroidal coordinate and additional terms to TEMPEST. The starting point is a fully 3D formulation of the GK + field equations, which we have. There are no major subtleties (except perhaps at an X pt).

MOVING FORWARD

(Some slides to start discussion)

R. Cohen, Moderator

ESL Workshop

Nov. 30, 2005

UCRL-PRES-216293

Work performed for U.S. DOE by UC LLNL under contract W7405-ENG-48

R. Cohen ESL-Intro Nov 30, 2005 -1-

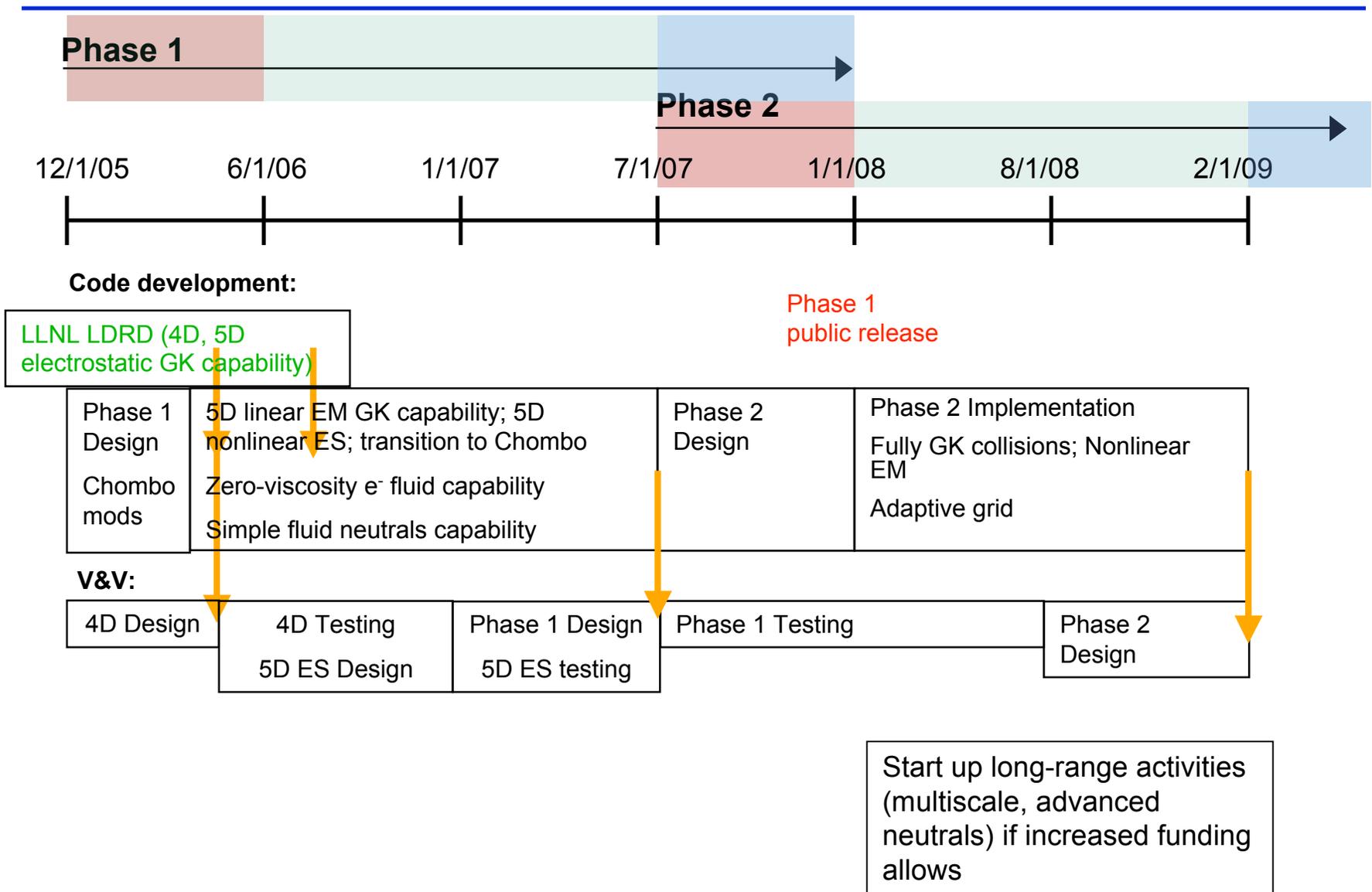
OPTIONS/FUNDING LEVELS

- In Sept. we responded to OFES request for statement of project scope at 3 funding levels: 550K, 750K, 950K from OFES with matching funds from OASCR.
 - These are presented in following slides
 - Now, \$550K × 2 is a likely upper limit for FY06
 - If we get this (or greater), proceed as planned.
 - If we have little or no OASCR funding, need a different plan.

Continuum edge code project: 3-year plans at several budget levels

- At \$1.1M (550K each OFES, OASCR)
 - Similar to stretched version of June 05 plan
 - Kinetic emphasis; turbulence emphasis; Simakov-Catto and hybrid deferred; simple neutrals; no multiscale work; research-quality code.
- At 1.5M (\$750K each office)
 - \$1.1M plan unstretched, plus Catto-Simakov neutrals in phase 1, increased V&V; hybrid in phase 2.
 - Some application to/V&V for ELMs
- At 1.9M (\$950K each office)
 - Adds multiscale, increased V&V, production-quality software; analysis and viz. tools
- Original proposal was for \$2M in year 1 and increases beyond that. Lost in all cases is kinetic neutrals models, some V&V, and at least partial loss of grid alignment adaptivity

\$1.1M/year plan



A modest proposal for what to do

- If no (or little) additional funding in ESL or LDRD:
 - Two-pronged effort within current computational framework (SAMRAI) to:
 - Develop a more complete 4D kinetic code (LDRD focus?)
 - Develop electrostatic 5D GK code (ESL focus?)
- With additional funding, decide what of original plan to restore, based on amount and source of added funds

Near-term application target (my personal observations)

- Likely '06 funding levels all below FSP proposal level, and we are no longer bound by FSP Call for Proposals.
 - ELMs, which played heavily in FSP proposal, should be largely deferred
 - Pedestal structure and edge turbulence are reasonable target applications
 - Is this a focussed-enough goal?