PETSc Solvers in Gyrokinetic Particle-in-Cell Methods for Tokamak Edge Plasmas

Mark Adams (Lawrence Berkely National Laboratory)
Seung-Hoe Ku (Princeton Plasma Physics Laboratory)
Robert Hager (Princeton Plasma Physics Laboratory)

and the EPSI team

Celebrating 20 Years with PETSc
XGC1 – Gyrokinetic particle-in-cell (PIC) code

- Gyrokinetic particle-in-cell (PIC) code designed for simulating edge plasmas in tokamaks
- Solves 5D gyrokinetic equations via
  - Ordinary differential equations for time advance of particles
  - Maxwell’s equation on unstructured triangular physical space grid, solved using PETSc for electro-static/electro-magnetic turbulence
- Multiscale physics (but full field)
  - Micro-turbulence
  - Background profile variation
  - Neutral and atomic physics

Celebrating 20 Years with PETSc
XGC1 – extreme scale (SciDAC partnership program)

- Ions and electrons: \( \sim O(10^{10} - 10^{11}) \) particles each
- Unstructured triangle mesh in 3D: \( \sim O(10^6 - 10^7) \) mesh points
- Fully non-linear Coulomb collisions
- Designed for leading HPC: Titan(OLCF), Edison(NERSC), Mira(ALCF)
  - Good performance scaling
  - Utilizing GPUs

XGC1 Performance: Weak Particle Scaling on DIII-D grid

Celebrating 20 Years with PETSc
Outline

- 5D Gyrokinetic Vlasov Equations (lots of hand waving)
- Poission Equation with adiabatic electrons
- I) (new) Gyrokinetic Poisson solver flux surface average
- II) (new) Hybrid kinetic-ion & fluid electron: implicit MHD
- Future directions XGC:
  - better numerics – tightly couple time integrators in PETSc
  - data centric processing for modern architectures
6D Vlasov equation: basically the whole world

- 6D Vlasov-Maxwell system

\[
\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} + \frac{\vec{F}}{m} \cdot \frac{\partial f}{\partial \vec{v}} = C
\]

\[
f = f(\vec{x}, \vec{v})
\]

- 6D PDE (phase space: 3D real space, 3D \(v\)-space)
- \(F\) : Lorenz Force \(\leftrightarrow\) Maxwell equations
- Collisions and source term on RHS
Gyrokinetics

Real particle motion
Lorentz Equation

Gyro Center Motion
(Gyrokinetics)

\[ \begin{align*}
\text{Point particle with charge } q \\
\text{Ring with radius } \rho \\
\text{magnetic moment } \mu
\end{align*} \]

- Ring with gyroradius ignoring gyro-phase
  - 6D \to 5D

Celebrating 20 Years with PETSc
XGC discretization

- Have somewhat complex geometry
- Extreme anisotropy from strong B field
- This has led to discretizations that split “perp” and “parallel” fields.
- Unstructured grids constructed with ODE solvers to make grid points follow field lines
  - Complex and approximate
  - Finite elements in perp plane
- Finite difference in parallel direction
- 64-128 toroidal planes.
5D Gyrokinetic Equations - Electrostatic

\[ f = f(\vec{X}, u_\parallel, u_\perp); \quad X: \text{guide center position} \]

\[
\frac{\partial f}{\partial t} + \dot{X} \cdot \frac{\partial f}{\partial X} + \dot{u}_\parallel \frac{\partial f}{\partial u_\parallel} = C
\]

\[
\dot{X} = \frac{1}{D} \left[ u_\parallel \vec{b} + \frac{mcu_\parallel^2}{qB} \nabla \times \vec{b} + \frac{c}{qB^2} \left\{ \vec{B} \times \left( \mu \nabla B - qE \right) \right\} \right]
\]

\[
\dot{u}_\parallel = -\frac{1}{mD} \left( \vec{b} + \frac{mcu_\parallel}{qB} \nabla \times \vec{b} \right) \cdot \left( \mu \nabla B + qE \right)
\]

Gyrokinetic Poisson Equation

\[
-\nabla_\perp \frac{mn_0}{eB^2} \nabla_\perp \Phi = n_i - n_e
\]

\[ D = 1 + \frac{mcu_\parallel}{qB} \vec{b} \cdot \nabla \times \vec{b} \]

\[ \vec{b} = \vec{B} / B \]

\[ \mu = \frac{mu_\perp^2}{2B} \]
5D Gyrokinetic Equations: simplified

- Particle motions (ODEs)

\[
\dot{x} = v_\parallel + v_D
\]

\[
\dot{v}_\parallel = E_\parallel + \mu \nabla B
\]

- $x$: 3D space coordinates
- $v_\parallel$: Parallel velocity to magnetic field
- $B$: Magnetic field strength
- $E_\parallel$: Parallel electric field
- $\mu$: Magnetic moment
- $n_e^0$: Electron density with zero potential
Adiabatic Electron Response

\[ n_e = n_e^0 \exp\left(\frac{-\Phi}{T(r)}\right) \]

- Electron density near Boltzmann F magnetic field line:
  - Parallel motion is dominating: \( v||/v_D > 10^4 \)
  - Travel time is shorter than wave time scale
- Delta F method: \( f = f_0 + \delta f \)

\[ \dot{X} = v|| + v_D \]

Dominant motion

Small drift

Electron guiding center

Magnetic field
Poisson solvers in XGC1

- Gyrokinetic Poisson Equation with adiabatic electrons
  - Old method, MG useful
  \[ -\nabla_{\perp} \frac{m n_0}{e B^2} \nabla_{\perp} \Phi + \Phi = n_i \]
- Real electrons
  - Need multigrid
  - Stable production solver
  \[ -\nabla_{\perp} \frac{m n_0}{e B^2} \nabla_{\perp} \Phi = n_i - n_e \]
- Two new solvers under development
  1. Flux surface electron equilibrium model (FSA)
     - More accurate electron model, electro-static
  2. Hybrid kinetic ions + fluid electrons
     - Faster in theory: implicit MHD, skip fast Alvene wave
     - Electro-magnetic
     - Future: fully kinetic electrons & ions; electro-magnetic
I. Perturbative calculation of electron density

- Maxwell distribution on flux surface, particles for $\delta f$
  - $f_0$ is of form, with $K$ kinetic energy:
  - $\langle f \rangle$ flux surface average (FSA)
- XGC1 calculates perturbation from Boltzmann density (i.e. adiabatic electron response) along field line, which equilibrate to flux surface

$$f_0 = C \exp \left( -\frac{K + e(\Phi - \langle \Phi \rangle)}{T} \right)$$

- $\Phi - \langle \Phi \rangle$ is potential variation along field line/surface
- Poisson equation:
  $$-\Delta \Phi + n_e^0 \exp \left( \frac{\Phi - \langle \Phi \rangle}{T} \right) = n_i - \delta n_e$$
- Poisson equation with linearization of exponential:
  $$-\Delta \Phi + n_e^0 \frac{\Phi - \langle \Phi \rangle}{T} = n_i - n_e^0 - \delta n_e$$
FSA accurate solver

- Add auxiliary variable for $\langle \Phi \rangle$
  - FieldSplit
- Create linearization for preconditioner matrix
  - Linearizing about phi=0
- Use matrix free operator for nonlinear version of solver
  - Preconditioned by linearized matrix
- Use PETSc FieldSplit and MatNest object
- Problem: $\langle \Phi \rangle$ is global
  - small number $\sim O(10^2)$ and linear
- Approach: compute explicit Schur compliment
  - Block factorization preconditioner, non-iterative
  - Total solve time $\sim 2 \times$ Laplacian solve time
  - But large setup cost that needs amortizing

\[
\begin{pmatrix}
\Delta + \left( n_e^0 / T_e \right) & -n_e^0 / T_e B_l \\
C_{\text{ave}} & -I
\end{pmatrix}
\begin{pmatrix}
\Phi \\
\langle \Phi \rangle
\end{pmatrix}
= \begin{pmatrix}
n_i - n_e^0 - \delta n_e \\
0
\end{pmatrix}
\]

Celebrating 20 Years with PETSc
Old Iterative solver vs Field Split

Solution from iterative solver after 10,000 iteration: 384 sec

Solution from field split solver: 0.086 sec

Celebrating 20 Years with PETSc
2. Hybrid fluid (MHD) electrons & kinetic ions

- Second branch (EM), less expensive, no kinetic electrons
- Evolve electron density with fluid formulations (cheaper)

\[ F(U, \dot{U}) = \begin{bmatrix} Q_1^s & Q_0^f + \overline{Q}_2^g & Q_3^s \\ 0 & -\frac{\eta}{\mu} \nabla_\perp^2 & \nabla_\parallel \\ I & 0 & -\frac{n_0 m_i}{B^2} \nabla_\perp^2 \end{bmatrix} \begin{bmatrix} n_1 \\ A \\ \phi \end{bmatrix} + \begin{bmatrix} \partial_t \begin{bmatrix} n_1 \\ A \\ \phi \end{bmatrix} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = 0 \begin{bmatrix} -C_1 \\ 0 \\ \delta n_i \end{bmatrix} = G(U) + C \]

\[ Q_0^f = B_0 \nabla_\parallel \frac{1}{e\mu B_0} \nabla_\perp^2 ; \quad Q_1^s = \frac{2}{m_e \Omega_e B_0^2} \overline{B}_0 \times \nabla B_0 \cdot \nabla T_0 \]

\[ \overline{Q}_2^g = \nabla \frac{j_0}{eB_0} \cdot \nabla \times \frac{\overline{B}_0}{B_0} ; \quad Q_3^s = \frac{2n_0}{B_0^3} B_0 \times \nabla B_0 \cdot \nabla + \nabla n_0 \cdot \left( \frac{B_0}{B_0^2} \times \nabla \right) \]

\[ C_1 = B_0 \nabla_\parallel \frac{n_0 u_i}{B_0} \]
Hybrid fluid (MHD) electrons & kinetic ions

- TS with IMEX solve (some slow terms moved to RHS)
- 2 auxiliary equations (for potential and current)
  - Verify w/ reduced prob: fast wave & slow growth mode
  - Fully implicit

\[
\begin{bmatrix}
0 & 0 & 0 & B_0 \nabla_\parallel \frac{1}{eB_0} \\
0 & 0 & \nabla_\parallel & 0 \\
M & 0 & -\frac{n_0 m_i}{B^2} \nabla_\perp^2 & 0 \\
0 & -\nabla_\perp^2 & 0 & \mu M
\end{bmatrix}
\begin{bmatrix}
n_1 \\
A \\
\phi \\
J
\end{bmatrix}
\begin{bmatrix}
\frac{\partial}{\partial t} n_1 \\
\frac{\partial}{\partial t} A \\
\frac{\partial}{\partial t} \phi \\
0
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
0 \\
0
\end{bmatrix}
\]
Future directions for XGC: numeric's & processing

- XGC is 10+ years old & established as flagship extreme scale Tokamak code
- Numeric's & software engineering need attention
- PIC processing for shift to data movement centric costs
- Started to develop PETScified XGC code – XGC2
  - XGC1 physics and basic numerical methods (PIC)
  - Streaming processing with loop fusion
  - Built on new hybrid structured/unstructured grid/discretization solvers (Toby Isaac, tomorrow)
- Particles with basic Tokamak dynamics
- FE solver grid decoupled from particle decomposition
- Pericles with red particle near outside flux surface
Future directions for XGC – use modern solvers

- Tightly couple time integrators in PETSc (IMEX)
  - Long term: need to put all code in PETSc
- Sketch of simple PIC, electro-static, one species

\[
\begin{bmatrix}
0 & -I \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
0 & 0 & 0 \\
0 & 0 & \frac{q}{m} \int & 0
\end{bmatrix}
\begin{bmatrix}
\begin{bmatrix} P_x \\ P_v \end{bmatrix} \\
\begin{bmatrix} \phi \\ \phi \end{bmatrix} \\
\begin{bmatrix} E \\ \rho \end{bmatrix}
\end{bmatrix}
\]

\[+
\begin{bmatrix}
\frac{\partial}{\partial t} & 0 \\
0 & \frac{\partial}{\partial t}
\end{bmatrix}
\begin{bmatrix}
P_x \\
P_v
\end{bmatrix} = S
\]
Distributed field – XGC redundantly stores fields!
- With about 10,000 particles/cell this is doable
- Think of PIC processing from the ground up
- Algorithm, use RK2 and at each stage
  - For all particles, deposit charge for Poisson solve
  - For all particles, deposit current for Ampere's law solve
  - For all particles, deposit density on velocity space grid for collision operator
  - Solve Poisson for potential
  - Solve Ampere's equation for magnetic potential
  - Solve collision operator
  - For all particles, push
  - For all particles, collect diagnostics

Stream through Particles

Random access to fields

Celebrating 20 Years with PETSc
Directions being explored to modernize XGC

- Stream particles (not cached)
- Use new “forest of octree” mesh and solvers in PETSc
  - Fast ultra scalable solver & Good data locality for grid operations
- Fuse particle processing loops
- Sketch of algorithm (using RK2):
  - Create particles, deposit charge
  - For each time step, for each RK stage irk=1:2
    - Solve for potential
    - For all particles in particle-list:
      - p = particle-list[i]; get E field at p.x (average on gyro radius); Push p;
      - If irk==2
        » If (p is still local)
          » particle-list[i] = p // write back
          » Deposit charge of p
        » Else: put in send-list; remove from particle-list
      - Else: Deposit charge of p // saves a copy of particles (big win)
    - If irk==2:
      - Send send-list
      - For each particle p received: add to particle-list; deposit charge of p
Thank you

Celebrating 20 Years with PETSc