Performance and Algorithms in Extreme Scale XGC Simulation

Advances in Solver and Meshing Technologies

M. Adams, M. Shephard, E. D'Azevedo, S. Ku, R. Hager, and the EPSI Team
Princeton Plasma Physics Laboratory, Rensselaer Polytechnic, Lawrence Berkeley Laboratory, Oak Ridge National Laboratory

Poisson Solvers in XGC1

- Gyrokinetic Poisson Equation with adiabatic electrons
  - Old method, *definite Helmholtz*, MG useful

$$-\nabla_\perp \frac{mn_0}{eB^2}\nabla_\perp \Phi + \Phi = n_i$$

- Real electrons
  - Need multigrid, *pure elliptic*
  - Stable production solver

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

- 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 with electro-magnetic terms**

1. Perturbative calculation of electron density

- Maxwell distribution on flux surface, particles for δf
  - f_0 is of form (with K kinetic energy): $f_0 = C\exp\left(-\frac{K + e(\Phi - \langle\Phi\rangle)}{T}\right)$
  - <> is flux surface average (FSA)
- XGC1 calculates perturbation from **Boltzmann density (i.e. adiabatic electron response) along field line, which equilibrate to flux surface**

$$n_e = n_e^0 \exp\left(\frac{\Phi - \langle\Phi\rangle}{T}\right) + \delta n_e$$

- $\Phi - \langle\Phi\rangle$ 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" solver with PETSc *FieldSplit*

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

- Add auxiliary variable for <Φ> in 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: <Φ> is global $Schur = -I + C_{ave}\left(\Delta + \frac{n_e^0}{T_e}\right)^{-1}\frac{n_e^0}{T_e}B_I$
  - small number ~ O(10²) and linear
- Approach: compute **explicit** Schur complement
  - Block factorization preconditioner, non-iterative
  - Total solve time ~2 X Laplacian solve time
  - But large setup cost that needs amortizing

2. Hybrid fluid (MHD) electrons & kinetic ions

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

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

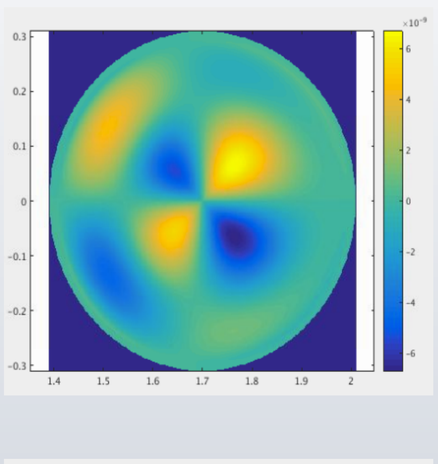
$$Q_0^f = B_0\nabla_\parallel \frac{1}{e\mu B_0}\nabla_\perp^2 \;;\; Q_1^s = \frac{2}{m_e\Omega_e B_0^2}\vec{B}_0 \times \nabla B_0 \bullet \nabla T_0$$

$$\overline{Q}_2^s = \nabla\frac{j_0}{eB_0} \bullet \nabla \times \frac{\vec{B_0}}{B_0} \;;\; Q_3^s = \frac{2n_0}{B_0^3}B_0 \times \nabla B_0 \bullet \nabla + \nabla n_0 \bullet \left(\frac{\vec{B_0}}{B_0^2} \times \nabla\right)$$

$$C_1 = B_0\nabla_\parallel \frac{n_0 u_i}{B_0}$$

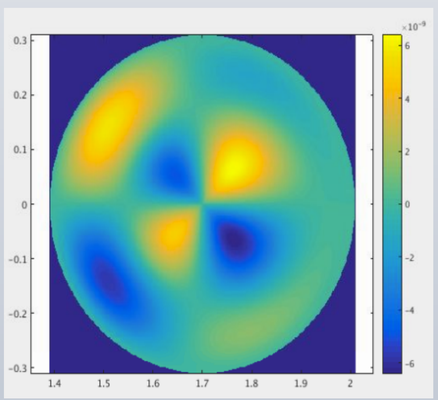
- PETSc time stepper (TS) with IMEX solve (some slow terms moved to RHS)
- 2 auxiliary equations (for potential and current)
  - Verify w/ reduced prob: fast wave & slow growth mode
  - Fully implicit
  - Again use PETSc Field split solvers

$$\begin{bmatrix} 0 & 0 & 0 & B_0\nabla_\parallel \frac{1}{eB_0} \\ 0 & 0 & \nabla_\parallel & 0 \\ M & 0 & -\frac{n_0 m_i}{B^2}\nabla_\perp^2 & 0 \\ 0 & -\nabla_\perp^2 & 0 & \mu M \end{bmatrix}\begin{bmatrix} n_1 \\ A \\ \phi \\ J \end{bmatrix} + \begin{bmatrix} \frac{\partial}{\partial t}n_1 \\ \frac{\partial}{\partial t}A \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



Explicit method

100 time steps

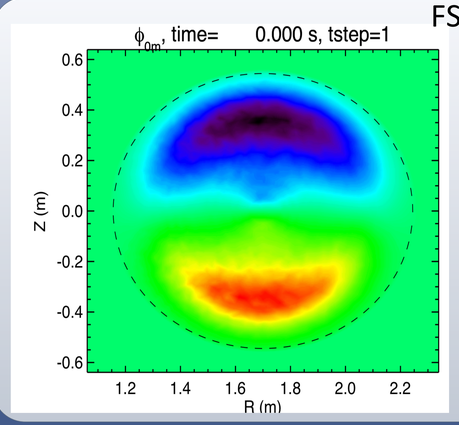


Implicit method

10 time steps

10X dt

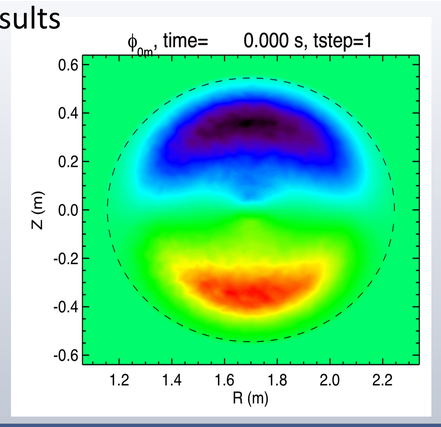
FSA solver results



Old Iterative solver

Solution after 10,000 iterations

384 sec



New PETSc solver

Solution after one iteration

0.086 sec

Meshing in Edge Physics Simulations

Must capture behaviors at multiple scales

- Employ overlapping two scale approach to solve gyrokinetic Vlasov equations
  - Tracking of ions and electrons through particle equations of motion
  - Gyrokinetic Maxwell equation numerically solved at the reactor scale to define fields that push particles
  - Must account for coupling of fields and particles

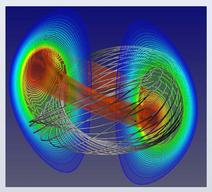
XGC code is a highly parallel code that implements this multiscale simulation

- Particle motion is updated using Runge-Kutta 4 step iteration or 3rd-order Predictor-Corrector
- Reactor fields solved using an unstructured mesh finite element method
- Mesh and particles must be properly coupled at each step in the simulation

Requirements Driving Meshing Efforts

Critical to account for the physics of the particle motion in the models and associated numerical solution methods

- Motion primarily confined to be along field lines of the magnetic field (on which the magnetic flux is constant)
- Motion of particles in toroidal direction much faster than in the poloidal plane



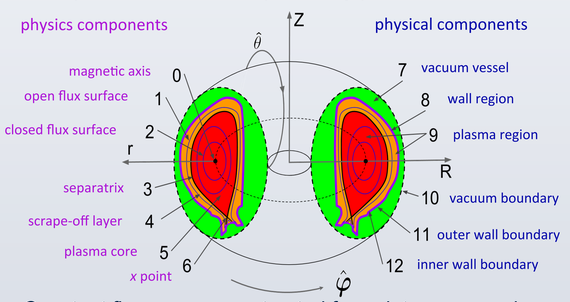
Effective numerical models account for the physics of the particle motion – leads to a set of mesh requirements

- Primary mesh is a 2D mesh resolving the poloidal plane
  - Most effective to carefully align mesh to follow magnetic field lines in the critical regions
  - Projection of particle motion on the poloidal plane controls mesh vertex placement on flux curves
  - Coupling of particle and mesh methods, and accuracy of mesh method dictate mesh fineness and gradation, as well as need for element shape control

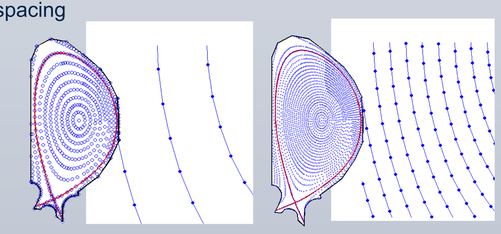
Geometry Definition

Automatic meshing requires unique geometry representation

- Boundary representation required
- Meshing requirements dictate geometric model meshed – it includes combination of physical and physics entities
- Flux surfaces are key physics component required
  - Require mesh vertices specifically placed on flux curves



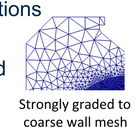
- Constant flux curves constructed from data measured on a uniform grid
  - Input flux (ψ) values are obtained from experimental data on a coarse uniform grid
  - Interpolative splines are constructed from this data to give continuous and smooth ψ values
  - Curves are then constructed to be perpendicular to the gradient of ψ and parallel to the magnetic field, meaning each curve has a constant ψ value
  - Multiple X-point configurations supported
  - Multiple control options for ψ curves used and nodal spacing



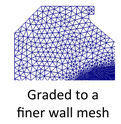
Meshing Generation Method

Combines multiple meshing functions

- General triangulation
  - Generate well shaped, graded mesh near reactor wall where geometry is complex

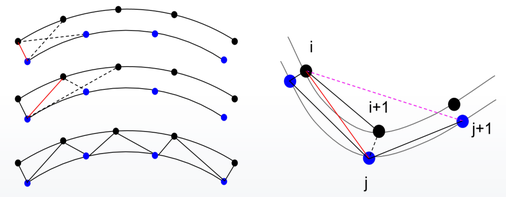


Strongly graded to coarse wall mesh

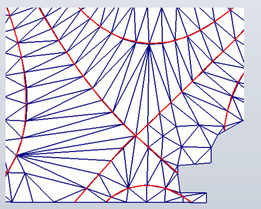


Graded to a finer wall mesh

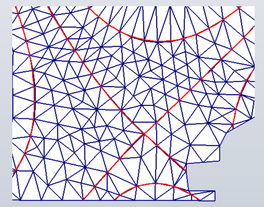
- One-element layered meshing procedure
  - Require depth of one element between critical flux curves
  - Given constant flux curves with vertices placed as required by the vertex to follow particles, a direct algorithm used to construct best shaped elements



- Mesh Control near X-points
  - Near X-point the one-element procedure between flux surfaces produces poorly shaped elements
    - Such element can degrade the local solution
    - Local physics is such that maintaining the one-element deep layer is not critical
  - Apply local general mesh modifications to control element shapes



initial layered mesh at X-point



mesh at X-point after mesh modification

- User interface procedures provided that include easy to specify mesh control functions

Mesh Generation Examples

- Original method took hours of combined human and machine time and mesh not completely satisifactory
- Run time (human plus machine) for new procedure reduced to minutes
- New procedure will support dealing with new and more complex reactor configurations (e.g., ITER)