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

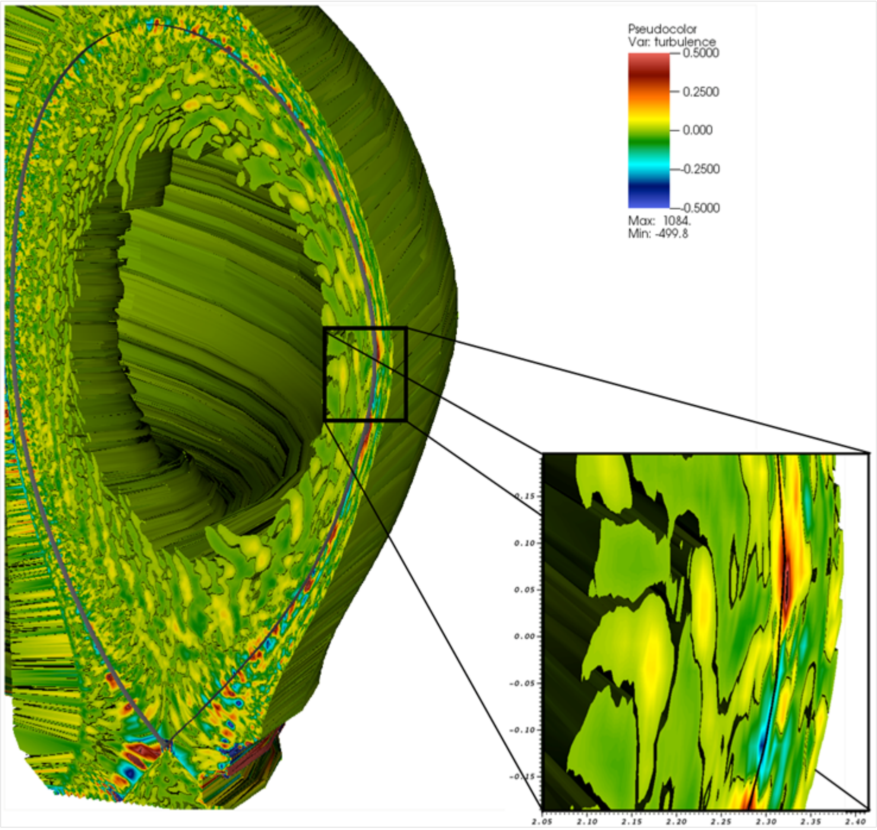
Mark Adams (Lawrence Berkely National Laboratory)
Seung-Hoe Ku (Princeton Plasma Physcis Laboratry)
Robert Hager (Princeton Plasma Physcis Laboratry)

and the EPSI team

Celebrating 20 Years with PETSc

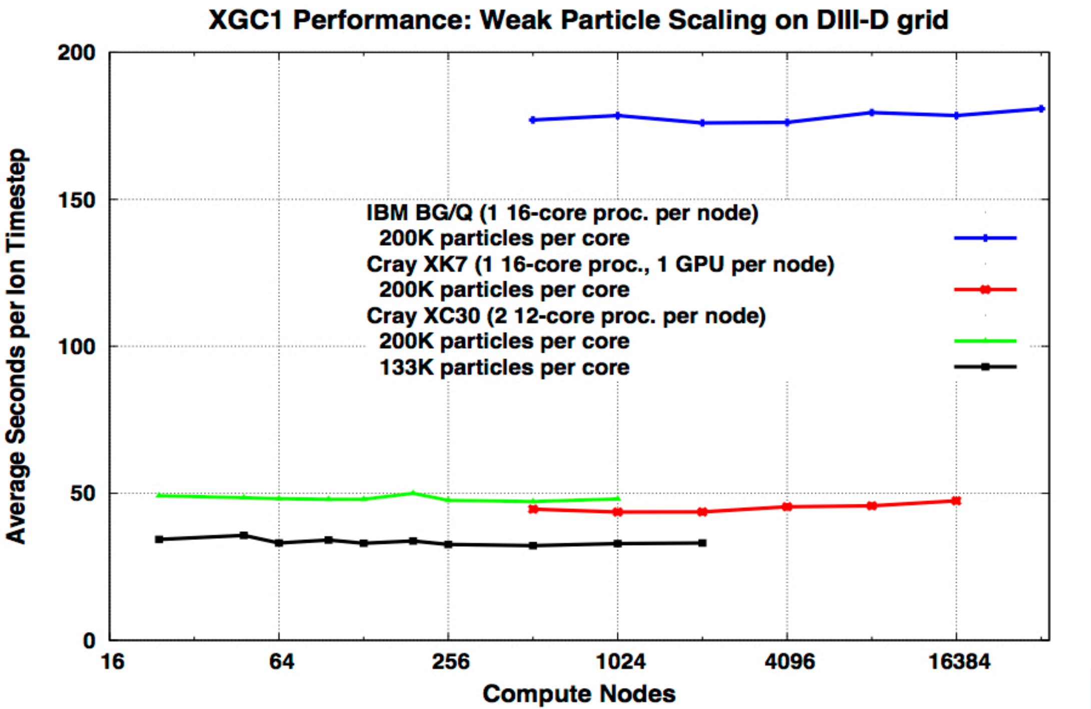
XGC1 – Gyrokinetic particle-in-cell (PIC) code

- 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



XGC1 – extreme scale (SciDAC partnership program)

- Ions and electrons : ~O(10^{10} – 10^{11}) particles each
- Unstructured triangle mesh in 3D : ~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



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

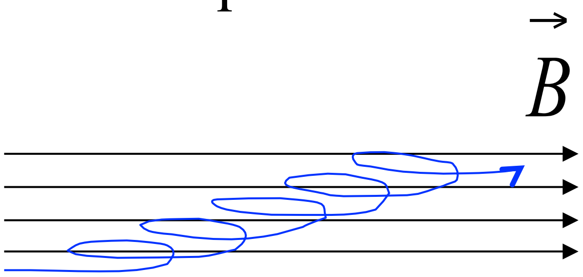
$$\frac{\partial f}{\partial t}+\vec{v}\cdot\frac{\partial f}{\partial \vec{x}}+\frac{\vec{F}}{m}\cdot\frac{\partial f}{\partial \vec{v}}=C$$

$$f=f(\vec{x},\vec{v})$$

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

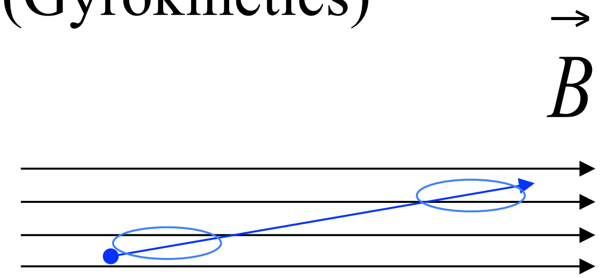
Gyrokinetics

Real particle motion
Lorentz Equation



Point particle with charge q

Gyro Center Motion
(Gyrokinetics)



Ring with radius ρ
magnetic moment μ

- Ring with gyroradius ignoring gyro-phase
  - 6D → 5D

XGC discretization

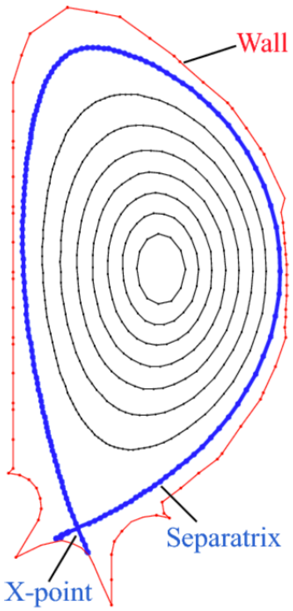


Figure 1. Magnetic flux surfaces of diverted tokamak geometry in a poloidal plane. X-point is shown on the separatrix.

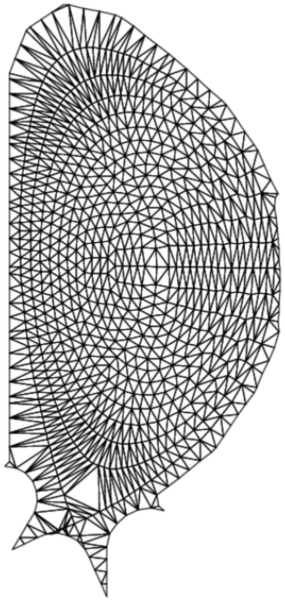


Figure 2. A sample unstructured triangular mesh with ITER magnetic field. The actual scale of the grid used in the simulations is approximately 30 times finer.

- 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

$$f = f(\vec{X}, \mathrm{u}_{\parallel}, u_{\perp});\ \mathrm{X}:\ \text{guide center position}$$

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

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

$$\dot{u}_{\parallel} = -\frac{1}{mD}\left(\hat{b} + \frac{mcu_{\parallel}}{qB} \nabla \times \hat{b} \right) \cdot \left(\mu \nabla B + q\bar{E} \right)$$

$$D = 1 + \frac{mcu_{\parallel}}{qB} \hat{b} \cdot \nabla \times \hat{b}$$

$$\hat{b} = \vec{B} / B$$

$$\mu = mu_{\perp}^{2} / 2B$$

Gyrokinetic Poisson Equation

$$-\nabla_{\perp} \frac{mn_{0}}{eB^{2}} \nabla_{\perp} \Phi = n_{i} - n_{e}$$

5D Gyrokinetic Equations: simplified

- Particle motions (ODEs)

$$\dot{\mathbf{x}} = \mathbf{v}_{\|} + \mathbf{v}_{D}$$

$$\dot{\mathbf{v}}_{\|} = E_{\|} + \mu \nabla B$$

$\mathbf{x}$: 3D space coordinates

$\mathbf{v}_{\|}$: Parallel velocity to magnetic field

B : Magnetic field strength

$\mathrm{E}_{\|}$: Parallel electric field

μ : magnetic moment

n_e^0 : electron density with zero potential

EPSi Edge Physics Simulation

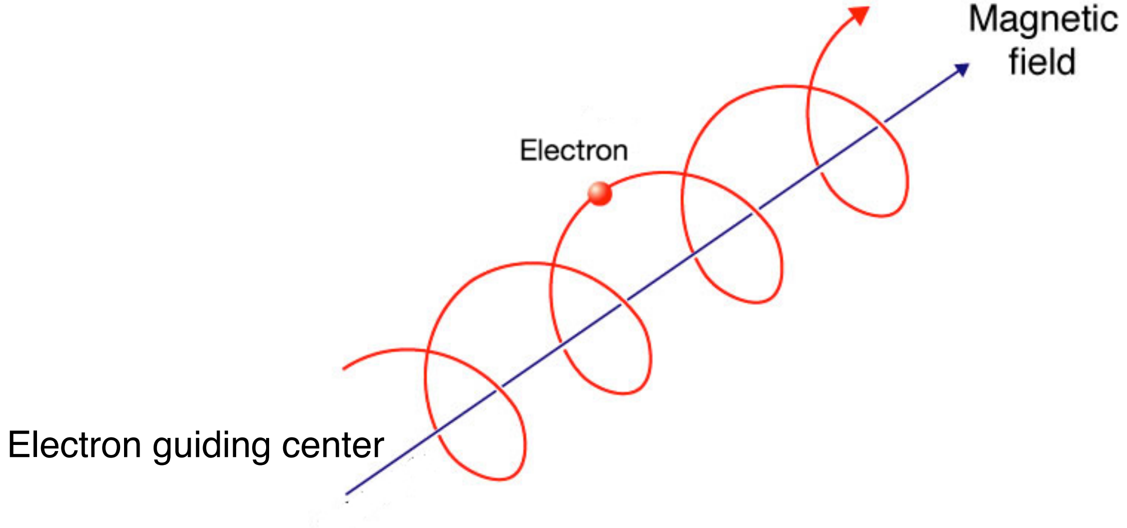
Adiabatic Electron Response

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

Dominant motion

$$\dot{\mathbf{x}} = \mathbf{v}_{||} + \mathbf{v}_D$$

Small drift



- 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: $f = f_0 + \delta f$

Poisson solvers in XGC1

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

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

- Real electrons
  - Need multigrid
  - 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; electro-magnetic

I. Perturbative calculation of electron density

- Maxwell distribution on flux surface, particles for δf
  - f_0 is of form, with K kinetic energy):
  - <> flux surface average (FSA)

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

- 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 accurate solver

- Add auxiliary variable for <Φ>

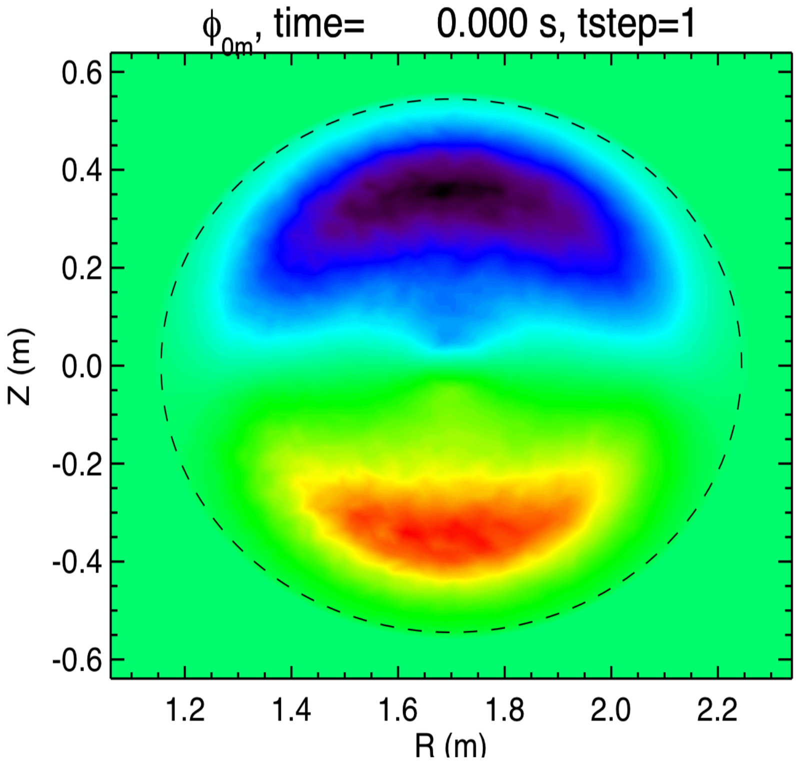
$$\begin{pmatrix} \Delta + \left(n_e^0/T_e\right) & -n_e^0/T_e B_I \\ C_{ave} & -I \end{pmatrix}\begin{pmatrix} \Phi \\ \langle\Phi\rangle \end{pmatrix} = \begin{pmatrix} n_i - n_e^0 - \delta n_e \\ 0 \end{pmatrix}$$

  - 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
  - small number ~ O(10^2) and linear

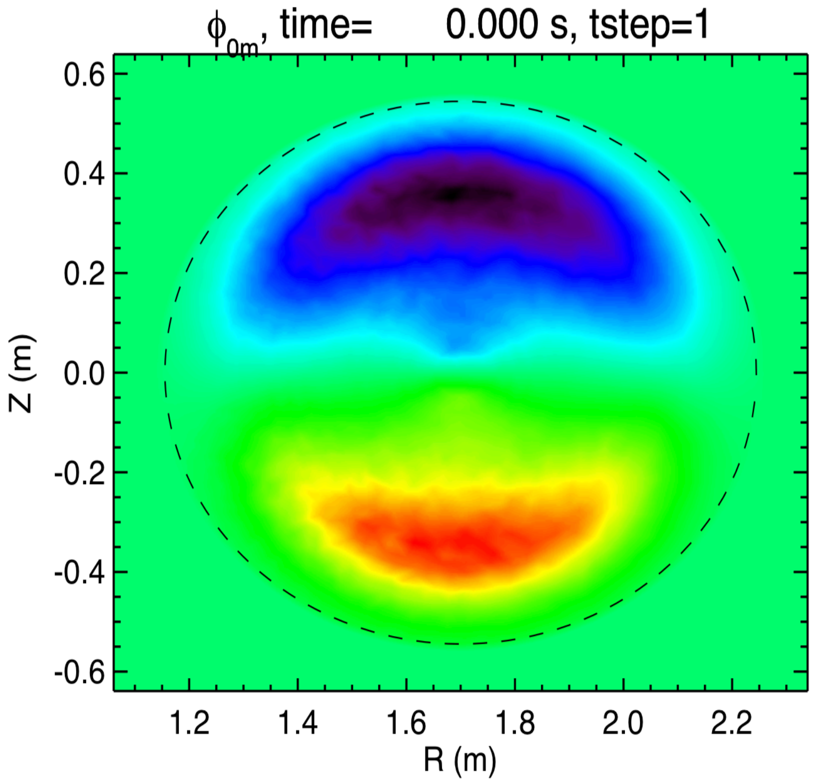
$$Schur = -I + C_{ave}\left(\Delta + \frac{n_e^0}{T_e}M\right)^{-1}\frac{n_e^0}{T_e}MB_I$$

- Approach: compute **explicit** Schur compliment
  - Block factorization preconditioner, non-iterative
  - Total solve time ~2 x Laplacian solve time
  - But large setup cost that needs amortizing

Old Iterative solver vs Field Split



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



Solution from field split solver: 0.086 sec

2. Hybrid fluid (MHD) electrons & kinetic ions

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

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

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

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

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

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

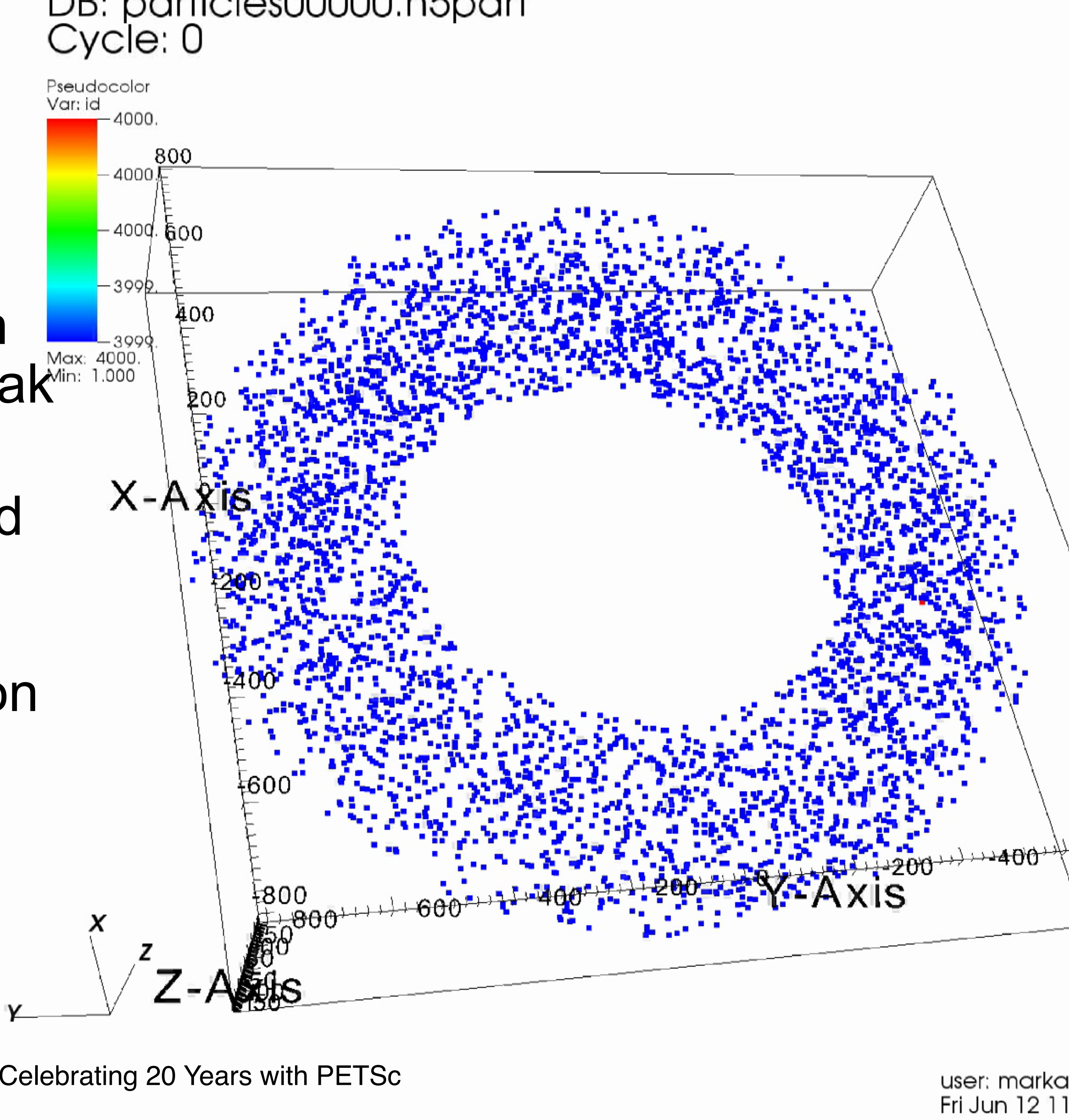
$$\begin{bmatrix} 0 & 0 & 0 & B_0 \nabla_{\parallel} \dfrac{1}{eB_0} \\ 0 & 0 & \nabla_{\parallel} & 0 \\ M & 0 & -\dfrac{n_0 m_i}{B^2} \nabla_{\perp}^2 & 0 \\ 0 & -\nabla_{\perp}^2 & 0 & \mu M \end{bmatrix} \begin{bmatrix} n_1 \\ A \\ \phi \\ J \end{bmatrix} + \begin{bmatrix} \dfrac{\partial}{\partial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

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)

EPSi
Edge Physics Simulation

- Particles with basic Tokamak dynamics
- FE solver grid decoupled from particle decomposition
- Pericles with red particle near outside flux surface



BERKELEY LAB

Future directions for XGC – use modern solvers

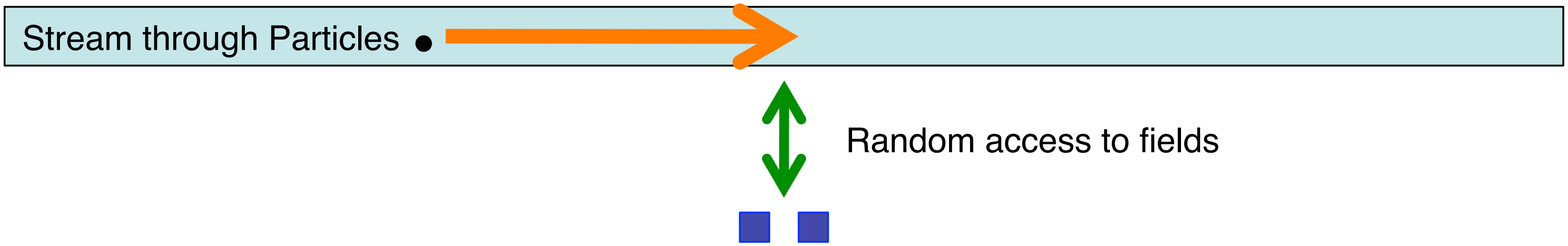
- Tightly couple time integrators in PETSc (IMEX)
  - Long term: need to put all code in PETSc
- Sketch of simple PIC, electro-static, one species

Particle position & vel

$$\begin{bmatrix} 0 & -I & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{q}{m}\int & 0 \\ 0 & 0 & \nabla^2 & 0 & -I \\ 0 & 0 & \nabla & I & 0 \\ \int & 0 & 0 & 0 & -I \end{bmatrix} \begin{bmatrix} P_x \\ P_v \\ \phi \\ E \\ \rho \end{bmatrix} + \begin{bmatrix} \frac{\partial}{\partial t} & 0 \\ 0 & \frac{\partial}{\partial t} \end{bmatrix} \begin{bmatrix} P_x \\ P_v \end{bmatrix} = S$$

Use PETSc semi-structured solvers, etc. & fuse loops

- 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

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 i *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