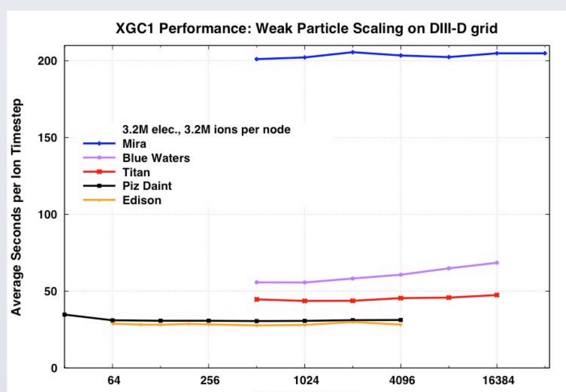
Algorithm and Performance Enhancements in XGC1

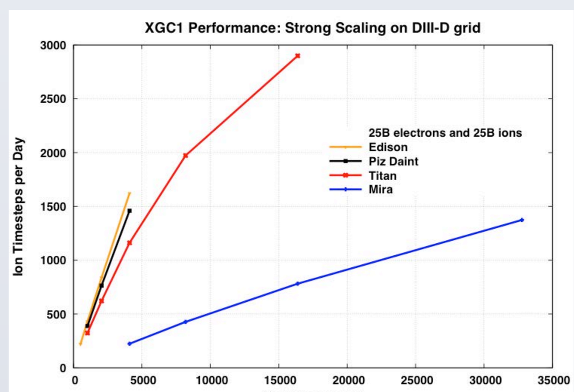
M. F. Adams[1,a], E. D'Azevedo[2,b], C. S. Chang[3], R. Hager[3], J. Hesthaven[4], S. Janhunen[3], S. Ku[3], J. Lang[3], B. Wang[5], P. Worley[2,b], and the EPSI Team

[1]LBNL, [2]ORNL, [3]Princeton Plasma Physics Laboratory, [4]Brown U. ([a]FASTMath,[b]SUPER), [5]Princeton U.

Platform Performance Comparison (collaboration with SUPER Institute)

The June 2013 version of XGC1_3 and a representative problem (DIII-D grid with kinetic electrons) were used to evaluate platform strengths and weaknesses for EPSi science simulations.





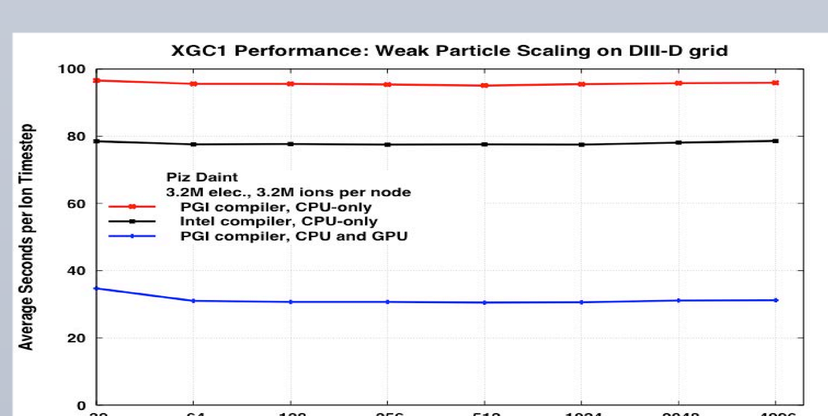
•Titan: Cray XK7 with *one* 2.2 GHz 16-core/8-FPU AMD processor and one NVIDIA Kepler K20 GPU per node and a Gemini interconnect. Titan is the most capable of these five systems for large EPSi science simulations.

•Blue Waters: Cray XE6 with *two* 2.3 GHz 16-core/8-FPU AMD processors per node and a Gemini interconnect.

•Edison: Cray XC30 with *two* 2.4GHz 12-core Intel processors per node and an Aries interconnect. Using 2-way hyperthreading, Edison has the most capable compute node, but is too small for the largest simulations.

•Mira: IBM BG/Q with *one* 1.6 GHz 17-core PowerPC processor per node. Even with 4-way hyperthreading, Mira is not competitive for particle-push dominated simulations. It is a useful resource for grid operation-dominated XGC1 simulations.

•Piz Daint: Cray XC30 with *one* 2.6 GHz 8-core Intel processor and one NVIDIA Tesla K20X GPU per node and an Aries interconnect. Piz Daint is an interesting compromise between Titan and Edison, approaching the performance of Edison, but is similarly too small.



Piz Daint CPU performance is hurt by dependence on the PGI compiler (required for the CUDA Fortran implementation of GPU code). Replacing CUDA Fortran with OpenACC or OpenMP 4 compiler directives could enhance both performance portability and code maintenance

Verification of 3D Poisson's equation using the method of manufactured solution

- Developed of a new fully 3D electro-magnetic solver in toroidal geometry (r,θ,ζ) for higher accuracy long wave length solutions: Must be verified.
- The existing 3D solver is 2D in (r,θ) with simple finite difference in ζ
- Initial verification in 3D Poisson solver
- The perturbed gyrokinetic Poisson's equation is $(\phi^* = \phi - \langle\phi\rangle)$

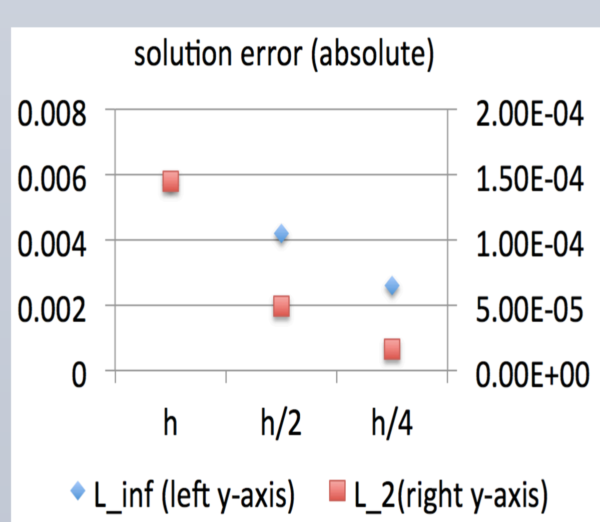
$$-\frac{n_0 m_i}{eB^2}\nabla_\perp^2\phi^* + \frac{n_0}{T_e}\phi^* = \delta\bar{n}_i - <\delta\bar{n}_i>,$$

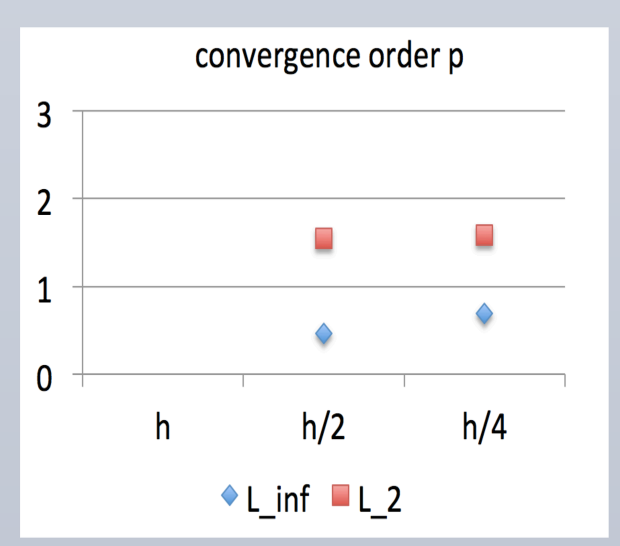
- Manufactured solution is

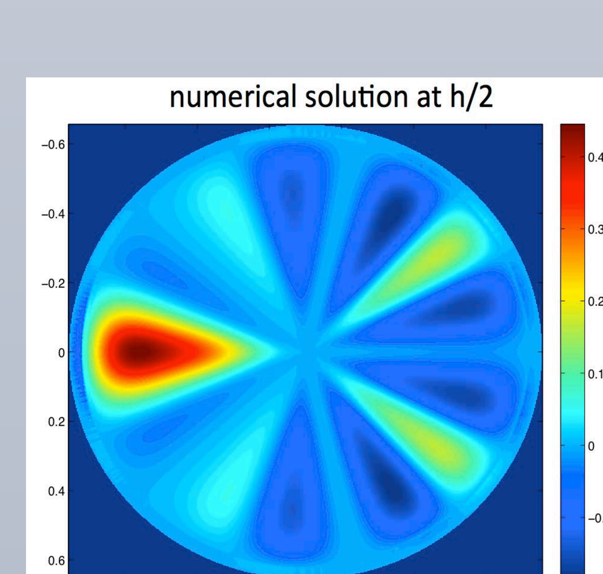
$$f(\xi,\theta,\zeta) = \sum_l A_l J_l(a_l\xi)\sum_m B_m\cos(m\theta)\sum_n C_n\cos(n\zeta)$$

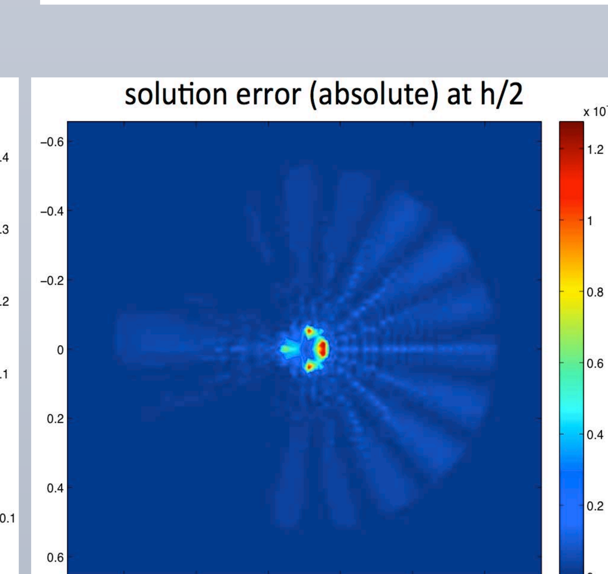
where $\zeta = (r - r_{\min})/(r_{\max} - r_{\min})$

- Source term calculated from manufactured solution
- Measure the 3D solver error in terms of designed norm and evaluate convergence order using $p = \ln(\varepsilon_{2h}/\varepsilon_h)$









Improving Performance of new Nonlinear Collision Operator (three way collaboration among FASTMath, SUPER and EPSI)

Challenge

- New nonlinear Fokker-Planck-Landau collision operator for highly non-Maxwellian edge plasma
- 2D direct velocity space solver at each cell requires Picard iteration and solution of many small sparse linear systems in PETSc
- Initial implementation of the new nonlinear collision solver proved impractical: It nearly doubled compute time of XGC1
- Insufficient work in a single grid cell to fully exploit 16-way OpenMP capability

Solution

- •Effective OpenMP parallelization across cells using thread-safe versions of PETSc, created by FASTMath (B. Smith), and PSPLINE, in collaboration with SUPER
- Multi-threaded nested OpenMP parallelization across grid cells achieved in collaboration with SUPER liaisons
- Collision solver accelerated by over 5X, and XGC1 by 1.7X

	Main loop time	Collision routine time
Multi threaded collision	200 s	32 s
Single threaded collision	340 s	171 s

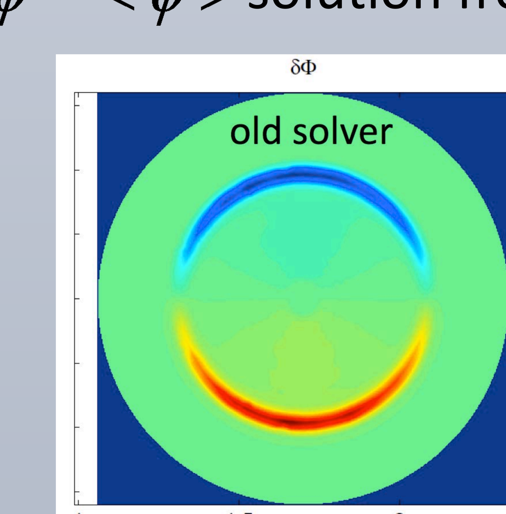
Gyrokinetic Poisson Solver (collaboration with FASTMath)

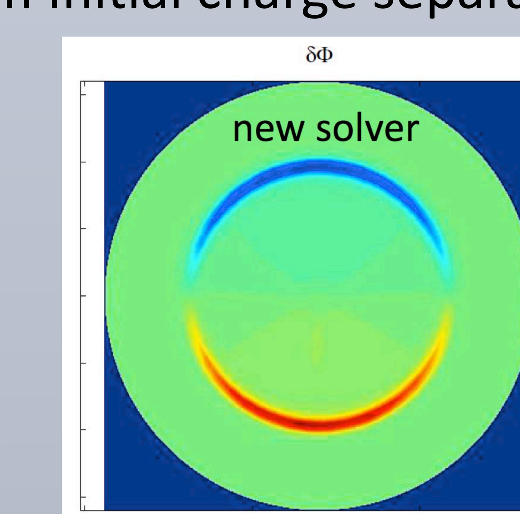
- Upgrade the gyrokinetic Poisson solver to one 2-field solver for ϕ and $<\phi>$, using PETSc's FieldSplit solvers

$$\begin{pmatrix} \frac{mn_0}{eB^2}\Delta + n_0 MC & -n_0 MB_I \\ D & -I \end{pmatrix}\begin{pmatrix} \phi \\ <\phi> \end{pmatrix} = \begin{pmatrix} \delta n_i - \delta n_e^{NA} + n_0(1-C) \\ 0 \end{pmatrix}$$

- Needed for nonlinear Poisson equation, to be used in the near future
  - Linearized around C=1
- Algebraic multigrid solvers perform well
- Anisotropic stiffness removed with two field formulation, more accurate perturbed physics
- Solutions do not have acceptable accuracy
- Hypothesis:
- Incompatible discretizations in finite element for ϕ and (low order) finite difference for flux surface average $<\phi>$
- Exacerbated by projection algorithm outside of PETSc solver:
  - Solve PETSc: $(A_{00}+M(I - BD))\phi'_0 = \rho_i$
  - $\phi_0 = BD\phi'_0$
  - Solve (turbulence) PETSc: $A\phi = \rho$
  - $\phi = \phi + \phi'_0 - \phi_0 = \phi + (I - BD)\phi'_0$
- Solutions
  - Consistent (high order) FE discretization of flux surface terms B & D
  - Explore whole algorithm in MATLAB code

$\phi - <\phi>$ solution from Initial charge separation





Acknowledgements:

The work was partially performed at ORNL, which is managed by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725; at LBNL under DE-AC02-05CH11231, and at PPPL under DE-AC02-09CH11466

Contact: Ed D'Azevedo dazevedoef@ornl.gov or Choong-Seock Chang cschang@pppl.gov