Multiphysics Challenges and Recent Advances in PETSc Scalable Solvers

Lois Curfman McInnes
Mathematics and Computer Science Division
Argonne National Laboratory

TU Darmstadt
July 2, 2012

Outline

- **Multiphysics Challenges**
  - Background
  - Outcomes of 2011 ICiS multiphysics workshop
- **PETSc Composable Solvers**
  - PCFieldSplit for multiphysics support
    - Core-edge fusion
    - Heterogeneous materials science
  - Hierarchical Krylov methods for extreme-scale
    - Reacting flow
- **Conclusions**

Acknowledgments

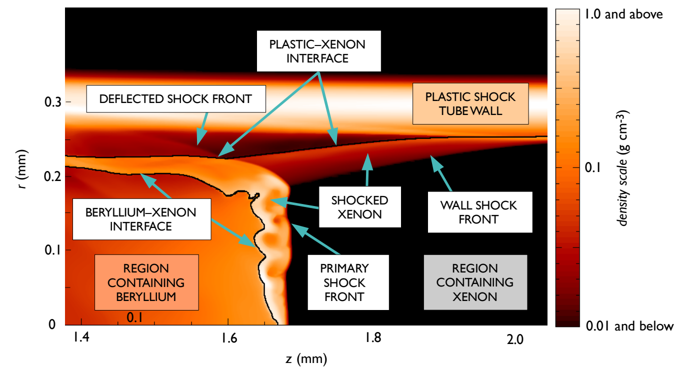
- **U.S. Department of Energy – Office of Science**
  - Base applied math program
  - Scientific Discovery through Advanced Computing (SciDAC): http://www.scidac.gov/
- **Collaborators**
  - D. Keyes, C. Woodward and all ICiS multiphysics workshop participants
  - S. Abhyankar, S. Balay, J. Brown, P. Brune, E. Constantinescu, D. Karpeev, M. Knepley, B. Smith, H. Zhang, other PETSc contributors
  - M. Anitescu, M. McCourt, S. Farley, J. Lee, T. Munson, B. Norris, L. Wang (ANL)
  - Scientific applications teams, especially A. El Azab, J. Cary, A. Hakim, S. Kruger, R. Mills, A. Pletzer, T. Rognlien

Office of Science
U.S. DEPARTMENT OF ENERGY

SciDAC
Scientific Discovery through Advanced Computing

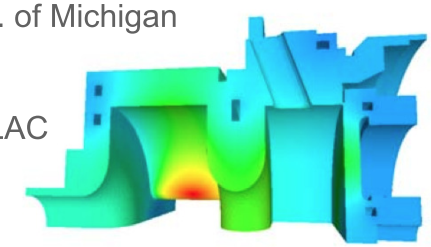
Some multiphysics application areas

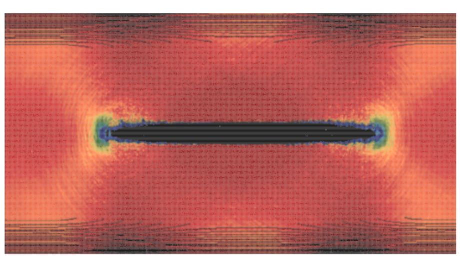
- Radiation hydrodynamics
- Particle accelerator design
- Climate modeling
- Fission reactor fuel performance
- Multiscale methods in crack propagation and DNA sequencing
- Fluid-structure interaction
- Conjugate heat transfer/neutron transport coupling in reactor cores
- Magnetic confinement fusion
- Surface and subsurface hydrology
- Geodynamics and magma dynamics
- Etc.



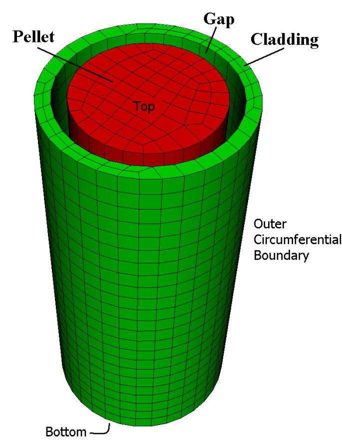
E. Myra, Univ. of Michigan

K. Lee, SLAC

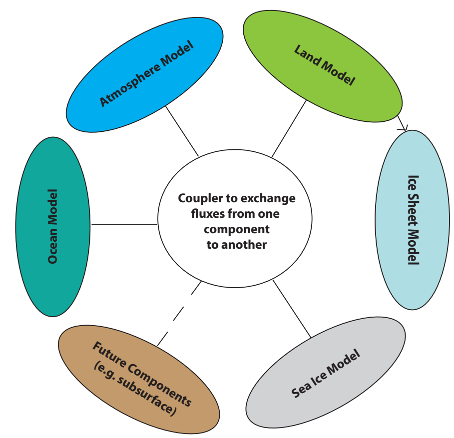




E. Kaxiras, Harvard



G. Hansen, SNL



K. Evans, ORNL

Multiphysics challenges … the study of ‘and’

“We often think that when we have completed our study of one we know all about two, because ‘two’ is ‘one and one.’ We forget that we still have to make a study of ‘and.’ ”

− Sir Arthur Stanley Eddington (1892−1944), British astrophysicist

Target computer architectures

- **Extreme levels of concurrency**
  - Increasingly deep memory hierarchies
  - Very high node and core counts
- **Additional complexities**
  - Hybrid architectures
  - Manycore, GPUs, multithreading
  - Relatively poor memory latency and bandwidth
  - Challenges with fault resilience
  - Must conserve power – limit data movement
  - New (not yet stabilized) programming models
  - Etc.

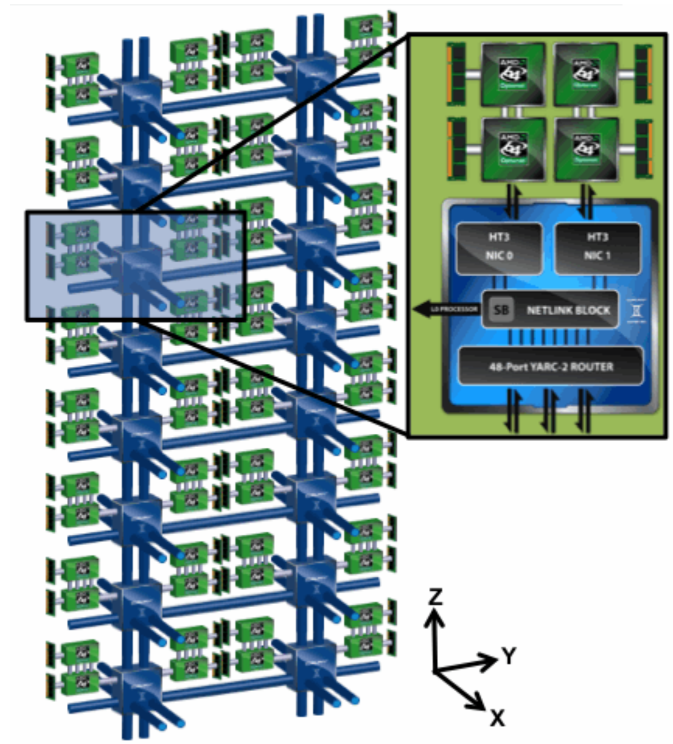


Image courtesy of Cray, Inc.

Multiphysics is a primary motivator for extreme-scale computing

Exaflop: 10^{18} floating-point operations per second

"The great frontier of computational physics and engineering is in the challenge posed by high-fidelity simulations of real-world systems, that is, in truly transforming computational science into a fully predictive science. Real-world systems are typically characterized by multiple, interacting physical processes (multiphysics), interactions that occur on a wide range of both temporal and spatial scales."

– *The Opportunities and Challenges of Exascale Computing*, R. Rosner (Chair), Office of Science, U.S. Department of Energy, 2010

Promote cross-fertilization of ideas (math/CS/apps) to address multiphysics challenges:

- **2011 Workshop sponsored by the Institute for Computing in Science (ICiS)**
  - Co-organizers D.E. Keyes, L.C. McInnes, C. Woodward
  - **https://sites.google.com/site/icismultiphysics2011/**
    - presentations, recommended reading by participants, breakout summaries
    - workshop report:
- ***Multiphysics Simulations: Challenges and Opportunities***
  - D. E. Keyes, L. C. McInnes, C. Woodward, W. D. Gropp, E. Myra, M. Pernice, J. Bell, J. Brown, A. Clo, J. Connors, E. Constantinescu, D. Estep, K. Evans, C. Farhat, A. Hakim, G. Hammond, G. Hansen, J. Hill, T. Isaac, X. Jiao, K. Jordan, D. Kaushik, E. Kaxiras, A. Koniges, K. Lee, A. Lott, Q. Lu, J. Magerlein, R. Maxwell, M. McCourt, M. Mehl, R. Pawlowski, A. Peters, D. Reynolds, B. Riviere, U. Rüde, T. Scheibe, J. Shadid, B. Sheehan, M. Shephard, A. Siegel, B. Smith, X. Tang, C. Wilson, and B. Wohlmuth
  - Technical Report ANL/MCS-TM-321, Argonne National Laboratory
  - **http://www.ipd.anl.gov/anlpubs/2012/01/72183.pdf**
  - Under revision for publication as a special issue of the *International Journal for High Performance Computing Applications*
- **Lots of exciting multidisciplinary research opportunities ☺**

Topics addressed by ICiS workshop

- Practices and Perils in Multiphysics Applications
- Algorithms for Multiphysics Coupling
- Multiphysics Software
- Opportunities for Multiphysics Simulation Research
- Insertion Paths for Algorithms and Software in Multiphysics Applications
- Multiphysics Exemplars and Benchmarks – modest start

discuss today: nonlinear systems in large-scale multiphysics

What constitutes multiphysics?

- Greater than 1 component governed by its own principle(s) for evolution or equilibrium
- Classification:
  - **Coupling occurs in the bulk**
    - source terms, constitutive relations that are active in the overlapping domains of the individual components
    - e.g., radiation-hydrodynamics in astrophysics, magnetohydrodynamics (MHD) in plasma physics, reactive transport in combustion or subsurface flows
  - **Coupling occurs over an idealized interface that is lower dimensional or a narrow buffer zone**
    - through boundary conditions that transmit fluxes, pressures, or displacements
    - e.g., ocean-atmosphere dynamics in geophysics, fluid-structure dynamics in aeroelasticity, core-edge coupling in tokamaks

Broad class of coarsely partitioned problems possess similarities to multiphysics problems:

Allow leveraging and insight for algorithms/software

- **Physical models augmented by variables other than primitive quantities in which governing equations are defined**
  - Building on forward models for inverse problems, sensitivity analysis, uncertainty quantification, model-constrained optimization, reduced-order modeling
    - probability density functions, sensitivity gradients, Lagrange multipliers
  - *In situ* visualization
  - Error estimation fields in adaptive meshing
- **Multiscale: Same component described by more than 1 formulation**
- **Multirate, multiresolution**
- **Different discretizations of same physical model**
  - Grafting a continuum-based boundary-element model for far field onto FE model for near field
- **Systems of PDEs of different types** (elliptic-parabolic, elliptic-hyperbolic, parabolic-hyperbolic)
  - Each of classical PDE archetypes represents a different physical phenomenon
- **Independent variable spaces handled differently or independently**
  - Algebraic character similar to a true multiphysics system

Crosscutting multiphysics issues …

Splitting or coupling of physical processes or between solution modules

- **Coupling via historical accident**
  - 2 scientific groups of diverse fields meet and combine codes, i.e., separate groups develop different "physics" components
  - Often the drive for an answer trumps the drive toward an accurate answer or computationally efficient solution
- **Choices and challenges in coupling algorithms: Do not know *a priori* which methods will have good algorithmic properties**
  - **Explicit methods** are prevalent in many fields for 'production' apps
    - relatively easy to implement, maintain, extend from a software engineering perspective
    - challenges with stability, small timesteps
  - **Implicit methods**
    - free from splitting errors, allow much larger timesteps
    - challenges in understanding timescales, software

Prototype algebraic forms and classic multiphysics algorithms

for deterministic problems with smooth operators for linearization

Coupled equilibrium problem:

$$F_1(u_1,u_2) = 0$$
$$F_2(u_1,u_2) = 0$$

Gauss-Seidel Multiphysics Coupling

Given initial iterate $\left\{u_1^0, u_2^0\right\}$
for k=1,2,..., (until convergence) **do**
  Solve for *v* in $F_1(v, u_2^{k-1}) = 0$; set $u_1^k = v$
  Solve for *w* in $F_2(u_1^{k-1}, w) = 0$; set $u_2^k = w$
end for

loosely coupled

Coupled evolution problem:

$$\partial_t u_1 = f_1(u_1,u_2)$$
$$\partial_t u_2 = f_2(u_1,u_2)$$

Multiphysics Operator Splitting

Given initial values $\left\{u_1(t_0), u_2(t_0)\right\}$
for n=1,2,..., N **do**
  Evolve 1 timestep in $\partial_t u_1 + f_1(u_1, u_2(t_{n-1})) = 0$ to obtain $u_1(t_n)$
  Evolve 1 timestep in $\partial_t u_2 + f_2(u_1(t_n), u_2) = 0$ to obtain $u_2(t_n)$
end for

Field-by-field approach leaves first-order-in-time splitting error in solution

Coupling via Jacobian-free Newton-Krylov approach

If residuals and their derivatives are sufficiently smooth, then a good algorithm for both the equilibrium problem and implicitly time-discretized evolution problem is:

Jacobian-free Newton-Krylov (JFNK)

Problem formulation:

$$F(u) \equiv \begin{pmatrix} F_1(u_1,u_2) \\ F_2(u_1,u_2) \end{pmatrix} = 0, \quad \text{where} \quad u = (u_1, u_2)$$

Newton's method

Given initial iterate u_0

for k=1,2,…, (until convergence) **do**

Solve $J(u^{k-1})\partial u = -F(u^{k-1})$

Update $u^k = u^{k-1} + \partial u$

end for

where $$J = \begin{bmatrix} \frac{\partial F_1}{\partial u_1} & \frac{\partial F_1}{\partial u_2} \\ \frac{\partial F_2}{\partial u_1} & \frac{\partial F_2}{\partial u_2} \end{bmatrix}$$

Assume that J is diagonally dominant in some sense and the diagonal blocks are nonsingular.

- Tightly coupled, can exploit black-box solver philosophy for individual physics components
- Can achieve quadratic convergence when sufficiently close to solution
- Can extend radius of convergence with line search, trust region, or continuation methods

Krylov methods

Krylov accelerator

- Projection methods for solving linear systems, *Ax=b,* using the Krylov subspace

$$K_j = span(r_0, Ar_0, Ar_0^2, \ldots, Ar_0^{j-1})$$

- Require *A* only in the form of matrix-vector products
- Popular methods include CG, GMRES, TFQMR, BiCGStab, etc.
- In practice, preconditioning typically needed for good performance

Matrix-free Jacobian-vector products

- **Approaches**
  - Finite differences (FD)
    - $F'(x)\, v = [\, F(x+hv) - F(x)] / h$
    - costs approximately 1 function evaluation
    - challenges in computing the differencing parameter, *h;* must balance truncation and round-off errors
  - Automatic differentiation (AD)
    - costs approx 2 function evaluations, no difficulties in parameter estimation
    - e.g., ADIFOR & ADIC
- **Advantages**
  - Newton-like convergence without the cost of computing and storing the true Jacobian
  - In practice, still typically perform preconditioning
- **Reference**
  - D.A. Knoll and D.E. Keyes, Jacobian-free Newton-Krylov Methods: A Survey of Approaches and Applications, 2004, *J. Comp. Phys*., 193: 357-397.

Challenges in preconditioning

- Cluster eigenvalues of the iteration matrix (and thus speed convergence of Krylov methods) by transforming *Ax=b* into an equivalent form:

  $$B^{-1}Ax = B^{-1}b \quad \text{or} \quad (AB)^{-1}(Bx) = b$$

  where the inverse action of *B* approximates that of *A*, but at a smaller cost
- How to choose *B* so that we achieve efficiency and scalability? Common strategies include:
  - Lagging the evaluation of B
  - Lower order and/or sparse approximations of B
  - Parallel techniques exploiting memory hierarchy, e.g., additive Schwarz
  - Multilevel methods
  - User-defined custom physics-based approaches

Software is the practical means through which high-performance multiphysics collaboration occurs

> **"The way you get programmer productivity is by eliminating lines of code you have to write."**
>
> – Steve Jobs, Apple World Wide Developers Conference, Closing Keynote Q&A, 1997

Multiphysics collaboration is unavoidable because full scope of required functionality for high-performance multiphysics is broader than any single person or team can deeply understand.

Challenges: Enabling the introduction of new models, algorithms, and data structures

- Competing goals of **interface stability** and **software reuse** with the **ability to innovate algorithmically** and develop new physical models

Two key aspects of multiphysics software design

- **Library interfaces that are independent of physical processes and separate from choices of algorithms and data structures**
  - Cannot make assumptions about program startup or the use of state
  - Cannot seize control of 'main' or assume MPI_COMM_WORLD
- **Abstractions for mathematical objects** (e.g., vectors and matrices), which enable dealing with composite operators and changes in architecture
  - Any state data must be explicitly exchanged through an interface to maintain consistency

Also

- Higher-level abstractions (e.g., variational forms, combining code kernels)
- Interface to support various coupling techniques
  - e.g., LIME, Pawlowski et al.

Outline

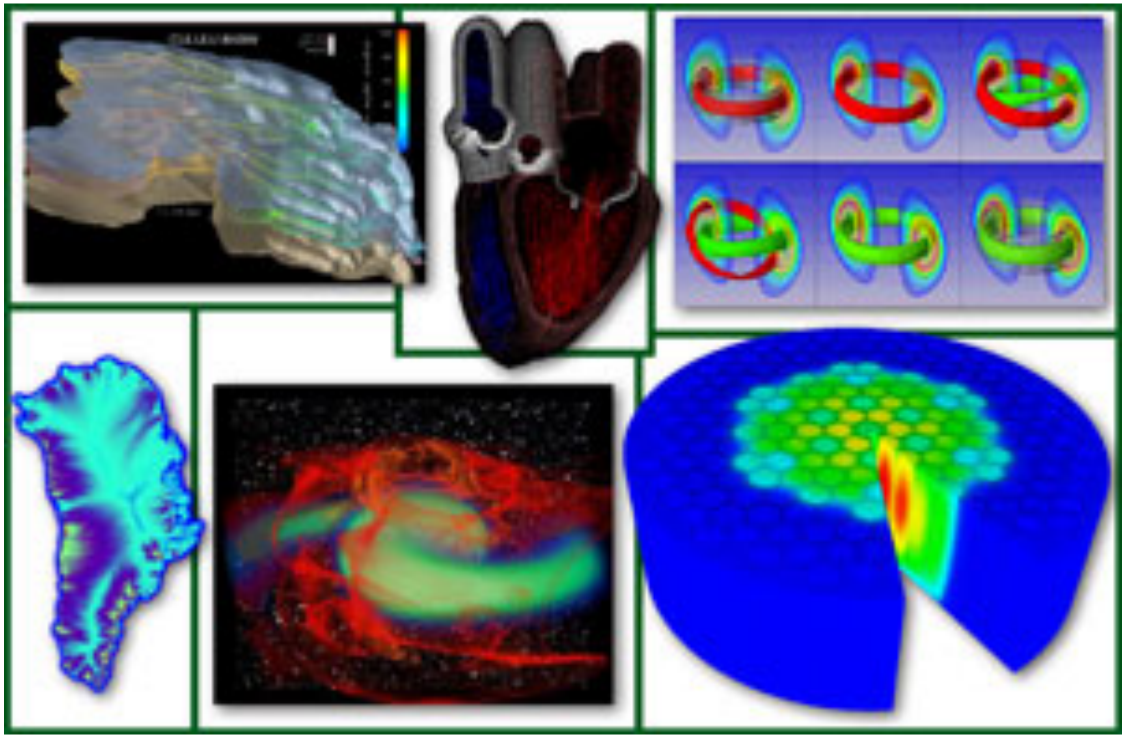
- **Multiphysics Challenges**
  - Background
  - Outcomes of 2011 ICiS multiphysics workshop
- **PETSc Composable Solvers**
  - PCFieldSplit for multiphysics support
    - Core-edge fusion
    - Heterogeneous materials science
  - Hierarchical Krylov methods for extreme-scale
    - Reacting flow
- **Conclusions**

Portable, Extensible Toolkit for Scientific computation

- **Focus: scalable algebraic solvers for PDEs**
  - Freely available and supported research code
  - Download via http://www.mcs.anl.gov/petsc
  - Usable from C, C++, Fortran 77/90, Python, MATLAB
  - Uses MPI; encapsulates communication details within higher-level objects
  - New support for GPUs and multithreading
  - Tracks the largest DOE machines (e.g., BG/Q and Cray XK6) but commonly used on moderately sized systems (i.e., machines 1/10th to 1/100th the size of the largest system)
  - currently @ 400 downloads per month, @ 180 petsc-maint/petsc-users queries per week, @ 40 petsc-dev queries per week
- **Developed as a platform for experimentation**
  - Polymorphism: Single user interface for given functionality; multiple underlying implementations
    - IS, Vec, Mat, KSP, PC, SNES, TS, etc.
- ***No optimality without interplay among physics, algorithmics, and architectures***

PETSc impact on applications

- **Applications include:** acoustics, aerodynamics, air pollution, arterial flow, bone fractures, brain surgery, cancer surgery, cancer treatment, carbon sequestration, cardiology, cells, CFD, combustion, concrete, corrosion, data mining, dentistry, earthquakes, economics, fission, fusion, glaciers, ground water flow, linguistics, mantel convection, magnetic films, materials science, medical imaging, ocean dynamics, oil recovery, page rank, polymer injection molding, polymeric membranes, quantum computing, seismology, semi-conductors, rockets, relativity, …
- Over 1400 cites of PETSc users manual
- 2008 DOE Top Ten Advances in Computational Science
- 2009 R&D 100 Award
- Several Gordon Bell awards and finalists



Software interoperability: A PETSc ecosystem perspective

MOOSE (INL)
Nuclear engineering

Libmesh (UTAustin)
PDEs

OpenFEM
fluids

Fluidity (England)
fluids

Deal II (UTAustin)
PDEs

others …

petsc4py (Argentina)
python

SLEPc (Spain)
eigenvalues

TAO (ANL)
optimization

PHAML (NIST)
Multilevel PDE solver

PETSc

hypre (LLNL)
preconditioners

SuperLU (LBNL)
direct solvers

Trilinos (SNL)
multigrid

MUMPS (France)
direct solvers

Elemental
direct solvers

FFTW
ffts

ParMetis
partitioners

PasTix (France)
direct solvers

Scotch (France)
partitioners

SUNDIALS (LLNL)
ODEs

others …

BLAS
LAPACK

MPI

PThreads

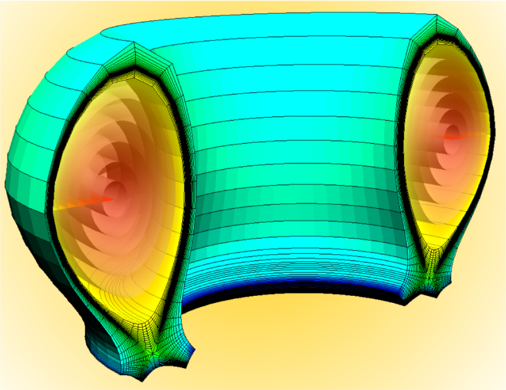
GPUs
CUSP

PETSc interfaces to a variety of external libraries; other tools are built ontop of PETSc.

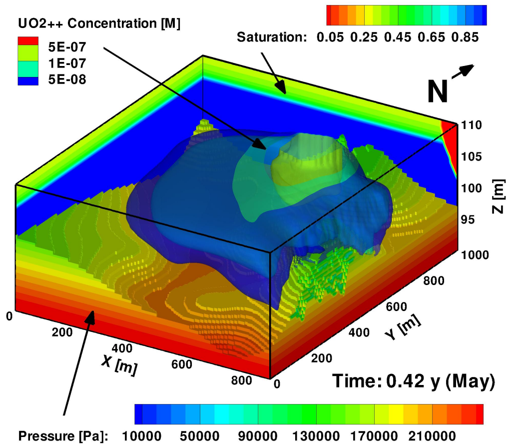
A few motivating applications

Time-dependent, nonlinear PDEs

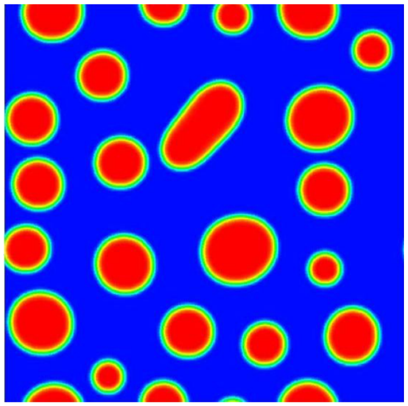
- Key phase: large nonlinear systems
  - Solve $F(u) = 0,$
  - where $F : R^n \rightarrow R^n$



Core-edge fusion,
J. Cary et al.



Reactive transport,
P. Lichtner et al.



Heterogeneous
materials,
A. Azab et al.

Additional multiphysics motivators: geodynamics gyrokinetics, ice sheet modeling, magma dynamics, neutron transport, power networks, etc.

What are the algorithmic needs?

- Large-scale, PDE-based problems
  - Multirate, multiscale, multicomponent, multiphysics
  - Rich variety of time scales and strong nonlinearities
  - Ultimately want to do systematic parameter studies, sensitivity analysis, stability analysis, optimization
- Need
  - Fully or semi-implicit solvers
  - Multilevel algorithms
  - Support for adaptivity
  - Support for user-defined customizations (e.g., physics-informed preconditioners)

Composable solvers are essential; many algorithmic and software challenges

- **Observation:**
  - There is no single, best, black-box (linear/nonlinear/timestepping) solver; effective solvers are problem dependent.
- **Need multiple levels of nested algorithms & data models to exploit**
  - Application-specific and operator-specific structure
  - Architectural features
  - Existing solvers for parts of systems
- **Solver composition:**
  - Solver research today requires understanding (mathematically and in software) how to compose different solvers together – each tackles a part of the problem space to produce an efficient overall solver. Algorithms must respond to: physics, discretizations, parameter regimes, architectures.

Our approach for designing efficient, robust, scalable linear/nonlinear/timestepping solvers

Need solvers to be:

- **Composable**
  - Separately developed solvers may be easily combined, by non-experts, to form a more powerful solver.
- **Hierarchical**
  - Outer solvers may iterate over all variables for a global problem, while inner solvers handle smaller subsets of physics, smaller physical subdomains, or coarser meshes.
- **Nested**
  - Outer solvers call nested inner solvers.
- **Extensible**
  - Users can easily customize/extend.

Protecting PETSc from evolving programming model details

- **Although the programming model for exascale computing is unclear/evolving, the model certainly will need to:**
  - Support enormous concurrency
  - Support "small" to "moderate" scale vectorization
  - Manage/optimize/minimize data movement and overlap with computation
  - Enable support for dynamic load balancing
  - Enable support for fault tolerance
  - Provide bridges to current programming models (MPI, OpenMP, TBB, PThreads, others)
- **Techniques we are using today to prepare code for future programming models**
  - Separating control logic from numerical kernels
  - Separating data movement from numerical kernels (perhaps data movement kernels needed)
  - Managing data in smallish chunks (tiles); allowing load-balancing by moving tiles and restarts on failed tile computations
  - Kernel model supports fusing of kernels
  - Kernel syntax independent of OpenMP, PThreads, TBB (only kernel launcher aware of system details)

Recent PETSc functionality (indicated by blue)

Time Integrators

Pseudo-Timestepping | Runge-Kutta | Strong Stability Preserving
General Linear | IMEX | Rosenbrock-W

Nonlinear Algebraic Solvers

Line Search Newton | Quasi-Newton (BFGS) | Nonlinear Gauss-Seidel | Nonlinear MG (FAS)
Trust Region Newton | Successive Substitutions | Nonlinear CG | Active Set VI

Krylov Subspace Solvers

Richardson | GMRES | Hierarchical Krylov | BiCG Stabilized
Chebychev | TFQMR | LSQR | SYMMLQ | CG | IBCGS

Preconditioners

Blocks (by field) | Additive Schwarz | ILU/ICC
Schur Complement | Algebraic Multigrid | Geometric Multigrid

Matrices

Compressed Sparse Row (AIJ) | Block AIJ | Matrix Blocks (MatNest)
Symmetric Block AIJ | Dense | GPU & PThread Matrices

Vectors **Index Sets**

In PETSc, objects at higher levels of abstraction use lower-level objects.

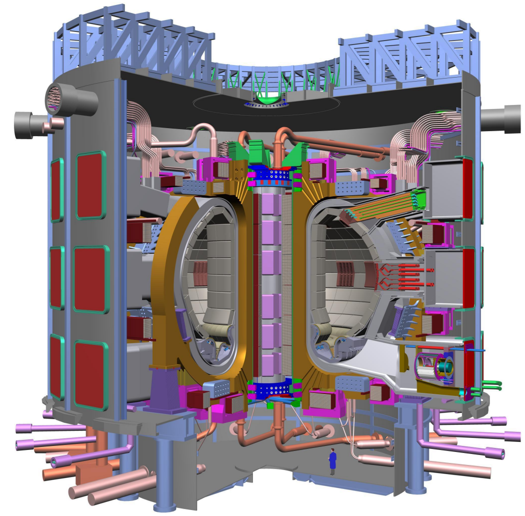
New PETSc capabilities for composable linear solvers

- **PCFieldSplit: building blocks for multiphysics preconditioners**
  - Flexible, hierarchical construction of block solvers
  - Block relaxation or block factorization (Schur complement approaches)
  - User specifies arbitrary pxp block systems via index sets
  - Can use any algebraic solver in each block
- **MatNest**
  - Stores submatrix associated with each physics independently
- **DM abstract class**
  - Provides information about mesh to algebraic solvers but does not impose constraints on their management
  - Solver infrastructure can automatically generate all work vectors, sparse matrices, transfer operators needed by multigrid and composite (block) solvers from information provided by the DM
- More details:
  - ***Composable Linear Solvers for Multiphysics***, J. Brown, M. Knepley, D. May, L.C. McInnes, B. Smith, Proceedings of ISPDC 2012, June 25-29, 2012, Munich, Germany, also preprint ANL/MCS-P2017-0112, Argonne National Laboratory
  - ***PETSc's Software Strategy for the Design Space of Composable Extreme-Scale Solvers***, B. Smith, L.C. McInnes, E. Constantinescu, M. Adams, S. Balay, J. Brown, Matthew Knepley and Hong Zhang, DOE Exascale Research Conference, April 16-18, 2012, Portland, OR, also preprint ANL/MCS-P2059-0312, Argonne National Laboratory

Outline

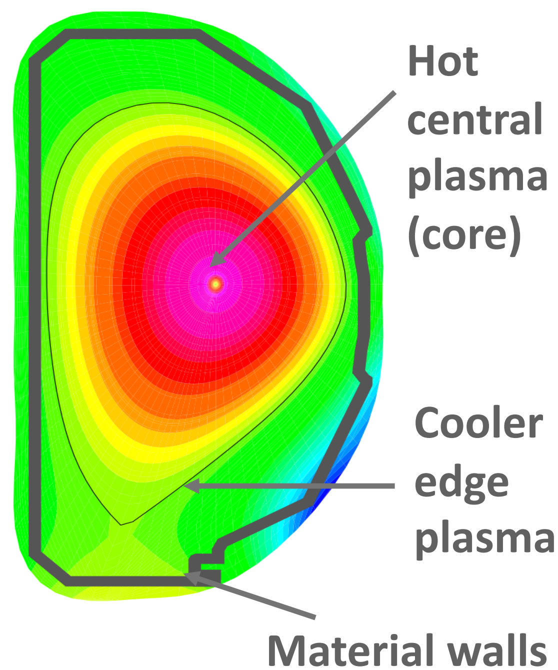
- **Multiphysics Challenges**
  - Background
  - Outcomes of 2011 ICiS multiphysics workshop
- **PETSc Composable Solvers**
  - PCFieldSplit for multiphysics support
    - Core-edge fusion
    - Heterogeneous materials science
  - Hierarchical Krylov methods for extreme-scale
    - Reacting flow
- **Conclusions**

SNES usage: FACETS fusion



ITER: the world's largest tokamak

- **FACETS:** Framework Application for Core-Edge Transport Simulations
  - PI John Cary, Tech-X Corp,
  - https://www.facetsproject.org/facets/
  - **Overall focus:** Develop a tight coupling framework for core-edge-wall fusion simulations
- **Solvers focus**: Nonlinear systems in:
  - **UEDGE** (T. Rognlien et al., LLNL): 2D plasma/neutral transport
  - **New core solver** (A. Pletzer et al., Tech-X)



Nonlinear PDEs in core and edge

Core: 1D conservation laws:

$$\frac{\partial q}{\partial t} + \nabla \bullet F = s$$

where q = {plasma density, electron energy density, ion energy density}

F = fluxes, including neoclassical diffusion, electron/ion temperature, gradient induced turbulence, etc.

s = particle and heating sources and sinks

Challenges: highly nonlinear fluxes

Edge: 2D conservation laws: Continuity, momentum, and thermal energy equations for electrons and ions:

$$\frac{\partial n}{\partial t} + \nabla \bullet (n_{e,i} v_{e,i}) = S^{p}_{e,i}$$

where $n_{e,i}$ & $v_{e,i}$ are electron and ion densities and mean velocities

$$nm_{e,i}\frac{\partial v_{e,i}}{\partial t} + m_{e,i} n_{e,i} v_{e,i} \bullet \nabla v_{e,i} = \nabla p_{e,i} + q n_{e,i}(E + v_{e,i} \times B/c) - \nabla \bullet \Pi_{e,i} - R_{e,i} + S^{m}_{e,i}$$

where $m_{e,i}, p_{e,i}, T_{e,i}$ are masses, pressures, temperatures
q, E, B are particle charge, electric & mag. fields
$\Pi_{e,i}, R_{e,i}, S^{m}_{e,i}$ are viscous tensors, thermal forces, source

$$\frac{3}{2} n \frac{\partial T_{e,i}}{\partial t} + \frac{3}{2} n v_{e,i} \bullet \nabla T_{e,i} + p_{e,i} \nabla \bullet v_{e,i} = -\nabla \bullet q_{e,i} - \Pi_{e,i} \bullet \nabla v_{e,i} + Q_{e,i}$$

where $q_{e,i}, Q_{e,i}$ are heat fluxes & volume heating terms
Also neutral gas equation

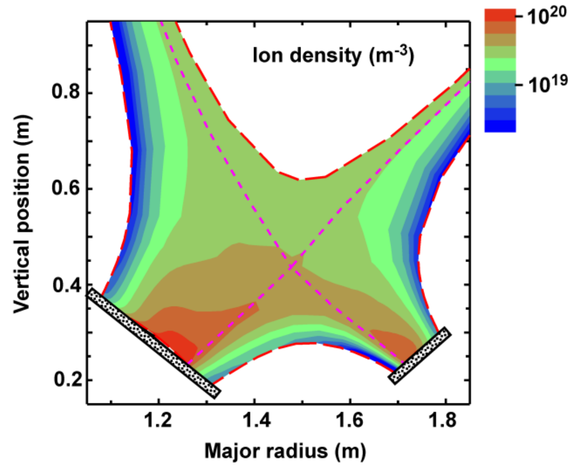
Challenges: extremely anisotropic transport, extremely strong nonlinearities, large range of spatial and temporal scales

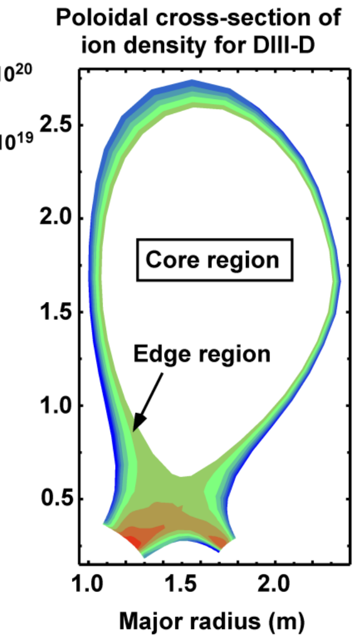
Dominant computation of each can be expressed as nonlinear PDE: Solve *F(u) = 0*, where *u* represents the fully coupled vector of unknowns

UEDGE demands robust parallel solvers to handle strong nonlinearities

Challenges in edge modeling

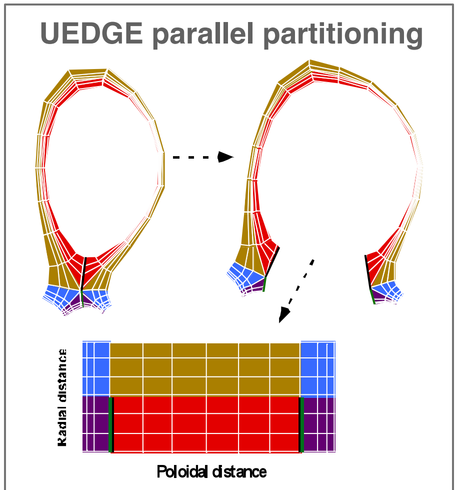
- Extremely anisotropic transport
- Extremely strong nonlinearities
- Large range of spatial and temporal scales





UEDGE (T. Rognlien, LLNL)

- Test case: Magnetic equilibrium for
- DIII-D single-null tokamak
- 2D fluid equations for plasma/neutrals
  - variables: n_i, u_{pi}, n_g, T_e, and T_i (ion density, ion parallel velocity, neutral gas density, electron and ion temperatures)
- Finite volumes, non-orthogonal mesh
- Volumetric ionization, recombination & radiation loss
- Boundary conditions:
  - core – Dirichlet or Neumann
  - wall/divertor – particle recycling & sheath heat loss
- Problem size 40,960: 128x64 mesh (poloidal x radial), 5 unknowns per mesh point



UEDGE approach: Fully implicit, parallel Newton solvers via PETSc

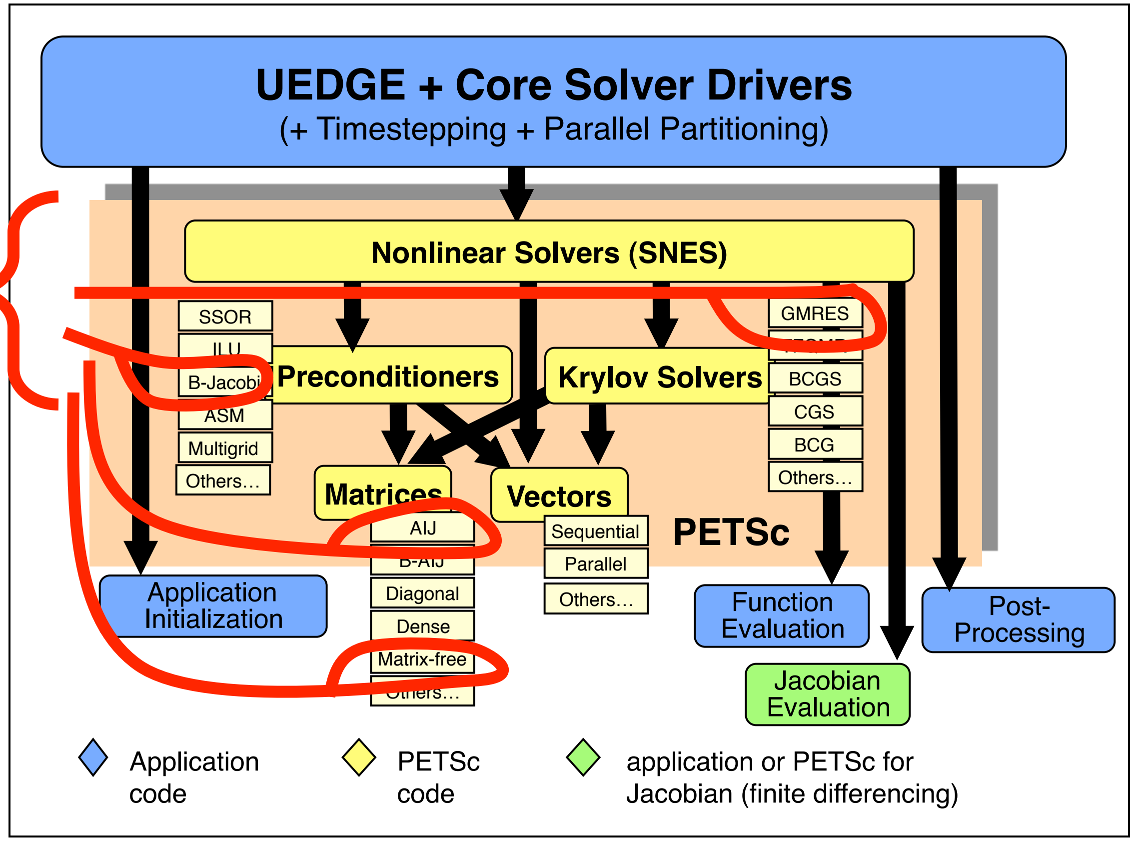
Solve F(u) = 0: Matrix-free Newton-Krylov:

$$F'(u^{l+1})\,\partial u^{l} = -F(u^{l-1})$$

$$u^{l} = u^{l+1} + \lambda\,\partial u^{l}$$

- Solve Newton system with preconditioned Krylov method

Options originally used by UEDGE



- Matrix-free: Compute Jacobian-vector products via finite difference approx; use cheaper Jacobian approx for preconditioner

Features of PETSc/SNES

- **Uses high-level abstractions for matrices, vectors, linear solvers**
  - Easy to customize and extend, facilitates algorithmic experimentation
    - Supports matrix-free methods
  - Jacobians available via application, Finite Differences (FD) and Automatic Differentiation (AD)
- **Application provides to SNES**
  - Residual:
    - `PetscErrorCode (*func) (SNES snes, Vec x,  Vec r, void *ctx)`
  - Jacobian (optional):
    - `PetscErrorCode (*func) (SNES snes,  Vec x, Mat *J, Mat *M, MatStructure *flag, void *ctx)`
- **Features of Newton-Krylov methods**
  - Line search and trust region globalization strategies
  - Eisenstat-Walker approach for linear solver convergence tolerance

Exploiting physics knowledge in custom preconditioners

PETSc: PCFieldSplit simplifies multi-model algebraic system specification and solution: Leveraging knowledge of the different component physics in the system produces a better preconditioner.

$$\begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \longrightarrow \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}$$

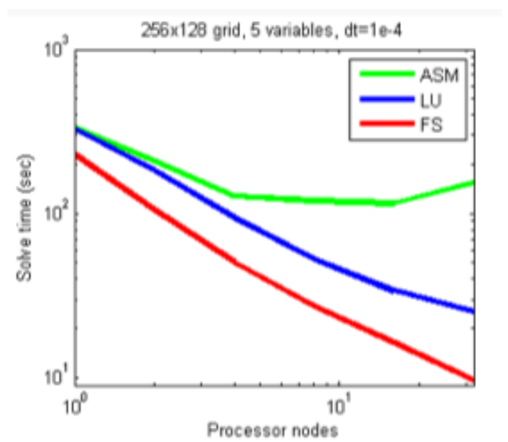
base ordering (all variables per mesh point) — physics-based reordering (for PC only) — components are separated

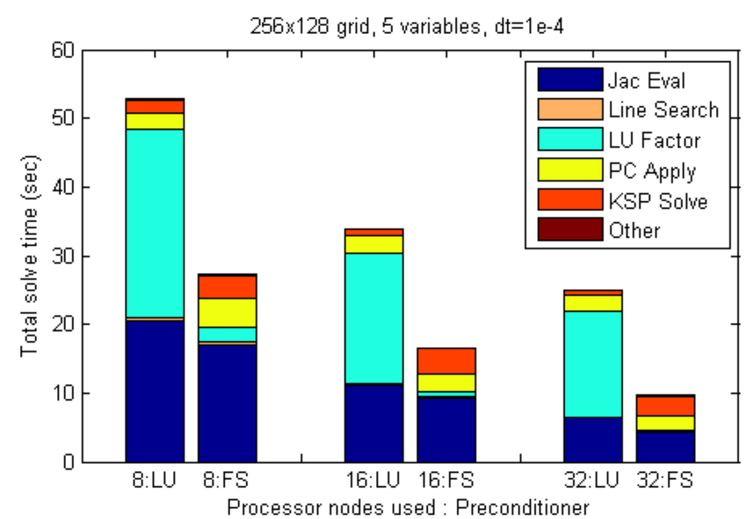
- A_1 Non-neutral terms only
- A_2 Neutral to Non-neutral
- A_3 Non-neutral to neutral
- A_4 Neutral terms only

UEDGE: Implicit time advance with Jacobian-free Newton-Krylov. **PCFieldSplit** overcomes a major obstacle to parallel scalability for an implicit coupled neutral/plasma edge model: greatly reduced parallel runtimes; little code manipulation is required (M. McCourt, T. Rognlien, L. McInnes, H. Zhang, to appear *in Proceedings of ICNSP, Sept 7-9, 2011, Long Branch, NJ*)

Combine component preconditioners:

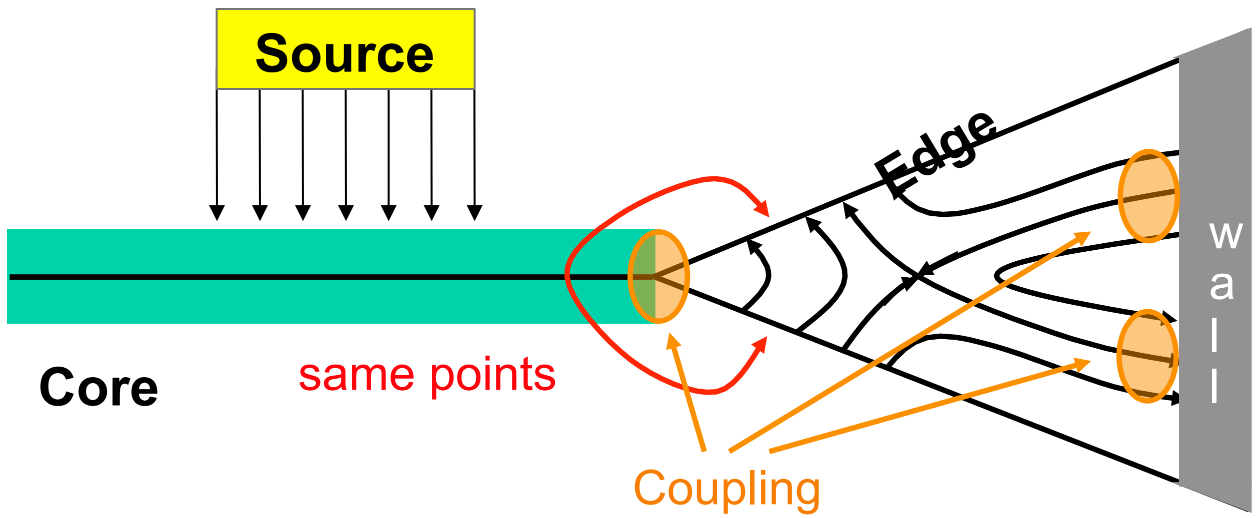
A1: four plasma terms solved with Additive Schwarz
A4: 1 neutral term solved with algebraic multigrid



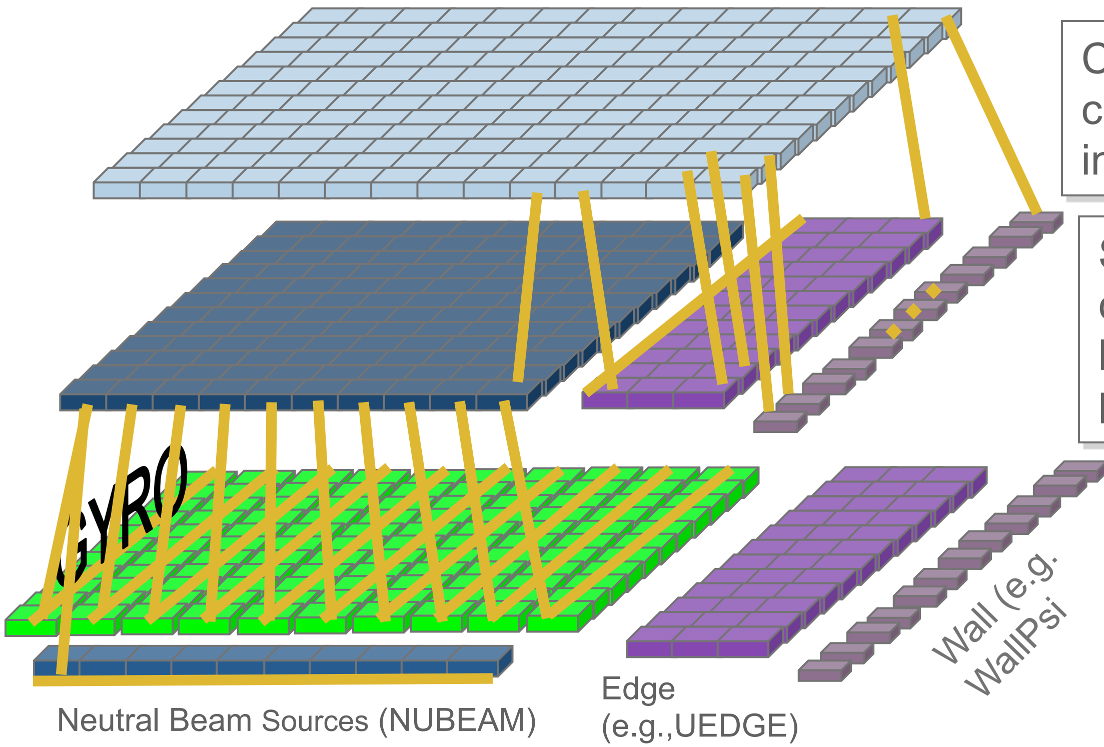


Idealized view: Surfacial couplings

- Core: collisionless, 1D transport system with local, only-cross-surface fluxes
- Edge: collisional 2D transport system
- Wall: beginning of a particle trapping matrix



Core + edge in FACETS framework



Core-edge-wall communication is interfacial

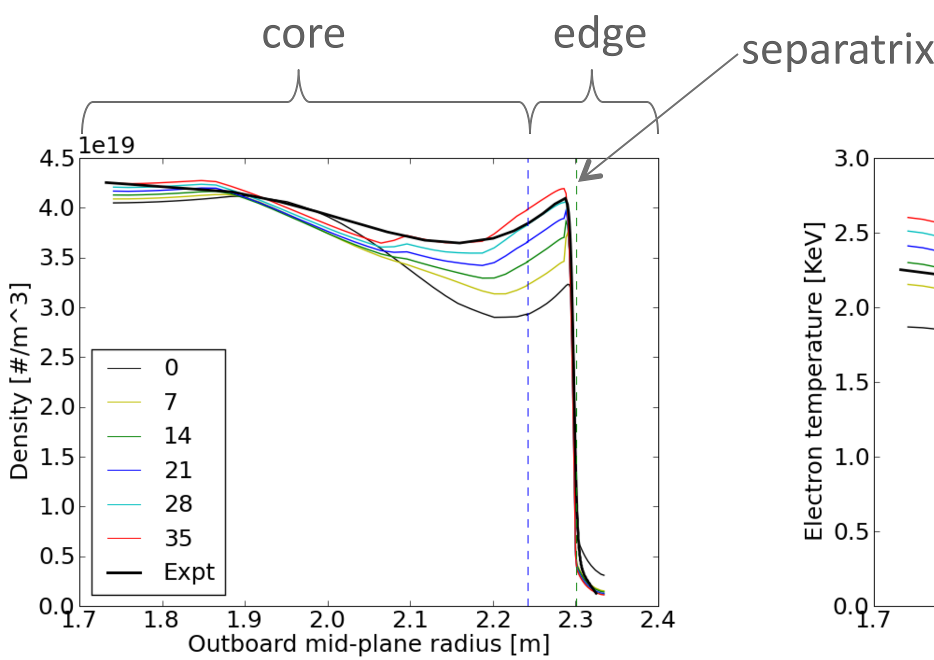
Sub-component communications handled hierarchially

Components use their own internal parallel communication pattern

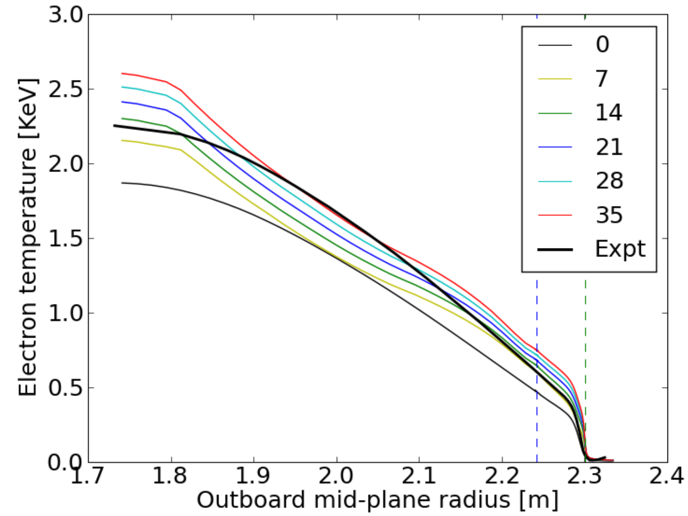
- Simulations of pedestal buildup of DIII-D experimental discharges

Coupled core-edge simulations of H-Mode buildup in the DIII-D tokamak

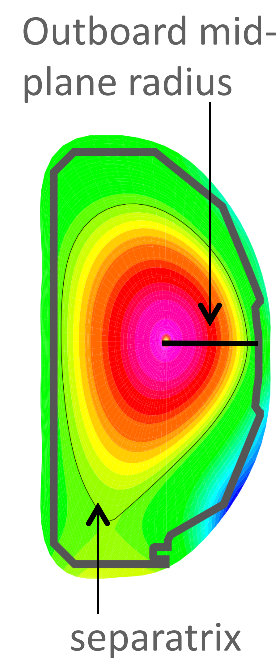
- Simulations of formation of transport barrier critical to ITER
- First physics problem, validated with experimental results, collab w. DIII-D



Time history of density over 35 ms



Time history of electron temp over 35 ms

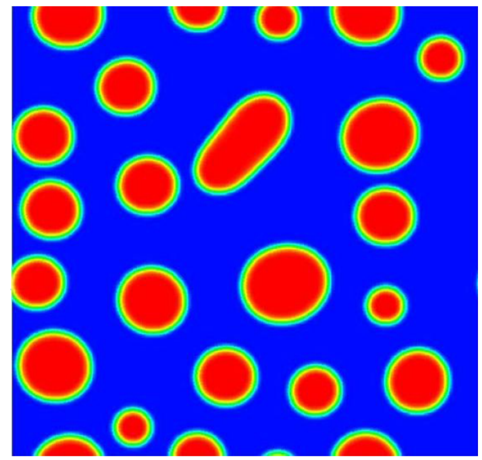


SNES Usage: Heterogeneous materials science

- **Partner EFRC: Center for Materials Science of Nuclear Fuel (PI T. Allen, U. Wisconsin)**
  - https://inlportal.inl.gov/portal/server.pt/community/center_for_materials_science_of_nuclear_fuel
- **SciDAC-e team:** S. Abhyankar, M. Anitescu, J. Lee, T. Munson, L.C. McInnes, B. Smith, L. Wang (ANL), A. El Azab (Florida State Univ.)
- **Motivation: CMSNF using phase field approach** for mesoscale materials modeling; results in differential variational inequalities (DVIs)
  - Prevailing (non-DVI) approach approximates dynamics of phase variable using a smoothed potential: stiff problem and undesirable physical artifacts
- **Broad aim:** Develop advanced numerical techniques and scalable software for DVIs for the resolution of large-scale, heterogeneous materials problems

CMSNF

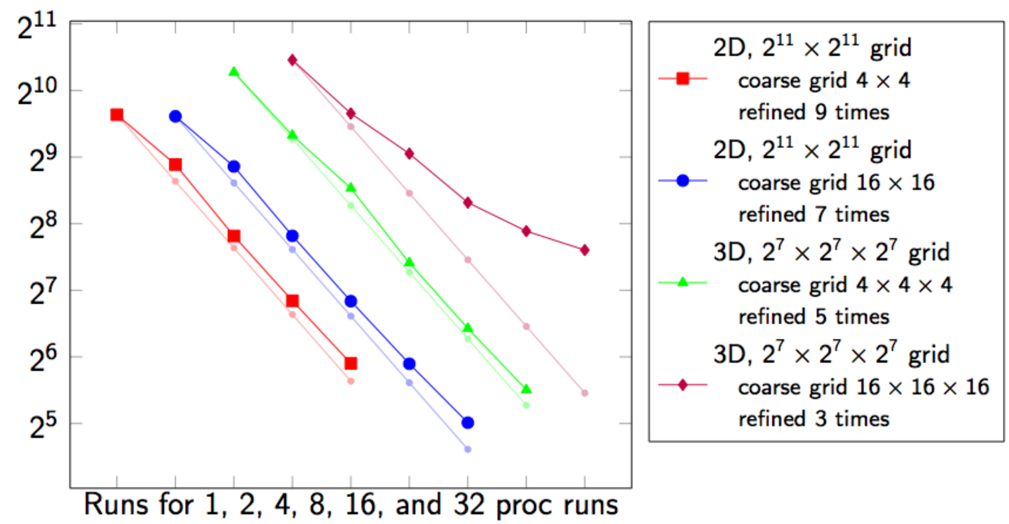
Center for Materials Science of Nuclear Fuel



radiation-generated voids in nuclear fuel

Initial computational infrastructure for DVIs

- Extended PETSc/SNES to include semi-smooth and reduced-space active set VI solvers (leveraging experience in TAO and PATH)
- Validated DVI modeling approach against baseline results of CMSNF for void formation and radiation damage
- Preliminary experiments and analysis for Allen-Cahn and Cahn-Hilliard systems: demonstrate **mesh-independent convergence rates** for Schur complement preconditioners using algebraic multigrid



./ex65 -ksp_type fgmres -pc_type mg -pc_mg_galerkin -mg_coarse_pc_type redundant -mg_coarse_redundant_pc_type lu -mg_levels_pc_type asm -da_refine 9 ...

- **Future:** Large-scale CMSNF problems, analysis of DVI approach, additional VI solvers, extending to other free-boundary physics, etc.

Reference: L. Wang, J. Lee, M. Anitesu, A. El Azab, L.C. McInnes, T. Munson, and B. Smith, *A Differential Variational Inequality Approach for the Simulation of Heterogeneous Materials*, Proceedings of SciDAC 2011.

Dynamically construct preconditioner at runtime

$$\begin{pmatrix} A & \tilde{M}_0 \\ -\tilde{M}_0 & \end{pmatrix} \begin{pmatrix} \cdots \\ \cdots \end{pmatrix} = \begin{pmatrix} \cdots \\ \cdots \end{pmatrix}$$

phase-field modeling
mesoscale materials
Allen-Cahn

See: src/snes/examples/tutorials/ex55.c

- **Schur complement iteration with multigrid on A block:**

```
./ex55 -ksp_type fgmres
-pc_type fieldsplit
-pc_fieldsplit_detect_saddle_point
-pc_fieldsplit_type schur
-pc_fieldsplit_schur_precondition self
-fieldsplit_0_ksp_type preonly
-fieldsplit_0_pc_type mg
-fieldsplit_1_ksp_type fgmres
-fieldsplit_1_pc_type lsc
```

- **Multigrid with Schur complement smoother using SOR on A block:**

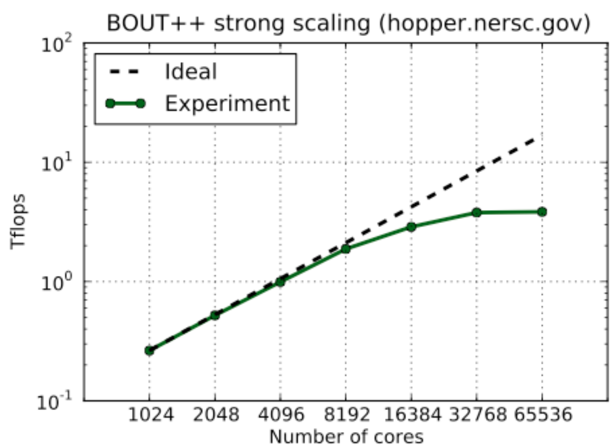
```
./ex55 -ksp_type fgmres -pc_type mg
-mg_levels_ksp_type fgmres
-mg_levels_ksp_max_it 2
-mg_levels_pc_type fieldsplit
-mg_levels_pc_fieldsplit_detect_saddle_point
-mg_levels_pc_fieldsplit_type schur
-mg_levels_pc_fieldsplit_factorization_type full
-mg_levels_pc_fieldsplit_schur_precondition user
-mg_levels_fieldsplit_0_ksp_type preonly
-mg_levels_fieldsplit_0_pc_type sor
-mg_levels_fieldsplit_0_pc_sor_forward
-mg_levels_fieldsplit_1_ksp_type gmres
-mg_levels_fieldsplit_1_pc_type none
-mg_levels_fieldsplit_ksp_max_it 5
-mg_coarse_pc_type svd
```

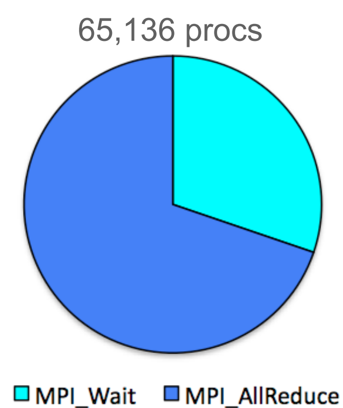
Outline

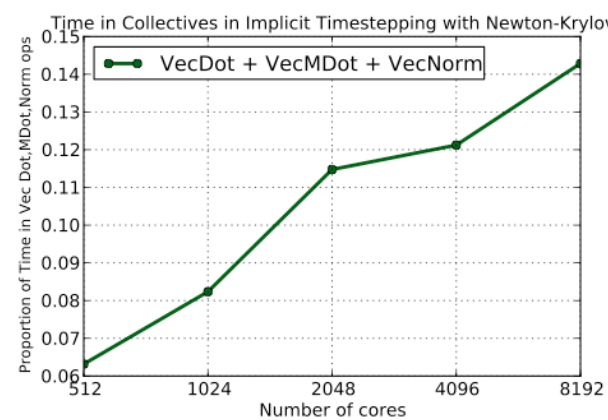
- **Multiphysics Challenges**
  - Background
  - Outcomes of 2011 ICiS multiphysics workshop
- **PETSc Composable Solvers**
  - PCFieldSplit for multiphysics support
    - Core-edge fusion
    - Heterogeneous materials science
  - Hierarchical Krylov methods for extreme-scale
    - Reacting flow
- **Conclusions**

Goal for hierarchical and nested Krylov methods: Scaling to millions of cores

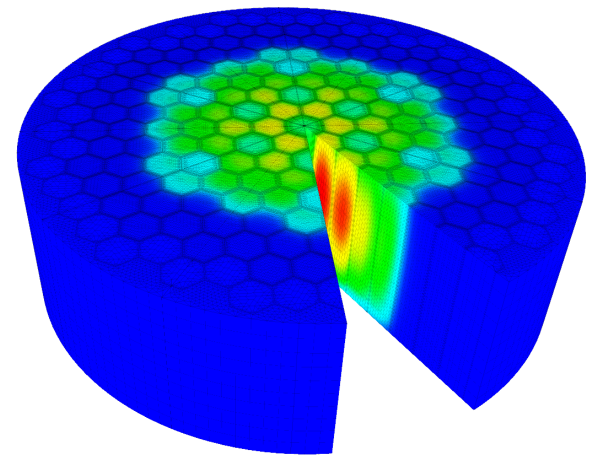
- **Orthogonalization is mathematically very powerful.**
  - Krylov methods often highly accelerate convergence of simple iterative schemes for sparse linear systerms (that do not have highly convergent multigrid or other specialized fast solvers).
- **But conventional Krylov methods encounter scaling difficulties on over 10k cores**
  - Require at least 1 global synchronization (via MPI_Allreduce) per iteration
  - Example: BOUT++ (fusion boundary turbulence, LLNL and Univ of York)
    - each timestep: Solve *F(u) = 0* using matrix-free Newton-Krylov
    - significant scalability barrier: MPI AllReduce() calls (in GMRES)
    - graphs courtesy of A. Koniges and P. Narayanan, LBNL







Hierarchical Krylov methods: Inspired by neutron transport success



- **UNIC exploits natural hierarchical block structure of energy groups (D. Kaushik, M. Smith, et. al)**
  - Scales well to over 290,000 cores of IBM Blue Gene/P and Cray XT 5, Gordon Bell finalist in 2009
    - outer solve: Flexible GMRES
    - inner solve: independent preconditioned CG solves
- **Hierarchical Krylov strategy**
  - Decrease reductions across entire system (where expensive)
  - Freely allow reductions across small subsets of the entire system (where cheap)

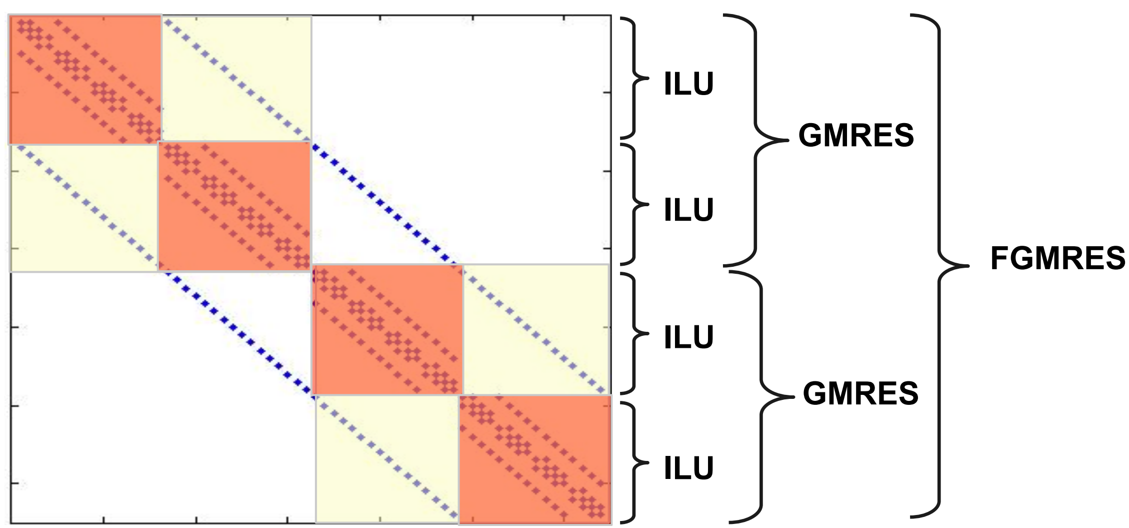
Hierarchical vector partitioning

Level	Parent vector, x, and child subvectors				Cores per (sub)vector
0	x, dim=64				4
1	$x_{(1)}$, dim=32		$x_{(2)}$, dim=32		2
2	$x_{(1,1)}$, dim=16	$x_{(1,2)}$, dim=16	$x_{(2,1)}$, dim=16	$x_{(2,2)}$, dim=16	1

Hierarchical Krylov approach to solve Ax = b

Two-level hierarchical FGMRES (**h-FGMRES**) with GMRES iterative preconditioner:

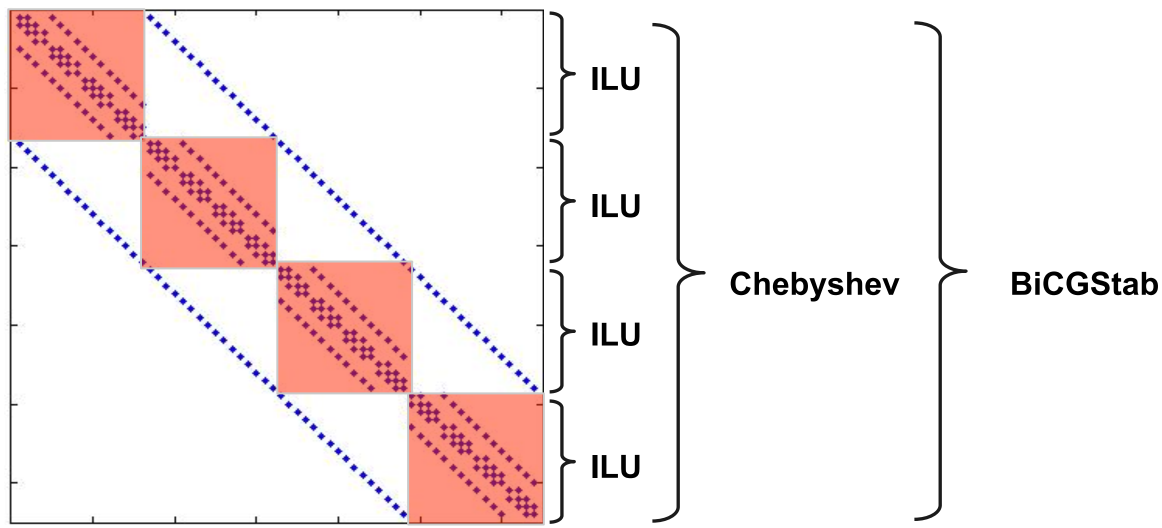
- Partition initial vector x_0 into a hierarchy; partition total cores with same hierarchy.
- Map the vector partition into the core partition.
- For j=0,1,… until convergence (on all cores) do
  - one FGMRES iteration to x_j (over the global system) with preconditioner:
  - For k=0,1,…,inner_its (concurrently on subgroups of cores) do
    - For each child of x_j, one GMRES + bjacobi + ilu(0)
  - End for
- End for



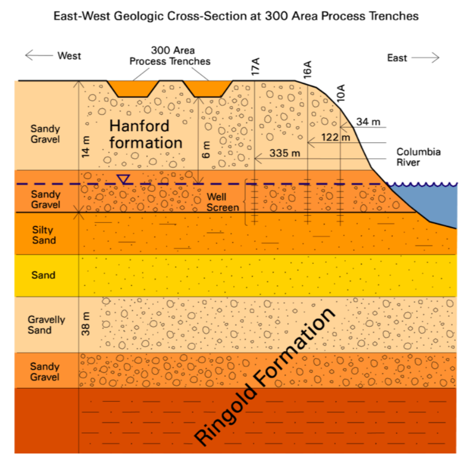
Nested Krylov approach to solve Ax = b

Nested BiCGStab
with Chebyshev linear iterative preconditioner:

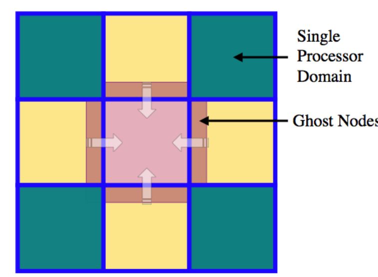
- For j=0,1,… until convergence (on all cores) do
  - One BiCGStab iteration over the global system with preconditioner:
  - For k=0,1,…,inner_its (on all cores) do
    - One Chebyshev + bjacobi + ilu(0)
  - End for
- End for



SNES usage: Reactive groundwater flow & transport



- **SciDAC project: PFLOTRAN**
  - PI P. Lichtner (LANL)
  - https://software.lanl.gov/pflotran
  - **Overall goal:** Continuum-scale simulation of multiscale, multiphase, multicomponent flow and reactive transport in porous media; applications to field-scale studies of geologic CO2 sequestration, contaminant migration
- **Model:** Fully implicit, finite volume discretization, multiphase flow, geochemical transport
  - Initial TRAN by G. Hammond for DOE CSGF practicum
  - Initial FLOW by R. Mills for DOE CSGF practicum
  - Initial multiphase modules by P. Lichtner and C. Lu



- **PETSc usage:** Preconditioned Newton-Krylov algorithms + parallel structured mesh management (B. Smith)

PFLOTRAN governing equations

Mass conservation: flow equations

$$\frac{\partial}{\partial t}(\phi s_\alpha \rho_\alpha X_i^\alpha) + \nabla \cdot \left[q_\alpha \rho_\alpha X_i^\alpha - \phi s_\alpha D_i^\alpha \rho_\alpha \nabla X_i^\alpha\right] = Q_i^\alpha$$

$$q_\alpha = -\frac{k k_\alpha}{\mu_\pi} \nabla(p_\alpha - W_\alpha \rho_\alpha g z) \qquad p_\alpha = p_\beta - p_{c,\alpha\beta}$$

Energy conservation equation

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \rho_\alpha U_\alpha + (1-\phi)\rho_r c_r T\right] + \nabla \cdot \left[\sum_\alpha q_\alpha \rho_\alpha H_\alpha - \kappa \nabla T\right] = Q_e$$

Multicomponent reactive transport equations

$$\frac{\partial}{\partial t}\left[\phi \sum_\alpha s_\alpha \Psi_j^\alpha\right] + \nabla \cdot \left[\sum_\alpha \Omega_\alpha\right] = -\sum_m v_{jm} I_m + Q_j$$

Total concentration

$$\Psi_j^\alpha = \delta_{\alpha l} C_j^\alpha + \sum_i v_{ji} C_i^\alpha$$

Total solute flux

$$\Omega_j^\alpha = (-\tau \phi s_\alpha D_\alpha \nabla + q_\alpha)\Psi_j^\alpha$$

Mineral mass transfer equation

$$\frac{\partial \phi_m}{\partial t} = V_m I_m \qquad \phi + \sum_m \phi_m = 1$$

PDEs for PFLOW and PTRAN have general form

$$\frac{\partial A}{\partial t} + \nabla \bullet F = s$$

Dominant computation of each can be expressed as:

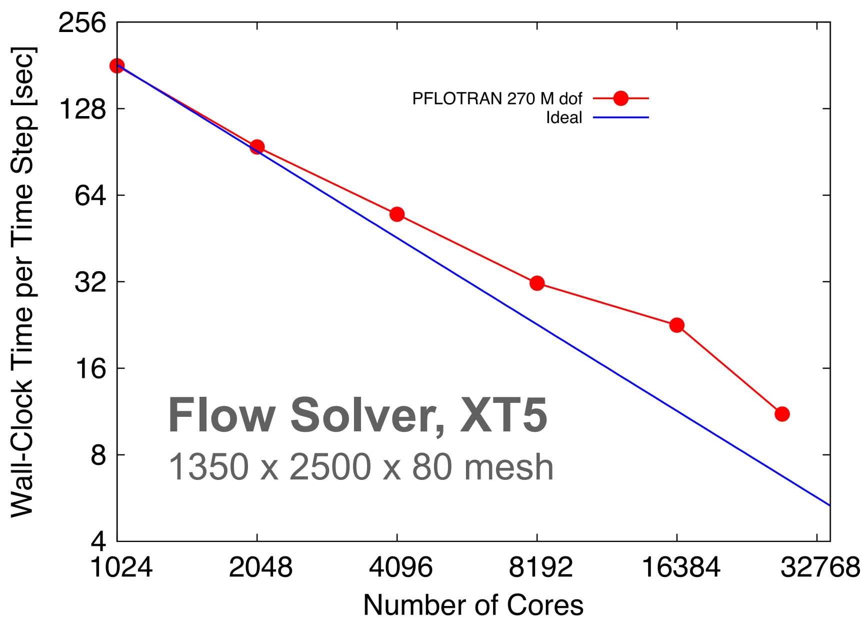
Solve F(u) = 0

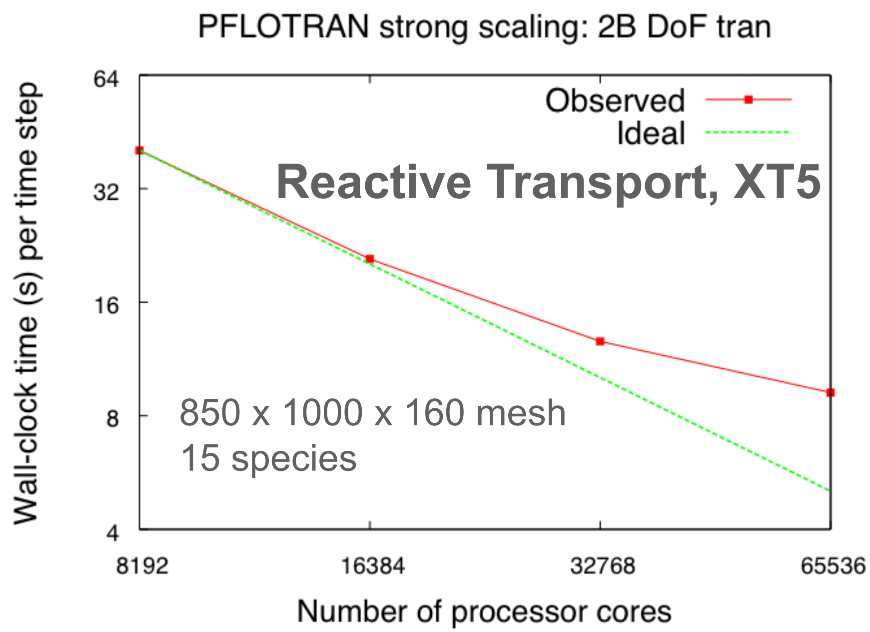
Cray XK6 at ORNL:

https://www.olcf.ornl.gov/computing-resources/titan/

- 18,688 (node) *16 (cores) = 299,008 processor cores
- 3.3 petaflops peak performance
- 2,300 trillion calculations each second. The same number of calculations would take an individual working at a rate of one per second more than 70 million years

SNES / PFLOTRAN scalability





- Results courtesy of R. Mills (ORNL)
- Inexact Newton w. line search using BiCGStab + Block Jacobi/ILU(0)
- PETSc/SNES design facilitates algorithmic research: reduced synchronization BiCGStab
- Additional simulations using up to 2 billion unknowns

Hierarchical FGMRES (ngp>1) and GMRES (ngp=1)
PFLOTRAN Case 1

Number of Cores	Groups of Cores	Time Step Cuts	% Time for Inner Products	Total Linear Iterations	Execution Time (sec)
4,096 512x512x512	1 **64**	0 **0**	39 **23**	11,810 **2,405**	146.5 **126.7**
32,768 (1024x1024x1024)	1 **128**	1 **0**	48 **28**	35,177 **5,244**	640.5 **297.4**
98,304 (1024x1024x1024)	1 **128**	7 **0**	77 **47**	59,250 **6,965**	1346.0 **166.2**
160,000 (1600x1600x640)	1 **128**	9 **0**	72 **51**	59,988 **8,810**	1384.1 **232.2**

./pfltran -pflotranin <pflotran_input>
-flow_ksp_type fgmres -flow_pc_type bjacobi -flow_pc_bjacobi_blocks ngp
-flow_sub_ksp_type gmres -flow_sub_ksp_max_it 6 -flow_sub_pc_type bjacobi
-flow_sub_sub_pc_type ilu

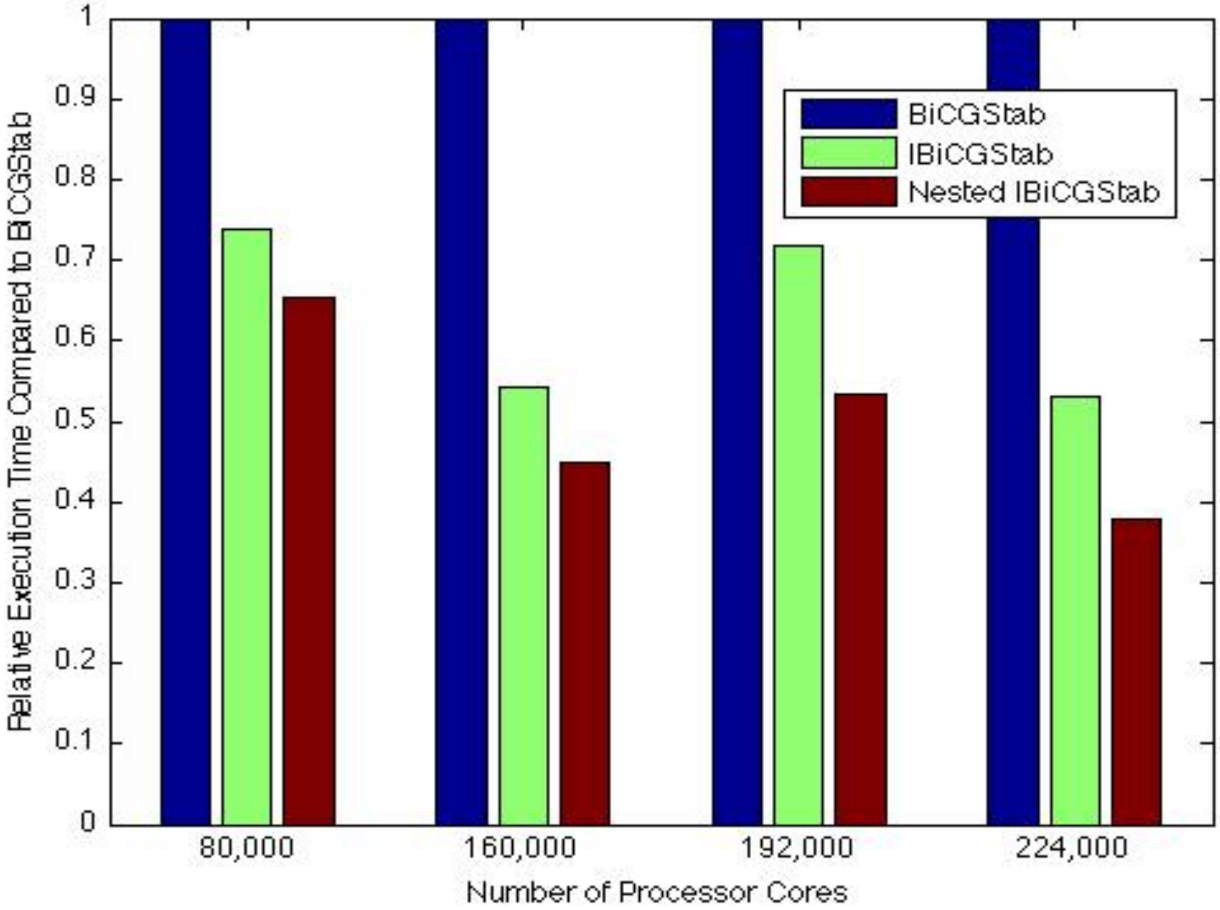
BiCGS, IBiCGS and Nested IBiCGS

Case 2, 1600x1632x640 mesh

Number of Cores	BiCGS Outer Iters	BiCGS Time (sec)	IBiCGS Outer Iters	IBiCGS Time (sec)	Nested IBiCGS Outer Iters	Nested IBiCGS Time (sec)
80,000	4304	81.4	4397	60.2	**1578**	**53.1**
160,000	4456	71.3	4891	38.7	**1753**	**31.9**
192,000	3207	41.5	4427	29.8	**1508**	**22.2**
224,000	4606	55.2	4668	29.3	**1501**	**20.9**

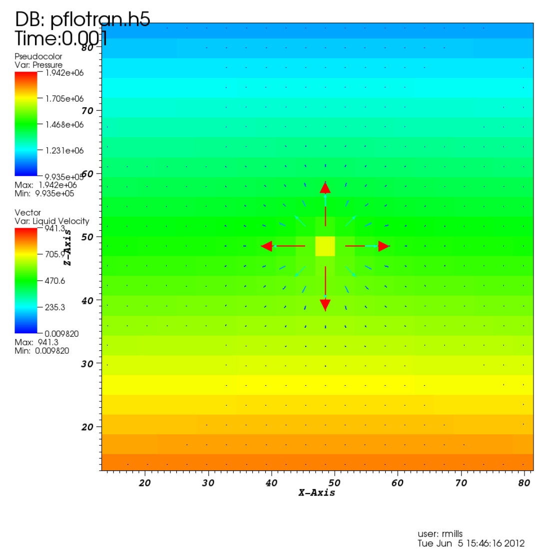
```
./pfltran -pflotranin <pflotran_input>
-flow_ksp_type ibcgs -flow_ksp_pc_side right -flow_pc_type ksp
-flow_ksp_ksp_type chebyshev -flow_ksp_ksp_chebychev_estimate_eigenvalues 0.1,1.1
-flow_ksp_ksp_max_it 2 -flow_ksp_ksp_norm_type none -flow_ksp_pc_type bjacobi
-flow_ksp_sub_pc_type ilu
```

Nested IBiCGStab/Chebyshev in PFLOTRAN



Hierarchical and nested Krylov approaches

- **Fewer global inner products & stronger preconditioners greatly improved PFLOTRAN performance on 10,000 through 224,000 cores of Cray XK6**
  - Hierarchical FGMRES (h-FGMRES) method with GMRES preconditioner
  - Nested BiCGStab method with a Chebyshev preconditioner
- Much research remains
  - Understanding details of performance at each level
  - Additional flexible Krylov methods
  - Physics-based partitioning

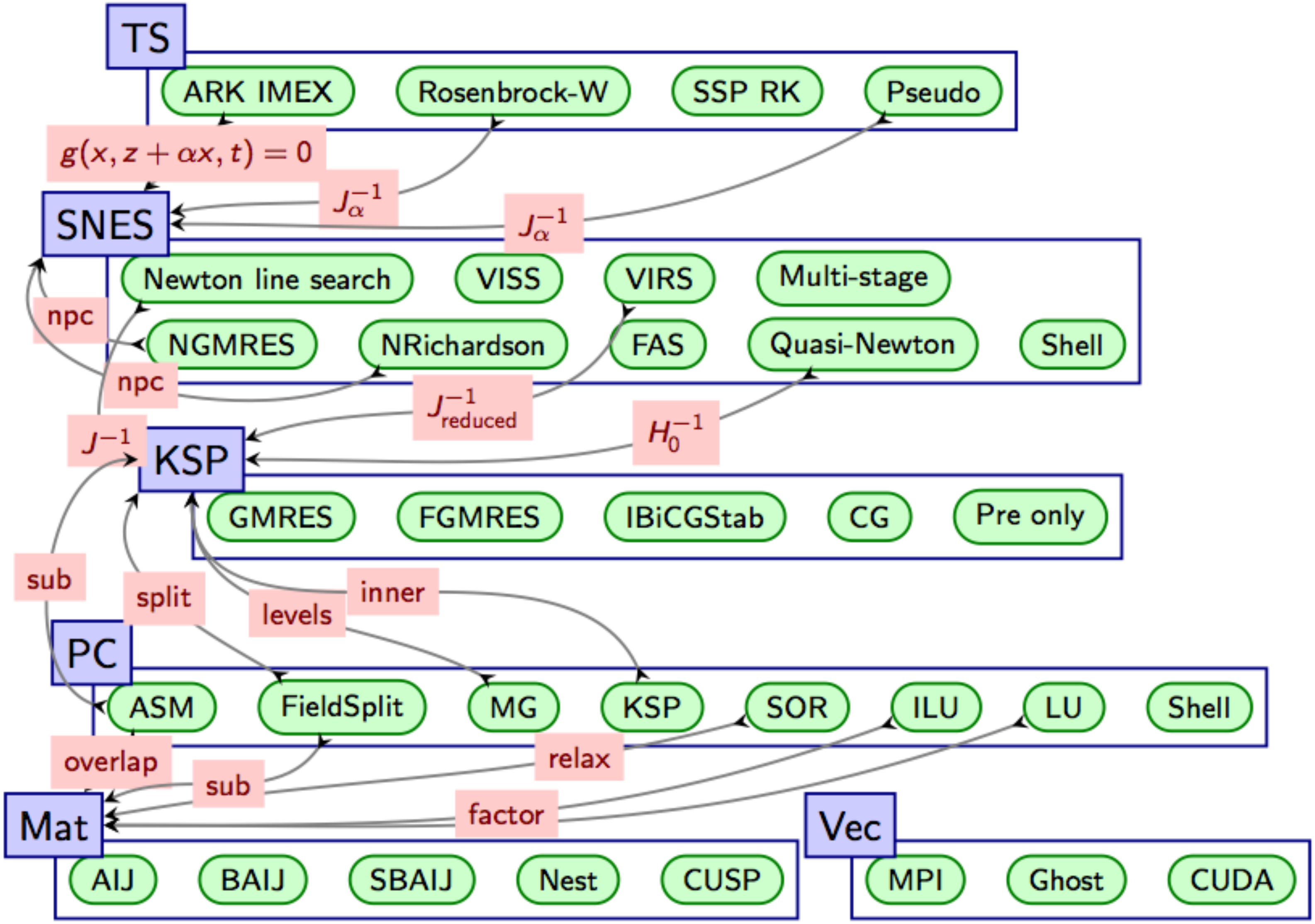


Reference: ***Hierarchical and Nested Krylov Methods for Extreme-Scale Computing***, L. C. McInnes, B. Smith, H. Zhang and R. T. Mills, preprint ANL/MCS-P2097-0512, Argonne National Laboratory, 2012.

PETSc: Composable approach for designing efficient, robust, scalable solvers

- **There is no single, best, black-box solver.**
- **Provide a uniform way to compose solvers from various pieces without changing the application code**
  - Algorithmic choices at runtime
  - Focus of today's presentation:
    - Linear solvers for multiphysics
    - Hierarchical Krylov methods for extreme-scale
  - Also new capabilities in nonlinear solvers (SNES)
    - Avoiding sparse matrices (memory bandwidth limited)
    - FAS (nonlinear multigrid), nonlinear GMRES, etc.
  - Also new capabilities in timestepping solvers (TS)
    - Balancing algorithmic convergence with data locality

Interactions among composable linear, nonlinear, and timestepping solvers



PETSc composable solvers

- **Multipronged approach: no optimality without interplay among physics, algorithms, and architectures**
  - Application support
  - Mathematical understanding
  - Customizable general-purpose software
- Further information

  http://www.mcs.anl.gov/petsc

Lots of multiphysics challenges and opportunities

Outcomes of 2011 ICiS multiphysics workshop:

- https://sites.google.com/site/icismultiphysics2011/
- including workshop report that addresses:

- **Practices and Perils in Multiphysics Applications**
  - Examples, cross-cutting issues
- **Algorithms for Multiphysics Coupling**
  - Linear and nonlinear solvers
  - Continuum-continuum coupling: temporal and spatial discretizations
  - Continuum-discrete coupling
  - Error estimation, uncertainty quantification
- **Multiphysics Software**
  - Current practices, common needs, successes
  - Challenges in design, difficulties in collaborative multiphysics software
- **Opportunities for Multiphysics Simulation Research**
  - Extreme-scale: Requires fundamentally rethinking approaches with attention to data motion, data structure conversion, and overall application design
- **Insertion Paths for Algorithms and Software in Multiphysics Applications**
- **Multiphysics Exemplars and Benchmarks** – modest start

Thanks again to:

- **U.S. Department of Energy – Office of Science**
- **Collaborators**
  - D. Keyes, C. Woodward and all ICiS multiphysics workshop participants
  - S. Abhyankar, S. Balay, P. Brune, J. Brown, E. Constantinescu, D. Karpeev, M. Knepley, B. Smith, H. Zhang, other PETSc contributors
  - M. Anitescu, M. McCourt, S. Farley, J. Lee, T. Munson, B. Norris, L. Wang (ANL)
  - Scientific applications teams, especially A. El Azab, J. Cary, A. Hakim, S. Kruger, R. Mills, A. Pletzer, T. Rognlien
- **All PETSc users**

Office of Science
U.S. DEPARTMENT OF ENERGY

SciDAC
Scientific Discovery through Advanced Computing