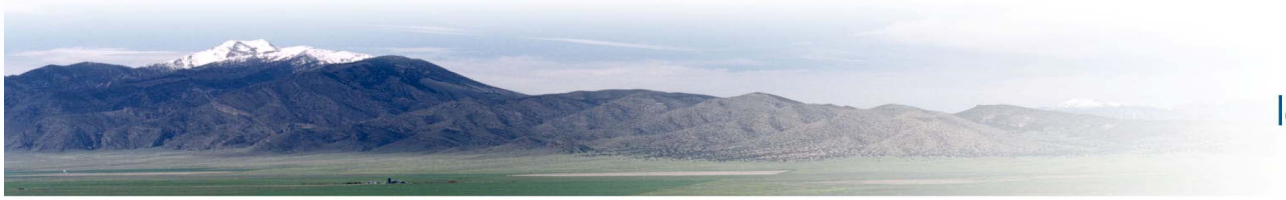
Parallel Algorithms and Software for Multiphysics Computational Nuclear Engineering

Dana Knoll and the Multiphysics Methods Group
Nuclear Science and Engineering Division
Idaho National Laboratory

Oregon St. Univ., Oct. 21, 2008



Idaho National Laboratory INL

Outline

- Multiphysics Computational Nuclear Engineering
- Operator Splitting and Analysis *
- Jacobian-Free Newton-Krylov (JFNK) Methods*
- Physics-based Preconditioning of JFNK*
- MOOSE, Multiphysics Object-Oriented Software Environment
- BISON, Fuel Performance Simulation
- PRONGHORN, Pebble-bed Gas Reactor Simulation

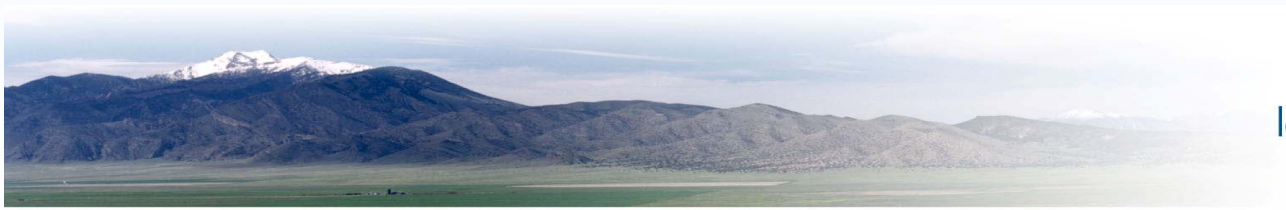
* *Each topic a talk of its own*

Some Requirements for Predictive Simulation

- Proper Physics Model / Equation Set
- Accurate spatial discretization / adequate grid refinement (3-D)
- High end computing platforms and parallel, scalable, algorithms.
- Accurate time integration (or coupling) for multiphysics systems
  - This area has received little attention.
  - Verification of temporal integration (confidence)
  - We must understand this area when we are taking $10^5 - 10^6$ time steps.
  - ”Code coupling” or first-order operator splitting the norm

Multiphysics Computational Nuclear Engineering: Two Examples

- Fuel Performance
  - Nonlinear Mechanics, Contact, Crack models
  - Thermal transport
  - Fission gas transport
  - Species transport
- Reactor design and transient analysis
  - Neutronics
  - Heat conduction
  - Fluid flow
  - Species transport and chemistry
  - Structural response



Linearization and Time Splitting (code coupling): The Standard Approach

- Model Problem: A and B represent two nonlinear processes

$$\frac{dT}{dt} = A(T)T + B(T)T \tag{1}$$

- Linearization

$$\frac{T^{n+1} - T^n}{\Delta t} = A(T^n)T^{n+1} + B(T^n)T^{n+1} \tag{2}$$

- Time Splitting (first-order), Same as ”code coupling”.

$$\frac{T^* - T^n}{\Delta t} = A(T^n)T^*, \quad \frac{T^{n+1} - T^*}{\Delta t} = B(T^n)T^{n+1} \tag{3}$$

- Both are sources of time integration error

Modified Equation Analysis (1 of 2)

- Modified Equation Analysis (MEA) example ($T_t \equiv \frac{dT}{dt}$)

$$T_t = T\,, \quad \frac{T^{n+1} - T^n}{\Delta t} = T^{n+1}\,. \tag{4}$$

- Taylor series used to eliminate T^n

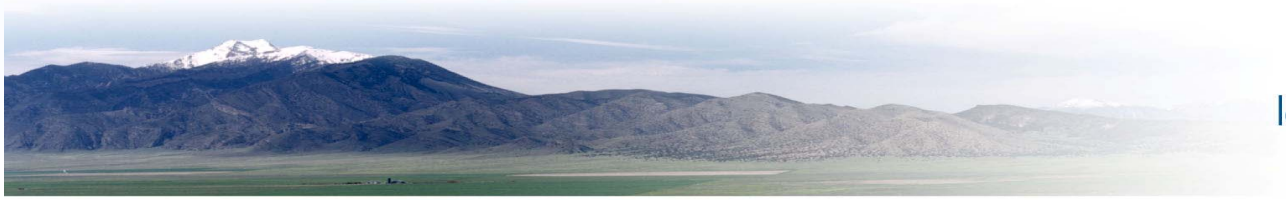
$$T^n = T^{n+1} - \Delta t T_t^{n+1} + \frac{\Delta t^2}{2} T_{tt}^{n+1} - \frac{\Delta t^3}{6} T_{ttt}^{n+1} + O(\Delta t^4)$$

- Original ODE on left side "modified" by nonzero RHS.

$$[T_t - T] = \frac{\Delta t}{2} T_{tt} + \mathcal{O}(\Delta t^2), \tag{5}$$

Modified Equation Analysis (2 of 2)

- History:
  - Nonlinear stability, Hirt, JCP, vol. 2, pp. 339-355 (1968)
  - Accuracy, Warming and Hyett, JCP, vol. 14, pp. 159-179 (1974)
- We will use semi-discrete (time) MEA to begin to analyze numerical errors which arise due to linearization and time splitting.
- D.A. Knoll, et. al., ”On balanced approximations for the time integration of multiple time scale systems”, *J. Comput. Phys.*, vol 185, pp. 583-611 (2003)



Splitting Linear Reaction - Diffusion (1 of 5)

- Model Linear Equation:

$$\frac{\partial T}{\partial t} - D\frac{\partial^2 T}{\partial x^2} = \alpha T, \quad T_{x=0} = T_L, \quad T_{x=1} = T_R$$

- Dynamical Time Scale

$$\frac{1}{\tau_{dyn}} \equiv -(\frac{1}{T}\frac{dT}{dt}) = -\frac{D}{T}\frac{\partial^2 T}{\partial x^2} - \alpha \approx \frac{1}{\tau_{dif}} + \frac{1}{\tau_{reac}},$$

- Normal Mode Time Scales

$$\tau_{dif} \equiv \frac{L^2}{D} \; ; \; \tau_{reac} \equiv -\frac{1}{\alpha},$$

and L is taken as the length of the domain (or solution structure).

Splitting Linear Reaction - Diffusion (2 of 5)

- First order, unsplit:

$$\frac{T^{n+1} - T^n}{\Delta t} - D\frac{\partial^2 T^{n+1}}{\partial x^2} = \alpha T^{n+1}$$

$$T^{n+1}_{x=0} = T_L, \quad T^{n+1}_{x=1} = T_R$$

- First order split (R-D):

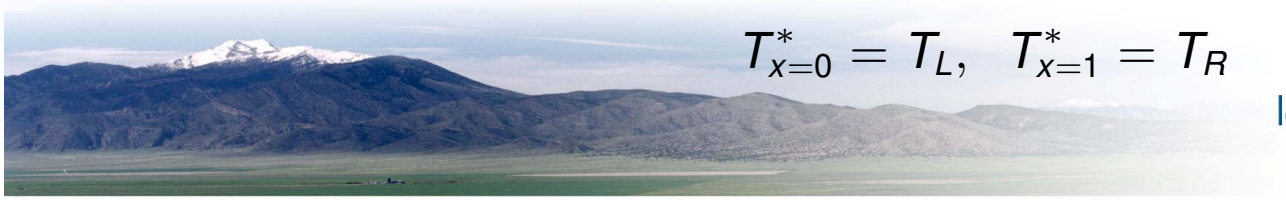
$$\frac{T^* - T^n}{\Delta t} = \alpha T^*, \quad \frac{T^{n+1} - T^*}{\Delta t} - D\frac{\partial^2 T^{n+1}}{\partial^2 x} = 0$$

$$T^{n+1}_{x=0} = T_L, \quad T^{n+1}_{x=1} = T_R$$

- First order split (D-R):

$$\frac{T^* - T^n}{\Delta t} - D\frac{\partial^2 T^*}{\partial^2 x} = 0, \quad \frac{T^{n+1} - T^*}{\Delta t} = \alpha T^{n+1}$$

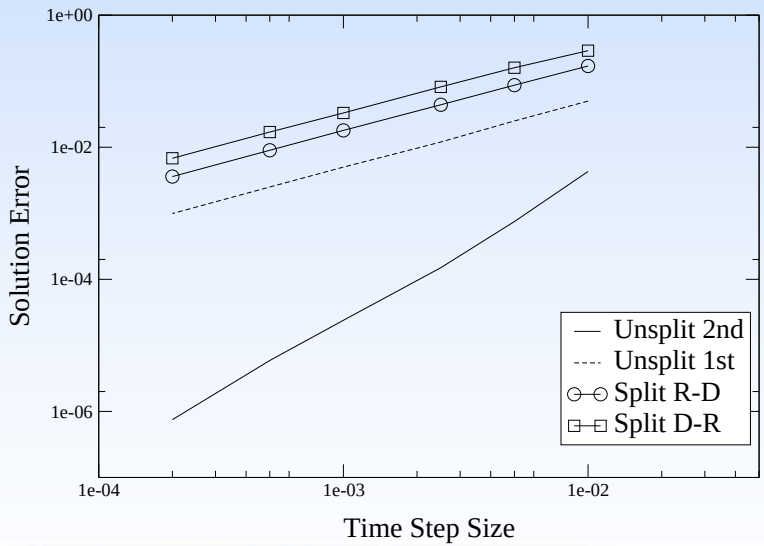
$$T^*_{x=0} = T_L, \quad T^*_{x=1} = T_R$$

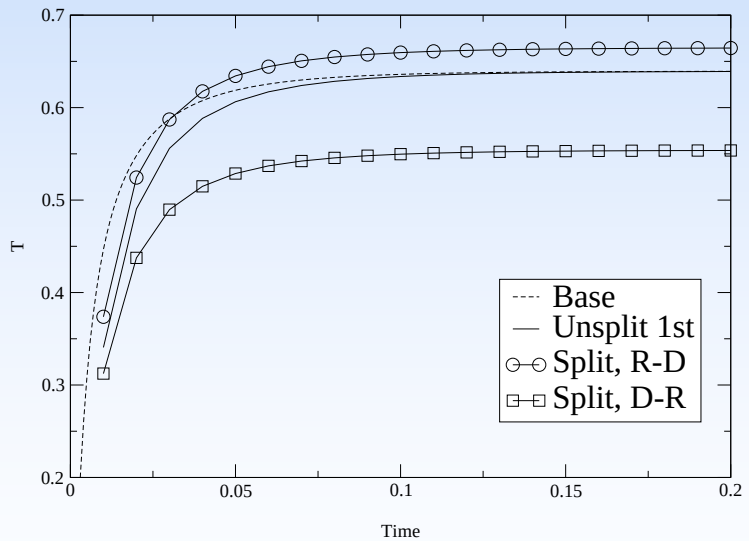


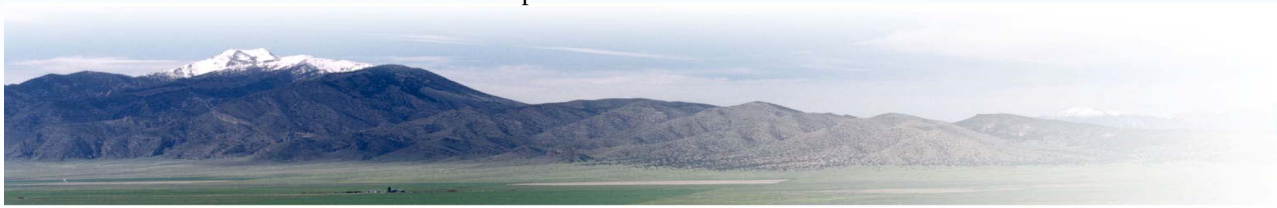
Splitting Linear Reaction - Diffusion (3 of 5)

$$T(t=0)=0, D=1, \alpha=-20, T_L=1, T_R=0,$$

$$\Delta t = 0.01, \alpha\Delta t = -0.2$$







Idaho National Laboratory INL

Splitting Linear Reaction - Diffusion (4 of 5)

- MEA on First order, unsplit, Yields:

$$[T_t - D\frac{\partial^2 T}{\partial x^2} - \alpha T] = \frac{\Delta t}{2} T_{tt} + O(\Delta t^2)$$

- MEA on R-D splitting Yields:

$$[T_t - D\frac{\partial^2 T}{\partial x^2} - \alpha T] = \frac{\Delta t}{2} T_{tt} + \alpha \Delta t D \frac{\partial^2 T}{\partial x^2} + O(\Delta t^2)$$

- MEA on D-R splitting Yields:

$$[T_t - D\frac{\partial^2 T}{\partial x^2} - \alpha T] = \frac{\Delta t}{2} T_{tt} + \alpha \Delta t D \frac{\partial^2 T}{\partial x^2} + O(\Delta t^2)$$

$$T_{x=0} = T_L/(1 - \alpha \Delta t), \quad T_{x=1} = T_R/(1 - \alpha \Delta t)$$

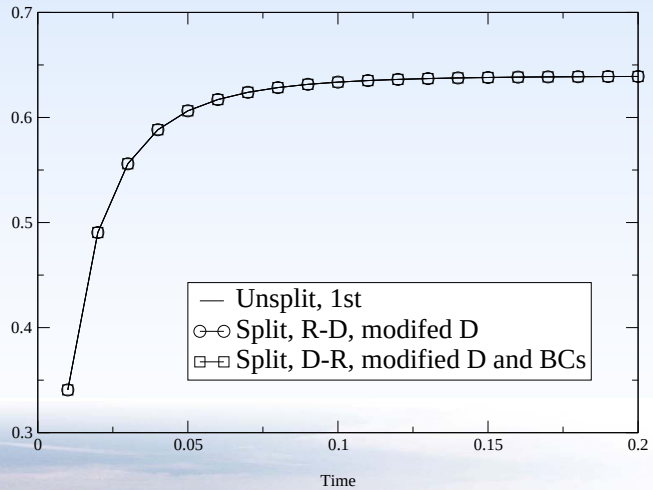
- Splitting errors scale with $\alpha \Delta t$, normal mode.

Splitting Linear Reaction - Diffusion (5 of 5)

- Use D^* in R-D splitting and equate the resulting modified equation to the modified equation from first order unsplit to get:

$$D^* = \frac{D}{1.0 - \alpha \Delta t}$$

- What is the result of using D^* in R-D split simulation ? and What is the result of using D^* and modified BCs in D-R split simulation ?



Linearization and Time Splitting: Some Conclusions

- Can appear to be inaccurate physics model under validation process (Dangerous !)
- Boundary conditions can be impacted
- Time step constraints for accuracy related to normal modes
- Second-order splitting is possible, but continues to be prone to numerical stability and accuracy challenges.
  - Numerical Stability Issues
    D.L Ropp and J.N. Shadid, *J. Comput. Phys.*, vol. 203, pp. 449-466 (2005)
  - Asymptotic Range / Time Step Size
    R.M., Rauenzahn, V.A. Mousseau, D.A. Knoll, *Comput. Phys. Comm.* vol. 172, pp. 109-118 (2005)
    V.A. Mousseau, D.A. Knoll, *Nuc. Sci. Eng.*, vol. 154, pp. 174-189 (2006)

Jacobian-Free Newton-Krylov Methods: A Modern Option

- **Pros:**
  - No splitting or linearization error
  - A clean way to include a variety of nonlinear phenomena
  - Can implement a variety of higher-order time discretizations
  - Opens the door for Sensitivity analysis, data assimilation ....
- **Cons:**
  - Solving Nonlinear problem with iteration
  - Solving full dimensional system implicitly
  - MUST produce effective preconditioner

Newton's Method : Equations

Newton's method solves a system of nonlinear equations of the form,

$$\mathbf{F}(\mathbf{x}) = 0,$$

by a sequence of steps defined by (each a linear problem)

$$\mathbf{J^k}(\mathbf{x}^k)\delta\mathbf{x}^k = -\mathbf{F}(\mathbf{x}^k),$$

where,

$$\mathbf{J}_{i,j}(\mathbf{x}^k) = \frac{\partial \mathbf{F}_i}{\partial \mathbf{x}_j^k},$$

and,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x}^k.$$

This is continued until

$$\|\mathbf{F}(\mathbf{x}^k)\| < tol_n \|\mathbf{F}(\mathbf{x}^0)\|$$

Jacobian-Free Newton-Krylov: Key Point

Krylov linear iterative methods (CG, GMRES ...) only need the action of the Jacobian matrix to construct the m^{th} iteration of $\delta\mathbf{x}^k$

$$\delta\mathbf{x}_m^k = a_0\mathbf{r}_0 + a_1\mathbf{J}\mathbf{r}_0 + a_2\mathbf{J}^2\mathbf{r}_0 + \ldots + a_m\mathbf{J}^m\mathbf{r}_0$$

where

$$\mathbf{r}_0 = \mathbf{J}^k\delta\mathbf{x}_0^k + \mathbf{F}(\mathbf{x}^k)$$

and

$$\delta\mathbf{x}_0^k = -\mathbf{P}^{-1}\mathbf{F}(\mathbf{x}^k)$$

where $\mathbf{P}^{-1}$ is the preconditioner.
The action of the Jacobian Matrix can be approximated with a single function evaluation

$$\mathbf{J}\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{x} + \epsilon\mathbf{v}) - \mathbf{F}(\mathbf{x})}{\epsilon}$$

Jacobian-Free Newton-Krylov: Key Refs

- Standard “PDE motivated” references:
  P. N. Brown and Y. Saad, *SIAM J. Sci. Stat. Comput.*, **11**, pp. 450-481 (1990)
  Tony F. Chan and Kenneth R. Jackson, *SIAM J. Sci. Stat. Comput.*, **5**, pp. 533-542 (1984)
- Also see the monograph,
  C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, 1995
- Recent JFNK review article from the application perspective
  D.A. Knoll and D.E. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, *J. Comput. Phys.*, **193**, pp. 357-397 (2004)

Preconditioning JFNK (1 of 2)

- Using right preconditioning one solves

$$(\mathbf{J}\mathbf{P}^{-1})(\mathbf{P}\delta\mathbf{x}) = -\mathbf{F}(\mathbf{x}). \tag{6}$$

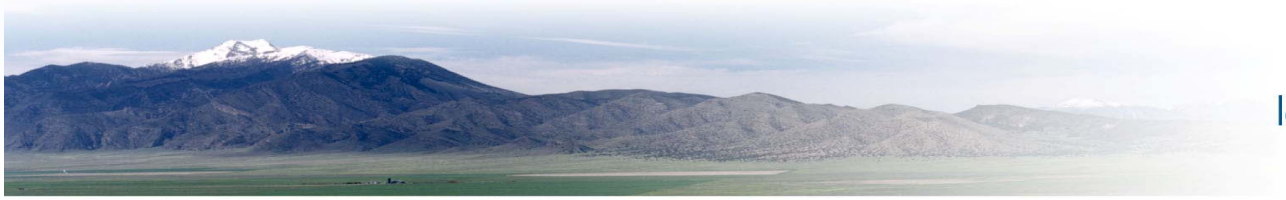
  P symbolically represents the preconditioning matrix (or process) and $\mathbf{P}^{-1}$ the inverse of preconditioning matrix.
- Actually realized through a two-step process. First solve

$$(\mathbf{J}\mathbf{P}^{-1})\mathbf{w} = -\mathbf{F}(\mathbf{x}), \tag{7}$$

  for **w** with the Krylov method. Then solve for $\delta\mathbf{x}$

$$\delta\mathbf{x} = \mathbf{P}^{-1}\mathbf{w}, \tag{8}$$

- Operationally only require the action of $\mathbf{P}^{-1}$ on a vector.

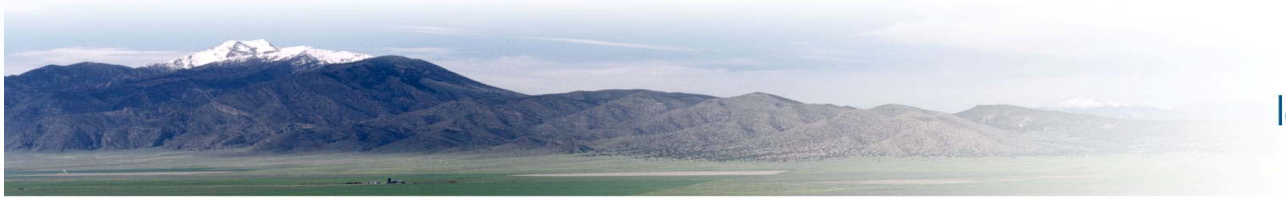


Preconditioning JFNK (2 of 2)

- Right-preconditioned matrix-free version is:

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{v} \approx [\mathbf{F}(\mathbf{x} + \epsilon\mathbf{P}^{-1}\mathbf{v}) - \mathbf{F}(\mathbf{x})] \; / \; \epsilon. \tag{9}$$

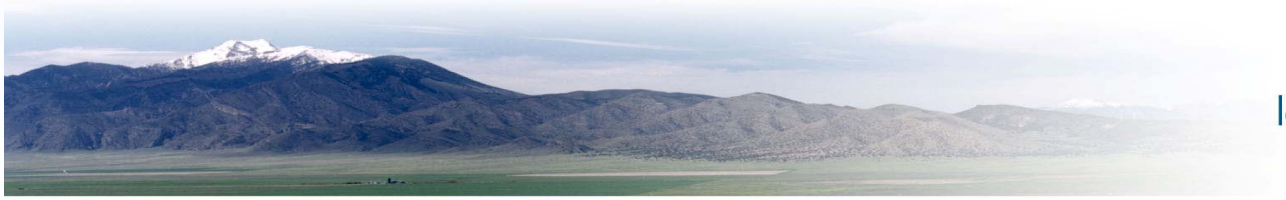
- Required in step 1 of previous slide
- Actually done in two steps ($\mathbf{v}$ is given):
  1. Preconditioning: Solve (approximately) for $\mathbf{y}$ in $\mathbf{y} = \mathbf{P}^{-1}\mathbf{v}$.
  2. Perform matrix-free product $\mathbf{J}\mathbf{y} \approx [\mathbf{F}(\mathbf{x} + \epsilon\mathbf{y}) - \mathbf{F}(\mathbf{x})] \; / \; \epsilon$.
- Only the matrix elements required for the action of $\mathbf{P}^{-1}$ are formed.



Physics-based Preconditioning: Motivation

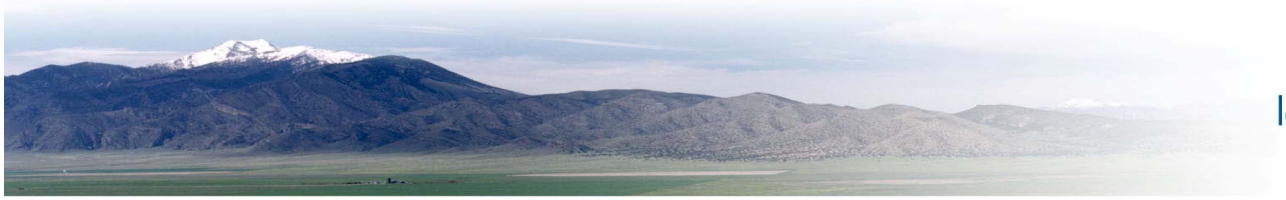
What is an effective choice for P^{-1} ?

- There exist numerous, legacy algorithms to solve multiphysics systems.
- Most are based on linearization and time splitting
- Typically developed with some insight into the time scales or physical behavior of the problem.
- As a benefit of this insight, a reduced implicit system, or a sequence of segregated implicit systems may be solved in place of the fully coupled system.
- We have shown that these legacy algorithms may make excellent preconditioners for JFNK.



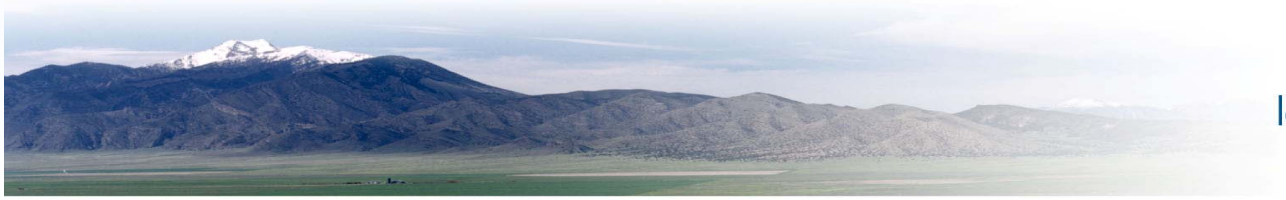
Physics-based Preconditioning: Examples

- Nonequilibrium Radiation Diffusion (Applied Astrophysics)
  V.A. Mousseau, D. A. Knoll and W.J. Rider, *J. Comput. Phys.*, **160**, pp. 743-765 (2000)
- Magnetohydrodynamics (Fusion and Space Weather)
  L. Chacon, D.A. Knoll, and J.M. Finn, *J. Comput. Phys.*, **178**, pp. 15-36 (2002)
- Low speed flow (Hurricane Simulation)
  J.M. Reisner, V.A. Mousseau, A. Wyszogrodzki and D.A. Knoll, *Mon. Wea. Rev.*, **133**, pp. 1003-1022 (2005)
- Solidifying flow (Metal Casting)
  K.J. Evans, D. A. Knoll and M.A. Pernice, *J. Comput. Phys.*, **219**, pp. 404-417 (2006)



Dual use of the method

- Not JFNK vs Linearization / Time Splitting
- Linearization / time splitting can be used as solver or preconditioner within a single code.
- Our view: Combination of JFNK and linearization plus time splitting is the most dependable route to time accurate (2^{nd} order), stable, and efficient methods.

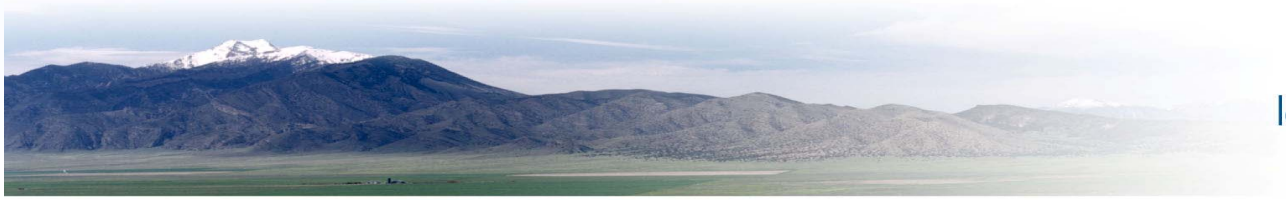


Multiphysics Framework Requirements

- 1D, 2D and 3D with same code
- Massively Parallel
- Fully Coupled
- Fully Implicit
- JFNK Finite Element Based
- Flexible Physics Interface
  - Ability to rapidly develop new capabilities
- Flexible Materials Database
- Error Estimation and Adaptivity
- Portable

MOOSE: *Multiphysics Object Oriented Simulation Environment*

- Meets all of the above requirements and more.

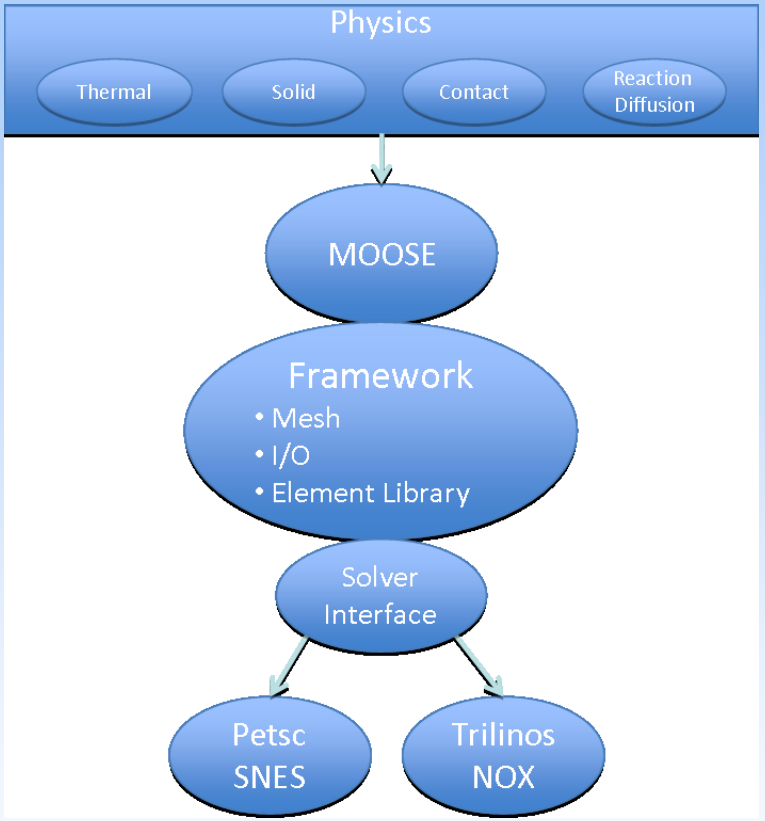


Parallel Algorithms

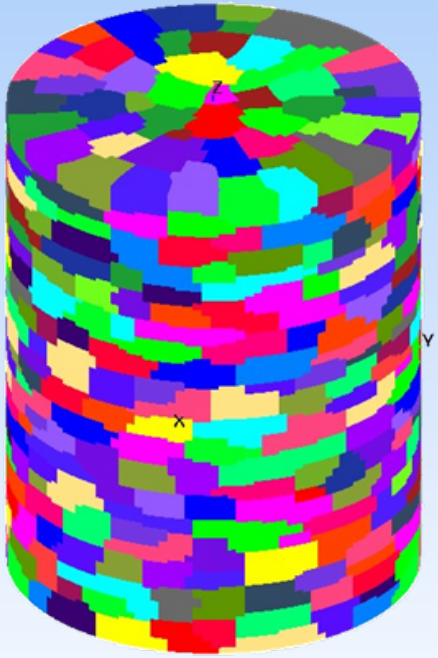
- First and second order Lagrange finite element discretization.
- Residual is calculated in parallel.
  - Off processor contributions assembled in parallel.
  - Element based domain decomposition.
- Parallel solver packages used for JFNK and elliptic system preconditioning.
  - PETSc, ANL
  - Trilinos, SNL

MOOSE – Multiphysics Object Oriented Simulation Environment

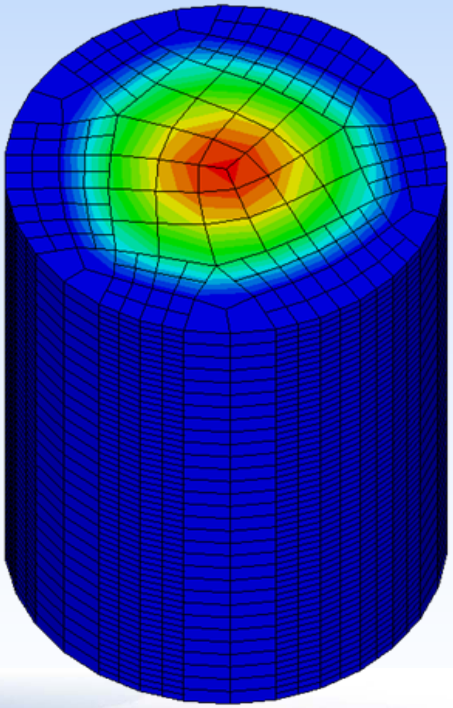
- Multiphysics framework for complex applications development
- Engineering application environment, not "research code"
  - Mesh generation, adaptation, and sensitivity analysis are tightly integrated to form a robust engineering application
- Plug-and-play modules
  - Simplified coupling
- MOOSE Physics Interface conceals framework complexity
- Utilizes state-of-the-art linear and non-linear solvers
  - Robust solvers are key for "ease of use"



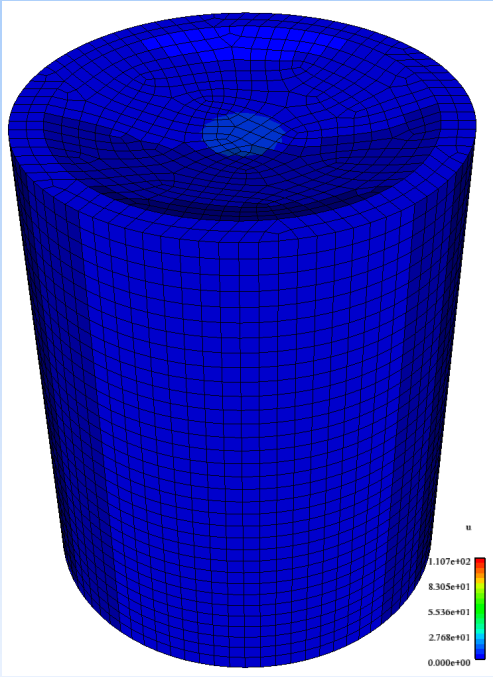
Advanced Capabilities



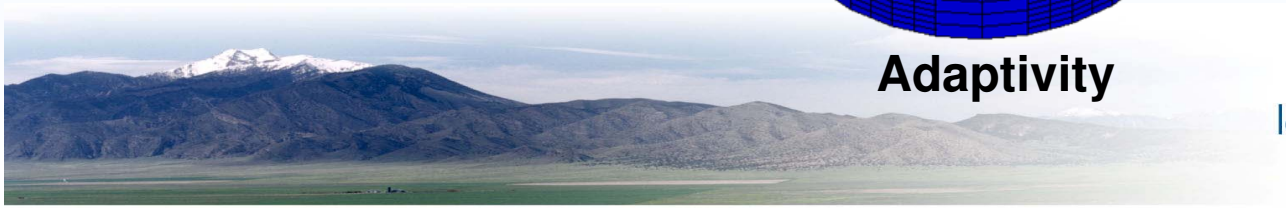
Parallel Decomposition



Adaptivity



Transient Analysis



Future Work

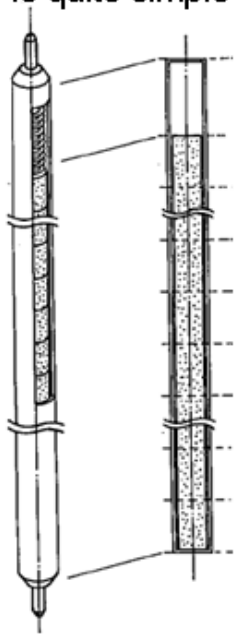
- Advanced physics based preconditioning.
  - De-coupled elliptic solves using multi-grid.
- Adjoint capability.
  - Enhances sensitivity analysis.
  - Enables goal oriented error estimation and adaptivity.
- Finite volume based discrete operator options
- Linking with other libraries.
  - Explore other solver and preconditioning packages.
  - Utilize material databases such as MATPRO.
- Code optimization.

BISON: Motivation and Target Application

- Motivation
  - Develop a predictive capability analysis for reactor fuel performance
  - Efficiently and accurately predict extended burn-up scenarios
- Target Application
  - 3-D, Full pin, extended burn-up cladding integrity
  - Comprehensive pellet–cladding interaction capability
- Capabilities
  - 3-D transient thermomechanics
- Development began in June 2008 !

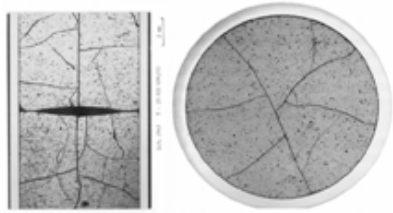
BISON: Nuclear Fuel Issues

At beginning of life, a fuel element is quite simple . . .



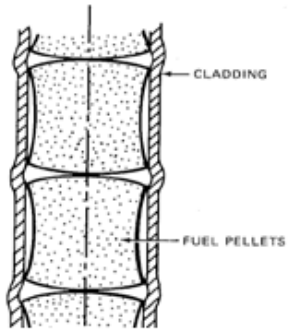
Nakajima et al, Nuc Eng Des, **148**, 41 (1994)

but irradiation brings about substantial complexity



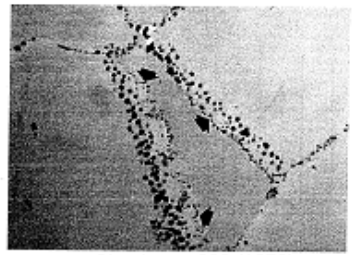
Michel et al, Eng Frac Mech, **75**, 3581 (2008)

Fuel Fracture



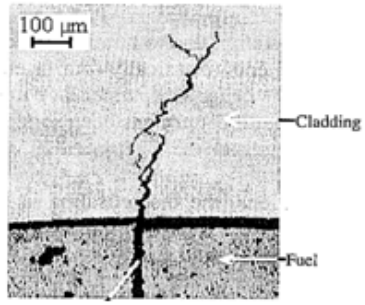
Olander, p. 584 (1978)

Multidimensional contact and deformation



Olander, p. 323 (1978)

Fission gas



Stress Corrosion Cracking Cladding Failure

BISON: Initial Equation Set (1 of 2)

- Nonlinear thermal conduction

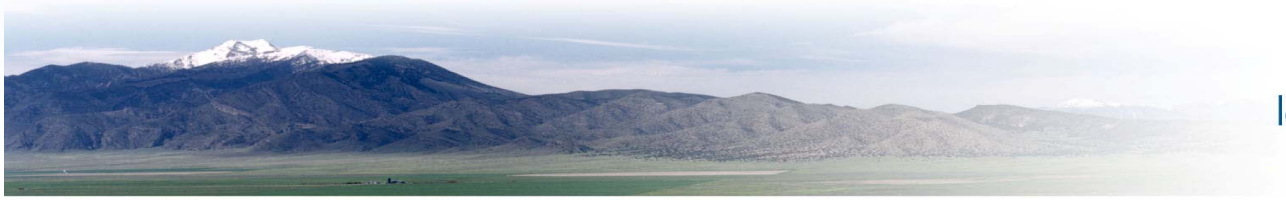
$$\rho C_p T_t + \nabla \cdot k \nabla T - Q = 0$$

  with appropriate boundary conditions

- Nonlinear oxygen non-stoichiometry

$$s_t + \nabla \cdot (D \nabla s + \frac{sQ^*}{FRT^2} \nabla T) = 0$$

  with appropriate boundary conditions



BISON: Initial Equation Set (2 of 2)

- Linear elasticity

$$\mathrm{A}^{\mathrm{T}}\mathrm{D}\mathrm{A}u + f = 0,$$

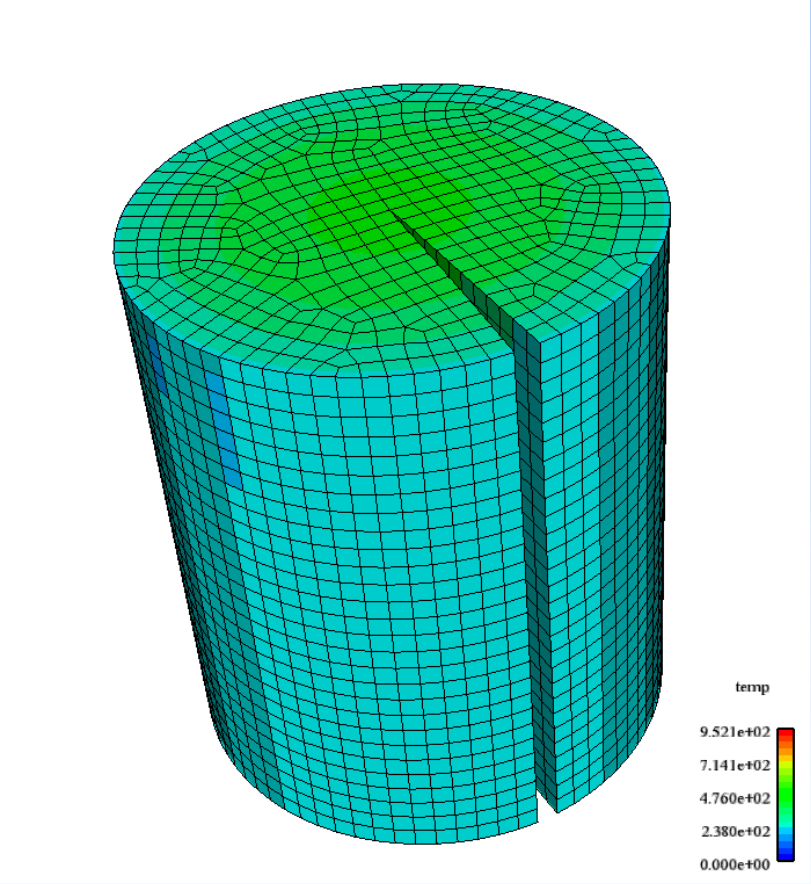
with

$$\mathrm{A} = \begin{bmatrix} \partial_x & 0 & 0 \\ 0 & \partial_y & 0 \\ 0 & 0 & \partial_z \\ \partial_y & \partial_x & 0 \\ 0 & \partial_z & \partial_y \\ \partial_z & 0 & \partial_x \end{bmatrix}, \quad \mathrm{D} = c_1 \begin{bmatrix} 1 & c_2 & c_2 & 0 & 0 & 0 \\ c_2 & 1 & c_2 & 0 & 0 & 0 \\ c_2 & c_2 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_3 \end{bmatrix}$$

and $c_1 = \frac{E(1-\nu)}{(1+\nu)(1-\nu)}, c_2 = \frac{\nu}{(1-\nu)}, c_3 = \frac{(1-2\nu)}{2(1-\nu)}$
with appropriate boundary conditions

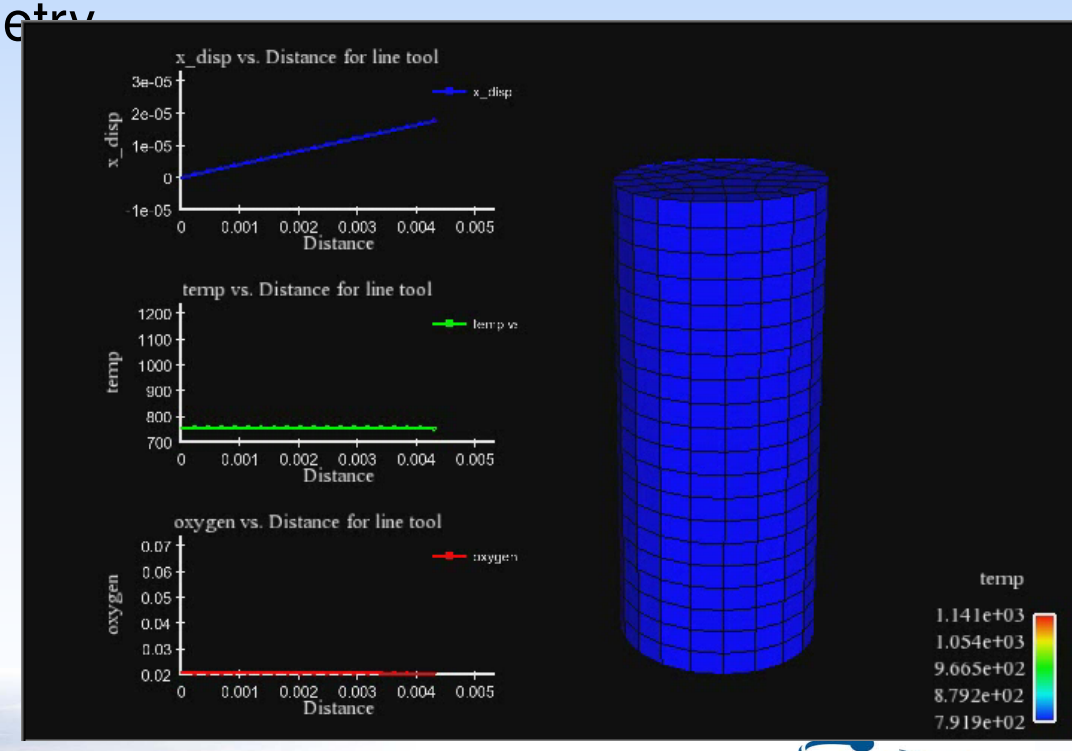
BISON: Example Results

- Coupled transient thermal–solid mechanics



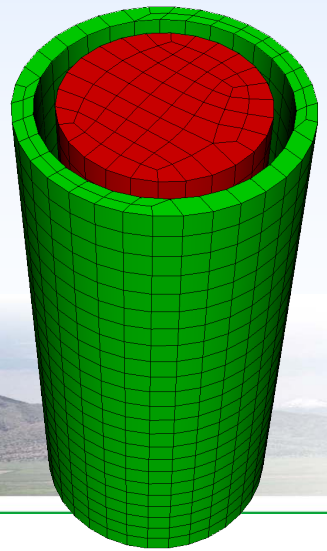
3D Transient Oxygen Diffusion Results

- Transient, fully-coupled thermomechanics and oxygen diffusion
- Inset shows 3D transient displacement X 100, colored by temperature. X-Y plots show radial displacement, temperature, and oxygen hyperstoichiometry



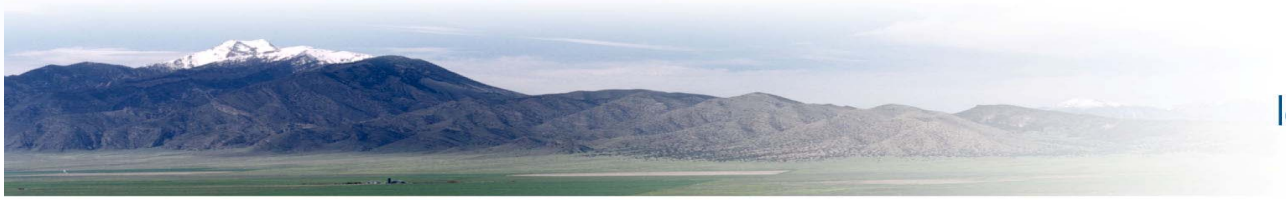
BISON: Near term plan

- Fission gas model, gap heat transfer, nonlinear mechanics, Dec 2008
- FRAPCON / FRAPTRAN test suite, March 2009
- 3-D Pellet-Cladding Mechanical Interaction, June 2009
- Benchmark with ABAQUS pellet-gap-clad model, July 2009
- 3-D multiphysics transient simulation in parallel, initial study of crack impacts on cladding deformation, Dec 2009 (1.5 years into project)



PRONGHORN: Motivation and Target Application

- Motivation
  - Base thermal solver for 3-D steady-state and transient pebble bed gas-cooled reactor. Many researches rely on standard codes (e.g. THERMIX) and use as a black box. (THERMIX is only 2D)
  - Coupled thermal-neutronics solver
- Target Applications
  - Use to study 3-D transient phenomena
  - study multiphysics boundary layer with extended physical model
- Development began in mid-September 2008 !



PRONGHORN: Initial Equation Set (1 of 2)

- Steady-state Thermal-Fluid Model (Darcy-like flow model) (solving for $P, \rho\vec{u}, T_f, T_s, \quad P = \rho R T_f$)
- Momentum (W is porous media friction model)

$$\nabla P - \epsilon\rho\vec{g} + W\rho\vec{u} = 0$$

- Continuity ($\nabla \cdot \rho\vec{u} = 0$)

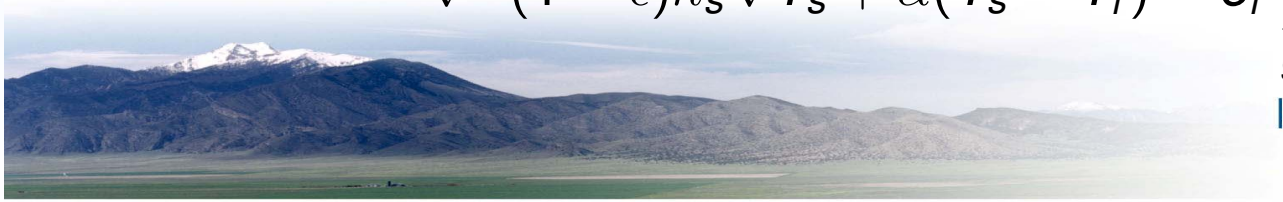
$$-\nabla \cdot \frac{1}{W}\nabla P + \nabla \cdot \frac{\epsilon\rho\vec{g}}{W} = 0$$

- Fluid Energy

$$\nabla \cdot (\rho c_{pf} u T_f) - \nabla \cdot \epsilon\kappa_f \nabla T_f + \alpha(T_f - T_s) = 0$$

- Solid Energy

$$-\nabla \cdot (1-\epsilon)\kappa_s \nabla T_s + \alpha(T_s - T_f) - e_f \sum_{g=1}^{g=G} \Sigma_{fg}\phi_g = 0$$

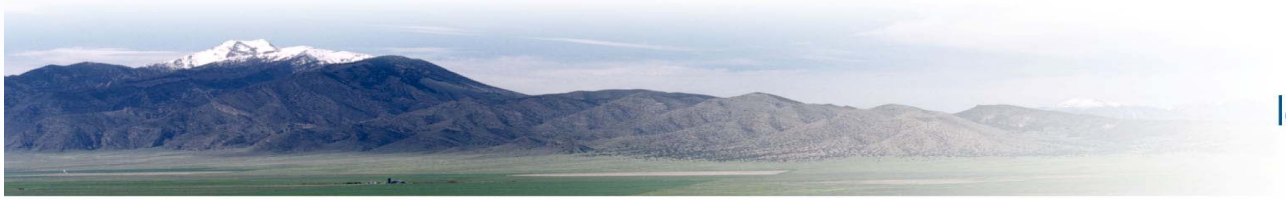


PRONGHORN: Initial Equation Set (2 of 2)

- Neutronics Model (Multigroup diffusion)

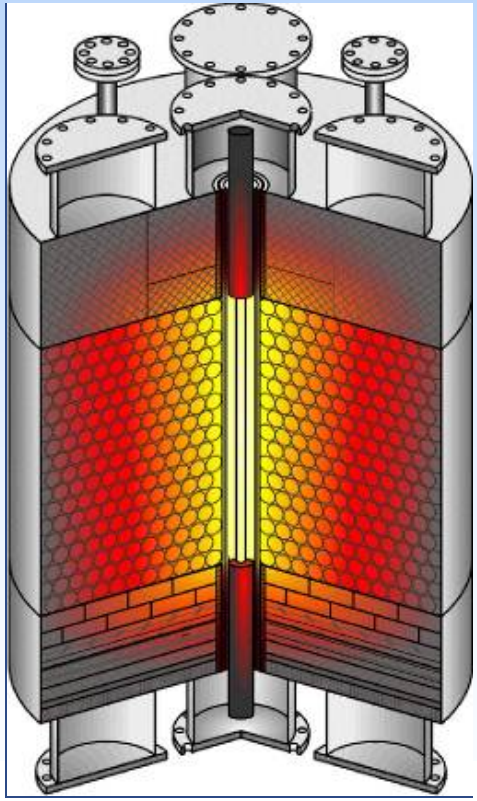
$$-\nabla \cdot D_g \nabla \phi_g + \Sigma_{ag}\phi_g - \chi_g \sum_{g'=1}^{g=G} \nu\Sigma_{fg'}\phi'_g - \sum_{g'=1,g'\neq g}^{g'=G} \Sigma_s^{g'\rightarrow g}\phi_{g'} = 0$$

- D Σ_a $\Sigma_f\Sigma_s$ are generally function of density and temperature (leads to nonlinear equations)



SANA Test

A series of thermal experiments were conducted at SANA test facility.

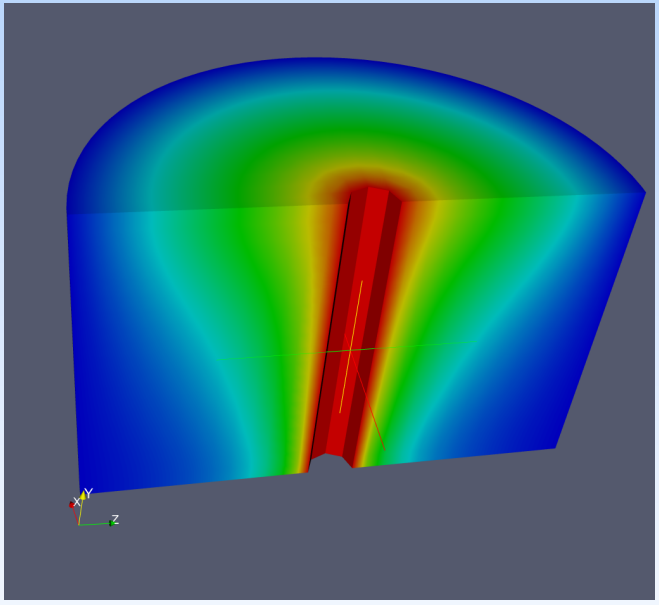


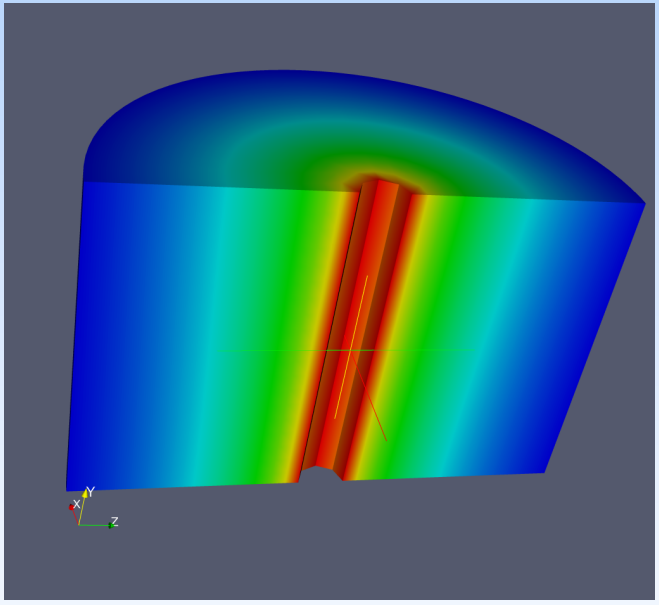
- Installed electrical power 50kW
- Diameter of the pebble bed 1.5m
- Height of the pebble bed 1.0m
- Complete height 3.2m
- Pebble diameter 60mm

Initial result SANA Test

- Fluid Temperature

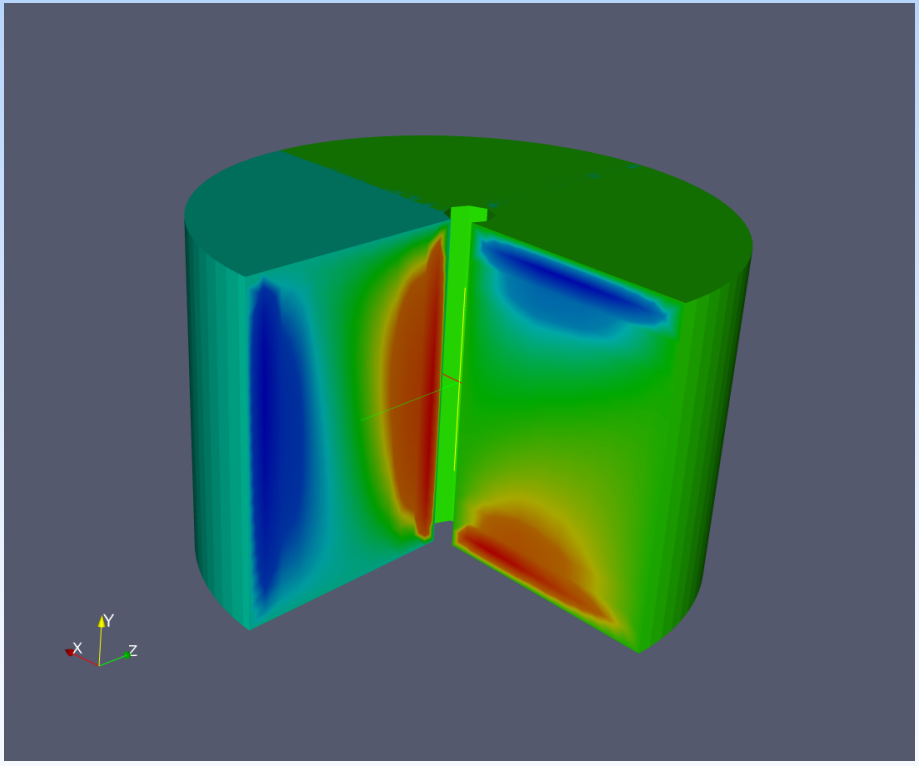
Solid Temperature





Initial result SANA Test, cont.

- Fluid Momentum



PRONGHORN: CPU effort and near term plan

- Rough computational cost
  - total DOF $\approx$ 90000 (6 DOFs/node)
  - CPU time $\approx$ 5min (on workstation with 4 procs)
- Initial validation of steady-state thermal solver with SANA experiment, Jan 2009
- Coupled thermal-neutronics simulation PBMR 400 benchmark, March 2009
- Couple to INL nodal neutronics model, March 2009
- 3-D multiphysics transient simulation in parallel, May 2009 (9 months into project)

Conclusions

- Multiphysics simulation is the next frontier in computational nuclear engineering
- Current algorithms leave open the question of accuracy and stability.
- Jacobian-Free Newton-Krylov (JFNK) methods and physics-based preconditioning provide a modern option.
- Multiphysics Object-Oriented Software Environment, MOOSE, is allowing for rapid, 3-D, parallel, multiphysics application tool development