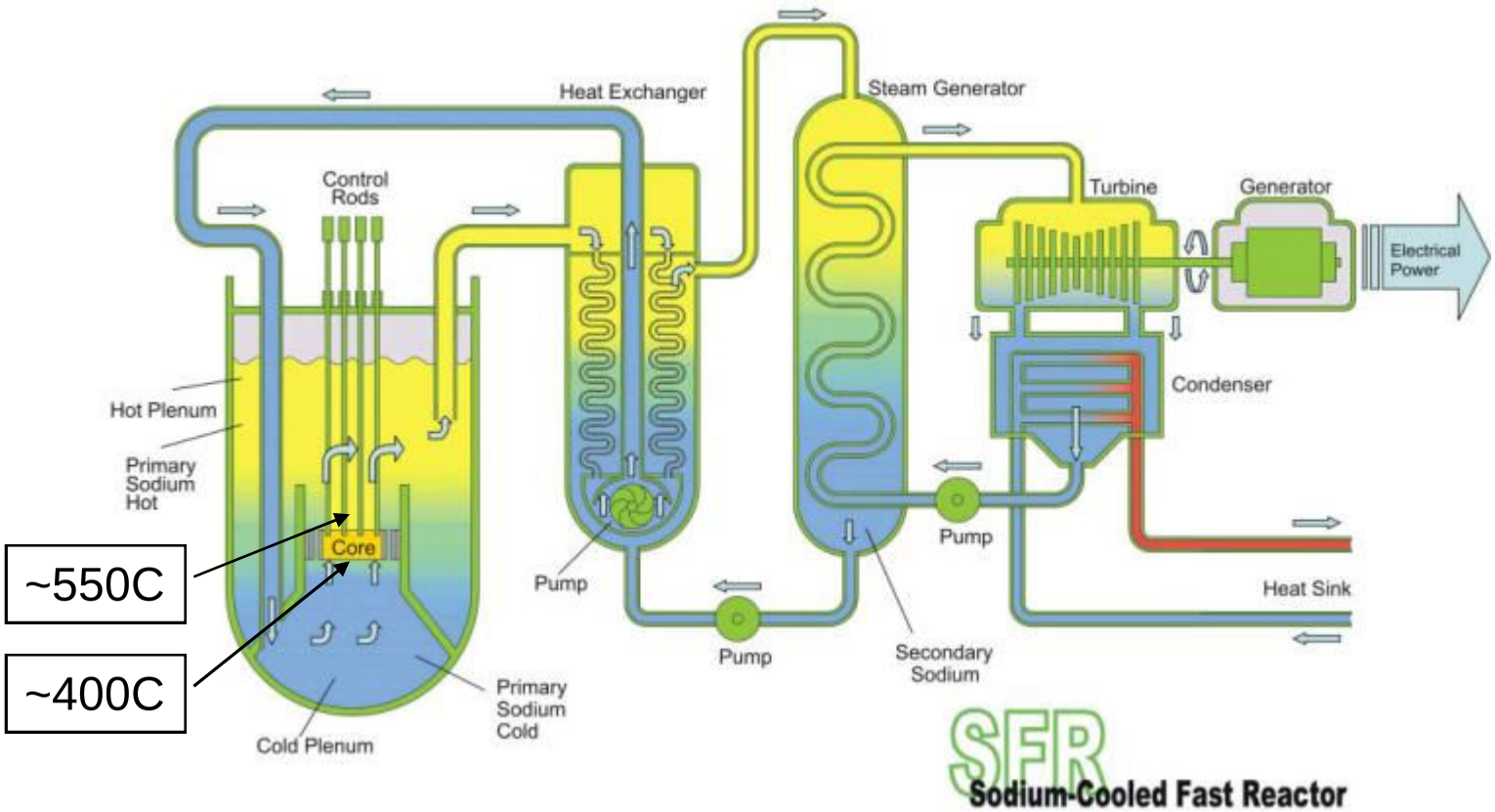
Fast Reactor Simulation

History, current state of the art, and open research questions

Sodium cooled fast reactor (SFR) Loop Design



Key point of fast vs. thermal reactors

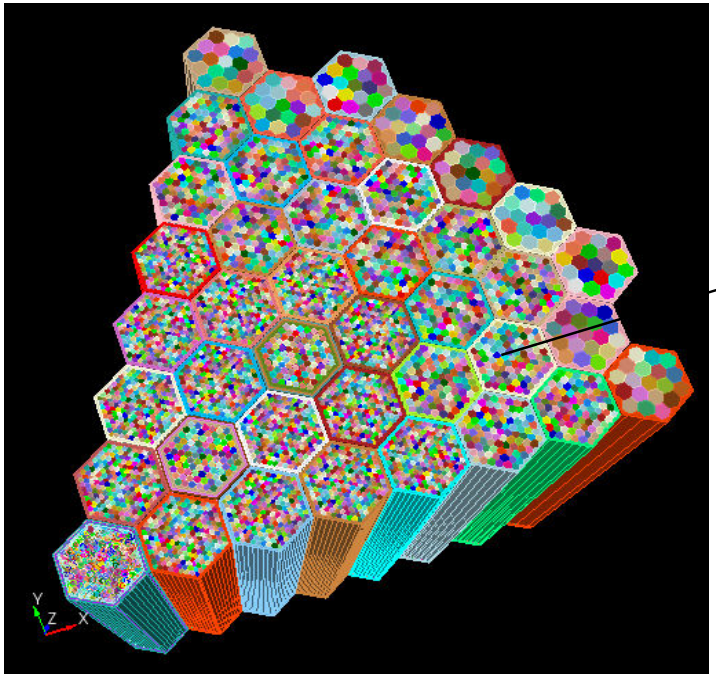
- Thermal reactors (e.g. LWRs)
  - Neutrons moderated to thermal energies (usually using water)
  - Higher probability of fission -> relatively low U-235 enrichment □(~3%)□□□
  - Also high probability of capture by U-238 -> buildup of transuranics
  - All existing commercial reactors are LWRs
  - Major spent fuel burden + inefficient resource usage

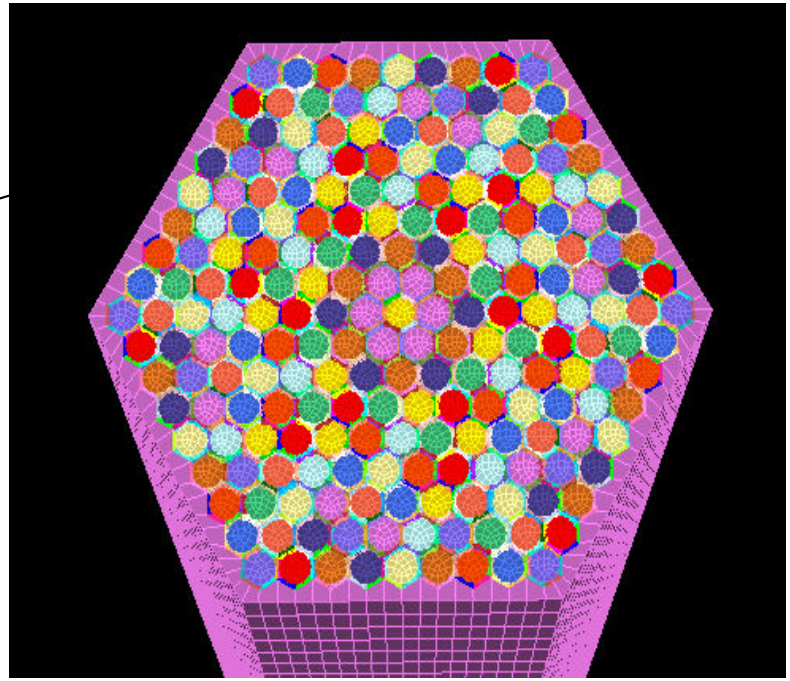
- Fast reactors (e.g. LMFBRs)
  - Neutron moderation minimized
  - Lower-probability of fission -> higher enrichment needed
  - Low probability of capture and ability to fission transuranics/breed plutonium
  - Key to closing fuel cycle + long-term resource management
  - Most built to date have been research or prototype

Fast reactors to date

- A number of fast reactors have been designed/operated over the last 50 years
  - Most have been research or prototype reactors
  - Yet to be successfully commercialized
- Major bottlenecks
  - Capital cost
  - Demonstration of safety
- LWR performance has benefited tremendously from decades of operational experience
- Want to use simulation to greatly accelerate for LMFBRs

Details on core geometry





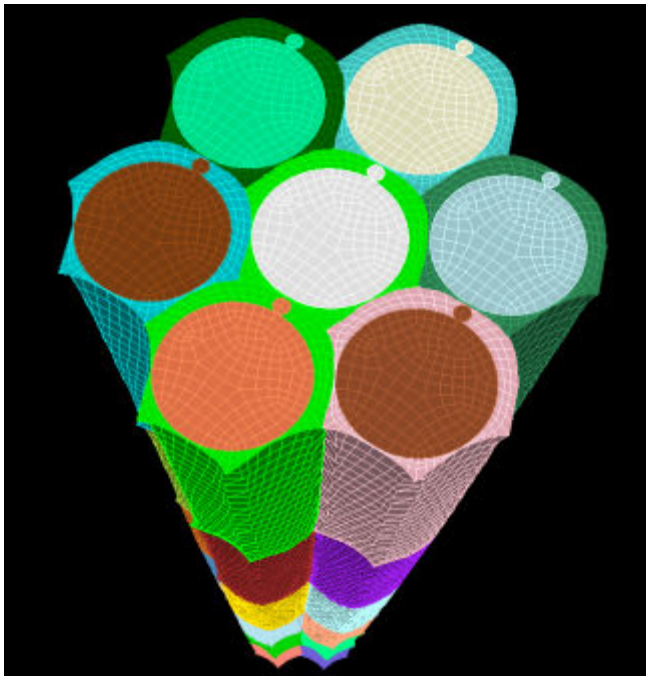
<u>1/6 ABTR core</u>

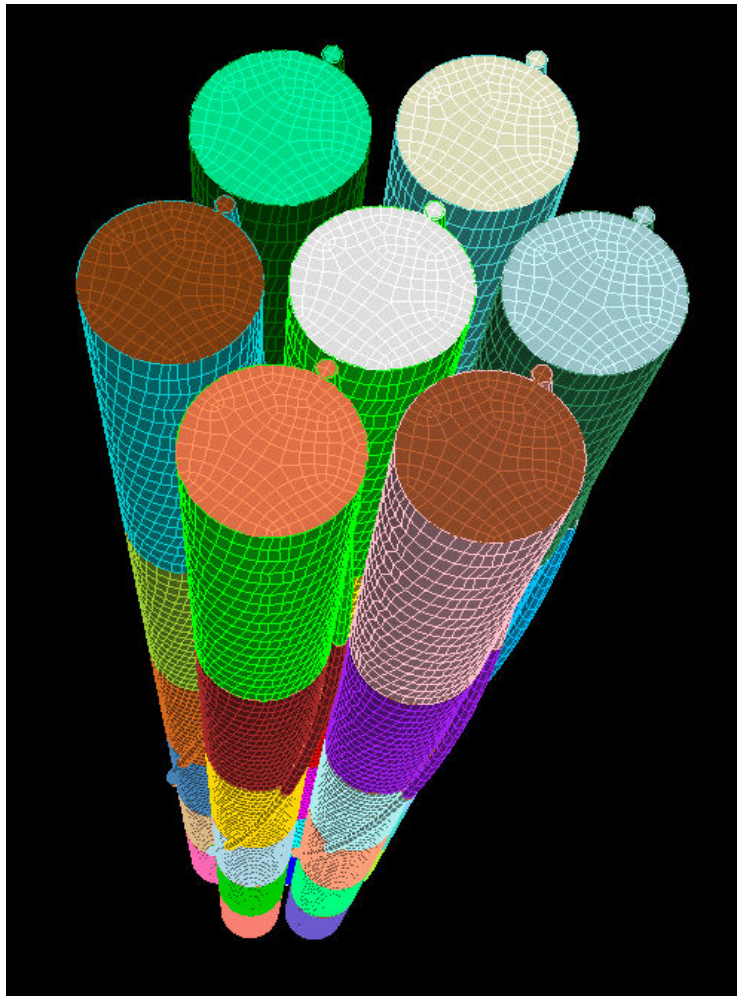
- 7k volumes (core, ctrl, reflect, shield)
- 43k-5m hex elements
- ~6 GB to generate using CUBIT

<u>217-pin fuel ass'y</u>

- Conformal hex mesh
- 1520 vols
- Multiple homogenization options, e.g. pins resolved

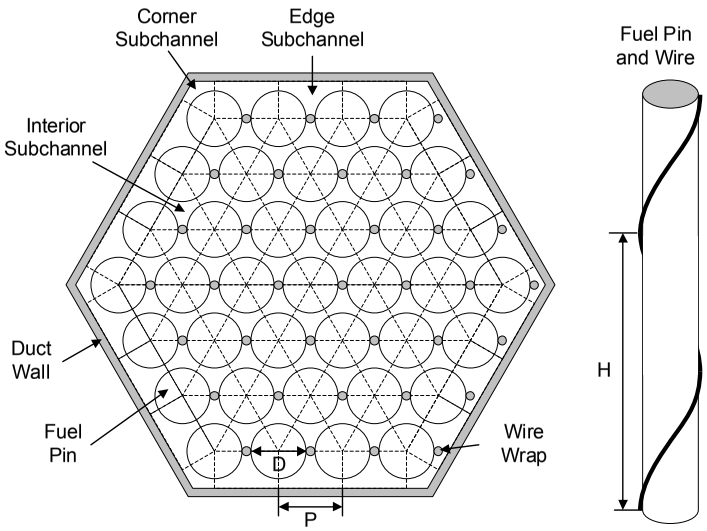
Wire-Wrapped Fuel Pin Assembly (Near-) Exact Geometry

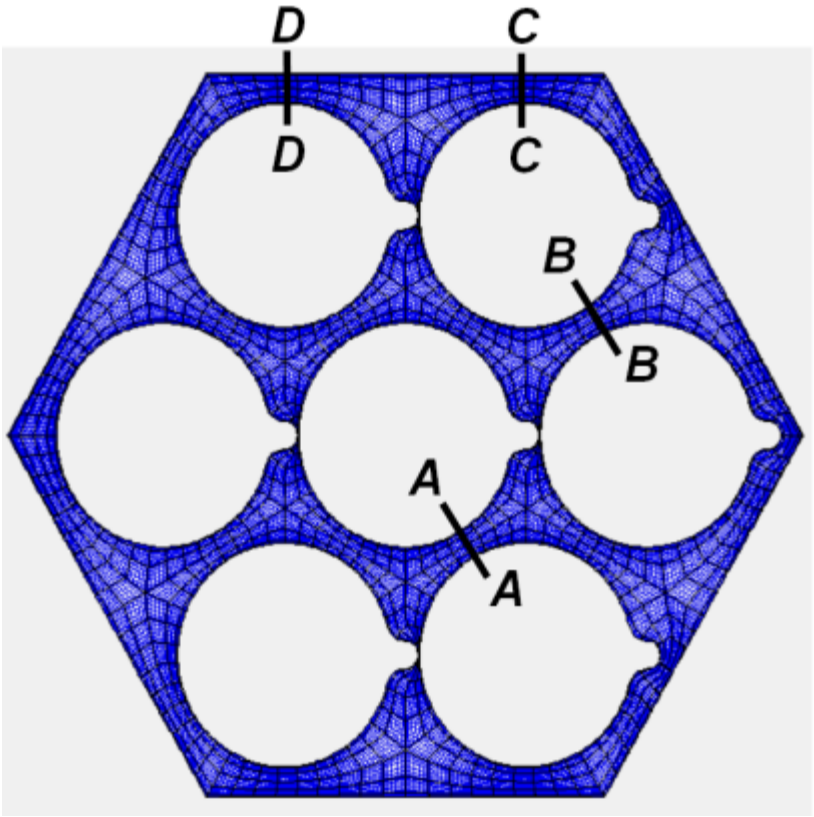


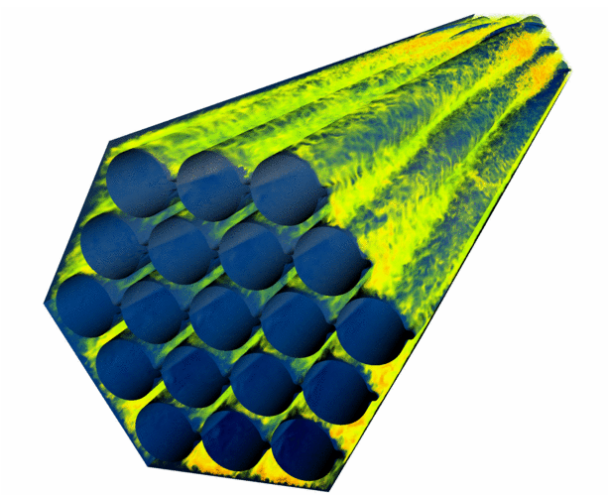


- ***Non-conformal*** hex mesh
- Uses “composite surfaces” for sides of each unit cell
- Space-filling geometry

Close-in of wire-wrapped fuel assembly







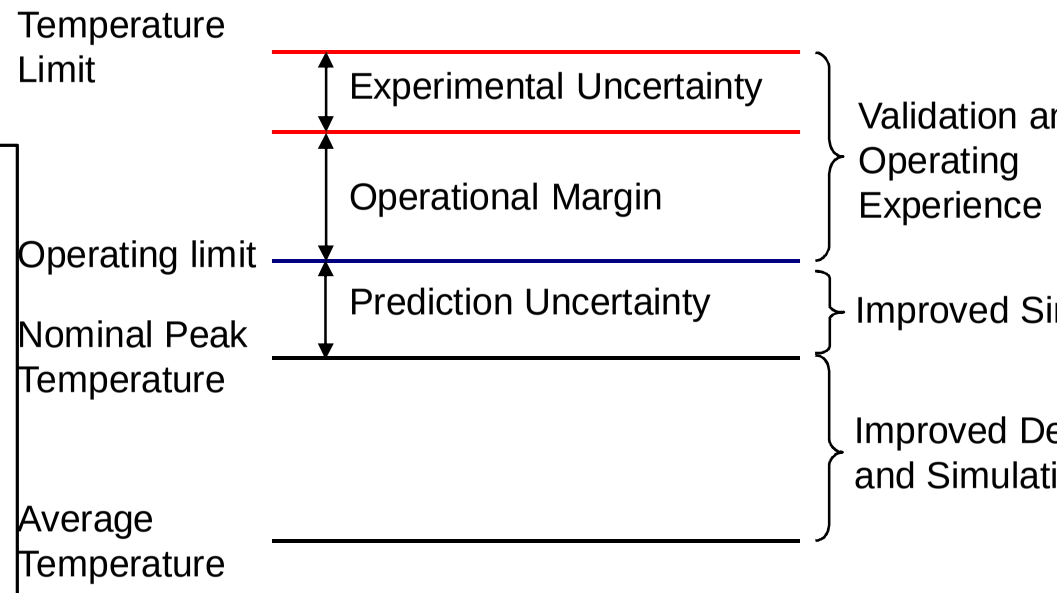
- Wire wrap used to space pins
- Has significant impact on pressure drop, mixing, cross flow

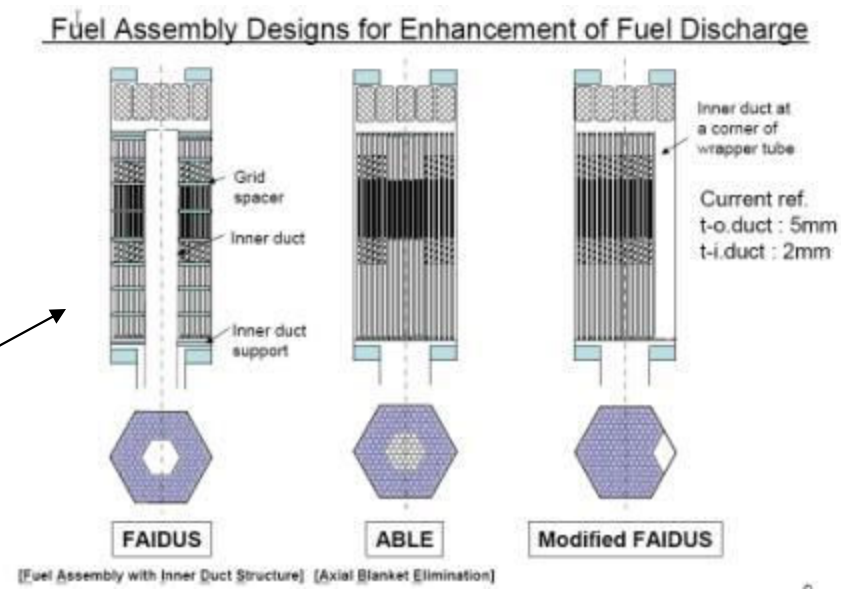
Current state of LMFBR modeling

- Two broad classes of problems -- safety and design
- Huge range of problems to be addressed within these
  - Mixing, shielding, power generation, structural feedback, fuel depletion, cladding failure, transient overpower, transient undercooling, fission product release, sodium boiling, etc etc
- All involve one or several of a handful of phenomena
  - Complex geometries
  - Neutron transport
  - Conjugate heat transfer (low Pr for LMFBR, mostly single phase)
  - Structural deformation
  - Fuel properties/behavior (Unal talk)
  - Lots of data -- cross sections, diffusivities, etc.
- > 1000 person-years of codes developed and deployed in 70s-80s to design early LMFBRs
  - Many codes/models exist since mostly one code/model per phenomenon

Really boiling it down

- Much of these phenomena address two overarching problems
  - Demonstrate increase of linear power to melting
  - Demonstrate unprotected (passive) safety features
- Two approaches
  - Advanced simulation leads to lower rule-of-thumb design margins for existing designs
  - Advanced simulation leads to design innovations with much better economics/safety



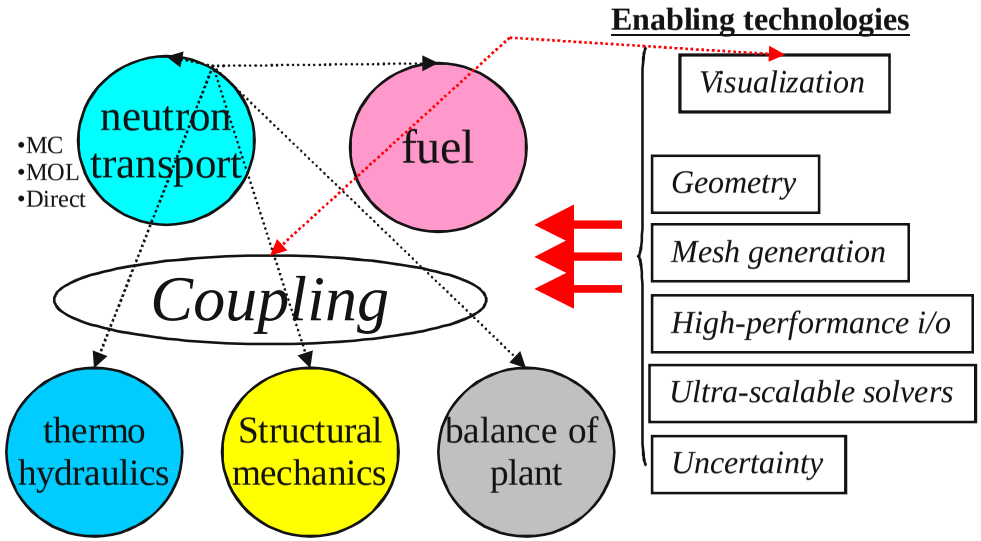


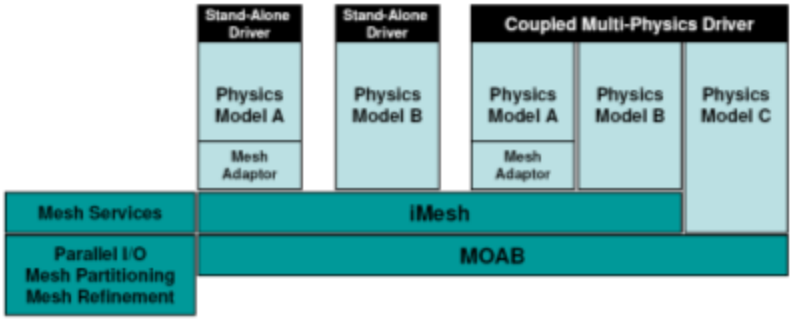
SHARP -- Simulation-based High-efficiency Advanced Reactor Prototyping

Existing models/framework

SHARP overall framework

- In fast reactors, safety depends on SM-neutronics feedback
- In thermal reactors, TH-neutronics feedback significant
- Want to take advantage of tools developed elsewhere
  - Minimize requirements for being used in framework & calculations
    - *Minimal changes to guts of physics code – library/driver*
    - *Each physics can choose its own mesh*
  - Coupling requires knowledge of both meshes & fields on them





Nek5000 & Initial Thermal Hydraulics Simulation Effort

- ANL code for fluids / heat transfer (*Fischer, Lottes, Thomas*)
  - DNS and LES
  - High-order spectral element based code
  - Scales to P > 10,000 processors
  - State of the art multigrid solvers
  - 2 decades of development / verification / validation
  - Supports conjugate heat transfer, variable properties, MHD, ALE, URANS

- Extensive reactor TH experience: (*Fanning, Pointer, Yang*)
  - RANS modeling – Star CD
  - Subchannel codes (SAS)

UNIC neutronics code

- PN2ND
  - Scaled to 64K procs on BG/P (more detail later)
  - Even-parity second-order formulation
  - Use spherical harmonics to represent angular distribution
  - Use finite elements for spatial approximation
  - Superior solution of homogeneous problems
- SN2ND
  - Scaled to 64K prcs on BG/P (more detail later)
  - Even-parity second-order formulation
  - Use discrete ordinates cubature to represent the angular distribution
  - Use finite elements for spatial approximation
  - Provides cheap memory matrix method
- MOCFE
  - Limited scalability in 3d (100s of procs)
  - First order integral transport formulation
  - Use discrete ordinates cubature to represent the angular distribution
  - Use finite elements for spatial approximation
  - Can handle thousands of groups due to matrix free formulation
  - Should be scalable to thousands of processors

Thermo mechanics modeling

- Just beginning simple calculations with LLNL code Diablo
- Little funding so far for this component but will become critical in near future.

Additional tools/technologies used so far in SHARP

- Meshing -- Cubit
- Visualization -- Visit
- Solvers -- Petsc
- Coupling mesh representation -- MOAB
- Language(s): C/FORTRAN mixed with MPI

Project goals

- What we are judged on
  - Carrying out simulations that predict general trends, elucidate fundamental phenomena, or more accurately calculate key physical quantities for more optimized design and licensing of SFRs.
- Very little reward for the tool itself so far
- Some remaining skepticism in community (though waning rapidly) that advanced simulation is correct approach
  - Little time to responsibly lay groundwork for a longer-term effort if not uniform belief in its effectiveness
- Better balance must be struck -- not all off-the-shelf technology nearly good enough. Must push forward math and cs research

General tools/technologies that we are not funded to pursue but would have big impact on project

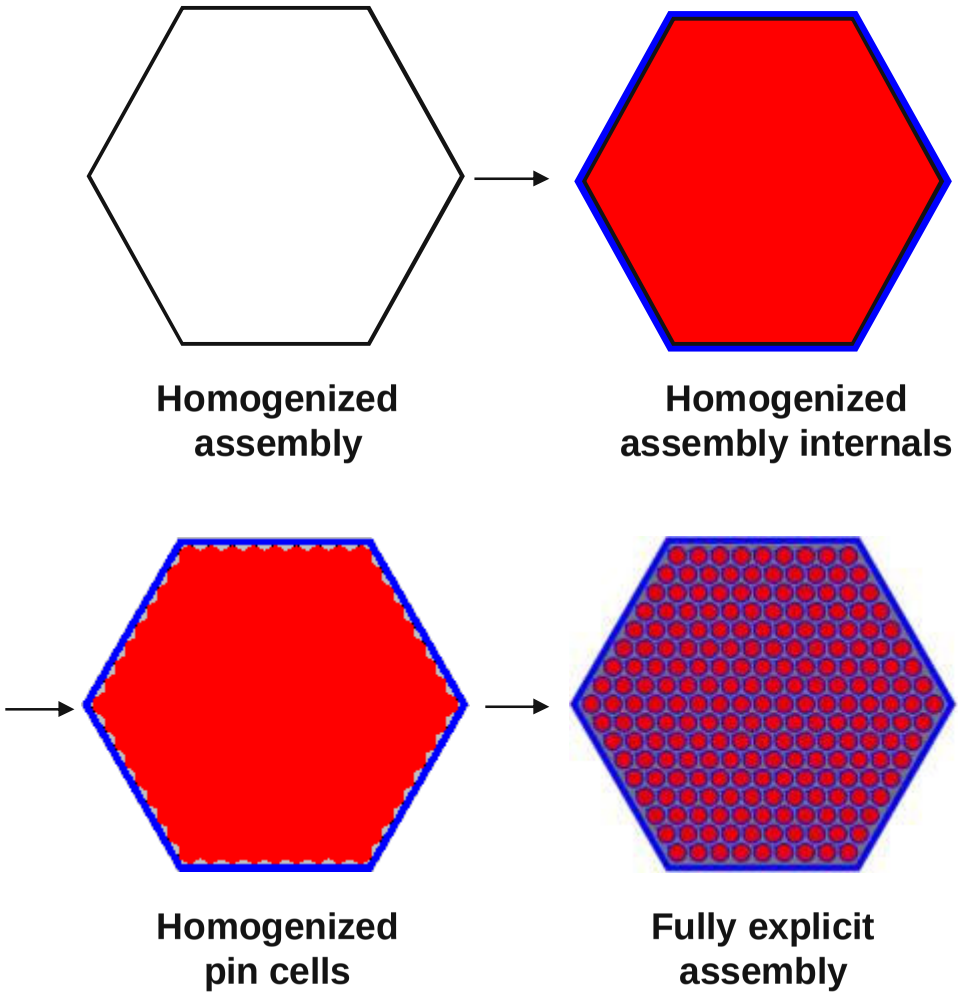
- Languages/compilers with more transparency in use of memory hierarchy
- Quicker tools for algorithmic prototyping/exploration beyond single proc
- Morphing of Matlab/Visit type capabilities for analysis
- *Quantifying uncertainty for multiphysics coupled simulations*
- Reduction of complexity of creating detailed meshes, including cut-cell techniques, better meshing tools/algorithms
- A good open source parallel coupler that does what I need it to
- More advanced parallel debugging tools with advanced memory snooping, etc.
- Faster computers

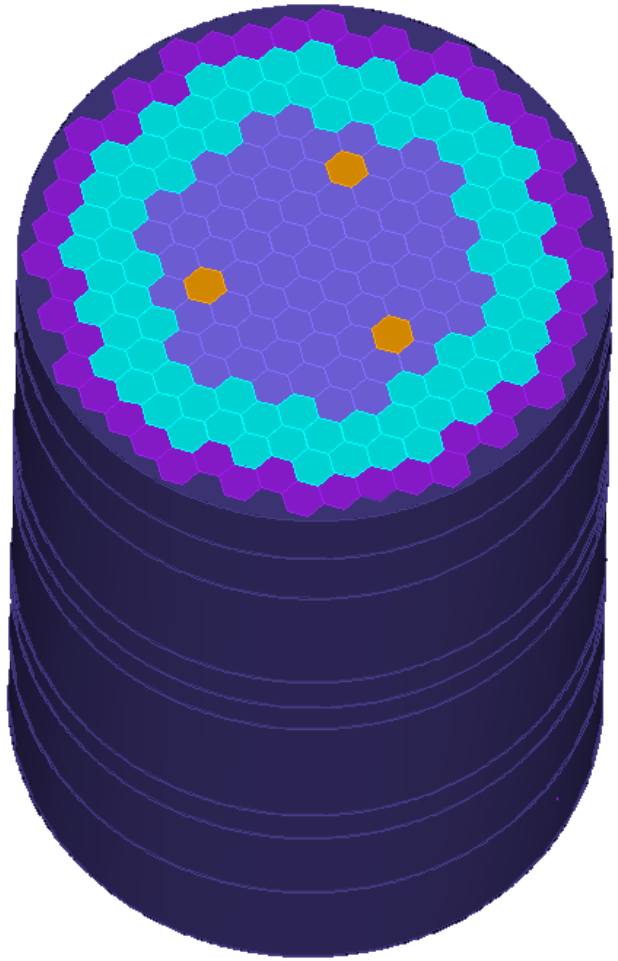
Application-specific problems where great applied mathematicians are needed for breakthroughs

- *Highly scalable transport methods -- improved preconditioners for PN, scalabale ray tracing algorithms for decomposed geometries, hybrid methods, scalable MC etc. towards a single reactor analysis tool for VHTR, SFR, LWR, etc.*
- *Multi-scale approach for heat transfer, transport,* bridging ab initio to engineering scale modeling for fuels, linking DNS, RANS, LES, and subchannel, etc.
- Adaptive transport methods (in energy, angle, space, and polynomial order)
- New ideas beyond multigroup for handling of energy variable
- *Accurate coupling techniques for fast transients in reactor accident scenarios*
- *Structured CFD on complex geometries*
- *Component architectures for tight/loose coupling*
- *Deriving subgrid fluid models for low-PR flows, sodium boiling from DNS*
- *Structural modeling for rod bowing, vessel expansion, etc.*

Neutron transport algorithms in depth

Homogenization at various levels





Neutron Transport: Two Broad Categories

- Stochastic (Monte Carlo)
- Deterministic
  - Even Parity Formulation (P_n-FE)
  - Discrete Ordinates (S_n)
    - *Satisfy the transport equation only along some predetermined angular directions*
  - Integral Form (Method of Characteristics)
    - *Ray tracing*
    - *Works well in low density regions*

Monte Carlo Methods

- Follow individual particle histories from birth (some source) to death (absorption and leakage)
- Nuclear data is expressed in continuum of energy
- Ability to handle complex geometries
- Calculate integral quantities over the whole domain
- Cons
  - Very time consuming for scattering dominant transport (e.g. optically thick medium)
  - Not appropriate when a detailed solution of neutron density throughout the geometry is needed

Deterministic Methods

- Efficient in scattering dominant situations
- Now can handle complex geometries (FEM) and heterogeneous materials
- Give detailed solutions throughout the domain
- Cons
  - Memory intensive (seven dimensional phase space)
  - Energy variable handled less accurately (than Monte Carlo methods)
  - Low density regions are problematic for second order methods

Some Notes on Cross Sections

- Vital source of inaccuracy in deterministic codes
- Very involved process to get cross sections homogenized over an energy group
  - All the physics knowledge goes here
- Large number of energy groups (~10,000) needed for fast reactors to improve accuracy and reduce uncertainties
- Need to repeat the cross section generation with thermal hydraulics feedback (temperature and density changes)

The Steady State Transport Equation (p46 in Lewis)

$$\hat{\Omega}\cdot\vec{\nabla}\psi(\vec{r},\hat{\Omega},E)+\Sigma_t(\vec{r},E)\psi(\vec{r},\hat{\Omega},E)=\int\int\Sigma_s(\vec{r},\hat{\Omega}'\rightarrow\hat{\Omega},E'\rightarrow E)\psi(\vec{r},\hat{\Omega}',E')d\Omega' dE'$$
$$+\frac{1}{k}\chi(\vec{r},E)\int\int\nu(\vec{r},E')\Sigma_f(\vec{r},E')\psi(\vec{r},\hat{\Omega}',E')d\Omega' dE'$$
$$+S(\vec{r},\hat{\Omega},E)$$

$\psi(\vec{r},\hat{\Omega},E)$	The neutron flux (neutron density multiplied by speed)
$\Sigma_t(\vec{r},E)$	The total probability of interaction in the domain
$\Sigma_s(\vec{r},\hat{\Omega}'\rightarrow\hat{\Omega},E'\rightarrow E)d\Omega dE$	The scattering transfer kernel
$\chi(\vec{r},E)$ $\nu(\vec{r},E)$ $\Sigma_f(\vec{r},E)$	The steady state multiplicative fission source
$S(r,\hat{\Omega},E)$	If a fixed source is present then k = 1
k	The multiplication eigenvalue

Multigroup Formalism

- To date the following has been ineffective
  - Polynomial expansions of the neutron flux dependence
  - Finite element expansions of the neutron flux dependence
- In general this is measured by the gain in accuracy relative to the cost associated with the solution algorithm
- Therefore, we chose to continue using the multigroup formalism

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega})+\Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\hat{\Omega})=\sum_{g'=1}^{G}\int\Sigma_{s,g'\to g}(\vec{r},\hat{\Omega}'\to\hat{\Omega})\psi_{g'}(\vec{r},\hat{\Omega}')d\Omega'$$
$$+\frac{1}{k}\chi_g(\vec{r})\sum_{g'=1}^{G}\int\nu_{g'}(\vec{r})\Sigma_{f,g'}(\vec{r})\psi_{g'}(\vec{r},\hat{\Omega}')d\Omega'$$
$$+S_g(\vec{r},\hat{\Omega})$$

Solving the Eigenvalue Problem

$A\,x = \lambda\,x$ Cast the transport equation as a pseudo matrix-vector operation

T = streaming/collision/scattering F = fission

$$\psi = \left[\psi_1(\vec{r},\hat{\Omega}) \quad \psi_2(\vec{r},\hat{\Omega}) \quad \cdots \quad \psi_G(\vec{r},\hat{\Omega})\right]^T$$

$$T\,\psi = \hat{\Omega}\cdot\vec{\nabla}\psi_g(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g(\vec{r},\hat{\Omega}) - \sum_{g'=1}^{G}\int \Sigma_{s,g'\to g}(\vec{r},\hat{\Omega}'\to\hat{\Omega})\psi_{g'}(\vec{r},\hat{\Omega}')d\Omega'$$

$$F\,\psi = \chi_g(\vec{r})\sum_{g'=1}^{G}\int \nu_{g'}(\vec{r})\Sigma_{f,g'}(\vec{r})\psi_{g'}(\vec{r},\hat{\Omega}')d\Omega'$$

$$T\,\psi = \frac{1}{k}F\,\psi$$

Standard eigenvalue notation:

$$A\,x = \lambda\,x$$

$$A = T^{-1}F$$

$$x = \psi$$

$$\lambda = k$$

Eigenvalue Solution Process (Inverse Power Method)

$$T\psi = \frac{1}{k} F\psi$$

$$\theta = F\psi \quad \text{Fission source}$$

$$T\psi = \frac{1}{k} F\psi$$

$$T\psi = \frac{1}{k}\theta$$

$$k\theta = F\,T^{-1}\theta$$

Last equation has the iterative solution:

$$\theta^{(i)} = \frac{1}{k^{(i-1)}} F\,T^{-1}\theta^{(i-1)} = F\left[T^{-1}\left(\frac{1}{k^{(i-1)}}\theta^{(i-1)}\right)\right]$$

$$k^{(i)} = k^{(i-1)}\frac{\left\langle w, \theta^{(i)}\right\rangle}{\left\langle w, \theta^{(i-1)}\right\rangle}$$

The w is an arbitrary vector which we choose to be **1** such that we get the total fission source.

Neutronics Code Solution Process

- Inversion of T is generally very expensive due to scattering coupling.
- Recast using the within group notation and sweep the groups
  - Logical approach given that iteration on fission is already necessary
  - “Downscatter” only region has straightforward solution (down sweep)
  - “Upscatter” region requires additional iteration (more down sweeps)

$$\hat{\Omega}\cdot\vec{\nabla}\psi_g^{(i)}(\vec{r},\hat{\Omega})+\Sigma_{t,g}(\vec{r})\psi_g^{(i)}(\vec{r},\hat{\Omega})=\int\Sigma_{s,g\to g}(\vec{r},\hat{\Omega}'\to\hat{\Omega})\psi_g^{(i)}(\vec{r},\hat{\Omega}')d\Omega'+Q_g^{(i)}(\vec{r},\hat{\Omega})$$

$$\begin{aligned}Q_g^{(i)}(\vec{r},\hat{\Omega})=&\sum_{\substack{g'\in 1,G\\ g'\neq g}}\int\Sigma_{s,g'\to g}(\vec{r},\hat{\Omega}'\to\hat{\Omega})\psi_{g'}^{(i)}(\vec{r},\hat{\Omega}')d\Omega'\\&+\frac{1}{k^{(i-1)}}\chi_g(\vec{r})\sum_{g'=1}^{G}\int\nu_{g'}(\vec{r})\Sigma_{f,g'}(\vec{r})\psi_{g'}^{(i-1)}(\vec{r},\hat{\Omega}')d\Omega'\\&+S_g(\vec{r},\hat{\Omega})\end{aligned}$$

Picking a Method

- Monte Carlo
  - Uses particle “walking” with random number decisions to decide path
  - Has been implemented in multi-group and continuous energy
- First order discrete ordinates (matrix free)
  - Domain is swept in specific directions, element by element
  - Element sized A matrix is formed and inverted during sweep
- First order method of characteristics (matrix free cp)
  - Series of parallel trajectories are used to pierce the domain
  - Solution along the ray is cheap and source by element is simple sum
- Second order discrete ordinates (generally a matrix method)
  - Block diagonal matrix makes method reasonable to store
- Second order spherical harmonics (generally a matrix method)
  - Includes all diffusion theory methods
  - Storage of matrices is quite expensive
- Second order angular finite elements
  - Rare usage due to complication of anisotropic scattering

Methods Employed in UNIC

- PN2ND
  - Even-parity second-order formulation
  - Use spherical harmonics to represent angular distribution
  - Use finite elements for spatial approximation
  - Superior solution of homogeneous problems
- SN2ND
  - Even-parity second-order formulation
  - Use discrete ordinates cubature to represent the angular distribution
  - Use finite elements for spatial approximation
  - Provides cheap memory matrix method
- MOCFE
  - First order integral transport formulation
  - Use discrete ordinates cubature to represent the angular distribution
  - Use finite elements for spatial approximation
  - Can handle thousands of groups due to matrix free formulation
  - Should be scalable to thousands of processors

Second Order Methods; Within Group Formulation

Angular flux is written in terms of even- and odd-parity components

$$\psi_g(\vec{r},\hat{\Omega}) = \psi_g^+(\vec{r},\hat{\Omega}) + \psi_g^-(\vec{r},\hat{\Omega})$$

Even- and odd-parity forms of the transport equation are obtained

Even-Parity $$\hat{\Omega}\cdot\vec{\nabla}\psi_g^-(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g^+(\vec{r},\hat{\Omega}) = W_g^+(\vec{r},\hat{\Omega}) + Q_g^+(\vec{r},\hat{\Omega})$$

Odd-Parity $$\hat{\Omega}\cdot\vec{\nabla}\psi_g^+(\vec{r},\hat{\Omega}) + \Sigma_{t,g}(\vec{r})\psi_g^-(\vec{r},\hat{\Omega}) = W_g^-(\vec{r},\hat{\Omega}) + Q_g^-(\vec{r},\hat{\Omega})$$

Within group Scattering

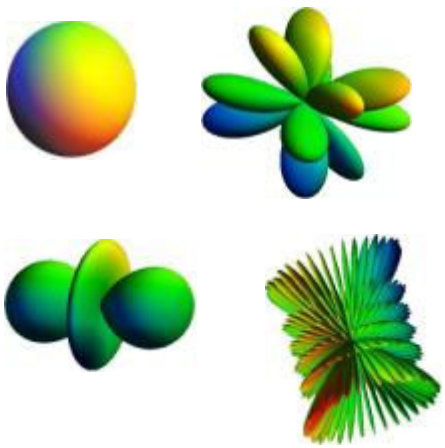
$$W_g^{\pm}(\vec{r},\hat{\Omega}) = \int \Sigma_{s,g\to g}^{\pm}(\vec{r},\hat{\Omega}\cdot\hat{\Omega}')\psi_g^{\pm}(\vec{r},\hat{\Omega}')d\hat{\Omega}'$$

$$= \int \sum_m \Sigma_{s,g\to g,m}^{\pm}(\vec{r})P_m^{\pm}(\hat{\Omega}\cdot\hat{\Omega}')\psi_g^{\pm}(\vec{r},\hat{\Omega}')d\hat{\Omega}'$$

External Source $$Q_g^{\pm}(\vec{r},\hat{\Omega}) = \tfrac{1}{2}Q_g(\vec{r},\hat{\Omega}) \pm \tfrac{1}{2}Q_g(\vec{r},-\hat{\Omega})$$

PN2ND

Spherical harmonics approximation

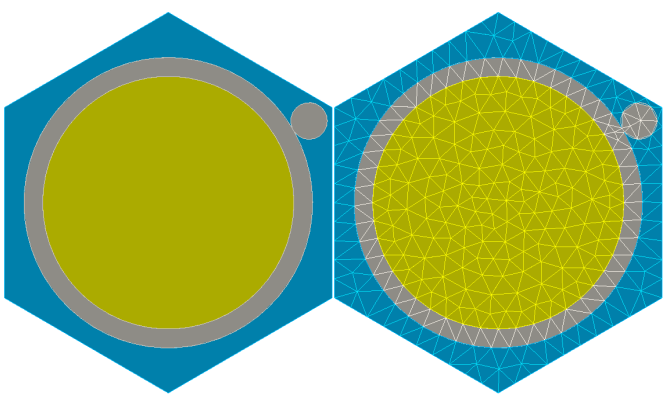


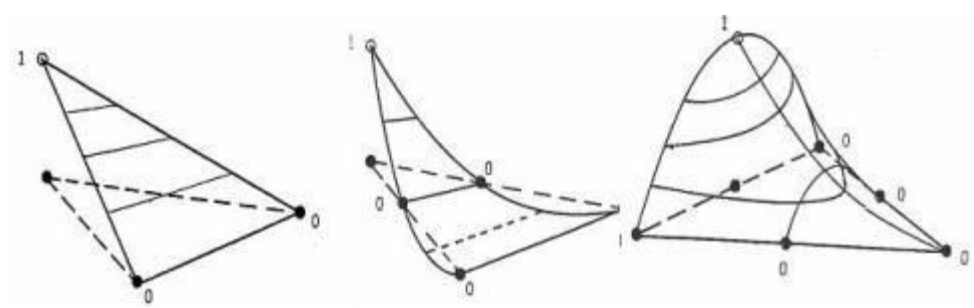
$$\psi_g^+(\vec{r},\Omega) = \sum_{l=0,2}^{L} \sum_{m=-l}^{l} Y_{l,m}(\hat{\Omega})\psi_{g,l,m}^+(\vec{r}) = \mathbf{Y}_+^T(\hat{\Omega})\psi_g^+(\vec{r})$$

$$\psi_g^-(\vec{r},\Omega) = \sum_{l=1,3}^{L} \sum_{m=-l}^{l} Y_{l,m}(\hat{\Omega})\psi_{g,l,m}^-(\vec{r}) = \mathbf{Y}_-^T(\hat{\Omega})\psi_g^-(\vec{r})$$

Finite element approximation

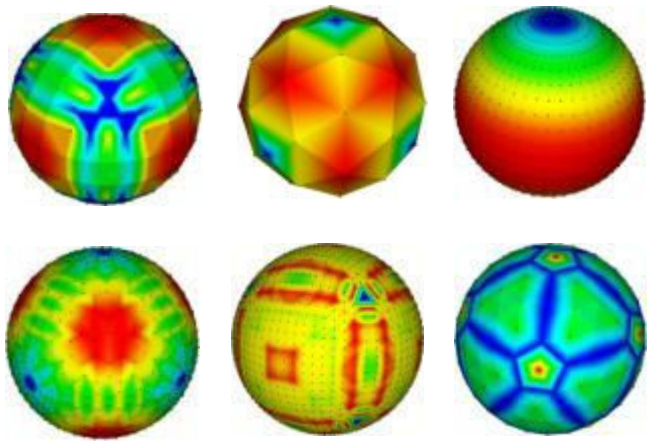
$$\psi_g^\pm(\vec{r},\Omega) = \mathbf{Y}_\pm^T(\hat{\Omega}) \otimes L^T(\vec{r})\,\psi_g^\pm$$





SN2ND

Discrete ordinates approximation



$$\psi_g^+(\vec{r},\Omega) \Rightarrow \varphi_g^+(\vec{r},\Omega_n) = L^T(\vec{r})\varphi_{g,n}^+$$

$$\sum_n Y_{l,m}(\hat{\Omega}_n) w_n = \delta_{l,0}\delta_{m,0}$$

$$\psi_g^+(\vec{r},\hat{\Omega}) = \boldsymbol{Y}_+^T(\hat{\Omega}) \otimes L^T(\vec{r})\,\psi_g^+$$

$$\psi_g^+ = \sum_n \boldsymbol{Y}_+^T(\hat{\Omega}_n)\varphi_{g,n}^+ w_n = M_+\varphi_g^+$$

$$\{\ \}_{+,n}^{SN} = \boldsymbol{Y}_+(\hat{\Omega}_n)\{\ \}_+^{PN} = N_{+,n}\{\ \}_+^{PN}$$

$$\psi_g^-(\vec{r},\Omega) = \boldsymbol{Y}_-^T(\hat{\Omega}) \otimes L^T(\vec{r})\,\psi_g^-$$

PN2ND & SN2ND Governing Equations

PN2ND $$A_g^{PN}\psi_g^+ = Q_g^+ + Q_g^-$$ $$\Xi_{g,m} = \left(\Sigma_t - \Sigma_{s,g\to g,m}^-\right)^{-1} \quad RULE: \begin{array}{c} g'\in 1,G \\ g'\neq g \end{array}$$

SN2ND $$A_g^{SN}\varphi_g^+ = N_+\left(Q_g^+ + Q_g^-\right)$$ $$\psi_g^+ = M_+\varphi_g^+$$ $$\Xi_{g,m} = \left(\Sigma_t\right)^{-1} \quad RULE: \quad g'\in 1,G$$

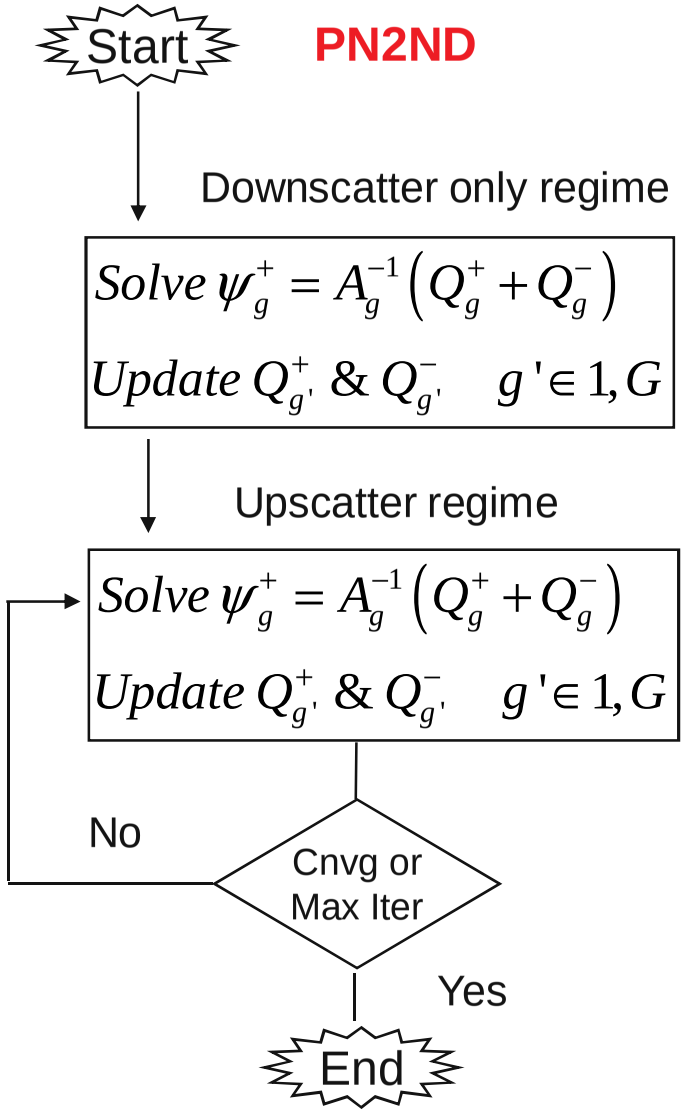
Group source equations

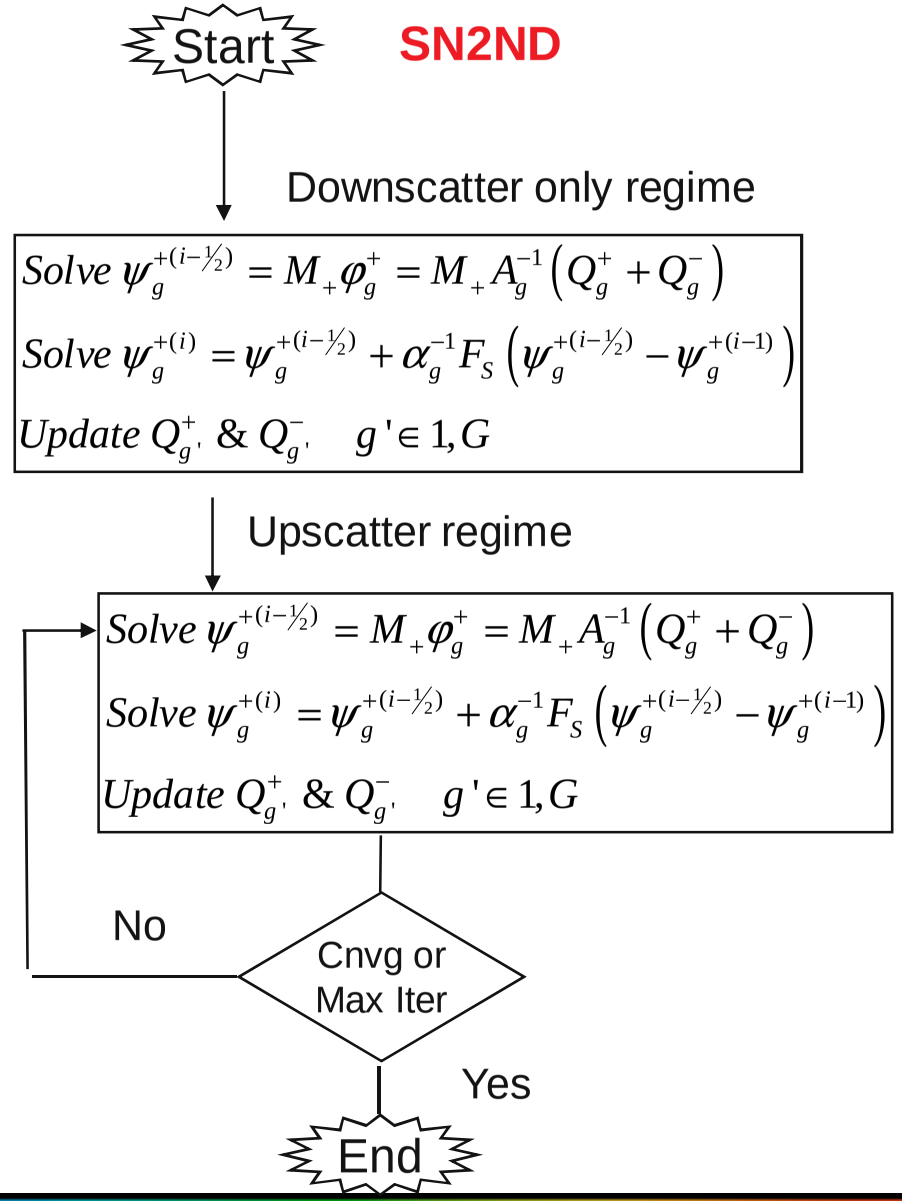
$$Q_{g,e}^{+\,(i)} = \sum_{RULE} \bar{\Sigma}_{s,g'\to g}^+ \otimes F_e\, \psi_{g',e}^{+\,(i)} + \frac{1}{k^{(i-1)}}\chi_{g,e}\sum_{g'=1}^{G}\nu\Sigma_{f,g',e}\left(J^+\otimes F_e\right)\psi_{g',e}^{+\,(i-1)} + \left(I^+\otimes F_e\right)S_{g,e}^+$$

$$\xi_{g,e,m}^{-\,(i)} = \Xi_{g,m}\sum_{K,L}\left(V_K V_L^T\right)_m \otimes P_{e,K,L}^T\, \psi_{g,e}^{+\,(i)} + \Xi_{g,m} Q_{g,e,m}^{-\,(i)}$$

$$Q_{g,e}^{-\,(i)} = \Xi_{g,m}\sum_{RULE}\Sigma_{s,g'\to g,m}^-\, \xi_{g',e,m}^{-\,(i)} + \Xi_{g,m}\left(V_K\right)_m \otimes U_{e,K,L}^T S_{g,e}^{-\,(i)}$$

Group Sweeping Algorithm



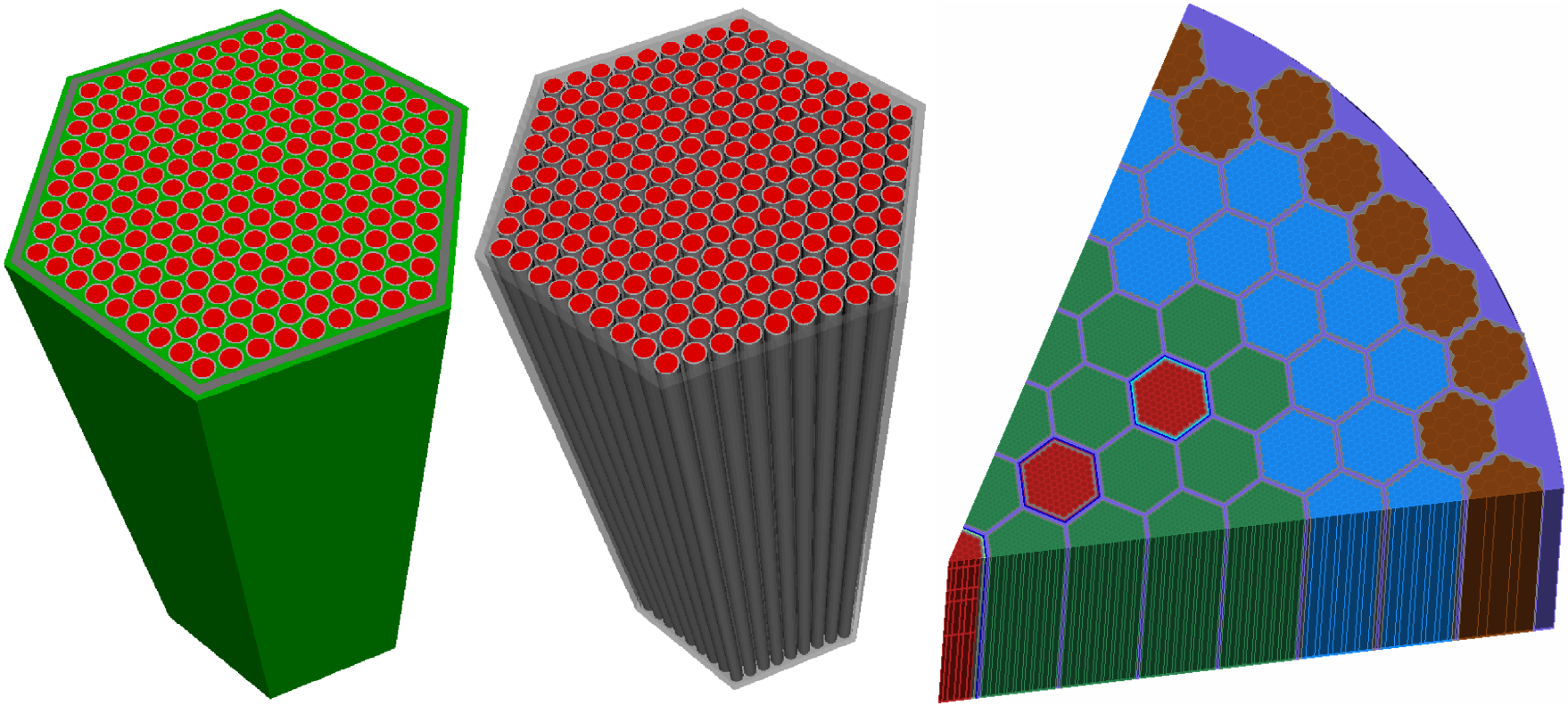


Features of Second Order Form Solutions in UNIC

- PN2ND and SN2ND solvers have been developed to solve the steady-state, second-order, even-parity neutron transport equation
  - PN2ND: Spherical harmonic method in 1D, 2D and 3D geometries with FE mixed mesh capabilities
  - SN2ND: Discrete ordinates in 2D and 3D geometries with FE mixed mesh capabilities
- These second order methods have been implemented on large scale parallel machines
  - Linear tetrahedral and quadratic hexahedral elements
  - Fixed source and eigenvalue problems
  - Arbitrarily oriented reflective and vacuum boundary conditions
  - PETSc solvers are utilized to solve within-group equations
    - *Conjugate gradient method with SSOR and ICC preconditioners*
    - *Other solution methods and preconditioners will be investigated*
  - Synthetic diffusion acceleration for within-group scattering iteration
  - Power iteration method for eigenvalue problem
    - *Various acceleration schemes are being investigated*
  - MeTiS is employed for mesh partitioning

General Geometry Capability in UNIC – ABTR Assembly

- Unstructured finite element meshes are employed
  - CUBIT package is the primary mesh generation tool
  - hexahedral and tetrahedral elements (linear and quadratic)
  - Further research is required for reducing mesh generation efforts and robust merging of the meshes of individual geometrical components

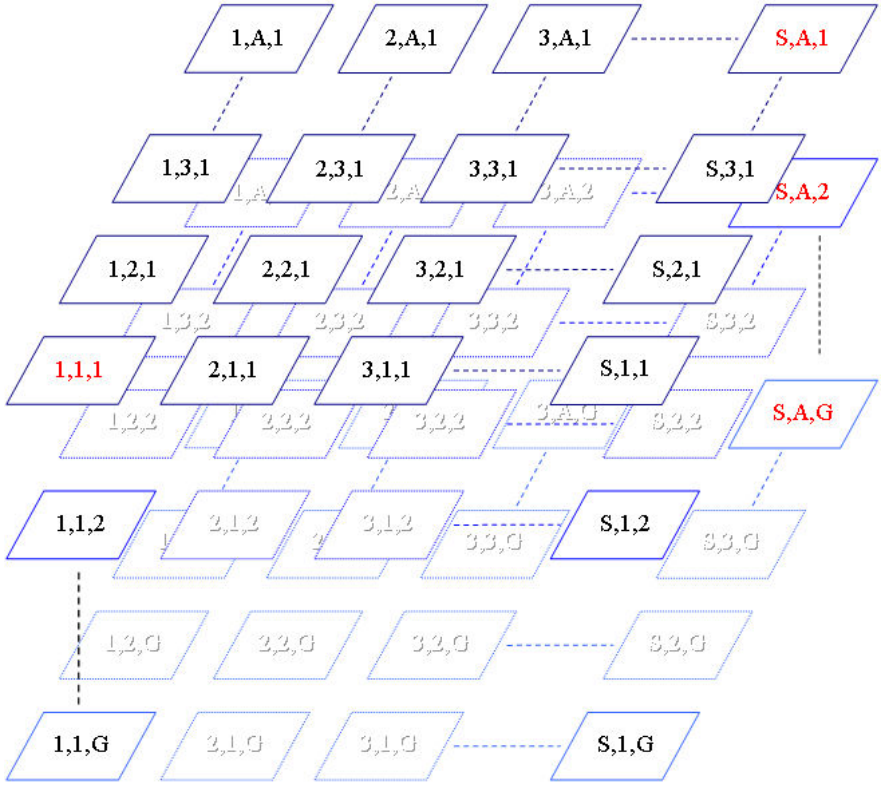


Key Features of the Implementation Strategy

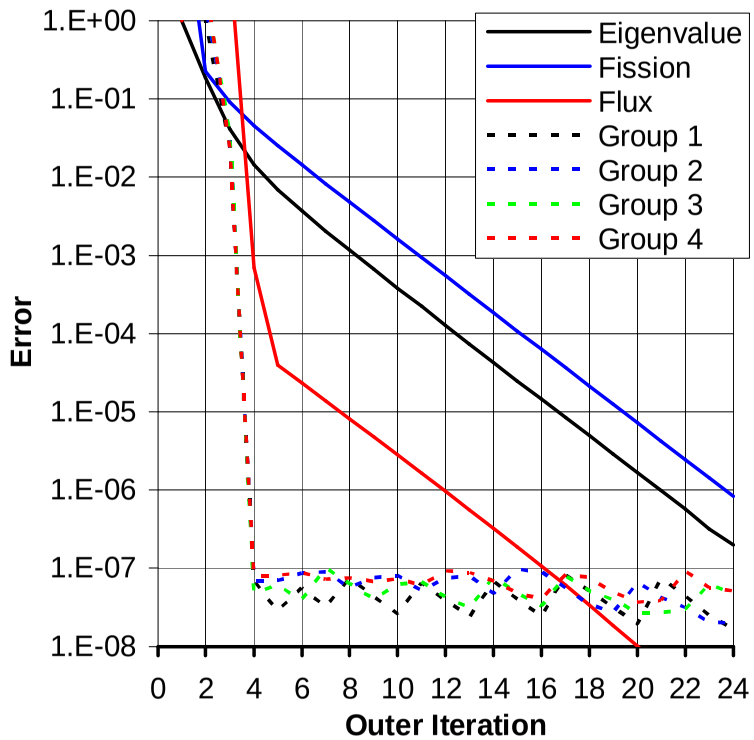
- Follow the “owner computes” rule under the dual constraints of minimizing the number of messages and overlapping communication with computation
- Each processor “ghosts” its stencil dependences in its neighbors
- Ghost nodes ordered after contiguous owned nodes
- Domain mapped from (user) global ordering into local orderings
- Scatter/gather operations created between **local sequential** vectors and **global distributed** vectors, based on runtime connectivity patterns
- Krylov-Schwarz operations translated into local tasks and communication tasks
- Profiling used to help eliminate performance bottlenecks in communication and memory hierarchy

Improved Parallelism and Acceleration

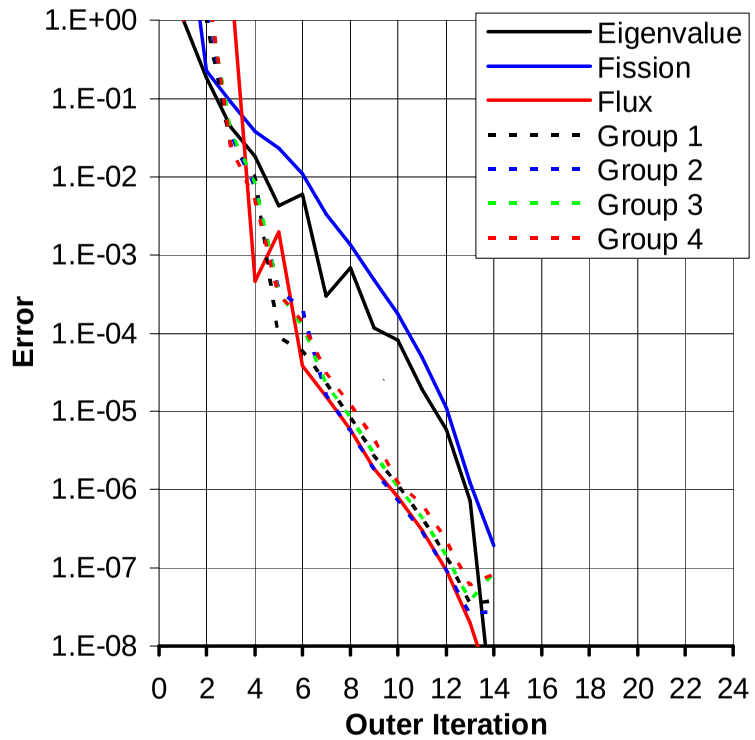
- Recently added capability to parallelize by space, angle, and energy in UNIC
  - Previous implementation only considered spatial parallelization
  - Memory limitations prevented large problems from being executed
- Implemented matrix-free CG solvers for PN2ND and SN2ND
  - New version uses single stenciled preconditioner matrix for each angle-group partition (far less memory)
  - Partitions can be solved simultaneously
  - Current fission source iteration scheme still uses Gauss Seidel in energy
- Added fission source acceleration
  - Tchebychev
  - Self-adjusting within-group flux error tolerance to minimize preconditioner effort
  - Time to solution reduced by 10+ factor



Sample Performance of Acceleration Schemes - Takeda Benchmark 4



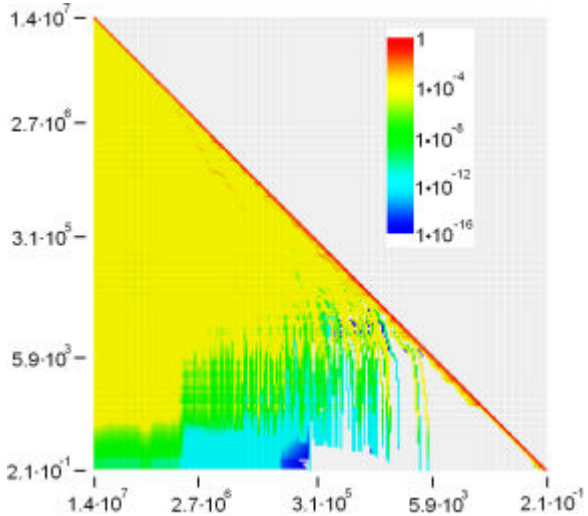
Without Tchebychev acceleration and a fixed tolerance of within-group flux error



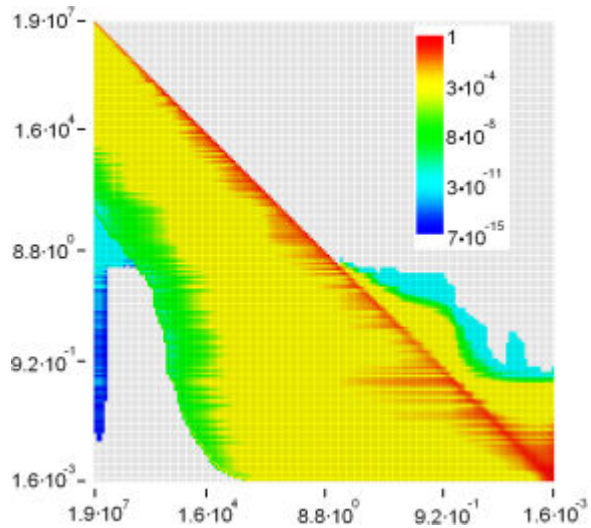
With Tchebychev acceleration and self-adjusting within-group flux error tolerance

Energy Group Parallelism Issues

Sodium Fast Reactor Fuel Assembly (230 Groups)



PWR Fuel Assembly (172 Groups)



Scattering Cross Section Stenciling

k-Eigenvalue Power Iteration

- **Steady state multigroup eigenvalue and/or fixed source iterations for one, two, and three dimensional unstructured finite element mesh geometries.**
- **Main loop (Power iteration)**

Begin Outer Iteration

Begin Loop over energy groups

Obtain group scattering+fission+fixed sources

Solve a symmetric positive definite linear system for flux

(preconditioned conjugate gradient)

End Loop over energy groups

Compute total fission source

Check for convergence in eigenvalue, angular flux, and sources

End Outer Iteration

Resource Requirements for UNIC Storage (PN2ND Solver)

- Massive linear systems
- *N* energy groups, *m* mesh points, *q* angular terms, and *p* nonzeroes per row, we need $m * q * p * 8 * 1.5$ bytes are required to store the matrix in compressed row format.
- Since this is a symmetric matrix, we need half of this.
- As an estimate, for 4000 nonzeroes in each row (p) for 10^8 rows ($m = 10^6$ and $q = 100$):
  - 2.4 terabytes (TB) of memory per group for the matrix alone (800 MB for one vector) and 24 petabytes (PB) for 10,000 groups.

Resource Requirements for UNIC Performance – A Crude Analysis

- The conjugate gradient (CG) method is $O[n * i]$ where

  n = number of non zeroes in the matrix $(p * m * q)$. For a well conditioned system, the number of iterations (i) needed to converge is small.
- Matrix vector product dominates the execution time
  - Two matrix-vector products per CG iteration
  - $2 * n$ floating point operations (flops) needed for each matrix vector product
  - Neglect the vector operations (dot products and scale) for serial case since $(n >> m * q)$

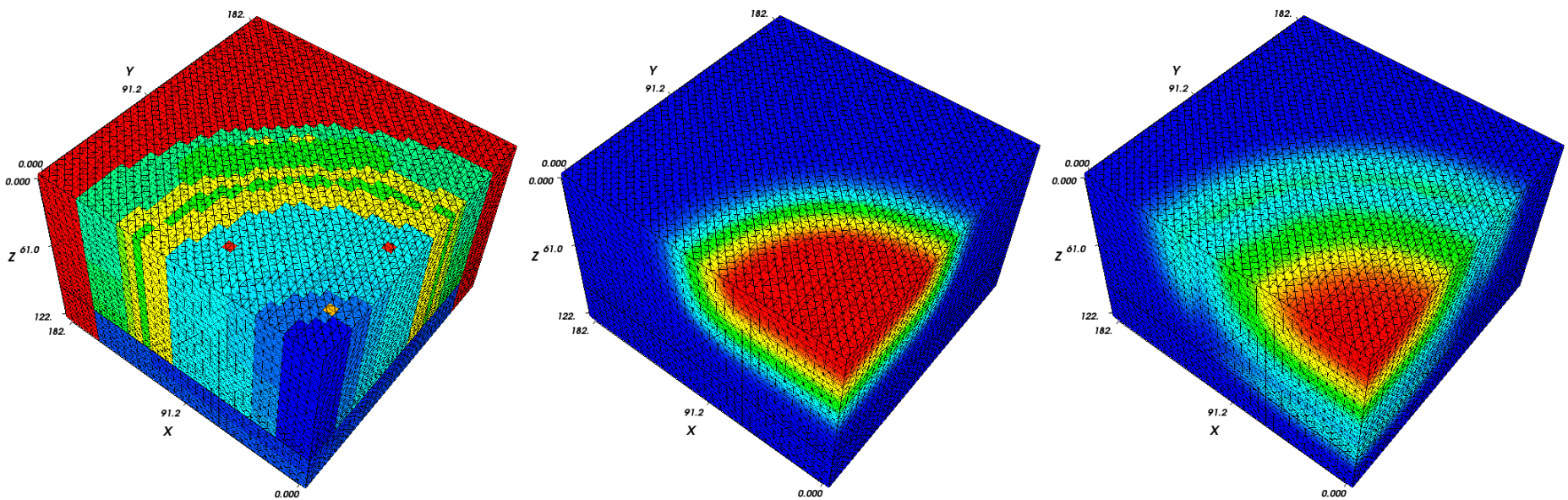
Resource Requirements for UNIC Performance (contd.)

- For i iterations of CG, O outer iterations, and N groups (assuming the same number of inner iterations for all the groups), the total flops are $4 * n * i * N * O$.
- CG performance is memory bandwidth limited
  - 10 - 20% of machine peak is practical
- *For* $i = 100$, $O = 50$, $N = 10{,}000$, and $n = 4 * 10^{11}$, the total work is 8×10^{19} flops (80,000 petaflops)
  - At 100 TFlop/s, 1111 hrs
  - At 1 PFlops/s, 111 hrs

Benchmark Problem Using UNIC PN2ND – ZPPR-15 Critical Experiment

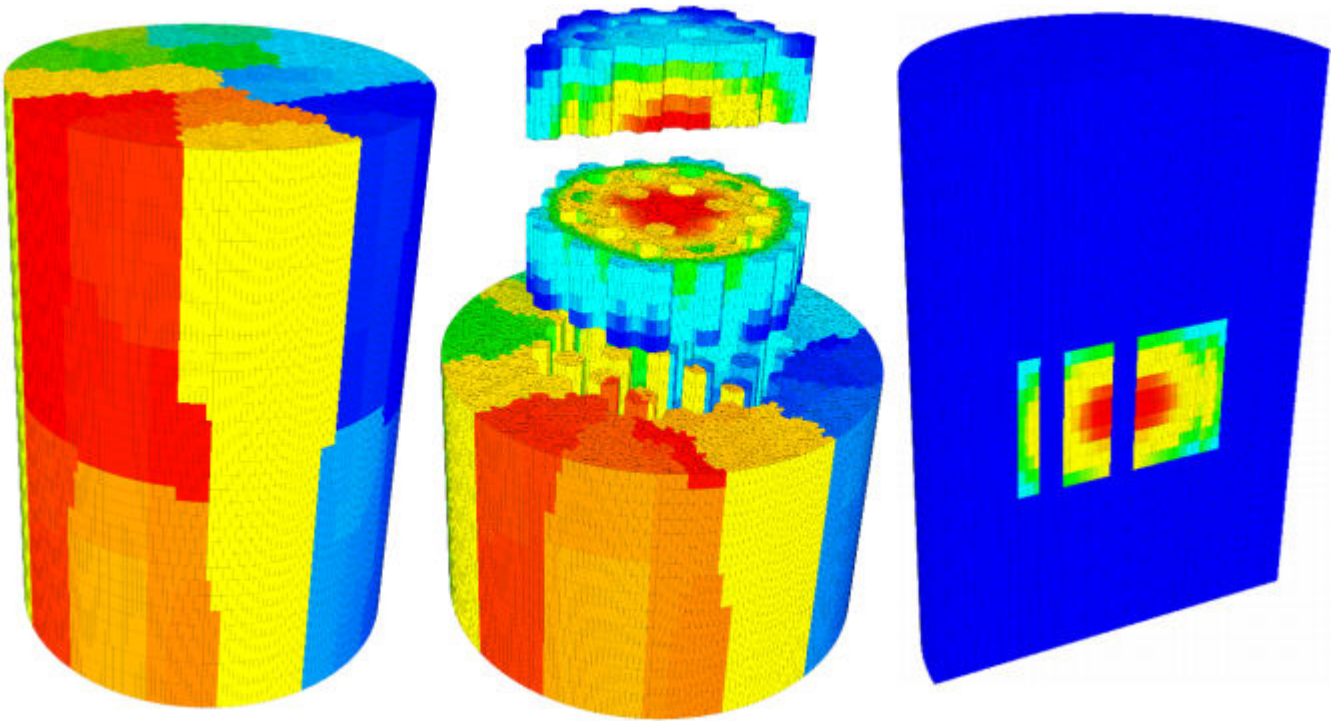
Mesh with 76,000 vertices, P3 scattering kernel and 230 energy groups

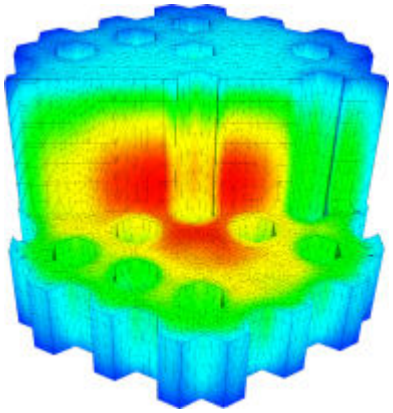
Flux expansion order	Scattering order	Eigenvalue
P_1	P_1	0.99258
P_3	P_3	0.99640
P_5	P_3	0.99651
Monte Carlo (VIM)		0.99647±0.00010



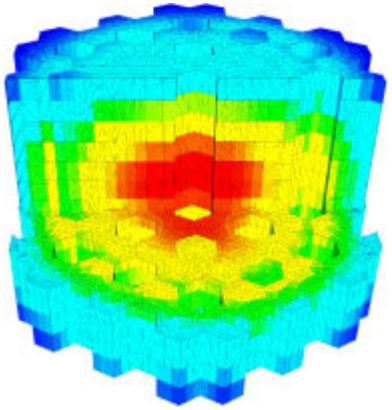
Computational Mesh and Example Flux Solutions of ZPPR-15 Critical Experiment

Power Distribution for ABTR Full Core Benchmark





Group 1 Flux

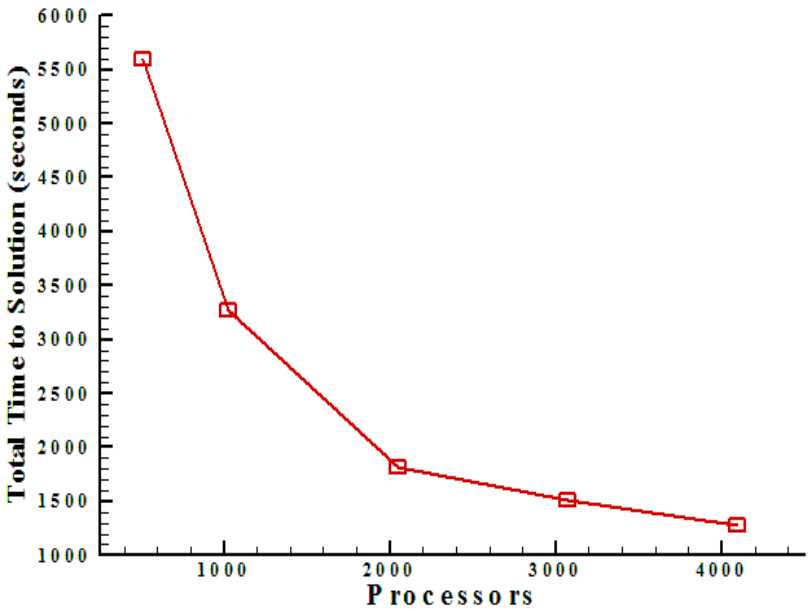


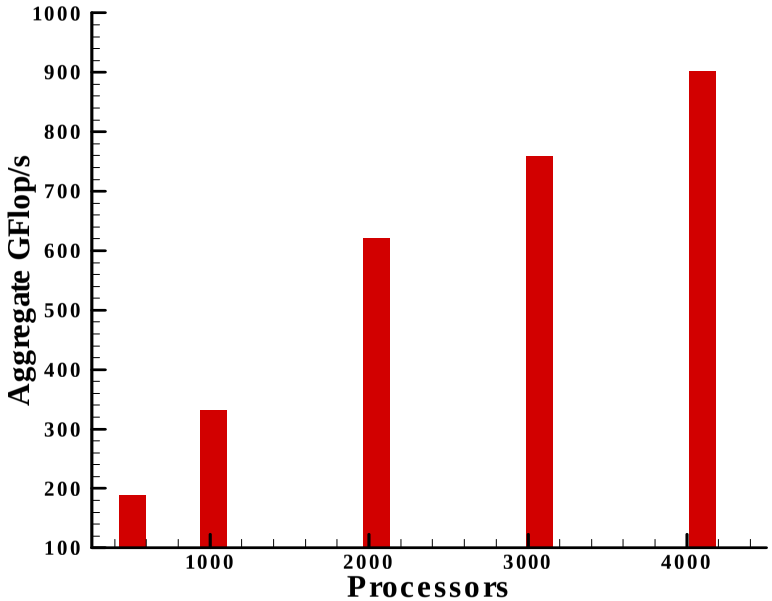
Power Distribution

Sample UNIC solution for ABTR. (Left) MeTiS decomposition for 512 processors (center) homogenized fuel regions are extracted and display the power distribution where the lower remnant displays the MeTiS decomposition and (right) vertical slice through the model showing the power distribution.

Performance of UNIC (PN2ND) on Cray XT4

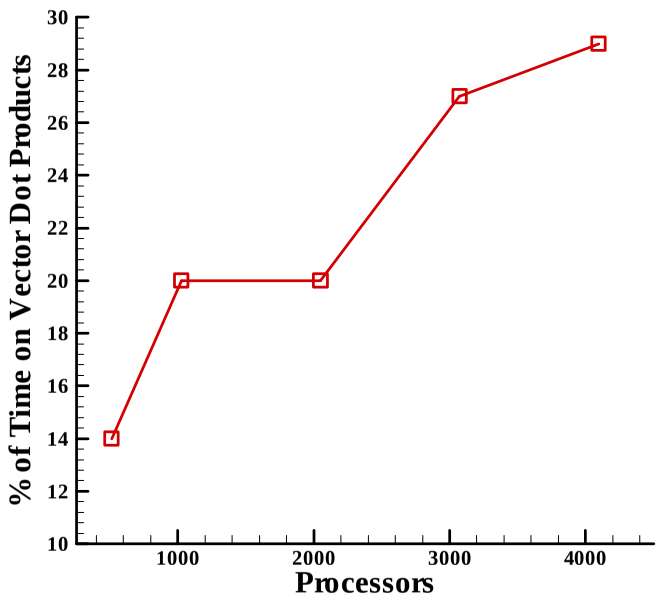
- Parallel performance from 512 to 4096 Cray XT4 processors
  - 120º periodic ABTR core with homogenized assemblies
  - 33 group P_5 calculation
  - Mesh contains 587,458 quadratic tetrahedral elements and 793,668 vertices
    - *About 12 million space-angle degrees of freedom per energy group*
  - About 900 GFlop/s (~4.2% of machine peak) on 4,096 processors

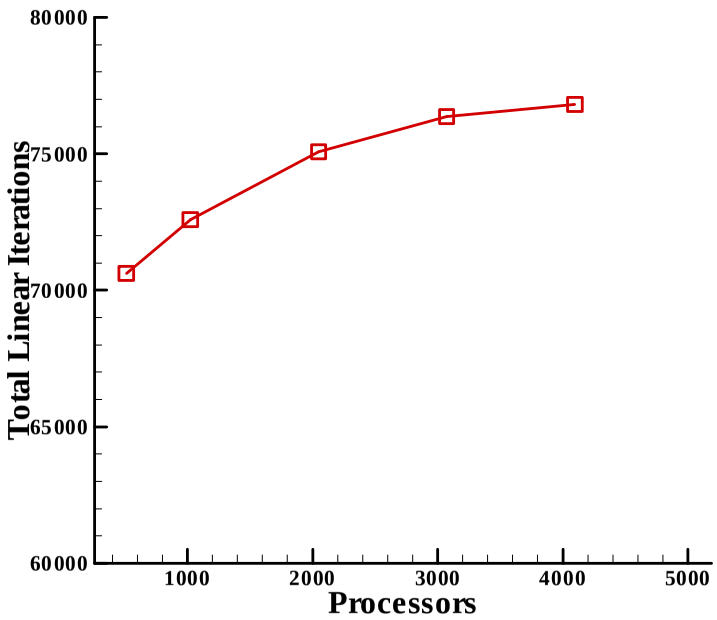




Performance bottlenecks

- Vector dot products
  - 14%-29%
  - Load imbalance
- Total linear iteration count
  - Increases as we grow the number of processors





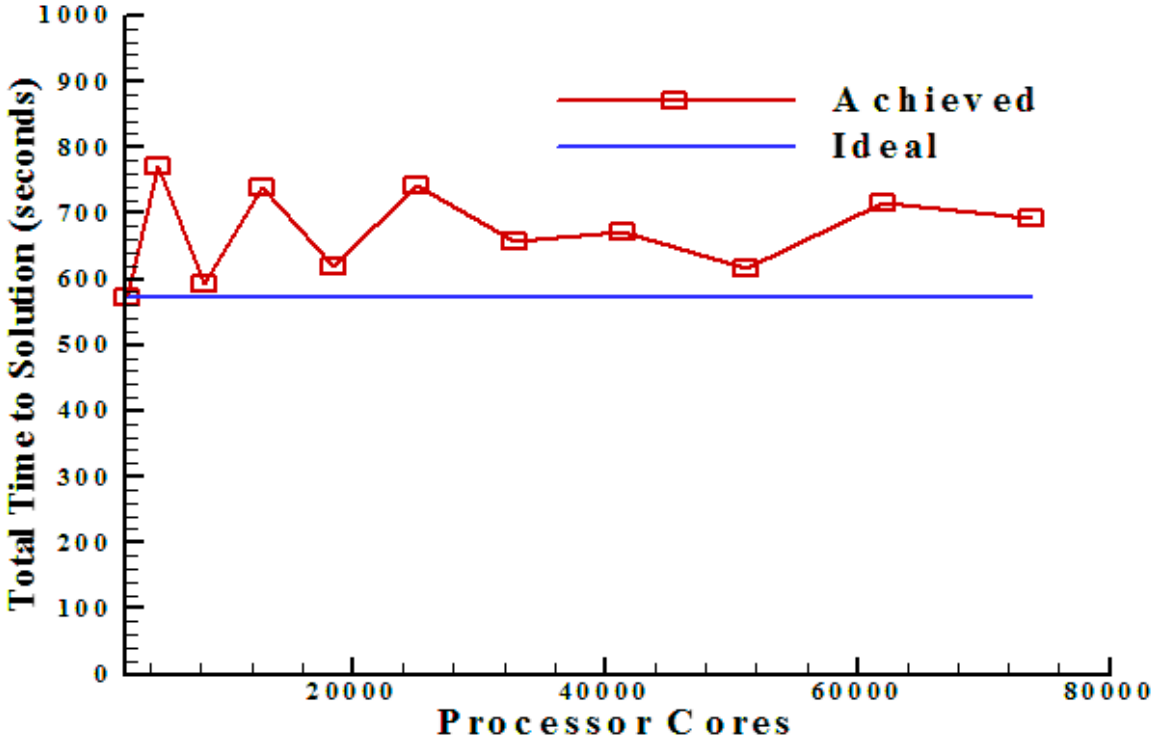
Angular Convergence of Eigenvalue of ABTR Core Using UNIC on BlueGene/P

- SN2ND Solver
  - 9 energy groups
  - Mesh (assembly homogenized)
    - *187,560 hexahedral quadratic elements and 785,801 vertices*
    - *Spread over 512 processor cores*

Processor Cores	Angular Resolution	Angles in 2π	Space-Angle DOF	Eigenvalue
2,048	1	4	3,143,204	1.00633
4,608	2	9	7,072,209	1.00823
8,192	3	16	12,572,816	1.00754
12,800	4	25	19,645,025	1.00776
18,432	5	36	28,288,836	1.00773
25,088	6	49	38,504,249	1.00781
32,768	7	64	50,291,264	1.00780
41,472	8	81	63,649,881	1.00782
51,200	9	100	78,580,100	1.00781
61,952	10	121	95,081,921	1.00782
73,728	11	144	113,155,344	1.00782

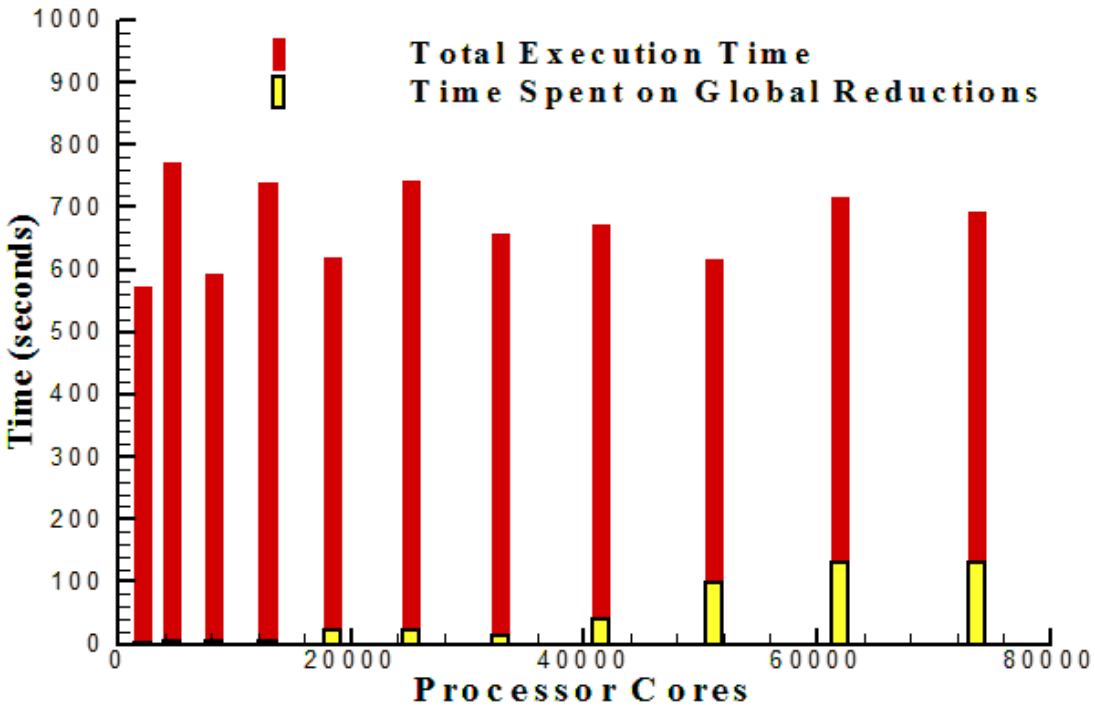
Weak Scaling of UNIC on BlueGene/P

- SN2ND Solver
- 2,048 to 73,728 Cores (Virtual Node Mode)



Weak Scaling of UNIC on BlueGene/P (contd.)

- Load imbalance in reduce operations
- Need to balance boundary vertices



Three Fundamental Limiting Factors to Peak Performance

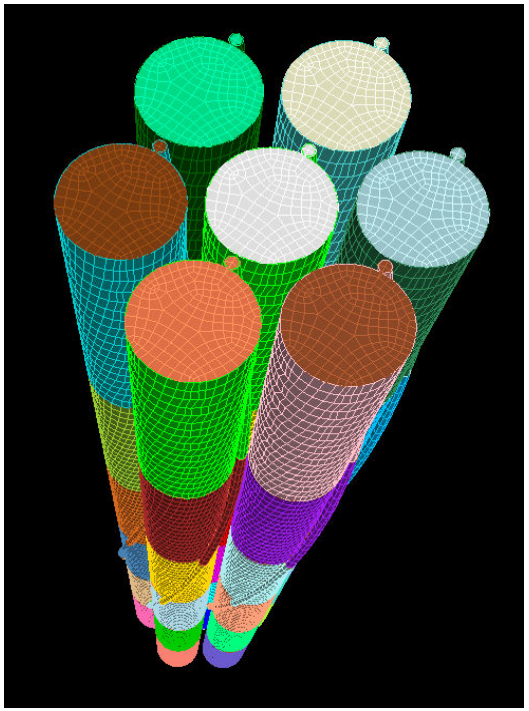
- Memory Bandwidth
  - Processor does not get data at the rate it requires
- Instruction Issue Rate
  - If the loops are load/store bound, we will not be able to do a floating point operation in every cycle even if the operands are available in primary cache
  - Several constraints (like primary cache latency, latency of floating point units etc.) are to be observed while coming up with an optimal schedule
- Fraction of Floating Point Operations
  - Not every instruction is a floating point instruction

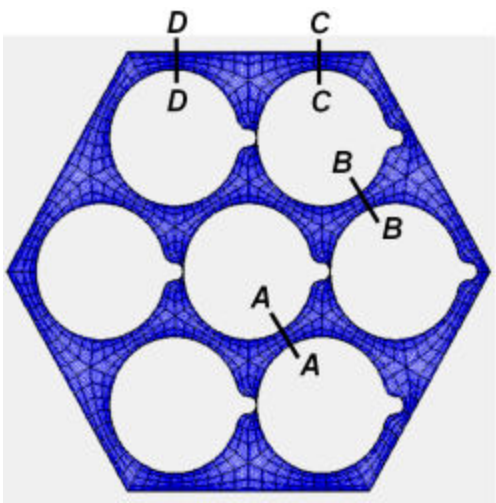
Path forward

- Load balancing to be handled through better partitioning
- Parallelize across groups to increase the amount of concurrency
- Explore the hybrid (mixed MPI/OpenMP) programming model
  - Better algorithmic convergence rate
- Memory reducing algorithms for matrix vector products
  - Tensor matrix vector product implementation
- More efficient custom preconditioners that take advantage of the matrix sparsity pattern

Wire-Wrapped Fuel Pin Assembly Setup Bottlenecks

- CAD-based geometry definition surprisingly difficult
  - “sweep with rotation” not supported/robust
- Various pieces of the process broke, forcing more decomposition
  - Swept+rotated surface, volume mesh
- No way to analytically specify geometry or mesh, though both conceptually straightforward
- MATLAB-based solution not scalable, portable, extensible
  - 37 pin, 217 pin problems needed too
- Real geometry would help with volume fraction calculation for wire homogenization





Approaches to TH analysis of subassemblies

DNS – direct numerical simulation of all scales	**parameter-free**	impractical
LES – large eddy simulation + dissipation	**parameter-free**	10^7 p. per channel
RANS – Reynolds-averaged Navier-Stokes	**tuning required**	10^5 p. per channel – *steady state*
Subchannel modeling	**empirical input**	100 p. per channel – *steady state*

- 400 x 200 subchannels in the core:
  - *Subchannel analysis will continue to be used for reactor design.*
  - *RANS will inform design process.*
  - *LES can help to validate / inform RANS and subchannel analysis.*

- *Hierarchical approach yields independent/redundant reference calculations*

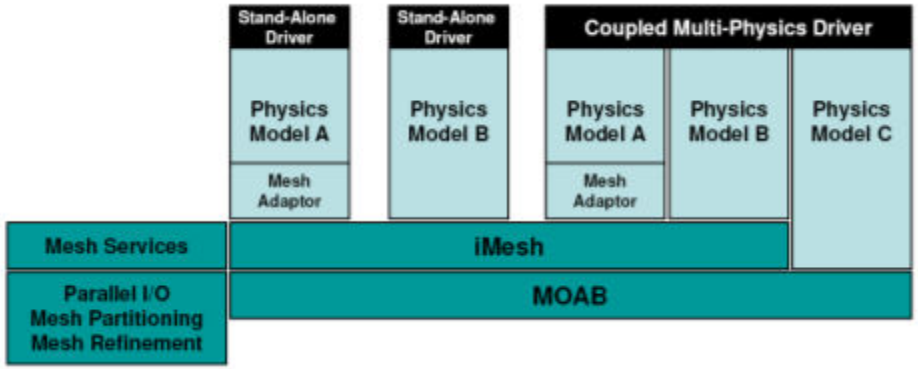
Approaches to TH analysis of subassemblies

DNS – direct numerical simulation of all scales	parameter-free	impractical
LES – large eddy simulation + dissipation	parameter-free	10^7 p. per channel
RANS – Reynolds-averaged Navier-Stokes	tuning required	10^5 p. per channel – *steady state*
Subchannel modeling	empirical input	100 p. per channel – *steady state*

- **400 x 200 subchannels in the core:**
  - *Subchannel analysis will continue to be used for reactor design.*
  - *RANS will inform design process.*
  - *LES can help to validate / inform RANS and subchannel analysis.*

- ***Hierarchical approach yields independent/redundant reference calculations***

SHARP framework Setup Bottlenecks



- Interoperability crucial
  - Interacting with multiple modules
  - Need variety of tools to operate on mesh & field data
- Coupling requires meshes & results in the same place
- Centralized parallel mesh infrastructure needed, to support coupling while allowing modules freedom
- Metadata needed to coordinate coupling (normalization/conservation)