Common Monte Carlo Design Tool (CMCDT) for Reactor Design and Analysis.

David P. Griesheimer[a], Presenter
The CMCDT Development Team[a,b]

April 9, 2008

[a] Bechtel Bettis, Inc.
[b] KAPL, Inc.

CMCDT/MC21 Development Team

- KAPL
  - Tom Sutton
  - Tim Donovan
  - Tim Trumbull
  - Pete Dobreff
  - Ed Caro
  - Kevin Sischo
- Bettis
  - Dave Griesheimer
  - Larry Tyburski
  - Dave Carpenter
  - Hansem Joo

Overview

- Monte Carlo Reactor Calculations
  - Historical
  - Future Directions
- Modern Challenges for Monte Carlo Methods
- CMCDT/MC21 Code Development Project
- Simplified Thermal Feedback for Monte Carlo

Introduction

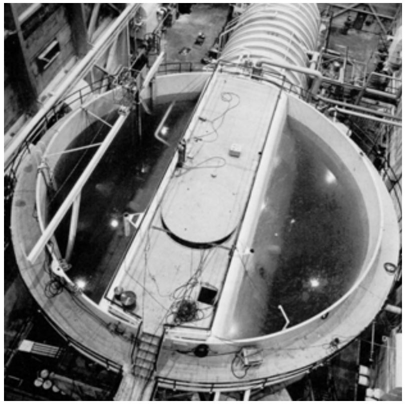
- Monte Carlo methods are currently the truth standard for radiation transport
  - Computationally expensive
  - Often reserved for complicated criticality or shielding problems

- Increasingly, Monte Carlo methods are being used for reactor analysis and design calculations
  - Even today Monte Carlo is often limited to verifying or qualifying results from deterministic diffusion or nodal models.

Historical Perspective

- Naval reactor laboratories (KAPL and Bettis) have long-term experience running Monte Carlo to calculate reactor:
  - Eigenvalue (k_{eff})
  - Flux / Power distributions
- Monte Carlo methods have played a major role in all naval nuclear design products since the early 1960's.
  - RECAP-1 (1963)
  - RCP01 (1978)
  - STEMB (1977)
  - RACER (1984)
- The "truth" modeling capability of Monte Carlo codes was an important step towards eliminating critical facilities.
- After 50 years of Monte Carlo development at Bettis and KAPL what is left to do?

NAUTILUS Prototype (1953)



Shippingport Atomic Power Station (1958)

Monte Carlo Future

- Objective:
  - Transition Monte Carlo from a stand alone static solver…
  - to an integrated reactor analysis package

- Provide support for
  - Modeling physical feedback effects
  - Criticality search capabilities to identify optimal
    - Assembly configurations
    - Fuel loadings
    - Control device positions
    - Soluble boron concentrations
  - Easily and quickly considering design permutations
  - Interfacing with external codes
    - Thermal analysis
    - Structural analysis
    - Safety analysis
    - Shielding analysis

Modern Challenges for Monte Carlo Methods

- Performance
  - Run time, variance reduction, parallelism
- Model complexity
  - Detail levels in full core models are expected to increase by 1 to 2 orders of magnitude
- Multiphysics feedback effects
  - Constant power depletion
  - Gamma heating
  - Thermal hydraulic feedback
  - Fuel growth
  - Rod search
  - Xenon feedback
- Become **THE** primary steady state analysis tool
  - Replace diffusion or nodal methods for all steady state design analyses.

Modern Challenges for Monte Carlo Methods

- Challenge 1: Performance
  - Monte Carlo methods are typically many times (~1000+) slower than corresponding deterministic solvers.
  - Monte Carlo algorithms are "embarrassingly parallel", meaning they scale well across many nodes on massively parallel scalar supercomputers (Beowolf / Linux Clusters)
  - New variance reduction techniques have the potential to increase the efficiency of Monte Carlo simulations, requiring fewer histories for a given level of accuracy.

Modern Challenges for Monte Carlo Methods

- Challenge 2: Model Complexity
  - Current codes can easily handle ~4 million compositions with ~100 million depleting nuclides.
  - The commercial nuclear industry, along with the DOE Global Nuclear Energy Partnership (GNEP) have projected that practical Monte Carlo nuclear design tools must handle 40-60 million compositions with up to 10 billion depletable isotopes.
  - Today a single calculation of this size would take ~5000 hours on a single processor.
  - Without substantial improvements in code efficiency, Monte Carlo methods will remain impractical for routine analysis of problems of this size until ~2030 (if Moore's law holds…).

Modern Challenges for Monte Carlo Methods

- Challenge 3: Multiphysics Feedback Capability
  - Basic feedback capabilities are required for Monte Carlo to compete with modern diffusion and nodal theory tools
    - Constant power depletion
    - Gamma heating
    - Thermal hydraulic feedback
    - Fuel growth
    - Rod search
    - Xenon feedback
  - Current research is looking at the first generation of multiphysics feedback treatments
    - Explicitly coupling Monte Carlo transport solver to external thermal or depletion solvers (MonteBurns, MCNP/StarCFD, etc.)
  - Future research will consider the effectiveness and efficiency of explicit code-to-code coupling and identify practical alternatives
    - In-line feedback solvers
    - Implicit solution techniques

Modern Challenges for Monte Carlo Methods

- Challenge 4: Become **<u>THE</u>** primary steady-state analysis tool
  - Move to Monte Carlo for all steady-state power distribution and reaction rate calculations
  - Benefits:
    - Increased accuracy and confidence in nuclear predictions
    - Elimination of cross section fitting and normalization steps
    - Exact geometry representation coupled with advanced model building tools will simplify the model creation process, requiring less engineer time to set up models
  - Challenges:
    - Limited by availability / capacity of computer resources
    - How to account for the presence of stochastic uncertainty in computed quantities?

Moving Forward

- How can we move forward towards our ultimate vision for Monte Carlo as a primary reactor design tool?
- Two options:
  - Continue to modify/improve existing MC codes... or
  - <u>Develop a new Monte Carlo code from scratch</u>

- In 2003, the Naval Reactor laboratories decided to begin joint development on a new Common Monte Carlo Design Tool (CMCDT).
  - Decision based on a desire to construct a new software foundation that will support years of new development.
  - Strong desire to adopt best practices from other successful Monte Carlo packages (RACER, RCP, MCNP, VIM, etc…)

Origins of CMCDT

- Development and programming on CMCDT began in mid 2004
- The CMCDT project has three components:

MC21

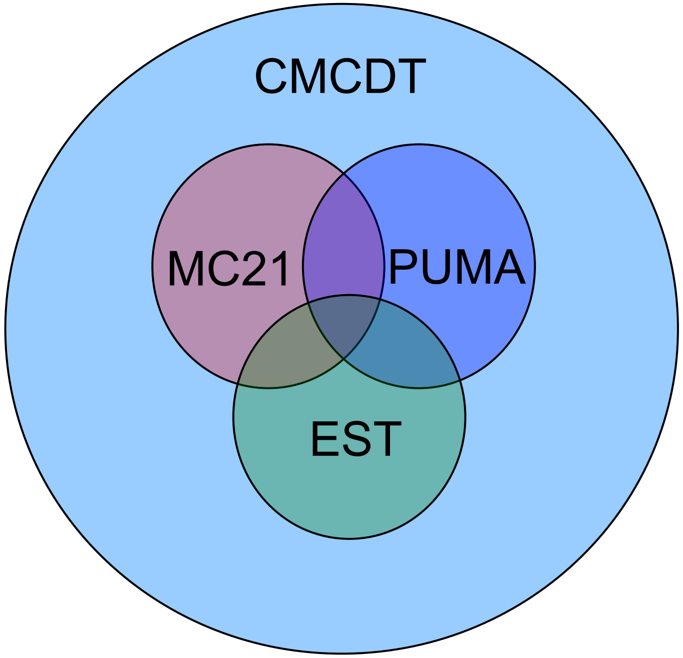
- Monte Carlo transport solver
- Cross section processing (NDEX)
- Scientific programming
- High performance computing

PUMA

- User interface – front / back end
- Model building / data processing
- Application (JAVA) programming

EST

- Engineering Support Team
- “Focus group” of senior engineers
- Guides development to meet user needs.



CMCDT scheduled to enter into production design usage in mid-2008.

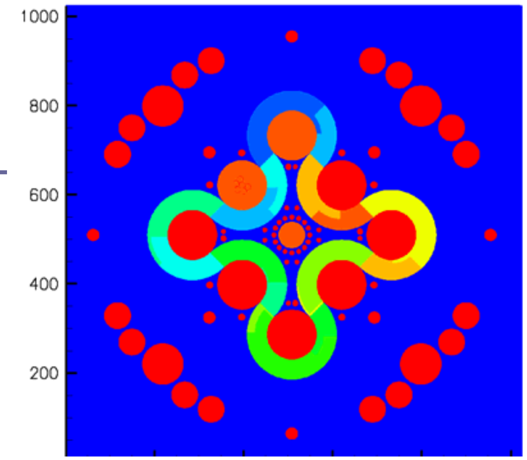
(after only 4 years of development!)

Motivation for a New Code

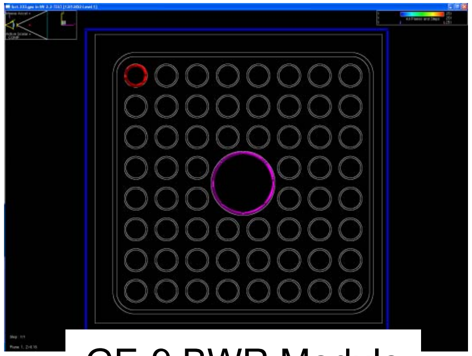
- Tight integration with a GUI modeling and post-processing system
  - Bettis and KAPL users learn just one interface
  - same GUI will be used for discrete ordinates code
- Designed and optimized for reactor design calculations
  - 100s of millions of tallies with minimal impact on run time
  - integrated support for depletion and automated control device movement for criticality searches
  - will eventually also support steady-state and peak xenon, thermal-hydraulic feedback and other features needed by reactor designers and analysts
  - neutron interaction physics treatments required for reactor calculations
- Modern coding
  - runs on a variety of single- and multi-processor platforms
    - uses both MPI and OpenMP
  - written in object-based Fortran 95 (GUI is written in Java)
    - easy to understand, maintain and modify

CMCDT Features

- Model Creation
  - PUMA provides model building, visualization and Q/A checking capabilities.
- Nuclear Data
  - NDEX & NDR provide a consistent and robust framework for generating nuclear data libraries for MC21
- MC21
  - State of the art nuclear physics treatments, and the capability to support tomorrow's physics analyses



Advanced Test Reactor (ATR)


GE-9 BWR Module

MC21 Transport Solver Kernel

- MC21 is a fully functioning Monte Carlo solver for neutron transport problems
  - Flexible geometry modeling system, which uses best practices from RACER, RCP01, MCNP5 and VIM
  - Includes all neutron physics required for Beginning of Life reactor analysis calculation
  - Generalized tally capability, which allows users to request detailed edits over the problem geometry
  - Directly coupled with sophisticated depletion solver (SPENT3)
  - Highly parallelized – performance scales linearly up to 100’s of nodes

CMCDT/MC21 Verification

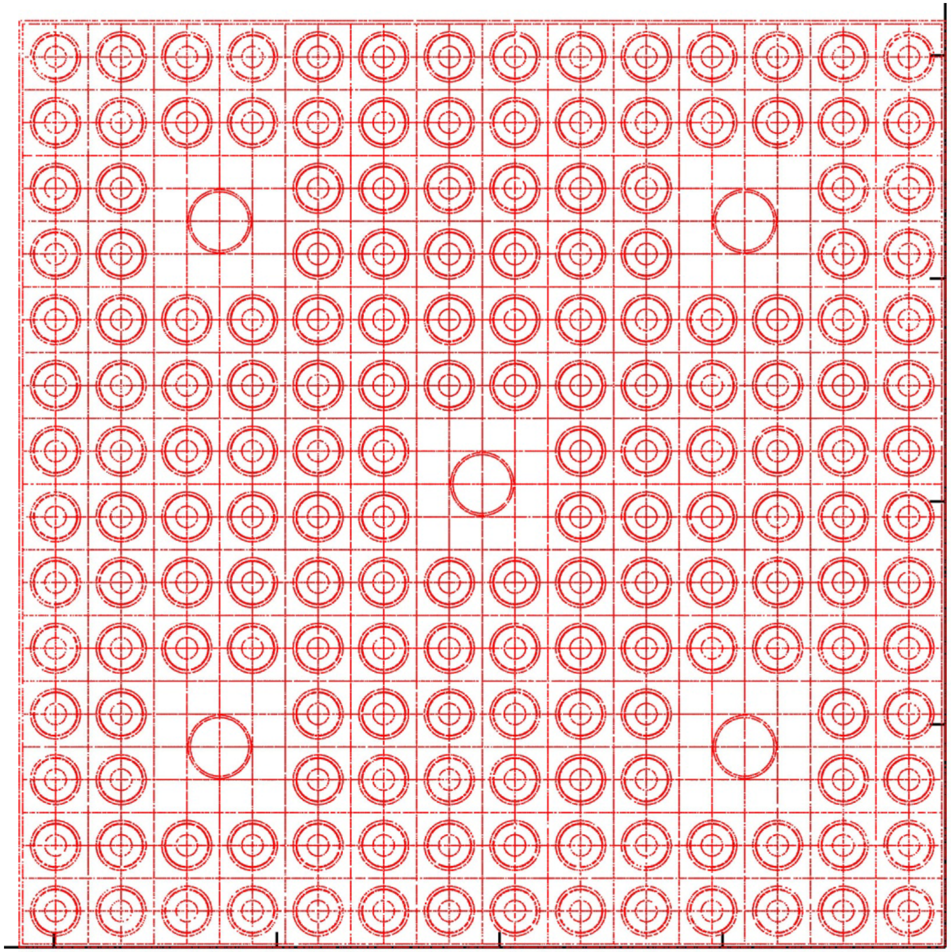
- As a part of the verification effort over 200 benchmark and reference models have been run in MC21.
- In all cases, results show excellent agreement with RCP01, RACER and MCNP5

Model* S(α,β) on (except godiva), no energy cutoff, no PTURR	**MC21 k-effective** 95% uncertainty on last sig. digits shown in parentheses	**MCNP5 k-effective** 1 StDev uncertainty on last sig. digits shown in parentheses
Godiva_1.18	0.9967(2)	0.9966(3)
HEU_MET_FAST_016	1.0007(4)	1.0002(2)
HEU_MET_FAST_030	1.0059(5)	1.0054(2)
HEU_SOL_THERM_001 case 1	0.9993(4)	0.9993(2)
* Original published MCNP Models have been modified slightly to ensure MC21 and MCNP consistency. Model names refer to ICSBEP identifiers for benchmark evaluations.		

CMCDT Objectives

- If CMCDT gives the same results as existing Monte Carlo codes, what was the point of writing a new code?
  - CMCDT provides a solid foundation for future growth and development in Monte Carlo methods.
  - Matching the capabilities of existing Monte Carlo analysis tools is simply a milestone.
  - Enabling future analyses that existing codes cannot handle is the real objective.

Calvert Cliffs Large Benchmark

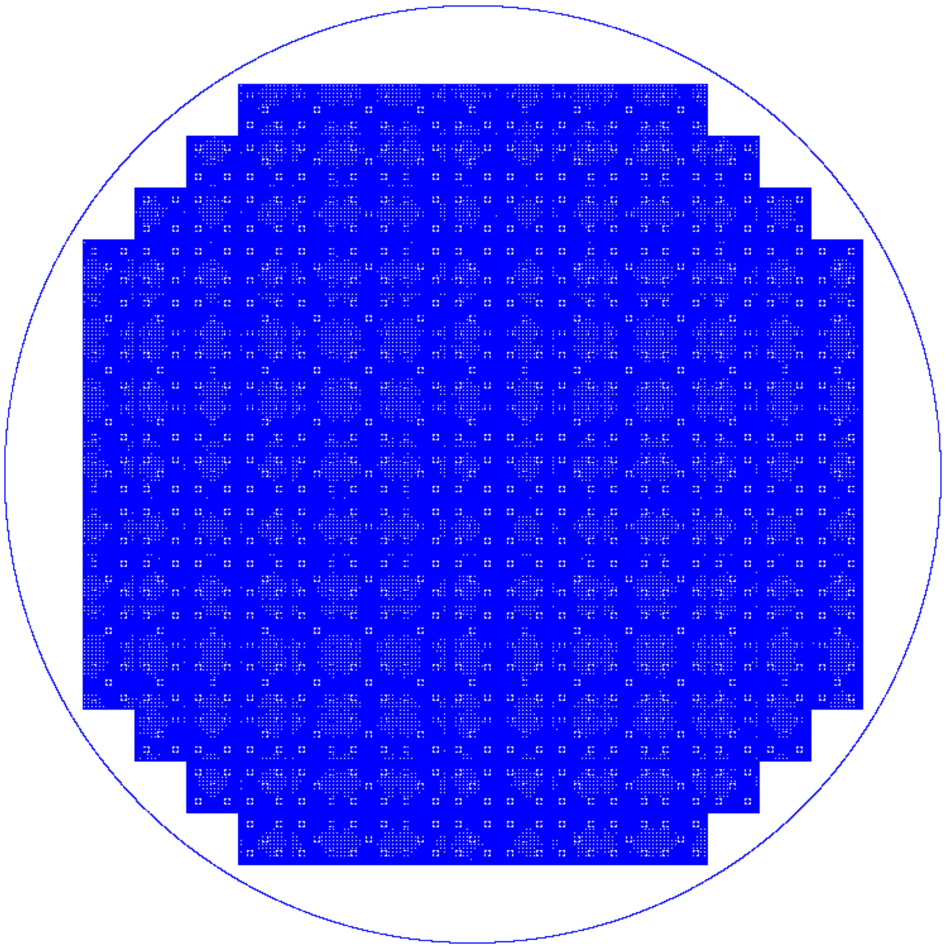


- 176 pins / assembly
- 8 depletion regions / pin
- 100 axial zones

- 140,000 depletion regions total

- 100 depletable isotopes
- 15 million tallies

Calvert Cliffs Large Benchmark



- 201 assemblies / core
- 28 million depletion regions total
- 2.83 billion tallies

- This is the size of the GNEP reference problem predicted for 2030.

- This problem is at the edge of our capability today.

Looking Forward

- Many achievements in novel Monte Carlo methods development
  - In-line depletion capability
  - Constant power depletion *(preparing draft for Math & Comp 2009)*
  - Rod search *(Math & Comp 2007)*
  - Fission source convergence acceleration *(Math & Comp 2007)*
  - Thermal feedback *(paper accepted for PHYSOR 2008)*

- Strong commitment to publishing results of our work
  - Newly developed algorithms and techniques are published in journals and conference proceedings
  - Algorithms are designed so that they can be applied to any Monte Carlo transport code, not just CMCDT/MC21

An Integrated Thermal Hydraulic Feedback Method for Monte Carlo Reactor Calculations

David P. Griesheimer[a], Dan F. Gill[b],
Jeffrey W. Lane[b], David L. Aumiller[a]

April 9, 2008

[a] *Bechtel Bettis, Inc.*

[b] *Pennsylvania State University, Department of Nuclear Engineering*

Thermal Feedback Overview

- Macroscopic cross sections are dependent on material temperature

$$\Sigma_t\left(E,T\right)=N\left(T\right)\sigma_t\left(E,T\right)$$

COOLANT → Material Density (impacts *N*)

Doppler Effect (impacts *σ*) ← FUEL

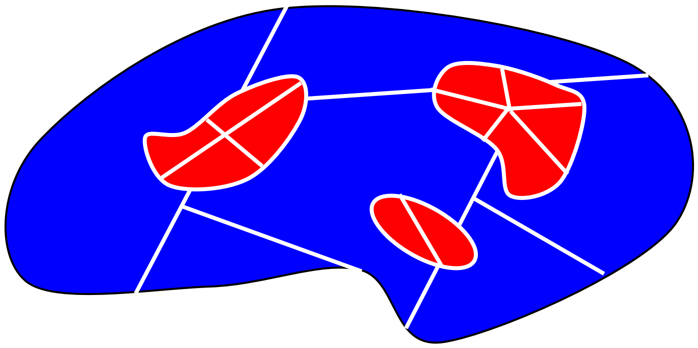
- For commercial LWR designs, temperature effects on power distributions and core reactivity are non-negligible
  - Temperature Defect: ~2-4% Δk/k
  - Power Defect: ~1-3% Δk/k

- For increased accuracy, local fuel and coolant variations should be taken into account during steady-state reactor calculations.
  - True **temperature feedback** requires iterations between the neutronic and thermal-hydraulic solvers to converge both temperature and flux.

MC Thermal Feedback Issues

- Monte Carlo Methods
- Advantages
  - Arbitrary geometry representation with combinatorial geometry (CG).
  - No structured grid restrictions.
  - Easy to calculate integrals over irregularly shaped regions.

Arbitrary Geometry Example



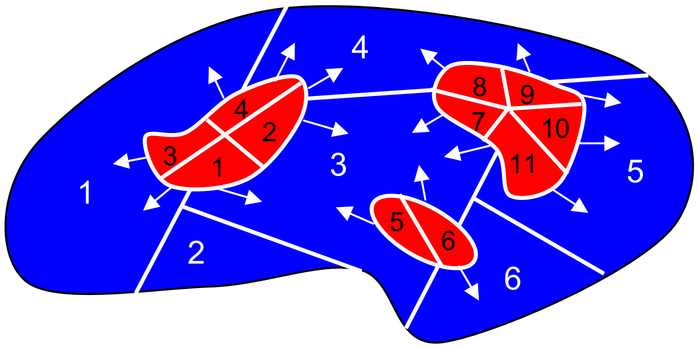
- Limitations
  - In CG representations there is no topographical information about connections between regions.
  - No way to tell how heat or coolant flows from region to region.

- Possible Solutions
  - Define a separate thermal solution grid overlay for geometry.
  - <u>Provide region connectivity information as problem input.</u>

MC Simplified Thermal Feedback

- Simplified thermal treatment is sufficient to capture important temperature effects
  - Users identify thermal source and sink regions in problem geometry.
  - Users define connectivity (heat and fluid flow paths) between thermal regions.

Arbitrary Geometry Example
(with user defined thermal regions)



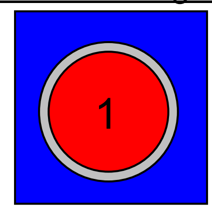
- Advantages
  - User controls how thermal regions are assigned and how heat and coolant flow in problem.
  - Uses the native Monte Carlo geometry, without the need for an additional thermal mesh.

- Disadvantages
  - Requires a lot of additional user input.
  - Nothing prevents the user from making poor or even unphysical assignments

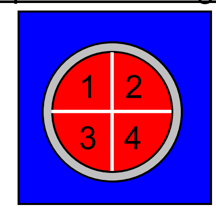
Source Region Definitions

- Thermal Source Region
  - Isothermal geometric region where heat is produced (fuel element).
  - Simplified heat transfer model assumes heat transfers directly from source to connected sink (coolant) region.
  - Thermal conduction between adjacent source regions is not allowed.
  - Fuel elements may be subdivided or collected together to form separate source regions.
  - Volumetric heat production in each source region is tallied during MC transport calculation.

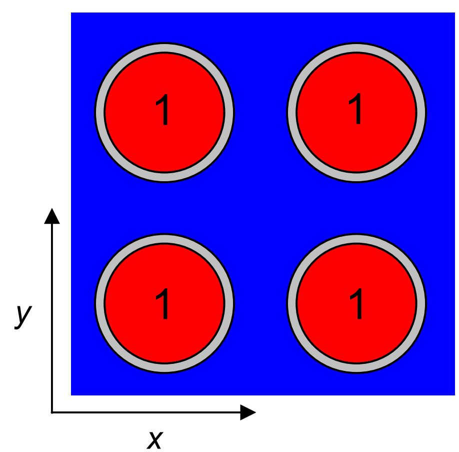
one fuel element
<u>one source region</u>



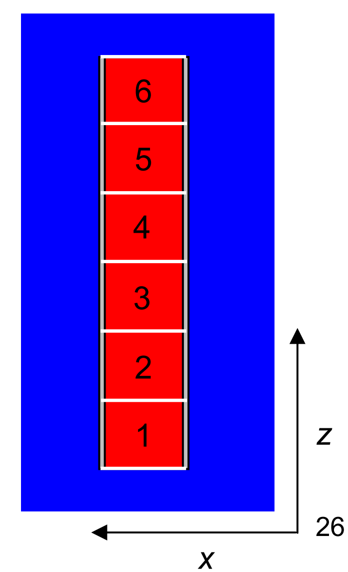
one fuel element
<u>multiple source regions</u>



multiple fuel elements
<u>one source region</u>



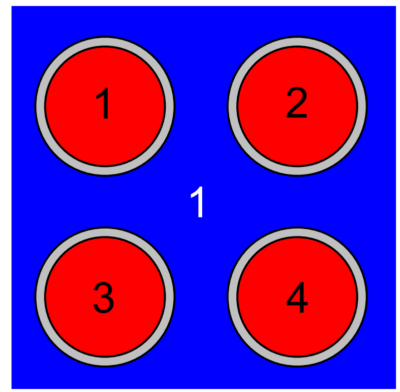
one fuel element
<u>multiple axial source regions</u>



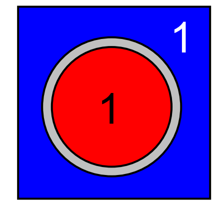
Sink Region Definitions

- Thermal Sink Region
  - Isothermal geometric region containing coolant.
  - Thermal conduction between adjacent sink regions is not allowed.
  - Coolant channels may be subdivided or grouped together to form separate sink regions.

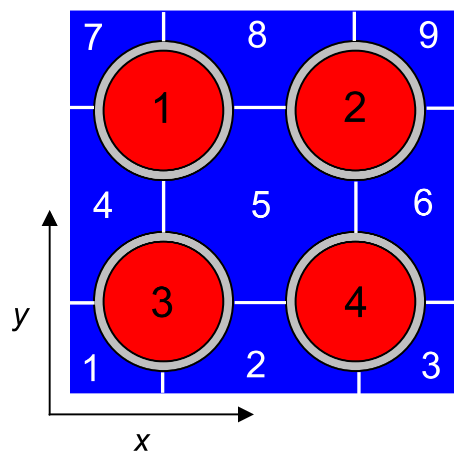
multiple source regions
one sink region



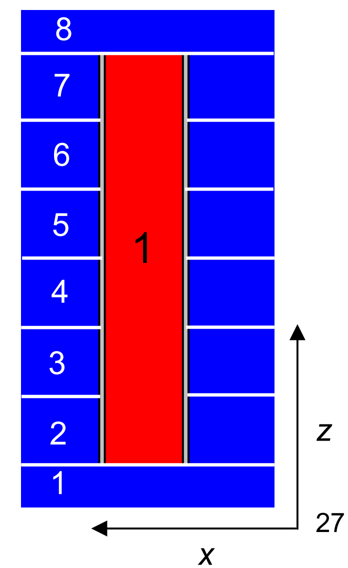
one fuel element
one source region



multiple sink regions
multiple source regions



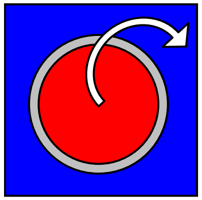
one fuel region
multiple axial sink regions



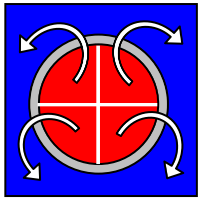
Source/Sink Heat Flow Connectivity

- Heat Flow Paths
  - Simplified heat transfer model assumes that heat produced in a source region is transferred directly to one or more sink regions.
  - The source/sink heat transfer ignores conduction effects through intervening materials.
  - These conduction effects are captured in a subsequent fuel heating calculation.

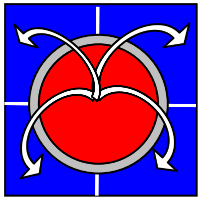
one source, one sink



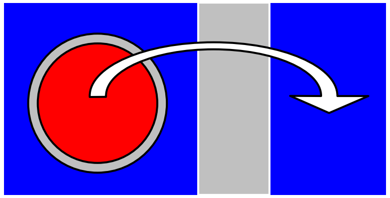
multiple sources, one sink



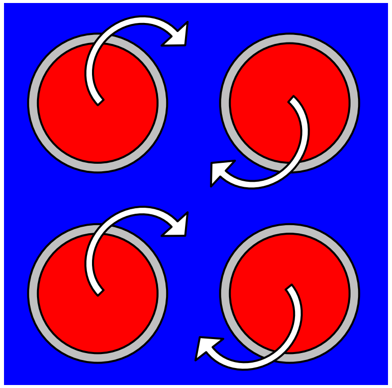
one source, multiple sinks



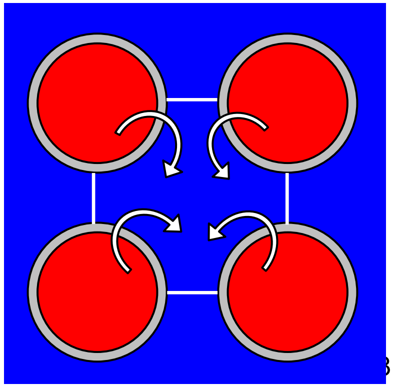
one source, unconnected sink(s)



multiple sources, common sink



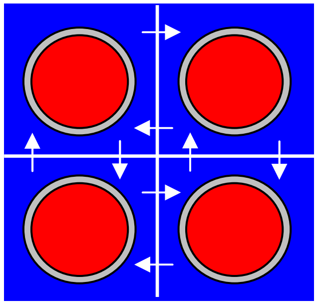
multiple sources, flow channel



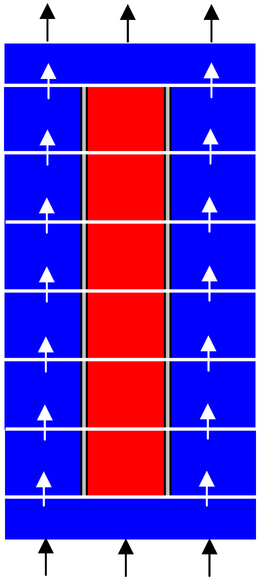
Sink/Sink Coolant Flow Connectivity

- Fluid Flow Paths
  - User defined connections between source regions determines how coolant flows through the system.
  - The flow path is used to determine the enthalpy rise of coolant as it travels through the reactor.
  - Users can specify axial, lateral, or mixing flow, with flow rates given as an absolute mass flow rate, or relative to total inlet flow into a region.

Lateral Flow

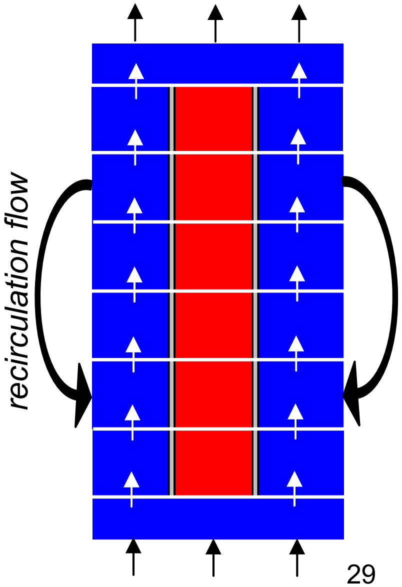


Axial Flow



inlet flow

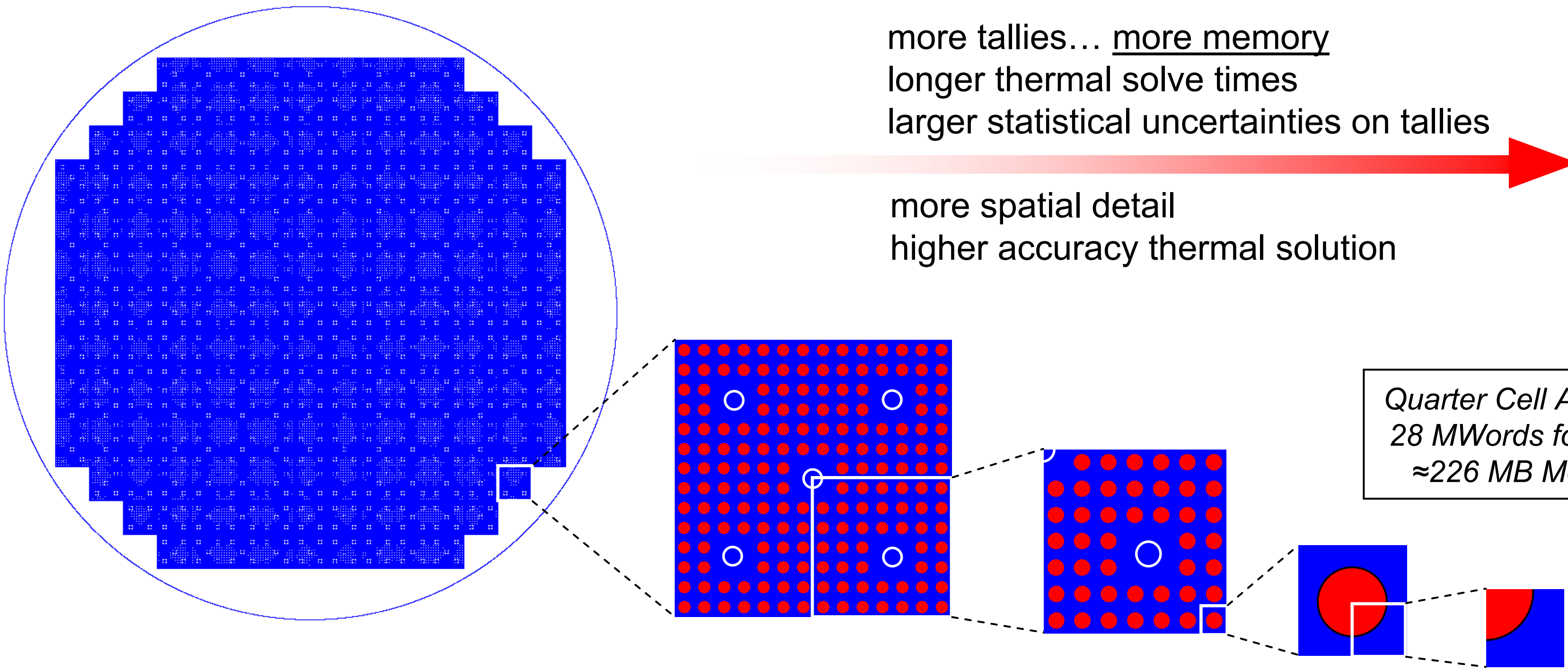
Mixing Flow



inlet flow

Flexibility in Thermal Solver

Calvert Cliffs Reactor – 50 Axial Zones



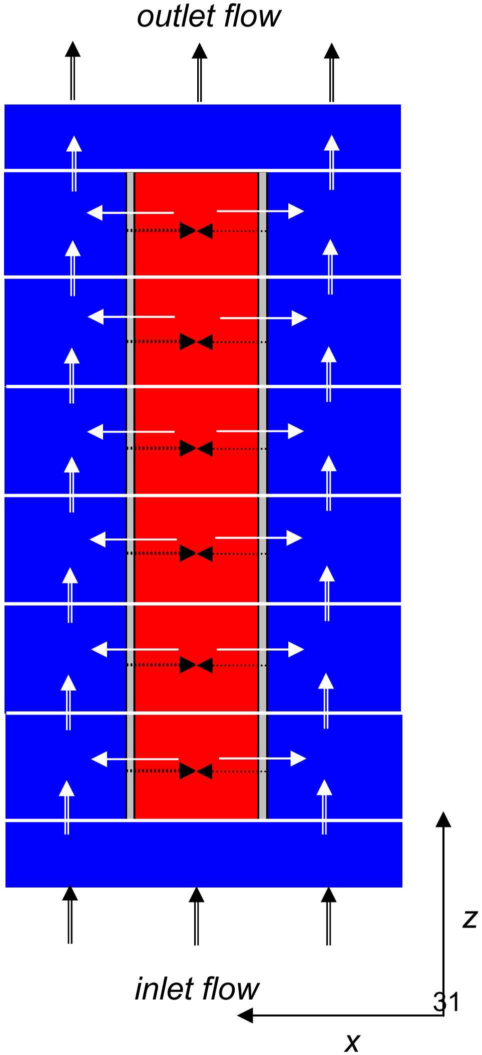
Quarter Cell Accuracy
28 MWords for tallies
≈226 MB Memory

Thermal Detail:	Full Core	Assembly	Q. Assembly	Unit Cell	Q. Cell
Source Regions:	50	10,050	40,200	7.08×10^6	28.3×10^6
Sink Regions:	50	10,050	40,200	7.27×10^6	29.1×10^6
Tally Memory:	0.4 kb	80 kB	321 kB	57 MB	226 MB

Thermal Solution Procedure

Three Step Thermal Solver

- Source/Sink Heat Transfer
  - Heat produced in each source region is transferred directly to connected sink region(s).
- Enthalpy Rise
  - Fluid flow / enthalpy rise calculation is performed
  - Produces updated temperature and coolant densities in each sink region.
- Fuel Heating (optional)
  - Solve 1-D thermal conductivity equation for source regions
  - Produces updated temperatures in each fuel source region.



Step 1: Heat Source Balance

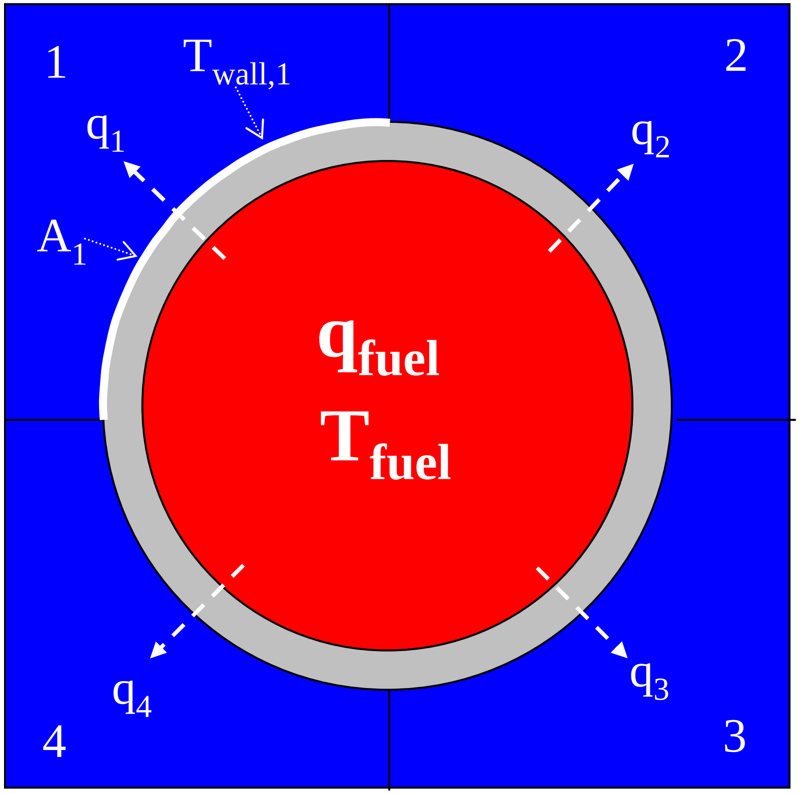
- Steady-state heat balance is used to determine what fraction of the total heat produced in the source region is delivered to each connected sink region.

- Heat is apportioned among adjacent coolant channels according to heat transfer coefficient, heat transfer surface area and interface wall temperature for each flow path.

$$q_i = h_i \left(T_{wall,i} - T_{fuel} \right) A_{surface,i}$$

- Energy balance is always preserved:

$$q_{Fuel} = \sum_{i}^{N} q_i$$



Step 2: Coolant Enthalpy Rise

- Apply simple enthalpy balance to each coolant region

$$q + \sum_{flow\,inlets} w_i \, h_{i,inlet} = h_{outlet} \sum_{flow\,outlets} w_j$$

  - w is mass flow rate of coolant into or out of sink region
  - q is total heat transferred into sink region from all adjacent source regions (obtained from step 1).

- Coolant regions are connected together through user defined coolant flow path:

$$w_{inlet,k} = w_{outlet,k-1}$$

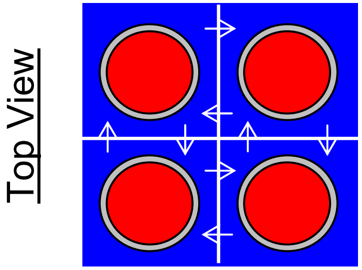
- Together, the coolant flow definitions and enthalpy balance form a sparse linear system of equations.

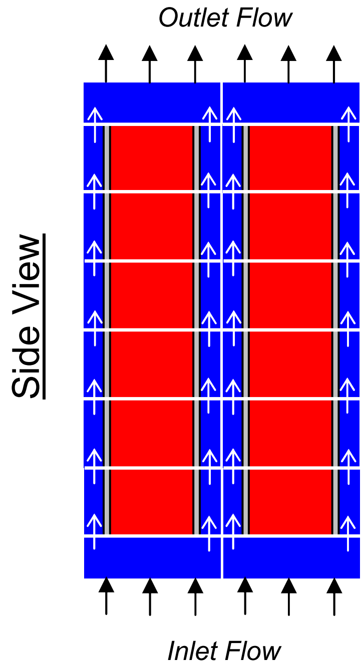
coolant mass flow matrix

Total heat deposition (by sink region)

$$\mathbf{M}\,\mathbf{h} = \mathbf{q}$$

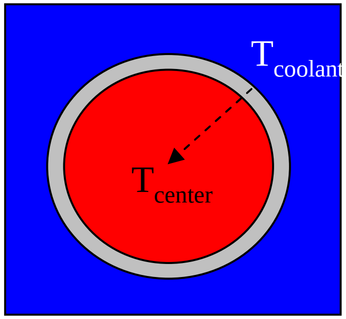
  - Solve with GMRES to obtain $\mathbf{h}$, the outlet coolant enthalpy in each sink region.
  - Coolant enthalpy is then converted to fluid properties (temperature, density, void fraction, heat transfer coefficient, etc.) with appropriate correlations.



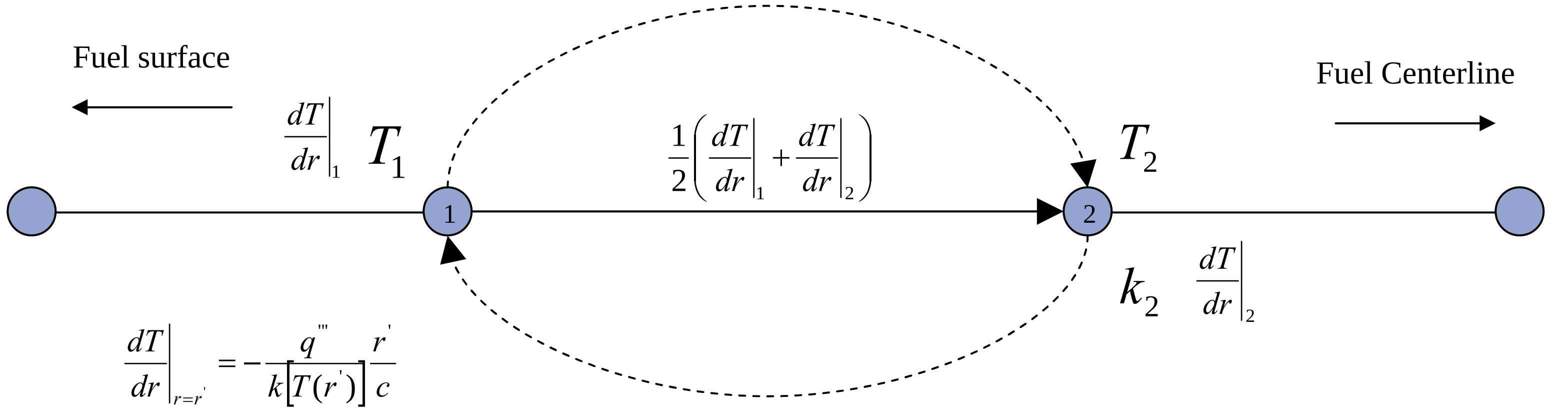


Step 3: Fuel Heating Calculation

- After coolant temperatures have been determined, use 1-D conduction models to calculate temperature rise in source regions.
  - Conduction models available for all three types of 1-D geometries (Cartesian, cylindrical, and spherical)
  - Effects of conduction through layer(s) of cladding, gap, or other structural material can be included by specifying an appropriate thermal resistivity value.



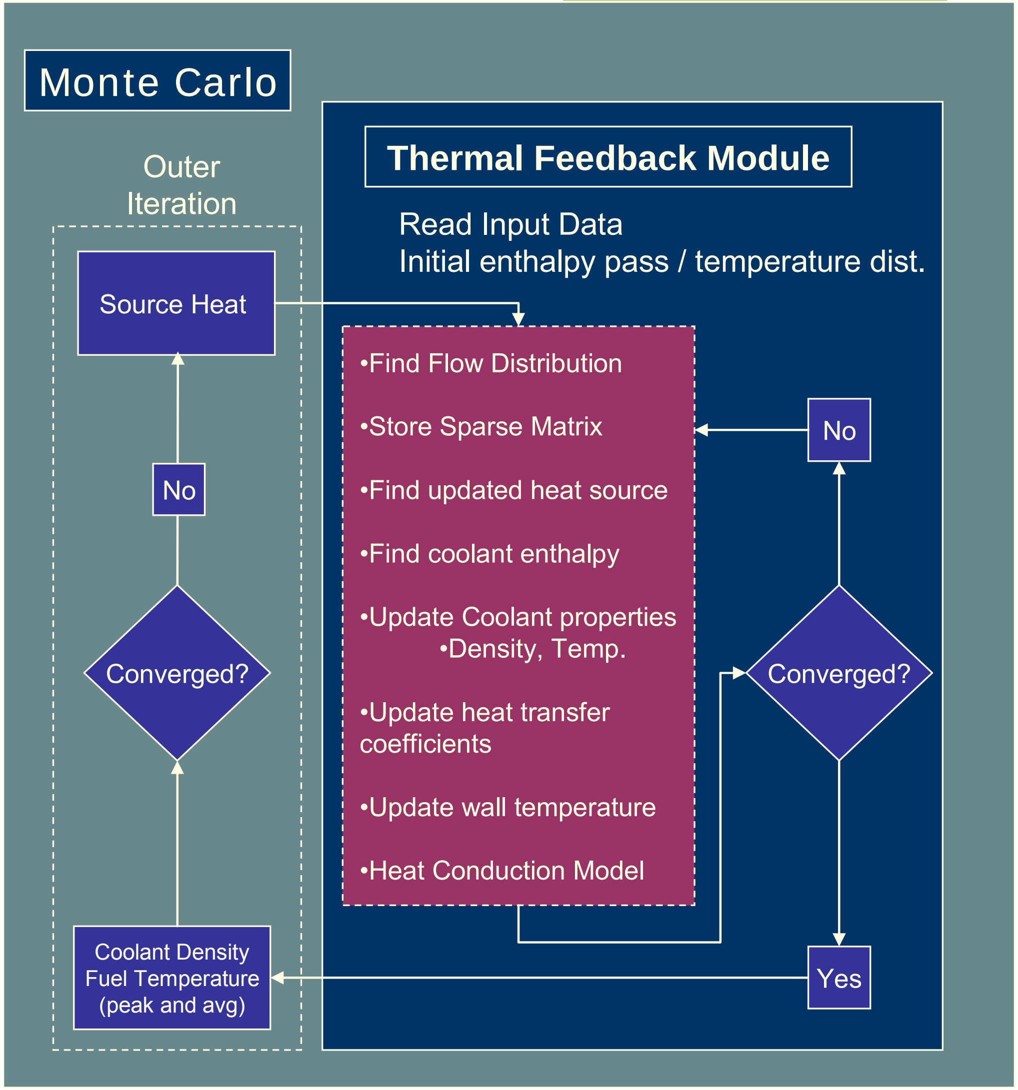
- Solution algorithm uses a forward marching predictor / corrector scheme to solve for temperature rise from the outer surface of the source region, inward.
  - Calculates source region temperature distribution.
  - Correctly accounts for variations in thermal conductivity with temperature.



$$\left.\frac{dT}{dr}\right|_{r=r'} = -\frac{q'''}{k\left[T(r')\right]}\frac{r'}{c}$$

Solution Process Overview

- Outer (Transport) Iterations
  - Solve MC transport calculation
  - Get heating tallies by source region
- Inner (Thermal) Iterations
  - Source/Sink heat balance
  - Enthalpy rise solution
  - Fuel heating calculation
- Method implemented and tested in an in-house continuous-energy Monte Carlo transport solver
  - Two benchmark problems
  - Only considered feedback effects of changes in coolant density.
  - Used constant temperature (543°K) cross sections for all regions.



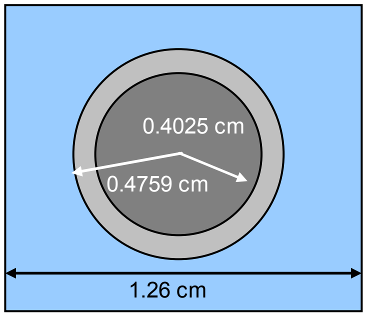
Simple PWR Fuel Pin

- Inlet Temp: 543 K
- Pressure: 15 MPa
- Flow Rate: 0.06843 kg/s

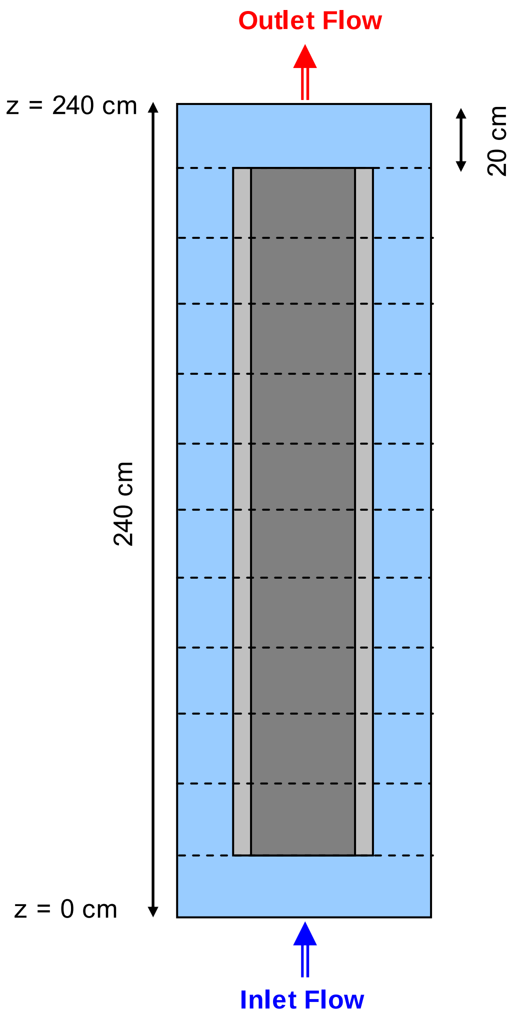
- 12 axial segments of 20 cm
- 26 Thermal Regions
  - 12 source, 14 sink
- 12 Outer (Transport Solve) Iterations
  - 1 source convergence
  - 10 thermal feedback
  - 1 edit iteration

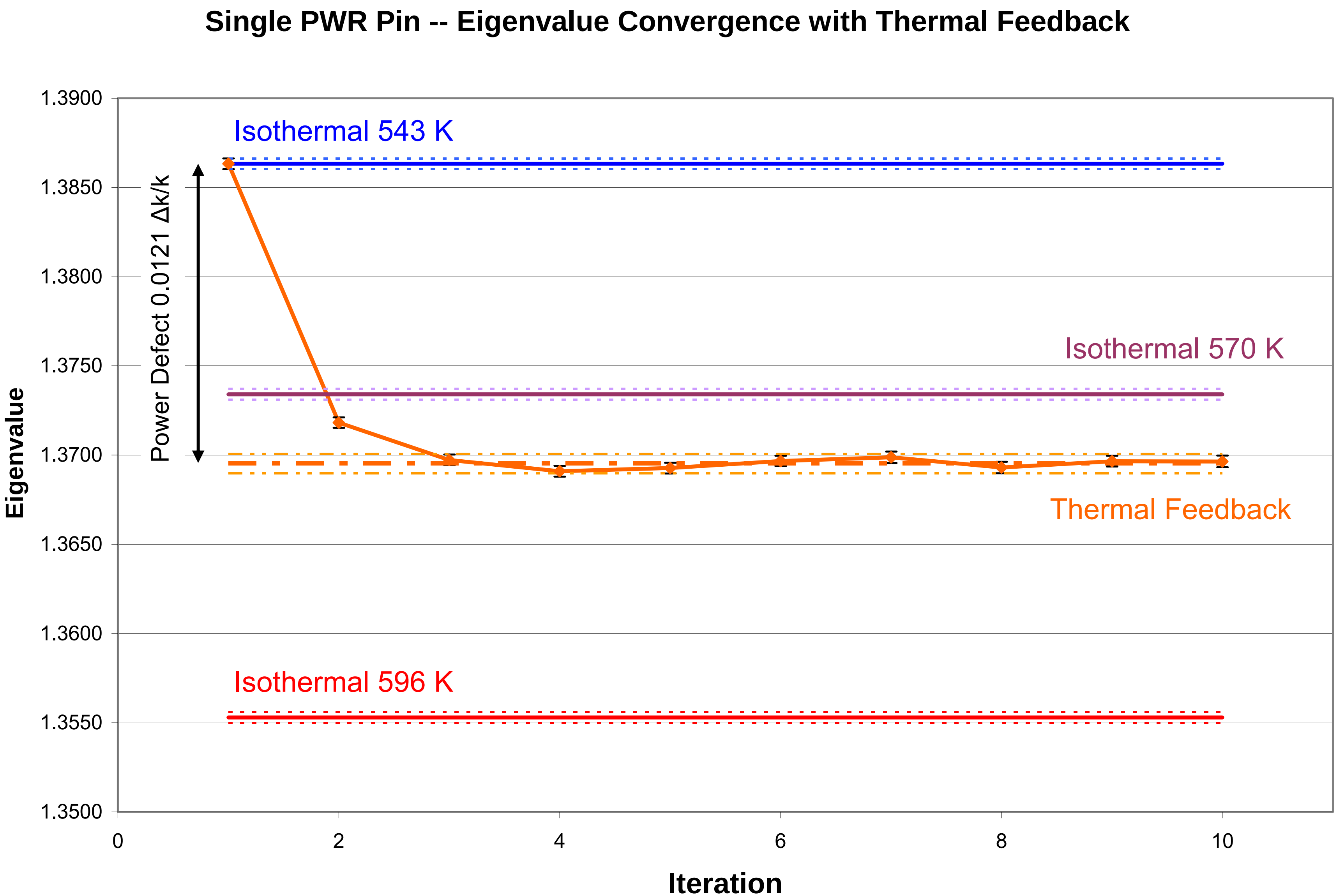
- Histories / batch = 40000
- Batches = 550
- Discard = 50
- Computer Statistics
  - 32 processors
  - 78.8 minutes
  - Avg. time of 0.01 seconds spent in thermal solver

Top View

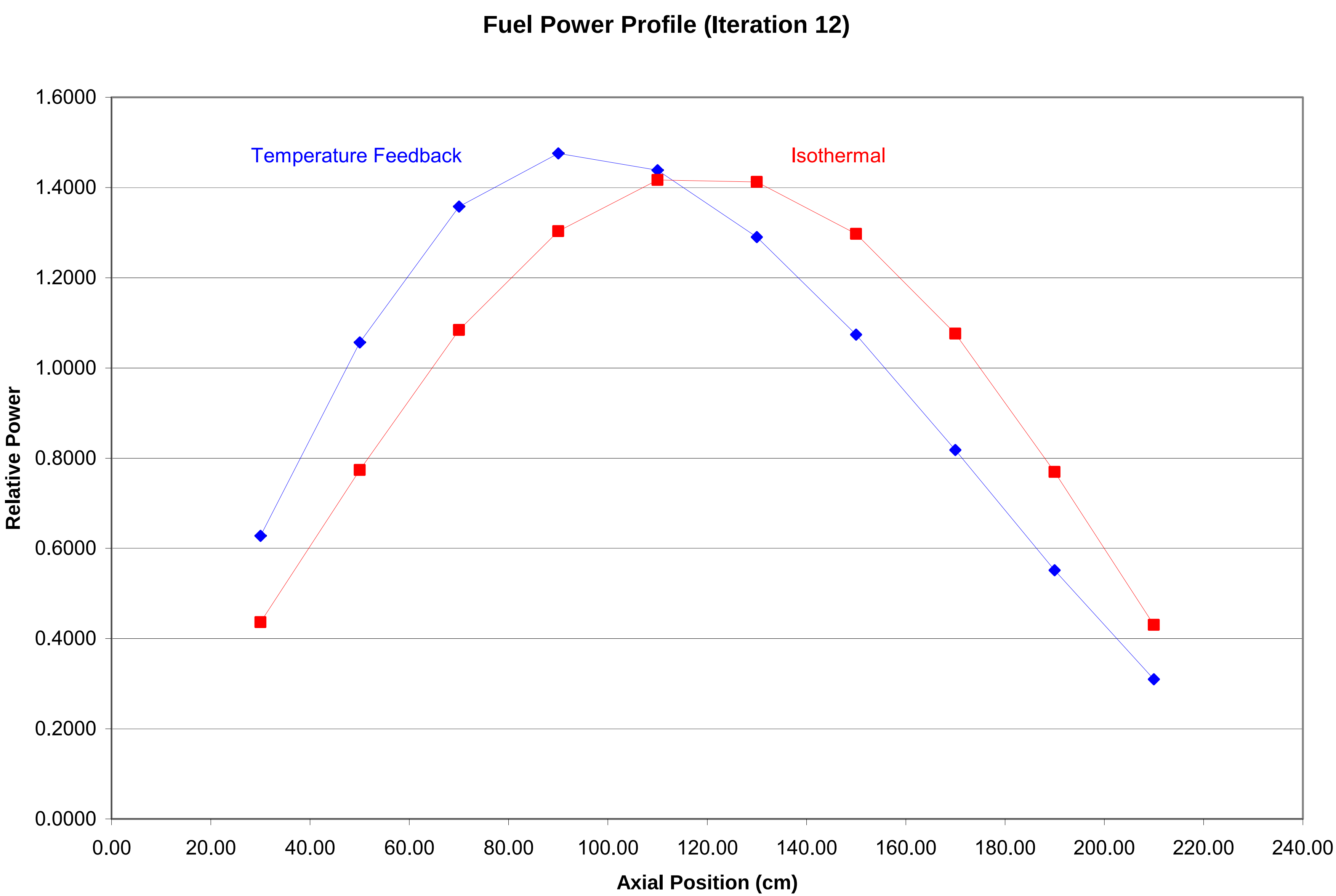


Side View





Temperature Defect (not shown) 0.0287 Δk/k



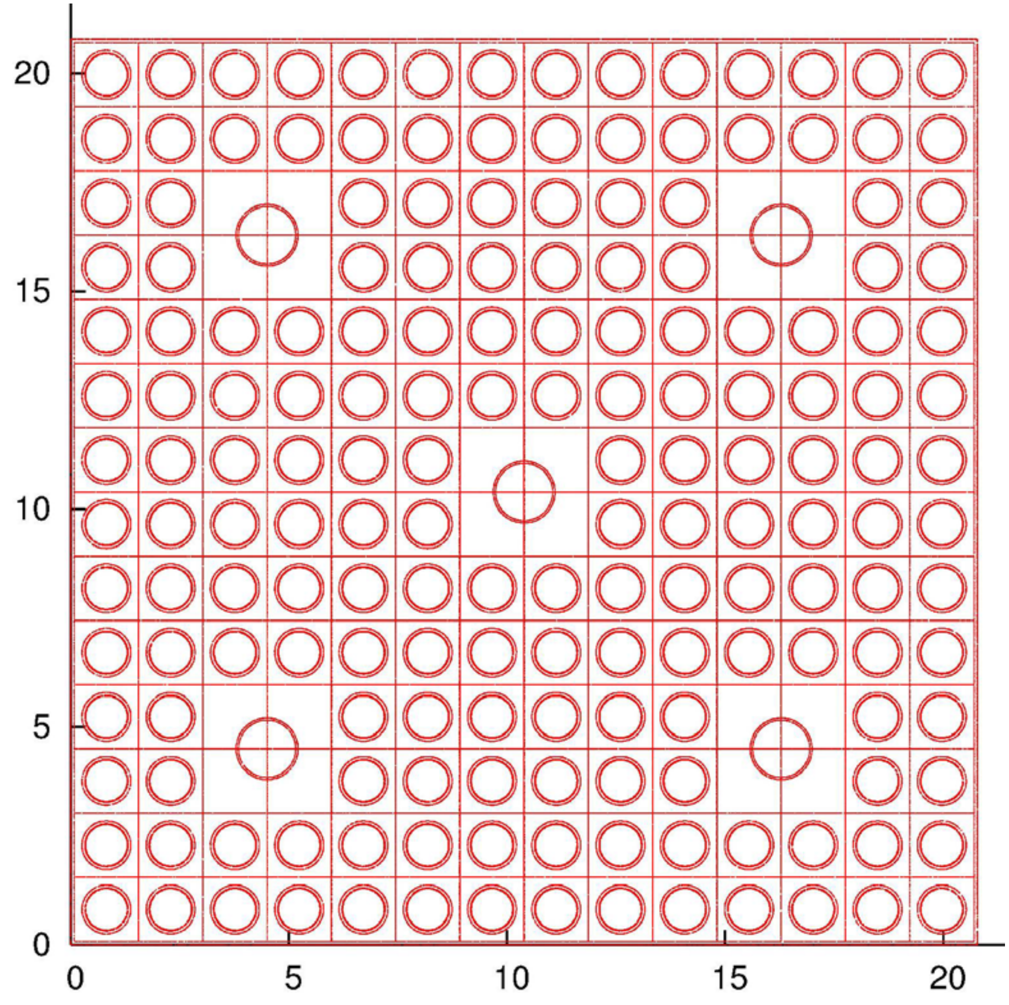
Relative uncertainty at 95% confidence level is less than 0.5% for all points

Calvert Cliffs Assembly

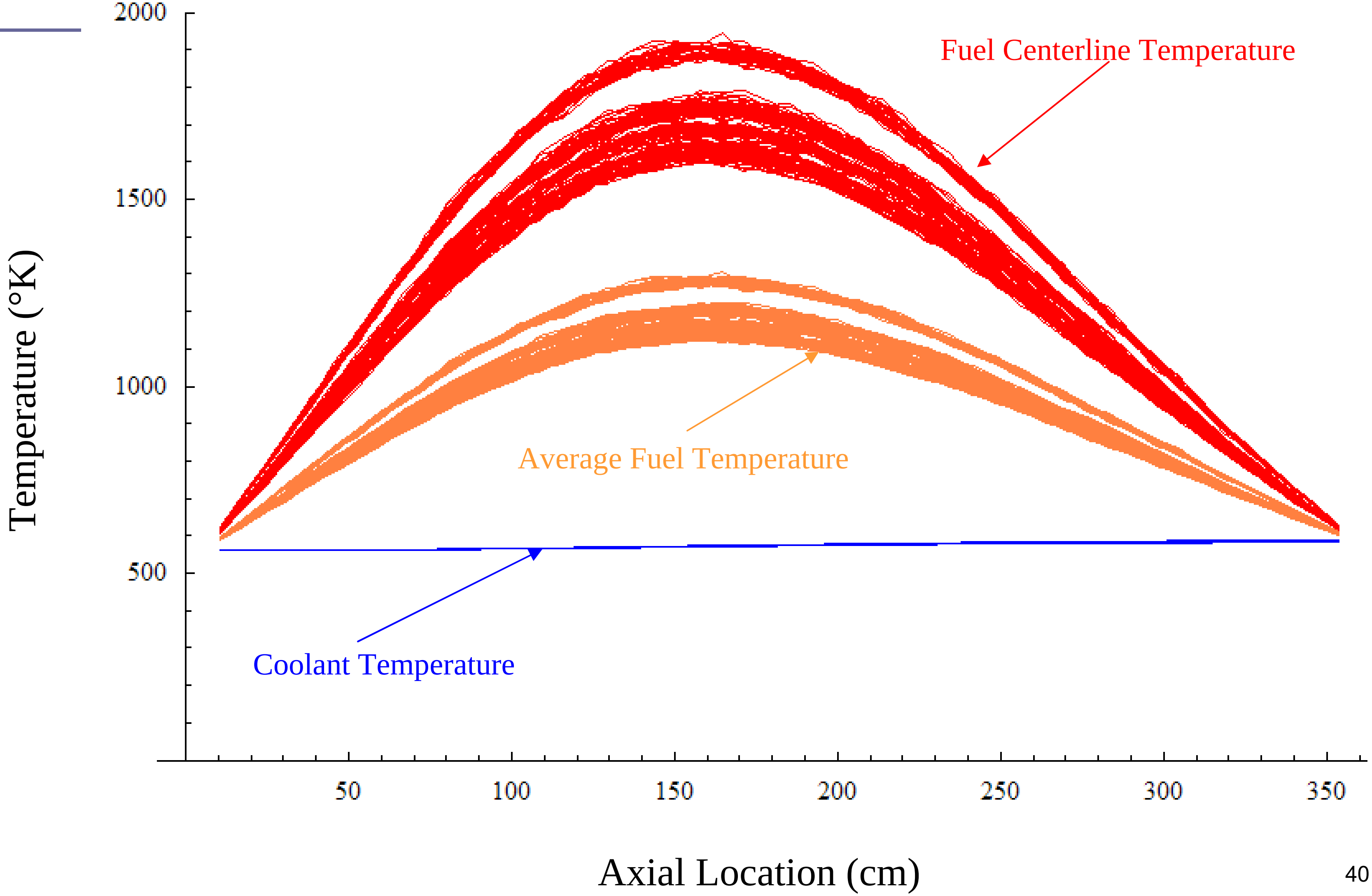
- Inlet Temp: 560 K
- Pressure: 15 MPa
- Flow Rate: 103.63 kg/s

- 50 axial segments of 7 cm
- 17,850 Thermal Regions
  - 8800 source, 9050 sink
- 12 Outer (Transport Solve) Iterations
  - 1 Flux
  - 10 Thermal Feedback
  - 1 edit

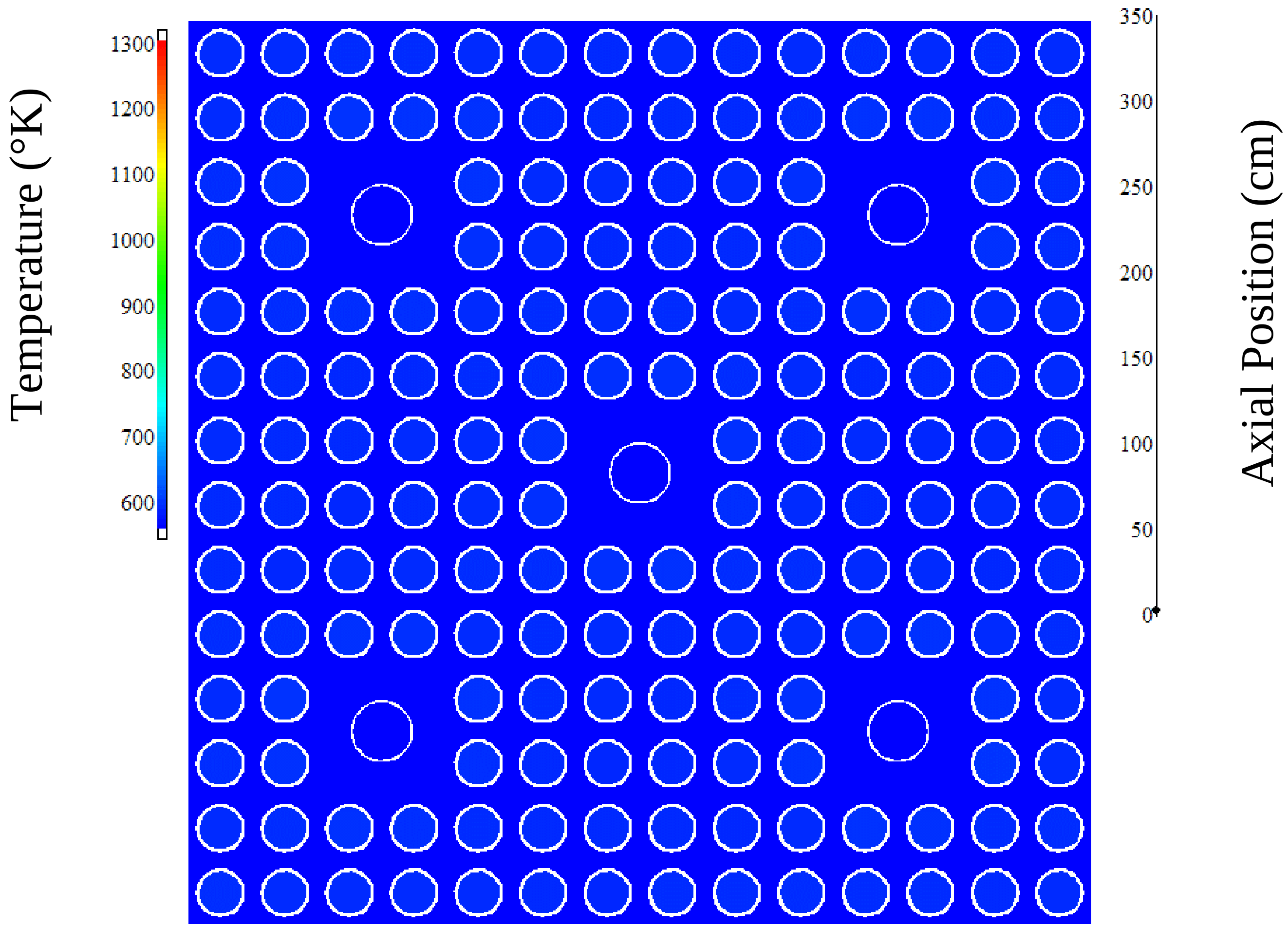
- Histories / batch = 250,000
- Batches = 250
- Discard = 50
- Computer Statistics
  - 64 processors
  - 117 minutes
  - Avg. time of 0.15 seconds spent in thermal solver



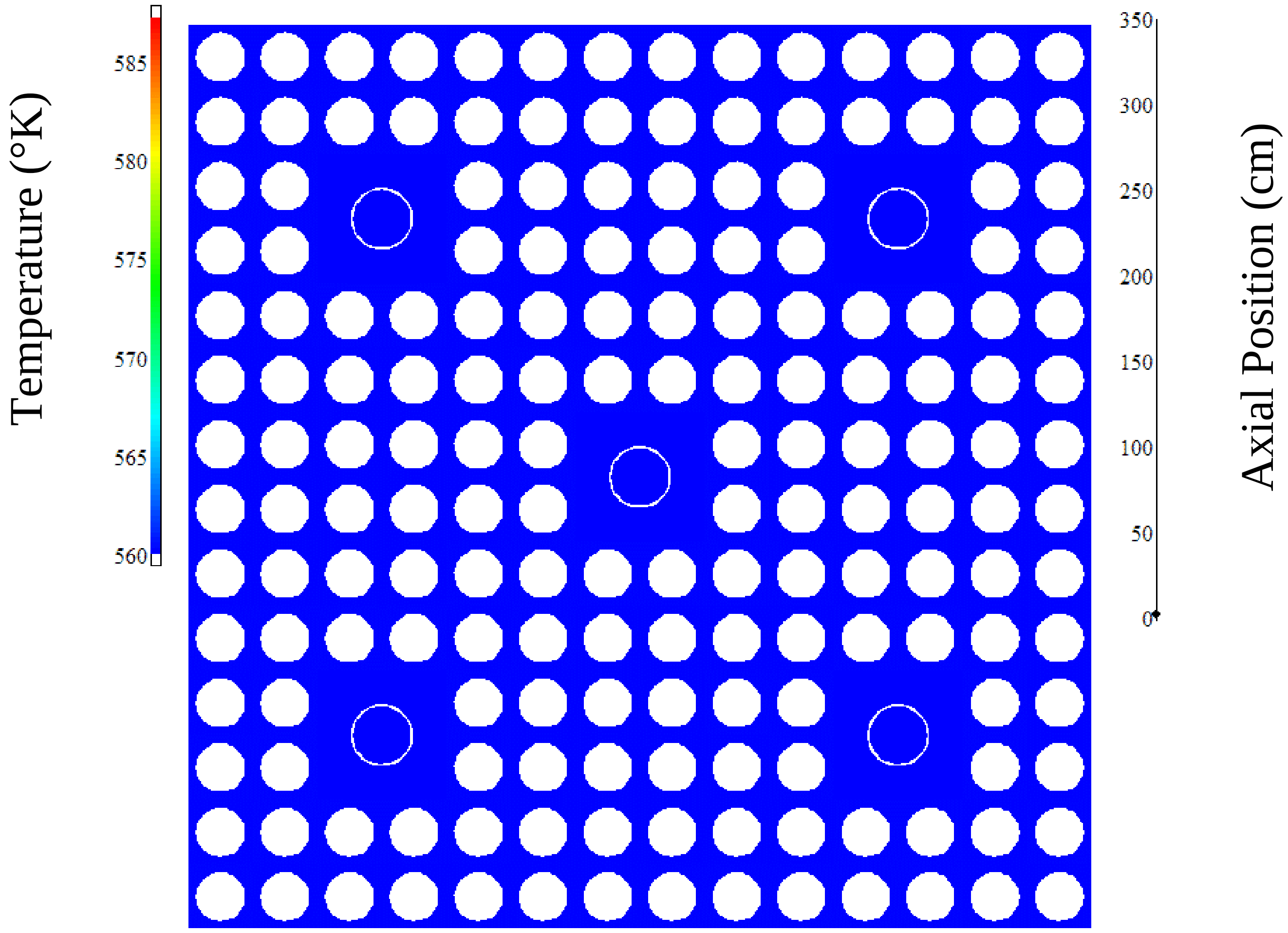
Calvert Cliffs Assembly – Axial Temperature Profile (Fuel & Coolant)



Calvert Cliffs Assembly – Axial Fuel & Coolant Temperature Distributions



Calvert Cliffs Assembly – Axial Coolant Temperature Distributions



Conclusions

- Prototype implementation of an integrated, in-line, thermal feedback capability for Monte Carlo reactor calculations
  - No external code coupling required
  - Uses native Monte Carlo combinatorial geometry; no additional “thermal mesh”
- The Thermal feedback module offers users flexibility in thermal modeling
  - Saves time and memory compared to traditional “full-blown” coupled MC/CFD
  - Maintains a “pay for what you use” philosophy, allowing users to trade accuracy / speed
- Simplified thermal treatments allow for reasonably accurate temperature distributions with minimal computational effort
  - Resulting temperature distributions are sufficient to capture most feedback effects for steady-state nuclear calculations
  - Thermal solution runs in a fraction of the time required for the MC transport simulation
  - Fuel heating calculation provides nuclear designers with rough estimates of fuel region temperatures without requiring a full thermal analysis design iteration
- Single pin and assembly results are in good agreement with expected qualitative behavior and with previously published results

Questions?

This research was performed under appointment to the Naval Nuclear Propulsion Fellowship Program sponsored by Naval Reactors Division of the U.S. Department of Energy.