

Update on the Development and Validation of MERCURY: A Modern, Monte Carlo Particle Transport Code

Richard Procassini, Janine Taylor, Scott McKinley, Gregory Greenman,
Dermott Cullen, Matthew O'Brien, Bret Beck and Christian Hagmann
Lawrence Livermore National Laboratory



ANS Topical Meeting in Mathematics and Computations
12 – 15 September 2005
Avignon, France

University of California



Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94551

Outline



- *Introduction*
- *Recent Algorithm Advances*
 - *3-D Combinatorial Geometry Particle Tracker*
 - *New Mesh-Based Particle Trackers*
 - *Complex Geometry Generation via Templates*
- *New Physics Capabilities*
 - *Improved Thermalization Model*
 - *Continuous Energy Cross Sections*
 - *$S(\alpha, \beta)$ Bound Molecular Scattering*
- *Recent Computer Science Enhancements*
 - *An Extensible, XML-Based Input Parameter Parser*
 - *A Dynamic Load Balancing Capability*
 - *Serial Run Times*
- *Future Directions*

Introduction



- MERCURY [1] is a modern Monte Carlo particle transport code being developed at LLNL. The intent is that MERCURY will eventually replace the older codes TART [2] and COG [3] as the next-generation, general-purpose radiation transport code at LLNL.
- The main physics capabilities of the MERCURY Monte Carlo transport code include:
 - ◆ Time dependent transport of several types of particles through a medium:
 - Neutrons (n)
 - Gammas (γ)
 - Light charged ions ($^1H, ^2H, ^3H, ^3He, ^4He$)
 - ◆ Particle tracking through a wide variety of problem geometries:
 - 1-D spherical (radial) meshes
 - 2-D r - z structured and quadrilateral unstructured meshes
 - 3-D Cartesian structured and tetrahedral unstructured meshes
 - 3-D combinatorial geometry

Introduction



- The main physics capabilities of the MERCURY (*continued*):
 - ◆ Multigroup and continuous energy treatment of cross sections
 - ◆ Population control can be applied to all types of particles
 - ◆ Static k_{eff} and α eigenvalue calculations for neutrons
 - ◆ Dynamic α calculations for all types of particles
 - ◆ All types of particles can interact with the medium via collisions, resulting in:
 - Deposition of energy
 - Depletion and accretion of isotopes resulting from nuclear reactions
 - Deposition of momentum (*to be added*)
- Near term enhancements of MERCURY will include:
 - ◆ Generalization of the current source capabilities
 - ◆ Generalization of the current tally capabilities, and addition of event history support
 - ◆ Post-processing of tallies will be provided by the CALORIS code
 - ◆ Addition of several variance reduction methods

Recent Algorithm Enhancements



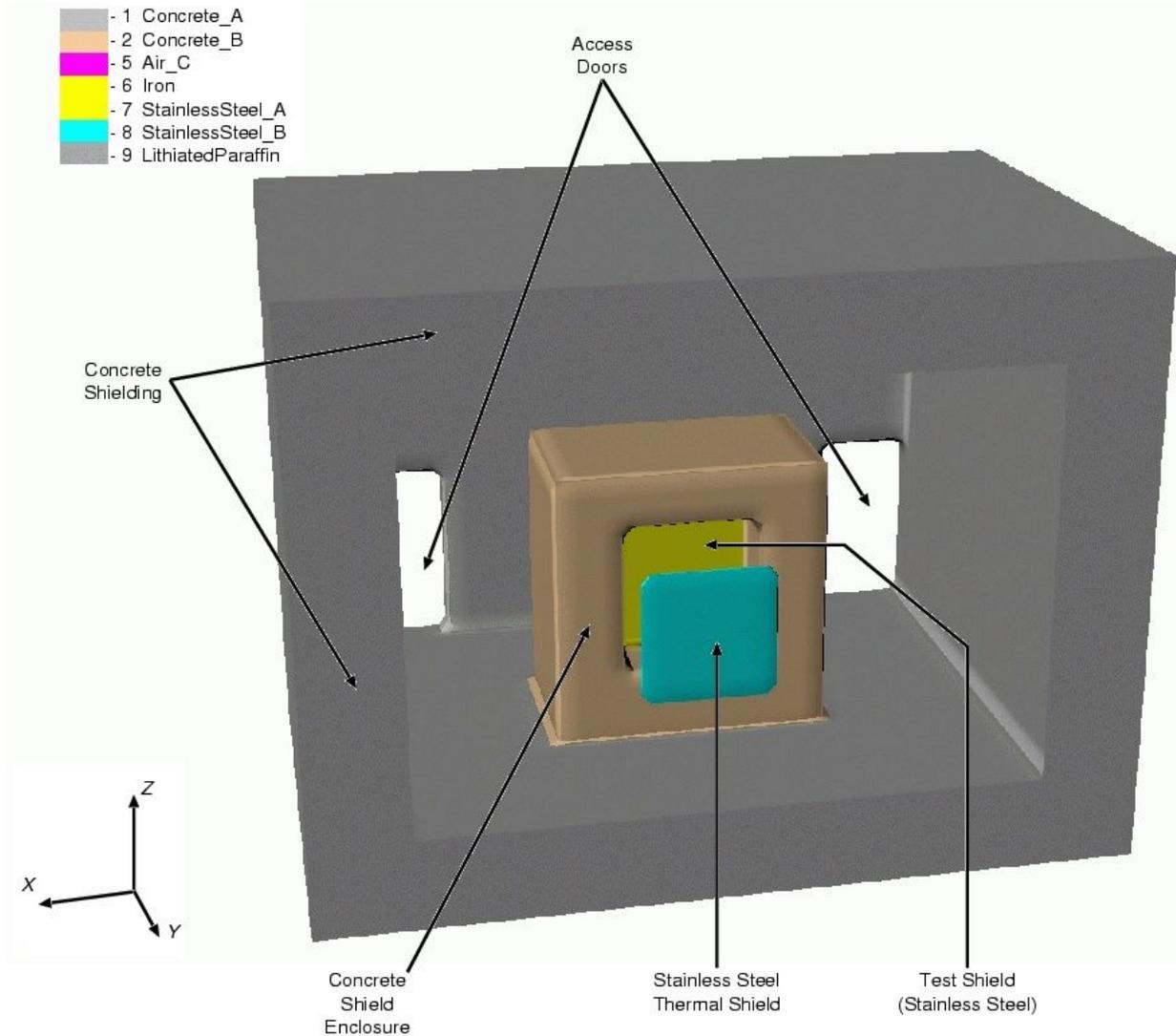
3-D Combinatorial Geometry Particle Tracker

- The new 3-D Combinatorial geometry (CG) currently supports:
 - Cell definitions using the logical aggregation of first-order (planar) and second-order (spherical, elliptical, cylindrical and conical) analytic surfaces
 - Logical aggregation of surfaces via an *implicit* Boolean 'AND' operation
- Future development plans for the CG tracker include:
 - Support for a broad range of logical operations, including Boolean 'OR' and 'NOT'. (Currently implemented, but not fully tested)
 - Support for fourth-order (toroidal) analytic surfaces, cubic spline-based surfaces of revolution, multidimensional spline surfaces and topographic surfaces
- A rudimentary capability to visualize 3-D CGs has also been added to MERCURY. :
 - A “graphics mesh” is superimposed upon the CG in order to produce images.
 - This CG-to-mesh mapping method can be *very* memory and flop intensive
 - A new technique which stores *and* visualizes the problem geometry in a CG-based data model is currently being implemented (See *Future Directions*)



Recent Algorithm Enhancements

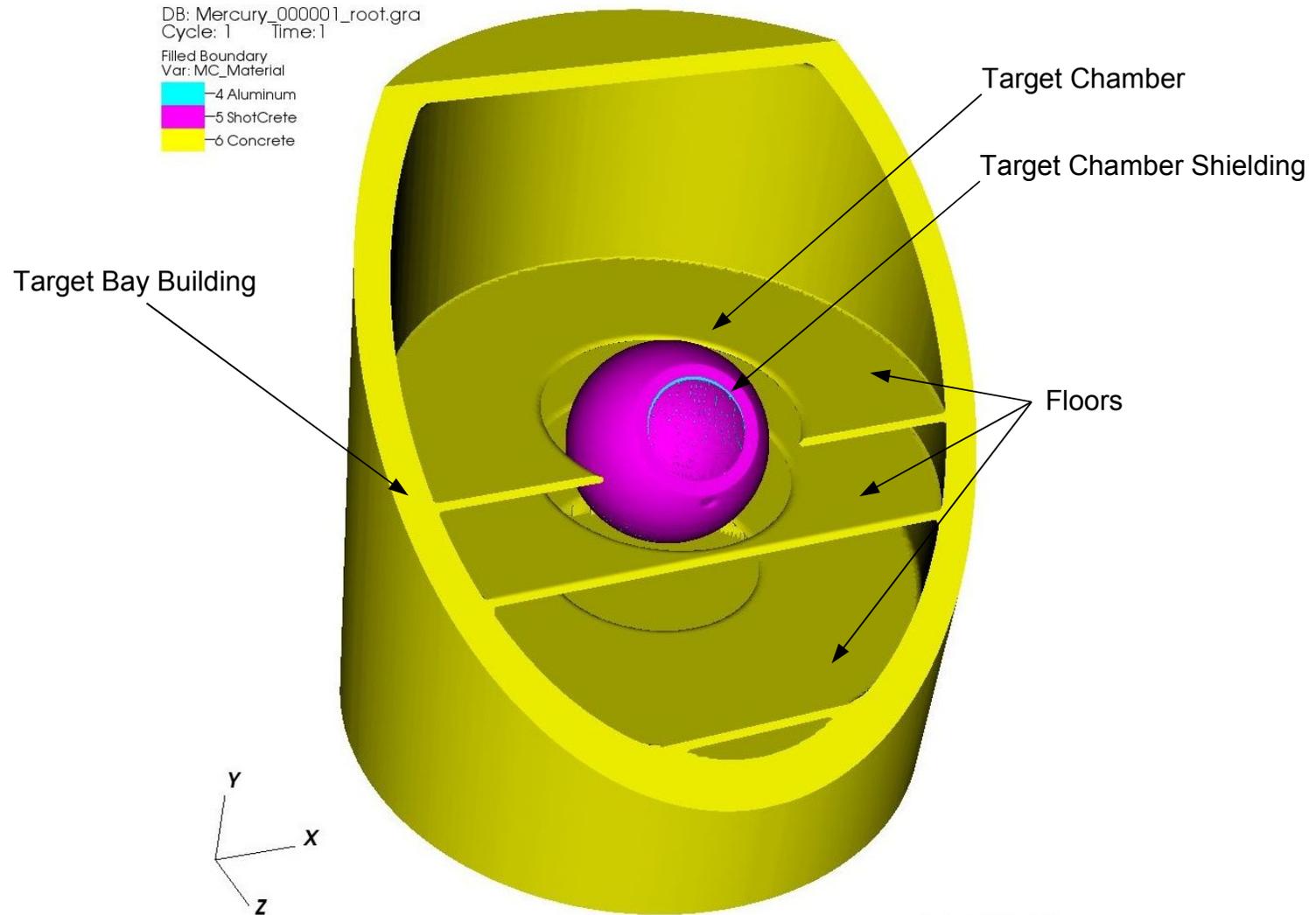
CG Model of a Fusion-Neutron Shielding Test Facility





Recent Algorithm Enhancements

CG Model of the Target Bay of the National Ignition Facility (NIF) Laser Fusion Facility



Recent Algorithm Enhancements



New Mesh-Based Particle Trackers

- Two additional mesh-based particle trackers have been developed:
 - ◆ A 1-D radial mesh tracker for modeling spherical systems:
 - This tracker uses components of the 3-D CG tracker.
 - It is implemented as concentric, nested 3-D spheres using the same second-order spherical surfaces that are used in 3-D CG systems.
 - ◆ A tracker which supports 2-D quadrilateral meshes:
 - The edges of the cells can be aligned at arbitrary angles to either of the (r, z) axes
 - This type of mesh is axisymmetric about the z axis, hence cell edges are actually second-order conical surfaces that may be degenerate in the form of cylinders or planes

Recent Algorithm Enhancements



Complex Geometry Generation via Templates

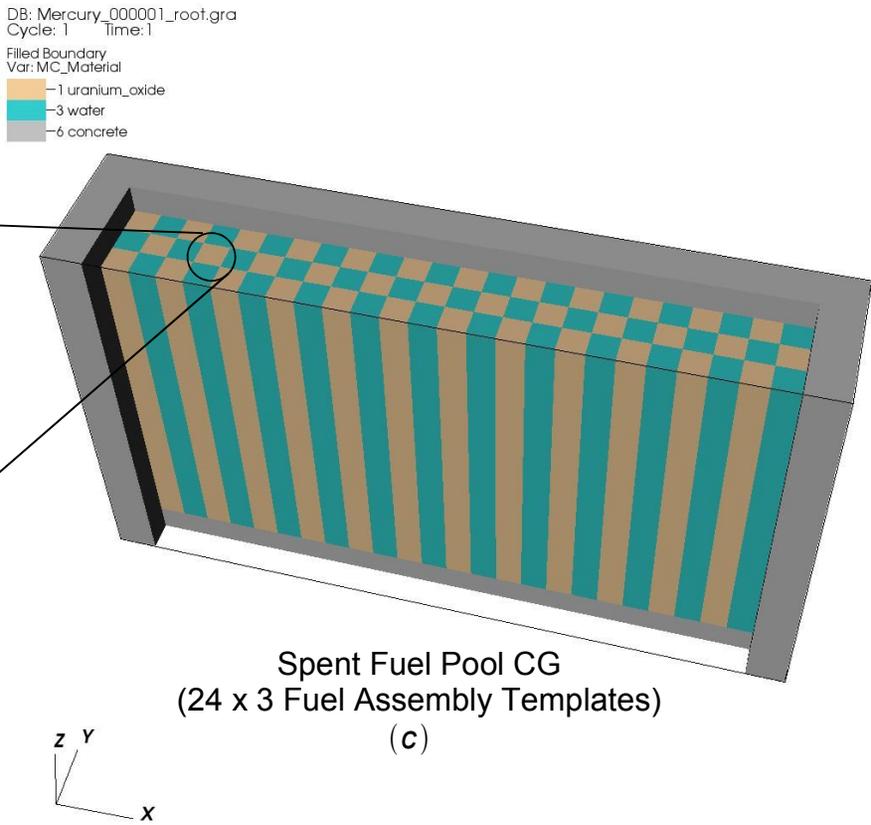
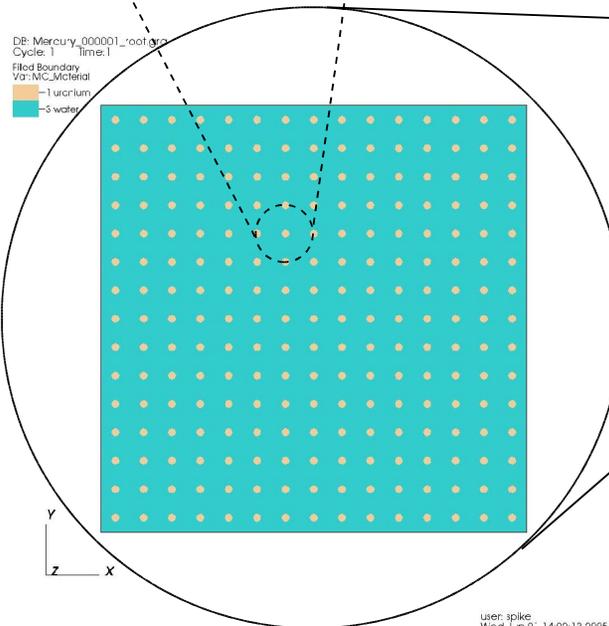
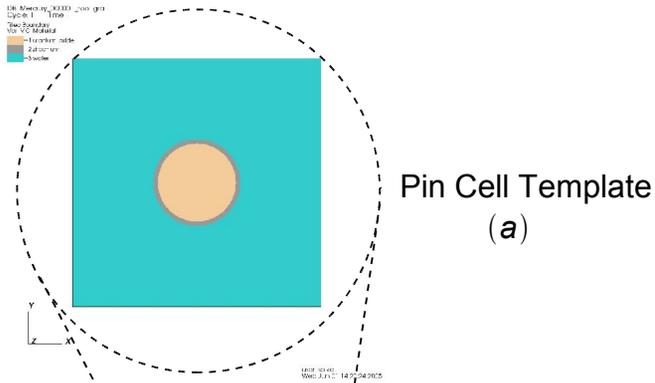
- Templates simplify the generation of complex CGs by providing a straightforward, recursive mechanism for defining hierarchical, repeated structures.
- Simple geometric structures can be defined, and then *referred to* in the definition of other, more-complex geometric structures.
- The intent is to minimize the number of geometric structures that need to be defined in order to create complex systems.
- Once all of the required templates have been defined, the user *instantiates* the actual cells through a small number of creation commands.
- The example shows the multistep process of creating a nuclear-reactor spent fuel pool [4] from simpler components:
 - ◆ The fuel pool is a 24×3 alternating array of reactor fuel and water assemblies
 - ◆ Each reactor assemblies is a 15×15 array of pin cells
 - ◆ A total of 24774 CG cells are generated using 62 templates and 1 create command



Recent Algorithm Enhancements

Template-Based CG Model of a Nuclear Reactor Spent Fuel Pool

Number of Reactor Assemblies:	72
Number of Pin Cells:	8100
Total Number of CG Cells:	24774
Number of Template Definitions:	62
Number of Creation Commands:	1



user: spike
Wed Jun 01 14:40:56 2005

New Physics Capabilities



Improved Thermalization Model

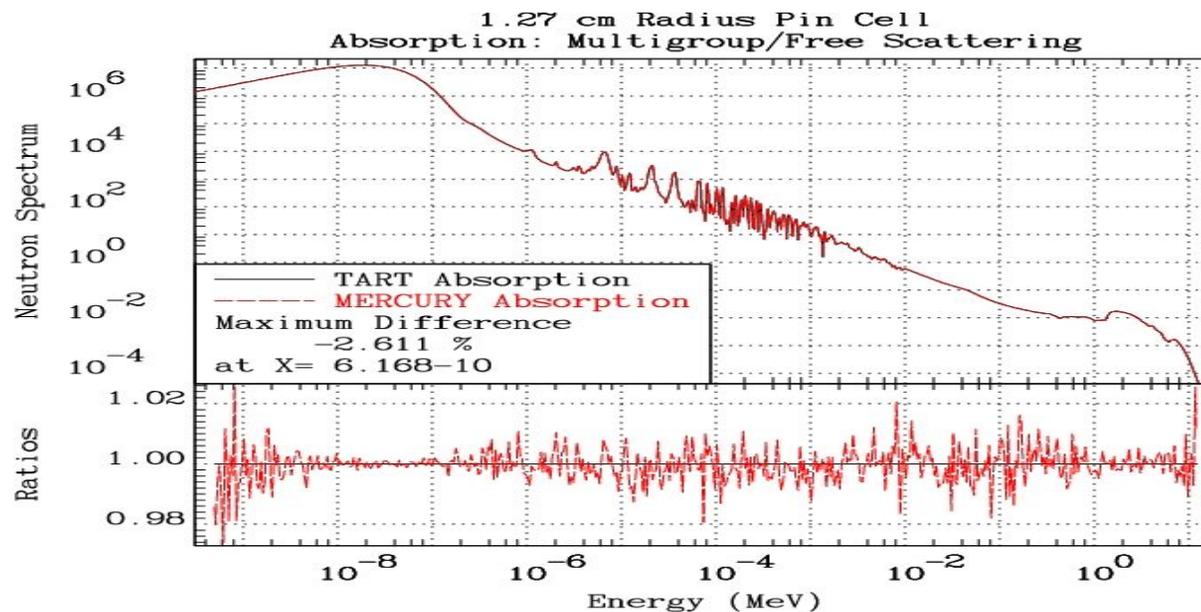
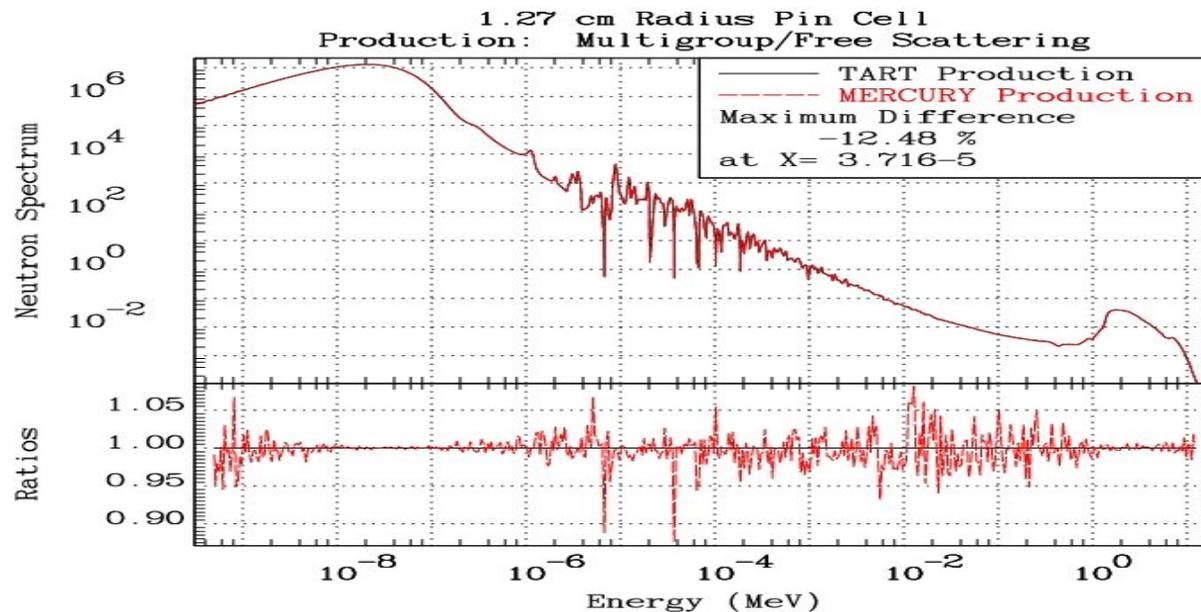
- A neutron free atom thermalization model has been added to the Monte Carlo All Particle Method (MCAPM) collision library and nuclear data server [5] which is used within MERCURY.
- The model assumes that the neutrons are elastically scattering off of a background Maxwellian distribution, with a thermal width of the Maxwellian is defined by the temperature of the medium.
- This is the same treatment that has been previously implemented in TART [6].
- The new elastic scattering model is more realistic than the “energy floor” model which is also available for use in MERCURY.
- The new model allows a particle to either elastically up- or down-scatter due to collisions with material that is not at room temperature.
- The “energy floor” model only allows the particle to “up-scatter” to the thermal energy, and only if the particle's energy was below the thermal energy.

New Physics Capabilities



- The new model is implemented as an extension of the existing collisional kinematics algorithms in MCAPM:
 - If a particle undergoes an elastic scattering event using room-temperature cross sections, MCAPM checks the ratio of the incident particle energy to the temperature of the background medium
 - If this ratio is $E_{inc}/(kT) < 1 \times 10^4$, then the energy of the outgoing neutron is resampled, assuming that the medium is heated to the correct temperature
- To test the new thermalization model, code-to-code comparisons were made with TART for a reactor pin cell criticality calculation [7]:
 - The pin cell has a uranium (^{238}U and ^{235}U , $\rho = 18.8 \text{ g/cm}^3$) pin of radius $r = 1.27 \text{ cm}$ in a square water ($\rho = 1.0 \text{ g/cm}^3$) pitch $\Delta = 5.08 \text{ cm}$ on each side
 - The pin cell is modeled with reflecting boundary conditions in each dimension
 - The heterogeneous nature of this problem makes it ideal to test thermalization of particles in the water moderator region

New Physics Capabilities



1.27 cm Pin Cell Calculations

Multigroup (616 Group) Cross Sections
 Consistent Evaluated Nuclear Data
 Free Atom (1001 Hydrogen) Scattering
 100 Million Particle Histories

MERCURY: $k_{\text{eff}} = 0.96080 \pm 0.00013$
 TART: $k_{\text{eff}} = 0.96066 \pm 0.00013$

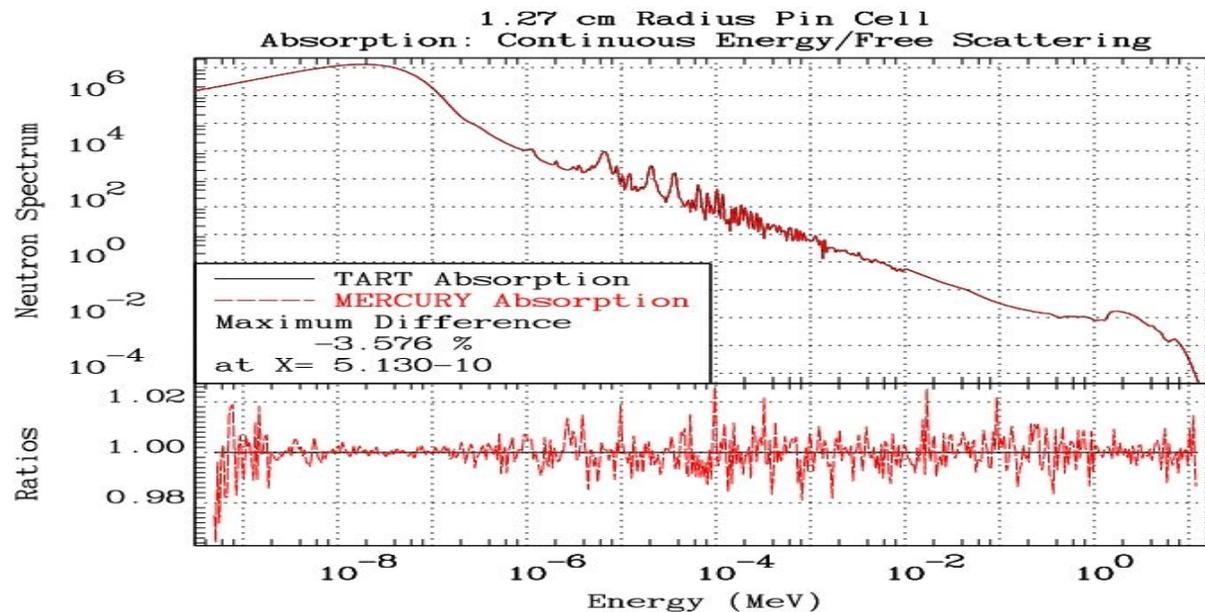
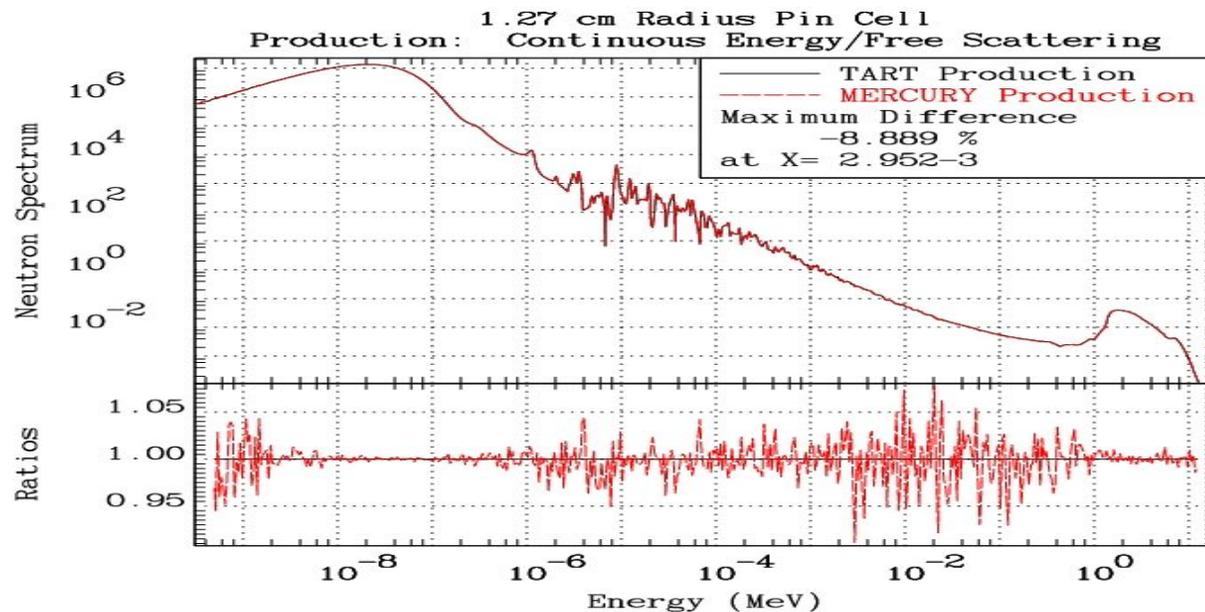
New Physics Capabilities



Continuous Energy Cross Sections

- The MCAPM library was extended to provide a continuous energy treatment of cross sections:
 - The pointwise data has always been available within the MCAPM data files, but access routines were only recently written
 - If the particle energy lies between two of the energy points at which the cross sections are tabulated, the library linearly interpolates the cross section based upon a linear interpolation of the incident particle energy

New Physics Capabilities



1.27 cm Pin Cell Calculations

Continuous-Energy Cross Sections
Consistent Evaluated Nuclear Data
Free Atom (1001 Hydrogen) Scattering
100 Million Particle Histories

MERCURY: $k_{\text{eff}} = 0.99674 \pm 0.00013$

TART: $k_{\text{eff}} = 0.99670 \pm 0.00013$

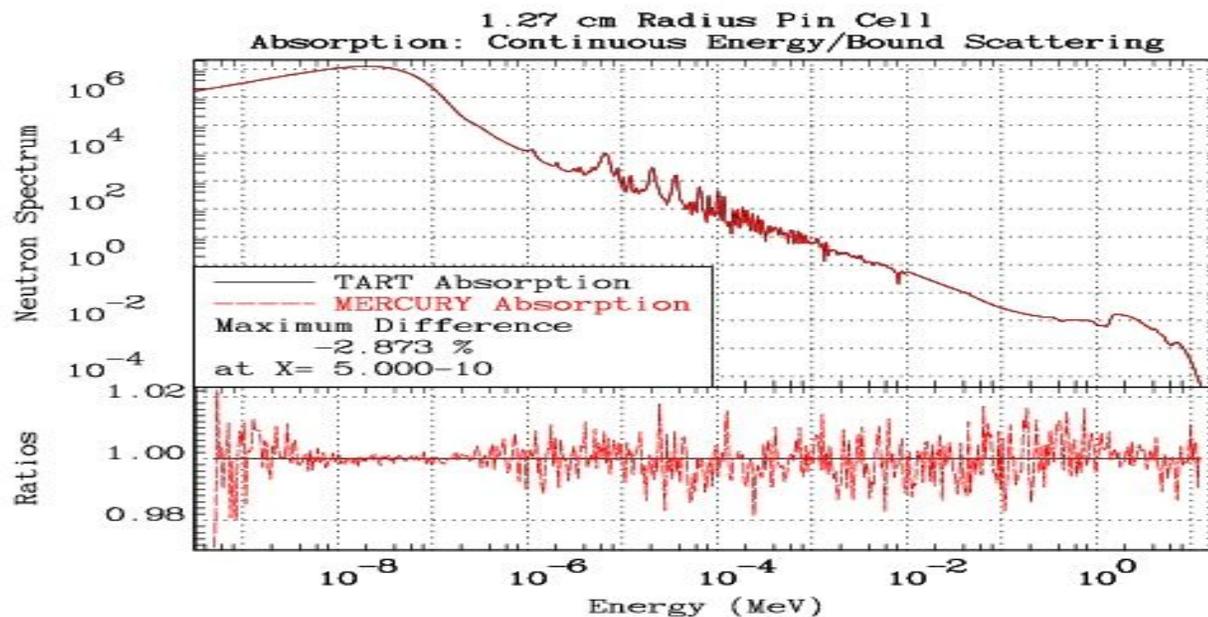
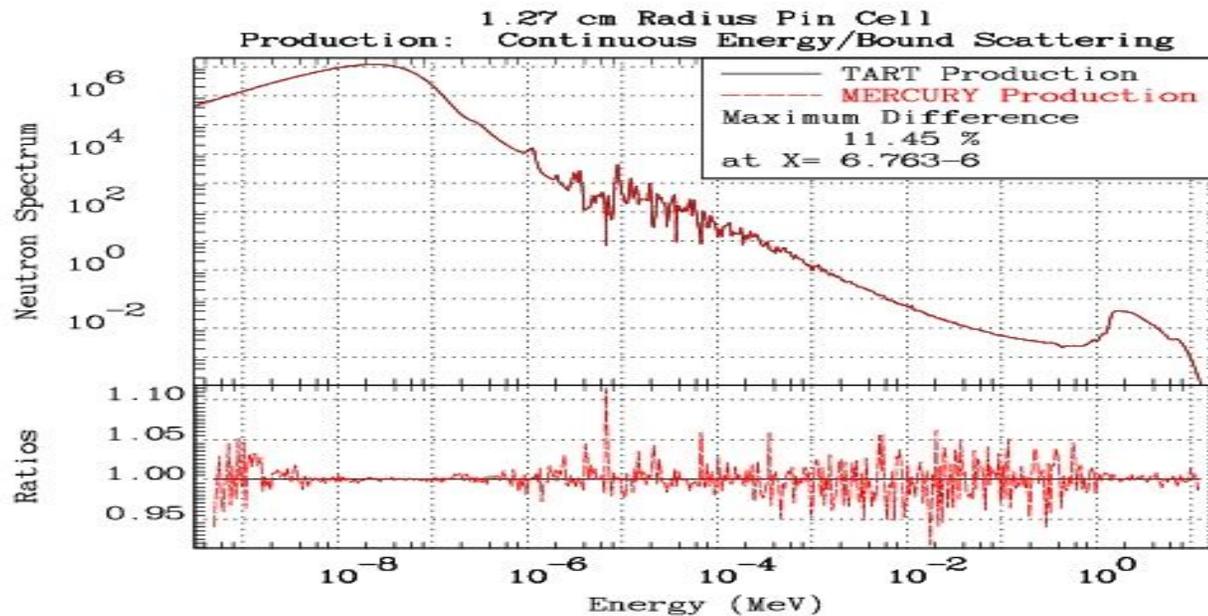
New Physics Capabilities



$S(\alpha, \beta)$ Bound Molecular Scattering

- The data describing scattering off of bound molecular systems is described in MCAPM with outgoing particle spectra that are energy-angle correlated.
- Since MCAPM already handled such secondary particle correlations, it was easy to include additional, fictitious isotopes that represented the following molecules into the MCAPM data files:
H in H₂O , H in CH₂ , D in D₂O , Be (Metallic) , Be in BeO , C (Graphite) and O in BeO
- These isotopes are otherwise identical to their free-atom scattering counterparts, except that below a threshold which is unique for each isotope (for example, $E \leq 4$ eV for H in H₂O), the elastic-scattering cross section goes to zero and is replaced by an inelastic-scattering cross section that represents scattering off of bound molecules.
- This is the same treatment that has been previously implemented in TART [8].

New Physics Capabilities



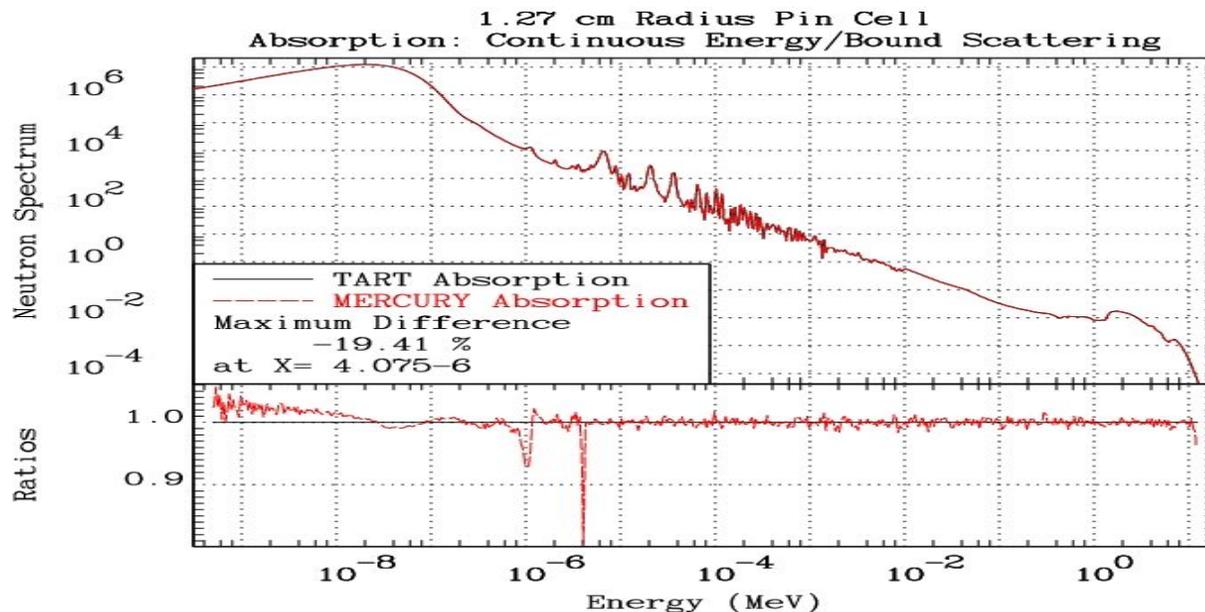
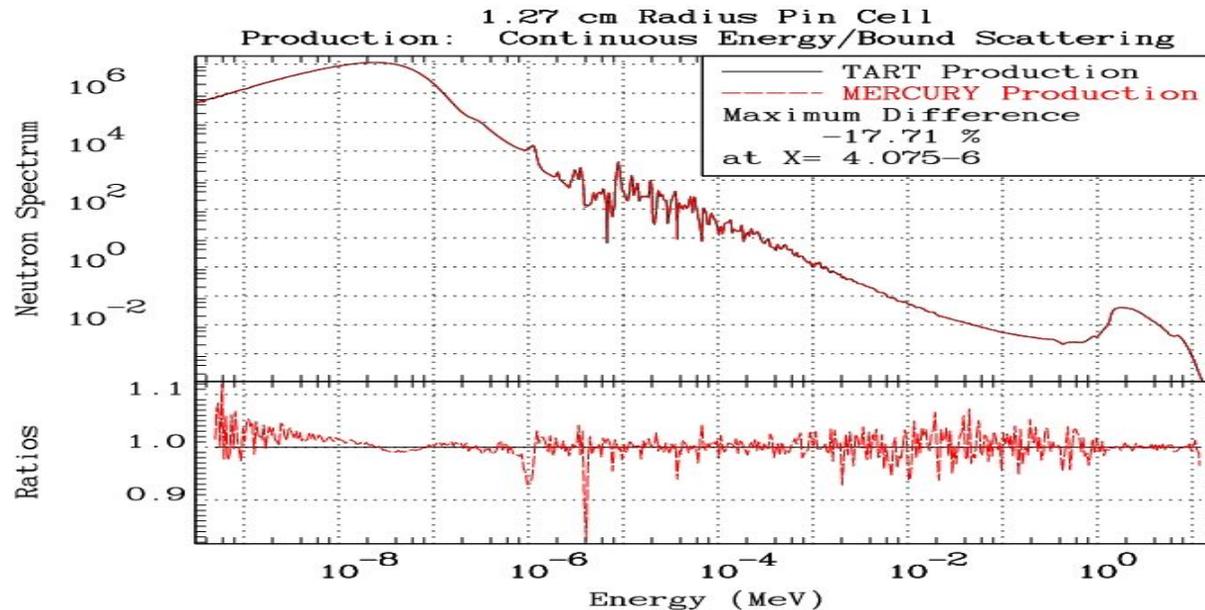
New Results

1.27 cm Pin Cell Calculations

Continuous-Energy Cross Sections
Consistent Evaluated Nuclear Data
Bound Atom (1801 Hydrogen) Scattering
100 Million Particle Histories

MERCURY: $k_{\text{eff}} = 0.94727 \pm 0.00012$
TART: $k_{\text{eff}} = 0.94717 \pm 0.00012$

New Physics Capabilities



Old Results

1.27 cm Pin Cell Calculations

Continuous-Energy Cross Sections
 Consistent Evaluated Nuclear Data
 Bound Atom (1801 Hydrogen) Scattering
 100 Million Particle Histories

MERCURY: $k_{\text{eff}} = 0.94688 \pm 0.00012$

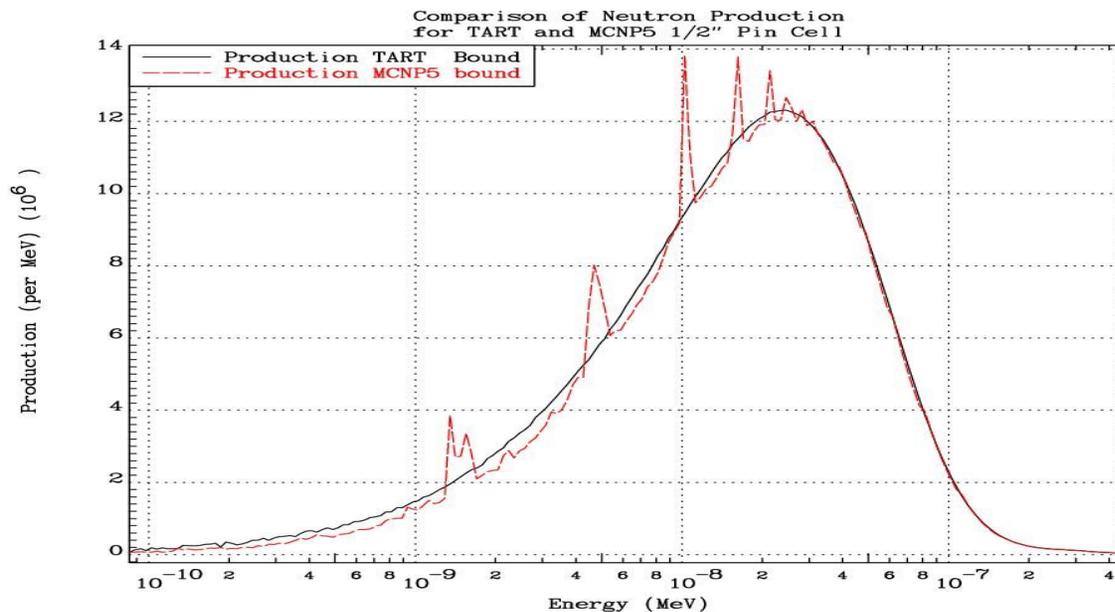
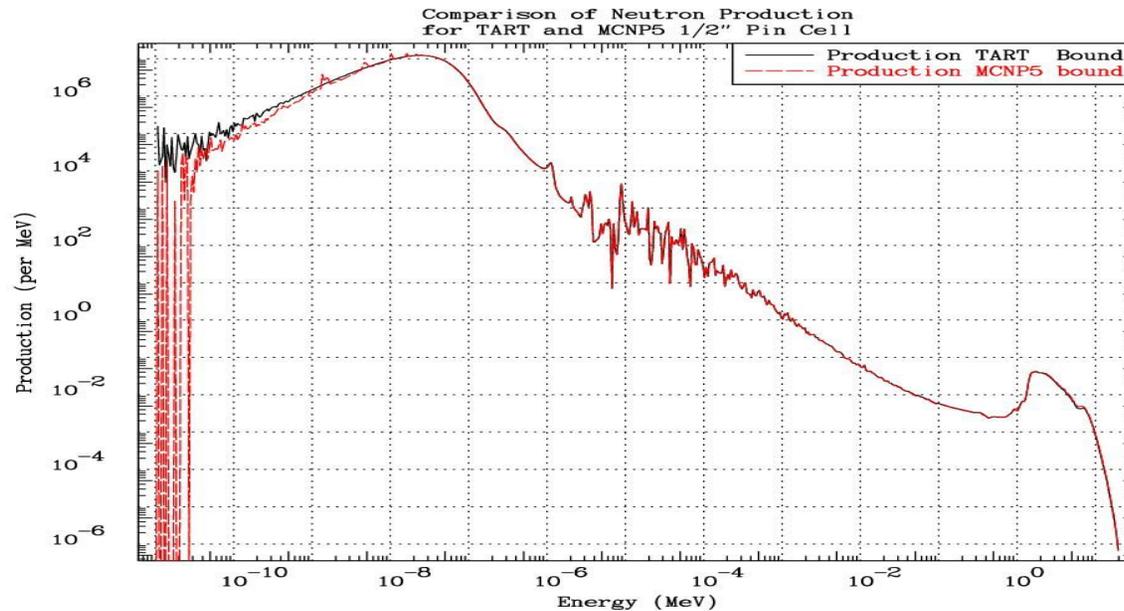
TART: $k_{\text{eff}} = 0.94730 \pm 0.00012$

Bugs were found, and corrected,
 in the data processing code for
 MCAPM (MERCURY) and the
 angle-energy correlated
 sampling routine in TART!

This is a good example of
 cross-code validation!



New Physics Capabilities



TART / MCNP Results

1.27 cm Pin Cell Calculations

Continuous-Energy Cross Sections
Consistent Evaluated Nuclear Data
Bound Atom (1801 Hydrogen) Scattering
100 Million Particle Histories

TART: $k_{\text{eff}} = 0.95946 \pm 0.00005$
MCNP: $k_{\text{eff}} = 0.96067 \pm 0.00004$
 $\text{DELTA}(k_{\text{eff}}) = 0.00121$

Notice the relatively good agreement in the integral quantity, k_{eff} , but the poor agreement in the production spectra! This clearly indicates that comparison of k_{eff} is not sufficient to determine if the code is accurately calculating transport in multiplying systems!

New Physics Capabilities



Bringing It All Together... Free and Bound Scattering Versions of Six Reactor Pin Cells
(Continuous Energy Cross Sections, Free Elastic or Bound “Inelastic” Thermal Scattering)

Code	User	1.27 cm / Free	1.27 cm / Bound	0.635 cm / Free	0.635 cm / Bound	0.3175 cm / Free	0.3175 cm / Bound
COG	M. Lee	1.0112(12)	0.9636(12)	1.0111(12)	0.9159(12)	1.0109(12)	0.9019(12)
COG	Heinrichs	1.0115(12)	0.9608(12)	1.0121(12)	0.9148(12)	1.0133(12)	0.9065(12)
KENO	Heinrichs	1.0092(38)	0.9628(5)	1.0133(7)	0.9163(7)	1.0133(8)	0.9050(7)
MCNP5	MacFarlane	1.01283(12)	0.96062(13)	1.01078(16)	0.91221(18)	1.01206(22)	0.89867(23)
MCNP5	M. Lee	1.01236(38)	0.96073(42)	1.01015(52)	0.91115(55)	1.01200(68)	0.89911(70)
MCNP5	Cullen	1.01294(40)	0.96059(42)	1.01133(56)	0.91204(58)	1.01299(74)	0.89932(73)
MCNP5	Cullen	1.01298(13)	0.96046(14)	1.01089(17)	0.91208(18)	1.01187(23)	0.89854(23)
MCNP5	Cullen	1.01279(4)	0.96067(4)	1.01094(6)	0.91212(6)	1.01187(7)	0.89882(8)
MCNP4C	Heinrichs	1.0101(05)	0.9597(06)	1.0105(08)	0.9133(08)	1.0103(09)	0.9027(11)
MCNP4B	Trkov	1.01071(6)	0.96061(6)	1.00932(9)	0.91380(9)	1.01038(11)	0.90201(11)
MCNPX21	Trkov	1.01075(6)	0.96071(7)	1.00927(8)	0.91402(9)	1.01023(11)	0.90199(11)
MCNPX24	Trkov	1.01292(6)	0.96211(7)	1.01100(9)	0.91496(9)	1.01178(11)	0.90264(11)
MCNPX24	Trkov	1.01292(6)	0.96055(6)	1.01100(9)	0.91207(9)	1.01178(11)	0.89874(12)
MCNPX24	Trkov	1.01292(6)	0.96044(6)	1.01100(9)	0.91167(9)	1.01178(11)	0.89850(11)
MCU	Kalugin	1.01680(30)	0.96378(30)	1.01632(40)	0.91404(40)	1.01556(40)	0.89842(30)
MONK8B	Dean	1.01300(5)	0.95990(5)	1.01130(5)	0.91140(5)	1.0132(2)	0.8991(1)
MONK9	Dean	1.0136(3)	0.9593(3)	1.0117(2)	0.9101(2)	1.0121(2)	0.8969(1)
MVP	Nagaya	1.01299(8)	0.96065(9)	1.01169(12)	0.91309(15)	1.01282(17)	0.90016(19)
TART04	Cullen	1.00916(50)	0.96114(50)	1.00983(50)	0.91548(50)	1.01072(50)	0.90333(50)
TART04	Cullen	1.00952(5)	0.96071(5)	1.00933(5)	0.91544(5)	1.01091(5)	0.90293(5)
TART04	M. Lee	1.0101(4)	0.9598(5)	1.0098(7)	0.9138(8)	1.0126(10)	0.9007(11)
TART04	M. Lee	1.00960(4)	0.96073(4)	1.00926(7)	0.91537(8)	1.01097(10)	0.90312(11)
TRIPOLI4	Y. Lee	1.01295(11)	0.96046(10)	1.01243(10)	0.91197(10)	1.01448(11)	0.89766(10)
VIM	Heinrichs	1.0157(8)	0.9609(8)	1.0150(09)	0.9131(09)	1.0168(11)	0.9020(11)
VIM	Blomquist	1.00525(7)	0.95939(7)	1.00680(9)	0.90877(9)	1.03463(10)	0.89629(10)
TART05 (best)	Cullen	1.01039(5)	0.95946(5)	1.01083(5)	0.91448(5)	1.01268(5)	0.90324(5)
TART05 (same)	Cullen	1.01081(5)	0.95984(5)	1.01101(5)	0.91478(5)	1.01269(5)	0.90349(5)
MERCURY (same)	Procassini	1.01100(6)	0.96004(6)	1.01097(6)	0.91483(7)	1.01288(6)	0.90351(7)
Mean		1.01168	0.96073	1.01103	0.91332	1.01313	0.90090
Standard Deviation		0.00223	0.00112	0.00180	0.00185	0.00448	0.00259
Minimum		1.00525	0.95930	1.00680	0.90877	1.01023	0.89629
Maximum		1.01680	0.96378	1.01632	0.91630	1.03463	0.90650
Range		0.01155	0.00448	0.00952	0.00753	0.02440	0.01021

Recent Computer Science Enhancements



An Extensible, XML-Based Input Parameter Parser

- MERCURY's previous input parameter parser was not easily extended to handle the complexity of defining 3-D combinatorial geometries.
- A new parser was developed which uses the XML data-description language and the CYCLOPS system [9].
- At its core, CYCLOPS is an XML-based data model that is closely coupled to:
 - ◆ A text-to-XML input translator
 - ◆ A graphical user interface (GUI)
 - ◆ A set of data-tree query and access routines which use the XERCES parser library [10] for manipulating document object model (DOM) data trees
- Use of the CYCLOPS/XERCES parsing system has greatly simplified the addition of new input-parameter blocks into MERCURY.

Recent Computer Science Enhancements

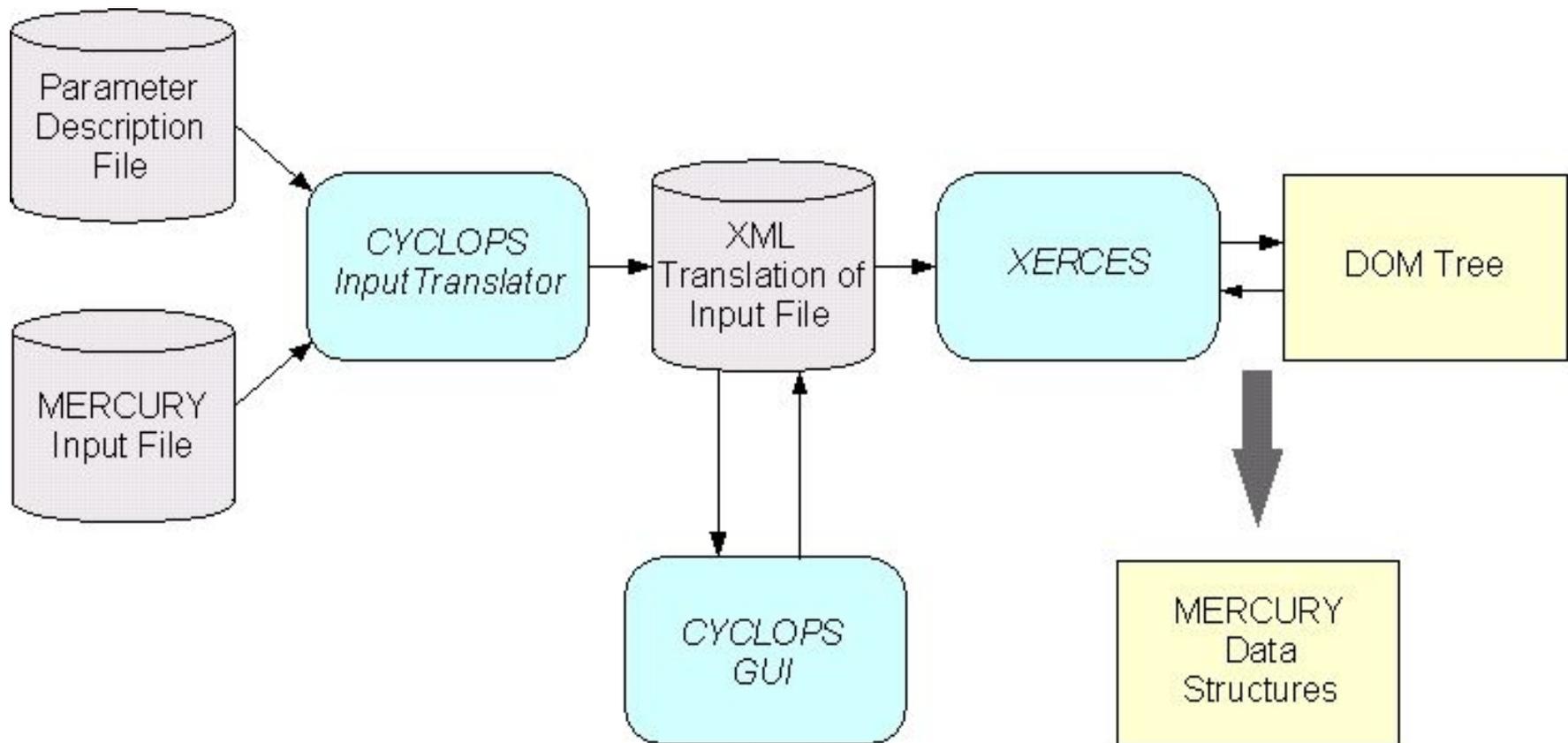


- This new approach allows one to add a new section to a data-model description file for each new block of input parameters.
- All of the file handling features, data querying and access capabilities of CYCLOPS/XERCES are modular and used for each of the data blocks.
- Additional benefits of using CYCLOPS/XERCES include:
 - ◆ A rudimentary level of validity checking of the input parameters
 - ◆ The CYCLOPS GUI, which allows users to interactively generate new, or modify existing, MERCURY input files
- The CYCLOPS GUI employs input-block-based windows, context sensitive help and tooltips which can simplify the process of building input files, and can be instructive for a new code user.

Recent Computer Science Enhancements



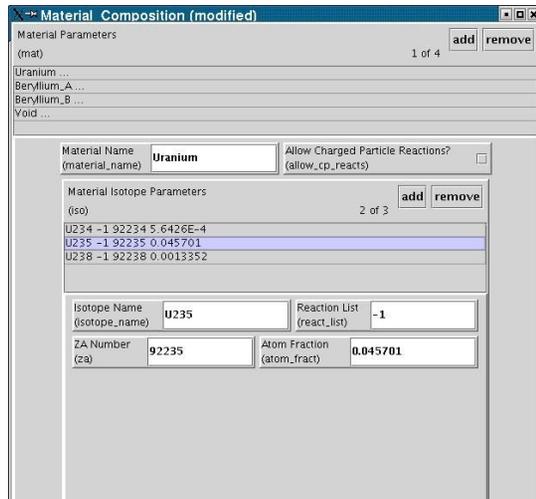
Flow Chart of the New XML-Based Input Parameter Parser in MERCURY



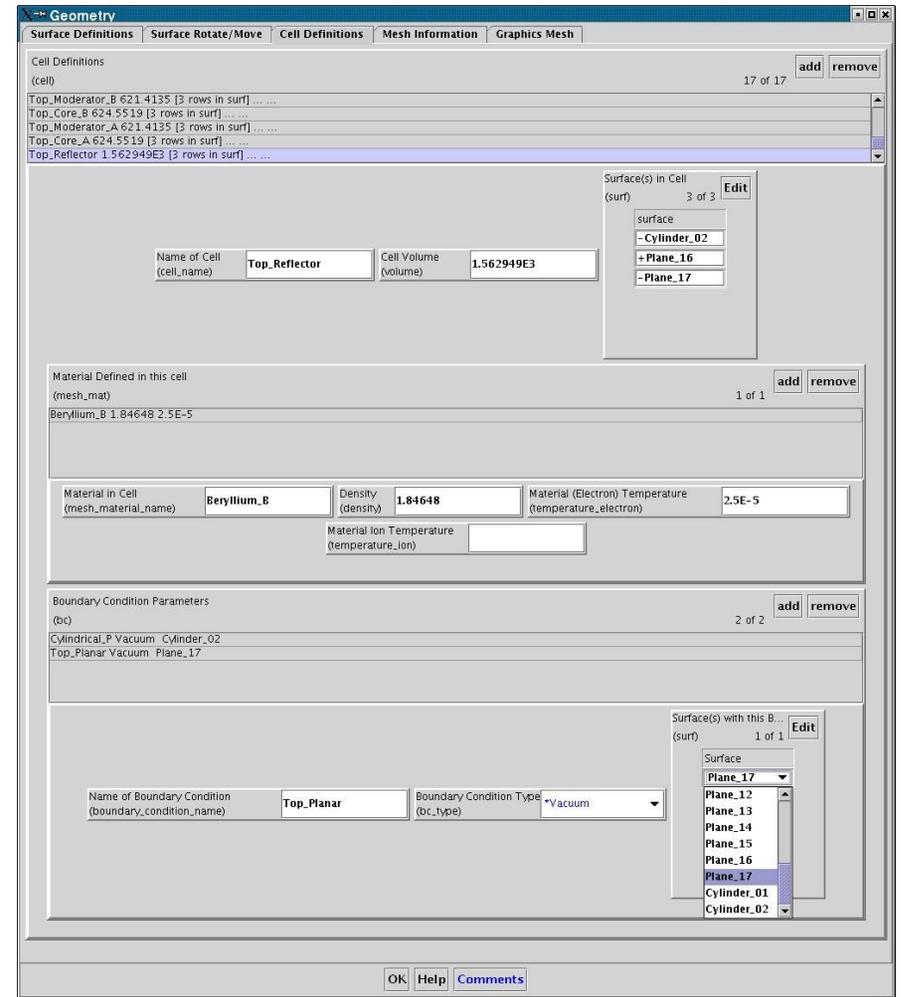


Recent Computer Science Enhancements

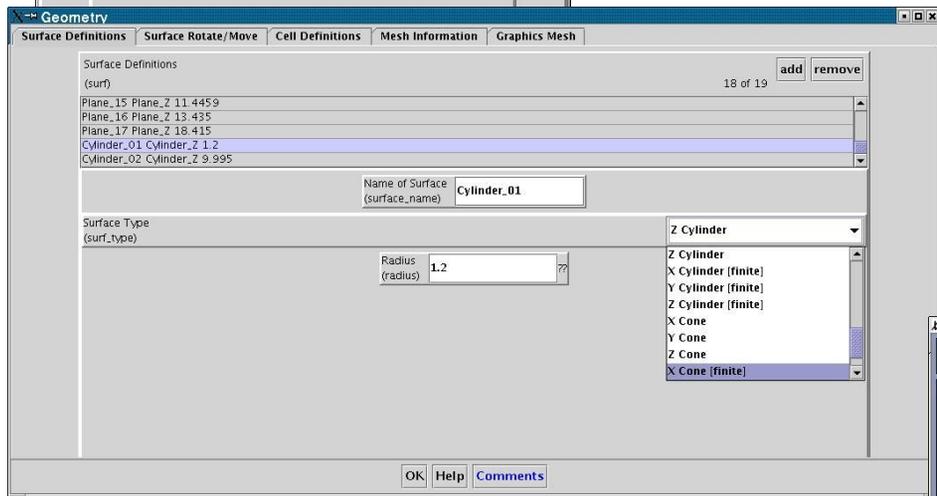
Three MERCURY Input Data-Block Windows from the CYCLOPS GUI



Material Composition Data Block Window (a)



Geometry (Surface Definition) Data Block Window (b)



Geometry (Cell Definition) Data Block Window (b)

Recent Computer Science Enhancements



A Dynamic Load Balancing Capability

- A major design requirements for MERCURY is the ability to run efficiently on a wide variety of parallel computing platforms.
- A multi-pronged approach to parallelism has been developed and implemented in MERCURY, including:
 - ◆ Spatial decomposition of the problem geometry into domains, and the assignment individual processors to work on specific domains - *Domain Decomposition* (spatial parallelism)
 - ◆ Storing the geometry information redundantly on each of the processors, and assigning each processor work on a different set of particles - *Domain Replication* (particle parallelism)
 - ◆ *A combination of both spatial and particle parallelism can be employed to achieve a scalable parallel calculations*

Recent Computer Science Enhancements



- Since particles migrate in space and time between different regions of a problem, it is a natural consequence of domain decomposition that not all spatial domains will require the same amount of computational work: Hence, the calculation becomes load imbalanced.
- In many applications, one portion of the calculation (cycle, iteration, etc.) must be completed by all processors before the next phase can commence.
- If one processor has more work than any of the other processors, the less-loaded processors must wait for the most worked processor to complete its work.
- A method has been developed to reduce this form of particle-induced load imbalance, which allows the number of processors assigned to a domain to vary dynamically in accordance with the amount of work on that domain [11].
- The particles that are located in a given spatial domain are then divided evenly among the number of processors assigned to work on that domain: the domain's replication level.

Recent Computer Science Enhancements



- The performance of parallel Monte Carlo transport calculations which use both spatial and particle parallelism is increased by dynamically assigning processors to the most worked domains.
- Since the particle work load varies over the course of the simulation, this algorithm determines each cycle if dynamic load balancing would speed up the calculation.
- If load balancing is required, a small number of particle communications are initiated in order to achieve load balance.
- This method has demonstrated a decrease in the parallel run time by more than a factor of two for certain criticality and source calculations [12].

For additional information on the dynamic load balancing algorithm, see our presentation Tuesday afternoon in the session on "Computational Methods on Advanced Computers".

Recent Computer Science Enhancements



Serial Run Times

- The serial run times are compared for a series of problems that were run using MERCURY (Version b.11), TART (Version 2005), COG (Version 10) and MCNP (Version 5) [13].
- Two criticality problems and one sourced problem were run:
 - ◆ *HEU-MET-FAST-002-001*: A fast, metallic critical assembly comprised of alternating stacked cylindrical disks of HEU fuel and beryllium moderator/reflector [14]
 - ◆ *HEU-MET-FAST-017-001*: A fast, metallic critical assembly comprised of a sphere of HEU fuel surrounded by a spherical shell of depleted uranium [14]
 - ◆ *Hydrogen Sphere*: An $r = 30$ cm sphere of $\rho = 0.1111$ g/cm³ hydrogen with an $E = 14.1$ MeV isotropic, point source at the center
- All calculations were run:
 - ◆ On one 2.4 Ghz Pentium-4 Xeon processor in an Intel Linux cluster (MCR) at LLNL
 - ◆ With *at least* 10^7 particle histories (The criticality calculations included 25 inactive and 1000 active generations, each with 10^4 particles.)
 - ◆ With a continuous energy treatment of the cross sections

Recent Computer Science Enhancements



Serial Run Times from MERCURY, TART, COG and MCNP for Three Problems

(2.4 Ghz Pentium-4 Xeon Processor / Red-Hat CHAOS-2 Linux OS)

Problem	HEU-MET-FAST-002-001 <i>Critical Assembly [14]</i>	HEU-MET-FAST-017-001 <i>Critical Assembly [14]</i>	Hydrogen Sphere <i>Sourced System</i>
MERCURY Run Time [sec]	3388.34	1162.70	4115.67
TART Run Time [sec]	358.33	771.96	556.15
COG Run Time [sec]	3532.02	1409.91	4388.17
MCNP Run Time [sec]	3649.20	1080.60	3318.60
TART / MERCURY Run Time Ratio	0.1058	0.6639	0.1351
COG / MERCURY Run Time Ratio	1.0424	1.2126	1.0662
MCNP / MERCURY Run Time Ratio	1.0770	0.9294	0.8063

- The serial performance of MERCURY is *comparable* to that of COG and MCNP, with run times varying by no more than ~21%.
- TART *significantly* outperforms MERCURY *only* for problems with small zone counts.
- An in-depth analysis of MERCURY's serial performance has *not* yet been undertaken.
- The advanced parallel capabilities of MERCURY permit it to solve other classes of problems which are not amenable to solution by TART, COG or MCNP.



Future Directions

- Planned physics and algorithm enhancements include:
 - ♦ Energy deposition and isotopic burnup capabilities (*recently completed*)
 - ♦ Extension of the current tally and source capabilities to allow the user to tally into, or sample from, an n -dimensional distribution, where the individual dimensions can be time, energy, angle(s), a 2-D or 3-D Cartesian mesh, CG surfaces, CG cells, etc
 - ♦ Use of the core tally module in a post-processing tool named CALORIS, which can:
 - Tally particles that are written to disk during a prior MERCURY calculation
 - Filter particles according to a set of criteria in order to develop a source for a subsequent MERCURY calculation
 - ♦ Addition of several variance reduction algorithms including geometry-based population control, weight windows, detector biasing, collisional survival biasing and an exponential transform.
 - ♦ Extension of the CG cell definition syntax to include other Boolean operators besides `AND`

Future Directions



- Planned physics and algorithm enhancements (*continued*):
 - The ability to embed meshes within combinatorial geometries
 - Particle trackers for 2-D $r - z$ cylindrical and 3-D Cartesian adaptive mesh refinement (AMR) meshes
 - A non-adjoint method for calculating the probability of a sustained chain reaction
- Planned computer science enhancements include:
 - A new CG data model that eliminates the need to superimpose a “graphics mesh” over a CG in order to visualize the geometry (*currently being implemented*)
 - The VISIT visualization tool [15] will be linked into MERCURY, providing:
 - A rapid means of geometry validation
 - Graphical display of the calculation results *in real time*.
 - Porting to run on Windows XP (*longer term*) and Macintosh OS.X (*shorter term*) systems

Future Directions



- Planned enhancements of the MCAPM nuclear data and collision library include:
 - ◆ An unresolved-resonance-region treatment
 - ◆ A multiband statistical method for modeling cross section resonances
 - ◆ A method for modeling delayed neutrons



References

- [1] R. J. Procassini and J. M. Taylor, "MERCURY User Guide (Version b.8)", Lawrence Livermore National Laboratory, Report UCRL-TM-204296 / Revision 1 (2005).
- [2] D. E. Cullen, "TART 2002: A Coupled, Neutron-Photon 3-D, Combinatorial Geometry Time Dependent Monte Carlo Transport Code", Lawrence Livermore National Laboratory, Report UCRL-ID-126455 / Revision 4 (2002).
- [3] R. Buck, E. Lent, T. Wilcox and S. Hadjimarkos, "COG User's Manual: A Multiparticle Monte Carlo Transport Code (Fifth Edition)", Lawrence Livermore National Laboratory, Internal Report (2002).
- [4] N. R. Smith, et al., *OECD/NEA Source Convergence Benchmark 1: Checkerboard Storage of Assemblies*, Nuclear Energy Agency, <http://www.nea.fr/html/science/wpncs/convergence/specifications/b1-checkerboard.pdf> (2001).
- [5] P. S. Brantley, C. A Hagmann and J. A. Rathkopf, "MCAPM-C Generator and Collision Routine Documentation (Revision 1.2)", Lawrence Livermore National Laboratory, Report UCRL-MA-141957 (2003).
- [6] D. E. Cullen, "THERMAL: A Routine Designed to Calculate Neutron Thermal Scattering (Revision 1)", Lawrence Livermore National Laboratory, Report UCRL-ID-120560 (1995).
- [7] D. E. Cullen, R. N. Blomquist, C. Dean, D. P. Heinrichs, M. A. Kalugin, M. Lee, Y-K Lee, R. MacFarlane, Y. Nagaya and A. Trkov, "How Accurately Can We Calculate Thermal Systems?", Lawrence Livermore National Laboratory, Report UCRL-TR-203892 (2004).
- [8] D. E. Cullen, L. F. Hansen. E. M. Lent and E. F. Plechaty, "Thermal Law Scattering Data: Implementation and Testing Using the Monte Carlo Neutron Transport Codes COG, MCNP and TART", Lawrence Livermore National Laboratory, Report UCRL-ID-153656 (2003).
- [9] N. H. Samuelson, "Cyclops User's Manual (Version 2.0)", Lawrence Livermore National Laboratory, Internal Report (2003).
- [10] The XERCES Team, *XERCES C++ Parser*, The Apache Software Foundation, <http://xml.apache.org/xerces-c/> (2004).
- [11] M. J. O'Brien, J. M. Taylor and R. J. Procassini, "Dynamic Load Balancing of Parallel Monte Carlo Transport Calculations", *Monte Carlo 2005: Topical Meeting in Monte Carlo*, Chattanooga, TN, American Nuclear Society (2005).
- [12] R. J. Procassini, M. J. O'Brien and J. M. Taylor, "Load Balancing of Parallel Monte Carlo Transport Calculations", *2005 Topical Meeting in Mathematics & Computations*, Avignon, France, American Nuclear Society (2005).
- [13] The X-5 Monte Carlo Team, "MCNP - A General, Monte Carlo N-Particle Transport Code (Version 5)", Los Alamos National Laboratory, Report LA-UR-03-1987 (2003).
- [14] J. B. Briggs, et al., "International Handbook of Evaluated Criticality Safety Benchmark Experiments (September 2003 Edition)", Nuclear Energy Agency, Report NEA/NSC/DOC(95)03 (2003).
- [15] The VISIT Team, *VISIT: Software that Delivers Parallel Interactive Visualization*, Lawrence Livermore National Laboratory, <http://www.llnl.gov/visit/> (2003).



For additional information, please visit our web site:

www.llnl.gov/mercury

Acknowledgments

This work was performed under the auspices of the
U. S. Department of Energy by the University of California
Lawrence Livermore National Laboratory
under Contract W-7405-Eng-48.