

LA-UR-16-20076

Approved for public release; distribution is unlimited.

Title: An Overview of the Monte Carlo Application ToolKit (MCATK)

Author(s): Trahan, Travis John

Intended for: University Seminar

Issued: 2016-01-07

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.



An Overview of the Monte Carlo Application ToolKit (MCATK)

Travis Trahan

January 12, 2016

UNCLASSIFIED

Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

Outline

- **MCATK Overview**
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

What is MCATK?

- A C++ component-based Monte Carlo neutron-gamma transport software library
 - Development began in 2008
- Designed to build specialized applications
- Designed to provide new functionality in existing general purpose Monte Carlo codes like MCNP
- Developed with Agile software engineering methodologies

Current MCATK Contributors

- Terry Adams
- Steve Nolen
- Jeremy Sweezy
- Travis Trahan
- Chris Werner
- Lori Pritchett-Sheats (CCS-2)
- Rob Aulwes (CCS-7)

UNCLASSIFIED

MCATK Motivation: Reduce Costs

- Reduce code duplication
- Reduce code complexity
- Reduce time to deliver custom solutions
- Leverage external packages
- Provide re-usable components
- Find defects earlier in the development cycle

UNCLASSIFIED

MCATK: Reusable Components

- Provide components for:
 - Existing codes (example: MCNP)
 - Stand alone applications (example: MCATK K_{eff} solver)
- Components are well tested
 - Unit testing – verification
 - Integrated physics tests – validation
- Flat API available for C and Fortran codes
- Object Oriented API for C++ codes

UNCLASSIFIED

Outline

- MCATK Overview
- **Development Strategy**
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

MCATK: Development

- Agile software development methodology
 - Incremental development
 - Short iteration cycles (2 weeks)
- Test driven development (TDD)
 - Test first philosophy
 - Unit testing with UnitTest++
 - Unit tests built and executed continuously (with each commit)
 - ~10 minute build
- Pair-programming, Colocation
 - Improves design and testing
 - Promotes knowledge sharing and collective ownership of code



UNCLASSIFIED

MCATK: Continuous Testing and Reporting

MCATK toolkitlib

Dashboard Calendar Previous Current Next Project

No update data as of Tuesday, March 24 2015 - 19:00 MDT

Nightly

Site	Build Name	Update			Configure			Build			Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass					
candycorn	linux-gnu4.6-Release	2	0	0	0	0	0	0	0	0	0	888 ¹¹	Mar 24, 2015 - 20:05 MDT	
cielito	linux-gnu4.6-Debug	0	0	0	0	0	0	0	0	0	0	994 ¹¹	Mar 24, 2015 - 20:12 MDT	
cielito	linux-gnu4.8-Release	0	0	0	0	0	0	0	0	0	0	882 ¹¹	Mar 24, 2015 - 20:22 MDT	
cielito	linux-intel14.0-Debug	0	0	0	0	0	0	0	0	0	0	964 ¹¹	Mar 24, 2015 - 20:32 MDT	
cielito	linux-intel14.0-Release	0	0	0	0	0	0	0	0	0	0	882 ¹¹	Mar 24, 2015 - 20:42 MDT	
moonlight	linux-gnu4.6-Debug	3	0	0	0	0	0	0	0	0	0	808 ¹¹	Mar 24, 2015 - 20:01 MDT	
moonlight	linux-gnu4.6-Release	0	0	0	0	0	0	0	0	0	0	994 ¹¹	Mar 24, 2015 - 20:12 MDT	
moonlight	linux-intel14.0-Debug	0	0	0	0	0	0	0	0	0	0	882 ¹¹	Mar 24, 2015 - 20:22 MDT	
moonlight	linux-intel14.0-Release	0	0	0	0	0	0	0	0	0	0	964 ¹¹	Mar 24, 2015 - 20:32 MDT	
pinto	linux-gnu4.6-Debug	0	0	0	0	0	0	0	0	0	0	888 ¹¹	Mar 24, 2015 - 20:04 MDT	
pinto	linux-gnu4.6-Release	0	0	0	0	0	0	0	0	0	0	994 ¹¹	Mar 24, 2015 - 20:11 MDT	
pinto	linux-intel14.0-Debug	0	0	0	0	0	0	0	0	0	0	882 ¹¹	Mar 24, 2015 - 20:22 MDT	
pinto	linux-intel14.0-Release	0	0	0	0	0	0	0	0	0	0	964 ¹¹	Mar 24, 2015 - 20:32 MDT	
retumbo	linux-gnu4.8-RelWithDebInfo	2	0	0	0	0	0	0	0	0	0	994		
titgroup	linux-gnu4.6-RelWithDebInfo	2	0	0	0	0	0	0	0	0	0	991		
vargnet	linux-intel14.0-RelWithDebInfo	0	0	0	0	0	0	0	0	0	0	964		

Continuous

Site	Build Name	Update			Configure			Build			Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass					
candycorn	linux-gnu4.6-Release	0	0	0	0	0	0	0	0	0	0	810 ¹¹		

Code coverage (nightly)

Site	Build Name	Update			Configure			Build			Test			Build Time
		Files	Error	Warn	Error	Warn	Not Run	Fail	Pass					
	linux-gnu4.6-Debug	3	0	0	0	0	0	0	0	0	0	808 ¹¹	Mar 24, 2015 - 20:42 MDT	
	linux-gnu4.6-Release	2	0	0	0	0	0	0	0	0	0	808	Mar 24, 2015 - 20:01 MDT	

Dynamic Analysis

Checker	Defect Count	Date
Valgrind	452	Mar 24, 2015 - 20:42 MDT

Nightly tests

- run longer
- include more tests
- variety of platforms/compilers

New entry generated after every commit

Code coverage (nightly)

Memory leaks etc. (even 3rd party libraries!) (nightly)

UNCLASSIFIED

Outline

- MCATK Overview
- Development Strategy
- **Available Algorithms**
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

Power Iteration

- k-Eigenvalue solver
- α -Eigenvalue solver
 - The α -Eigenvalue is the inverse of the reactor period
 - Uses “time absorption” i.e. $\alpha/velocity_{0n}$
- Validation suite covers all of the criticality test problems from the MCNP validation suite.

Note: Shapiro-Wilk test for normality shows that at the 5% significance level, the differences between codes cannot be shown to be non-normally distributed.

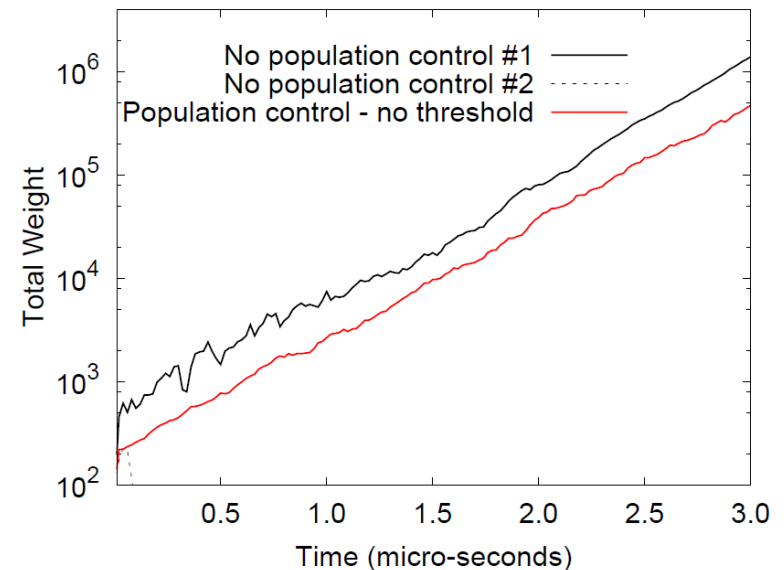
Case	MCNP		MCATK		$\Delta k_{eff} / \sigma_{\Delta}$
	k_{eff}	Std. Dev.	k_{eff}	Std. Dev.	
BAWXI2	1.0092	0.0013	1.0064	0.0013	-1.52
BIGTEN	0.9919	0.0009	0.9944	0.0010	1.86
FLAT23	0.9992	0.0010	0.9982	0.0011	-0.67
FLAT25	1.0014	0.0010	1.0030	0.0010	1.13
FLATPU	1.0003	0.0010	0.9998	0.0010	-0.35
FLSTF1	0.9838	0.0017	0.9868	0.0018	1.21
Godiva	1.0000	0.0008	0.9998	0.0009	-0.17
GodivaR	1.0197	0.0011	1.0170	0.0012	-1.66
HISHPG	1.0115	0.0010	1.0113	0.0009	-0.15

Work performed by Jesse Giron, ASU

UNCLASSIFIED

Time-Dependent Solver

- Performs dynamic α -eigenvalue calculations
- Particles are added to “census” if they survive to the end of the time step
- Population control is applied to the census between time steps
- This prevents exceeding memory limits for supercritical systems, and prevents the population from dying off in subcritical or stochastic systems (i.e., systems with small neutron populations)



Total weight vs. time for a super-critical analytic problem using the MCATK time-dependent algorithm

UNCLASSIFIED

Fission Chain Analysis

- Each source particle begins a fission chain
- Every descendent of the initiating particle is marked as being part of the same chain
- Tally quantities by fission chain
 - Total chain length
 - Number of fission events
 - Number of particles alive in a chain at a given time
- The fission chain analysis is particularly useful for analysis of stochastic systems (systems in which the neutron population is very small)
 - Distribution of neutron populations at a specified time
 - POI/POE
 - Multiplication

UNCLASSIFIED

Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- **Problem Modeling (Sources, Geometry, Data)**
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

Sources

- Point sources
- Mesh sources
- Time-varying distributions
- Spontaneous fission
 - Can use average multiplicity or sample from actual multiplicity data
- Surface source files
 - Link to MCNP

UNCLASSIFIED

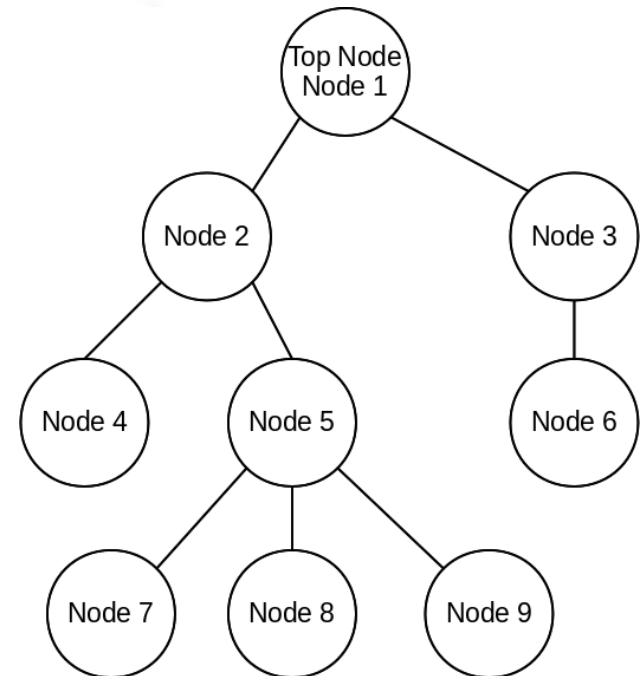
MCATK Geometry

- LNK3DNT mesh files
- Mesh specification through the API
- Mesh types supported:
 - Spherical: R
 - Cylindrical: R, RZ, RZ Θ
 - Cartesian: X, XY, XYZ
- 3-D Solid Body Geometry
 - Available geometric primitives:
 - Spheres
 - Cylinders (right finite cylinder)
 - Boxes (right parallel piped)
 - Cones (truncated circular cones)
 - Each primitive can have rotations and translations

UNCLASSIFIED

More on Solid Body Geometry

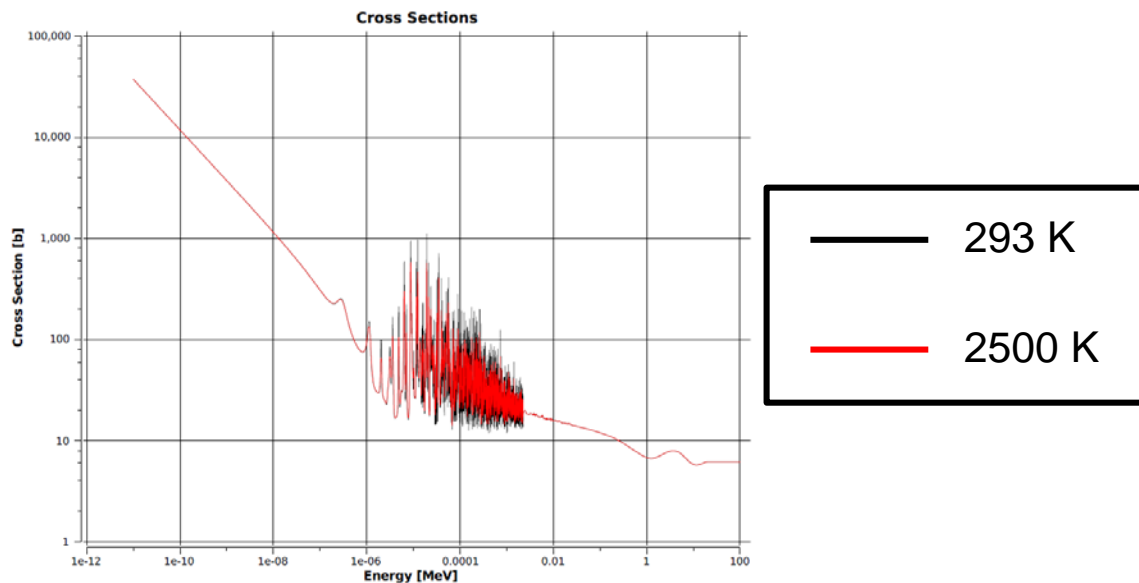
- MCATK uses a scene graph hierarchy
 - Each solid in the scene is assigned to a node in a tree
 - Each node is a child of (contained by) the node above
 - Similar to CERN's ROOT and GEANT4 geometries
- Nodes have the following properties:
 - Each node is assigned a geometric primitive
 - Each node can have other nodes registered as children
 - A node can have an optional transformation that will be inherited by its children
 - Children are allowed to overlap, but MCATK applies an order of precedence
 - Parent containment and child precedence can be used to replicate combinatorial operators
- A node can also be assigned a mesh object, but these nodes cannot have any children



UNCLASSIFIED

Cross Section Data and Multi-Temperature Treatment

- MCATK uses continuous energy data read from ACE data files
- Multi-temperature treatment:
 - MCATK is able to read in cross sections processed at more than one temperature
 - It picks the cross section table that was processed at the temperature that is closest to, but does not exceed the cell temperature
 - This capability is expected to be a step towards Doppler broadening on-the-fly and other multi-temperature treatments



UNCLASSIFIED

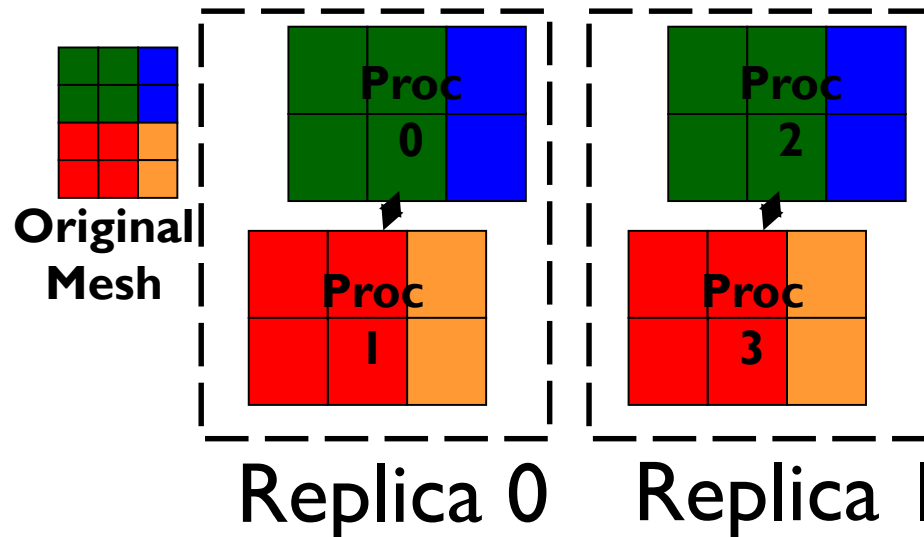
Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- **Parallelism**
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

Parallel Models

- MPI (distributed memory) only
 - Domain decomposed
 - Domain replicated (particle decomposed)
 - Hybrid



- Threading (shared memory) models currently being implemented (C++11 threads)
- Many-core threading models on the near horizon

UNCLASSIFIED

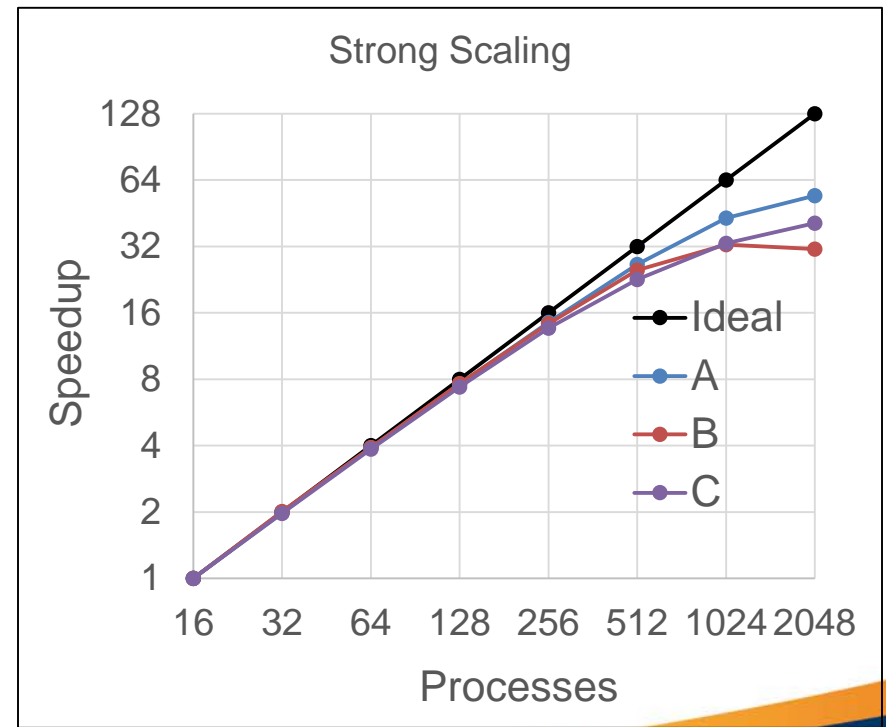
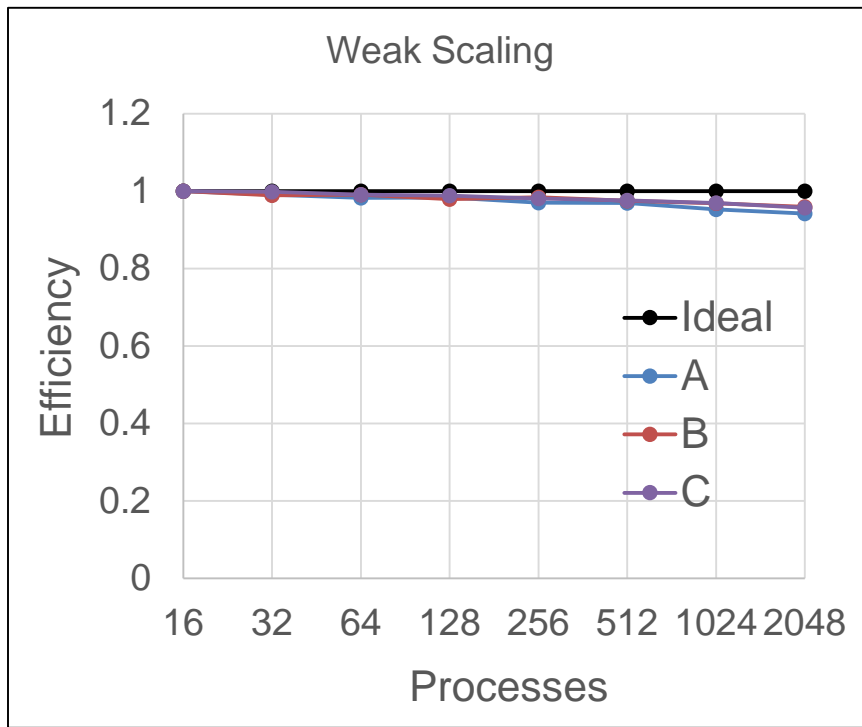
Parallel Performance

- A - Godiva - 1-D spherical calculating k_{eff} with the *power-iteration algorithm*
 - 1,000 Active/50 Settles
 - 10,000 particles/generation/process (weak)
 - 500,000 particles/generation (strong)
- B - Godiva - 1-D spherical calculating alpha time constant and neutron fluence rates with the *time-dependent algorithm*:
 - 250,000 particles/proc (weak) : 500 steps @ 0.1sh
 - 1.5 M particles (strong) : 2500 steps @ 0.1 sh
- C - ZEBR8H - 2-D cylindrical calculating k_{eff} with the *power-iteration algorithm*:
 - 1,000 Active Cycles/50 Settles
 - 5,000/generation/proc (weak)
 - 100,000 particles/generation (strong)
- Performance tests are run as part of the weekly test suite
 - Tests may not involve enough work for all processors

UNCLASSIFIED

Parallel Performance

- Good weak scaling up to 2048 procs
- Strong scaling tests may not involve enough work for all processors
- To this point, the focus of MCATK has been on capability rather than performance



UNCLASSIFIED

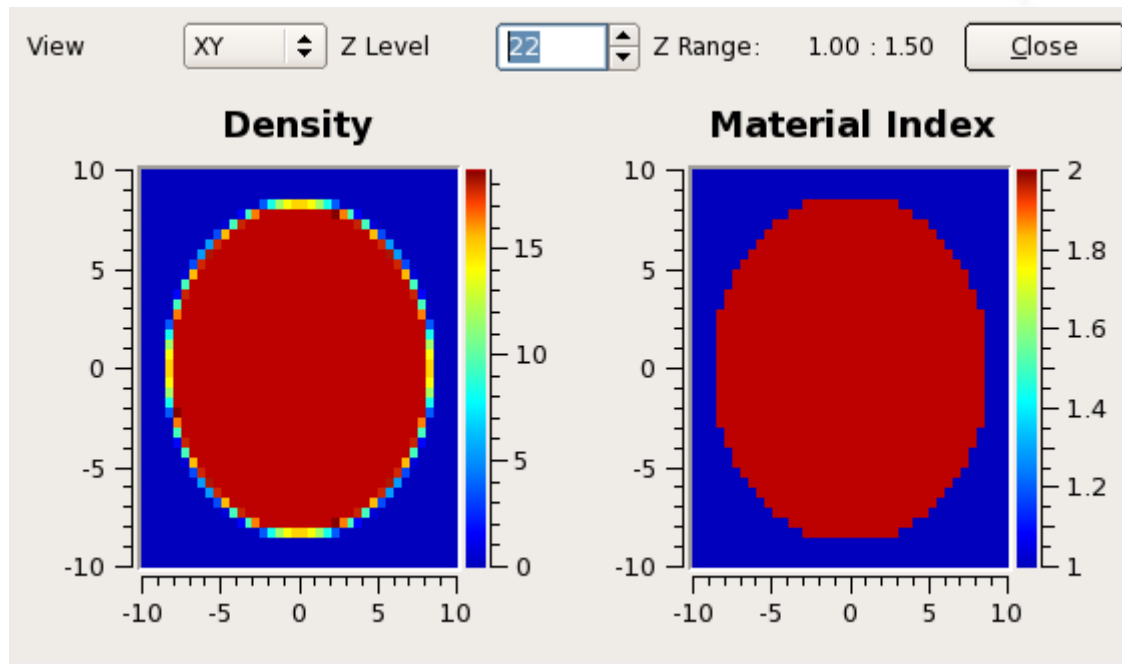
Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- **Miscellaneous Tools/Features**
- Example MCATK Application
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

LNK3DNT Viewer

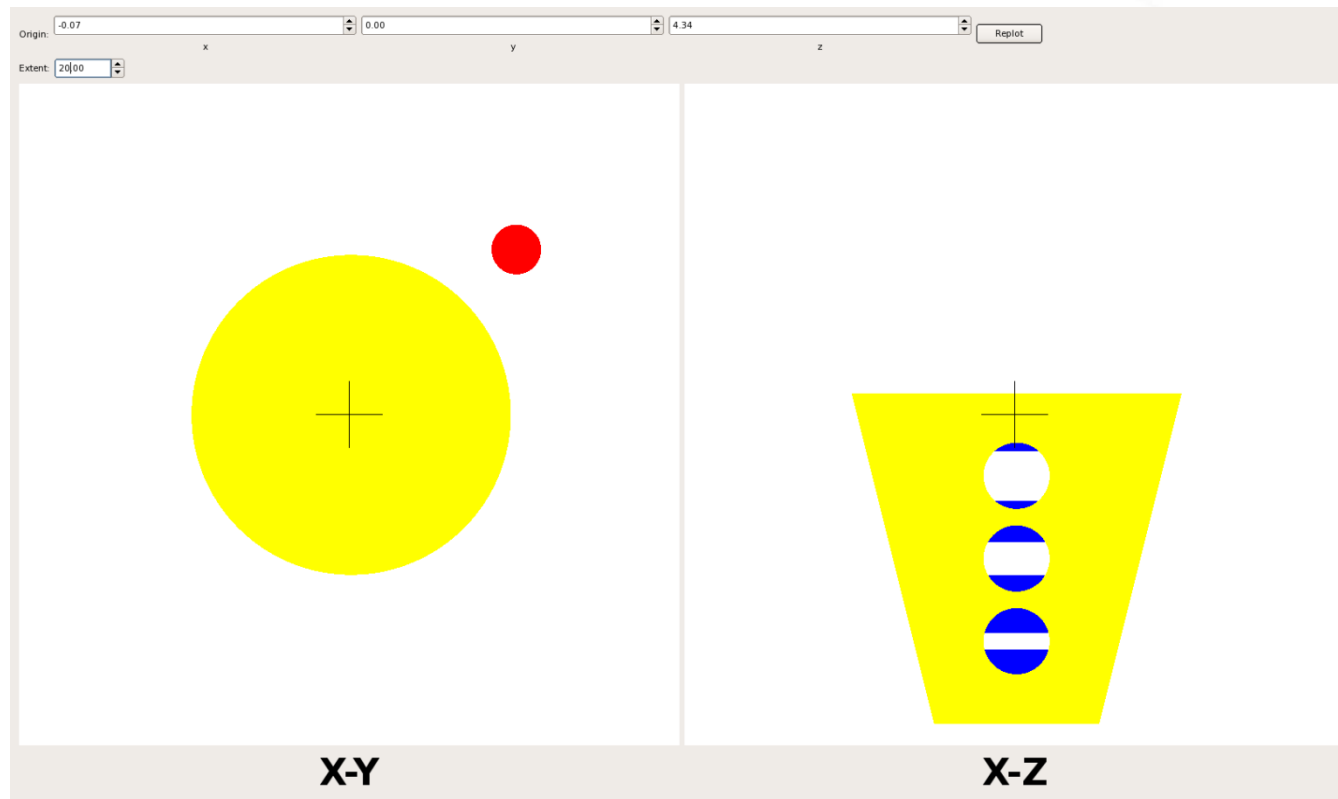
Godiva critical assembly via LNK3DNT file



UNCLASSIFIED

Geometry Plotting Tool

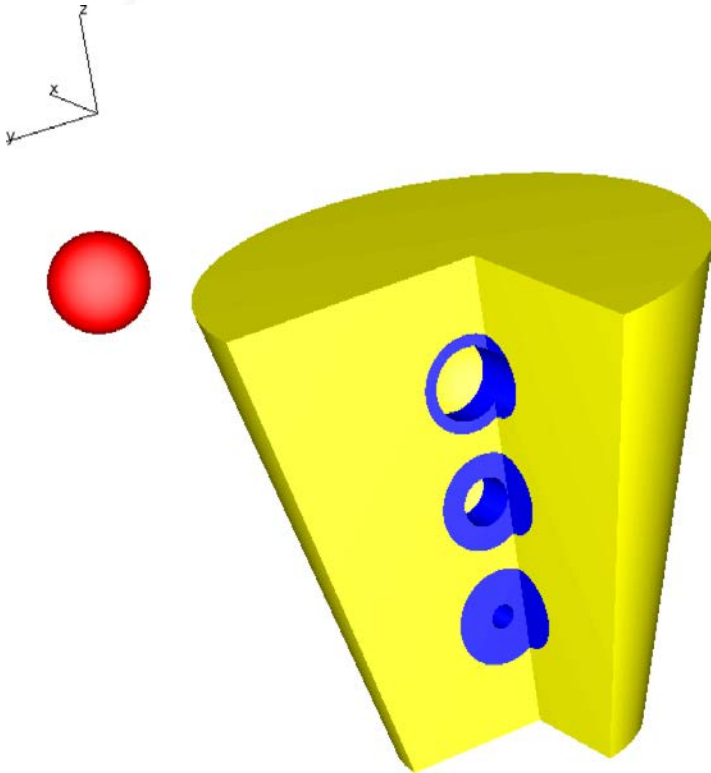
- Similar to MCNP's plotting tool
 - 2 cross-sectional views with a common origin



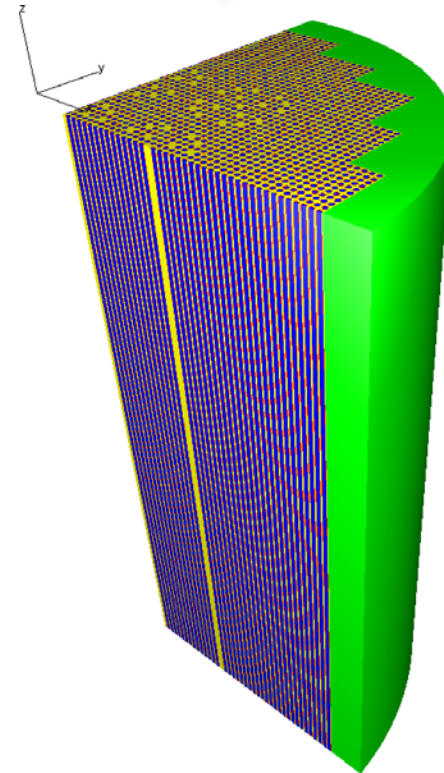
UNCLASSIFIED

Geometry Rendering Tool

- 3-D Renderer
- Optional chop box for interior visualization



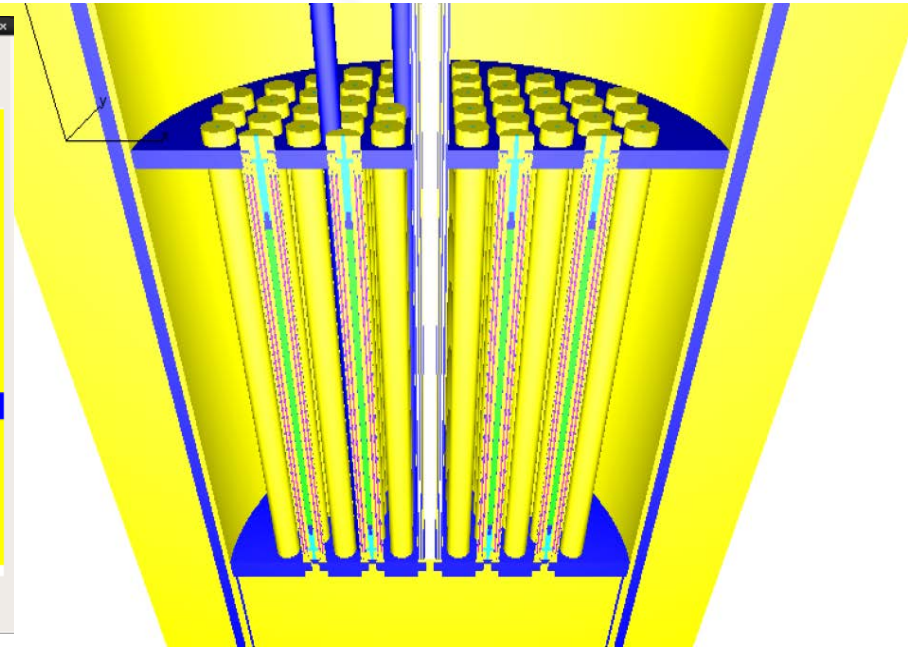
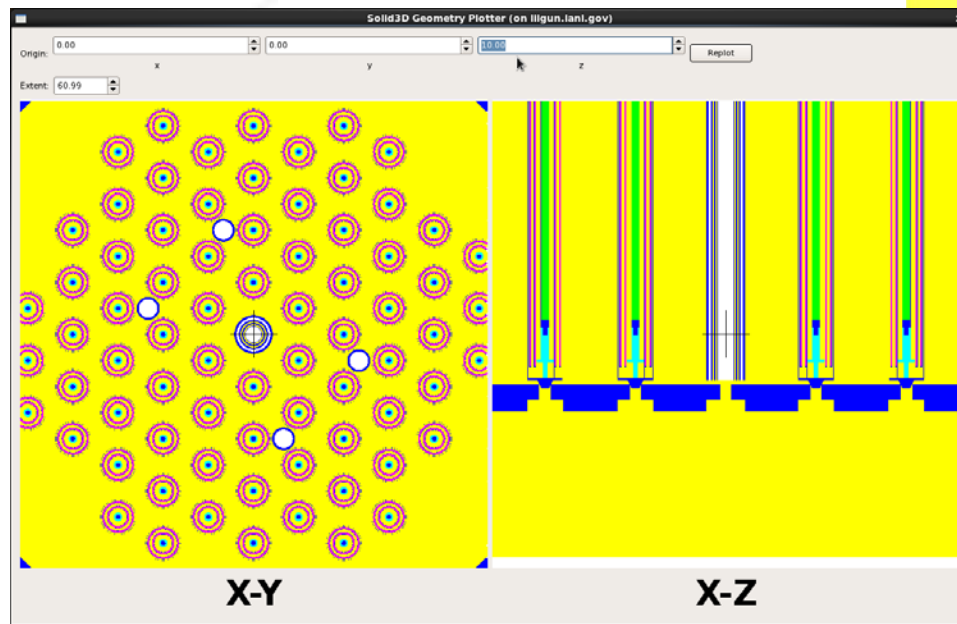
Radiographic Test Object



BAWXI2 Critical Benchmark

UNCLASSIFIED

Solid Body Geometry Enables Modelling More Complex Systems Than Meshes

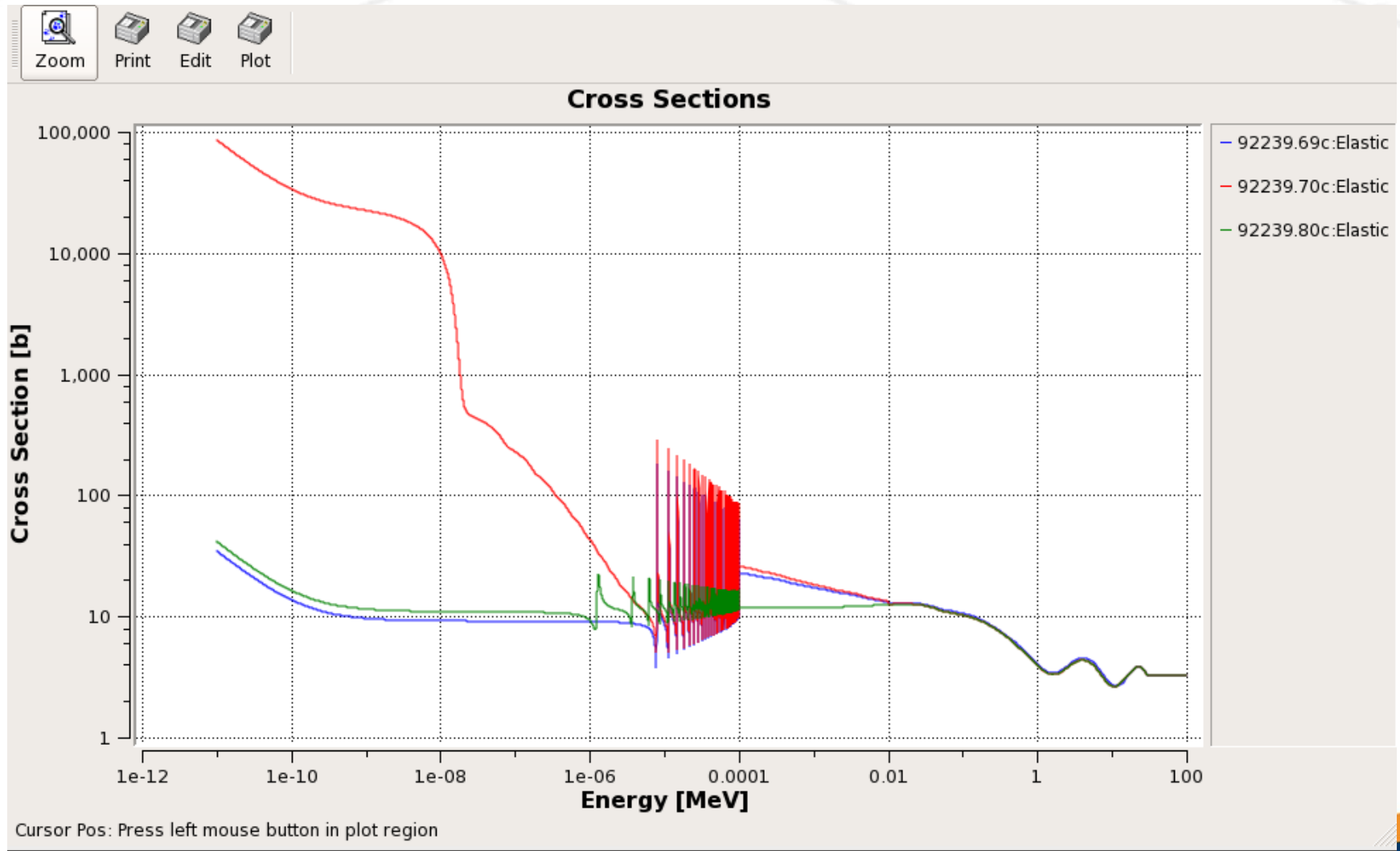


ICT2C3 Critical Benchmark

Work performed by Jesse Giron, ASU

UNCLASSIFIED

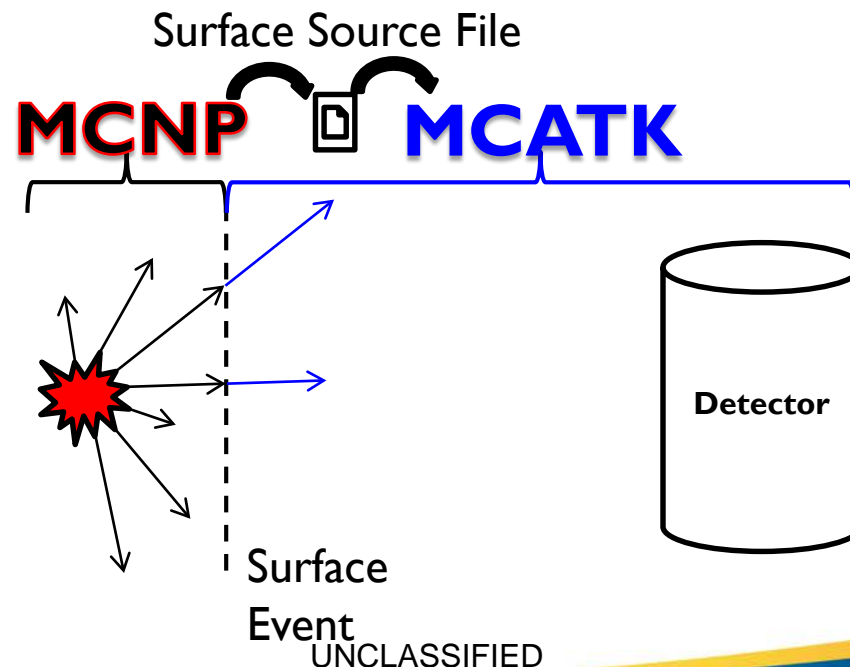
Cross Section Viewer



UNCLASSIFIED

Surface Source Write/Read Capability

- Compatible with MCNP SSW/SSR feature
 - Serves as a link between MCNP and MCATK
- Saves the state of each particle that exits the geometry
 - Could be extended to save particle states at other boundaries (e.g., interior surfaces or time/energy boundaries)
- Particles can be read in and treated as a source



Surface Source File Manager

- The SurfaceSrcFileManager application can be used to:
 - Compare the contents of two surface source files
 - Convert a file from MCATK format to MCNP format and vice versa
 - Merge multiple surface source files into a single file
 - Sort the particles on a file according to their current time (useful for time-dependent calculations)
 - View the contents of a surface source file

```
./SurfaceSrcFileManager view mcatk file.ssr 1
```

Particle 1:

```
NParticles:          10000
NStartingParticles:  10000
Original Compilation Date: 07/29/14
Original Revision Number: r3377
File created:       07/29/14
16:12:02
```

```
ID:          0
Particle Type: 8
Surface:     99999
Weight:      1
Energy:      14
Time:        0.0005
Position:    (0,0,0)
Direction:   (1,0,0)
```

UNCLASSIFIED

Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- **Example MCATK Application**
- Recent Areas of Research
- Summary and Future Work

UNCLASSIFIED

Example Applications

- Plotting tools
- ParallelEigen and SerialEigen Drivers
- Perturbation Applications: setup materials once, then solve a problem on multiple geometries that contain the same materials
- Multiple-Algorithm Solver
 - Use the same materials and geometry with multiple solution algorithms
 - Static k_{eff}
 - Static α
 - Time-dependent α

UNCLASSIFIED

Multiple-Algorithm Solver

```
namespace mpi = boost::mpi;

int main(int argc, char* argv[]){
    mpi::environment env( argc, argv);
    mpi::communicator world;

    mcatk::ProgramOptions po(argc, argv); // parse command line arguments
    mcatk::Setup driver( po.getInputName(), world.rank() ); // parse input file

    driver.buildMaterials(); // setup all problem materials
    driver.buildMesh(); // setup problem mesh

    driver.solveEigen(); // perform static keff and static alpha calculation
    driver.solveTimeDependent(); // perform time-dependent alpha calculation

    return 0;
}
```

UNCLASSIFIED

Multiple-Algorithm Solver

```
void Setup::build_materials(){
    std::vector<int> material_id;
    int HEU_id = 0; double HEU_density=18.79;

    ContinuousMat* HEU_mat = new ContinuousMat();
    HEU_mat->addByAtomicFraction( "92235.70c", 0.94 );
    HEU_mat->addByAtomicFraction( "92238.70c", 0.06 );
    multimaterial.add_material( *HEU_mat, HEU_id );
    layer_density.push_back(HEU_density);
    material_id.push_back(HEU_id);

    multimaterial.useTotalNu();
    multimaterial.enableImplicitCapture();
    multimaterial.useThermalScatterTreatment(true);

    pMaterialProperties = new MaterialProperties( material_id, layer_density,
    material_id.size() );
}
```

UNCLASSIFIED

Multiple-Algorithm Solver

```
void Setup::buildMesh(){
    typedef TransportMesh<1,Spherical> Mesh_t;
    Mesh_t::Vertices_t verts;
    // Initialize verts...
    pMesh = new Mesh_t( verts );
}

void Setup::buildSource(){
    pSource.reset( new mcatk::IntrinsicSource("u238") );
    pSource->useIsotropicDirection();
    SourceManager::instance().AddSource( pSource, "U238_SpontaneousFissionSource" );
}
```

UNCLASSIFIED

Multiple-Algorithm Solver

```
void Setup::solveEigen(){
    StaticAlpha EigenSolver;
    StaticAlpha::KeffAlgoPtr_t pKeff = EigenSolver.getKeffAlgorithm();
    EigenSolver.setAlphaCycles( numAlphaActiveCycles );
    // set other StaticAlpha options...

    buildSource(); // setup initial source for the first cycle
    AlgorithmAssistant<StaticKeff,TransportMesh<1,Spherical>> helper( *pMesh,
        *pKeff, SourceManager::instance() ); // AlgorithmAssistant handles parallelism
    and sources

    EigenSolver.setupForParallel( helper );
    EigenSolver.setupAlgorithm( *pMesh, multimaterial, *pMaterialProperties );
    helper.useLoadBalancing();

    // perform static keff and static alpha calculations...
```

UNCLASSIFIED

Multiple-Algorithm Solver

```
void Setup::solveTimeDependent(){
    TimeDependent TD;
    double start_time=0.0;
    TD.initializeCensusTime( start_time );

    buildSource();

    AlgorithmAssistant<TimeDependent,TransportMesh<1,Spherical>> helper( *pMesh,
    TD, SourceManager::instance() ); // AlgorithmAssistant handles parallelism and
    sources

    TD.setupAlgorithm( *pMesh, multimaterial, *pMaterialProperties );
    TD.setNTargetParticles( numTDHistories );
    helper.useLoadBalancing();

    // perform time-dependent calculation...
```

UNCLASSIFIED

Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- **Recent Areas of Research**
- Summary and Future Work

UNCLASSIFIED

What is a Stochastic System?

- Weak sources (i.e., low neutron populations)
- Fission chains do not overlap
- Average behavior of the neutron population is not sufficient to describe the system

- Examples:
 - Pulsed nuclear reactors (e.g., Godiva, Caliban)
 - Criticality accident scenarios

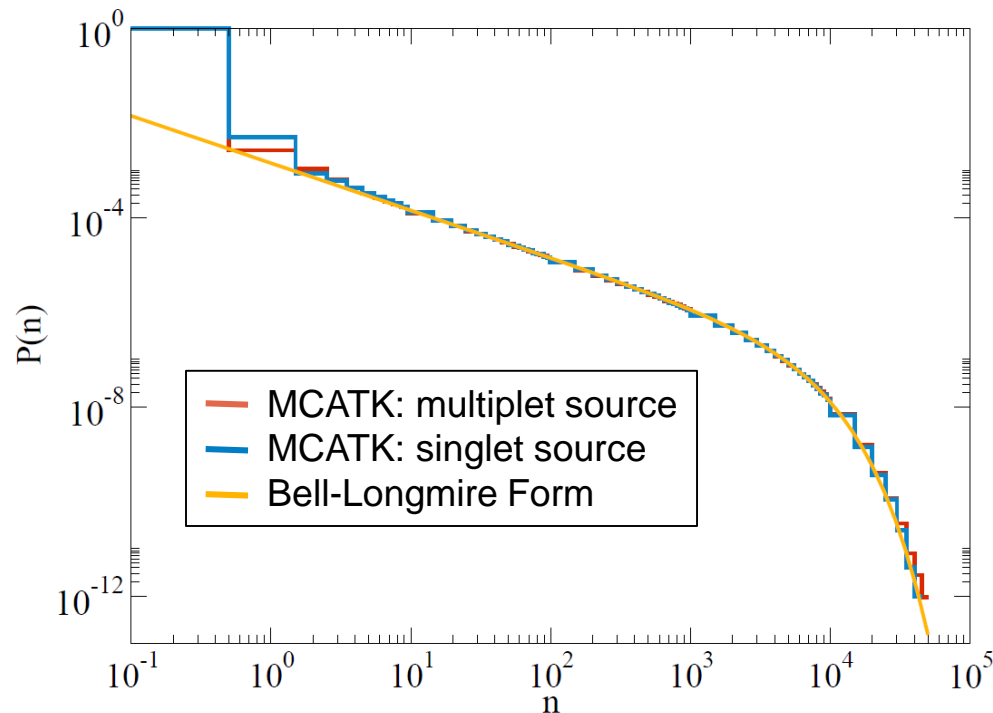
UNCLASSIFIED

Slide 39

Instantaneous Neutron Population is Defined by a Distribution

- MCATK can be used to tally the probability that a fission chain is comprised of n neutrons at a given time, $P(n)$.

PDF of Neutron Number, n , for a Pu Sphere



Work performed by Erin Fichtl (LANL)

UNCLASSIFIED

Deterministic Analysis of Stochastic Systems

- Deterministic methods for characterizing stochastic systems first developed by Bell
- Deterministic analysis has been implemented in PARTISN and PANDA
- Deterministic equations take the form of coupled adjoint transport equations with modified source terms
- Can calculate:
 - Single particle probabilities of initiation / extinction / survival (POI / POE / POS)
 - Moments of the neutron population distribution resulting from single neutrons or neutron sources

UNCLASSIFIED

Slide 41

Monte Carlo Analysis of Stochastic Systems

- Fission chains do not overlap
- Simulate the random histories of complete fission chains and observe their stochastic behavior
- Can tally:
 - Detector response or dose for every fission chain
 - Total / net / leakage multiplication
 - POI / POE / POS
 - Moments of the neutron population distribution resulting from single neutrons or neutron sources
- POI capabilities have been implemented in Mercury, a research version of MCNP, and the Monte Carlo Application ToolKit (MCATK)

UNCLASSIFIED

Slide 42

Fission Chain Algorithm Basics

- Each source particle gets an ID, which is inherited by all progeny to identify them as being a part of the same chain
- At any given time, we know:
 - Total chain length, *ChainLength*
 - Instantaneous chain population, N
- In principle, any standard tally can be accumulated on a chain-by-chain basis

UNCLASSIFIED

Slide 43

Static Analysis: System Multiplication

- Total Multiplication: Total neutrons produced per source neutron

$$M_T = Avg(ChainLength)$$

- Net Multiplication: Net neutrons produced per source neutron (excludes neutrons lost to fission and NXN reactions)

$$M_N = Avg(ChainLength - NumChainFissions - NumChainNXN)$$

- Leakage Multiplication: Number of neutrons that leak per source neutron

$$M_L = Avg(ChainLeakage)$$

UNCLASSIFIED

Slide 44

Static Analysis: Probability of Initiation/Extinction

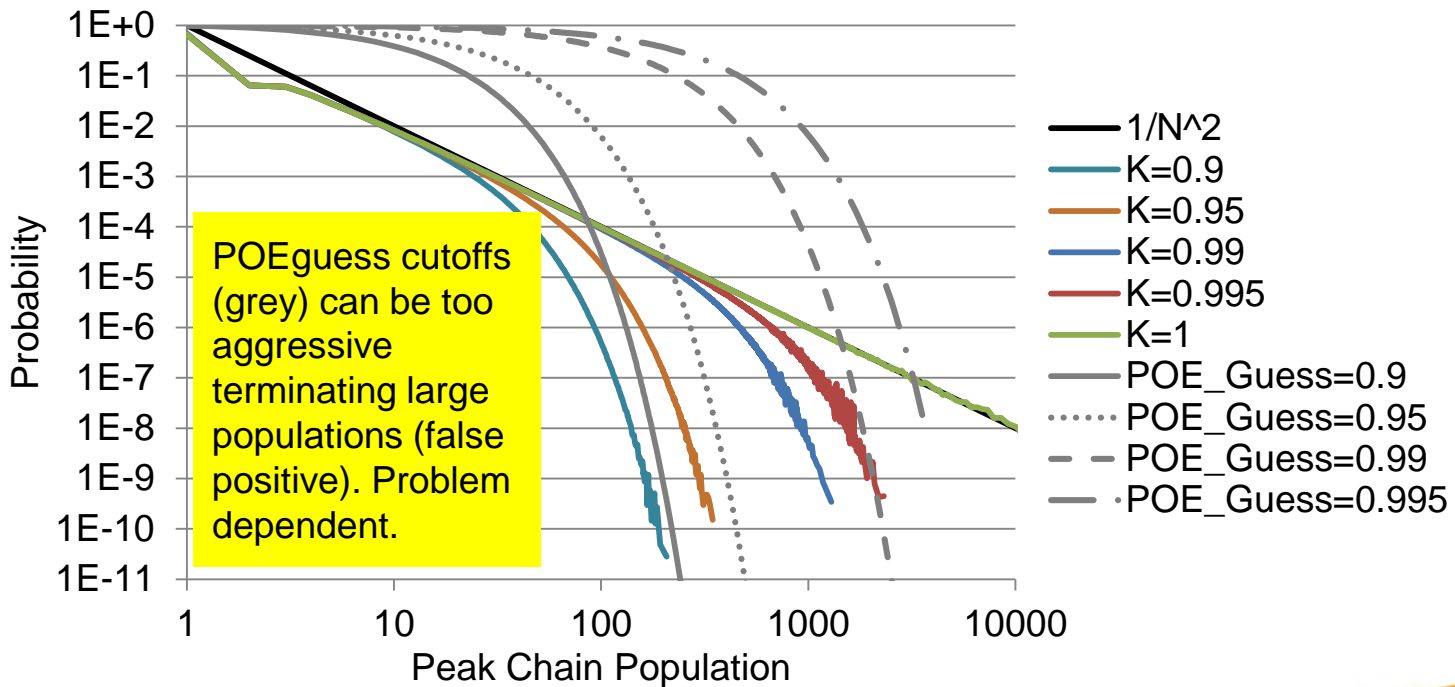
- Probability of Initiation ($1 - \text{Probability of Extinction}$):
Probability that a fission chain persists for an infinite time
- Cannot track an infinitely long chain, must terminate chains
- Cutoff method for POI:
 - Specify a cutoff length or population above which a chain is terminated
 - Score the terminated chain to POI
 - Overestimation bias
- Importance methods for POE:
 - Give each chain an importance and a weight
 - Larger chains are less important to POE
 - Play Russian Roulette on entire fission chains

UNCLASSIFIED

Slide 45

Static Analysis: Probability of Initiation/Extinction

- Booth Importance: $I = \frac{1}{(POE_{Guess})^{N-1}}$
- “N Left” Importance: $I = \frac{C}{N^2}$



UNCLASSIFIED

Slide 46

Dynamic Analysis Quantities of Interest

- Time-integrated or time-dependent quantities on a chain-by-chain basis (e.g. total dose or instantaneous dose rate)
- Burst wait time of pulsed reactor experiments
- Probability of Survival: probability that a fission chain persists until some designated time

UNCLASSIFIED

Slide 47

Dynamic Analysis Algorithm

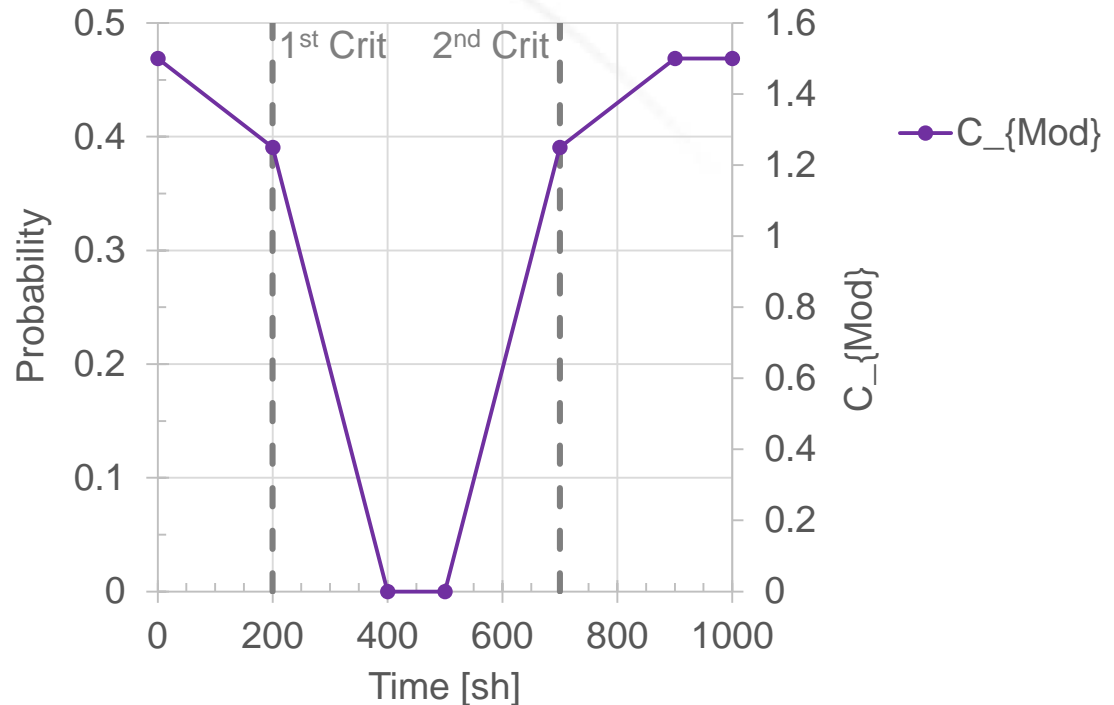
- We are interested in fission chain behavior over the entire transient, so we cannot terminate chains
- We do not want chains to grow too large (memory constrained), but we do want to allow them to become small and die off
- Algorithm summary:
 - Initially, track all chains in an analog fashion
 - When the chain population exceeds a threshold, *apply population control on a chain-by-chain basis*
 - Can roulette if the chain grows, or split if the chain shrinks
 - If the average weight of particles in the chain drops below the initial source particle weight due to splitting, track analog again

UNCLASSIFIED

Slide 48

Dynamic Test Problem: Definition

- Numerical test problem defined by R. Baker and studied again by E. Fichtl
- Bare uranium sphere with time-varying enrichment:
$$\rho_{U235} = 14 - 6C_{Mod}$$
$$\rho_{U238} = 1 + 6C_{Mod}$$
- Source: 1 MeV isotropic point source at center of sphere (different than Baker)

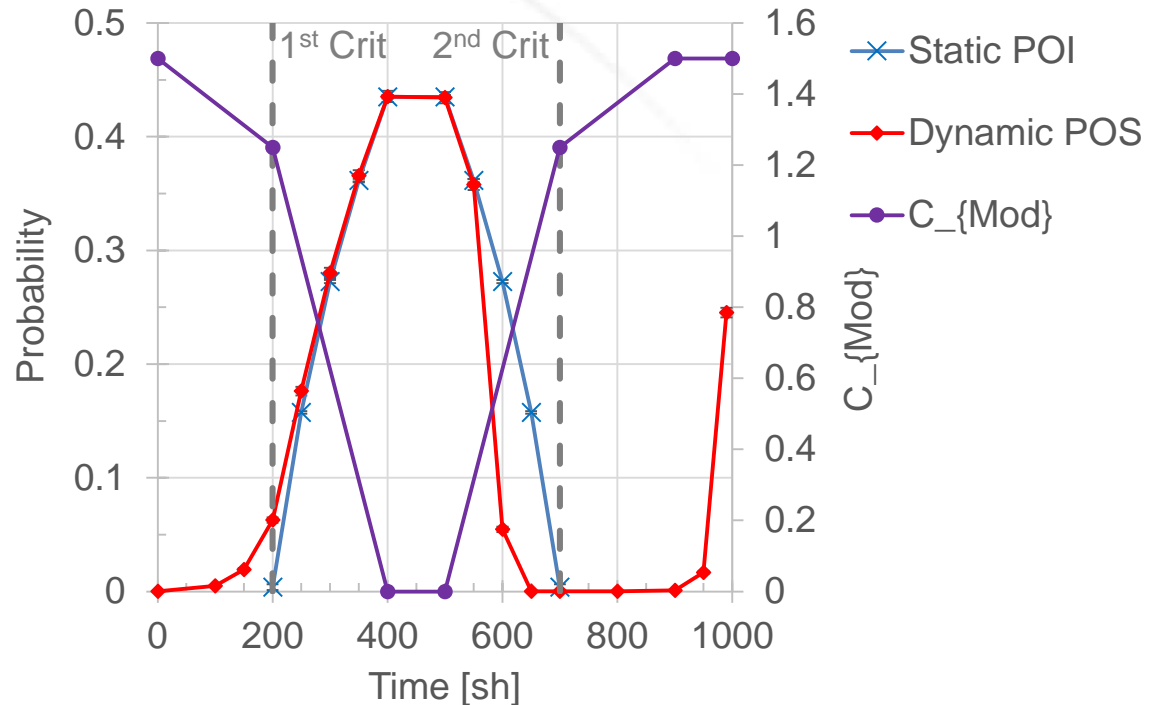


UNCLASSIFIED

Slide 49

Dynamic Test Problem: Results

- POI is 0 until first criticality, but POS is greater than 0 even before first criticality
- POI is much larger than the POS prior to second criticality
- POS goes to 1 at the final transient time



Static analysis is insufficient to describe a transient system

UNCLASSIFIED

Dynamic Test Problem: Results

Comparison of Dynamic Simulations with Different Combing Thresholds (1e7 Fission Chains).

Comparison of Dynamic Simulations with Different Numbers of Chains (1e3 Combing Threshold).

Combing Threshold	POS	Peak Neutron Weight	Peak MC Population
1e2	6.75e-5	1.93e50	4.65e6
1e3	6.62e-5	2.04e50	4.49e7
1e4	6.62e-5	1.98e50	4.48e8
1e5	6.62e-5	1.97e50	4.46e9

Number of Fission Chains	POS	Peak Neutron Weight	Peak MC Population
1e5	4.00e-5	8.60e49	2.84e5
1e6	5.70e-5	1.83e50	3.85e6
1e7	6.62e-5	2.04e50	4.49e7
1e8	6.22e-5	1.87e50	4.21e8

- Peak neutron weight shows that analog transport would be impossible
- Must use enough fission chains AND a high enough combing threshold to converge

UNCLASSIFIED

Slide 51

Outline

- MCATK Overview
- Development Strategy
- Available Algorithms
- Problem Modeling (Sources, Geometry, Data)
- Parallelism
- Miscellaneous Tools/Features
- Example MCATK Application
- Recent Areas of Research
- **Summary and Future Work**

UNCLASSIFIED

MCATK: Summary

- MCATK physics:
 - Continuous energy neutron-gamma transport with multi-temperature treatment
 - Static eigenvalue (k and α) algorithms
 - Time-dependent algorithm
 - Fission chain algorithms
- MCATK geometry:
 - Mesh geometries
 - Solid body geometries
- MCATK provides:
 - Verified, unit-tested Monte Carlo components
 - Flexibility in Monte Carlo applications development
 - Numerous tools such as geometry and cross section plotters
- Public availability is a possibility

UNCLASSIFIED

MCATK: What's Next?

- Variance Reduction
- New parallel models
 - Threaded parallel model
 - Many-core parallel model
- New physics treatments
 - S(alpha,beta) treatment
 - Unresolved resonance treatment
- New tallies
 - Energy deposition
 - Next-event estimators (i.e., point detectors)
 - Additional fission chain tallies

UNCLASSIFIED

Where to Learn More

MCATK Citation Paper:

T. ADAMS, S. NOLEN, J. SWEEZY, A. ZUKAITIS, J. CAMPBELL, T. GOORLEY, S. GREENE, R. AULWES, “Monte Carlo Application ToolKit (MCATK),” *Annals of Nuclear Energy*, **82**, 41 (2015).

MCATK at M&C/SNA/MC:

J. SWEEZY, S. NOLEN, T. ADAMS, T. TRAHAN, L. PRITCHETT-SHEATS, “Monte Carlo Applications ToolKit (MCATK): Advances for 2015,” *Proc. M&C 2015*, Nashville, Tennessee, April 19-23, 2015.

S. NOLEN, “Using Fission Chain Analysis to Inform Probability of Extinction/Initiation Calculations with MCATK,” *Proc. M&C+SNA+MC 2015*, Nashville, Tennessee, April 19-23, 2015, American Nuclear Society (2015).

MCATK at ANS:

T.J. TRAHAN, S.D. NOLEN, “A Monte Carlo Algorithm for Fission Chain Analysis of Dynamic Stochastic Systems,” *Trans. Am. Nuc. Soc.*, **113**, 661 (2015).

Contact Info: Travis Trahan, tjtrahan@lanl.gov

UNCLASSIFIED

MCATK vs MCNP Timing

Problem	MCNP Time [s]		MCATK Time [s]		MCATK Time / MCNP Time	
	1 Proc	16 Procs	1 Proc	16 Procs	1 Proc	16 Procs
GodivaR	537.04	44.92	1837.91	140.31	3.42	3.12
UH3C6	167.25	18.86	492.20	37.22	2.94	1.97
ICT2C3	967.32	83.86		364.61		4.35

Each problem run with 200 inactive cycles, 300 active cycles, and 10000 particles per cycle.

UNCLASSIFIED