

Monte Carlo Application ToolKit (MCATK): Advances for 2017

Travis J. Trahan, Terry R. Adams, Rob T. Aulwes, Steven D. Nolen, Jeremy E. Sweezy, Chris J. Werner

Los Alamos National Laboratory, PO Box 1663, Los Alamos, NM, 87545

tjtrahan@lanl.gov, tadams@lanl.gov, rta@lanl.gov, sdnolen@lanl.gov, jsweezy@lanl.gov, cwerner@lanl.gov

Abstract - *The Monte Carlo Application ToolKit (MCATK) is a component-based Monte Carlo neutron-photon transport library developed at Los Alamos National Laboratory (LANL). This paper presents features that were added to the code since 2015. The significant features that were added to the code include: expanded source modeling, photon transport, redesign and expansion of tally capabilities, weight windows, new methods of stochastic systems analysis, a Python interface for setting up material, geometry, and source XML input files, and new driver applications.*

I. INTRODUCTION

The Monte Carlo Application ToolKit (MCATK) is a component-based Monte Carlo neutron-photon transport library developed at Los Alamos National Laboratory (LANL) [1]. In 2015, a paper was presented at M&C+SNA+MC to provide an overview of recently implemented features in MCATK [2]. Similarly, the purpose of this paper is to present features that have been added to the code since that paper was written.

The significant features that were added to the code include: expanded source modeling, photon transport, redesign and expansion of tally capabilities, weight windows, new methods of stochastic systems analysis, a Python interface for setting up source, material, and geometry XML input files, and new driver applications.

Section II provides a brief overview of MCATK and features available prior to this update. In Section III we define a common test problem that will be used to demonstrate each of the new features one-by-one. The new features are described and demonstrated in Section IV. We conclude in Section V with a discussion of future work and anticipated features.

II. MCATK OVERVIEW

This section contains an overview of the major capabilities of MCATK that were available as of the last summary paper [2]. MCATK is a modern C++, component-based software library for Monte Carlo particle transport developed at LANL since 2008 [1]. The purpose of MCATK is to provide robust, well-tested, reusable software components for Monte Carlo transport applications. The components may be used to rapidly develop specialized Monte Carlo applications, or to provide new functionality to existing Monte Carlo codes such as MCNP [3].

Particles are transported using continuous energy data from ACE formatted nuclear data files produced by NJOY [4]. Problem geometries can be represented by 1-D (r or x), 2-D (rz or xy), and 3-D ($rz\theta$ or xyz) meshes, or by solid body geometries. Solid body geometry representations in MCATK are created from simple volumes (boxes, cones,

cylinders, and spheres) [5] as opposed to the MCNP constructive solid geometry, which is surface-based. From a user perspective, however, the MCATK solid body geometry may be viewed as analogous to modeling a geometry in MCNP using only macrobodies.

MCATK has two eigenvalue solution modes for calculating k - and α -eigenvalues [6], a time-dependent mode (similar to MCNP's fixed source mode, but with population control to enable simulations of systems at any level of criticality) [7], and a fission chain analysis mode for calculating properties of stochastic systems, including multiplication, probability of initiation [8], and moments of the neutron and fission number distributions [9].

III. COMMON TEST PROBLEM DEFINITION

Many of the features presented in the following section are demonstrated using a common test problem. The problem geometry is a cylinder with a radius of 100 cm and a height of 50 cm. The geometry is divided into 10 cylinders, each 5 cm thick, for tallying purposes. The tally regions are numbered 1-10, with region 1 extending from the base of the cylinder ($z=0$ cm) to ($z=5$ cm), and the others numbered consecutively such that region 10 is at the far end of the cylinder ($z=45$ cm to $z=50$ cm). The cylinder is homogeneous and composed entirely of light water at a density of 1g/cc. The source is a planar neutron source at the base of the cylinder, i.e., a disc of radius 100 cm. The angular distribution of the source is a uniform cone with angle cosines between 0.9 and 1.0 relative to the incident normal vector. The energy distribution of the neutrons is a Watt fission spectrum for thermal-induced fission in uranium-235. The Watt spectrum parameters are those recommended in Appendix D of the MCNP6 User's Manual [10]: $a=0.988$, $b=2.249$. A time cutoff of 50 sh is used. Twenty million histories are simulated for all problems.

IV. NEW CAPABILITIES IN MCATK

In this section, we provide an overview of features that were implemented since the last M&C meeting in 2015.

1. Expanded Source Modeling

Several new source specifications are now available in MCATK. Previously, sources could only be defined on a space-energy grid, with uniform sampling in each space-energy cell, or as point, monoenergetic sources that were either monodirectional or isotropic in angle. For more complicated sources, users were required to write their own custom sampling functions in C++. Any number of custom sampling functions can be assigned to a particular source. Although custom sampling is still allowed, the use of this feature requires sufficient knowledge of C++, which may be a barrier to some users.

Now, many additional source definitions are implemented in MCATK (see Table I) and source customization is no longer needed except for very advanced sources. Sources are created using a standardized interface for specifying independent space, energy, angle, and time distributions. Defining non-standard sources, e.g., a source for which separate particle attributes are dependent on one another, is done by writing a custom sampling function and assigning it to a source. The interface of the “Source” object makes this function assignment quite simple.

The new source specifications were applied to the test problem defined in Section III. The problem was run using both MCNP and MCATK. The MCATK source uses the histogram angle, Watt energy, disc space, and burst time distributions. Except for the burst time distribution, these distributions are all new. The neutron fluxes in each region are compared in Table II. We see that the codes are in good agreement, meaning that the relative errors are nearly identical, and fluxes lie within ~1-3 combined relative errors. Therefore, we conclude that the source specification in MCATK is functioning correctly.

Table I. Available Source Definitions in MCATK.

	Old Sources	New Sources
Angle	Monodirectional Isotropic	Monodirectional
		Isotropic
Energy	Monoenergetic Histogram	Histogram
		Linear
		Monoenergetic
		Discrete Lines
		Gaussian
Space	Point Histogram	Maxwellian
		Watt
		Point
		Histogram
		Solid Body Node
		Line
		Parallelogram
		Right Parallelepiped
Time	Burst Histogram Uniform	Disc (Hollow/Solid)
		Annulus/Cylinder
		Spherical Shell
Misc.	Spontaneous Fission	Burst Histogram Uniform
	Custom	Spontaneous Fission Custom

Table II. Comparison of Neutron Fluxes [# /cm²] Using New Source Definitions

Region	Energy < 1.0 MeV		Energy > 1.0 MeV	
	MCATK (Avg. +/- Rel. Err.)	MCNP (Avg. +/- Rel. Err.)	MCATK (Avg. +/- Rel. Err.)	MCNP (Avg. +/- Rel. Err.)
1	3.1320E-5 +/- 0.0002	3.1330E-5 +/- 0.0002	1.9828E-5 +/- 0.0002	1.9828E-5 +/- 0.0002
2	1.8355E-5 +/- 0.0003	1.8364E-5 +/- 0.0003	9.8861E-6 +/- 0.0004	9.8768E-6 +/- 0.0004
3	7.9290E-6 +/- 0.0006	7.9176E-6 +/- 0.0006	4.7078E-6 +/- 0.0007	4.6987E-6 +/- 0.0007
4	3.4440E-6 +/- 0.0009	3.4346E-6 +/- 0.0009	2.2505E-6 +/- 0.0010	2.2405E-6 +/- 0.0010
5	1.5432E-6 +/- 0.0014	1.5441E-6 +/- 0.0014	1.0931E-6 +/- 0.0015	1.0878E-6 +/- 0.0015
6	7.1419E-7 +/- 0.0020	7.1242E-7 +/- 0.0020	5.3997E-7 +/- 0.0021	5.3799E-7 +/- 0.0021
7	3.3807E-7 +/- 0.0029	3.3738E-7 +/- 0.0029	2.7225E-7 +/- 0.0030	2.7099E-7 +/- 0.0030
8	1.6493E-7 +/- 0.0042	1.6325E-7 +/- 0.0042	1.4001E-7 +/- 0.0042	1.3923E-7 +/- 0.0042
9	8.0769E-8 +/- 0.0060	8.0950E-8 +/- 0.0060	7.2377E-8 +/- 0.0059	7.2947E-8 +/- 0.0059
10	3.5830E-8 +/- 0.0089	3.6156E-8 +/- 0.0088	3.7845E-8 +/- 0.0080	3.7343E-8 +/- 0.0080

Table III. Comparison of Photon Fluxes [#/cm²] Using Coupled Neutron-Photon Transport

Region	Energy < 1.0 MeV		Energy > 1.0 MeV	
	MCATK (Avg. +/- Rel. Err.)	MCNP (Avg. +/- Rel. Err.)	MCATK (Avg. +/- Rel. Err.)	MCNP (Avg. +/- Rel. Err.)
1	6.5252E-08 +/- 0.0096	6.5412E-08 +/- 0.0097	1.0633E-07 +/- 0.0072	1.0577E-07 +/- 0.0073
2	8.8587E-08 +/- 0.0094	8.7564E-08 +/- 0.0093	1.0412E-07 +/- 0.0074	1.0499E-07 +/- 0.0074
3	9.1255E-08 +/- 0.0093	9.2261E-08 +/- 0.0093	8.2234E-08 +/- 0.0079	8.2982E-08 +/- 0.0078
4	8.5602E-08 +/- 0.0095	8.5845E-08 +/- 0.0095	6.3055E-08 +/- 0.0087	6.2141E-08 +/- 0.0085
5	7.6041E-08 +/- 0.0101	7.5787E-08 +/- 0.0101	4.7702E-08 +/- 0.0095	4.7099E-08 +/- 0.0095
6	6.6483E-08 +/- 0.0108	6.5196E-08 +/- 0.0108	3.6035E-08 +/- 0.0103	3.6006E-08 +/- 0.0104
7	5.3724E-08 +/- 0.0118	5.3961E-08 +/- 0.0118	2.7916E-08 +/- 0.0113	2.7212E-08 +/- 0.0111
8	4.2565E-08 +/- 0.0132	4.3146E-08 +/- 0.0131	2.1627E-08 +/- 0.0123	2.1297E-08 +/- 0.0123
9	3.1947E-08 +/- 0.0147	3.2387E-08 +/- 0.0148	1.6994E-08 +/- 0.0136	1.6607E-08 +/- 0.0134
10	2.0456E-08 +/- 0.0162	1.9796E-08 +/- 0.0164	1.3222E-08 +/- 0.0145	1.2805E-08 +/- 0.0144

2. Photon Transport

Coupled neutron-photon transport is now possible in MCATK. Most nuclear data evaluations do not contain correlated photon emission information. As a result, photons produced from neutron reactions are sampled independently of the actual neutron interaction that occurs. Thus, photons generated from neutron interactions are not correlated to the exiting state of the neutron, and indeed may be created even when the sampled neutron interaction does not create any photons. This treatment, which is standard, is correct in aggregate, but incorrect on a microscopic level.

The four dominant photoatomic interactions are simulated by MCATK: incoherent scattering, coherent scattering, photoelectric interactions, and pair-production interactions. Photonuclear interactions are not modeled at this time. Additionally, electron transport is not yet possible in MCATK, but is planned for future work. As a result, photoelectric interactions are modeled as photon absorption, and pair-production results in the emission of two 0.511 MeV photons in opposite directions at the site of the initial interaction. No approximate treatments for the generation of bremsstrahlung photons – e.g., thick-target bremsstrahlung (TTB) - are implemented at this time.

We demonstrate coupled neutron-photon transport again using the problem from Section III. The source is still a neutron source, so photons are created only through neutron interactions. In order to compare the codes, the physics not yet available in MCATK must be turned off in MCNP. The disabled physics are: Compton Doppler broadening, fluorescence from photoelectric events, and the TTB approximation. Photon fluxes for both codes are shown in Table III. Again, the agreement is good. A comparison of event logs (not shown) shows that the number of (n,xg), photoelectric, pair-production, and photon leakage events are all within 0.6% between the two codes. The number of scatter events is not written to the MCNP event log and cannot be compared. We conclude that photon transport is implemented correctly in MCATK.

3. Tally Capabilities: Redesign and Expansion of Available Tally Types

Previously, the only tallies available were integral quantities (e.g., eigenvalues, system multiplication, probability of initiation, etc.) and track-length estimates of the flux on a space-energy grid. While the energy grid was arbitrary, the space grid could only correspond to the geometry mesh, meaning that fluxes could only be tallied for problems modeled using a mesh geometry.

The track-length tally was completely redesigned, and makes use of the filters and scores concepts used by the MC21 and OpenMC codes [11-12]. Filters are a means of subdividing tallies into bins (e.g., energy bins). Currently, the available filters enable binning of tallies by time, energy, and space. The tallies can be binned in space by their transport mesh cell index for mesh geometries, by their geometry node index for solid body geometries, or by their cell index in an independent spatial grid overlaid on the geometry. Other filters will be implemented in the future. The flux tallies in Sections IV.1 and IV.2 made use of the redesigned track length tallies using a spatial filter by geometry node index and an energy filter with two bins.

Scores are similar to the concept of tally multipliers in MCNP, and are a way of tallying quantities that are a product of some function and the flux. For example, reaction rates are obtained by tallying the product of the track-length and the desired reaction cross section. At present, scoring functions have been implemented for energy deposition, i.e., heating, and for reaction rates. More scoring functions will be implemented in the near future.

We anticipate implementing collision and surface tallies in MCATK. The concepts of filters and scores are extensible to these tally types, with the caveat that some filters and scores are only applicable to certain tally types. The user interfaces for these tallies, filters, and scores are still being refined.

A neutron heating tally was accumulated for the coupled neutron-photon test problem. The tally uses a time filter (1 sh time bins), a spatial filter (by geometry node

index) and a heating scoring function. Figure 1 shows the cumulative energy depositions from for three regions at varying distances from the source. We see good agreement between MCATK and MCNP.

The total photon reaction rate was also tallied for the coupled neutron-photon test problem (Fig. 2). This tally also uses a time filter (1 sh time bins), a spatial filter (by geometry node index) and a total cross section multiplier to obtain a reaction density. Again, the two codes agree well. There is more noise in the photon tallies because there are fewer photon events than neutron events in this problem, and because the neutron tally is cumulative with very little late-time energy deposition compared to the large initial deposition, which masks the late-time noise.

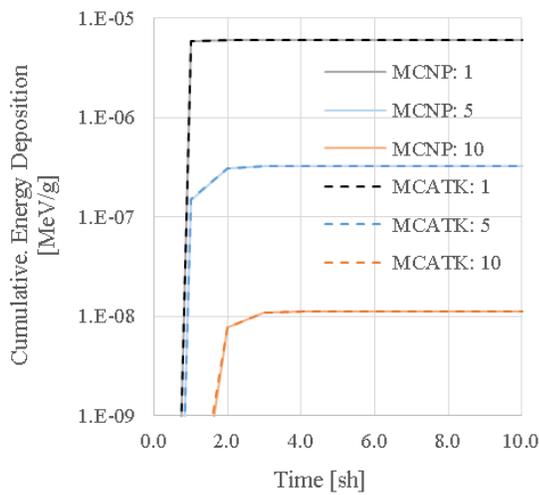


Fig 1. Cumulative energy deposition in regions 101, 105, and 110. MCATK results in darker, dashed lines. MCNP results in lighter, solid lines.

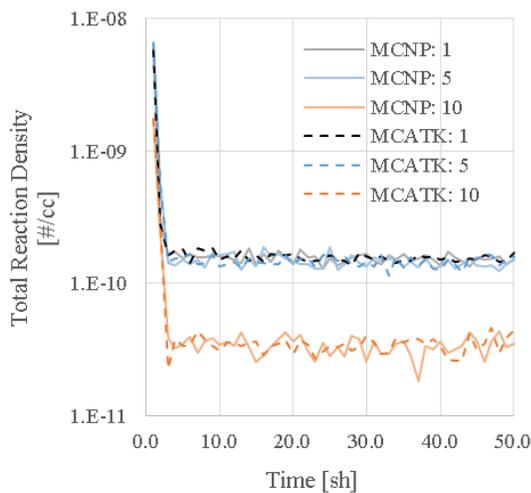


Fig 2. Total reaction density in regions 101, 105, and 110. MCATK results in darker, dashed lines. MCNP results in lighter, solid lines.

The next event estimator, also known as a point-detector, was implemented in MCATK for photons. The point-detector calculates the expected contribution to a tally at a point from every collision. It is particularly useful when tallies are desired in regions where few particle tracks occur, as the expected contribution at the location of the point detector can be calculated from events taking place far away. The implementation of a neutron next event estimator is planned, but requires the ability to calculate the probability of exiting a reaction at a particular angle, which is more complex for neutron interactions than photon interactions.

An array of next event estimators can be used to simulate a radiograph using high energy photons. A simulated radiograph can be created with or without contributions from collided particles, i.e., with or without contributions from collided particles. We use the photon next event estimator to simulate source-only (uncollided contribution only) radiographs.

The first radiograph is of the Radiographic Test Object (RTO) [13]. The RTO is an aluminum cone containing three spheres each of different material, and each containing a cylindrical void (Fig. 3). The fourth sphere located outside the cone is for orientation. Figure 4 shows a source-only (uncollided) photon radiograph of the RTO. An 1124x1124 array of next event estimators is placed 392 cm behind the object, while a point, isotropic, 14 MeV source is placed 133 cm in front of the object. The angular distribution of the source is isotropic in a cone that circumscribes the detector array.

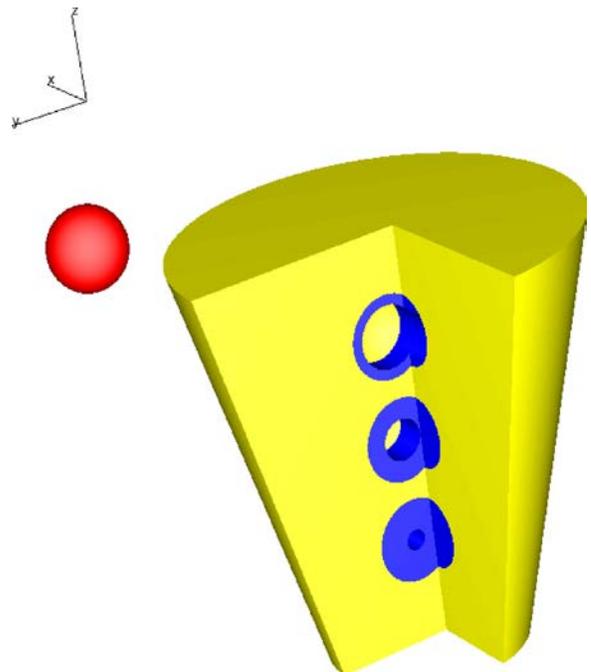


Fig 3. The RTO rendered using MCATK's rendering tool.

4. Weight Windows

Weight windows are now implemented in MCATK. Using weight windows, users can increase the number of Monte Carlo tracks in important regions, resulting in lower variance in those regions.

Users can overlay a spatial grid (spherical, cylindrical, or Cartesian) over the problem geometry and assign a lower limit of acceptable weights for particles in each grid cell. The user also specifies an upper bound factor, a maximum split factor, and a survival factor. The upper bound of the weight window is the upper bound factor times the lower bound. Particles are not allowed to split more than the maximum split factor in a single event. The weight of particles after surviving rouletting is the survival factor times the lower weight bound.

Weight window games are applied after collision events and, in time-dependent mode, are combined with population control at the end of every time step. Weight windows are not applied at surface crossing events. By contrast, the MCNP default is to apply weight windows at collisions and at surface crossings, though this behavior can be modified.

When particles have weights above the upper limit of the window, they are split into multiple lower weight particles such that the weight of the split particles is within the window (unless that would require splitting in excess of the maximum split factor). Conversely, when particles have weight less than the lower bound of the window, Russian roulette is applied to the particle. If the particle survives the roulette game, its weight is increased to the survival factor times the lower weight bound.

Weight windows were applied to the test problem from Section III. In this case, the problem is run exclusively with neutrons. The energy-integrated neutron fluxes and associated figure of merit ($FOM = 1/(\sigma^2 * \text{time})$) with and without weight windows are listed in Table IV. The weight windows were defined such that the *center* of the weight window in the first region is 1 (unchanged), and halved for each subsequent region (i.e., 1, 1/2, 1/4, ... 1/512). For consistency, MCNP was run such that weight windows were only applied after collisions, and not after surface crossings. The MCATK computation time increases from 19.58 minutes to 90.50 minutes when weight windows are activated. The MCNP computation time increases from 6.35 minutes to 20.86 minutes when weight windows are turned on. The fluxes do not change significantly when weight windows are activated, and the errors decrease deep in the cylinder as expected. Furthermore, the two codes are in good agreement. We conclude that weight windows have been implemented properly in MCATK.

We do see that the use of weight windows *decreases* the figure of merit. This result is no cause for concern; it simply indicates two things. First, the manual generation of weight windows is difficult, and typically requires iteration to obtain reliable weight windows. These weight windows were chosen more or less arbitrarily, and without iteration to

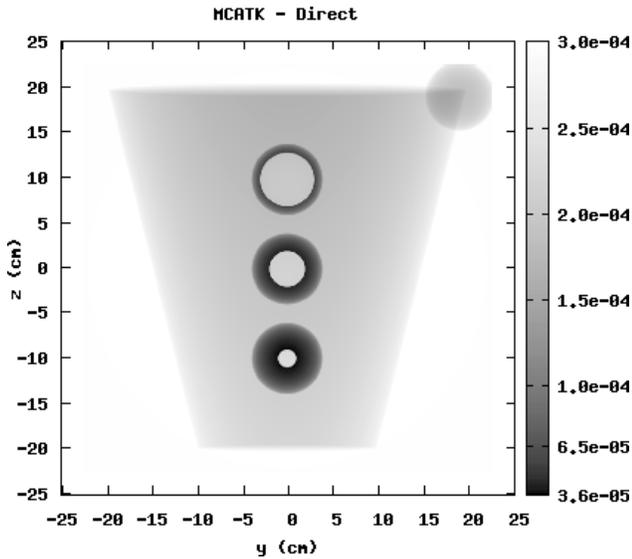


Fig 4. Source-only radiograph of the RTO generated by MCATK.

Figure 5 shows a source-only (uncollided) radiograph of the Zubal phantom model of the top portion of a human head [14-15]. The image is created with a 1000x1000 array of next event estimators placed 80 cm to the right of the center of the head. The point, 30 MeV photon source is located 900 cm to the left of the center of the head, and is isotropic within a cosine range of 0.9 to 1.0.

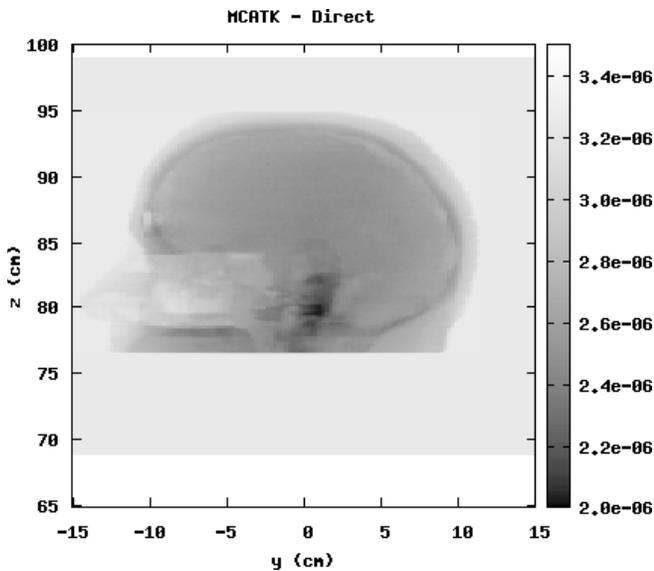


Fig 5. Source-only radiograph of the Zubal head phantom

Table IV. Comparison of Neutron Fluxes [$\#/\text{cm}^2$] With and Without Weight Windows.

Reg.	MCATK			MCNP		
	No Weight Windows (Avg. +/- Rel. Err.)	Weight Windows (Avg. +/- Rel. Err.)	Rel. FOM	No Weight Windows (Avg. +/- Rel. Err.)	Weight Windows (Avg. +/- Rel. Err.)	Rel. FOM ^a
1	5.1148E-05 +/- 0.0001	5.1149E-05 +/- 0.0001	0.2377	5.1158E-05 +/- 0.0001	5.1154E-05 +/- 0.0001	0.3
2	2.8241E-05 +/- 0.0003	2.8239E-05 +/- 0.0002	0.2532	2.8224E-05 +/- 0.0003	2.8224E-05 +/- 0.0002	0.7
3	1.2637E-05 +/- 0.0005	1.2628E-05 +/- 0.0004	0.2919	1.2619E-05 +/- 0.0005	1.2617E-05 +/- 0.0004	0.5
4	5.6945E-06 +/- 0.0007	5.6948E-06 +/- 0.0006	0.3392	5.6741E-06 +/- 0.0008	5.6772E-06 +/- 0.0006	0.5
5	2.6363E-06 +/- 0.0011	2.6377E-06 +/- 0.0008	0.3943	2.6275E-06 +/- 0.0011	2.6279E-06 +/- 0.0009	0.5
6	1.2542E-06 +/- 0.0017	1.2547E-06 +/- 0.0011	0.4560	1.2511E-06 +/- 0.0017	1.2489E-06 +/- 0.0012	0.61
7	6.1033E-07 +/- 0.0024	6.1057E-07 +/- 0.0016	0.5216	6.0929E-07 +/- 0.0024	6.0800E-07 +/- 0.0016	0.68
8	3.0494E-07 +/- 0.0034	3.0336E-07 +/- 0.0021	0.5879	3.0271E-07 +/- 0.0034	3.0264E-07 +/- 0.0021	0.80
9	1.5315E-07 +/- 0.0048	1.5335E-07 +/- 0.0028	0.6489	1.5322E-07 +/- 0.0048	1.5297E-07 +/- 0.0028	0.89
10	7.3675E-08 +/- 0.0066	7.3263E-08 +/- 0.0037	0.7054	7.3157E-08 +/- 0.0067	7.3261E-08 +/- 0.0037	1.00

^aMCNP output of relative error has a fixed number of digits (4), and so only one or two significant figures are available for the FOM. More digits were available for the MCATK tally output and FOM calculation.

improve them. A poor selection by the user, as was made here, can often result in decreased performance. For this reason, we have already begun implementation of a simple weight window generator that uses particle tracks to determine where more particles are needed. More sophisticated generators will be considered in the future. Second, the ability to apply weight windows at surface crossing events and collision events should be implemented in MCATK. It is worth noting that when MCNP is run a third time with the same weight windows, but using the default behavior of applying them at surface crossing and collision events, the FOM increases for cells far from the source.

5. Stochastic Systems Analysis: Calculation of Moments of the Neutron and Fission Number Distribution

The fission chain analysis algorithm is used to study stochastic systems (i.e., systems in which neutron behavior can vary significantly from its average behavior). The algorithm, which previously included multiplication and probability of initiation capabilities [8], was extended to calculate moments of the neutron and fission number distributions. The neutron number is the number of neutrons alive in a system at a given time, while the fission number is the number of fissions that have occurred in a system up to a particular time. In a stochastic system, a distribution of neutron and fission numbers may be observed, and this distribution cannot be sufficiently characterized by its mean alone. MCATK can now calculate the mean, variance, skewness, and kurtosis of these distributions. This novel Monte Carlo capability provides an alternative to deterministic methods for calculating the statistical moments of the neutron and fission number distributions. This new feature is presented in detail in a separate paper [9].

6. Python Interface for Setup of Material and Geometry XML Input Files

Many of the classes in the MCATK software library are designed to be serializable, that is, writable to a file (or binary data stream for message passing). Major problem components, including geometry, materials, and sources are serializable to XML formatted files. This is done automatically using the Boost.Serialization library [16]. These files can be read (deserialized) as well as written, and therefore are useful as input files to initialize a problem.

The auto-generated XML formatted files are difficult to read and more difficult to write from scratch. For this reason, it is recommended that the geometry, materials, and sources be set up using the MCATK library, and then written to XML files using the automated capabilities of the library. However, this means that, when using the MCATK C++ interface, every change to the geometry or materials requires recompiling the MCATK problem setup C++ code.

In order to streamline problem setup, a Python interface has been created using the Boost.Python library. Unlike C++, Python is an interpreted language, so no compilation is required. The geometry, materials, and sources can be set up using the Python interface, which looks nearly identical to the C++ interface, but compilation is not required every time the geometry or materials are changed.

As the designs of new tally objects in MCATK are finalized, they will be made serializable to an XML file. A new object for specifying algorithm parameters (e.g., number of cycles for a k-eigenvalue calculation) that is also serializable is currently being developed. The Python interface will be extended, enabling total problem setup, i.e., algorithms, geometry, materials, sources, and tallies, using the Python interface.

Both the Python and the original C++ interface for problem setup are designed to be immediately readable even by first time code users. For example, the python code:

```
from MaterialsMode import *
mat1 = MaterialSetupFileEntry()
mat1.setID(99);
mat1.setName("leu");
mat1.setMassFrac();
mat1.addIsotope("92235",0.05);
mat1.addIsotope("92238",0.95);
matFile = MaterialSetupFile()
matFile.add(mat1)
matFile.write("Materials.xml")
```

creates a material setup object (`mat1`), gives it an ID number (99) and a name ("leu"), specifies that all atomic fractions are to be interpreted as mass fractions, and adds two isotopes (u-235 and u-238 with 5% and 95% mass fractions respectively). A material setup file object is also created (`matFile`), the problem material(s) are added to the material setup file object, and then the material setup file object writes itself to an XML file which can later be read during problem setup.

An example of specifying a Watt energy distribution and disc spatial distribution for a source object using the MCATK Python interface might look like:

```
src1 = Source()
src1.specifyWattEnergyDistribution(0.988, 2.249)
radius = 10.0
center = Vector3D(1.0,2.0,3.0)
axis = "y-axis"
Src.specifyDiscSpatialDistribution(radius,
    base,axis)
```

whereas the MCNP source set up for the same distribution might be:

```
SDEF ERG=D1 POS=1.0 2.0 3.0 RAD=D2 AXS=0 1 0
SP1 -3 0.988 2.249
SI2 10.0
SP2 -21 1
```

where distribution -3 on the SP1 card specifies a Watt spectrum, and distribution -21 1 on the SP2 card specifies power law sampling an exponent of 1. This example is representative of the two codes: the MCNP input is more concise, but the MCATK input is more intuitive.

7. New Driver Applications

MCATK is designed to be, first and foremost, a toolkit allowing users to build customized applications. This means that MCATK provides C++ classes and methods (with either C++, C, Fortran, or Python interfaces) that are callable from custom applications or existing codes. Nevertheless, it is beneficial to have standard driver applications, both to serve as examples for setting up various problem types in MCATK, and to provide a stable platform for performing verification and validation (V&V).

Some older driver applications, `serialEigen` and `parallelEigen`, were originally written for V&V of the k-eigenvalue algorithm [17]. Although they were extended

to make use of all of the available algorithms in MCATK, the problem setups allowed by these drivers are still very limited, and they are not modular enough to provide a flexible platform for testing new features or connecting in custom components.

A new driver application, was created that has much of the functionality of the previous drivers, but with several key differences. First, it was designed from the beginning to make use of the XML input files discussed in the previous subsection. To summarize, the XML input files will define: algorithms, geometry, materials, sources, and tallies. Second, the driver uses several new features not available in the previous drivers. Third, and most importantly, this driver was written to be significantly more modular than the previous drivers.

The modularity of the new driver has several benefits. Each module, e.g., the module for source algorithm setup, provides a clean example for users hoping to setup up various components of MCATK for custom applications. Furthermore, the modular design enables simple replacement of modules for testing and customization. For example, to apply a new tally type to a realistic problem (to supplement our simple unit testing), one need only replace the `SetupTally` and `PrintTally` objects with custom alternatives, leaving all other modules untouched. Template header files are provided defining the interface for these objects, so users and developers need only write the body of the functions while conforming to the template interface.

All of the results in this paper excluding the radiographs were obtained using the new driver with various custom modules

V. SUMMARY AND FUTURE WORK

This paper provided an overview of new features implemented in MCATK since the "Advances for 2015" paper [2]. The new features discussed here are: new source specifications, coupled neutron-photon transport, new tally capabilities, weight windows, calculation of moments of the neutron population for stochastic systems, a Python interface for problem setup, and a new driver application.

Additional work remains on several of these features. A weight window generator is being implemented to mitigate the risk of choosing poor weight window bounds. Surface and collision tallies must be implemented to supplement the track length tally. The tally interface must be finalized, and all tallies must be made serializable to an XML file. The Python interface must be extended until an entire (non-customized) problem, i.e., geometry, material, source, tally, and algorithm specification, can be defined through this interface. The new driver application must be made to make use of the tally and algorithm XML input files when they become available (it already makes use of the other XML input files).

Several other features are under development, including shared-memory parallelism and on-the-fly multi-temper-

ature treatments. In addition, there are plans to implement a number of other features. These include the implementation of new physics such as thermal scattering ($S(\alpha,\beta)$) and unresolved resonance treatments.

ACKNOWLEDGMENTS

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Los Alamos National Security, LLC, for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396.

REFERENCES

1. 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).
2. J. SWEEZY, S. NOLEN, T. ADAMS, T. TRAHAN, L. PRITCHETT-SHEATS, "Monte Carlo Application ToolKit (MCATK): Advances for 2015", *Proc. M&C+SNA+MC 2015*, Nashville, Tennessee, April 19-23, 2015, American Nuclear Society (2015) (CD-ROM).
3. T. GOORLEY, et. al., "Initial MCNP6 Release Overview", *Nuclear Technology*, **180**(3), 298 (2012).
4. R.E. MACFARLANE, D.W. MUIR, "The NJOY Nuclear Data Processing System Version 91", LA-12740-M, Los Alamos National Laboratory, Los Alamos, NM (1994).
5. T.J. TRAHAN, J.E. SWEEZY, J.F. GIRON, "Verification of Solid Body Geometries in MCATK", *Proc. PHYSOR 2016*, Sun Valley, Idaho, May 1-5, 2016, American Nuclear Society (2016).
6. S. NOLEN, J. SWEEZY, T. ADAMS, "Integral Criticality Estimators in MCATK (U)", LA-UR-12-25458, Los Alamos National Laboratory (2012).
7. S. NOLEN, T. ADAMS, J. SWEEZY, A. ZUKAITIS, "Time-Dependent Tracking in the Monte Carlo Application ToolKit", *Proc. SNA+MC 2013*, Paris, France, October 27-31, 2013.
8. 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) (CD-ROM).
9. T.J. TRAHAN, "Monte Carlo Calculation of Moments of the Neutron Number and Fission Number for Stochastic Systems in MCATK", *Proc. M&C 2017*, Jeju, South Korea, April 16-20, 2017, American Nuclear Society (2017) (SUBMITTED).
10. D.B. PELOWITZ et. al., "MCNP6TM User's Manual", LA-CP-13-00634, Los Alamos National Laboratory, Los USA (2013).
11. T.M. SUTTON, et. al., "The MC21 Monte Carlo Transport Code", *Proc. M&C + SNA 2007*, Monterey, California, April 15-19, 2007, American Nuclear Society (2007) (CD-ROM).
12. P.K. ROMANO, "OpenMC: A State-of-the-Art Monte Carlo Code for Research and Development", *Ann. Nucl. Energy*, **82**, 90 (2015).
13. E. SHORES, C. SOLOMON, J. ZUMBRO, D. FLAMIG, G. TERRONES, "Radiographic Test Problem for MCNP and Other Mesh-Based Applications," LA-UR-12-24562, Los Alamos National Laboratory (2012).
14. I.G. ZUBAL, C.R. HARRELL, E.O. SMITH, A.L. SMITH, "Two dedicated software, voxel-based, anthropomorphic (torso and head) phantoms", *Proc. International Workshop, National Radiological Protection Board*, Chilton, UK, July 6-7, 1995, National Radiological Protection Board (1996).
15. J.F. EVANS, T.E. BLUE, N. GUPTA, "Absorbed dose estimates to structures of the brain and head using a high-resolution voxel-based head phantom", *Med. Phys.*, **28**(5), 780 (2001).
16. "BOOST C++ Libraries", <http://www.boost.org>.
17. J. SWEEZY, T. ADAMS, S. NOLEN, "Validation and Verification of the Monte Carlo Application ToolKit (MCATK) for Keff Eigenvalue Calculations (U)", LA-UR-13-20269, Los Alamos National Laboratory, Los Alamos, NM, (2013).