

The New MCNP6 Depletion Capability

Michael L. Fensin¹, Michael R. James¹, John S. Hendricks¹, John T. Goorley²
¹*D-5²XCP-3 MCNP Code Development Project, MS C921*
Los Alamos National Laboratory, Los Alamos, New Mexico, 87545
Tel: 505-606-0145, Fax: 505-665-2897, Email: mfensin@lanl.gov,

Abstract –*The first MCNP based inline Monte Carlo depletion capability was officially released from the Radiation Safety Information and Computational Center as MCNPX 2.6.0. Both the MCNP5 and MCNPX codes have historically provided a successful combinatorial geometry based, continuous energy, Monte Carlo radiation transport solution for advanced reactor modeling and simulation. However, due to separate development pathways, useful simulation capabilities were dispersed between both codes and not unified in a single technology. MCNP6, the next evolution in the MCNP suite of codes, now combines the capability of both simulation tools, as well as providing new advanced technology, in a single radiation transport code. We describe here the new capabilities of the MCNP6 depletion code dating from the official RSICC release MCNPX 2.6.0, reported previously, to the now current state of MCNP6. NEA/OECD benchmark results are also reported.*

The MCNP6 depletion capability enhancements beyond MCNPX 2.6.0 reported here include: (1) new performance enhancing parallel architecture that implements both shared and distributed memory constructs; (2) enhanced memory management that maximizes calculation fidelity; and (3) improved burnup physics for better nuclide prediction.

MCNP6 depletion enables complete, relatively easy-to-use depletion calculations in a single Monte Carlo code. The enhancements described here help provide a powerful capability as well as dictate a path forward for future development to improve the usefulness of the technology.

I. INTRODUCTION

Over the past several years, there have been several publications on Monte Carlo linked depletion methods, advertising varied implementation strategies for externally linking some version of MCNP, TRIPOLI, MVP, etc. to a depletion calculator such as ORIGEN, CINDER and/or PEPIN.¹⁻¹⁰ The main reason for the continued interest in this field is the belief that by using particle simulation with combinatorial geometry and continuous energy cross sections, the Monte Carlo method will best simulate complex 3-d geometries, with exotic material combinations and highly anisotropic flux behavior, expected to be encountered in test reactors and new advanced reactor systems such as: small modular reactors (SMRs) and Generation 3+ and 4 systems.¹¹⁻¹⁴

Deterministic flux calculators have historically been the method of choice for industry inline depletion calculations.¹⁵⁻¹⁸ The deterministic method uses various approximations to discretize the phase space of the Boltzman transport equation. These approximations, such as multi-group representation of the cross section, angular

averaging (S_n or diffusion theory), and spatially approximating smooth curved surfaces with triangular or square meshes, influence the flux solution accuracy.¹⁹ Nonetheless, for industry, these approximations were (and continue to be) tuned to a plethora of operating reactor data and the computational errors were deemed to be “acceptable enough” for reactivity type calculations necessary to license a reactor (i.e. cycle length, power distribution, safety margin, etc.).¹⁵⁻¹⁸ Deterministic methods are generally computationally less expensive than the Monte Carlo method; and therefore because reactor designers may be required to run hundreds to thousands of calculations to license a core, qualified fast running deterministic methods make the most sense for typical light water reactor (LWR) core design. But what if a designer was not just interested in reactivity? What if the designer was interested in a system that did not have a large amount of experimental data for qualifying the simulation accuracy?

The Monte Carlo method is well suited for looking at “details” as the simulation process has fewer approximations during the particle transport. “Details”

represents any calculation involving high anisotropy, large streaming effects and/or when cross section fidelity is extremely important such as when computing: (a) low capture cross section high decay yield isotopes used in a material characterization for nonproliferation; (b) material combinations that result in appreciable spectra over varying significant resonances such as high burnup or advanced clad systems; and (c) fuel/reflector interface for highly leaky systems such as SMRs.^{11-14, 20} The Monte Carlo method can also be used to compliment deterministic solutions by qualifying the design space of implemented approximations in the deterministic solution technique.²¹

As mentioned before, several externally linked technologies exist for computing Monte Carlo linked depletion solutions.¹⁻¹⁰ These technologies utilize various scripts for linking a transport code to a depletion solver. In most cases the author of the script only supports development of the linking script and has no access to the codes being linked. To accommodate robustness, these scripts usually coordinate several files to generate the decks for each stage of the calculation. The coordination usually depends on a specific directory structure that may or may not be automated during installation as well as an input structure that utilizes rules that may or may not be confined to the rules of the other codes further obfuscating the typical calculation. Furthermore, flux calculations and depletion solutions for reactors involve an immense amount fidelity that is extremely data heavy (i.e. many isotopes and reactions); and therefore once the proper physics can be tallied, the real limitation is memory management and performance, which may have nothing to do with the linking script.

To best accommodate these limitations, the first MCNP based inline Monte Carlo depletion capability was officially released from the Radiation Safety Information and Computational Center as MCNPX 2.6.0.²² The capability utilized a consistent, easy-to-use and easy-to-install framework that supports the development of the link, transport and depletion solver such that physics, performance enhancements and memory management improvements are more tractable and easier to implement.

Both the MCNP5 and MCNPX codes have historically provided a successful combinatorial geometry based, continuous energy, Monte Carlo radiation transport solution for advanced reactor modeling and simulation.^{22, 23} However, due to separate development pathways, useful simulation capabilities were dispersed between both codes and not unified in a single technology (i.e. MCNPX burnup and MCNP5 Shannon entropy). MCNP6, the next evolution in the MCNP suite of codes, now combines the capability of both simulation tools, as well as providing new advanced technology, in a single radiation transport code.²⁴ We describe here the new capabilities of the MCNP6 depletion code dating from the official RSICC release, MCNPX 2.6.0, reported previously, to the now current state of MCNP6.

The MCNP6 depletion capability enhancements beyond MCNPX 2.6.0 reported here include: (1) new performance enhancing parallel architecture that implements both shared and distributed memory constructs; (2) enhanced memory management that maximizes calculation fidelity; and (3) improved burnup physics for better nuclide prediction.

II. PARALLEL ARCHITECTURE

At the Advances in Nuclear Fuel Management conference in 2009, preliminary reactor modeling work identified that running the depletion solver in a serial loop caused the time dependent nuclide density calculation to rival computational expense of the actual transport solution.²⁵ Though CINDER90 took seconds to run, running hundreds of materials could take hours.

Eq. 1a-c displays the depletion equations:

$$\frac{dN_m(t)}{dt} = -N_m(t)\beta_m + \bar{Y}_m + \sum_{k \neq m} N_k(t)\gamma_{k \rightarrow m} \quad (\text{Eq. 1a})$$

$$\beta_m = \lambda_m + \sum_r \int \sigma_{m,r}(E)\Phi(E,t)dE \quad (\text{Eq. 1b})$$

$$\gamma_{k \rightarrow m} = \sum_{m \neq k} L_{km}\lambda_k + \sum_{m \neq k} \sum_r \int Y_{km,r}(E)\sigma_{k,r}(E)\Phi(E,t)dE \quad (\text{Eq. 1c})$$

The reaction rate term, in the destruction and creation operators, depends upon time-dependent flux, and the time-dependent flux depends upon the time-dependent number density, making these coupled equations non-linear (coupling is between isotopes). Therefore to solve these equations, we assume reaction rates are constant over a time step, leading to the destruction and creation operators being constant over a time step, making equation 1a a coupled first order differential equation with constant coefficients. The depletion solution therefore marches through updating fluxes at each time step, using time step lengths that are only as long as can be assumed that the nuclide density does not change enough to significantly alter the flux (i.e. flux shape and magnitude should not significantly change over a time step). Using these assumptions, there are no transverse leakage terms in depletion equations, and the solution depends only on the integral scalar flux in a given region. Therefore the depletion solution for each region is completely independent of any other region, making the solution very amenable to parallelization. In MCNPX 2.7.A, a distributed memory paradigm was implemented, using the

Message Pass Interface (MPI) to distribute the depletion calculation over several nodes to maximize computational performance.²⁶ Fig. 1 displays the MPI work distribution algorithm. If the user is not parallelizing the depletion calculation, a serial loop is executed over all burn regions. If the user is parallelizing the burnup calculation, the user then has two options: (1) if the user has more materials than available processors, the load is distributed evenly amongst processors (i.e. compute the range of regions between M1 and M2); (2) if the user has more available processors than regions, a single calculation is executed on each processor in which M2 is less than or equal to the number of available processors. Notice that the parallelization scheme also utilizes the master for doing useful work (1+S includes master).

```

IF (MPI) then
  M1=(1+S+CS*M)/(1+S)
  M2=(1+CS)*M/(1+S)
  IF ((1+S))>=M) then
    M1=1+CS
    M2=M1
  ENDIF
ELSE
  M1=1
  M2=M
ENDIF
    
```

Fig. 1 MPI work distribution algorithm

Because of the extreme independence of the solution method, it was hypothesized that the parallelization would result in linear speedup; however, bottlenecks were identified. Theoretically, the CINDER90 interface need only be sent interaction rates, fluxes, and atom densities (along with other variables to identify isotopes, flag predictor corrector, and compute various normalization coefficients), and the CINDER90 interface need only send out atom densities (along with other variables for computing region specific quantities). Because these reaction rate and flux arrays are large, and because a copy must be sent to each slave processor in a linear loop, for large scale calculations involving many regions, there exists a bottleneck in the send and receive procedures resulting in a “not-exactly linear” speedup in implementation. Furthermore, by only using MPI, a copy of each array, used as intent in only, is now loaded on each processor, even when several processors share a common piece of RAM (i.e. a node containing 4 processors can share one piece of common RAM). This wasted memory usage can limit the amount of fidelity used in a calculation (i.e. less memory available for using more burnable regions).

To limit the bottleneck, we could have chosen to use tree collection procedures available in MPI-2 for parallelizing the collection; however, we would have still have been stuck with the wasted memory allocation

problem. A combination of MPI and threading was already available in MCNP5 for regular transport calculations, utilizing MPI with OPENMP.²³ Therefore in MCNP6 we chose to also implement this paradigm for parallelizing the burnup calculation. A collection of burnable regions is sent to a node via MPI and then those burnable regions are further threaded, using OPENMP, across the available processors. The work distribution algorithm for each thread within each node is displayed in Fig. 2. The algorithm is similar to Fig 1, except now load is distributed evenly for each node and thread.

```

IF (MPI) then
  M1=((1+S)*T+(CS*T+CT)*M)/((1+S)*T)
  M2=(1+CT+CS*T)*M/((1+S)*T)
  IF ((1+S)*T)>=M) then
    M1=1+CT+T*CS
    M2=M1
  ENDIF
ELSEIF (THREADING .AND. .NOT. MPI) then
  M1=(T+(CS*T+CT)*M)/T
  M2=M*(1+CT+CS*T)/T
  IF (M2>=M) M2=M
  IF ((1+S)*T)>=M) then
    M1=1+CT
    M2=M1
  ENDIF
ELSE
  M1=1
  M2=M
ENDIF
    
```

Fig. 2. Threading with MPI work distribution algorithm

A simple test case using 28 concentric spheres, with 28 burnable regions containing 76 total nuclides per region was executed using the single processor mode, across several threads on a single node, across several nodes on a single thread per node and a combination of shared and distributed memory across several nodes and threads per node. The settings for each case were 5000 particles per cycle, for 33 cycles skipping the first 2 cycles. Table I shows the increase in performance when using a combination of MPI and threading. Comparing the single processor case to the 1 node 8 thread case, we see a speedup of 4.88 times. The 1 node 8 thread case is also ~50% faster than the 8 node 1 thread case, which is evidence of the bottleneck in only using MPI instead of threading. The 3 node 8 thread case is ~33% faster than the 24 node 1 thread case, which is not as large a speedup as comparing the 1 node 8 thread case to the 8 node 1 thread case. Using MPI for any number of nodes initiates communication logic, which is in itself part of the bottleneck. Also included is the 3 node 1 thread case, which appears to have an almost linear speedup (actual linear speedup would be 3.0); however, the 8 node 1 thread case definitely does not have linear speedup as more communication is involved to reach more of the slaves. Because the burnup calculations are independent between

regions, large arrays passed in by MPI can all be made THREADSHARED and therefore do not require further superfluous copying on the shared RAM. The threading improves computational performance by: (1) decreasing the amount of distributed memory sends which decreases the computational expense of the main bottleneck (sending information to and from threads is much faster than communicating to separate distributed memory space); and (2) decreasing the amount of needed memory at a slave.

TABLE I

Computational Speed from Distributed and Shared Memory

Nodes	Threads	Computational Speedup*
1	1	na
1	8	7.66
8	1	4.88
3	1	2.28
24	1	9.00
3	8	13.38

* Single Processor = C; Test =A; Speedup = C/A.

III. MEMORY MANAGEMENT

The initial purpose of the MCNPX code was to combine MCNP4B and the LAHET 2.8 codes, to transport all particles and all energies, in support of the Accelerator Production of Tritium (APT) project.²⁷ Because tabular ENDF/B data did not exist in the higher (>100 MeV) regime, the MCNPX code implemented physics models, which use various event estimator codes, to predict interaction rates at high energies.²⁷ Because MCNPX offered the ability to mix and match tabular data with physics models, such that a particle could be simulated at any energy, the arrays associated with these auxiliary event estimator codes (as well as interface arrays used to communicate with auxiliary codes) were allocated regardless of whether they were needed or not.

Furthermore, during transport secondary particles may be created from inelastic reactions, banked, and then transported (if the particle is present on the mode card). MCNPX takes the banked particles and stores information about the particle, in arrays, such that they can be emitted at the termination of the interacting particle history. The storage array information is saved on a per initial history basis. If the amount of banked particles exceeds the size of the storage array, MCNPX would write the particle information to a file, which slows down the calculation through use of I/O. To accelerate high energy calculations, involving the creation of showers of particles per interaction per starting history (>> 1000 particles), MCNPX 2.7.C increased, by an order of magnitude, the amount of particles that could be saved in the bank per history. This adjustment was to be made statically and not

physics dependent, and therefore greatly increased the allocatable memory for storage arrays.

In a typical eigenvalue reactor calculation (mode n p), the energy of an emitted neutron is not expected to exceed 20 MeV (as $\chi(E)$ has an extremely low probability at > 20 MEV), and because the amount of secondary particles generated per history is not expected to be large, banked secondaries from neutron only transport are only generated through (n, 2n) and (n, 3n) events. It is true that the amount of banked secondaries per history can increase through use of variance reduction, such as splitting; however, in typical eigenvalue calculations, variance reduction is useless, as we are interested in computing global quantities such as k_{eff} or reaction rates in every region. Therefore, if examining isotopes containing ENDF/B transport data, there should be no reason to implement a high energy event estimator model. If simulating interactions that do not result in many banked secondaries, then the storage space for these banked events should be minimized.

In MCNPX 2.7.D, a memory reduction capability was introduced that used a combination of options on the phys:n and phys:p cards to eliminate physics model allocation as well as intelligently set banked secondary allocation based on problem dependent physics.^{28, 29} On the phys:n card, if the maximum particle energy (phys:n 1st entry) is less than the maximum energy for using tabular data (phys:n 5th entry in MCNPX, 8th entry in MCNP6), then the code will never encounter a particle energy that requires a physics model (the code will interpolate the higher energy cross section from tabular data); however, the code may still need physics models if using photonuclear physics as the code will use tabular data for nuclides with a specified extension but use models for every other nuclide. Therefore to initiate the memory reduction capability in MCNPX 2.7.D, the user had to set the 5th entry on the phys:n card greater than the 1st entry, and also turn off photonuclear physics if running both neutron and photon transport calculation (phys:p 4th entry, which is off by default). MCNP6 includes the capability of MCNPX 2.7.D as well as eliminating more arrays associated with non neutron photon transport (i.e. heavy ion and electron transport) if the user only transports neutrons and photons (i. e. using the settings mentioned for the MCNPX 2.7.D capability as well as setting the 2nd entry on the phys:p to zero; turning of electron generation from photons causing bremsstrahlung photon generation to be neglected). MCNP6 also expunges all reactions from the ACE libraries that are not directly used for burnup saving about ~8% of the total cross section allocation space.

A test case using 600 concentric spheres, with 600 burnable regions containing 277 total nuclides per region, was run using neutrons only to test the impact of the memory reduction capability. Table II shows the increase in memory savings comparing the base MCNPX 2.7.D capability to the MCNPX 2.7.D. memory reduction

capability and MCNP6 memory reduction capability. The memory reduction capability in MCNP6 saves nearly an order of magnitude of space that can be used to greatly increase the amount of available memory for more burnable regions.

TABLE II

Memory Savings from Memory Reduction Capability.

Case	RAM usage during runtime		% Savings*
	[GB]*	[GB]	
MCNPX 2.7.D	3.80	na	na
MCNPX 2.7.D M	0.78	3.02	79.47%
MCNP6 M	0.43	3.37	88.68%

M = Memory Reduction Option turned on

* During runtime = after cross section processing (xact)

* (Calculated/Measured-1)*100

IV. BURNUP PHYSICS ENHANCEMENTS

Three burnup physics enhancements were incorporated into MCNPX 2.7.0, and thus also in MCNP6, since the release of MCNPX 2.6.0.³⁰ These enhancements include: (1) lowering the thermal fission cutoff upper band limit to 1 eV for assessing burn region energy dependent fission yield; (2) using actual (n, γ) instead of summed capture for computing (n, γ) collision rates for CINDER90; and (3) correcting isomer branching based upon a combination of continuous energy integrated (n, γ) from MCNP and computed 63-group energy integrated (n, γ^*) from CINDER90.

In MCNPX 2.6.B a capability was introduced to select a burn region dependent thermal, fast or high energy spectra based fission yield for CINDER90.^{31, 32} The fission yields in CINDER90 were based from ENDF/B VI.0 and therefore thought to best represent a thermal reactor, fast reactor and fusion spectra. Initially, the energy bounds were set at 1 MeV and 14 MeV (if below 1 MeV use thermal; if between 1 and 14 MeV use fast, if greater than 14 MeV use high energy). The bounds were arbitrarily set to these values to capture the minor amount of fission events in a thermal reactor occurring between 1 eV to 1 MeV; however, when modeling epithermal systems, where using the fast yields is more correct, this approximation fails. Therefore in MCNPX 2.7.D the thermal cutoff was lowered to 1 eV.

MCNPX automatically computes the total absorption reaction (not including fission) during each track traverse and collision and stores this information for accelerating reaction sampling. Initially, the burn capability attempted to approximate the (n, γ) using total capture in order to accelerate looking up these reactions during burnup reaction tracking in transport. This approximation is

usually correct for most heavier nuclides as (n, γ) dominates all capture reactions by orders of magnitude; however, for light nuclides such as B-10 the dominant reaction can be (n, α) (or other capture events like (n, p) , (n, t) , etc.), and therefore this approximation has since been eliminated in MCNPX 2.7.D.

MCNPX 2.6.0 over predicted (n, γ) contribution because the tallied (n, γ) in MCNPX was total (n, γ) and not adjusted for isomer branching. At ICAPP 2008, it was stated that due to the energy dependent nature of the isomer branching, the future focus would be to include ENDF/B File 9 MT 102 in the ACE file and alter MCNPX to process this information.¹ Figure 3 displays the energy dependence, and fidelity, of the isomer branching to, for ENDF/B VII.0, Am-242, Am-242m, Am-244, Am-244m.³³

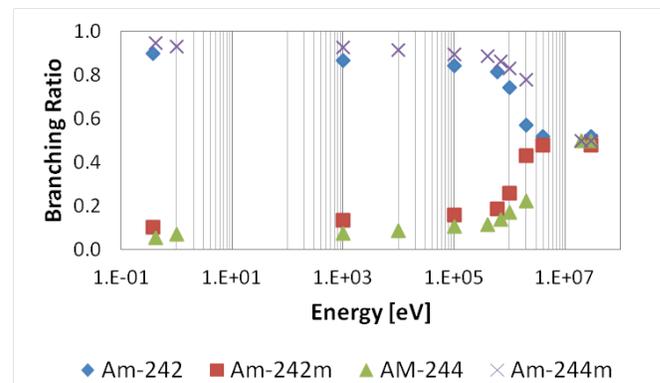


Fig. 2. Energy dependent isomer branching

The VESTA code actually does post process File 9, the isomer branching ratios, and File 10, cross sections for the production of the isomer state, to compute the actual branching based upon ENDF/B and JEFF data.³⁴ Though the isomer branching is energy dependent (changing drastically at ~ 1 MeV), the fidelity of this energy dependence in the file is actually not greater than the fidelity of the multi-group cross sections in CINDER90 (which used a combination of File 9 and File 10 “like” data to compute the 63-group cross section). Therefore in MCNPX 2.7.B, a new method was developed that leverages the 63-group (n, γ^*) reactions from CINDER90 to adjust the continuous energy integrated (n, γ) cross sections computed in MCNPX. Eq 2 displays the new method.

$$(n, \gamma)_{Corrected} = \left(1 - \frac{(\sigma_{n, \gamma^*} \Phi)_C}{(\sigma_{n, \gamma} \Phi)_M} \right) \times (\sigma_{n, \gamma} \Phi)_M \quad (\text{Eq. 2})$$

This method therefore provides energy dependence of the isomer branching without having to: (1) change the format of the ACE files and the NJOY code; (2) accommodate more storage in the cross section arrays; and

(3) increase computational expense by having to look up more information on the ACE file.

IV. H. B. ROBINSON BENCHMARK

Geometry and burnup specifications used for the H. B. Robinson benchmark were taken from the Oak Ridge National Laboratory Report, ORNL/TM-12667.³⁵ The calculation setup (i.e. time steps, boundary conditions, etc.) was taken from ref. 1. The benchmark calculation uses an infinitely reflected 15 by 15 UO₂ fueled, Zircaloy-4 clad Pressurized Water Reactor (PWR) fuel assembly. Fig. 3 shows a diagram of the computational model. In the actual calculation there is no excess water region; the outer pin cell boundary on the outer pins is the reflective surface.

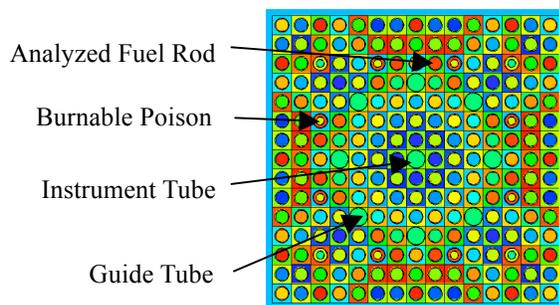


Fig. 3. H. B. Robinson infinitely reflected lattice model.

Cases A-D represents the different burnup cases from the benchmark: (1) Case A = 16.02 GWD/MTU; (2) Case B = 23.8 GWD/MTU; (3) Case C = 28.47 GWD/MTU; (4) Case D = 31.66 GWD/MTU. MCNP6 is compared to best available results from SCALE/SAS2H, MCNPX 2.6.0 and MONTEBURNS.^{1, 35, 36} The results of each Case for each code are displayed in Table III-VI.

Each benchmark calculation was run using a separate set of ENDF/B (V-VII.0) cross sections generated at a separate set of temperatures using different tolerance parameters in the cross section processing codes (details of cross section generation are listed in refs. 1, 35, and 36). All MCNP6 results are representative of MCNPX 2.7.0. Thus MCNP6 in Tables III-VI represents MCNPX 2.7.0 and MCNP6; MCNPX in Tables III-VI represents MCNPX 2.6.0. At lower burnups, Cases A and B, MCNP6 does not compute U-235, U-236, Pu-239, Pu-241 and Cs-137 as well as MCNPX 2.6.0 and SCALE (results are similar to MONTEBURNS). For Case C, MCNP6 computes similar results to MONTEBURNS, which are superior to MCNPX 2.6.0 and SCALE/SAS2H; however, at higher burnups, Case D MCNP6 computes the best results for almost every isotope (except Np-237).

TABLE III

Percent Difference* between Measured and Computed Nuclide Compositions for H. B. Robinson Benchmark Case A.

Case A				
16.02 GWD/MTU				
Isotope	MCNP6	MCNPX	SCALE	MONTEBURNS
235U	3.73	0.42	0.60	2.62
236U	-3.43	-1.76	-1.50	-3.37
238U	0.06	0.12	0.10	0.17
238Pu	-2.69	-3.41	1.50	2.29
239Pu	5.59	0.27	7.00	2.01
240Pu	2.66	3.32	-1.50	4.22
241Pu	7.68	3.57	5.90	7.04
237Np	-3.23	-6.13	6.00	-2.76
99Tc	8.49	10.91	12.40	11.35
137Cs	-3.06	-1.12	0.20	-1.64

* (Calculated/Measured-1)*100

TABLE IV

Percent Difference* between Measured and Computed Nuclide Compositions for H. B. Robinson Benchmark Case B.

Case B				
23.8 GWD/MTU				
Isotope	MCNP6	MCNPX	SCALE	MONTEBURNS
235U	3.71	-0.58	1.40	4.11
236U	-2.70	-1.90	-2.20	-3.09
238U	-0.60	-0.54	-0.60	-0.53
238Pu	-4.22	-3.86	0.90	0.83
239Pu	2.50	-0.37	7.70	1.31
240Pu	1.62	0.59	-4.20	1.61
241Pu	5.44	2.82	6.00	4.97
237Np	-4.88	-7.31	5.50	-5.55
99Tc	5.70	6.76	8.60	8.34
137Cs	-2.82	-1.88	-0.80	-2.22

* (Calculated/Measured-1)*100

Because of the assumptions used in constructing the benchmark and use of different data for each calculation, one cannot easily conclude that MCNP6 is the superior technology for this specific calculation. Furthermore, in all cases, no code best predicts all isotopes. For example, in Case A, MCNP6 has not burned up enough U-235; however, MCNP6 has transmuted more U-238 resulting in more Pu-239 and Pu-241. The creation and destruction of all isotopes is dictated by spectrum and shielding of one isotope to another; therefore it is difficult to determine the specific reaction where the methods are differing. Furthermore, the difference in data or calculation setup may be generating the largest difference.

TABLE V

LA-UR 11-07032

Percent Difference* between Measured and Computed Nuclide Compositions for H. B. Robinson Benchmark Case C.

Case C				
28.47 GWD/MTU				
Isotope	MCNP6	MCNPX	SCALE	MONTEBURNS
235U	-3.27	-11.80	-4.90	-2.44
236U	1.84	3.72	2.20	1.24
238U	0.47	0.47	0.50	0.54
238Pu	-11.04	-14.72	-6.50	-7.01
239Pu	-0.64	-9.22	5.30	-1.77
240Pu	2.09	-5.42	-4.90	1.14
241Pu	-5.08	-11.03	0.50	-4.72
237Np	3.03	2.43	14.30	2.45
99Tc	11.45	9.58	14.60	14.94
137Cs	0.11	-0.38	3.90	0.70

* (Calculated/Measured-1)*100

TABLE VI

Percent Difference* between Measured and Computed Nuclide Compositions for H. B. Robinson Benchmark Case D.

Case D				
31.66 GWD/MTU				
Isotope	MCNP6	MCNPX	SCALE	MONTEBURNS
235U	-0.08	-9.66	3.30	5.98
236U	0.17	1.18	-0.40	-1.51
238U	-0.73	-0.73	-0.80	-0.89
238Pu	-8.58	-10.69	2.60	1.97
239Pu	-0.20	-8.66	12.80	6.00
240Pu	1.32	-6.52	-4.10	2.65
241Pu	-2.56	-8.79	9.10	2.71
237Np	1.58	3.08	18.40	7.91
99Tc	7.79	5.53	11.20	11.90
137Cs	-2.45	-3.09	1.50	-1.44

* (Calculated/Measured-1)*100

Using MCNP6, each actinide and Cs-137 was computed to within a few percent, and Tc-99 was computed to within 12%, which is only slightly better than the other codes. However, one can conclude that the physics updates in MCNP6 do not produce worse results; and since these physics enhancements help to better represent the actual model, these improvements should improve accuracy in more complicated calculations.

V. CONCLUSIONS

With the merger of MCNPX and MCNP5, MCNP6 is now the next evolution in the MCNP suite of codes, and the depletion capability in MCNP6 is the next generation in complete, relatively easy-to-use Monte Carlo linked burnup. The new parallel architecture, using both THREADING and MPI as compared to MPI only, offers significant speedup in burnup calculations by speeding up both particle transport and the burnup calculation. The tests presented here show speedups of 30%-50% from using a combination of THREADING and MPI as compared to using MPI alone. The new memory management capability significantly reduces the memory footprint of each burn region allowing for more burn regions per gig of RAM to improve calculation fidelity. For the simple 600 region test case mentioned in this work, memory usage was improved by nearly an order of magnitude. Finally, the new physics enhancements provide a more correct representation of the burnup physics, as compared to MCNPX 2.6.0. Calculation results of the H. B. Robinson benchmark show that SCALE/SAS2H, MCNPX 2.6.0, MONTEBURNS and MCNP6 produces similar results for 16-28 GWD/MTU burnups and MCNP6 produces superior results at 31.66 GWD/MTU. The enhancements described here help provide a powerful capability as well as dictate a path forward for future development to improve the usefulness of the technology.

VI. FUTURE WORK

The memory reduction capability eliminates 22 large dynamically allocated arrays. Over 64 subroutines/modules allocate variables in MCNP6. Therefore future work will focus on eliminating excess allocation from the rest of the MCNP6 code. Furthermore, large book keeping arrays for tracking variance reduction summary information are dimensioned by the product of number cells, nuclides per cell and number of summary reactions; therefore these tracking arrays are enormous for large problems. Since variance reduction tracking is meaningless for typical reactor eigenvalue calculations, eliminating these tracking arrays can further increase memory savings. A preliminary capability to remove these arrays was tested, and resulted in a further >200 MB of savings for the 600 burn region test case (total memory reduction savings greater than an order of magnitude). However, eliminating these arrays causes a computational hit as "if" tests are required throughout transport; therefore further testing is required before introducing this capability into a production version of MCNP6. Furthermore as problems get larger, data arrays may become so large that storing a complete array on a single node may become impractical, and future implementations of burnup may require data decomposition across several nodes. This implementation will require severe restructuring of the

code, but still should be examined to accommodate larger scale calculations.

ACKNOWLEDGMENTS

This work was supported by the DOE– NNSA – Advanced Simulation and Computing program. The authors would also like to acknowledge the help and support provided by the members of both the D-5 and XCP-3 MCNP radiation transport teams.

NOMENCLATURE

The nomenclature is listed in the order in which each variable appears:

S = total number of slave nodes

T = total number of threads per node

CS = current slave number

CT = current thread number on a node

M = number of burn regions

M1 = initial burn region in range

M2 = final burn region in range

$N_m(t)$ = time dependent isotope density of nuclide m

β_m = destruction coefficient for nuclide m

$\gamma_{k \rightarrow m}$ = creation coefficient for nuclide m from nuclide k

\bar{Y}_m = feed or removal rate

λ_m = decay constant for isotope m

$\sigma_{m,r}(E)$ = energy dependent microscopic interaction rate for nuclide m of reaction type r

$\Phi(E, t)$ = energy and time dependent flux

L_{km} = probability of isotope k decaying into nuclide m

$Y_{km,r}$ = probability of isotope k transmuting into nuclide m by inelastic reaction type r.

ϕ = neutron flux

$(n, \gamma)_{Corrected}$ = corrected capture rate

$(\sigma_{n,\gamma} \Phi)_C$ = CINDER90 isomer production rate

$(\sigma_{n,\gamma} \Phi)_M$ = MCNPX computed capture rate

REFERENCES

1. M. L. FENSIN, J. S. Hendricks, S. Anghaie, "The Enhancements and Testing for the MCNPX 2.6.0 Depletion Capability," *Nuclear Technology*, **170**, pg 68-79 (2010).
2. P. K. ROMANO, B. Forget, T. H. Newton, "Extending MCODE capabilities for Innovative Design Studies at the MITR," *Trans of ANS*, **99**, pg 659-660 (2008).
3. Y. K. LEE, B. Roesslinger and A. Tsilanizara, "Tripoli-4 Pepin Depletion Code and Its First Numerical Benchmarks for PWR High-Burnup UO2 and MOX Spent Fuel," *Proc. Mathematics and Computation, Supercomputing, Reactor Physics and Nuclear and Biological Applications*, Avignon, France, (2005).
4. W. HAECK and B. Verboomen, "An Optimum Approach to Monte Carlo Burn-Up," *Nuclear Science and Engineering*, **156**, pg. 180-196 (2007).
5. C. TIPPAYAKUL, K. Ivanov, S. Misu, "Improvements of MCOR: New Monte Carlo Depletion Code System for Fuel Assembly Reference Calculations," *PHYSOR*, Vancouver, Canada (2006).
6. W. HAECK, "VESTA User's Manual – Version 2.0.0", IRSN Report DSU/SEC/T/2008-331 Indice A, France (2009).
7. H. R. TRELLE and D. I. Poston, "User's Manual, Version 2.0 for MONTEBURNS, Version 5B," Los Alamos National Laboratory report LA-UR-99-4999 (1999).
8. G. S. CHANG, "MCWO-Linking MCNP and ORIGEN2 For Fuel Burnup Analysis," *Trans ANS Topical Meeting on Monte Carlo*, Chattanooga, TN (2005).
9. J. CETNAR, W. Gudowski, and J. Wallenius, "User Manual for Monte-Carlo Continuous Energy Burnup (MCB) Code—Version 1," draft report, Department of Nuclear and Reactor Physics, Royal Institute of Technology, Stockholm, Sweden (2005).

10. Y. NAGAYA, K. Okumura, T. Mori, "A Monte Carlo Neutron/Photon Transport Code MVP Version 2," *Trans of ANS*, **95**, pg. 662-663 (2006).
11. L. COUSIN and W. Haeck, "Validating the VESTA Depletion Interface Using Ariane Chemical Assay," *Trans of ANS*, **101**, pg 694-695 (2008).
12. S. GHRAYEB, K. Ivanov, S. Levine, E. Loewen, "Development of Monte Carlo Models to Investigate Thorium-Based Fuel in Sodium Cooled Fast Reactors," *Trans of ANS*, **99**, pg 203-204 (2008).
13. J. W. STERBENTZ, G. L. Hawkes, J. T. Maki, D. A. Petti, "Monte Carlo Depletion Calculation for the AGR-1 TRISO Particle Irridiation Test," *Trans of ANS*, **102**, pg 495-496 (2011).
14. Y. KIM and D. Hartanto, "A Physics on a LEU-Loaded Small Modular Fast Reactor," *Trans of ANS*, **105**, pg 1120-1122 (2011).
15. M. EDENIUS, K. Ekberg, B. H. Forssen, and D. Knott, "CASMO-4, A Fuel Assembly Burnup Program, User's Manual," Studsvik/SOA-95/15, Studsvik of America, Inc. (1995).
16. T. P. SHANNON, J. K. Wheeler, and G. Touvannas, "TGBLA/PANACEA and CASMO/MICROBURN analyses of GE9B/GE10 Fuel in the Quad Cities," *Trans of ANS*, **74**, pp. 287-290 (1996).
17. D. KNOTT, E. Wehlage, "Description of the LANCER02 Lattice Physics Code for Single-Assembly and Multibundle Analysis," *Nuclear Science and Engineering*, **155**, pp. 331-354 (2007).
18. H. C. HURIA and R. J. Buechel, "Recent Improvements and New Features in the Westinghouse Lattice Physics Codes," *Trans. ANS*, **72**, pp. 369-371 (1995).
19. E. E. LEWIS and W. F. Miller, Jr., *Computational Methods of Neutron Transport*, ANS, Inc., La Grange Park, Illinois (1993).
20. M. L. FENSIN, W. E. Koehler, S. J. Tobin, "MCNPX Simulation of a Passive Prompt Gamma System to be Used in a Spent Fuel Plutonium Assay Strategy," *Trans ANS*, **102**, pg 431-433 (2010).
21. F. P. ESPEL, C T.ippayakul, S. Misu, K. Ivanov, "MCOR as Reference Tool for Benchmarking Spectral Codes," *Trans ANS*, **99**, pg 700-702 (2008).
22. D. B. PELOWITZ, editor," MCNPX User's Manual Version 2.6.0," Los Alamos National Laboratory Report LA-CP-07-1473 (2008).
23. X-5 Monte Carlo Team, "MCNP -- A General Monte Carlo N-Particle Transport Code, Version 5" Los Alamos National Laboratory Report LA-CP-03-0245 (2003).
24. T. GOORLEY, M. James, T. Booth, F. Brown, J. Bull, L.J. Cox, J. Durkee, J. Elson, M. Fensin, R.A. Forster, J. Hendricks, H.G. Hughes, R. Johns, B. Kiedrowski, R. Martz, S. Mashnik, G. McKinney, D. Pelowitz, R. Prael, J. Sweezy, L. Waters, T. Wilcox, T. Zukaitis, "Initial MCNP6 Release Overview: MCNP6 version 1.0," submitted to Journal of Nuclear Technology, Los Alamos National Laboratory Report LA-UR-11-05198
25. M. L. FENSIN, S. J. Tobin, N. P. Sandoval, S. J. Thompson, M. T. Swinhoe, "A Monte Carlo Linked Depletion Spent Fuel Library for Assessing Varied Nondestructive Assay Techniques for Nuclear Safeguards," *ANFM IV*, Hilton Head Island, South Carolina (2009).
26. D. B. PELOWITZ, J. S. Hendricks, J. W. Durkee, M. R. James, M. L. Fensin, G. W. McKinney, S. G. Mashnik, L. S. Waters, "MCNPX 2.7.A Extensions," Los Alamos National Laboratory Report LA-UR-08-07182 (November 6, 2008).
27. H. G. HUGHES, L. S. Waters, K. J. Adams, M. B. Chadwick, J. C. Comly, S. C. Frankle, J. S. Hendricks, R. C. Little, R. E. Prael, P. G. Young, "MCNP - The Lahet/MCNP Code Merger," *SARE-3, Simulation of Accelerator Radiation Environments*, Tsukuba, Japan (1997).
28. M. L. FENSIN, "MCNPX Memory Reduction Patch—FY 2009," Los Alamos National Laboratory Report: LA-UR-09-06308 (October 2009).
29. D. B. PELOWITZ, J. W. Durkee, J. S. Elson, M. L. Fensin, J. S. Hendricks, M. R. James, Russell C. Johns, G. W. McKinney, S. G. Mashnik, J. M. Verbeke, L. S. Waters, "MCNPX 2.7.D Extensions," Los Alamos National Laboratory Report LA-UR-10-07031 (October 20, 2010).
30. D. B. PELOWITZ, editor," MCNPX User's Manual Version 2.7.0," Los Alamos National Laboratory Report LA-CP-11-00438 (2011).
31. J. S. HENDRICKS, G. W. McKinney, H. R. Trellue, J. W. Durkee, J. P. Finch, M. L. Fensin, M. R. James, D. B. Pelowitz, L. S. Waters, F. X. Gallmeier, Julian-Christophe David, "MCNPX, Version 2.6.B," Los

LA-UR 11-07032

Alamos National Laboratory Report LA-UR-06-3248
(June 1, 2006).

32. W. B. WILSON, T. R. England, E. D. Arthur, C. A. Beard, C. D. Bowman, L. N. Engel, A. Gavron, D. C. George, L. Daemen, H. G. Hughes, III, W. W. Kinnison, R. J. Labauve, D. M. Lee, H. Lichtenstein, P. W. Lisowski, D.W. Muir, A. P. Muir, A. P. Palounek, R. T. Perry, E. J. Pitcher, R. E. Prael, R. J. Russel, G. Sanders, L. S. Waters, P. G. Young and J. J. Ziocok, "Accelerator Transmutation Studies at Los Alamos with LAHET, MCNPX and CINDER90," *Proceedings of the Workshop on Simulation of Accelerator Radiation Environments*, Santa Fe, NM, USA (1993).
33. M. B. CHADWICK, et al., "ENDF/B-VII.0: Next Generation Evaluated Nuclear Data Library for Nuclear Science and Technology", *Nuclear Data Sheets*, **107**, pg. 2931-3060 (2006).
34. W. HAECK, B. Cochet, L. Aguiar, "Isomeric Branching Ratio Treatment for Neutron-Induced Reactions," *Trans ANS*, **103**, pg 693-695 (2010).
35. O. W. HERMANN, S. M. Bowman, M. C. Brady, C. V. Parks, "Validation of the SCALE System for PWR Spent Fuel Isotopic Composition Analysis," Oak Ridge National Laboratory Report: ORNL/TM-12667, Oak Ridge, TN, USA (1995).
36. H. R. TRELLEUE, "Development of Monteburns: A Code that Links MCNP and ORIGEN2 in an Automated Fashion for Burnup Calculations," Los Alamos National Laboratory Report LA-13514-T (1998).