

LA-UR-26-27215

Approved for public release; distribution is unlimited.

Title: Updating the BURN Card for Version 6.4 of the MCNP Code

Author(s): Josey, Colin James

Intended for: Meeting: 2026 MCNP User Symposium, 2026-08-24 - 2026-08-26 (Los Alamos, New Mexico, UNITED STATES)

Issued: 2026-08-11



Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by Triad National Security, LLC for the National Nuclear Security Administration of U.S. Department of Energy under contract 89233218CNA000001. 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.



Updating the BURN Card for Version 6.4 of the MCNP Code

Colin Josey

MCNP User Symposium 2026

August 24-26 2026



Managed by Triad National Security, LLC for the U.S. Department of Energy's NNSA.

Aug. 24-26 2026

Summary

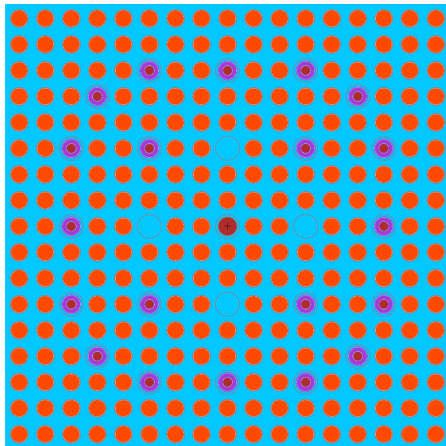
- The next version of Cinder (Cinder 2.0) is about 80% done.
- Work pivoted to rewriting MCNP's `BURN` solver to use Cinder 2.0.
- This new solver:
 - is substantially more precise.
 - uses substantially less memory.
 - has fixed all known bugs.
 - has a Python API to access and manipulate the data.

This new solver will be available in MCNP 6.4.

Precision

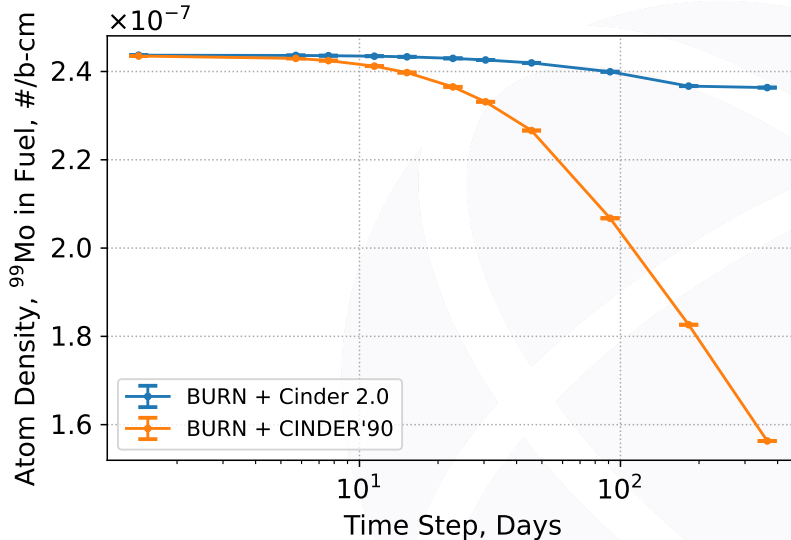
Comparisons of CINDER'90 and Cinder 2.0

- Ran single assembly from BEAVRS benchmark [1].
- Burned fuel and burnable absorbers as single region each.
- Used CINDER'90 data to keep things equal.
- Time step = $\frac{365}{2^n}$ days for 365 days.



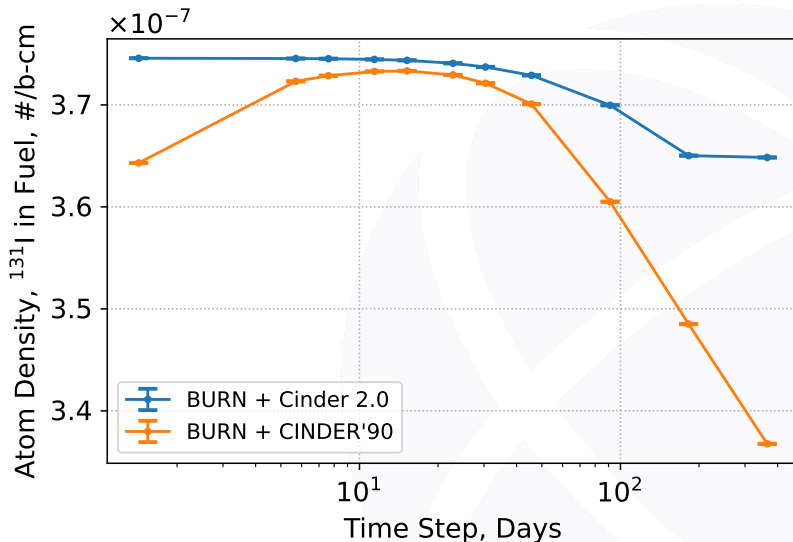
BEAVRS Assembly P09

CINDER'90 vs. Cinder 2.0 - ^{99}Mo



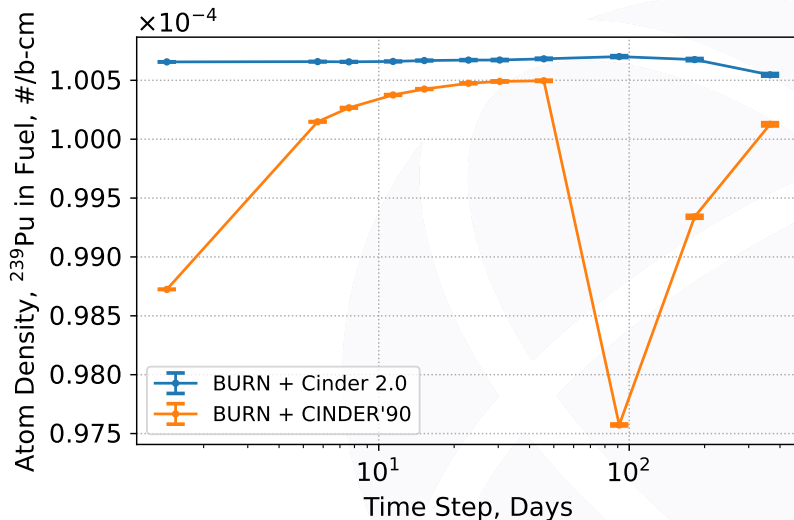
Cinder 2.0 converges much faster, but they get to the same result here.

CINDER'90 vs. Cinder 2.0 - ^{131}I



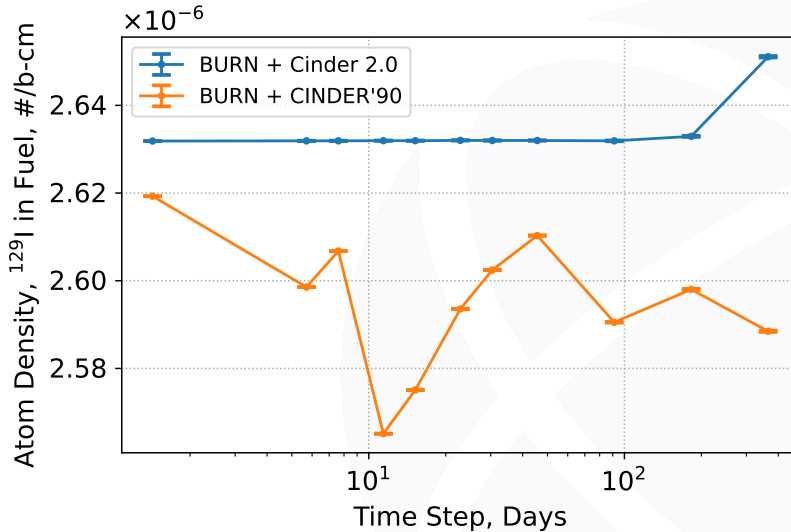
Cinder 2.0 converges, while CINDER'90 has issues as time steps get shorter.

CINDER'90 vs. Cinder 2.0 - ^{239}Pu



CINDER'90 is having issues with ^{239}Pu .

CINDER'90 vs. Cinder 2.0 - ^{129}I

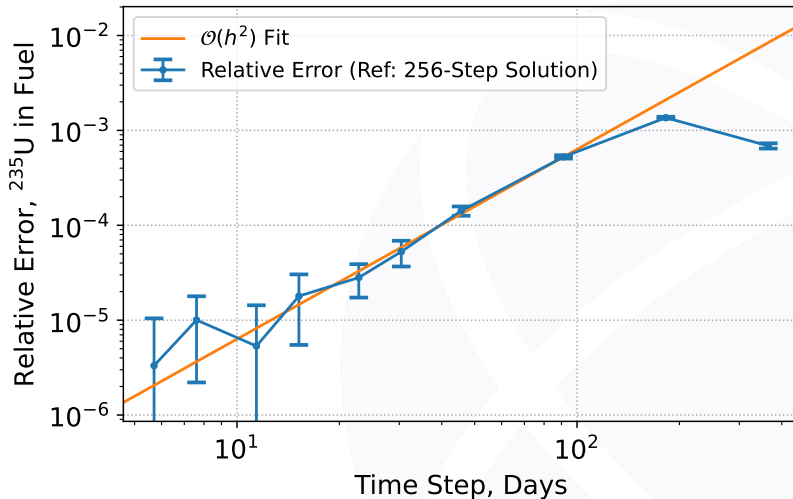


Some nuclides cause the old solver to have severe issues.

Convergence Analysis

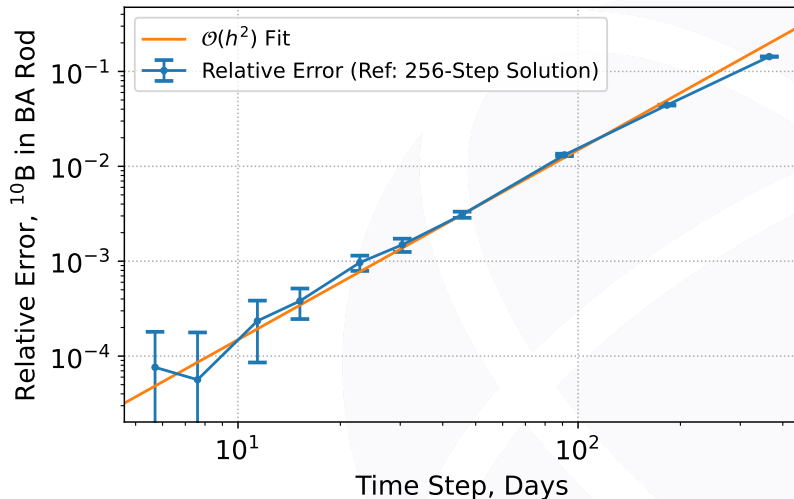
- The MCNP-Cinder coupling uses the CE/CM (Constant Extrapolate / Constant Midpoint, [2]) numerical integration scheme for now.
- This algorithm is second-order (converges $\mathcal{O}(h^2)$).
- To verify, took the 256-time-step solution as the “reference”.
(256 steps was the upper bound I could manage with the compute resources at hand, so asymptotic behavior was not observed with all nuclides.)
- Ran each simulation 50 times with different seeds to get uncertainties.

Convergence of Cinder 2.0 - ^{235}U



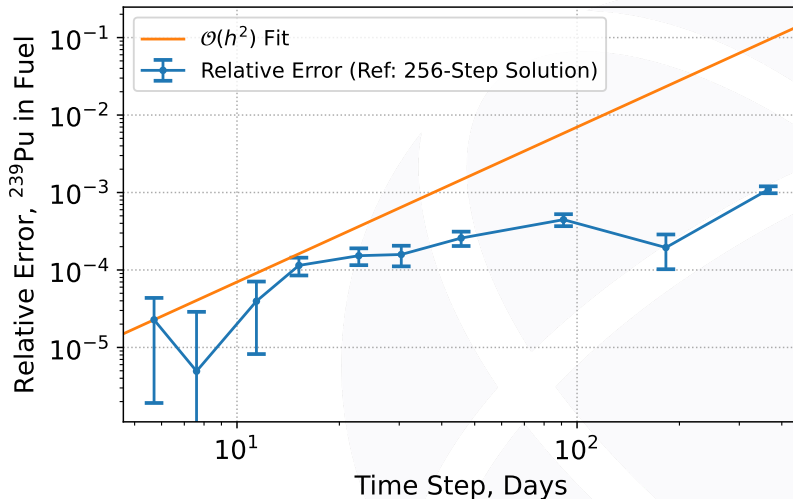
Convergence is $\mathcal{O}(h^2)$ for ^{235}U as expected.

Convergence of Cinder 2.0 - ^{10}B



Same applies to ^{10}B in the burnable absorber rod.

Convergence of Cinder 2.0 - ^{239}Pu

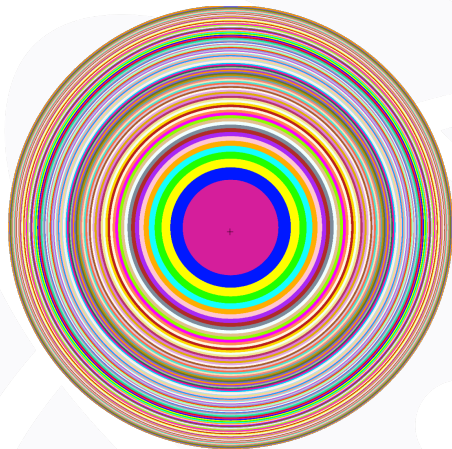


Convergence for ^{239}Pu is not yet asymptotic, but it is still converging.

Performance

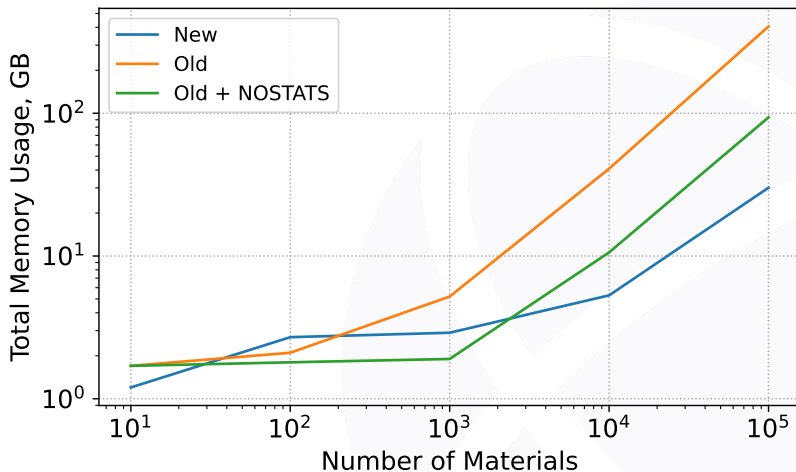
Memory Usage

- Memory usage in MCNP with many materials is problematic.
- Alongside optimization to BURN memory layout, optimizations made to other structures.
- Substantial improvement made to 6.4 relative to 6.3.1.
- Further improvements possible but not yet planned.



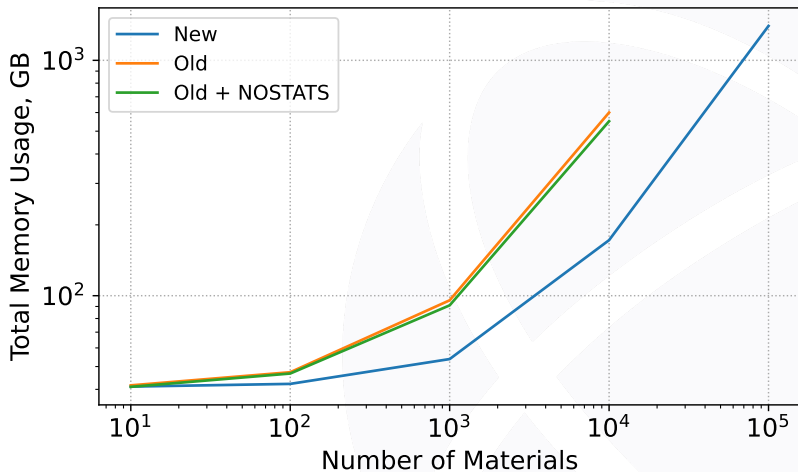
Memory Test Model: GODIVA [3]
segmented into 100 equal volume cells

Memory Usage, OpenMP



New solver takes a bit more memory for small problems. For large problems, 3-10x reduction in memory, without sacrificing reaction tally uncertainties.

Memory Usage, MPI



3.5x reduction in memory for large problems with MPI.

User Interface

New Print Table Format

Summary for all burnable materials:

step	time (days)	dt (days)	power (MW)	k-eff	flux (n/cm2-s)	burnup (GWd/tHM)
1	0.000E+00	1.825E+02	1.768E+01	1.02070	3.405E+14	0.000E+00
2	1.825E+02	1.825E+02	1.768E+01	1.01264	3.521E+14	7.629E+00
final	3.650E+02			0.99919		1.522E+01

Summary for material 7:

step	time (days)	dt (days)	power (MW)	burnup (GWd/tHM)
1	0.000E+00	1.825E+02	1.768E+01	0.000E+00
2	1.825E+02	1.825E+02	1.768E+01	7.629E+00
final	3.650E+02			1.522E+01

New Print Table Format - 2

Inventory for material 7:

No.	Isomer	Mass (g)	Activity (Ci)	Spec.Act. (Ci/g)	Atom Den. (a/b-cm)	Atom Fr.	Mass Fr.
1	U-238	4.058E+05	1.364E-01	3.362E-07	2.201E-02	3.175E-01	8.444E-01
2	O-16	5.684E+04	0.000E+00	0.000E+00	4.585E-02	6.615E-01	1.183E-01
3	U-235	7.310E+03	1.580E-02	2.162E-06	4.015E-04	5.792E-03	1.521E-02
4	Pu-239	1.865E+03	1.157E+02	6.204E-02	1.007E-04	1.453E-03	3.880E-03
5	U-236	1.044E+03	6.752E-02	6.468E-05	5.709E-05	8.237E-04	2.172E-03
6	Xe-136	4.528E+02	0.000E+00	0.000E+00	4.297E-05	6.200E-04	9.422E-04
7	Pu-240	4.233E+02	9.607E+01	2.270E-01	2.276E-05	3.284E-04	8.808E-04
8	Xe-134	2.984E+02	0.000E+00	0.000E+00	2.874E-05	4.147E-04	6.209E-04
9	Ba-138	2.575E+02	0.000E+00	0.000E+00	2.408E-05	3.475E-04	5.358E-04
10	La-139	2.438E+02	0.000E+00	0.000E+00	2.264E-05	3.266E-04	5.073E-04
11	Cs-137	2.414E+02	2.100E+04	8.698E+01	2.274E-05	3.281E-04	5.024E-04

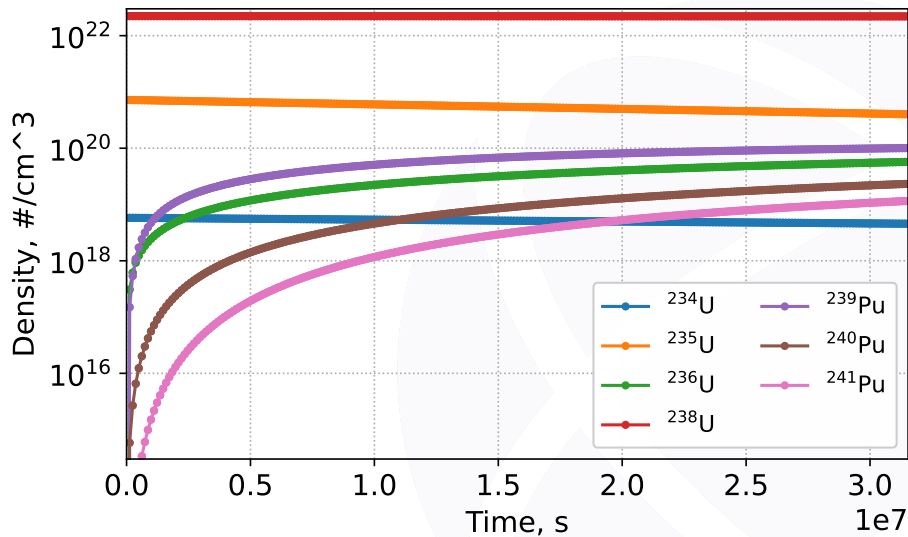
[...]

All nuclides in the decay chain can be printed, based on print settings.

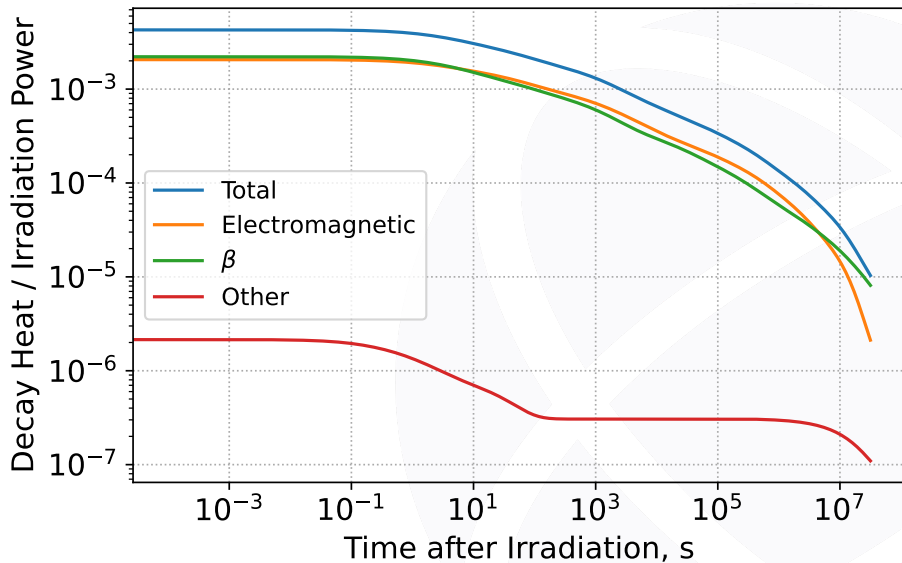
Python API (Draft) - Example

```
1 result = read_from_runtape("runtpe.h5", mat_name=1)
2
3 identifiers = [Identifier(Z=92, A=234), ...]
4
5 plt.figure()
6
7 # Plot atom densities as a function of time:
8 for identifier in identifiers:
9     nuc = result.atom_density(identifier, "edges")
10    nuc.plot(label=identifier.latex_str())
11
12    plt.xlabel(f"Time, {nuc.x_units}")
13    plt.ylabel(f"Density, {nuc.y_units}")
14
15 plt.legend()
16 plt.gca().set_yscale("log")
17 plt.show()
```

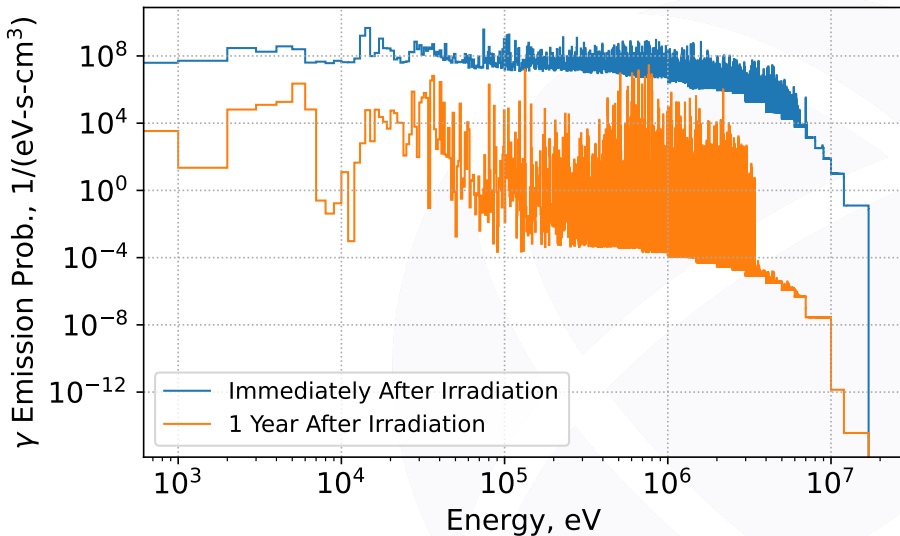
Python API (Draft) - Plot from Example



Python API (Draft) - Other Capabilities



Python API (Draft) - Other Capabilities



Python API (Draft) - MCNP SDEF

Generating a photon SDEF card from MCNP BURN output:

```
1 C Scale all tallies by 4.101610879e+18 to get units of per-second.
2 sdef par=p erg=d1065 $ Add geometry sampling here.
3 sc1065 Photon spectra from mat = 1
4 si1065 S 822 1012 829 471 375 574 988 580 [...]
5
6 [...]
7
8 sc1021 Np-239 photon emission, source = JEFF-4.0, decay = beta
9 si1021 L 0.0143 0.09955 0.10376 0.117
10 sp1021 D 0.5694448368679705 0.1279909736310227
11          0.20577276982570744 0.09679141967529936
12 sc1022 Pu-239 photon emission, source = JEFF-4.0, decay = alpha
13
14 [...]
```

The SCD special tally treatment can be used to tally contributions by each nuclide.

Other Components

- All data and results now in HDF5 format.
- Now supports selecting a data library using the `library=[...]` option.
 - Includes the C'90 library.
 - Experimenting with a JEFF-4.0-based library.
 - AARE at ORNL has been testing their own library for 100 MeV+ sources.

Future Work

- Replace MATMOD with something that is easy to use.
- Improve tallying performance through new reaction caching routines.
- Work on V&V test sets.
- Clean up and improve Python API in collaboration with other teams at LANL.
- Rewrite the ACT card to use this new infrastructure.
- After that, investigating avenues for expansion of capability.

References

- [1] N. Horelik, B. Herman, M. Ellis, S. Kumar, J. Liang, B. Forget, and K. Smith, “Benchmark for evaluation and validation of reactor simulations (BEAVRS), v2.0.2,” technical report, MIT Computational Reactor Physics Group, 2018.
- [2] A. Isotalo and V. Sahlberg, “Comparison of neutronics-depletion coupling schemes for burnup calculations,” *Nuclear Science and Engineering*, vol. 179, no. 4, pp. 434–459, 2015.
- [3] “International Criticality Safety Benchmark Evaluation Project Handbook 2015,” Tech. Rep. ICSBEP-2015, OECD Nuclear Energy Agency, Paris, France, 2015.