

Current Status and Recent Development of Serpent 2

10th International Serpent UGM

Munich, Germany, Oct. 27–39, 2020

Jaakko Leppänen, Research Professor, VTT

- ▶ New features in recent updates
- ▶ Updates 2.1.32, 2.1.33 and Serpent 2.2
- ▶ Features in 2.1.32:
 - Miscellaneous
 - History variations
 - Burnup-dependent transformations
 - Managing history and branch calculations
 - Monte Carlo based leakage correction

Major new capabilities implemented in recent updates:

- ▶ Advanced energy deposition modes (`set edepmode`)¹
- ▶ Domain decomposition (`set dd`)²
- ▶ Weight-window based variance reduction scheme (`wwin`) and a response matrix method-based importance solver (`wwgen`)³
- ▶ Sensitivity calculations
- ▶ Transient capabilities

For more information, see: Serpent Wiki, Serpent UGM 2018 & 2019

¹ R. Tuominen, V. Valtavirta, and J. Leppänen. "New energy deposition treatment in the Serpent 2 Monte Carlo transport code.". *Ann. Nucl. Energy* **129** (2019), 224–232.

² M. García and J. Leppänen. "A Collision-based Domain Decomposition scheme for large-scale depletion with the Serpent2 Monte Carlo code.". *Ann. Nucl. Energy* (submitted).

³ J. Leppänen. "Response Matrix Method–Based Importance Solver and Variance Reduction Scheme in the Serpent 2 Monte Carlo Code.". *Nucl. Technol.* **205** (2019), 1416–1432.

The last update (2.1.31) was released in May 2019.

The next update 2.1.32 is being prepared for release:

- ▶ Includes no revolutionary new capabilities, but several bug fixes and useful features.
- ▶ When released, development of major new features will be frozen for two months to collect bug reports and comments from users.
- ▶ 2.1.33 will become the final update of Serpent 2.1 (and the base version of Serpent 2.2).

The goal is to release Serpent 2.2 in early 2021:

- ▶ Single-user licenses for both RSICC and OECD/NEA.
- ▶ Distribution to some non-member countries by NEA/IAEA agreement.
- ▶ Commercial SLAs already cover 2.1.X and 2.2.X and their updates.

Additions to existing features:

- ▶ Support for continuous $S(\alpha, \beta)$ formalism for thermal scattering
- ▶ Option to write secondary photons to file (`set csw`)
- ▶ Capability to fill STL solids with universes
- ▶ Option to write STL search mesh to file to speed up initialization (`set stlfile`)
- ▶ Detector mesh transformation (`trans d`)
- ▶ Lattice transformation (`trans l`)
- ▶ Triangular lattice (type 14)
- ▶ Triangular prism surface type (tric)
- ▶ $1/k_{\text{eff}}$ scaling for nubar in transient simulations
- ▶ Variance reduction for transient simulations (branchless collision method)
- ▶ FIMA in burnup output

Additions to existing features:

- ▶ More options for CRAM solver
- ▶ Flux-limited TRC diffusion coefficients
- ▶ User-defined radial limits in for depletion zone division (`div subr`)
- ▶ General-purpose mesh definition (`datamesh`)
- ▶ New symmetry options for ADF calculation (NSEW, 60, 120)
- ▶ New options for micro-depletion (see Antti's presentation)
- ▶ New FET capabilities (see Ana's presentation)

New plotter capability:

- ▶ Command-line plotter (`-ip` command line option)
- ▶ Interactive python-based plotter (see Simppa's demo)

Reactor calculations often involve run-time changes in the input, for example:

- ▶ Reactivity control: boron concentration, control rod movement
- ▶ Fuel shuffling

Existing options:

- ▶ Nuclide iterations:

```
set iter nuc CYCLES KEFF NZAI ZAI1 ZAI2 ... [ NMAT MAT1 MAT2 ... ]
```

practical for critical boron iterations

- ▶ Restart files:

```
set rfw OPT [ FILE ]
```

```
set rfr STEP FILE
```

Input file can be modified between runs, but can be cumbersome to use.

A new option is to apply history variations:

- ▶ Based on the the branch capability (`branch card`) and a new `hisv card`
- ▶ Serpent stops the burnup calculation at a given point, applies the variation and continues to the next step

```
% --- Branch variations with the usual format:
```

```
branch Tlo stp fuel -10.045 800 % fuel temperature to 800 K
branch Thi stp fuel -10.045 1000 % fuel temperature to 1000 K
```

```
% --- History variations:
```

```
hisv
1.0 1 Tlo % burnup point, number of variations, list of variations
2.0 1 Thi
3.0 1 Tlo
```

Features in 2.1.32: Burnup-dependent transformations

For control rod movement an easier option is to apply burnup-dependent transformations:

- ▶ Can be applied to similar to regular transformations
- ▶ Burnup point provided as an additional input parameter
- ▶ New input card `transb`

```
% --- Lift control rod bank CR1 (negative entries refer to days):
```

```
transb    0.0 u CR1 0 0    0.0  
transb -164.0 u CR1 0 0   24.0  
transb -328.0 u CR1 0 0   48.0  
transb -492.0 u CR1 0 0   72.0  
transb -656.0 u CR1 0 0   96.0  
transb -819.0 u CR1 0 0  120.0
```

Group constant generation requires running assembly level calculations in such way that all fuel types, burnups and TH conditions etc. are covered.

This includes momentary changes in local operating conditions (“branches”):

- ▶ Fuel temperature (low, nominal, high)
- ▶ Coolant temperatures and/or density (low, nominal, high)
- ▶ Boron concentration (low, nominal, high)
- ▶ Control rods (inserted, withdrawn)
- ▶ ...

And variations to account for burnup-related effects (“histories”):

- ▶ Coolant temperature and/or density (inlet, average, outlet)
- ▶ Boron concentration (BOC, average, EOC)
- ▶ ...

Serpent provides the `branch` and `coef` cards to perform the momentary branch variations automatically:

- ▶ Capability to replace materials or universes, change material temperatures and densities, apply transformations, options for poison calculation
- ▶ Coefficient matrix defines the restart points and combination of branches
- ▶ Output written in a separate file (`INPUT.coe`)

Advantages and limitations:

- + Removes the need to write separate inputs for each branch.
- Separate inputs still needed for histories.
- Thousands of branches in a single history executed as a single run (no branch-level parallelization).⁴

Sufficient approach if computer capacity is limited (number of nodes < number of histories), but not the best approach for large clusters.

⁴ J. Leppänen and R. Mattila. "Validation of the Serpent-ARES Code Sequence Using the MIT BEAVRS Benchmark – HFP conditions and fuel cycle 1 simulations." Ann. Nucl. Energy 96 (2016), 324–331.

The new `casematrix` card is an extension to the old `coef` card:

- ▶ Defines restart points and a coefficient matrix similar to `coef`
- ▶ Additional entries for history variations

```
casematrix NAME
      NHIS HIS1 HIS2 ...
      NBU BU1 BU2 ...
      NBR1 BR11 BR12 ...
      NBR2 BR21 BR22 ...
      ...
```

Advantages:

- ▶ All history calculations can be defined in a single input using additional `branch` cards.
- ▶ New command line option allows running individual histories / branches for optimal parallelization:

```
sss2 -casematrix MTX HIS COE INPUT
```

```
% --- Case matrix definition:
```

```
casematrix mtx0
3 lowThis nomThis highThis           % 3 coolant temperature history variations
9 0 0.5 1 5 10 15 20 25 30           % restart at 9 burnup points
3 no_cr aic boc                       % 3 control rod branches
6 case0 case1 case2 case3 case4 case5 % 6 state point variations
```

Run a single history (-1 for coefficient index):

```
sss2 -omp 64 -casematrix mtx0 1 -1 pwr
sss2 -omp 64 -casematrix mtx0 2 -1 pwr
sss2 -omp 64 -casematrix mtx0 3 -1 pwr
```

Run a single branch:

```
sss2 -omp 64 -casematrix mtx0 1 1 pwr
sss2 -omp 64 -casematrix mtx0 1 2 pwr
sss2 -omp 64 -casematrix mtx0 1 3 pwr
```

Features in 2.1.32: Managing history and branch calculations

To submit history calculations in a cluster:

```
#!/bin/bash
#SBATCH --cpus-per-task=64
#SBATCH --partition=all
#SBATCH --array=1-3

case $SLURM_ARRAY_TASK_ID in
    1) ARGS='1 -1 pwr';;
    2) ARGS='2 -1 pwr';;
    3) ARGS='3 -1 pwr';;
esac

srun sss2 -omp $SLURM_CPUS_PER_TASK -casematrix mtx0 $ARGS
```

Features in 2.1.32: Managing history and branch calculations

To submit branch calculations in a cluster:

```
#!/bin/bash
#SBATCH --cpus-per-task=32
#SBATCH --partition=all
#SBATCH --array=1-486

case $SLURM_ARRAY_TASK_ID in
    1) ARGS='1 1 pwr';;
    2) ARGS='1 2 pwr';;
    3) ARGS='1 3 pwr';;
    ...
    486) ARGS='3 162 pwr';;
esac

srun sss2 -omp $SLURM_CPUS_PER_TASK -casematrix mtx0 $ARGS
```

Infinite-lattice calculations (e.g. group constant generation) are subject to a fundamental problem:

- ▶ When the assembly is surrounded by an infinite lattice of identical copies, the system can be far from critical.
- ▶ The k -eigenvalue criticality source simulation forces the system to criticality by adjusting the fission number.
- ▶ This creates a bias in the solution, which is also reflected in the homogenized group constants.

The conventional approach for deterministic codes is to apply a leakage correction:

- ▶ Solution of a multi-group FM-equation for critical buckling:

$$\Sigma_{t,g}\phi_g + D_g B^2 \phi_g = \chi_g + \sum_{g'} \Sigma_{s,g' \rightarrow g}^0 \phi_{g'} \quad (1)$$

- ▶ Use critical multi-group spectrum to collapse multi-group data to few-group structure.

Similar approach is available in Serpent (`set fum`) option for calculating group constants in the critical spectrum.

Alternative Monte Carlo based approach:

- ▶ Interpret the buckling-term in:

$$\Sigma_{t,g}\phi_g + D_g B^2 \phi_g = \chi_g + \sum_{g'} \Sigma_{s,g' \rightarrow g}^0 \phi_{g'}$$

as a cross section:

$$\Sigma_L(E) = D(E)|B^2| \quad (2)$$

that corresponds to an additional loss ($B^2 > 0$) or production ($B^2 < 0$) reaction.

- ▶ The multi-group diffusion coefficient $D_g \approx D(E)$ is obtained using the CMM method, and the value of B^2 is iterated to achieve criticality.

Advantages:

- ▶ FM equation is solved using continuous energy Monte Carlo (albeit using a multi-group evaluation for the leakage cross section).
- ▶ The leakage correction is applied to everything, including transmutation cross sections used for burnup calculation.

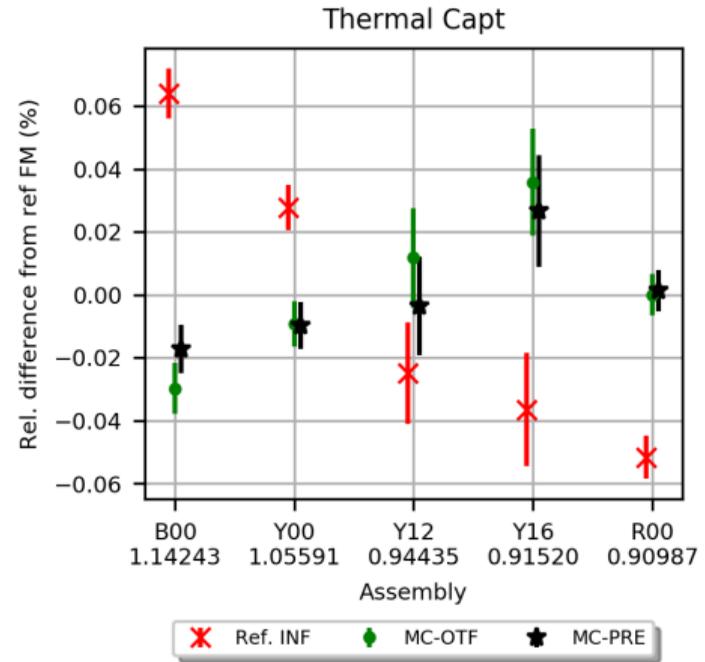
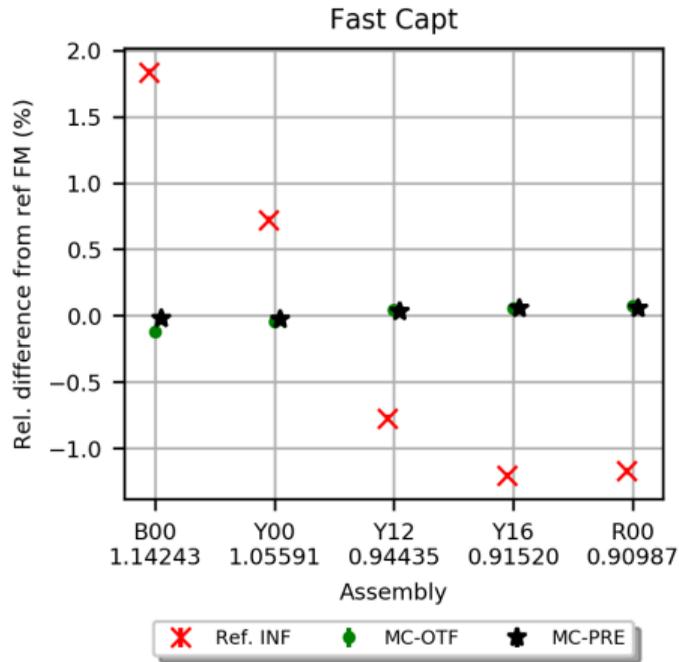
Features in 2.1.32: Monte Carlo based leakage correction

Test case: BEAVRS benchmark⁵ assembly calculations, comparison of critical bucklings.

	B00	Y00	Y12	Y16	R00
k_{∞} ref.	1.14255	1.05583	0.94436	0.91536	0.90992
B^2 ref.	2.4145E-03	9.5698E-04	-9.7421E-04	-1.4919E-03	-1.5825E-03
B^2 MC-PRE	2.3834E-03	9.4839E-04	-9.6870E-04	-1.4885E-03	-1.5765E-03
B^2 MC-OTF	2.4534E-03	9.5963E-04	-9.6861E-04	-1.4915E-03	-1.5757E-03

⁵ N. Horelik and B. Herman. "Benchmark for Evaluation and Validation of Reactor Simulations.". MIT Computational Reactor Physics Group. 2012.

Features in 2.1.32: Monte Carlo based leakage correction



Fast (**left**) and thermal (**right**) capture cross section evaluated by the different approaches.

The end – thank you for your attention!

Jaakko.Leppanen@vtt.fi

<http://montecarlo.vtt.fi>