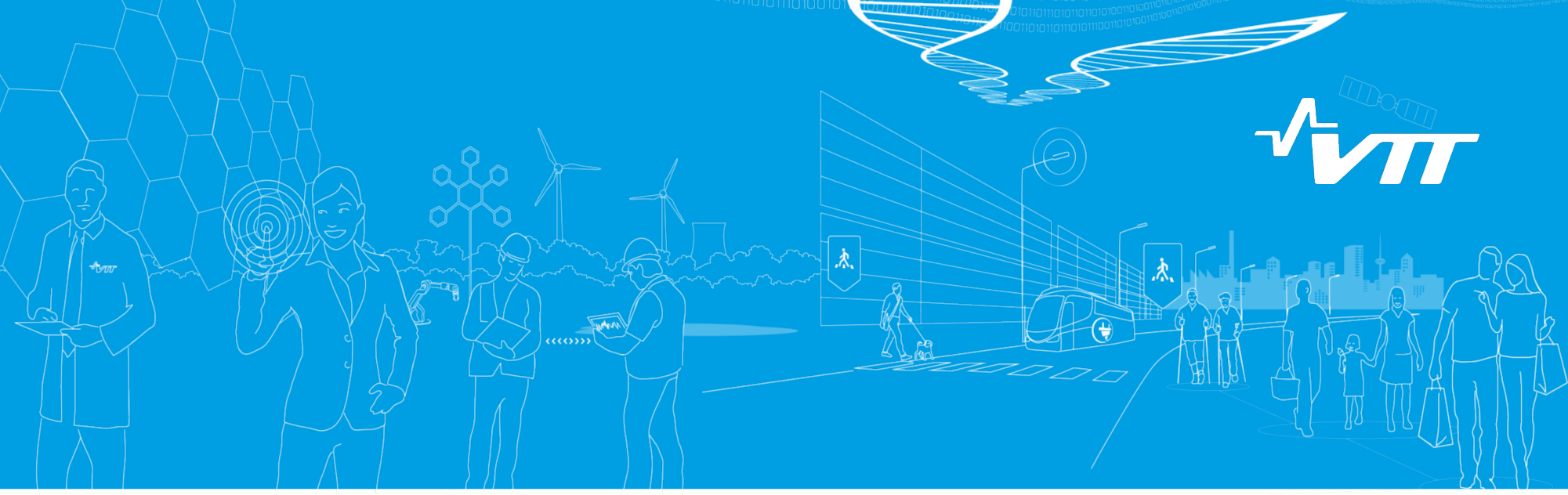


Automated calculation sequence for group constant generation in Serpent 2

4th International Serpent UGM, Cambridge, UK, Sept. 17-19, 2014

Jaakko Leppänen

VTT Technical Research Center of Finland

Outline

- What is spatial homogenization?
- Homogenization methods in Serpent 2
- Covering all state points
- Automated calculation sequence in Serpent 2
- Example case: Serpent-ARES code sequence
- Estimation of computational cost
- Summary and conclusions

What is spatial homogenization?

Modeling of an operating nuclear reactor is a complicated task:

- Full-scale solution to neutron transport problem is computationally expensive
- Transport problem becomes non-linear when feedbacks from material temperatures and densities are taken into account
- Same applies to changes in fuel composition with increasing burnup

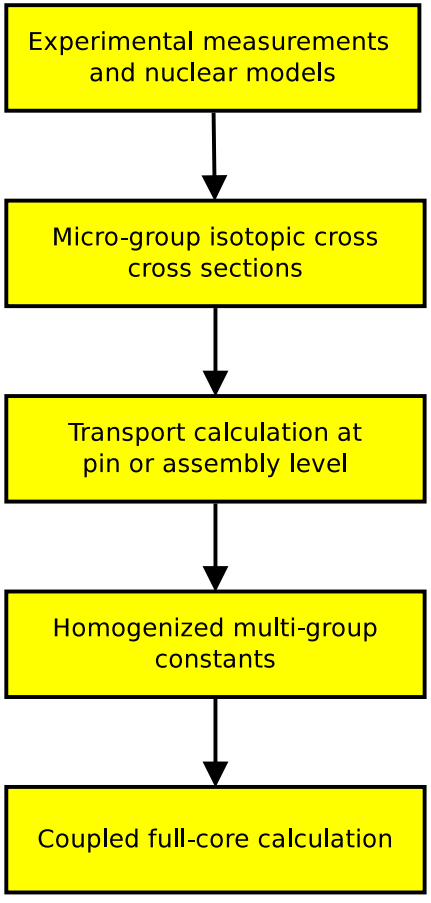
In practice, the solution to the coupled problem is obtained by iteration between:

- Solution of linearized transport problem
- Solution of heat transfer and coolant flow
- Solution of Bateman depletion equations

There exists high-fidelity solution methods for the independent problems (e.g. continuous-energy Monte Carlo and CFD), which can be combined for the solution of the coupled problem, but the approach is not practical for routine design and safety analyses

What is spatial homogenization?

Instead, the solution is obtained by gradually increasing the scale of the system, while simplifying the physics. The calculation sequence essentially consists of two parts:



i) Spatial homogenization:

- Interaction physics at the fuel assembly level is condensed into a set of representative group constants
- Geometry is homogenized, energy dependence condensed into few energy groups
- Local reaction rate balance is preserved
- Traditionally based on 2D transport calculation

ii) Full-core calculation:

- Group constants from spatial homogenization are used as the building blocks for a simplified full-core calculation
- In LWR applications typically based two-group nodal diffusion methods
- Neutronics solution is obtained at an acceptable computational cost, which enables iterative solution to the coupled problem

What is spatial homogenization?

Traditional methods for spatial homogenization:

- Deterministic transport solution, for example, MOC or CP
- Two-dimensional infinite-lattice geometry

Advantages of using continuous-energy Monte Carlo for homogenization:

- No major approximation for geometry or physics
- Straightforward, no intermediate solutions to account for self-shielding effects
- Inherently three-dimensional
- Same code and cross section data can be used for any fuel or reactor type

Serpent is one of the first Monte Carlo codes designed from the beginning for the purpose of spatial homogenization

Homogenization methods in Serpent 2

Serpent 2 has the capability to produce homogenized multi-group constants for nodal diffusion reactor simulator calculations:

- Homogenized reaction cross sections
- Scattering matrices
- Diffusion coefficients
- Assembly discontinuity factors
- Form factors for pin-power reconstruction

Methods exist for the calculation of effective delayed neutron fractions, but the routines are not specifically designed for homogenization:

- Based on the iterated fission probability (IFP) method
- Calculation covers the entire geometry (not applicable to assembly colorsets)

Homogenization methods in Serpent 2

The calculation is based on the preservation of local reaction rate balance and two built-in deterministic solvers:

i) B_1 critical spectrum calculation:

- Homogenized reaction cross sections are calculated using an intermediate micro-group structure (by default WIMS 69-group structure)
- B_1-equations are formed and solved by critical buckling iteration[1]
- The result is a leakage-corrected micro-group spectrum, which is used for collapsing the cross sections into group constants using the final macro-group structure (by default 2 energy groups)

ii) Homogeneous diffusion flux solver for ADF's and form factors:

- Based on the solution of two-dimensional diffusion equation in the homogenized region, using net currents as boundary conditions
- Used when homogenization is performed for assembly colorset or reflector (non-zero net current)
- More about this in Maria's presentation

[1] E. Fridman and J. Leppänen. "*On the Use of the Serpent Monte Carlo Code for Few-Group Cross Section Generation.*" Ann. Nucl. Energy **38** (2011), 1399–1405.

Homogenization methods in Serpent 2

The methodology has been tested with different code sequences and reactor types:

- Serpent-DYN3D: VVER-440[2], HTGR[3], SFR[4]
- Serpent-PARCS: PWR[5], SFR[6], RBWR[7]
- Serpent-ARES: MIT BEAVRS Benchmark (PWR)[8]

This is not the complete list – more examples at this meeting and PHYSOR 2014

---

[2] S. Duerigen and E. Fridman. "*The Simplified P3 Approach on a Trigonal Geometry of the Nodal Reactor Code DYN3D.*" Kerntechnik **77** (2012), 226–229.

[3] S. Baier et al. "*Extension and application of the reactor dynamics code DYN3D for Block-type High Temperature Reactors.*" Nucl. Eng. Design **271** (2014), 431–436.

[4] E. Fridman and E. Shwageraus. "*Modeling of SFR Cores With Serpent-DYN3D Codes Sequence.*" Ann. Nucl. Energy **53** (2013), 354–363.

[5] M. Hursin et al. "*Comparison of Serpent and CASMO-5M for Pressurized Water Reactors Models.*" In proc. M&C 2013. Sun Valley, ID, 2013.

[6] L. Ghasabyan. "*Use of Serpent Monte Carlo Code for Development of 3D Full-Core Models of Gen-IV Fast-Spectrum Reactors and Preparation of Group Constants for Transient Analyses with PARCS/TRACE Coupled System.*" M.Sc. Thesis, Royal Institute of Technology. 2013.

[7] A. Hall et al. "*Advanced Neutronics Methods for Analysis of the RBWR-AC.*" Trans. Am. Nucl. Soc. **108** (2013), 771–774.

[8] J. Leppänen, R. Mattila, and M. Pusa. "*Validation of the Serpent-ARES Code Sequence Using the MIT BEAVRS Benchmark – Initial Core at HZP Conditions.*" Ann. Nucl. Energy **69** (2014), 212–225.

Homogenization methods in Serpent 2

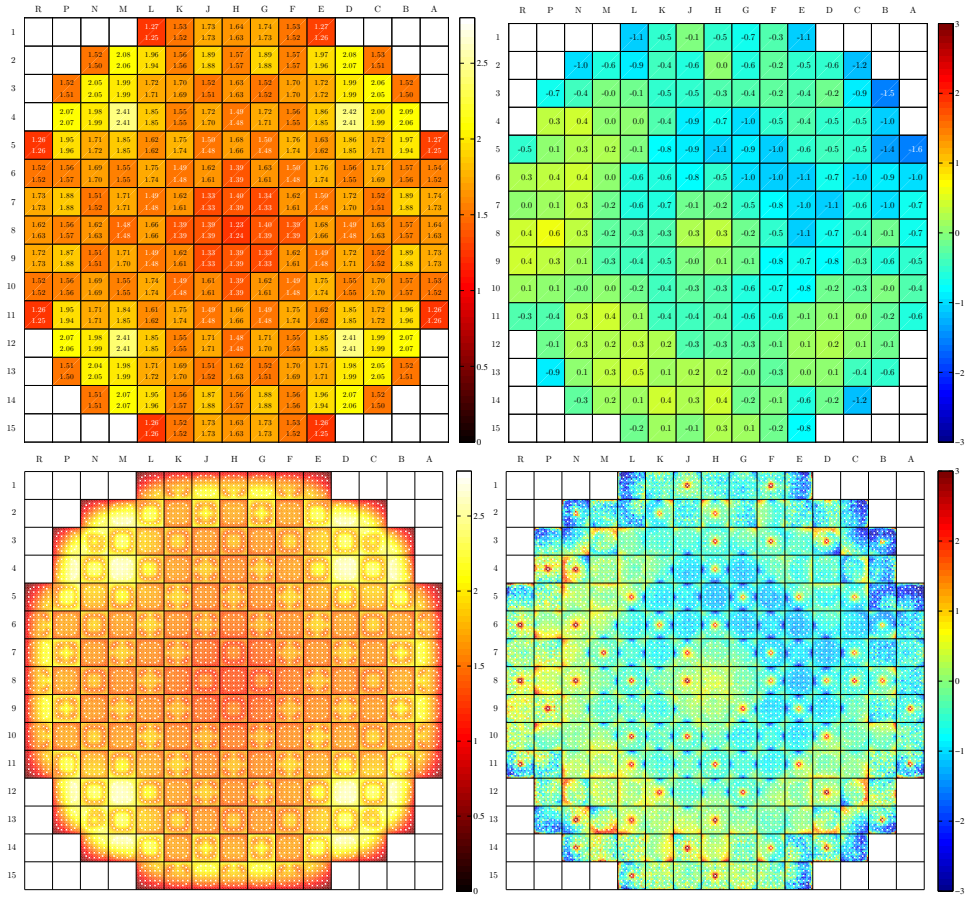


Figure 1: Results of Serpent-ARES calculations compared to Serpent 3D reference results: assembly (top row) and pin level (bottom row) power distributions (left column) and relative differences (right column).

Covering all state points

Most of the previous studies were limited to hot zero-power conditions and initial cores. In reality, spatial homogenization is complicated by the fact that group constants depend on:

- Local operating conditions: fuel temperature, moderator temperature (PWR), coolant void fraction (BWR), boron concentration (PWR)
- Fuel burnup
- Operating history: local void fraction (BWR) or moderator temperature (PWR), boron concentration (PWR), position of control rods (BWR)

In order to provide the sufficient building blocks for the full-scale simulation, spatial homogenization must cover the full scope of reactor operating conditions

Covering all state points

Covering all assembly types, burnup points and operation condition requires hundreds or thousands of runs. There exists codes and scripts for automating the task:

SerpentXS – Python-based wrapper script developed by Bryan Herman from MIT for Serpent-PARCS code sequence[9]

S2P – Matlab-based wrapper script developed by Levon Ghasabyan from PSI for Serpent-PARCS code sequence[10]

PyNE – Python-based utility code package developed by Anthony Scopatz from University of Wisconsin-Madison[11]

GenPMAXS – The preprocessor code for PARCS (the latest version has Serpent support)

Problem: *All these scripts were developed for Serpent 1, and they are not compatible with the new variables in Serpent 2 (according to my knowledge).*

[9] http://canes.github.io/SerpentXS/index.html

[10] https://github.com/tumregels/stop

[11] https://github.com/pyne/pyne

Automated calculation sequence in Serpent 2

To simplify group constant generation, an automated calculation sequence is currently being developed for Serpent 2:

- Based on restart calculations, performed after the main burnup cycle
- Capable of handling branches to different state points
- Writes a separate output file, easily read to processing scripts
- Covers only branch calculations, history calculations must be run separately
- Available since version 2.1.21, important updates in 2.1.22.

The work is still under way, and there is a separate topic for discussion at the Serpent forum: http://ttuki.vtt.fi/serpent

Automated calculation sequence in Serpent 2

The calculation sequence is based on branches (’branch’ card), used to inflict changes into original input file:

- Change in material density and temperature (’stp’ entry)[12]
- Replace one material with another (’repm’ entry)
- Replace one universe with another (’repu’ entry)

And a coefficient matrix (’coef’ card):

- Defines burnup points for which branch calculations are performed
- Defines combinations of branches run for each burnup point

[12] The code automatically retrieves the correct cross sections from the directory file and performs Doppler-broadening on the data. Variations in moderator temperature can be inflicted by replacing the entire material.

Simplified input example

```
% -- Nominal state branch (do nothing):

branch nom

% -- Fuel temperature branches (change in material temperature):

branch fueH stp fuel24 -10.457 1600
branch fueC stp fuel24 -10.457 625

% -- Coolant boron concentration branches (replace material):

branch borL repm cool cool_loB
branch borH repm cool cool_hiB

% -- Control rod insertion branch (replace universe):

branch CR repu T R

% -- Coefficient matrix (13 burnup points, 3x3x2 branch combinations):

coef 13
0 5 10 15 20 25 30 35 40 45 50 55 60
3 nom fueH fueC
3 nom borL borH
2 nom CR
```

Example case: Serpent-ARES code sequence

Some preliminary tests have been carried out for the Serpent-ARES code sequence. Observations:

- It was possible to define the branches in such way that all state points were covered, but the calculations resulted in several combinations that were not used for anything
- A post-processing script was relatively easy to write for the output file, but the format had to be changed a few times along the way – the format is likely to change in near future updates as well
- Use of variables was a convenient way to pass history information into the processing script

The methods were also successfully applied to HZP initial core calculations with Serpent-TRAB3D (a transient code developed at VTT)

Example case: Serpent-ARES code sequence

ARES reads group constants for each control rod type, burnup and void (BWR) or moderator temperature (PWR) history and the values for the local homogeneous flux solution inside each node are obtained by table interpolation and polynomial fit:

$$\begin{aligned}\Sigma = \Sigma_0 &+ (a\rho + b)\rho + (cT_\mathrm{f} + d)\,T_\mathrm{f} + (eT_\mathrm{m} + f)\,T_\mathrm{m} + g\rho T_\mathrm{f} \\ &+ (hT_\mathrm{m} + i)\rho T_\mathrm{m} + jT_\mathrm{f}\,T_\mathrm{m} + kH_\mathrm{cr} + (lB + m)B \\ &+ (nH_\mathrm{bo} + o)H_\mathrm{bo} + (pB^2\,T_\mathrm{m}^2 + qB^2\,T_\mathrm{m} + vBT_\mathrm{m}^2 + wBT_\mathrm{m}) \\ &+ (rB + s)BT_\mathrm{f} + (tB + u)B\rho\,,\end{aligned}$$

where Σ_0 is the value at the nominal state, $a \ldots w$ are the polynomial coefficients and

- ρ is the relative coolant density - nominal value
- T_f is the square root of fuel temperature - nominal value
- T_m is the moderator temperature
- H_cr is the control rod history (BWR)
- B is the boron concentration (PWR)
- H_bo is the boron history (PWR)

Example case: Serpent-ARES code sequence

The previous HZP initial core BEAVRS[13] benchmark calculations were repeated with cross sections parametrized for boron concentration:

- Good results for radial and axial power distribution compared the 3D Serpent reference calculation
- Critical boron concentrations and control rod bank worths consistent with benchmark values

The next stage is to proceed to HFP and burnup:

- Critical boron for HFP state with equilibrium xenon consistent with benchmark value
- Comparison to Serpent 3D calculation under way (fuel temperature and moderator density/temperature from ARES to Serpent reference calculation)
- Processing script is still missing the calculation of cross terms

[13] http://crpg.mit.edu/pub/beavrs

Example case: Serpent-ARES code sequence

Table 1: Critical boron concentrations (ppm).

Configuration	ARES	Ref.	A-R
ARO	989	975	14
D	926	902	24
C,D	824	810	14
A,B,C,D	683	686	-3
A,B,C,D,SE,SD,SC	486	508	-22

Table 2: Control rod bank worths (pcm).

Bank	Serpent 3D	ARES	Ref.	A-S	A-R
D	781 (5)	787	788	6	-1
C	1244 (5)	1245	1203	1	42
B	1199 (6)	1197	1171	-2	26
A	538 (6)	528	548	-10	-20
SE	497 (5)	497	461	0	36
SD	785 (6)	783	772	-2	11
SC	1120 (6)	1093	1099	-27	-6

Estimation of computational cost

Monte Carlo simulation is computationally expensive compared to deterministic transport methods, and the results are always accompanied by statistical error. This raises two questions:

i) How many neutron histories need to be run for spatial homogenization to reduce the statistical errors to such level that random noise is not reflected in the simulator calculation?

ii) What is the overall computational cost for generating the full set of group constants, covering all assembly types, burnups and state points?

These questions are addressed in a paper presented at PHYSOR 2014 in two weeks,[14] using the BEAVRS benchmark as a test case.

[14] J. Leppänen and R. Mattila. "*On the Practical Feasibility of the Continuous-energy Monte Carlo Method for Spatial Homogenization.*" In proc. PHYSOR 2014. Kyoto, Japan, Sept. 28 - Oct. 3, 2013.

Estimation of computational cost

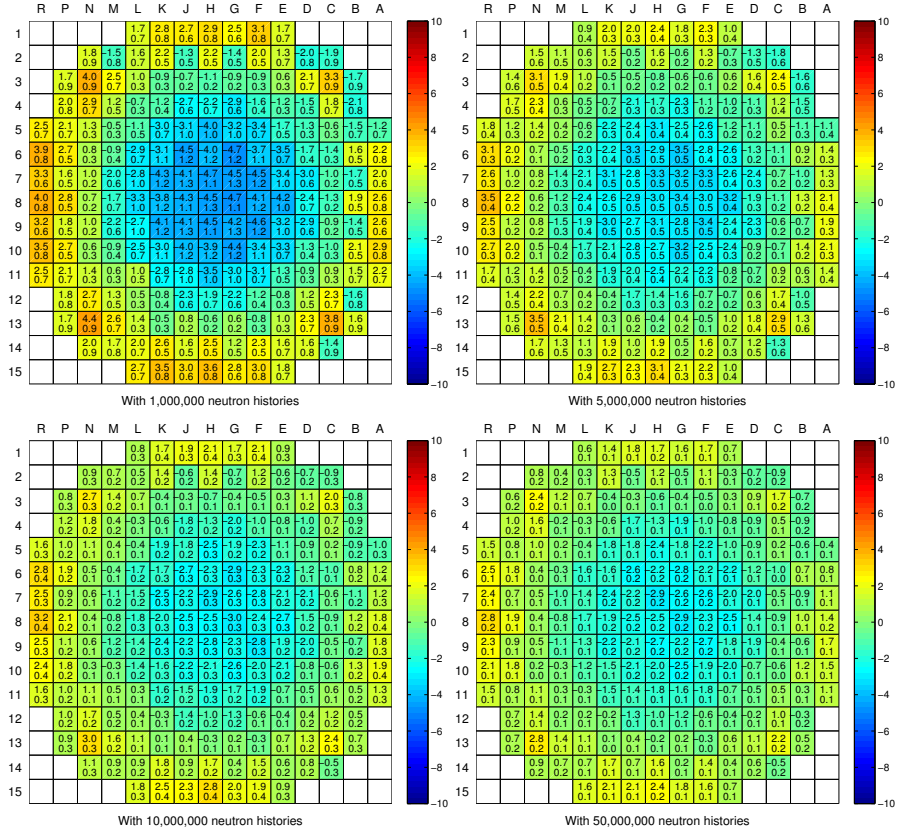


Figure 2: Differences in radial assembly power distributions at core mid-plane between Serpent-ARES and Serpent 3D reference solution. Group constant generation with different number of neutron histories. Top value: maximum difference, bottom value: standard deviation of differences. Calculations performed with 20 sets of independently generated group constants.

Estimation of computational cost

The overall calculation time was extrapolated from single assembly calculations:

- Running times from Serpent burnup calculation
- 9 assembly types, 9 history calculations, 16 burnup points
- 27 or 54 branch points, depending on assembly type (all combinations from state-point matrix)

Table 3: Running times of single history and single branch for the representative fuel assembly (2.4 wt-% ^{235}U fuel with 12 burnable absorber pins), together with the estimated total running time required for producing the full set of group constants. All calculations were run with 12 CPU cores (3.47 GHz Intel Xeon workstation).

Neutron histories	Running times (min) T_{burn}	T_{bra}	Estimated total (days)
1M	30.5	0.5	18.5
5M	91.5	1.8	62.8
10M	169.5	3.5	119.3
50M	790.9	16.6	566.9

Estimation of computational cost

Observations and conclusion of the study:

- Increasing the number of neutron histories beyond 10 million in homogenization does not significantly change the results of the ARES calculation
- Random asymmetry in ADF's can inflict large systematic tilt in the core power distribution, which can be avoided by averaging over the symmetries[15]
- Covering all state points was estimated to take a few CPU months of calculation time, which can be significantly reduced by running separate history calculations in different computer nodes
- In reality, the overall computational cost is considerably lower, since all state-point combinations are not used

[15] Serpent has an option for this in the 'ADF' card

Summary and conclusions

- Traditional calculation sequence used for modeling operating nuclear reactors is based on spatial homogenization
- Using continuous-energy Monte Carlo codes for the task has several advantages over traditional deterministic approach
- Serpent was originally designed specifically for for spatial homogenization, which still remains as one of the two main focus areas for future development
- Implementation of an automated sequence for performing branch calculations is under way
- The results have been promising, but more user experience is needed!
- Serpent seems like a viable option for spatial homogenization, even when burnup and all state points are covered, but a full-scale demonstration is yet to be accomplished


TECHNOLOGY FOR BUSINESS