20th AER Symposium on VVER Reactor Physics and Reactor Safety

20 - 24 September 2010
Espoo, Finland

2.2 SERPENT MONTE CARLO REACTOR PHYSICS CODE

Jaakko Leppänen, VTT
Tietotie 3, FI-02150 Espoo
Finland
Tel: +358 40 593 9076, Email: Jaakko.Leppanen@vtt.fi
http://montecarlo.vtt.fi

ABSTRACT

Serpent is a three-dimensional continuous-energy Monte Carlo reactor physics burnup calculation code, developed at VTT Technical Research Centre of Finland since 2004. The code is specialized in lattice physics applications, but the universe-based geometry description allows transport simulation to be carried out in complicated three-dimensional geometries as well. The suggested applications of Serpent include generation of homogenized multi-group constants for deterministic reactor simulator calculations, fuel cycle studies involving detailed assembly-level burnup calculations, validation of deterministic lattice transport codes, research reactor applications, educational purposes and demonstration of reactor physics phenomena. The Serpent code has been publicly distributed by the OECD/NEA Data Bank since May 2009 and RSICC in the U.S since March 2010. The code is being used in some 35 organizations in 20 countries around the world. This paper presents an overview of the methods and capabilities of the Serpent code, with examples in the modelling of VVER-440 reactor physics.

1.INTRODUCTION

Monte Carlo codes can perform transport calculations in complicated three-dimensional geometries using continuous-energy cross sections, which allows the modelling of neutron interactions at the microscopic level without major approximations. This capability makes the calculation method well suited for its traditional applications, including criticality safety analyses, shielding and dosimetry calculations, detector modelling and the validation of deterministic transport codes. The common factor in these applications is the need to model the physics of the transport process to within maximum accuracy, often regardless of the computational effort.

Due to the excessive use of CPU time, the Monte Carlo method is not a practical choice for coupled neutronics / thermal hydraulics calculations. Monte Carlo codes can, however, be used for the previous step in the calculation chain, namely the generation of input data for deterministic reactor simulator codes. Group constant generation is a relatively new field of application for the calculation method, and both the computational challenges and the main advantages differ quite significantly from traditional Monte Carlo problems.

Unlike in criticality safety analyses, for example, the main goal is not to simulate neutron transport in a complicated realistic system, but to produce source terms and interaction parameters that represent the fuel once it is loaded in the reactor core. The calculations are typically carried out in an infinite two-dimensional lattice of identical fuel assemblies, and repeated for every assembly type in different operating conditions. This multi-stage calculation scheme is based on the theory of homogenization [1,2], and it is the standard approach to solving coupled large-scale reactor physics and dynamics problems.

Homogenization is a repetitive routine procedure that basically requires condensing isotopic high energy-resolution interaction parameters into a set of case-specific multi-group constants. The task also requires tracking the isotopic changes in the materials during the irradiation cycle. The main advantage of using Monte Carlo codes for homogenization is not so much the accuracy of the calculation method, but rather its versatility. The same code and cross section data can be used for modelling any fuel or reactor configuration without losing the reliability of the calculation scheme. This is not always the case with deterministic lattice transport codes, which often rely on various application-specific approximations.

The main reason why Monte Carlo codes are not more widely used for lattice physics applications is probably the prohibitively long running time, especially when burnup calculation is involved. Another reason is that most of the publicly available general-purpose codes are simply incapable of calculating all the input parameters required for full-core reactor simulator calculations, or simulate fuel burnup. These challenges gave rise to the development of the Serpent code at VTT Technical Research Centre of Finland in 2004 [3]. One of the main goals of the project is to show that despite its limitations regarding computational efficiency, the continuous-energy Monte Carlo method may become a viable option to deterministic transport codes within the near future.

This paper is a general overview of the capabilities and calculation methods used in the Serpent code. A brief history of the Serpent project is given in Section 2. Methods and capabilities are introduced in Section 3, and Section 4 presents some example results related

to VVER reactor physics. Future challenges are discussed Section 5 and Section 6 is left for the conclusions.

2.HISTORY

The Serpent project was started at VTT Technical Research Centre of Finland in 2004, under the working title "Probabilistic Scattering Game", or PSG. All publications dated before October 2008 refer to the code using this name. The main motivation at the time was to develop the capability to produce homogenized multi-group constants for deterministic nodal diffusion codes using the continuous-energy Monte Carlo method. This capability was considered essential for the modelling of next-generation reactor systems, as the applications often lie beyond the capabilities of traditional LWR codes. It was also thought that the development of a dedicated Monte Carlo lattice physics code could extend the applications of the calculation method, and bring new possibilities for LWR analyses as well.

The early versions of the code were developed without burnup calculation capability, and the main focus was in the interaction physics and the production of homogenized multi-group constants. Significant effort had to be put especially in the calculation of diffusion coefficients, as the continuous-energy Monte Carlo method is not easily combined with the diffusion approximation. The early code development was also the topic of a doctoral thesis, completed in 2007 [3].

Depletion routines were added in the "version 2" of the code in 2008, the name was changed from PSG to Serpent, and a website was established for the project at http://montecarlo.vtt.fi. A limited pre-release version was also distributed to some research institutes for testing purposes. The code was released at the OECD/NEA Data Bank in May 2009. RSICC distribution in North America began one year later, in March 2010. At the same time, a public discussion forum was set up for Serpent users at http://ttuki.vtt.fi/serpent.

Serpent is licensed free of charge for non-commercial research and educational use. The code is mainly distributed by the two data centres, but some users have acquired Serpent by bilateral license agreements with VTT. Currently the user community consists of some 35 organizations in 20 countries around the world.

3.METHODS AND APPLICATIONS

Serpent was originally developed and mainly intended for lattice physics calculations, similar to the presently-used second-generation deterministic lattice transport codes. The suggested applications include:

- Generation of homogenized multi-group constants for deterministic reactor simulator calculations
- Fuel cycle studies involving detailed assembly-level burnup calculations
- Research reactor applications
- Validation of deterministic lattice transport codes

The transport capabilities of the continuous-energy Monte Carlo method naturally extend beyond the capabilities of traditional deterministic lattice physics codes. The simulation is not limited to two-dimensional assembly geometries, and the code can be used for modelling any three-dimensional full-core configuration as well. The interaction physics covers all reactor types, including both thermal and fast-spectrum systems.

3.1. Neutron Tracking and Interaction Physics

Serpent uses a combination of the conventional surface-to-surface ray tracing and the Woodcock delta-tracking method [4] for simulating neutron transport through the geometry. The delta-tracking method is essentially a rejection sampling technique that enables the random walk to be continued over several material regions without stopping the neutron at each boundary surface. This method has proven fast and efficient in lattice geometries, especially when combined with conventional techniques [5]. The geometry routine is built on a universe-based approach, very similar to other Monte Carlo codes, such as MCNP [6] and KENO-VI [7] (the universe concept is equivalent with the unit definition in SCALE). This type of geometry model allows the description of practically any two- or three-dimensional fuel or reactor configuration.

The interaction physics in Serpent is based on classical collision kinematics and ENDF reaction laws. The code reads continuous-energy cross sections and unresolved resonance probability tables from ACE format library files, which are also used by MCNP. The reaction cross sections are reconstructed using a single unionized energy grid for all nuclides [8]. This approach leads to a dramatic increase in efficiency, especially when modelling irradiated fuel consisting of over 200 actinide and fission product cross sections, as the number of time-consuming grid search iterations is reduced to minimum.

3.2. Burnup Calculation

Simulation of fuel depletion is a cyclic process that requires the calculation of isotopic one-group transmutation cross sections for each depleted material at each depletion step. This data is combined with radioactive decay constants and fission yields, and formulated into a set of coupled first-order differential equations. Serpent has two built-in methods for solving these depletion equations. The first option is the linear chains method, also referred to as Transmutation Trajectory Analysis (TTA) [9]. The second alternative, used by default, is the Chebyshev Rational Approximation Method (CRAM) [10], an advanced matrix exponential solution specifically developed for the Serpent code.

The calculation of one-group transmutation cross sections requires a very large number of reaction rate tallies, which results in a significant penalty in CPU time. To overcome this problem, Serpent can also generate the data indirectly after the transport simulation, by collapsing continuous-energy cross sections using a flux spectrum collected for each burnable material. Similar methods have been developed for other Monte Carlo based burnup calculation codes as well [11,12]. To maximize the resolution, Serpent collects the flux spectra using the same unionized energy grid that is used to tabulate the cross sections. The main drawback of the method is that additional storage space is needed for the material fluxes and that all information on statistical accuracy is lost. In order to account for the self-shielding effects in the unresolved resonance region, cross sections produced by probability table sampling are handled separately.

All parameters needed to form the depletion equations are automatically calculated and set up without additional user effort. The radioactive decay constant and fission yield data is read from standard ENDF format libraries and Serpent can be run as a completely stand-alone burnup calculation code without dependence on external depletion solvers or pre-generated data sets.

3.3. Parallel Calculation Mode

Serpent has the capability to use the Message Passing Interface (MPI) for parallel calculation. Parallelization is implemented by dividing the neutron population between the parallel tasks and combining the results after the transport cycle. If the number of depleted materials is large in burnup calculation, the processing and depletion routines between the burnup steps may take a significant fraction of the overall calculation time. To speed up the calculation, these processes are also divided between the MPI tasks.

3.4. Output

Since Serpent is primarily intended as a lattice physics code, several assembly level parameters are calculated by default. This data includes:

- Effective and infinite multiplication factors calculated using analog and implicit estimators
- Homogenized multi-group reaction cross sections
- Group-transfer probabilities and scattering matrices
- Diffusion coefficients calculated using two fundamentally different methods
- Pn scattering cross sections up to order 5
- Assembly discontinuity factors for boundary surfaces and corners in square and hexagonal fuel lattices
- Assembly pin-power distributions
- Point reactor kinetics parameters
- Physical and effective delayed neutron fractions and decay constants in 6 or 8 precursor groups
- Normalized flux, power and reaction rates integrated over geometry
- Parameters for the six-factor formula
- Various parameters related to the Monte Carlo transport simulation

All result estimates are accompanied by the associated relative statistical errors.

Fission source entropies are available for convergence studies and user-defined detectors (tallies) can be set up for calculating various integral reaction rates. The output in the burnup calculation mode consists of isotopic compositions, transmutation cross sections, activities and decay heat data. The results are given both as material-wise and total values. Group constants and all the other output parameters are calculated and printed for each depletion step.

All numerical output is written in Matlab m-format files to simplify the post-processing of the data. The code also has a geometry plotter feature and a reaction rate plotter, which is convenient for visualizing the neutronics in thermal systems.

3.5. Additional Features

In addition to the standard Monte Carlo techniques and group constant generation, Serpent uses a number of specialized calculation routines. The code has an explicit particle fuel model for high-temperature gas-cooled reactor geometries [13,14]. This model allows the accurate description of random particle distributions inside graphite pebbles or cylindrical compacts, without any approximations. The model works at multiple geometry levels, and it can also be used with pebble distributions inside pebble-bed reactor cores. The routine uses a three-dimensional search mesh to speed up neutron tracking and it is capable of handling millions of random units without significant increase in calculation time compared to regular lattice models. The explicit model has been successfully used in full-core pebble-bed reactor calculations, in combination with a DEM (Discrete Element Method) solver to generate the random distribution of fuel pebbles [15].

To account for the temperature dependence of neutron interactions, Serpent has a built-in Doppler-broadening routine that allows cross sections to be adjusted to higher temperatures [16]. The build-up of Xe-135 fission product poison can be simulated without running full-scale burnup calculation using a built-in iteration routine. Code calculates equilibrium concentrations of Xe-135 and I-135 from the production, decay and absorption rates of the isotopes during the transport simulation.

3.6. Major Limitations

The optimization of the calculation routines for lattice physics applications results in some limitations in terms of generality. The delta-tracking method necessitates the use of the collision flux estimator for calculating integral reaction rates. The conventional approach is to use the track-length estimator, which has a better efficiency in small and optically thin cells and in regions of low collision density. For this reason the Serpent code is not well suited for detector and dosimetry calculations, at least to the extent of general-purpose Monte Carlo codes.

The unionized energy grid structure combines the energy points of all the constituent nuclides to avoid repeating the grid search for the partial grids. The drawback of the efficiency gain is that computer memory is wasted for storing redundant data. The memory demand easily grows to several gigabytes in burnup calculation, due to the number of actinide and fission product cross sections involved. The problem becomes even worse in the parallel calculation mode if several MPI tasks are sharing the same memory space. Serpent uses two methods, based on grid thinning and the double-indexing of the energy grids [8] to reduce the overall memory demand. Memory usage is not considered a limiting factor for typical assembly burnup calculations containing less that a few hundred depletion zones, but it may rule out the modelling of larger systems.

4.EXAMPLE RESULTS

Serpent has been used for various tasks, ranging from lattice physics calculations to the modelling of complicated research reactor geometries. The following subsections present example results related to the physics of VVER-440 reactors.

4.1. Serpent vs. MCNP Validation

One of the best ways to validate the transport routines in Serpent is to compare the results to reference values produced by MCNP [6]. The two codes share the same cross section library format, and consequently, the same laws of physics. This eliminates any additional uncertainties from the comparison. The MCNP code itself has decades of development history and it has been extensively used and validated against experimental data, which makes it the perfect reference code for Serpent validation.

Table 1 shows the comparison of infinite multiplication factors, prompt neutron lifetimes and homogenized four-group cross sections calculated using Serpent 1.1.13 and MCNP5. The geometry is an infinite 2D model of a TVEL-type VVER-440 fuel assembly with 3.6% enriched uranium oxide fuel and 650 ppm coolant boron concentration. The Serpent cross sections are produced by default, along with the associated relative statistical errors. The MCNP results are calculated by combining the data from multiple reaction rate tallies. The combined statistical errors are conservative and over-estimated, due to the lack of covariances between the tally results.

It is seen that the differences between the codes are below the 95% confidence intervals of the Serpent results, which is an indication that the ENDF interaction laws in the ACE format data are correctly implemented. The results are quite typical for this type of comparison. Figures 1 and 2 show the comparison of flux spectra integrated over the entire geometry for the same calculation case. It is clearly seen that the differences are mostly statistical noise.

The calculations with both codes were run using 10 million neutron histories. The Serpent calculation took 18 minutes on a single-processor 3.0 GHz AMD Opteron PC. The MCNP calculation run for 258 minutes on the same machine. The factor of 15 between the running times is the result of optimizing Serpent for lattice physics calculations using the Woodcock delta-tracking method and the unionized energy grid approach. It should also be noted, however, that the running time for MCNP is significantly affected by the relatively large number of reaction rate tallies required to produce the results and the code runs faster when used for criticality calculation only.

4.2. VVER-440 Local Power Peaking Experiment Benchmark

The second example is a computational benchmark related to VVER-440 local power peaking experiments carried out in 2004-2005 at the LR-0 reactor operated by NRI Rež plc. in Czech Republic [17]. The reactor configuration consists of the joint region between a VVER-440 control assembly and the fuel follower, surrounded by multiple rings of standard fuel assemblies. The purpose of the experiment was to measure the local power peaking in fuel rods adjacent to the joint region where neutron moderation is enhanced due to the presence of excess water.

The benchmark calculations were carried out in 2007, using MCNP4c and PSG, the predecessor of the Serpent code. A complete description of the calculations is found Ref. [18]. Results for one of the experimental configurations is plotted as an example in Figure 3. It is seen that the calculations are close to the experimental measurements, and the fact that both codes yield consistent values suggests that the small differences are most likely due to approximations in the computational model.

Table 1: Comparison of infinite multiplication factors, prompt neutron lifetimes and homogenized multi-group cross sections calculated by MCNP5 and Serpent 1.1.13. Relative statistical errors and differences are in percent. The four energy groups represent fast fission, slowing-down, resonance and thermal regions, respectively. Calculated using JEFF-3.1.1 based cross section libraries.

Param.	g	MCNP5		Serpent 1.1.13		Δ (%)
k_{eff}		1.26989	(0.015)	1.27008	(0.014)	0.015
l_{p}		18.15	(0.045)	18.14	(0.029)	-0.015
Σ_{tot}	1	2.2179E-01	(0.057)	2.2178E-01	(0.011)	-0.004
	2	5.1730E-01	(0.028)	5.1741E-01	(0.007)	0.020
	3	7.7515E-01	(0.028)	7.7517E-01	(0.007)	0.003
	4	1.2064E+00	(0.050)	1.2061E+00	(0.013)	-0.027
Σ_{fiss}	1	2.8256E-03	(0.064)	2.8254E-03	(0.043)	-0.009
	2	4.1053E-04	(0.028)	4.1049E-04	(0.021)	-0.008
	3	5.7941E-03	(0.036)	5.7955E-03	(0.040)	0.025
	4	5.9894E-02	(0.050)	5.9886E-02	(0.028)	-0.013
Σ_{capt}	1	7.1002E-04	(0.072)	7.1037E-04	(0.052)	0.049
	2	1.9226E-03	(0.036)	1.9226E-03	(0.029)	0.004
	3	1.8919E-02	(0.045)	1.8914E-02	(0.044)	-0.026
	4	2.9597E-02	(0.050)	2.9591E-02	(0.017)	-0.021
Σ_{scatt}	1	2.1825E-01	(0.057)	2.1812E-01	(0.011)	-0.062
	2	5.1497E-01	(0.028)	5.1507E-01	(0.007)	0.020
	3	7.5044E-01	(0.028)	7.5046E-01	(0.007)	0.003
	4	1.1169E+00	(0.050)	1.1166E+00	(0.014)	-0.029
$\nu\Sigma_{\mathrm{fiss}}$	1	7.9071E-03	(0.064)	7.9062E-03	(0.045)	-0.011
	2	1.0050E-03	(0.028)	1.0049E-03	(0.020)	-0.009
	3	1.4102E-02	(0.036)	1.4106E-02	(0.040)	0.025
	4	1.4591E-01	(0.050)	1.4589E-01	(0.028)	-0.013
ν	1	2.7983E+00	(0.071)	2.7983E+00	(0.006)	-0.002
	2	2.4480E+00	(0.028)	2.4480E+00	(0.000)	-0.001
	3	2.4339E+00	(0.042)	2.4339E+00	(0.000)	0.000
	4	2.4362E+00	(0.057)	2.4362E+00	(0.000)	0.000
$1/v$	1	5.3098E-10	(0.057)	5.3101E-10	(0.010)	0.004
	2	2.6320E-09	(0.028)	2.6330E-09	(0.014)	0.035
	3	1.7644E-07	(0.036)	1.7644E-07	(0.023)	0.002
	4	2.3559E-06	(0.050)	2.3556E-06	(0.013)	-0.015

5.FUTURE PROSPECTS

The Serpent code has been developed in one form or another for about five years now. Although the source code has been completely rewritten twice, the basic structure of the program has remained relatively unchanged. The current, publicly distributed version of Serpent is largely based on its predecessor, PSG [3], which was developed without considering fuel depletion or parallel calculation. The burnup calculation routines and MPI implementation were added later, without changing the basic structure of the source code.

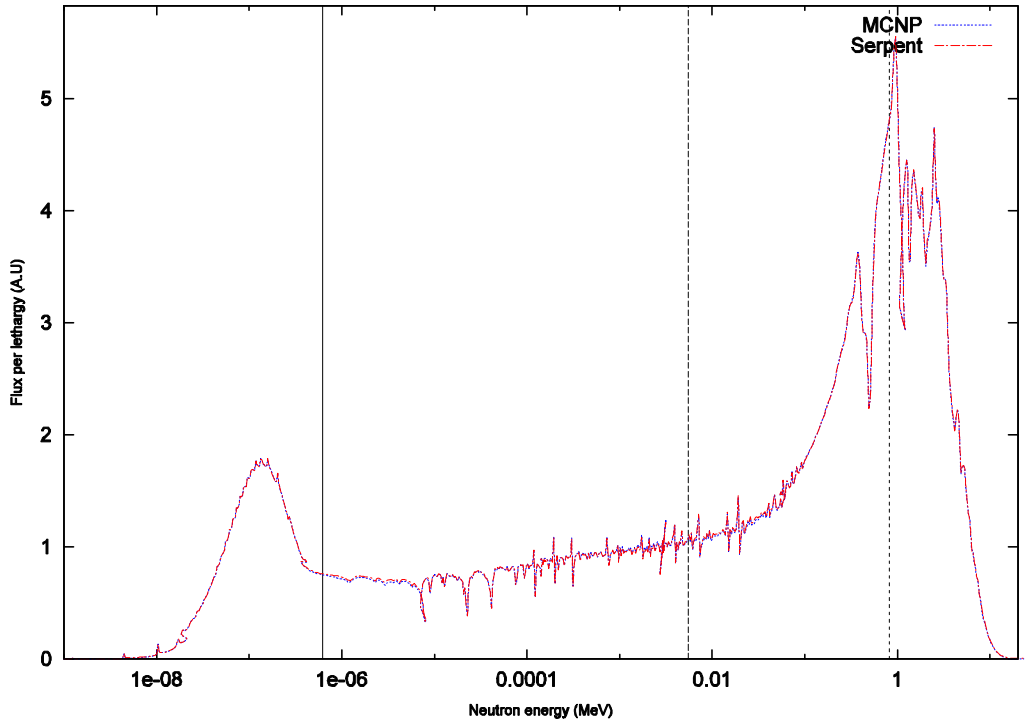

Figure 1: Flux spectrum integrated over a VVER-440 fuel assembly. Calculated using MCNP5 and Serpent 1.1.13 using JEFF-3.1.1 based cross section libraries.

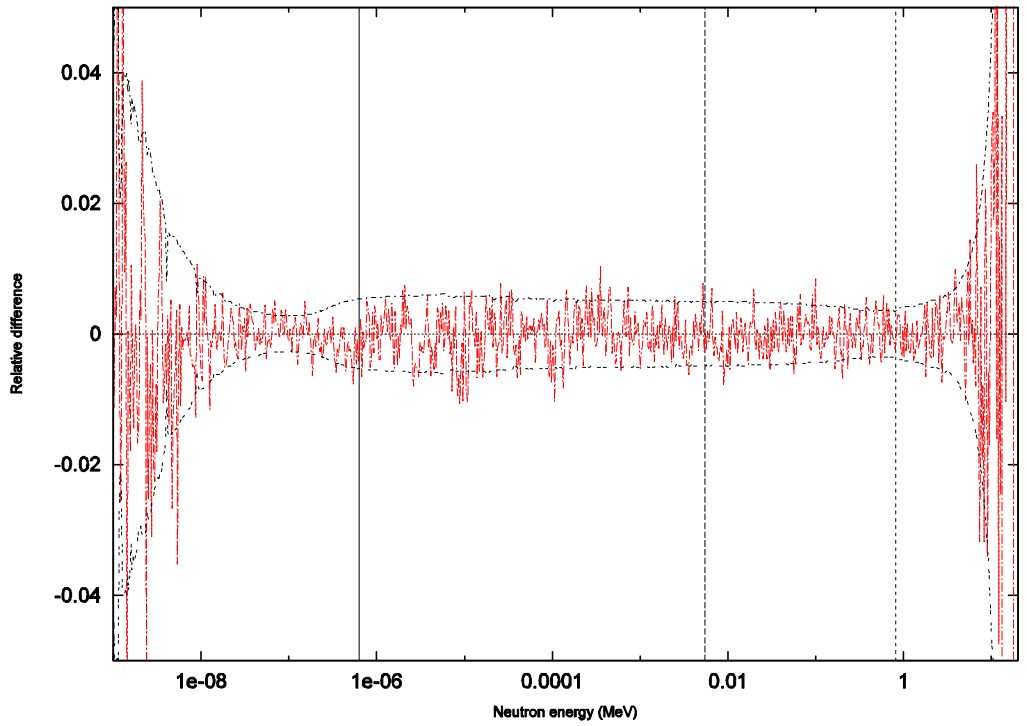

Figure 2: Relative differences between MCNP5 and Serpent results in Figure 1. The black curve shows the 95% confidence interval of the MCNP result.

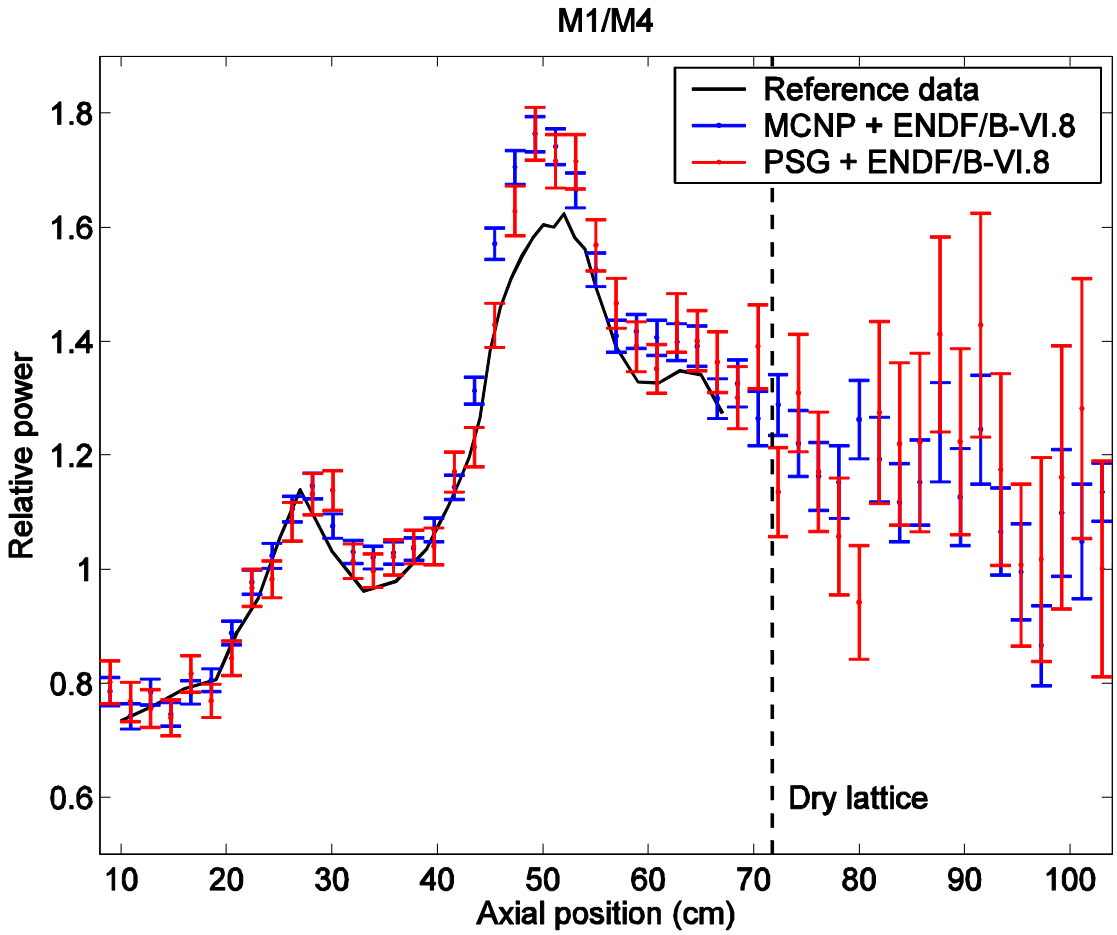

Figure 3: Measured and calculated axial power distribution in a fuel rod adjacent to the water gap in the joint region between a VVER-440 control assembly and fuel follower.

Performing burnup calculations requires a tremendous amount of data to be handled as efficiently as possible, avoiding all potential bottlenecks that might lead to increase in the overall running time. When the methodology was developed, the best practices had to be figured out along the way, as new calculation routines were built on top of existing source code. The result was that the solutions became overly complicated and error-prone, and that reductions in simulation time were accomplished at the cost of excessive memory usage.

Similar problems were encountered in the implementation of the MPI routines. Parallel calculation requires communication and exchange of data between separate calculation tasks performing similar operations at the same time. The solution currently used in Serpent is simple and straightforward: all input data is shared at the beginning of the simulation and the results are combined at the end. Although this approach leads to efficient parallelization by minimizing the communication between the hosts, it has at least two major drawbacks. First, the Monte Carlo simulation performed using a single CPU cannot be reproduced event-by-event, even with the same random number sequence. This means that the methods cannot be validated without running a very large number of neutron histories to minimize statistical fluctuation in the results. The second problem is related to memory management. Two parallel tasks cannot share the same address space, and memory usage is consequently multiplied by the number of CPU cores performing the calculation.

The limit in memory usage is usually hit either when increasing the number of depletion zones in burnup calculation, or the number of MPI tasks when running the code in the parallel mode. With a typical PC workstation, Serpent can handle several hundred burnable materials, which is more than enough for typical lattice physics applications. This, however, may not be sufficient for the future. Recent publications [19] and communication with Serpent users suggests that there is an increasing interest in using Monte Carlo codes for more and more challenging tasks, including massively parallelized full-core calculations [20]. Because of the fundamental limitations in memory management, it is likely that the development of computer hardware will outrun the calculation methods within a few years. This is especially the case when more and more cores are added to CPU:s, and Serpent can only utilize a fraction of them due to the incapability to use shared memory in parallel calculation.

Tackling the problem requires significant changes in the program structure, which can be considered as one of the major challenges for future code development.

6.SUMMARY AND CONCLUSIONS

The Serpent Monte Carlo reactor physics burnup calculation code has been developed at VTT Technical Research Centre of Finland since 2004. At the beginning of the project, the main goal was to develop a new lattice physics code, capable of producing homogenized group constants for deterministic reactor simulator calculations using the continuous-energy Monte Carlo method. New capabilities, including built-in burnup calculation routines have been added in the code along the way.

Serpent is publicly distributed by two data centres: the OECD / NEA Data Bank and RSICC in the U.S. The user community consists of some 35 organizations in 20 countries around the world. As the number of users has steadily grown since the public release in May 2009, so has the diversity of applications. Lattice physics calculations and croup constant generation can no longer be taken as the primary goal for future development. Modelling of large and complicated systems in massively parallelized computing environments is a major challenge, which requires re-thinking some of the fundamental approaches in the code structure, and taking into account the lessons learned from previous development.

REFERENCES

[1] K. Koebke, *A New Approach to Homogenization and Group Condensation*, International Atomic Energy Agency, IAEA-TECDOC-231 (1978).

[2] K. Smith, *Spatial Homogenization Methods for Light Water Reactor Analysis*, Ph.D. Thesis, Massachusetts Institute of Technology (1980).

[3] J. Leppänen, *Development of a New Monte Carlo Reactor Physics* Code, D.Sc. Thesis, Helsinki University of Technology (2007).

[4] E. R. Woodcock, *Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry*, Argonne National Laboratory, ANL-7050 (1965).

[5] J. Leppänen, *Performance of Woodcock Delta-Tracking in Lattice Physics Applications Using the Serpent Monte Carlo Reactor Physics Burnup Calculation Code*, Ann. Nucl. Energy, **37** (2010) 715-722.

[6] X-5 Monte Carlo Team, *MCNP – A General Monte Carlo N-Particle Transport Code, Version 5*, Los Alamos National Laboratory, LA-UR-03-1987 (2003).

[7] D. F. Hollenbach et al., K*ENO-VI: A General Quadratic Version of the KENO Program*, Oak Ridge National Laboratory, ORNL/TM-2005/39 Version 6 Vol. II, Sect. F17 (2009).

[8] J. Leppänen, *Two Practical Methods for Unionized Energy Grid Construction in Continuous-Energy Monte Carlo Neutron Transport Calculation*, Ann. Nucl. Energy, 36 (2009) 878-885.

[9] J. Cetnar, *General solution of Bateman equations for nuclear transmutations*, Ann. Nucl. Energy, **33** (2006) 640-645.

[10] M. Pusa and J. Leppänen, *Computing the Matrix Exponential in Burnup Calculations*, Nucl. Sci. Eng., **164** (2010) 140-150.

[11] E. Fridman, E. Shwageraus, and A. Galperin, *Efficient Generation of One-Group Cross Sections for Coupled Monte Carlo Depletion Calculations*, Nucl. Sci. Eng., **159** (2008) 37-47.

[12] W. Haeck, and B. Verboomen, *An Optimum Approach to Monte Carlo Burnup*, Nucl. Sci. Eng., **156** (2007) 180-196.

[13] J. Leppänen and M. DeHart, *HTGR Reactor Physics and Burnup Calculations Using the Serpent Monte Carlo Code*, Trans. Am. Nucl. Soc. **101** (2009) 782-784

[14] J. Leppänen, *HTGR Modelling Capabilities in the Serpent Monte Carlo Code*, In Proc. IYNC-2010. Cape Town, South Africa, Jul. 12-18 2010.

[15] H. Suikkanen, V. Rintala and R. Kyrki-Rajamäki, *An Approach for Reactor Physics Modelling of Randomly Packed Pebble Beds*, In Proc. HTR-2010. Prague, Czech Republic, October 18-20 2010.

[16] T. Viitanen, *Implementing a Doppler-Preprocessor of Cross Section Libraries in Reactor Physics Code Serpent*, M.Sc. Thesis, Helsinki University of Technology (2009).

[17] R. Jošek, E. Novák and V. Rypar, *WWER-440 Local Power Peaking Experiment: Benchmark and Material Specification*, UJV Z 1651, NRI, (2006).

[18] J. Leppänen, *VVER-440 Local Power Peaking Experiment Benchmark*, VTT Technical Research Centre of Finland VTT-R-02688-07 (2007).

[19] K. Smith, *Reactor Core Methods*, Invited lecture at the M&C 2003 International Conference, April 6-13, 2003, Gatlinburg, TN, USA (2003).

[20] E. Hoogenboom, W. Martin and B. Petrovic, *Monte Carlo Performance Benchmark for Power Density Calculation in a Full Size Reactor Core*, OECD/NEA (2009).