International Conference on Mathematics, Computational Methods & Reactor Physics (M&C 2009)
Saratoga Springs, New York, May 3-7, 2009, on CD-ROM, American Nuclear Society, LaGrange Park, IL (2009)

BURNUP CALCULATION CAPABILITY IN THE PSG2 / SERPENT MONTE CARLO REACTOR PHYSICS CODE

Jaakko Leppänen and Maria Pusa
VTT Technical Research Centre of Finland
Tietotie 3 Espoo, P.O.Box 1000, FI-02044 VTT, Finland
Jaakko.Leppanen@vtt.fi; Maria.Pusa@vtt.fi

ABSTRACT

The PSG continuous-energy Monte Carlo reactor physics code has been developed at VTT Technical Research Centre of Finland since 2004. The code is mainly intended for group constant generation for coupled reactor simulator calculations and other tasks traditionally handled using deterministic lattices physics codes. The name was recently changed from acronym PSG to "Serpent", and the capabilities have been extended by implementing built-in burnup calculation routines that enable the code to be used for fuel cycle studies and the modelling of irradiated fuels.

This paper presents the methodology used for burnup calculation. Serpent has two fundamentally different options for solving the Bateman depletion equations: 1) the Transmutation Trajectory Analysis method (TTA), based on the analytical solution of linearized depletion chains and 2) the Chebyshev Rational Approximation Method (CRAM), an advanced matrix exponential solution developed at VTT. The first validation results are compared to deterministic CASMO-4E calculations. It is also shown that the overall running time in Monte Carlo burnup calculation can be significantly reduced using specialized calculation techniques, and that the continuous-energy Monte Carlo method is becoming a viable alternative to deterministic assembly burnup codes.

Key Words: Serpent, PSG, Monte Carlo burnup calculation, TTA, CRAM

1. INTRODUCTION

The increasing computer capacity has made the Monte Carlo method an interesting option for various reactor physics applications in which deterministic transport codes have traditionally been used. One of such applications is homogenization, in which the geometry and the interaction physics in the reactor core are reduced to such a level that simple and efficient deterministic methods, such as the few-group nodal diffusion method, can be used for coupled full-core calculations.

Homogenization is not easily carried out using most general-purpose Monte Carlo codes, mainly because the calculation of certain few-group constants lies beyond their standard tally capabilities. As a solution, it was decided in 2004 at the VTT Technical Research Centre of Finland, to develop a completely new continuous-energy Monte Carlo code specifically for homogenization and other lattice physics applications. The project started under the working title "Probabilistic Scattering Game", or PSG [1]. The name was changed to "Serpent" [2] in 2008 and a public release to the OECD/NEA Data Bank is being prepared. *

The code has the capability to calculate homogenized multi-group constants, assembly discontinuity factors, pin-power distributions, diffusion coefficients and various kinetic and

* A User's Manual and the detailed description of the code are available at the Serpent website http://montecarlo.vtt.fi

effective delayed neutron parameters, all without any additional user input. The calculation methods have been validated against both Monte Carlo and deterministic transport codes with good results. The main limitation for the practical use of PSG was the lack of burnup calculation capability. This flaw has now been corrected in the new Serpent code by introducing built-in depletion routines based on two fundamentally different methods. The first option is the Transmutation Trajectory Analysis Method (TTA) [3], based on the analytical solution of linearized depletion chains, and the second an advanced matrix exponential solution using the new Chebyshev Rational Approximation Method (CRAM) [4].

Group constant generation is a repetitious process, often requiring hundreds of runs in order to cover all assembly types and operating conditions within the reactor core. Calculation is slow using the Monte Carlo method, and when burnup calculation is involved, the overall running time easily becomes a limiting factor. This paper presents the burnup calculation methodology in the Serpent code and reviews the techniques used for reducing the calculation time compared to widely-used coupled Monte Carlo burnup calculation codes.

2. OVERVIEW OF THE SERPENT CODE

Serpent can be characterized as a continuous-energy Monte Carlo reactor physics burnup calculation code. The code is specifically designed and optimized for lattice physics calculations, although the universe-based geometry model allows the description of any two- or three-dimensional system. The code automatically calculates all few-group constants and kinetic and delayed neutron parameters needed for coupled nodal diffusion calculations. User-defined detectors (tallies) can be set up for calculating integral flux and reaction rates in cells, materials and universes, or using a super-imposed three-dimensional mesh.

The transport simulation is based on the k-eigenvalue criticality source method, which limits the applications to self-sustaining systems. A combination of conventional surface-to-surface ray-tracing and the Woodcock delta-tracking method [5] is used for neutron transport. This approach takes full advantage of the efficiency of delta-tracking in complicated geometries, without the penalties resulting from the presence of localized heavy absorbers. The methodology has proved particularly efficient in infinite-lattice calculations.

Serpent reads neutron interaction data from ACE format cross section library files. All reaction cross sections are reconstructed using the same unionized energy grid structure for all nuclides. This has the advantage that time-consuming energy grid iteration is reduced to minimum, which results in a tremendous speed-up in the calculation. The unionized energy grid is constructed by combining the individual grids of all nuclides. The drawback of this approach is that the grid becomes excessively large in burnup calculation and the cross section data may require several gigabytes of computer memory. To overcome this problem, methodology has been developed for reducing the grid size without loss of accuracy [6].

3. METHODOLOGY FOR BURNUP CALCULATION

Serpent uses built-in calculation routines for burnup calculation. The code can be used as a completely stand-alone application, without any external coupling to a separate depletion code.

The coupling option exists as well, but mainly for validation purposes. The methods used in the built-in calculation routines are introduced below.

Burnup calculation is a cyclic process, consisting of two steps. The first step is the transport cycle, in which the rates of all neutron-induced transmutation reactions are calculated, in this case, using standard Monte Carlo techniques. This data is then combined with radioactive decay constants and fission product yields, read from nuclear data libraries. The isotopic changes in the irradiated materials are described by the Bateman equations

$$\frac{dN_j}{dt} = \sum_{i \neq j} \lambda_{ij} N_i - \lambda_j N_j \ , \quad N_j(0) = N_0 \ , \quad j = 1, \ldots, n \ , \tag{1}$$

where N_j is the atomic density of nuclide j, n is the total number of nuclides and λ_{ij} are the generalized transmutation coefficients characterizing the rates of neutron-induced reactions and spontaneous radioactive decay. The second stage of the process consists of solving these equations, after which the procedure is repeated using the updated material compositions.

The λ_{ij} coefficients are assumed constant over each burnup step. The reaction rates used in the calculation are hence characteristic of the flux spectrum at the beginning of each step, which may lead to an underestimation of self-shielding effects as the nuclide concentrations increase. This error can be compensated by reducing the step size. Another widely-used solution is the predictor-corrector method, in which the transmutation coefficients are corrected by repeating the transport calculation using the (predicted) material compositions at the end of the step and taking the average of the two sets of values. The predictor-corrector method is used by default in the Serpent code.

3.1. Calculation of One-group Transmutation Cross Sections

The transmutation constants for neutron-induced reactions can be written as the product of the one-group neutron flux and the microscopic one-group cross section for transmutation $i \rightarrow j$:

$$\lambda_{ij} = \phi \sigma_{ij} \ . \tag{2}$$

The flux depends on normalization and the cross section is calculated by averaging the corresponding microscopic continuous-energy cross section over the flux spectrum in the irradiated material:

$$\sigma_{ij} = \frac{\int_V \int_E \phi(\mathbf{r}, E) \sigma_{ij}(E) d^3 r dE}{\int_V \int_E \phi(\mathbf{r}, E) d^3 r dE} \ . \tag{3}$$

The integrals in Eq (3) are easy to calculate using standard Monte Carlo flux estimators.

Neutron interaction data is available for almost 300 actinides and fission products. If a large number of nuclides is included in the burnup calculation, the total number of transmutation cross sections is in the order of 1000. If the corresponding reaction rates are tallied within the transport cycle, the result is a major increase in the overall calculation time.

To overcome this penalty, Serpent uses a technique previously implemented in coupled Monte Carlo burnup calculation codes [7, 8]. The flux spectrum is tallied using a super-fine energy bin

structure, which is then used for calculating the averaged one-group cross sections after the transport cycle. Since the code uses a unionized energy grid for all reaction cross sections, the same energy grid is a natural choice for the spectrum calculation as well. The original data is represented to within maximum accuracy and the only approximation is that average, rather than linear-interpolated values are used for the integration between two tabulated energy points. Test calculations have shown that the differences are negligible for a typical grid consisting of well over 100,000 points. The reduction in calculation time is in the order of a factor of 4.

3.2. Fission Yield and Radioactive Decay Data

Equation (2) applies as such for non-fission reactions only. For fission, the one-group transmutation coefficients are defined as

$$\lambda_{ij} = \gamma_{ij}\phi\sigma_{\mathrm{f},i} \ , \tag{4}$$

where $\sigma_{\mathrm{f},i}$ is the one-group total fission cross section for actinide i and γ_{ij} is the yield of fission product j. Serpent reads fission yield data from raw ENDF format files. The yields are energy-dependent and usually tabulated for two or three different neutron energies. Linear interpolation is used between the points to calculate effective yields corresponding to the flux spectrum.

The transmutation coefficients for radioactive decay are equal to the decay constant multiplied by the branching ratio for reaction $i \rightarrow j$. This data, along with decay heat values are read from raw ENDF format files.

3.3. Solution of the Depletion Equations

When the Bateman equations (1) are written for each nuclide, the result is a set of coupled first-order differential equations. The couplings between nuclides form long and complicated transmutation chains with multiple branches. The magnitudes of the transmutation coefficients vary extensively and the corresponding half-lives range from milliseconds to billions of years. Also the time step varies. When the fuel is irradiated in the reactor, the time step is usually counted in days. For a radioactive inventory calculation the time span may extend to thousands or even millions of years. The complexity of the depletion equations and the numerical problems resulting from the extensive variation in the coefficients have inspired various solutions over the years. Serpent uses two fundamentally different methods, introduced in the following.

3.3.1. TTA method

The Bateman equations (1) can be solved analytically if the complicated transmutation chains are first resolved into a set of linear sub-chains, or trajectories. This is the idea in the Transmutation Trajectory Analysis method (TTA). According to the theory [3], the concentration of the k:th nuclide in a linear chain starting from nuclide 1 is given by

$$N_k(t) = \frac{N_1(0)}{\lambda_k} \sum_{i=1}^{k} \lambda_i \prod_{\substack{j=1 \\ j \neq i}}^{k} \left(\frac{\lambda_j}{\lambda_j - \lambda_i} \right) \exp(-\lambda_i t) \ , \tag{5}$$

where $N_1(0)$ is the concentration of the initial nuclide at $t = 0$. The advantage of the TTA method is that neither the extensive variation in the transmutation coefficients nor the length of the time step have any impact on the numerical accuracy of the calculation. The method is also relatively easy to implement in its basic form using a recursive loop.

The main problem with TTA, as it is currently implemented in the Serpent code, is that the method cannot treat chains that form a closed cycle. Equation (5) breaks down if the same nuclide is encountered twice in the chain, which occurs, for example, with consecutive (n,γ) and (n,2n) reactions. [†] Further, the overall calculation time easily becomes excessive if all trajectories are followed until a stable nuclide is encountered. Cut-offs have to be enforced to terminate unimportant chains, which complicates the calculation routines and may lead to unexpected results due to loss of data.

3.3.2. CRAM method

Perhaps the most popular approach to solving the Bateman equations (1) is to formulate the problem in matrix notation as

$$\mathbf{n}' = \mathbf{A}\mathbf{n} \; , \quad \mathbf{n}(0) = \mathbf{n}_0 \; , \tag{6}$$

where $\mathbf{n}(t) \in \mathbb{R}^n$ is the concentration vector and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the burnup matrix containing the decay and transmutation coefficients of the nuclides in the irradiated material. The general solution of this system can be written

$$\mathbf{n}(t) = e^{\mathbf{A}t}\mathbf{n}_0 \; , \tag{7}$$

where the exponential of the matrix $\mathbf{A}t$ is defined by the power series expression

$$e^{\mathbf{A}t} = \sum_{k=0}^{\infty} \frac{1}{k!} \left(\mathbf{A}\mathbf{t}\right)^k \tag{8}$$

and $\mathbf{A}^0 = \mathbf{I}$ is the identity matrix.

There are numerous algorithms for computing the matrix exponential but many of them are ill-suited for burnup calculations, where the decay and transmutation coefficients vary extensively. Short-lived nuclides are especially problematic because they can increase the matrix norm and induce eigenvalues of arbitrarily large magnitude. This characteristic as well as the length of the time step are the key aspects that should be taken into consideration when choosing the matrix exponential method.

Fortunately, all eigenvalues of the burnup matrix seem to generally remain bounded near the negative real axis. This observation is exploited in the CRAM method, which can be characterized as the best rational approximation on the negative real line and can further be interpreted as a numerical contour integral in the left complex plane [4]. This approach is optimal to the extent of allowing arbitrary large eigenvalues on the entire negative real axis and can therefore readily treat even the most short-lived nuclides simultaneously with the long-lived nuclides. In addition, the practical maximum time step value can be used in CRAM without compromising the computational accuracy. It is also computationally very effective making it an attractive method for solving the Bateman equations.

[†] Cetnar has derived an equation that takes into account the occurrence of closed loops [3]. The extended methodology will be included in the Serpent code within the near future.

4. EXAMPLE CALCULATIONS

The methodology in the Serpent code was tested in an LWR assembly burnup calculation. An OECD/NEA depletion calculation benchmark [9] was chosen as the basis for test case. The geometry is an infinite two-dimensional lattice of standard 17×17 UOX PWR fuel assemblies with 16 burnable absorber pins containing gadolinium. When each pin is treated as a separate depletion zone and burnable absorber pins divided into 10 annular sub-regions to account for the rim-effects, the total number of burnable materials becomes 65, taking into account the 1/8 symmetry of the configuration.

The fuel was irradiated at a constant 38.6 kW/kgU power density to 40 MWd/kgU burnup. The burnup step was 0.5 MWd/kgU at the early part of the irradiation cycle and 2.5 MWd/kgU after the excess gadolinium in the burnable absorber pins was completely burnt out. The total number of burnup steps was 42 and predictor-corrector calculation was run for each step. A total of 3 million active neutron histories (500 criticality cycles of 6000 neutrons) were run for each Monte Carlo simulation.

The Serpent calculations were carried out using ENDF/B-VI.8 and JEF-2.2 based cross section libraries and compared to reference CASMO-4E [10] results. Calculations were also carried out using Monteburns [11], but due to the limitations in the number of burnable materials, the results are compared separately.

4.1. Results

Figure 1 shows the infinite multiplication factor as function of fuel burnup, calculated using CASMO-4E and Serpent. It is shown that the results are consistent up to about 15 MWd/kgU burnup, after which there is a clear systematic and increasing over-prediction in k_∞. When the nuclide concentrations are compared, discrepancies originating from U-235 and the two main burnable absorber isotopes (Gd-155 and Gd-157) can be ruled out. The differences to CASMO-4E are practically negligible, as can bee seen in Figures 2 and 3.[‡] Instead, the discrepancies can be traced back to Pu-239 and two fission product poisons.

The comparison of Pu-239 concentrations in Figure 4 shows a constant difference in the build-up rates between CASMO-4E and Serpent. It is quite certain that this over-prediction results from the capture rate of U-238, since the same difference is seen in the concentration of this nuclide as well. This discrepancy could originate from the cross section data, or it could be related to differences in the flux spectrum.

The atomic densities of xenon-135 are plotted in Figure 5. There is a sudden change in the concentration calculated by CASMO-4E when the burnup step is increased from 0.5 to 2.5 MWd/kgU. This change originates from the special treatment used for equilibrium xenon calculation and the effect is reflected in the differences between the codes as well. This change could partially explain why the discrepancies in k_∞ begin to grow after the step size is increased at 15 MWd/kgU burnup.

[‡]The seemingly large differences at low concentrations result from the fact that, unlike CASMO-4E, Serpent does not separate burnable absorber and fission product gadolinium.

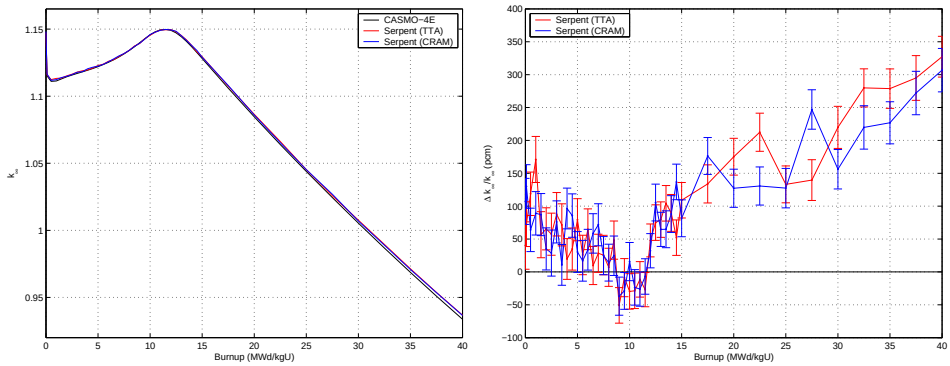


Figure 1. Infinite multiplication factor as function of fuel burnup (left) and relative differences compared to CASMO-4E. The errorbars represent one-sigma confidence intervals (ENDF/B-VI.8).

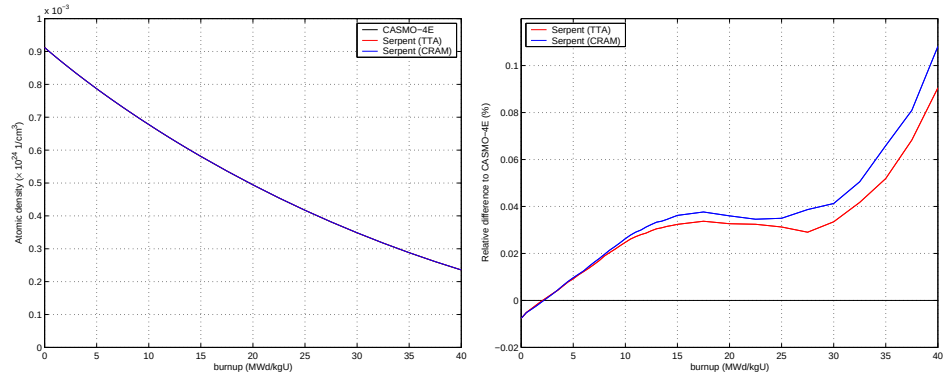


Figure 2. Assembly-averaged U-235 concentrations as function of fuel burnup (left) and relative differences compared to CASMO-4E (ENDF/B-VI.8).

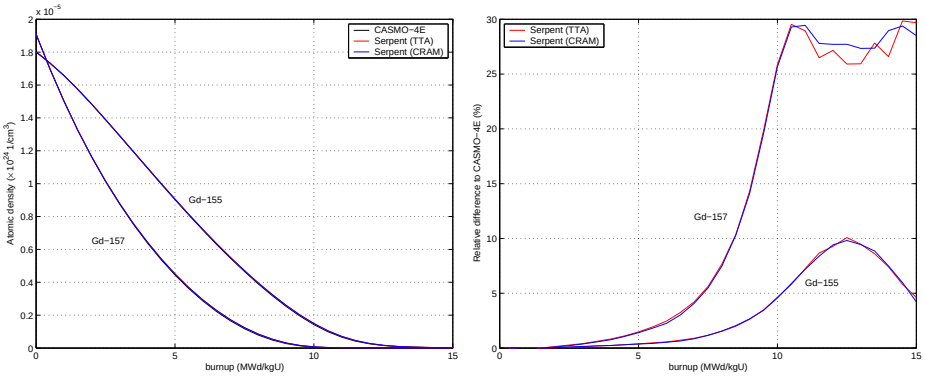


Figure 3. Assembly-averaged Gd-155 and Gd-157 concentrations as function of fuel burnup (left) and relative differences compared to CASMO-4E (ENDF/B-VI.8).

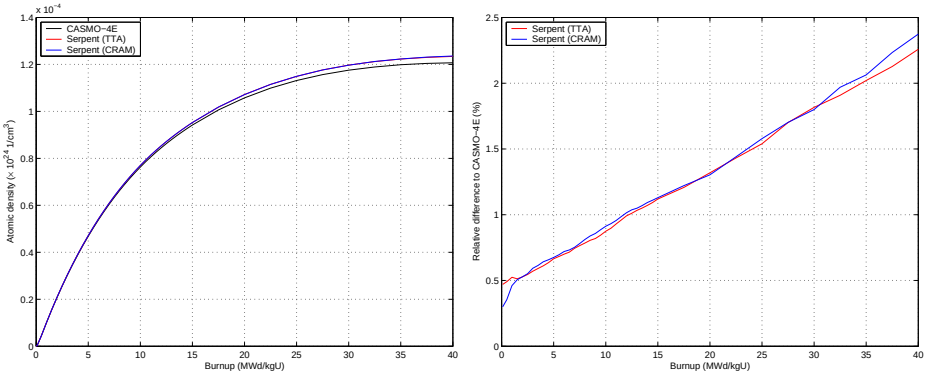


Figure 4. Assembly-averaged Pu-239 concentrations as function of fuel burnup (left) and relative differences compared to CASMO-4E (ENDF/B-VI.8).

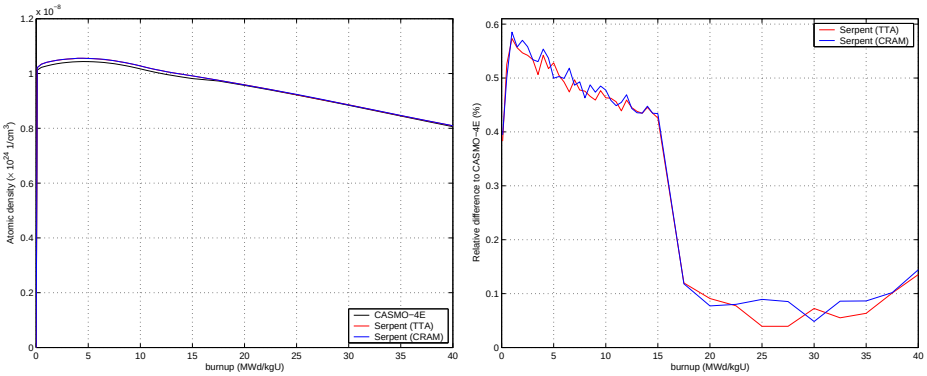


Figure 5. Assembly-averaged Xe-135 concentrations as function of fuel burnup (left) and relative differences compared to CASMO-4E (ENDF/B-VI.8).

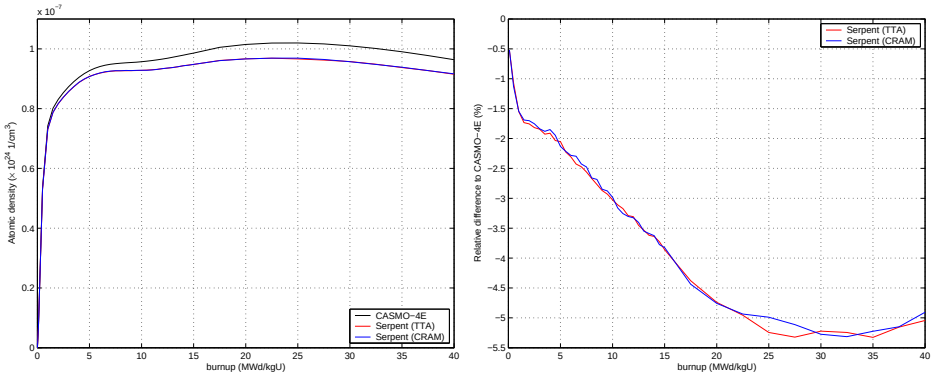
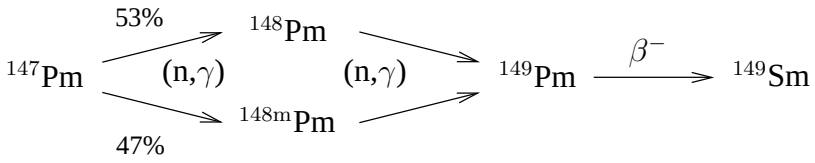


Figure 6. Assembly-averaged Sm-149 concentrations as function of fuel burnup (left) and relative differences compared to CASMO-4E (ENDF/B-VI.8).

The concentrations of samarium-149 are compared in Figure 6, and there is a clear difference in the saturated value. This nuclide is produced directly in fission and in the β decay of other fission products in the 149 mass chain. A significant fraction is also produced from promethium isotopes in a transmutation chain originating from Pm-147:



The fission yield of Pm-147 is relatively high and the yields of Pm-148 and Pm-148m practically negligible. The result is that the concentration of Sm-149 is somewhat sensitive to the capture rates in the promethium chain, and it turns out that the differences can be traced back all the way to the (n,γ) cross section of Pm-147, which is about 20% higher in the Serpent calculation.

It is not clear which of the above three is the dominant source of discrepancy for k_∞, or how the deviations are coupled by the flux spectrum. There may also be differences in the fission product yields, which makes the comparison even more complicated. It should also be noted that the neutron capture rate in gadolinium isotopes clearly exceeds all fission product poisons except Xe-135 when burnable absorbers are used. It is hence reasonable to assume that any differences begin to contribute only after the initial gadolinium has been completely burnt out, which is exactly what is seen in Figure 1.

The two fundamentally different solution methods used by the Serpent code give consistent results, which is an indication that the calculation routines work as expected. In fact, the only significant difference between TTA and CRAM is seen in the U-235 concentration in Figure 2. This difference is explained by the fact that the closed loops formed with U-234 and U-236 by consecutive (n,γ) and (n,2n) reactions are cut short in the TTA calculation.

The above results were calculated using ENDF/B-VI.8 based cross section libraries. The differences in k_∞ are smaller using JEF-2.2 data, but the same trends are clearly shown. The build-up rate of Pu-239 is slightly over-predicted by Serpent and the concentration of Sm-149 remains lower. The sudden change in the xenon equilibrium is also seen in the results. The fact that the observed discrepancies are independent of the data evaluation may suggest that they originate from the calculation method instead. It is clear that there are fundamental differences between deterministic and Monte Carlo neutron transport codes and it is only natural that these differences are reflected in the results as well.

4.2. Calculation Time

The Serpent burnup calculation using the CRAM method was completed in less than 19 hours on a single-processor 3.0 GHz Intel Xeon PC workstation. The overall CPU time is divided between calculation routines that can be associated with data processing, transport simulation and the solution of the depletion equations. The division is shown in Table I.

Table I. The overall CPU time (in minutes) divided between processes in the example PWR assembly burnup calculation using the Serpent code. 65 depletion zones, 42 burnup steps with predictor-corrector calculation, 3 million neutron histories in the Monte Carlo simulation.

	TTA		CRAM	
Data processing	348	(24%)	350	(31%)
Transport simulation	766	(53%)	766	(69%)
Depletion calculation	332	(23%)	3	(<1%)
Total	1446		1119	

Transport simulation takes the largest fraction of the overall CPU time. Like in all Monte Carlo calculation, this time depends mainly on the total number of neutron histories run. Most of the data processing time is used for calculating macroscopic material total cross sections after each burnup step.[§] The total CPU time is hence dependent on the number of irradiated materials as well. The difference between the two depletion methods is quite dramatic. The time spent for solving the depletion equations is practically negligible using the CRAM method, while the TTA solution uses a total of over 5 hours for the same task. It should be noted, however, that the TTA calculation time is strongly dependent on the cut-off criterion used for terminating unimportant transmutation chains. Shorter chains result in a shorter calculation time, but may induce larger errors in the final compositions. The selection of the appropriate cut-off criterion is one of the main difficulties in the TTA method and a somewhat conservative value was used in the test calculations.

4.3. Comparison to Monteburns

The Serpent calculations were also compared to Monteburns [11], a coupling code based on MCNP5 [12] and ORIGEN2 [13]. The comparison was carried out separately, mainly because a simplified geometry model had to be used due to the limitations in the number of burnable materials in the Monteburns calculation. Also the predictor-corrector methods used by the two codes are different, so the calculations were run without the corrector steps.

It turned out that there were some major discrepancies in the depletion of burnable absorber isotopes in the Monteburns results compared to both Serpent and CASMO-4E. The origin of these discrepancies remained unclear, but they most likely result from errors in the input files. Despite the failure to produce valid results, the calculations can be considered representative enough for the comparison of running times. When the number of histories per cycle was reduced from 6000 to 1000, the Monteburns calculation took about 40 hours to complete. The corresponding Serpent calculation time was only 67 minutes. This difference clearly shows how the overall running time of continuous-energy Monte Carlo codes can be reduced significantly by using methods dedicated

[§] Serpent pre-calculates material-wise total, total capture, elastic, (n,2n), fission, fission neutron production and total energy production cross sections to avoid the summation over material compositions during the transport cycle.

to burnup calculation. In fact, previous comparisons have shown a difference as high as a factor of 80 between Serpent and Monteburns.

5. SUMMARY AND CONCLUSIONS

The PSG2 / Serpent Monte Carlo reactor physics code has been developed at VTT for group constant generation and other lattice physics applications since 2004 [1, 2]. The methodology was recently extended to burnup calculation and the code is now capable of performing full-scale assembly burnup calculations as a completely stand-alone application. Serpent uses two fundamentally different methods for solving the Bateman equations: the Transmutation Trajectory Analysis method (TTA) [3], based on the analytical solution of linearized depletion chains and the Chebyshev Rational Approximation Method (CRAM) [4], an advanced matrix exponential solution developed at VTT.

Serpent was tested by comparing to the deterministic CASMO-4E code in a PWR assembly burnup calculation. The results showed a good consistency in k_∞ and average material compositions, although there were some systematic discrepancies as well. The observed differences in the build-up rates of Pu-239 and fission product poisons are most likely to originate from cross section data, or from the flux spectra used for calculating the one-group transmutation coefficients. The two codes also use different methods for equilibrium xenon calculation and there are probably differences in the fission product yields as well. More comparison calculations are needed for thoroughly resolving the origins of these discrepancies. It should be noted, however, that the differences between two such fundamentally different calculation methods may never be reduced to zero. This, of course, is not even the main goal when developing new calculation techniques.

The two methods used for solving the depletion equations in the Serpent code showed remarkable consistency, which is an indication that there are no major flaws in the main calculation routines. The CRAM method can be considered superior in terms efficiency, but the difference is likely to reduce if a less conservative cut-off criterion is used for terminating the unimportant TTA chains. The current implementation of the TTA method cannot treat chains that form closed loops. Although this flaw produced only minor differences in the results, an improved calculation routine will be introduced in the Serpent code in the near future.

The PWR assembly burnup calculation was completed in less than 19 hours when the Serpent code was run on a 3.0 GHz single-processor PC workstation. This calculation time is not comparable to deterministic codes, but it clearly shows that the Monte Carlo method can be used for realistic full-scale assembly burnup calculations, without having to wait for several days or weeks for the results. This offers interesting possibilities for the future as well, especially since Monte Carlo codes are not restricted to a single fuel or reactor type and the use of continuous-energy cross sections brings the best available knowledge on neutron interactions directly in the transport calculation.

REFERENCES

[1] J. Leppänen, *Development of a New Monte Carlo Reactor Physics code*, D.Sc. thesis, Helsinki University of Technology (2007). VTT Publications 640.

[2] J. Leppänen, *PSG2 / Serpent - A Monte Carlo Reactor Physics Burnup Calculation Code, User's Manual (February 2, 2009)*, VTT Technical Research Centre of Finland (2008).

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

[4] M. Pusa and J. Leppänen, "Computing the Matrix Exponential in Burnup Calculations," Article submitted to *Nucl. Sci. Eng.*

[5] E. R. Woodcock et al., "Techniques Used in the GEM Code for Monte Carlo Neutronics Calculations in Reactors and Other Systems of Complex Geometry," *Proceedings of the Conference on Applications of Computing Methods to Reactor Problems*, ANL-7050. Argonne National Laboratory (1965).

[6] J. Leppänen, "Two Practical Methods for Unionized Energy Grid Construction in Continuous-Energy Monte Carlo Neutron Transport Calculation," Article submitted to *Ann. Nucl. Energy*.

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

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

[9] B. Roque et al., *Specification for the Phase 1 of a Depletion Calculation Benchmark devoted to Fuel Cycles*, NEA/NSC/DOC(2004)11, OECD/NEA (2004).

[10] J. Rhodes, K. Smith and M. Edenius, "CASMO-4E Extended Capability CASMO-4 User's Manual". SSP-01/401, Studsvik Scandpower (2001, proprietary).

[11] D. I. Poston and H. R. Trellue, *User's Manual, Version 2.0 for Monteburns, Version 1.0*, LA-UR-99-4999, Los Alamos National Laboratory (1999).

[12] X-5 Monte Carlo Team. *MCNP – A General Monte Carlo N-Particle Transport Code, Version 5*, LA-CP-03-0245, Los Alamos National Laboratory (2003).

[13] A. G. Croff. *A User's Manual for the ORIGEN2 Computer Code*, ORNL/TM-7175, Oak Ridge National Laboratory (1980).