DEVELOPMENT OF A DYNAMIC SIMULATION MODE IN SERPENT 2 MONTE CARLO CODE

Jaakko Leppänen
VTT Technical Research Centre of Finland
P.O. Box 1000 FI-02044 VTT, Finland
Jaakko.Leppanen@vtt.fi

ABSTRACT

This paper presents a dynamic neutron transport mode, currently being implemented in the Serpent 2 Monte Carlo code for the purpose of simulating short reactivity transients with temperature feedback. The transport routine is introduced and validated by comparison to MCNP5 calculations. The method is also tested in combination with an internal temperature feedback module, which forms the inner part of a multi-physics coupling scheme in Serpent 2. The demo case for the coupled calculation is a reactivity-initiated accident (RIA) in PWR fuel.

Key Words: Serpent, dynamic Monte Carlo, multi-physics, temperature feedback, RIA

1. INTRODUCTION

The Serpent Monte Carlo reactor physics burnup calculation code has been in public distribution since 2009, and has users in 84 universities and research organizations in 28 countries around the world. Most of the user applications are related to either fuel assembly level calculations, such as group constant generation or fuel depletion, or the modeling of small research reactor cores. Main focus in code development, however, has moved from stand-alone neutronics simulations to multi-physics applications, involving a two-stage calculations scheme [1] consisting of an internal temperature feedback module and a universal interface for external coupling to computational fluid dynamics (CFD), thermal hydraulics system codes and fuel performance codes. The new applications also require development in the basic calculation routines, such as the introduction of time-dependence in neutron transport simulation – a feature that has until recently been completely missing in the Serpent code.

This paper presents the work carried out for the implementation of a time-dependent (dynamic) neutron transport mode in the development version of the code, Serpent 2. The methods are based on a standard external source simulation, with sequential population control for the modeling of prompt super-critical systems. Some theoretical background on time-dependent simulations is given in Section 2, before introducing the implemented methodology in Section 3. The methods are validated by comparison to MCNP5 calculations in Section 4. To make things more interesting, the new simulation mode is also coupled to the internal temperature feedback module in an effort to simulate a dynamic response to a prompt super-critical power excursion in PWR fuel.

2. BACKGROUND

Monte Carlo reactor physics calculations are typically run as criticality source simulations, which closely resemble a self-sustaining fission chain reaction. Sub-critical and non-multiplying systems can also be

simulated as external source problems, which is typical for radiation shielding applications. Even though the operations performed for individual neutron histories are practically identical in the two simulation modes, there are some considerable differences in the way the neutron population is handled as a whole. These differences, and their implications for time dependence, are briefly discussed in the following.

2.1. Time Dependence in *k*-eigenvalue Criticality Source Simulation

Neutron transport simulation in the criticality source mode is run in cycles, or generations, in which the source distribution is iteratively regenerated from the fission neutron distribution of the previous cycle. The iteration is started from an initial guess and the average population size is maintained either by scaling the source term with the criticality eigenvalue, or by applying population control on the banked source neutrons before the next cycle is run. Since the simulation proceeds from one fission event to the next, neutron histories are not bound in time, which in turn implicates that flux and reaction rates are assumed to remain constant over any given time interval. The simulation mode essentially corresponds to solving the k-eigenvalue form of the static transport equation:

$$\hat{\mathbf{\Omega}} \cdot \nabla\Phi + \Sigma_{\mathrm{tot}}\Phi = S + \frac{1}{k}F\,, \tag{1}$$

where the scattering and fission source terms are written in simplified form and all variables are omitted. The k-eigenvalue criticality source method always simulates the chain reaction without time dependence. If the system is not inherently in a steady state condition (i.e. $k \neq 1$), it is forced into one by adjusting the fission source.

2.2. Time Dependence in External Source Simulation

All neutron histories in an external source simulation are started from the same user-defined source distribution, which fixes the initial position, direction, energy, weight and time of each starting neutron. Since the histories are not batched into generations, the simulation can be divided into explicit time intervals. In deterministic transport theory, the simulation mode corresponds to solving the time-dependent form of the transport equation:

$$\frac{1}{v}\frac{\partial}{\partial t}\Phi + \hat{\mathbf{\Omega}} \cdot \nabla\Phi + \Sigma_{\mathrm{tot}}\Phi = S + F + G\,, \tag{2}$$

where G is the external source. There is no population control applied to neutron histories, and fission reactions are explicitly simulated by branching the history. All histories eventually die out in sub-critical systems, but in the case of super-criticality, the histories tend to develop an infinite number of branches. This, in turn, means that time cut-off must be enforced to terminate the simulation. Since the growth rate is exponential, the simulation of prompt super-critical reactivity excursions is practically limited to very short periods of time.

3. DYNAMIC SIMULATION MODE IMPLEMENTED IN SERPENT 2

Even though Serpent 1 and earlier versions of Serpent 2 include time as one of the neutron state variables, the value is only used for calculating analog estimates for certain time constants. Development of an actually time-dependent simulation mode started with version 2.1.9, in which time cut-off was added in external source mode and time binning included in detectors (tallies). As discussed in the previous section,

this type of approach is sufficient for modeling sub-critical systems and short time periods in super-critical configurations. The main purpose of the methods introduced in the following is to extend the capability to prompt super-critical reactivity excursions, where population size may increase by several orders in magnitude. The methodology was implemented in code version 2.1.10.

3.1. Time-dependent Simulation with Sequential Population Control

If the fission chain reaction in a super-critical system is allowed to diverge, some form of population control must be applied on the neutron histories. This procedure must be carried out in such way that the total weight is preserved, and no biases are introduced in energy, space or time.* In practice this means that a statistically representative sample of limited size is kept continuing the simulation, and excess neutrons are discarded. The most obvious way to accomplish this is to divide the simulation into a number of discrete time intervals. Neutrons that reach the end of the interval are banked, similar to fission neutrons in a criticality source simulation. After all histories have completed the interval, population control is applied on the banked neutrons. Randomly selected neutrons are discarded, until the population size matches the initial source size at the beginning of the simulation. Total weight of the population is preserved by re-normalizing the weights before the next interval is run.

Since the population is allowed to increase within each interval, the number of scores, and therefore the statistical accuracy depends on the interval length. Increasing the number of intervals over a given time period may considerably reduce the accuracy in a super-critical system, since a larger fraction of neutron histories are terminated by population control before developing branches. For the same reason the simulation is completed faster, and the loss of accuracy can be compensated by increasing the size of the starting population. The relation between statistics and interval length is not straightforward.

In addition to super-critical systems, the same procedure can also be applied to a sub-critical chain reaction. Population control then duplicates randomly selected neutrons, until the initial source size is matched. The effect of time intervals on statistical accuracy is the opposite compared to super-critical systems: increasing the number of intervals improves the statistics, but increases the overall running time. Even though not necessary for completing a sub-critical simulation, this method is convenient when used for transient systems, where the criticality state changes from super- to sub-critical over time.

The development of the dynamic transport mode in Serpent 2 is currently limited to fast transients and prompt neutrons. Delayed neutron emission considerably extends the time scale of the simulations, which requires special techniques for handling prompt fission chains originated from the decay of delayed neutron precursors [3].

3.2. Normalization

The absolute values of all tallies in a Monte Carlo simulation depend on the number neutron histories run. The ratio of any two tallies, however, remains constant (to within statistics), regardless of the history count. The results can therefore be normalized by fixing the numerical result of a single tally to a user-given

*This may seem like a trivial task, but in reality, even the conventional k-eigenvalue criticality source simulation is inherently biased if the system is not critical. As pointed out by Cullen et al. [2], these biases result from the fact that neutrons are not selected to population control in a completely random manner. The problem is also encountered in all deterministic methods solving the k-eigenvalue form of the transport equation, and it originates from the assumption that a non-steady state system can be modeled without time dependence.

value. A typical normalization for reactor calculations is to fix reactor power to total energy production tally, after which all other tallies can be normalized by multiplication with the ratio of the two values. Serpent performs this type of normalization automatically.

The situation becomes a bit more complicated when the simulation involves time dependence. The value of the total energy production tally in a super-critical system, for example, depends on time cut-off. In order to apply the conventional normalization, the user needs to know the total energy released during the transient, which may be just what the simulation was set up to calculate. In the dynamic simulation mode developed for Serpent 2 this problem is overcome by simply fixing the normalization during the first simulated time interval, which represents the initial state. Reactor power, total flux, reaction rates, etc., are then allowed to increase or decrease in time, depending on the criticality state of the system.

3.3. Coupling to Internal Temperature Feedback Module

The main intended use for the dynamic mode is the simulation of fast reactivity transients in combination with the internal temperature feedback module [4], currently under development in Serpent 2. This routine solves the temperature profile inside cylindrical fuel pins, based on local power, user-provided heat transfer data and a fixed boundary condition at the cladding-coolant interface. In a criticality source simulation the temperature distributions are solved after each cycle, and the solution passed back to neutronics for the next cycle. The methodology takes advantage of a built-in on-the-fly temperature treatment routine [5], which allows material temperature distributions to be modeled as arbitrary continuous functions of the spatial coordinates.

When applied to a dynamic simulation, the iteration between heat transfer and neutronics is not performed between static criticality cycles, but rather between two consecutive time intervals. The simulation starts from the initial state, which also fixes the normalization. Since tally calculation involves integration over time, fission power and fuel temperatures are assumed to remain constant over each time interval. The power produced during the previous interval is used for solving the temperature distribution for the next interval. The errors induced by this approximation depend on how the interval lengths relate to the time rate of change in the system, as will be demonstrated by the example in Section 4.2.[†]

4. TEST CALCULATIONS

The test calculations presented in this section serve two purposes: 1) to validate the time-dependent transport routines implemented in Serpent 2.1.10, and 2) to demonstrate how the new dynamic simulation mode can be used in combination with the internal temperature feedback module to simulate fast reactivity transients.

4.1. Validation by Comparison to MCNP5

Since Serpent reads neutron interaction data in the form of ACE format cross section libraries, the most natural choice as a reference code for validation purposes is MCNP [6], which is also capable of running time-dependent external source simulations. Sharing the same cross section libraries practically eliminates

[†]It should be noted that the temperature feedback module is currently based on a steady-state solution of the heat conductivity equations, which in a transient case essentially corresponds to infinitely fast heat transfer inside the fuel pin. The example calculation presented in Section 4.2 is included in this paper merely as a test case for the neutronics.

all data-originated uncertainties, and any possible differences in the results can be attributed to methodological flaws.

Two sets of comparison calculations were carried out, representing fast and thermal systems, respectively. The experimental configurations were selected from the International Handbook of Evaluated Criticality Safety Benchmark Experiments [7], with slight modifications to induce sub- and prompt super-critical conditions. Neutron population size as function of time was calculated with Serpent 2.1.10 and MCNP5, using a $1/v$ response function for a standard cell flux tally. All calculations were carried out with ENDF/B-VII based cross section libraries. Delayed neutron emission was switched off, which is the default in MCNP external source simulations. Since the shortest precursor half-lives are in the range of 50-100 ms, this approximation is not believed to have any effect on the results.

4.1.1. Fast system

The configuration representative of a fast system is a natural-uranium reflected plutonium sphere (Flattop-Pu, benchmark PU-MET-FAST-006 in [7]). Sub-critical condition was induced by reducing the reflector thickness by 60%, and prompt super-critical condition by increasing the core radius by 5%. The three variants are summarized in Table I. A total of 2 million source neutrons were run in the external source simulations with Serpent 2.1.10 and MCNP5. The dynamic simulation was divided into 10 time intervals, and run with 20 and 40 million neutron histories in the critical and super-critical cases, respectively, to produce similar level of statistical accuracy as the two reference calculations (see Sec. 3.1). Initial source was an isotropic 1 MeV point source, placed in the core center. Time cut-off was set to 400 ns ($\sim$10 prompt neutron lifetimes), and neutron population was tallied in 500 time bins of 0.8 ns each.

The results are plotted in Figure 1. After a short initial transient peak, the systems converge in their fundamental modes. Population saturates at a constant level in the critical case, while an exponential increase and decrease is observed in the super- and sub-critical cases. The curves produced by MCNP5 and Serpent 2.1.10 overlap, and a closer comparison reveals that the differences are within the range of statistical accuracy. The two Serpent calculations also produce consistent results, which suggests that the sequential population control performed after each time interval preserves the statistical behavior.

4.1.2. Thermal system

Time dependence in a thermal system was tested by modeling an unreflected cylindrical tank of uranium nitrate solution (STACY-30 experiment, benchmark LEU-SOL-THERM-007 in [7]). Criticality was adjusted by changing the surface level, as summarized in Table II. A total of 1 million source neutrons were run in the external source simulations with Serpent 2.1.10 and MCNP5. In the dynamic mode, source size was increased to 2 and 4 million for the critical and super-critical cases, respectively, and the simulation was divided into 10 time intervals. Initial source was an isotropic 1 eV point source, placed in the center of the cylindrical container, 10.0 cm from the bottom. Time cut-off was set to 3 ms ($\sim$7 prompt neutron lifetimes), and neutron population was tallied in 500 time bins of 6 μs each.

The results are plotted in Figure 2. It appears that convergence to fundamental mode takes a bit longer than the simulated period. The criticality calculation presented in Table II estimated the critical configuration to be sub-critical by about 500 pcm, which might also explain why the population does not saturate to a constant level. In any case, Serpent results are in a good agreement with both MCNP5 and each other.

Table I. Fast-spectrum test cases in the Serpent 2 / MCNP5 comparison. Geometry is from benchmark PU-MET-FAST-006 in [7]. Criticality and time constants were calculated using MCNP5.

Configuration	Modification	Calculated k_{eff}	Calculated l_{p}
Critical	None	0.99922 ± 0.00005	64.12 ± 0.01 ns
Sub-critical	Reflector thickness -60%	0.95516 ± 0.00009	21.47 ± 0.03 ns
Prompt super-critical	Core radius +5%	1.04298 ± 0.00010	64.63 ± 0.01 ns

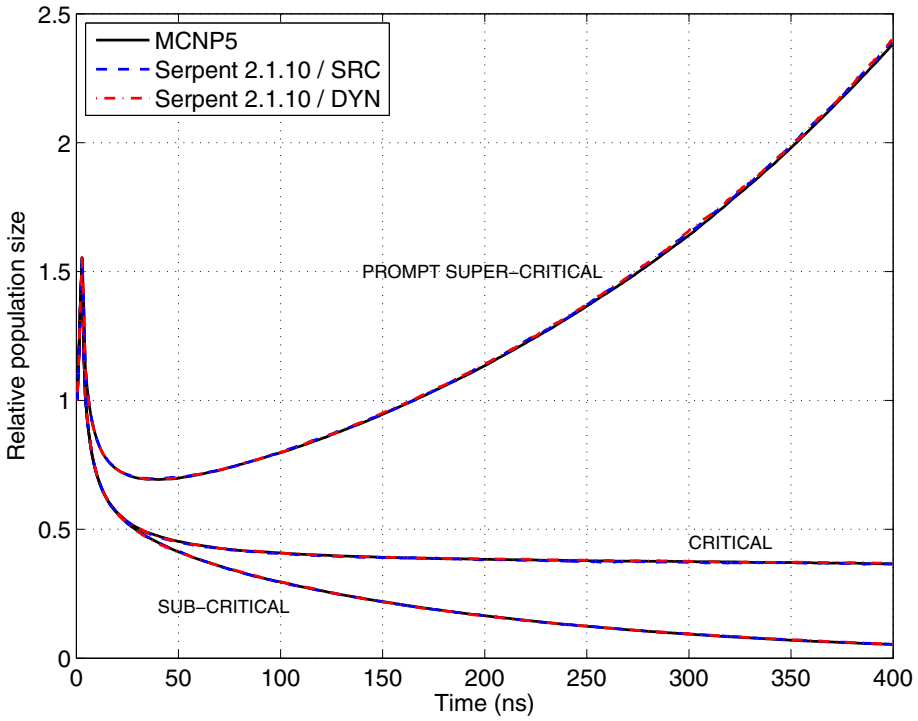

Figure 1. Neutron population inside the plutonium sphere in the Flattop-Pu experiment as function of time. Base case and sub- and prompt super-critical variants. Two sets of Serpent calculations were run: conventional external source simulation with time cut-off (SRC) and dynamic simulation divided into 10 equally-spaced time intervals (DYN).

Table II. Thermal-spectrum test cases in the Serpent / MCNP5 comparison. Geometry is from benchmark LEU-SOL-THERM-007 in [7]. Criticality and time constants were calculated using MCNP5.

Configuration	Modification	Calculated k_{eff}	Calculated l_{p}
Critical	None	0.99463 ± 0.00010	0.420 ± 0.007 ms
Sub-critical	Surface level -10.0 cm	0.94869 ± 0.00010	0.408 ± 0.007 ms
Prompt super-critical	Surface level +10.0 cm	1.02189 ± 0.00010	0.427 ± 0.006 ms

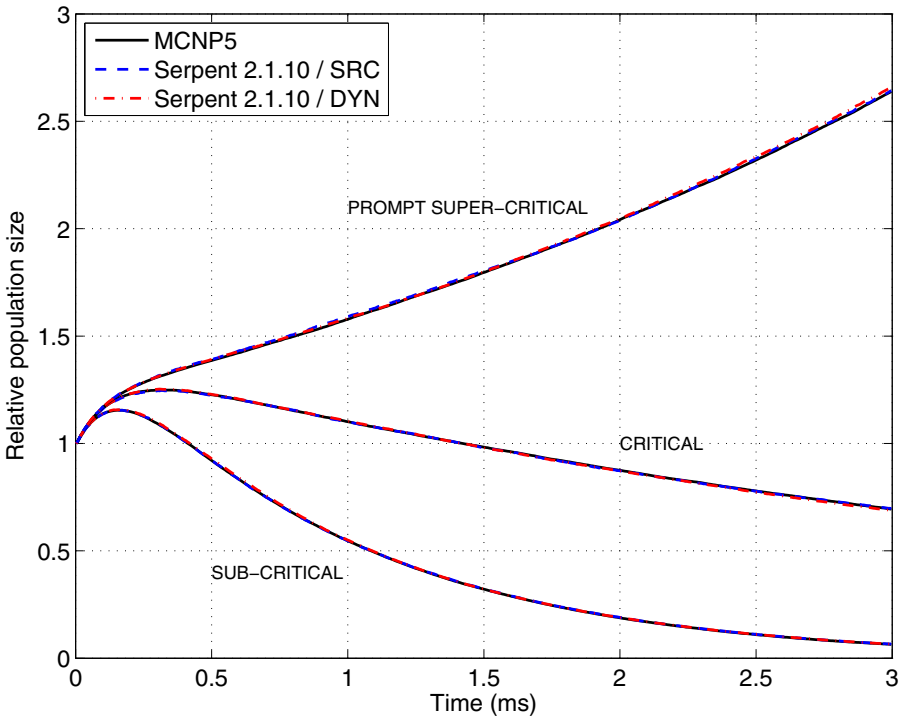

Figure 2. Neutron population in the uranium nitrite solution in the STACY-30 experiment as function of time. Base case and sub- and prompt super-critical variants. Two sets of Serpent calculations were run: conventional external source simulation with time cut-off (SRC) and dynamic simulation divided into 10 equally-spaced time intervals (DYN).

4.2. RIA Demo

The new dynamic calculation mode with temperature feedback was put to test in a simulation of a PWR reactivity-initiated accident (RIA). The methodology is still under development, and the calculations presented in the following are subject to three approximations:

1. Thermal hydraulics feedback is ignored and a fixed boundary condition is applied at the cladding-coolant interface
2. Delayed neutron emission is ignored
3. Heat transfer equations are solved in static form, which essentially means that the generated heat is dissipated instantaneously throughout the system.

Even though the first two approximations can be justified by the short duration (10 ms) of the transient, the third one can not. The example calculation should therefore be viewed as a test case for the new simulation mode, rather than a predicted physical response to a reactivity transient.

The geometry model in the test case is a 2-dimensional 17×17 PWR fuel assembly of the EPR type [8]. The initial state represents a critical hot zero-power condition, in which the coolant is at 0.7 g/cm^3 density and all materials are at 327°C (600 K) temperature. Coolant boron concentration required to keep the system at criticality is about 3500 ppm. In this state the system has a positive reactivity coefficient for coolant boiling, and the simulated transient is initiated by reducing the coolant density by 50% at the beginning of the time-dependent simulation. The reactivity insertion is not only instantaneous, but also extremely large, in the order of 4400 pcm, or 6.0$. Prompt neutron lifetime in the system is about 0.014 ms, which (according to a point-kinetic approximation) means that the population size doubles in every 0.2 ms.

The calculation was performed in two parts. A criticality source simulation was first run for the critical configuration to produce a source distribution for the initial state of the dynamic simulation, which was then run for a period of 10 milliseconds. All fuel pins in the assembly were treated as a single material, divided into 10 annular regions with equal volume for the calculation of local power distribution. A simple heat transfer model based on fixed thermal conductivities was used with the internal temperature feedback module.[‡] A total of 2 million neutron histories were run, and the calculation was repeated using 20, 40, 60 and 80 intervals for the dynamic simulation.

The results are plotted in Figures 3 and 4. It is seen that the exponential power excursion initiated by the reactivity insertion at $t = 0$ is terminated by a feedback effect after about 5 milliseconds. Linear pin power saturates at 500 W/cm and fuel center-line temperature to just below 1800°C. Population size increases by a factor of 2200, which means that a conventional external source simulation would be a very impractical method for running the calculation. Dividing the transient into only 20 time intervals creates some oscillation in the curves, as the simulation is unable to keep up with the changing reactivity. Increasing the number damps the oscillations, and after 80 time intervals there was no visible change in the results.

[‡]Heat transfer correlations in the internal temperature feedback module are compared in another paper submitted to this conference [9]

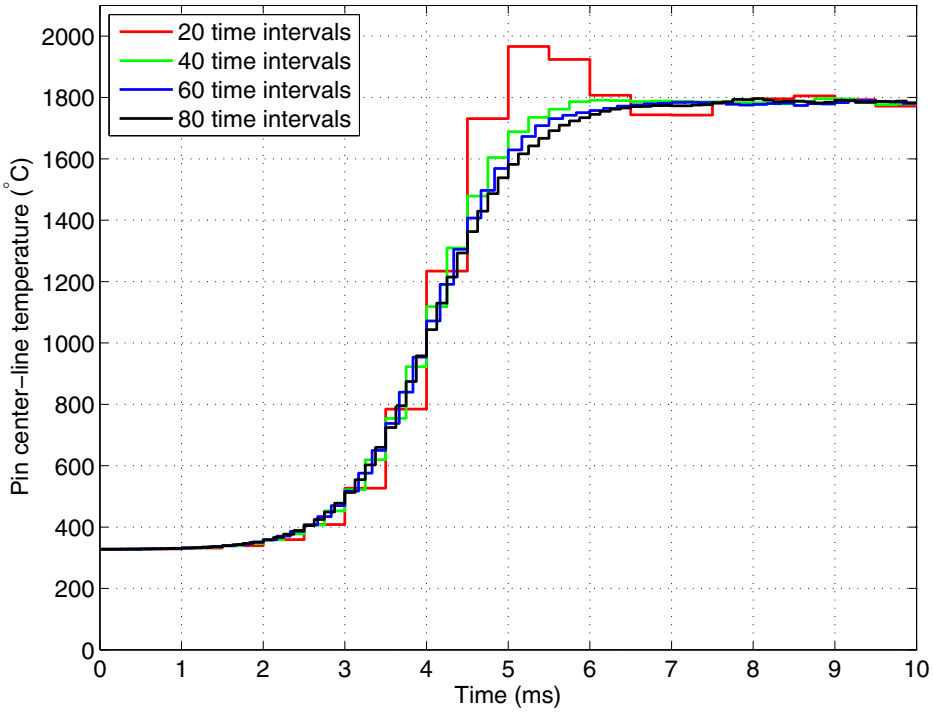

Figure 3. Fuel center-line temperature as function of time in the RIA calculation.

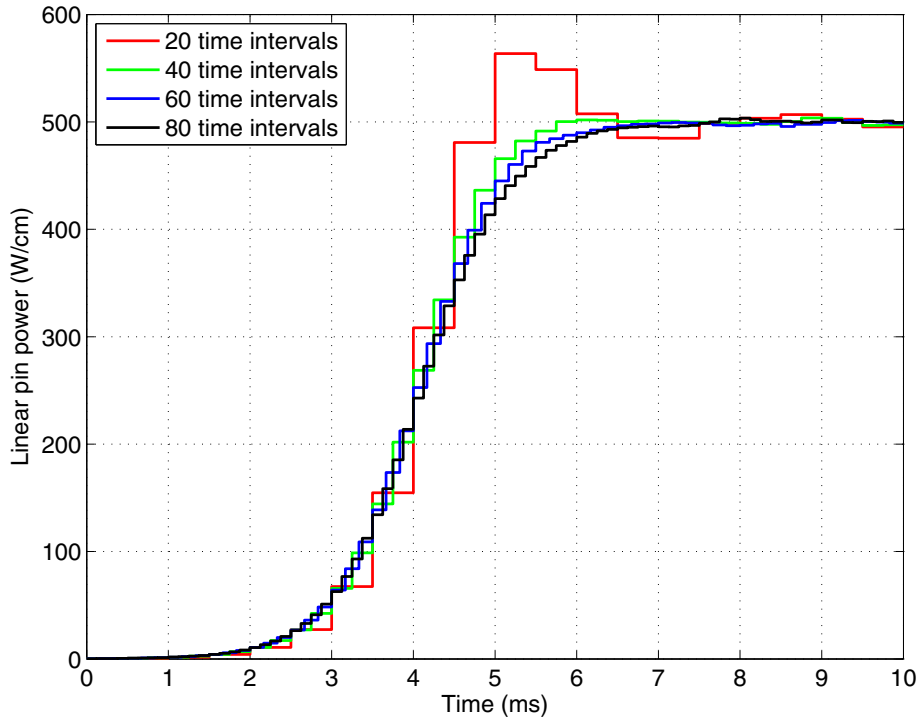

Figure 4. Linear pin-power as function of time in the RIA calculation.

Solving the heat conductivity equations takes less than 0.1% of the overall calculation time, so there is no significant time penalty from increasing the number of intervals. The statistics of the pin-power tallies are affected, however, as the scores are divided between a larger number of bins. Further analysis on running times and statistics is excluded in this paper, because the on-the-fly temperature treatment routine used by the feedback module is currently being optimized for performance [10], and any numbers presented here would soon become obsolete. The topic is revisited after the work is complete, and the heat conductivity model has been extended to time-dependent systems.

5 SUMMARY AND CONCLUSIONS

Considerable effort in the future development of the Serpent Monte Carlo code is devoted to multi-physics applications. The methodology is based on a two-stage calculation scheme, in which temperature feedback inside fuel pins is handled by an internal calculation routine and thermal hydraulics solution provided by external coupling. This paper presented the implementation of a dynamic neutron transport mode, which is a necessity for simulating reactivity transients with feedback effects. The methodology is based on standard external source simulation with sequential population control for prompt super-critical systems. The time-dependent simulation modes were validated by comparison to MCNP5 calculations. A demonstration involving a reactivity-initiated accident in PWR fuel also showed that the new simulation mode can handle reactivity excursions in which the population size is increased by several orders of magnitude.

In order to become a practical simulation tool for fast reactivity transients, the internal temperature feedback module in Serpent 2 must be extended to handle time dependence in heat conduction. This work will begin within the near future. Even with time-dependent temperature feedback the simulation is limited by the approximations made on delayed neutron emission and thermal hydraulics. Extending the methodology to slow transients and fully coupled systems will require some considerable work on both neutronics and the external coupling. These topics are included in the long-term development plan of the Serpent 2 code.

ACKNOWLEDGMENTS

This work has been funded from the EU High Performance Monte Carlo Reactor Core Analysis (HPMC) project, the Numerical Multi-Physics (NUMPS) project of the Academy of Finland, and the KÄÄRME project under the Finnish National Research Programme on Nuclear Power Plant Safety, SAFIR-2014.

REFERENCES

[1] J. Leppänen, T. Viitanen and V. Valtavirta, *Multi-physics Coupling Scheme in the Serpent 2 Monte Carlo Code*, Trans. Am. Nucl. Soc. **107**, pp. 1165-1168 (2012).

[2] D. E. Cullen, *Static and Dynamic Criticality: Are They Different?*, UCRL-TR-201506, Lawrence Livermore National Laboratory, 2003.

[3] B. L. Sjenitzer and J. E. Hoogenboom, *A Monte Carlo Method for Calculation of the Dynamic Behaviour of Nuclear Reactors*, P. Nucl. Sci. Technol., **2**, pp. 716-721 (2011).

[4] V. Valtavirta, *Designing and implementing a temperature solver routine for Serpent*, M.Sc. Thesis, Aalto University, 2012.

[5] T. Viitanen and J. Leppänen, *Explicit treatment of thermal motion in continuous-energy Monte Carlo tracking routines*, Nucl. Sci. Eng., **171**, pp. 165-173 (2012).

[6] 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.

[7] *International Handbook of Evaluated Criticality Safety Benchmark Experiments*, NEA/NSC/DOC(95)03, OECD/NEA, 2010.

[8] G. Sengler, et al., *EPR Core Design*, Nucl. Eng. Design, **187**, pp. 79-119 (1999).

[9] V. Valtavirta, *Internal Neutronics-Temperature Coupling in Serpent 2 – Reactivity Differences Resulting from choice of Material Property Correlations*, In Proc. M&C 2013.

[10] T, Viitanen, *Optimizing the Implementation of the Explicit Treatment of Target Motion – How Fast Can it Get?*, In Proc. M&C 2013.