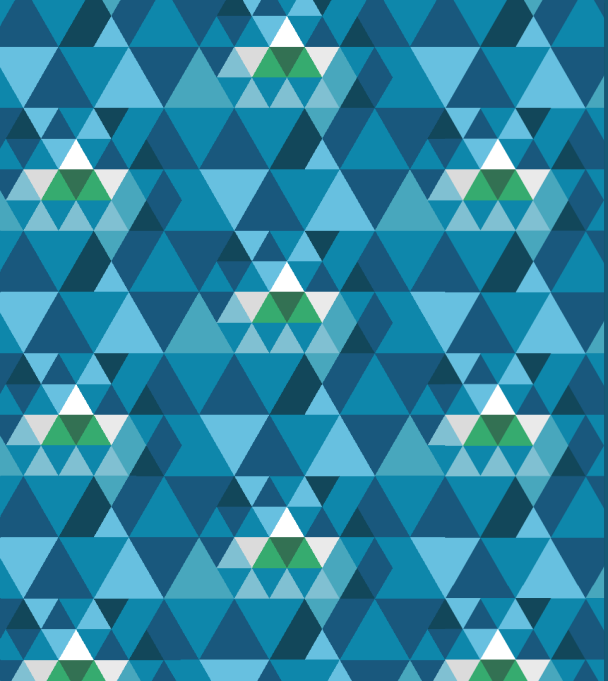


Past, present and future challenges of developing the Serpent Monte Carlo Code

Seminar at the National Centre for Nuclear Research

Otwock - Świerk, Poland, 7.10.2019

Jaakko Leppänen, Research Professor, VTT

Outline

- Part I: Challenges related to the different stages of Serpent development
  - Background
  - PSG – the predecessor of Serpent
  - Serpent 1
  - Serpent 2
- Part II: Current status and future work
  - Serpent team and user community
  - Development of multi-physics capabilities
  - Kraken computational framework
  - Expanding to new applications

Serpent: Overview

Serpent is a continuous-energy Monte Carlo transport calculation code:

- Developed at VTT Technical Research Centre of Finland, Ltd since 2004
- Originally intended for lattice physics calculations, in particular spatial homogenization
- Built-in burnup calculation capability
- Most typical applications: group constant generation, fuel cycle studies, research reactor modeling
- Considerable effort has been devoted to coupled multi-physics applications (fuel performance, thermal hydraulics and CFD code coupling)
- In resent years the scope of applications has been broadened beyond reactor physics, in particular to radiation transport and fusion neutronics

VTT

Part I: Challenges related to the different stages of Serpent development

Background

The idea of developing a new Monte Carlo code dedicated to spatial homogenization came from practical need:

- In 2003–2004 a small research project at VTT involved transient simulations for a super-critical water cooled reactor using the TRAB3D (in-house) nodal diffusion code
- The calculations required homogenized group constants, which could not be produced using the conventional deterministic lattice physics codes at our disposal
- Attempts of using MCNP for this task were not very successful
- Monte Carlo burnup calculation was still a new thing and subject to serious limitations, so generating group constants for burned fuel using MCNP was unthinkable

The project was dropped (I think?), but it left a few ideas on how to make things more efficiently with continuous-energy Monte Carlo, if the scope of applications was narrowed down to group constant generation.

Background

Monte Carlo simulations are typically used for solving complicated problems that can be divided into multiple simple sub-tasks.

Radiation and particle transport problems are a very good example:

- Solving the particle flux (transport equation) is a complicated task
- Simulating individual particle histories is relatively straightforward

In practice, Monte Carlo particle transport simulation requires:

i) Geometry routine (simple linear algebra and vector calculus)

ii) Physics model, handling the individual interactions with target nuclides (collision kinematics and a lot of data)

iii) Collecting the results (simple statistics)

When divided into small enough sub-tasks, this problem does not seem too difficult at all!

PSG – the predecessor of Serpent

The development started at around September 2004, and the predecessor of Serpent was called PSG ("Probabilistic Scattering Game").[1]

At the beginning the work was based on several simplifications:

- The geometry model was limited to very simple 2D fuel assembly geometries
- Particle tracking was handled using the Woodcock delta-tracking method
- Interaction data was read from ACE format data libraries using a common energy grid for all nuclides
- Only the most important reaction modes (fission, capture, elastic scattering) were included
- Built-in routines for group constant generation, no user-defined tallies

Some of these features, originally implemented for the sake of simplicity, later turned out to be quite efficient and are still used in Serpent today.

Some details are described in the extra slides at the end of this presentation.

[1] To be precise, there was a short-lived predecessor of PSG, simply called "MC".

The idea of using the continuous-energy Monte Carlo method for spatial homogenization seems relatively simple at first.

Calculation of homogenized few-group cross sections involves two flux integrals:

$$\Sigma_g = \frac{\int_V \int_{E_g}^{E_{g-1}} \Sigma(\boldsymbol{r}, E)\phi(\boldsymbol{r}, E) dV dE}{\int_V \int_{E_g}^{E_{g-1}} \phi(\boldsymbol{r}, E) dV dE} \tag{1}$$

which can be evaluated using standard cell flux tallies. Assembly discontinuity factors (ADF's) can also be obtained using standard Monte Carlo techniques.

Scattering matrices can be calculated using analog estimators for the group-to-group scattering rate (by counting the number of simulated events).

Methods exist for calculating β_{eff} (e.g. the Meulekamp method[2] and later IFP[3]).

[2] R. K. Meulekamp and S. C. van der Marck. "*Calculating the Effective Delayed Neutron Fraction with Monte Carlo.*". Nucl. Sci. Eng. **164** (2006), 142–148.

[3] J. Leppänen et al. "*Calculation of Effective Point Kinetics Parameters in the Serpent 2 Monte Carlo Code.*". Ann. Nucl. Energy **65** (2014), 272–279.

What became a major issue, however, was the calculation of diffusion coefficients:

- Diffusion coefficient is defined by an approximation (Fick's law):

$$\mathbf{J}_g(\boldsymbol{r}, t) = -D_g \nabla \phi_g(\boldsymbol{r}, t) \tag{2}$$

  without direct connection to any continuous-energy reaction cross section
- The methodology has to be consistent with leakage models and ADF's
- Several methods were tried, but the results were not completely satisfactory
- Eventually it took several years before the calculated diffusion coefficients would yield satisfactory results with nodal diffusion codes[4]
- New methods are still being developed for Serpent 2

One of the problems was that diffusion theory and nodal diffusion methods are poorly described in text books! Some of the methods used in deterministic codes have not been published until fairly recently.[5]

---

[4] J. Leppänen, M. Pusa, and E. Fridman. "*Overview of methodology for spatial homogenization in the Serpent 2 Monte Carlo code.*". Ann. Nucl. Energy **96** (2016), 126–136.

[5] K. Smith. "*Nodal diffusion methods and lattice physics data in LWR analyses: Understanding numerous subtle details.*". Ann. Nucl. Energy **101** (2017), 360–369.

PSG – the predecessor of Serpent

PSG was first introduced at the XII Meeting on Reactor Physics in the Nordic Countries in Halden, Norway, in May 2005:[6]

- Four test cases: BWR, VVER-440 and mixed MOX/UOX assembly calculations, and the Godiva sphere
- Calculation of k_{eff} and homogenized few-group constants
- Comparison to MCNP and CASMO-4E
- The calculation was still missing a lot of physics, but the differences to MCNP were generally below a few percent

Another early paper was presented at the M&C 2005 conference.[7] The work became the topic of a DSc thesis, completed in 2007.[8]

---

6 J. Leppänen. "*Diffusion Code Group Constant Generation Using the Monte Carlo Method.*". In proc. XII Meeting on Reactor Physics in the Nordic Countries. Halden, Norway, May 17–18, 2005.

7 J. Leppänen. "*A New Assembly-Level Monte Carlo Neutron Transport Code for Reactor Physics Calculations.*". In proc. M&C 2005. Avignon, France, Sept. 12-15, 2005.

8 J. Leppänen. "*Development of a New Monte Carlo Reactor Physics Code.*". DSc Thesis, Helsinki University of Technology, 2007.

By 2007, PSG had several features still included in Serpent:

- 3D universe-based geometry model
- Physics based on ENDF reaction laws (except for unresolved resonance probability table sampling)
- Unified energy grid with grid thinning
- Automated generation of group constants
- User-defined detectors (tallies)
- Geometry and mesh plotters
- Parallelization using MPI

The major deficiency was the lack of burnup calculation capability, which required re-writing the entire source code.

The new version was named Serpent (for no particular reason), and it was also decided to make this code available for public distribution.

The idea of expanding from transport to burnup calculations is straightforward, and basically involves sequential coupling between two solvers:

1) Solution of transport problem to obtain cross sections for transmutation and fission reactions
2) Solution of depletion problem to obtain material compositions for the next burnup step

There are efficient and accurate methods for the solution of the Bateman depletion equations, typically falling under two categories:

1) Analytic solution of linearized depletion chains
2) Matrix exponential solutions

The first solution implemented in Serpent was based on the linear chains method (or Transmutation Trajectory Analysis, TTA),[9] followed soon after by a new matrix exponential solution (Chebyshev Rational Approximation Method, or CRAM).[10]

[9] J. Cetnar. "*General Solution of Bateman Equations for Nuclear Transmutations*.". Ann. Nucl. Energy **33** (2006), 640–645.

[10] M. Pusa. "*Numerical Methods for Nuclear Fuel Burnup Calculations*.". DSc Thesis, Aalto University, 2013.

In practice things get much more complicated when burnup is involved, because of additional challenges:

- How to form the transmutation chains starting from the initial composition, and how to decide which cross sections to read?
- Where to obtain the additional data? – decay constants, fission yields and isomeric branching ratios are not included in ACE format data files
- How to handle isomeric branching (e.g. formation of $^{242m}\mathrm{Am}$)? – ACE data format does not even have an identifier for isomeric states
- How to calculate transmutation cross sections in the most efficient way? – directly tallying thousands of reaction rates results in a large computational overhead
- How to handle automated division into depletion zones?
- How to obtain material volumes?

All these problems could be left for the user to figure out, but it was decided that the methodology should be automated as far as possible.

In the end, implementation of burnup capability in Serpent turned into a complicated data management and book-keeping problem, with very little to do with physics!

Another major challenge was verification – even though the “laws of physics” in the transport simulation were shared with MCNP,[11] there was no perfect reference code for burnup calculations because of additional uncertainties coming from:

- Fundamental nuclear data (decay, fission yields, isomeric branching)
- Formulation of transmutation chains
- Coupling algorithms (Euler, predictor-corrector, etc.)

Validation by comparison to experimental data is complicated by uncertainties related to measurements, e.g. large error margins and uncertainties related to operating history (power density, spectrum, burnup).

[11] In neutron transport calculation the differences between Serpent and MCNP should be within statistics, as long as the geometries and material definitions are consistent and the same ACE files and physics options are used. There are some differences in the photon physics models, which is why similar agreement cannot be expected for gamma transport simulations.

Development of Serpent 1

Serpent 1 was released to the OECD/NEA Data Bank in May 2009 and RSICC in March 2010. By September 2010 Serpent was used in 35 organizations in 20 countries around the world.

Almost immediately after the public release users started to complain about issues related to excessive memory usage:

- The unified energy grid may grow to 1,000,000 points in burnup calculation
- Grid thinning reduces memory consumption, but leads to loss of data
- Memory is also used for storing pre-calculated material-wise cross sections, which limits the total number of depletion zones
- Parallelization by MPI means that memory footprint is multiplied by the number of tasks running within the same node

Result: tricks implemented to speed up 2D assembly burnup calculations were not applicable to geometries containing hundreds or thousands of depletion zones.

The MPI parallelization issue was even more severe, because it meant that it was impossible to take full advantage of available CPU capacity.

Development of Serpent 2

The solution was to re-write the source code once again, taking into account the challenges related to:

- Small (2D assembly) and large (full-core) burnup calculation problems by introducing various optimization modes[12]
- Hybrid parallelization using MPI between nodes and OpenMP between CPU cores

The practical implementation created a whole new set of book-keeping and data management problems.

It was also decided early on to start looking at completely new types of applications:

- Multi-physics coupling to thermal hydraulics, CFD and fuel performance codes
- Radiation transport and shielding applications involving photon simulations, complicated geometries and variance reduction

Development of Serpent 2 started in late 2010 and is still on-going. The “beta version” 2.1.0 was released in 2012 (current update 2.1.31).

The release of a new base version 2.2.0 is scheduled for early 2020.

[12] J. Leppänen and A. Isotalo. “*Burnup Calculation Methodology in the Serpent 2 Monte Carlo Code.*”. In proc. PHYSOR 2012. Knoxville, TN, Apr. 15-20, 2012.

Part II: Current status and future work

Serpent developer team

Serpent development is carried out by 4+ researchers at VTT:[13]

- Jaakko Leppänen
- Ville Valtavirta
- Toni Kaltiaisenaho
- Riku Tuominen
- ...

With significant contributions from the user community.

[13] The list should also include Maria Pusa and Tuomas Viitanen, who are no longer working at VTT.

Serpent user community

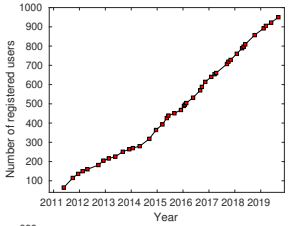
User community in numbers:

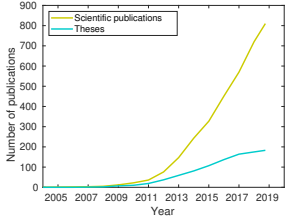
- 962 registered users on mailing list
- 224 organizations (48% universities)
- 44 countries
- Typical user: MSc or PhD student
- 180 theses on Serpent-related topics since 2007
- 810 peer-reviewed journal and conference papers since 2005

Serpent website: http://montecarlo.vtt.fi

Serpent discussion forum: http://ttuki.vtt.fi/serpent

Serpent Wiki: serpent.vt.fi/mediawiki/[14]





[14] Serpent wiki has useful resources for users, including an on-line tutorial – http://serpent.vtt.fi/mediawiki/index.php/Tutorial

Serpent user community

Group photos from previous International Serpent User Group Meetings: Dresden, 2011; Madrid, 2012; Berkeley, 2013; Cambridge, 2014; Knoxville, 2015; Milan, 2016; Gainesville, 2017; Espoo, 2018.

The 2019 UGM is hosted by Georgia Tech in Atlanta, USA next week.

On-going and future work

Serpent development is currently divided between three major topics:

i) Advanced methodology for spatial homogenization

ii) Coupled multi-physics applications

iii) New applications, including radiation shielding and fusion neutronics

The first topic continues the original idea of using the Monte Carlo method for group constant generation:

- The methodology has been put to practice using various reactor simulator codes: DYN3D, PARCS, ARES, TRAB3D, HEXTRAN, HEXBU, APROS, ...
- The majority of recent work has been focused on improving the results of nodal calculations using Serpent-generated cross sections (diffusion coefficients, ADF's, leakage models)
- Some studies on fuel behavior coupling in assembly burnup calculations

Development of multi-physics capabilities

Modeling an operating nuclear reactor requires solving a coupled non-linear problem between:

- Neutronics
- Thermal hydraulics
- Fuel behaviour
- Fuel depletion

The most common approach is to apply operator splitting, and iterate between linearized single-physics solutions.

Methods used for neutronics can be roughly divided in two categories:

- Reduced-order methods (spatial homogenization + nodal diffusion for the full-scale solution)
- High-fidelity methods (direct Monte Carlo full-scale solution)

Even though reduced-order methods are likely to remain the “work horse” in reactor analysis, coupled high-fidelity methods are becoming a feasible option for many problems along with the development of computer capacity and high-performance parallel computing.

Development of multi-physics capabilities

Coupled high-fidelity multi-physics simulations implies coupling the Monte Carlo neutronics solution to CFD, thermal hydraulics and fuel performance codes.

In Serpent this is accomplished using a multi-physics interface, that separates state-point information from the actual geometry model:

- For the code user this means that no modifications are needed in the geometry input, as density and temperature distributions are handled using separate data structures
- For the Monte Carlo tracking routine this means that density and temperature distributions can be handled efficiently using a rejection sampling based algorithm (variation of delta-tracking)

For the code developer this means yet another complicated data management and book-keeping problem!

Development of multi-physics capabilities

A few early milestones in the development of multi-physics capabilities:

- Implementation of a universal multi-physics interface (2012)[15]
- Capability to handle non-uniform density distributions and the TMS on-the-fly temperature treatment routine (2012–2015)[16,17]
- Development of an unstructured mesh based multi-physics interface and coupling to CFD (OpenFOAM) calculations (2013)[18]
- Fuel performance code coupling and transient simulations (2012–)[19]

The work continues within the McSAFE project in EU Horizon 2020.

---

15 J. Leppänen, T. Viitanen, and V. Valtavirta. "*Multi-Physics Coupling Scheme in the Serpent 2 Monte Carlo Code*.". Trans. Am. Nucl. Soc. **107** (2012), 1165–1168.

16 J. Leppänen. "*Modeling of Nonuniform Density Distributions in the Serpent 2 Monte Carlo Code*.". Nucl. Sci. Eng. **174** (2013), 318–325.

17 T. Viitanen. "*Development of a stochastic temperature treatment technique for Monte Carlo neutron tracking*.". DSc Thesis, Aalto University, 2015.

18 J. Leppänen et al. "*Unstructured Mesh Based Multi-physics Interface for CFD Code Coupling in the Serpent 2 Monte Carlo Code*.". In proc. PHYSOR 2014. Kyoto, Japan, Sept. 28 - Oct. 3, 2014.

19 V. Valtavirta. "*Development and applications of multi-physics capabilities in a continuous energy Monte Carlo neutron transport code*.". DSc Thesis, Aalto University, 2017.

Kraken computational framework

In 2017 we started working on a new Serpent-based computational framework “Kraken” for reactor core physics calculations.[20]

Lessons learned from several years of work on group constant generation using Serpent with both good and bad experience:

- The capability to generate cross sections using the Monte Carlo method is not enough to guarantee good results
- The methods used for for spatial homogenization must be consistent with the methods used for nodal calculations (this is the strength of CASMO-SIMULATE)

Conclusion: VTT needs a completely new reduced-order calculation system, along with the associated expertise, to take advantage of the full potential of Serpent in spatial homogenization.

The same computational framework servers as the basis for high-fidelity calculations, and forms a platform for the development of new methodologies.

[20] V. Valtavirta et al. “*Kraken—An Upcoming Finnish Reactor Analysis Framework*.”. In proc. M&C 2019. Portland, OR, Aug. 25–29, 2019.

Kraken computational framework

The Kraken framework consists of independent modular solvers (VTT and third-party), coupled together via a common core-level multi-physics interface.

Capability to run two types of code sequences:

I) **Reduced-order sequence** – based on a reduced-order (e.g. nodal diffusion) neutronics solution and spatial homogenization, intended for routine design and safety analyses

II) **High-fidelity sequence** – based on a heterogeneous (Monte Carlo) transport solution, intended for best-estimate analyses and verification of reduced-order methods

The reduced-order sequence is based on new solvers, while the development of the high-fidelity sequence continues previous work on multi-physics coupling.

The entire core physics framework will be coupled to system-scale analyses via a second power plant level interface.

Kraken computational framework

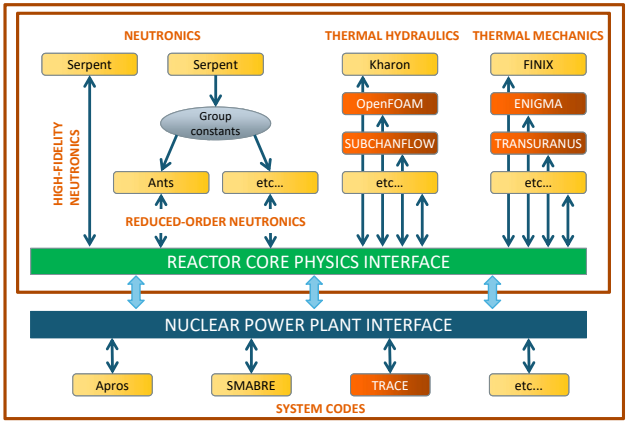


Figure 1: Overview of the Kraken computational framework for core physics calculations. All modular solvers operate on a common platform that couples the physical solutions together via a core-level multi-physics interface, and to external boundary conditions via the NPP interface. Most of the modular solvers will be interchangeable, so that the code sequence can be tailored to the specific needs of each application.

Kraken computational framework

Recent milestones in Serpent / Kraken development:

- Development of a new nodal solver “Ants” was started in 2017, and the first results were presented at PHYSOR2018.[21]
- Acceleration of fission source convergence in Serpent by a response matrix method-based solution for initial source guess[22]
- Functional expansion tallies for multi-physics coupling[23]
- Advanced heat deposition models[24]
- Transients with delayed neutrons[25]

---

[21] V. Sahlberg and A. Rintala. “*Development and First Results of a New Rectangular Nodal Diffusion Solver of Ants.*”. In proc. PHYSOR 2018. Cancun, Mexico, Apr. 22–26, 2018.

[22] J. Leppänen. “*Acceleration of Fission Source Convergence in the Serpent 2 Monte Carlo Code Using a Response Matrix Based Solution for the Initial Source Distribution.*”. In proc. PHYSOR 2018. Cancun, Mexico, Apr. 22–26.

[23] B. Wendt et al. “*Advancement of functional expansion capabilities: Implementation and optimization in Serpent 2.*”. Nucl. Eng. Design **334** (2018), 138–158.

[24] R. Tuominen, V. Valtavirta, and J. Leppänen. “*New energy deposition treatment in the Serpent 2 Monte Carlo transport code.*”. Ann. Nucl. Energy **55** (2013), 312–321.

[25] V. Valtavirta, M. Hessan, and J. Leppänen. “*Delayed Neutron Emission Model for Time Dependent Simulations with the Serpent 2 Monte Carlo Code - First Results.*”. In proc. PHYSOR 2016. Sun Valley, ID, May 1–6, 2016.

Kraken computational framework

On-going work:

- Methods for sensitivity and uncertainty analyses based on the collision history-based approach[26]
- Domain decomposition for large-scale burnup calculations
- Development of new leakage models for burnup calculations
- Parametrization of group constants
- Stability of 3D burnup calculations
- ...

[26] M. Aufiero et al. "*A collision history-based approach to sensitivity/perturbation calculations in the continuous energy Monte Carlo code SERPENT.*". Ann. Nucl. Energy **85** (2015), 245–258.

Development of a photon transport mode

Originally the main motivation for expanding to photon transport was to simulate gamma heating in multi-physics calculations:

- Accurate deposition of fission energy requires accounting for direct and indirect components of prompt and delayed heating
- Prompt indirect neutron and photon heating may become important especially in fast transients

Development of photon physics routines:

- First introduced in 2015[27]
- Coupled neutron-photon transport mode in 2017[28]
- Photonuclear reactions in 2019[29]

See extra slides at the end of this presentation for some more details on photon physics.

---

[27] T. Kaltiaisenaho. "*Implementing a photon physics model in Serpent 2.*". MSc Thesis, Aalto University. 2016.

[28] J. Leppänen et al. "*Implementation of a Coupled Neutron / Photon Transport Mode in the Serpent 2 Monte Carlo Code.*". In proc. M&C 2017. Jeju, Korea, Apr. 16-20, 2017.

[29] T. Kaltiaisenaho. "*Photonuclear Reactions in Serpent 2 Monte Carlo Code.*". In proc. M&C 2019. Portland, OR, Aug. 25–29, 2019.

Development of a photon transport mode

Accurate heat deposition models with gamma heating remain as one of the major near-term development goals.

However, the photon transport mode and other related capabilities have also enabled expanding the applications of Serpent from reactor physics to new fields, in particular:

- Radiation shielding and dose rate calculations
- Fusion applications

Specific challenges related to radiation transport calculations:

- Geometries are often complicated and irregular
- Shielding calculations require extensive use of variance reduction

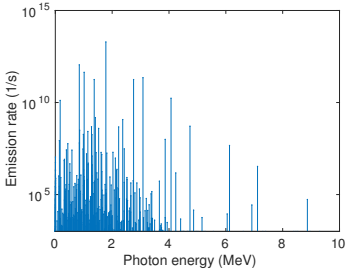
In addition, calculation of dose rates and detector responses requires special techniques that are not commonly applied in reactor analysis.

Radioactive decay source mode

Prompt gammas produced in neutron reactions can be accounted for using the coupled neutron-photon transport mode. The radioactive decay source mode was implemented to account for delayed radiation emitted by activated materials.

The source term is determined by:

- Isotopic material compositions, either user-defined or obtained from a previous burnup / activation calculation
- Decay constants of radioactive nuclides, read from ENDF-format decay data file
- Emission spectra, read from ENDF-format decay data file



In addition to photon line spectra, the source term can include neutron emitting reactions (e.g. spontaneous fission) and bremsstrahlung produced by beta-decay.[30]

[30] T. Kaltiaisenaho. "*Implementing a beta bremsstrahlung source in Serpent.*". VTT-R-00953-18, VTT Technical Research Centre of Finland, Ltd. (2018).

Radioactive decay source mode

The radioactive decay source mode is most conveniently used with burnup or activation calculation (radioactive material compositions read from a binary restart file).

The same input can be used in both calculations without major modifications:

- Isotopic material compositions automatically converted into elemental
- Source normalization done automatically based on total emission rate

Example applications:

- Radiation shielding calculations for CASTOR canister with spent nuclear fuel (carried out within the SAFIR2018/KATVE project)[31]
- Shut-down dose-rate calculations for the ITER fusion reactor[32,33]

[31] S. häkkinen. "*Shielding analysis of a spent fuel storage cask using variance reduction with Serpent.*". VTT-R-00796-18, VTT Technical Research Centre of Finland, Ltd. (2018).

[32] P. Siren and J. Leppänen. "*Expanding the Use of Serpent 2 to Fusion Applications: Development of a Plasma Neutron Source.*". In proc. PHYSOR 2016. Sun Valley, ID, May 1–6, 2016.

[33] J. Leppänen and T. Kaltiaisenaho. "*Expanding the Use of Serpent 2 to Fusion Applications: Shut-down Dose Rate Calculations.*". In proc. PHYSOR 2016. Sun Valley, ID, May 1–6, 2016.

Advanced geometry types

The basic CSG geometry type used in Serpent is usually sufficient for reactor applications:

- Most reactor geometries can be described using planes and cylinders
- Regular structures are easy to model using standard universes and lattices
- Special geometry type for HTGR fuels (particles and pebbles)[34]

Advanced types available for irregular and unstructured geometries:

- Mesh-based geometry type developed together with the OpenFOAM multi-physics interface was introduced in 2013[35]
- CAD-based geometry type based on STL file format implemented in 2014[36]

Implementing a voxel-based geometry model is also an option if medical applications become important at some point.

---

34 V. Rintala et al. "*Modelling of Realistic Pebble Bed Reactor Geometries Using the Serpent Monte Carlo Code.*". Ann. Nucl. Energy **77** (2015), 223–230.

35 J. Leppänen and M. Aufiero. "*Development of an Unstructured Mesh Based Geometry Model in the Serpent 2 Monte Carlo Code.*". In proc. PHYSOR 2014. Kyoto, Japan, Sept. 28 - Oct. 3, 2014.

36 J. Leppänen. "*CAD-based Geometry Type in Serpent 2 – Application in Fusion Neutronics.*". In proc. M&C + SNA + MC 2015. Nashville, TN, Apr. 19–23, 2015.

Advanced geometry types

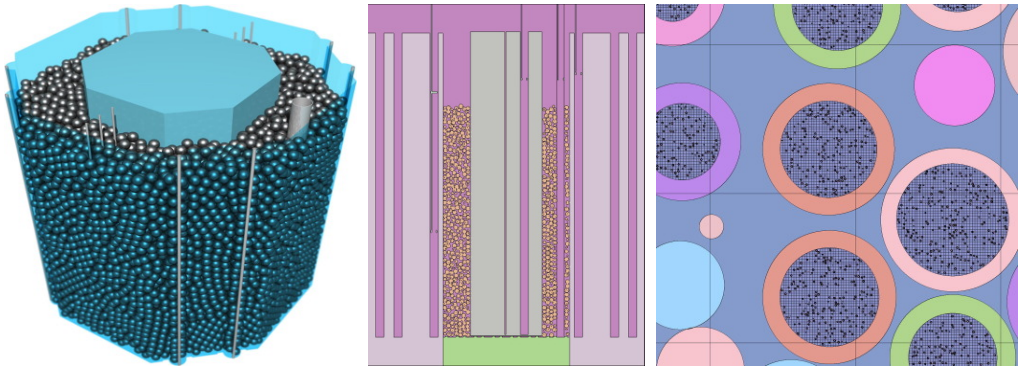


Figure 2: Explicit stochastic particle fuel model in Serpent 2 (ASTRA critical experiment). The coordinates of fuel particles inside the pebbles and the positions of pebbles inside the core are provided by the user, and the geometry is modeled without approximations. In this study, the stochastic pebble configuration was obtained from DEM simulations.

Advanced geometry types

The CAD-based geometry type in Serpent is based on the STL file format:

- Solid bodies constructed from triangulated surfaces
- Widely used for 3D printing – supported by most CAD tools
- Geometries consisting of one or several STL solids are handled as universes
- Background universe for undefined regions (no need to define empty space)
- Tracking routine can handle gaps between solids without problems
- Adaptive search mesh to speed-up geometry routine

The STL geometry type can be used in combination with other types (CSG, mesh-based, stochastic pebble/particle fuel model).

Advanced geometry types: CAD-based geometry type

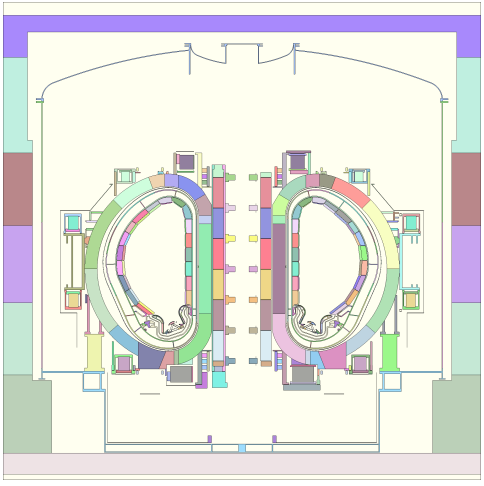


Figure 3: Serpent geometry plot of an STL geometry type Different CAD solids are plotted with different colors. (C-Lite model of the ITER fusion reactor comprised of 11 components, 1,548 solids, 1,842,576 points, 614,192 triangular facets).

Advanced geometry types: CAD-based geometry type

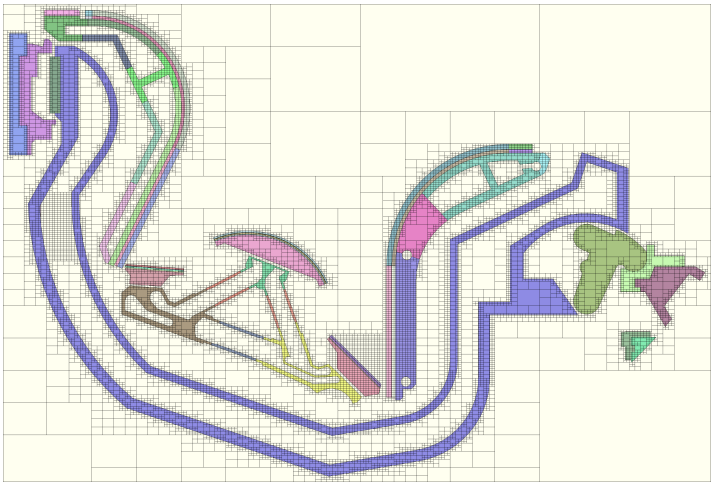


Figure 4: Serpent geometry plot of an STL geometry type (divertor cassette, part of the ITER C-Lite model). Different CAD solids are plotted with different colors. The search mesh is automatically adapted around the triangulated surface.

Advanced geometry types: the next challenge

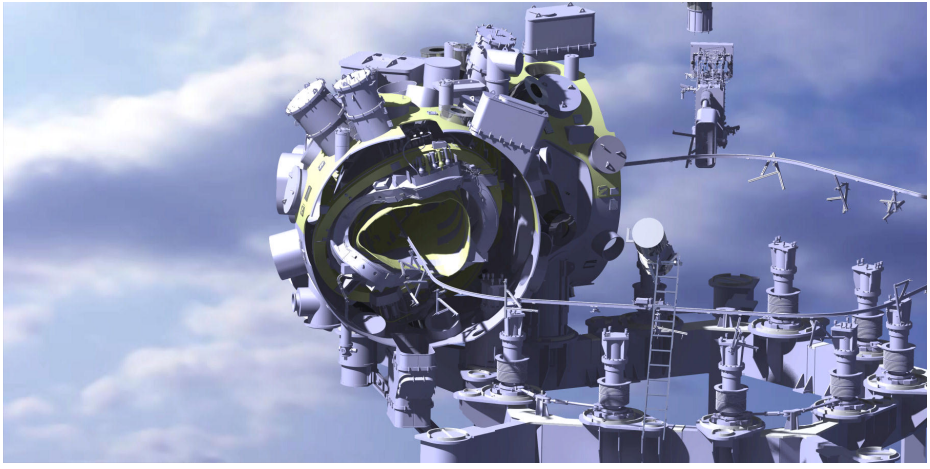


Figure 5: CAD-based model of the Wendelstein 7-X stellarator.

Variance reduction

Radiation shielding calculations using the Monte Carlo method require extensive use of variance reduction.

A weight-window based variance reduction scheme was first introduced in 2016:[37]

- Support for MCNP WWINP format weight-window mesh generated using some deterministic tool
- Built-in light-weight response matrix based importance solver[38]

The built-in solver obtains coupling coefficients for the response matrix method from a forward Monte Carlo simulation, and provides the importances (adjoint solution) for calculating the weight-window boundaries:

- Applicable to both photon and neutron transport problems
- Support for single and multiple detector responses and global variance reduction (GVR)
- Rectangular, cylindrical, hexagonal and self-adaptive octree mesh[39]

---

[37] J. Leppänen, T. Viitanen, and O. Hyvönen. "*Development of a Variance Reduction Scheme in the Serpent 2 Monte Carlo Code.*". In proc. M&C 2017. Jeju, Korea, Apr. 16-20, 2017.

[38] J. Leppänen. "*Response Matrix Method–Based Importance Solver and Variance Reduction Scheme in the Serpent 2 Monte Carlo Code.*". Nucl. Technol. (2019, in press). DOI: 10.1080/00295450.2019.1603710.

[39] J. Leppänen and M. Jokipii. "*Global Variance Reduction Scheme with Self-Adaptive Weight-Window Mesh in the Serpent 2 Monte Carlo Code.*". In proc. M&C 2019. Portland, Oregon, USA, Aug. 25–29, 2019.

Potential radiation transport applications for Serpent 2

Serpent has a lot of potential for radiation transport applications, for example:

- Back-end of the nuclear fuel cycle (storage, transportation and final disposal of spent fuel, decommissioning of reactors)
- Industrial applications involving radioactive sources
- Shielding design (hot cells, sources, reactors, accelerators, space applications)
- Medical applications of radiation (radiotherapy, medical imaging)
- Dose rate calculations involving complicated source terms (e.g. radioactive fallout)

Serpent is still used almost exclusively for reactor physics, and radiation transport is a relatively new application for us as well, so all user feedback is greatly appreciated!

Summary: Lessons learned from 15 years of Serpent development

VTT

Personal observations:

- Developing your own calculation code is extremely interesting, and can be very rewarding as well!
- Sometimes it can be better not to know everything, because then you can find new solutions (re-inventing the wheel is not a bad way to learn)
- It is not possible to plan everything beforehand – the work often proceeds with a live, learn and fix later -type of approach
- Most of the actual work is problem-solving, not physics or mathematics
- The most complex problems are often related to the handling of large data sets and complying with the limitations of computers (that also evolve in time)
- Code development does not scale linearly – as the project grows and more people get involved the pace of development slows down

If you have the chance build your professional career on code development, I can warmly recommend it!

Thank you for your attention!

Questions? – Jaakko.Leppanen@vtt.fi

http://montecarlo.vtt.fi

Extra slides

The most obvious part of any Monte Carlo transport code is the simulated random walk, which follows a very simple algorithm:

1) Sample distance to next collision from an exponential distribution
2) Move neutron to collision site
3) Sample interaction

If the sampled interaction is scattering, the procedure restarts from beginning with new energy and direction of motion.

What makes things a bit more complicated is that the sampled path length:

$$l = -\log(\xi)/\Sigma \tag{3}$$

is not statistically valid if the neutron crosses the boundary between two materials.

The traditional solution applied in the surface-tracking algorithm is to stop the neutron at the boundary surface and sample a new path using the total cross section of the next material.

Methodology in PSG: Woodcock delta-tracking method

VTT

An alternative to surface-tracking is the Woodcock delta-tracking method.[40] The procedure relies on the concept of a *virtual collision*, which is a fictive interaction that preserves both energy and direction of motion.

Since virtual collisions do not change the random walk in any way, the material total cross section Σ can be adjusted with an arbitrary virtual collision cross section Σ_0:

$$\Sigma'(\boldsymbol{r}, E) = \Sigma(\boldsymbol{r}, E) + \Sigma_0(\boldsymbol{r}, E) \tag{4}$$

without changing the outcome of the simulation. It is then possible to adjust the cross sections of all material regions in the system such that:

$$\Sigma'_1(E) = \Sigma'_2(E) = \Sigma'_3(E) \cdots = \Sigma_{\mathrm{m}}(E) \tag{5}$$

where Σ_{m} is called the *majorant* cross section.

In practice, it is not necessary to define the virtual collision cross sections at all if the majorant is simply taken as the maximum of all material totals at each energy point:

$$\Sigma_{\mathrm{m}}(E) = \max\left[\Sigma_i(E)\right] \tag{6}$$

40 E. R. Woodcock et al. "*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.

Methodology in PSG: Woodcock delta-tracking method

The point of having a macroscopic cross section that is uniform throughout the geometry is that when used for sampling path lengths:

$$l = -\log(\xi)/\Sigma_{\mathrm{m}} \tag{7}$$

the values are statistically valid regardless of the number of material boundaries crossed.

At the end point of the sampled path the tracking routine performs rejection sampling. The probability to accept the collision is given by ratio of the physical total cross section to the majorant:

$$P = \frac{\Sigma(\boldsymbol{r}, E)}{\Sigma_{\mathrm{m}}(E)} \tag{8}$$

If the collision is rejected, a new path length is sampled from (7) and the neutron is moved to the next collision site candidate.

Since the majorant cross section is always larger than or equal to the total cross section, the path lengths sampled in delta-tracking are, on the average, shorter than those sampled with surface-tracking.

The average physical distance between two collisions is preserved, as the paths are extended over multiple virtual collisions.

Methodology in PSG: Woodcock delta-tracking method

Delta-tracking is essentially a rejection sampling technique, used for sampling values from a piece-wise continuous probability distribution where the points of discontinuity are not known.

The advantages of delta-tracking include:

- Easy to implement (no need to calculate surface distances)
- Can be faster than surface-tracking in geometries where the neutron mfp is long compared to dimensions (no need to stop the track at each boundary crossing)
- Error tolerant to small gaps between geometry cells
- Can be used for modeling inhomogeneous material compositions (majorant is independent of position, total cross section needed only at discrete locations)

Methodology in PSG: Woodcock delta-tracking method

But there are also drawbacks:

- Efficiency becomes poor in the presence of localized heavy absorbers (control rods, burnable absorber pins, etc.)
- Rules out the use of track-length estimate (TLE) of neutron flux – integral reaction rate estimates need to be calculated using the potentially less efficient collision estimator (CFE)
- Cannot use information on regional connectivity in cell search – tracking in universes comprised of a very large number of cells becomes slow

The performance of CFE is comparable to TLE in most reactor physics applications, when reaction rates are calculated in regions of high collision density.[41] In practice things get more complicated since also virtual collisions can be scored.[42]

Problems with localized heavy absorbers can be overcome by switching to surface-tracking when the efficiency of the rejection sampling loop becomes poor.

---

41 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.

42 J. Leppänen. "*On the use of delta-tracking and the collision flux estimator in the Serpent 2 Monte Carlo particle transport code*.". Ann. Nucl. Energy **105** (2017), 161–167.

Methodology in PSG: Woodcock delta-tracking method

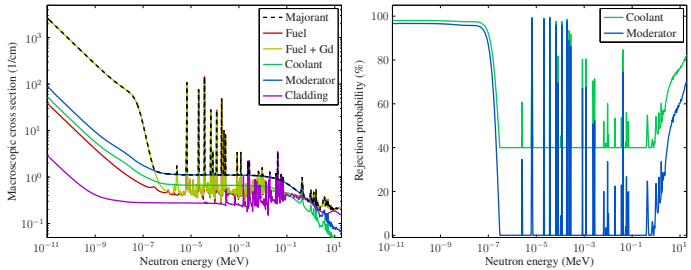


Figure 6: Left: Majorant and macroscopic cross sections in a system with localized heavy absorber (Gd-fuel pins in BWR assembly). The majorant is dominated by the high capture cross sections of ^{155}Gd and ^{157}Gd, even though the burnable absorber pins occupy a relatively small volume of the geometry. Right: Rejection probability in coolant and moderator where neutrons spend most of their lifetime. The efficiency of the rejection sampling scheme becomes poor especially at low energy.

Methodology in PSG: Interaction physics

One of the advantages of continuous-energy Monte Carlo method is its capability to use the best available knowledge on neutron interactions without major approximations.

In practice, the “laws of physics” are hidden behind cross sections and probability distributions describing the emission of secondary particles.

The ACE data format (originally developed for MCNP) was selected for Serpent because of several reasons:

- MCNP and NJOY were already used at VTT
- The data format is well documented
- Simplifies verification (no additional uncertainties when compared to MCNP)

Instead of using this data as-is, all cross sections were reconstructed using a common energy grid:

- Capability to pre-calculate macroscopic material-wise cross sections to avoid summing over partials during the transport simulation
- Energy grid search has to be performed only once for each neutron energy
- Majorant cross section is easy to calculate from the pre-calculated material totals

The first two tricks lead to a significant speed-up in calculation, especially when the number of nuclides is large.[43]

The same tricks also lead to major issues with excessive memory consumption a few years later.

The first version of the code used a log-uniform energy grid, but this was later replaced by merging the grid points of individual nuclides into a single unified grid.

[43] 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.

Photon physics model in Serpent 2

Photon physics model covers elements from hydrogen to Einsteinium (Z = 99) and energy range from 1 keV to 100 MeV.

Photon cross sections are read from ACE format data libraries, also used by MCNP (e.g. mcplib12).

Additional sources are used for interaction data:

- Most of the interaction data is from ENDF-B-VII.1 (form factors, incoherent scattering functions, photoelectric cross sections and atomic relaxation data)
- Other sources for data not found in ENDF-B-VII.1 (Compton profiles and bremsstrahlung data)

The physics model for neutrons is practically identical to that used in MCNP, but for photons the results cannot be expected to match perfectly.

Photon physics model in Serpent 2

Rayleigh scattering: (elastic scattering from the electron cloud of an atom)

- Direction is sampled using the form factor approximation

Compton scattering: (inelastic scattering from an atomic electron)

- Direction is sampled using the incoherent scattering function approximation
- Doppler broadening of the photon energy is taken into account (caused by the momentum distribution of the electron), important below 1 MeV

Photoelectric effect:

- Electron shell is selected with a probability given by its cross section, all sub-shells are included

Pair production:

- The energies and directions of the electron and positron are sampled using appropriate theoretical distributions
- Positron annihilation at rest generates two 0.511 MeV photons

Photon physics model in Serpent 2

Atomic relaxation:

- Compton scattering and photoelectric effect cause vacancies in electron shells
- Relaxation cascade through radiative (fluorescence) and non-radiative (Auger, Coster–Kronig) transitions
- Transitions are sampled according to the probabilities given by ENDF/B-VII.1, all possible transitions are included

Thick-target bremsstrahlung approximation:

- Electrons are generated through Compton scattering, photoelectric effect, pair production and non-radiative transitions
- Bremsstrahlung photon production is important especially for high-Z atoms at energies above $\sim$1 MeV
- The number of bremsstrahlung photons and their energies are sampled from the distributions given by the continuous slowing down approximation (CSDA)
- Angular distribution is omitted; the direction of the bremsstrahlung photon is equal to the direction of the electron

Photon detectors

Serpent provides conventional Monte Carlo cell flux, surface current etc. detector (tally) capabilities for neutrons and photons.

Built-in response functions for photons:

- Mass-energy attenuation coefficients from NIST data
- Flux-to-effective dose conversion factors from ANSI/ANS 6.1.1-1977, ICRP-21 (1971), ICRP-74 (1996), and ICRP-116 (2010)

In addition there are two photon-specific detector types:

- Analog energy deposition detector for photon heating / absorbed dose
- Photon pulse height detector for modeling detector responses

The standard tally functionality in Serpent also includes user-defined response functions.

Serpent cannot explicitly transport secondary electrons produced in photon interactions, but the simulation can be coupled to Geant4 via surface sources (for more information, contact: Toni.Kaltiaisenaho@vtt.fi).

Variance reduction: practical demonstration

Task: Calculate photon dose rates on the outer surface of a transport cask containing spent nuclear fuel assemblies.

Practical Challenges and applied solutions:

Source term complexity: Photon source consists of gamma radiation emitted by hundreds of radionuclides in the irradiated fuel. – *Apply radioactive decay source mode to form source term automatically.*

Geometry complexity: The storage cask is a complicated structure. – *Use a CAD model and import the geometry into Serpent in STL format.*

Heavy shielding: Poor statistics in analog simulation. – *Apply built-in importance solver and GVR.*

The simulation had to be divided into three parts:

1) Assembly burnup calculation
2) Production of a weight-window mesh
3) Final transport calculation

Even so, the amount of manual work is reduced to minimum.

Variance reduction: practical demonstration

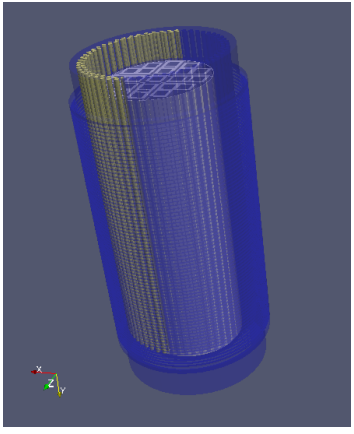


Figure 7: CAD-based geometry model of the CASTOR storage cask. The cask is 490 cm high and 240 cm in diameter.

Variance reduction: practical demonstration

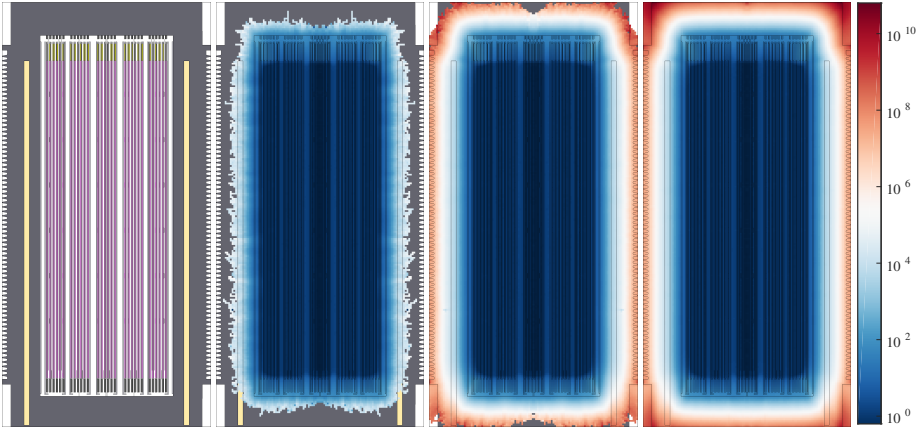


Figure 8: Axial geometry plot of the storage cask model and the importance mesh produced by the automated GVR routine. The transport cask consists of three STL solids: stainless steel basket (light gray), polyethylene neutron moderator rods (yellow) and cast iron shell (dark gray). The fuel assemblies inside the cask were modeled as a traditional CSG geometry. The geometry was fully populated after three GVR iterations with 50 million photon histories.

Variance reduction: practical demonstration

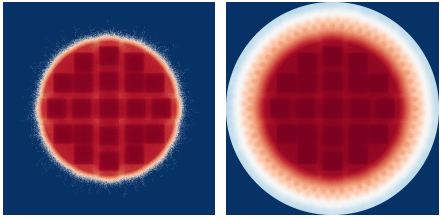


Figure 9: Radial dose rate distribution in the cask. Left: without VR. Right: with VR.

Results:

- Average dose rate on surface 9.29 $\pm$0.03 μSv/h
- Weight-window generation by 3 GVR iterations using 50 million photon histories, completed in 29 minutes
- Final calculation using 1 billion histories, completed in 5.7 hours
- Analog reference calculation using 100 billion histories, completed in 142 hours
- Factor of 5000 gain in figure-of-merit using variance reduction