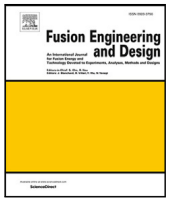
Fusion Engineering and Design 200 (2024) 114191

Contents lists available at ScienceDirect

Fusion Engineering and Design

journal homepage: www.elsevier.com/locate/fusengdes



Computation of sensitivity coefficients in fixed source simulations with SERPENT2

P. Griveaux [a], M. Hursin [a,*], I. Kodeli [b], D. Leichtle [c], A. Pautz [a]

[a] *École Polytechnique Fédérale de Lausanne, Switzerland*
[b] *Culham Center for Fusion Energy, United Kingdom*
[c] *Karlsruhe Institute of Technology, Germany*

ARTICLE INFO

Keywords:
Sensitivity coefficients
Nuclear data library
Neutronics
Monte Carlo
SERPENT2

ABSTRACT

Within the scope of the implementation of a nuclear data pipeline aiming at producing the best possible evaluated nuclear data files, a major point is the production of relevant sensitivity coefficients when including integral benchmark information. Thanks to recent code modifications in the Monte Carlo code SERPENT2, it is now possible to produce these coefficients in fixed source simulations. The manuscript describes the verification of this implementation against the deterministic transport code SUSD3D. The study is completed by an analysis of the computational cost (running time and memory allocation) associated with such calculations with SERPENT2. The relative difference between the sensitivity coefficients produced by SERPENT2 and SUSD3D is of the order of the percent at most, except for the low energy range where the lack of neutrons prevents from reducing the Monte Carlo uncertainties. The computational cost of such calculation is similar to the one of criticality calculation mode, although the OPENMP scalability should be further improved.

1. Introduction

The implementation of a nuclear data pipeline (an automated sequence of scripts) [1] to produce the best possible evaluated nuclear data files (ENDF) has been under development recently to improve nuclear data libraries. It can be done by combining energy dependent and integral measurements using a Bayesian calibration process [2]. The computation of sensitivity coefficients to the integral benchmark information (reaction rates for example) is a key issue in producing such pipeline, and is the core of this work. More specifically, the nuclear data of interest in this work are the ones related to structure materials used in fusion devices such as Iron, Tungsten, Nickel and Chromium.

SERPENT2 is a Monte Carlo code developed by the Valtion Teknillinen Tutkimuskeskus (VTT), Finland. It has already been extensively used to compute sensitivity coefficients to nuclear data in criticality calculation mode (see for instance [3]). However, prior to the present work, it was not possible to determine these coefficients in fixed source mode. Recently, a capability to compute of sensitivity coefficients in fixed source type simulations has been implemented in the code. This work aims at providing an assessment of SERPENT2's newest sensitivity coefficients capabilities in fixed source problems, and of its computational cost. The rest of this paper is organised as follows: Section 2 describes the theoretical and practical aspects behind the calculation of sensitivity coefficients in fixed source mode in SERPENT2. The assessment of the new developments is done by comparing with deterministic code SUSD3D on a toy problem case defined in Section 3. The results of a computational cost study in terms of running time and memory consumption are developed in Section 4 and conclusions are drawn in Section 5.

2. Sensitivity coefficients

Several Monte Carlo methods have been developed recently to determine the sensitivity of a given nuclear data cross section (usually in a group-wise formalism) to a response function R in criticality calculations [3–7]. Such capabilities can in principle be extended to fixed source calculations provided that some small modifications are made to the Iterated Fission Probability (IFP) algorithm implemented in the SERPENT2 code [3].

When doing Generalised Perturbation Theory (GPT) calculations in criticality source mode, responses are ratios of scores computed at the same time. Those scores are stored at the end of a given generation (bank) for all considered latent generations (ancestors). The impact of the perturbations on the total neutron population does not matter as it is an eigenvalue problem, the magnitude of the neutron population is an arbitrary parameter. In fixed source simulations however, the perturbations might impact the subcritical multiplication and change

* Corresponding author.
E-mail address: mathieu.hursin@epfl.ch (M. Hursin).

https://doi.org/10.1016/j.fusengdes.2024.114191
Received 16 August 2023; Received in revised form 12 January 2024; Accepted 19 January 2024
Available online 23 January 2024

the population. As a result some sort of normalisation of the neutron population is required.

Let us consider a 1 keV neutron source next to a fuel lattice containing soluble boron, with a U-235 fission rate detector at a given location in the model. The goal is to evaluate the sensitivity of the detector response to the total cross section of B-10 in the coolant. For such calculations in criticality source mode, each time a neutron contributes to the fission rate detector we would essentially add to the scoring buffer the net number of collisions in boron times the detector response value, e.g. U-235 fission cross section at the considered energy. The net number of collisions is calculated as a sum of all collisions in the neutron history (up to the number of latent generations). Later, when the results are collected the value in this buffer gets divided by the actual detector value.

In external source mode, the population size will change due to our B-10 perturbation affecting the subcritical multiplication factor. However, the same process can be used provided that the 'net number of collisions in boron' is determined from the whole collision history back to the initial neutron emitted by the source. In the specific case of Serpent2, neutrons emitted by fissions or (n, xn) processes are put in simulation queue (instead of bank as in criticality source simulations) and they get simulated before the modelling of the next source particle. This greatly simplifies collision history tracking.

Then, if the `set srcrate` option in textscSerpent2 is used, neutron source rate is prescribed regardless of the perturbations and gives directly access to the sought sensitivity coefficients.

This requires the user to specify from the input the total neutron source through the `srcrate` option. The use of other source options (`set powdens`, `set power`, `set flux`, `set genrate`, `set fissrate`, `set absrate`, `set lossrate`) is also possible but would require additional normalisation of the denominator in the reaction rate ratio response.

Finally this approach to determine first order sensitivity coefficient in the fixed source mode of Serpent2 could be extended to exact perturbation theory at low computational cost provided a small amount of additional modifications in the code [8].

These developments mentioned here are not specific to the Serpent2 code and could be ported to other Monte Carlo codes relying on IFP when performing GPT calculations.

With the assistance of Serpent2 developers, these modifications have been implemented in the version 2.1.32 of the code. Further modifications were also needed to improve the memory consumption of the calculations. These last modifications are included in version 2.2.1.

3. Assessment of Serpent2 's sensitivity computation accuracy

Among the two objectives of this work, the first was the assessment of Serpent2's computational capabilities to determine sensitivity coefficients in fixed source mode. It has been done by comparing with susd3d [9] results. This deterministic code is based on the GPT. In order to prevent spatial biases, a simple, spherically symmetric geometry shown in Fig. 1 is considered, which was already used in the past for the validation of M/C sensitivity code MCSEN [10,11]. Note that for criticality and β_{eff} sensitivity and uncertainty analysis an intercomparison of different deterministic and Monte Carlo codes, including Serpent2 was already done in the past [12].

The model is a simplified representation of the IPPE (Institute of Physics and Power Engineering) Iron sphere benchmark. A point source is surrounded by a 28 cm-thick spherical shell of pure Fe-56. A flux detector is positioned at 680 ± 0.5 cm of the source, thus in a 1 cm-thick shell. The response of the detector is a group-wise scalar flux in the region of interest. The corresponding energy grid is a coarse 7-group energy grid, with upper energy boundaries 0.1, 1, 5, 7, 10, 13.84, 14.19, in MeV. On the other hand, the sensitivity coefficients are provided on the 175 energy groups VITAMIN-J group structure. Fe-56 cross sections are perturbed, using JEFF-3.3 nuclear data library for all

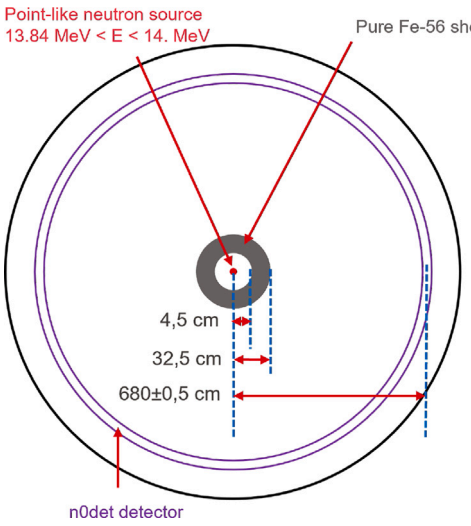


Fig. 1. Geometry of the model to compare Serpent2 against susd3d. The source is at the centre, surrounded by a shell of pure Fe-56 in grey, itself surrounded by a thin spherical shell for the flux detector delimited in purple, encompassed in a spherical domain defined by the inside of a sphere, represented in black.

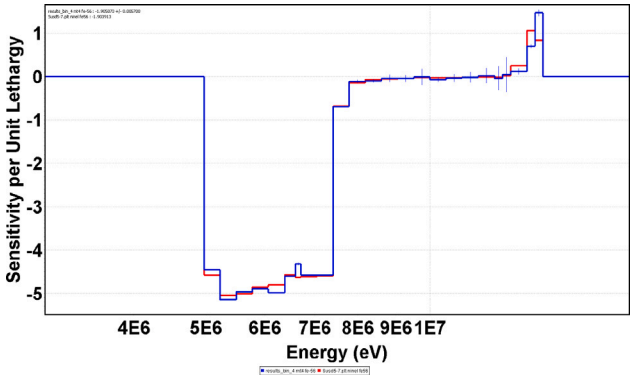


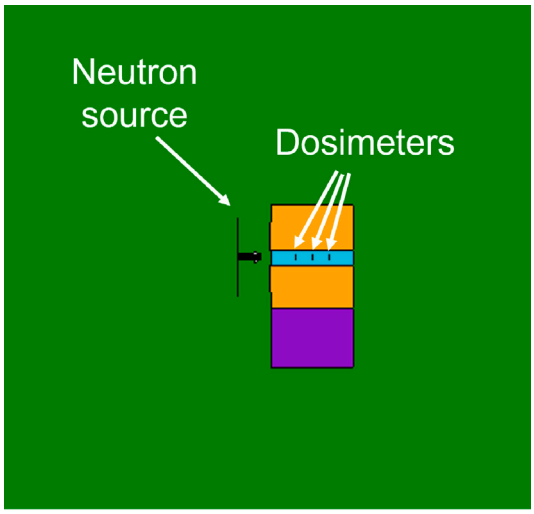
Fig. 2. Sensitivity of the 5 MeV $< E <$ 7 MeV neutron flux to inelastic scattering of Fe-56. Serpent2 is in blue, with uncertainty bars, and susd3d is in red.

Serpent2 computations. The reference sensitivity values are generated with susd3d using the same nuclear data library and PARTISN discrete ordinates transport code, all part of the XSUN package [13].

Fig. 2 displays the sensitivity of the neutron flux between 5–7 MeV to the inelastic scattering. This reaction has been chosen among all others as it has the biggest contribution to the total sensitivity. Sensibility coefficients (resp. by unit lethargy) are dimensionless as they are ratios of relative variation ($S = \frac{dR/R}{d\sigma/\sigma}$) (resp. $S/\Delta u = \frac{S}{ln(\Delta E/E)}$, which is still dimensionless). The bin-integrated sensitivity values (sum over all energy bins of the current sensitivity profile) are -1.905 ± 0.006 for Serpent2, and -1.904 for susd3d. In spite of the presence of some discrepancies, the sensitivity profile shows an overall good agreement between the two codes, and the integrated values are consistent within the Monte Carlo uncertainty of Serpent2.

In Appendix, other profiles showing the main contributions to the sensitivity of neutron flux for each energy bin are displayed. It should be noted that the energy range of the flux varies from profile to profile. The range is chosen to illustrate energy regions with significant flux sensitivity to the considered nuclide/reaction channel.

The relative difference between the two codes is at most of the order of the percent except for the lowest energy bin where significant uncertainties are still present due to the lack of particles in this region, see for instance Fig. 8. This Appendix includes as well Table 2 with the



Colour	Corresponding material
	Vacuum
	Air
	DENSIMET 176 (Tungsten alloy)
	DENSIMET 180 (Tungsten alloy)
	Aluminium
	Stainless steel
	Manganese
	Zirconium

Fig. 3. Benchmark configuration. The source is in front of a block of two Tungsten alloys on an aluminium support.

integrated sensitivities of SUSD3D and SERPENT2 along with the relative uncertainties for the latter code.

4. Computational cost

The second purpose of this work is to quantify the computational cost of sensitivity calculations with SERPENT2 in fixed source mode. All simulations have been done on a computer equipped with an Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz processor having 48 cores and a RAM capacity of 128 GB. A first computational study was done after the enabling of SERPENT2's sensitivity computational capabilities in fixed source problems. It showed the significant memory cost of the simulations. Additional improvements were made to the code by the development team and allowed to drastically reduce this cost. Specifically, the allocation of the scoring buffers used to determine the sensitivity coefficients were done only for the last tracked sensitivity generation, instead of doing it for all generations as required in criticality calculations. Nonetheless, the cost of sensitivity coefficients computations remains significant even with these last modifications, as summarised in Table 1.

The test case previously used to compare code results is simple. One would be interested in having an idea of the costs on a real case which was the actual purpose of the project. This is why the case that is considered to carry out the computational cost study is the Iron setup (meaning that the results are those of the Iron foil dosimeters) of FNG (Frascati Neutron Generator) tungsten benchmark, that can be found in SINBAD (Shielding Integral Benchmark Archive and Database) [14]. Fig. 3 shows the benchmark configuration.

The aforementioned cases (both with and without sensitivity calculations) will be taken as the 'reference cases' for the computational study. Their running parameters were 2×10^5 neutrons per batch, 200 batches and the simulations were launched in serial mode (on one OPENMP thread). All cross sections of tungsten isotopes 180, 182, 183, 184, 186 are considered during the sensitivity calculation, e.g. their sensitivity coefficients will be determined. The influence of several quantities is studied: the number of neutrons per batch (neutron multiplier), the number of threads on which the simulation is parallelised, the individual cost of several specific cross sections as well as the number of batches.

Table 1
Nominal values of running time and memory allocation for a simulation of the SINBAD FNG tungsten benchmark [15] (using the Iron foil dosimeters) with and without sensitivity computations.

	Running time (h)	RAM (GB)
Without sensitivity	0.46	2.2
With sensitivity	0.71	17
With/without	1.5	7.9

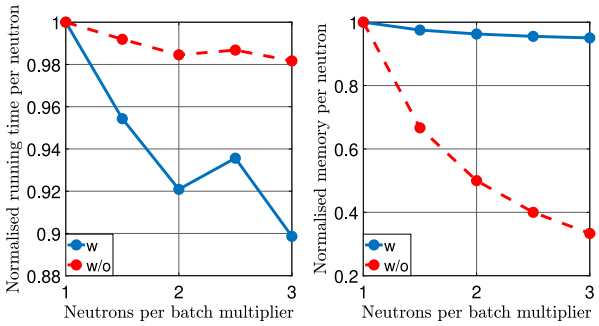


Fig. 4. Running time and memory allocations normalised to the number of neutrons per batch, respectively at the left and right, as functions of the neutron per batch multiplier. 2×10^5 neutrons per batch are used in each simulation. The dashed red line corresponds to the simulations without sensitivity computations, while the solid blue line corresponds to the simulations with sensitivity computations. 'w' (resp. 'w/o') stands for 'with sensitivity computations' (resp. 'without sensitivity computations').

Fig. 4 displays the running time per neutron history and memory allocation per neutron history as functions of the number of neutrons per batch (multiple of 2×10^5). Values are normalised to the reference case described above. The different numbers of neutrons per batch taken in the simulations are 2×10^5, 3×10^5, 4×10^5, 5×10^5 and 6×10^5. The normalised running time per neutron (runtime in the following) shows decreasing trends with and without sensitivities, meaning that increasing the number of neutrons per batch decreases the time cost per neutron, which is expected. This decline is more visible for the simulations when sensitivity coefficients are computed: multiplying the number of neutrons by 3 reduces the running time by 10% instead of only 2% without sensitivities. The presence of a peak for a neutron per batch multiplier of 2.5 is not fully understood at the time of writing. It is however reproducible and as such corresponds to a deterministic effect. Concerning the memory, it is clear that having more neutrons per batch lowers the memory cost per neutron. Nevertheless, the influence of the sensitivity coefficients computation is much stronger. While multiplying by 3 the number of neutrons per batch divides the memory cost per neutron by approximately 3 without sensitivities, it is barely reduced (less than 10%) with them.

Fig. 5 shows the scaling of the inverse running time and memory allocation with the number of threads on which the simulation is run ($y = x$ in red, all other lines corresponding to SERPENT2). The simulation has been successively parallelised on 1, 5, 10, 15, 20, 25, 30, 35 and 40 threads. The dashed (resp. solid) blue line corresponds to the reference case without (resp. with) sensitivities. Only for the study of this scaling, two additional series of simulations were done with other reference cases that differ from the first only in the sense that they have 4×10^5 (dotted green curve) and 6×10^5 (dashed–dotted magenta curve) neutrons per batch instead of only 2×10^5. Looking first at the memory, it is barely varying for all simulations (slope of the order of 10^{-3}) as the number of thread increases, which is expected as parallelising

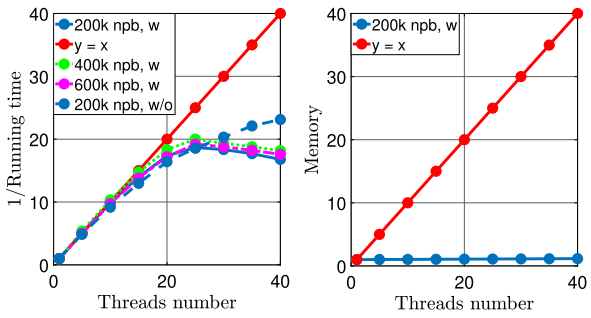


Fig. 5. Running time and memory allocation, respectively at the left and right, as functions of the number of threads on which the simulation is run. The $y = x$ line is in solid red, SERPENT2 data are in dashed (resp. solid) blue for the simulations with 2×10^5 neutrons per batch without (resp. with) sensitivity computations, in dotted green for the simulations with 4×10^5 neutrons per batch and in dashed-dotted magenta for the simulations with 6×10^5 neutrons per batch. The memory slopes are of the order of 10^{-3} for all simulations. 'w' (resp. 'w/o') stands for 'with sensitivity computations' (resp. 'without sensitivity computations') and 'Xk npb' stands for 'X thousand neutrons per batch'.

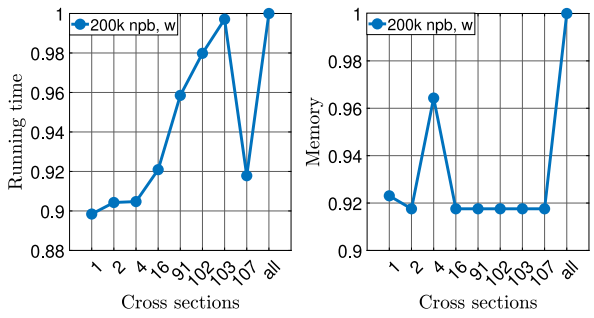


Fig. 6. Running time and memory allocation, respectively at the left and right, as functions of the cross section that is perturbed for each simulation. 2×10^5 neutrons per batch are used in each simulation. Each number corresponds to a reaction channel (MT) 1: total cross section, 2: elastic scattering, 4: inelastic scattering, 16: $(n,2n)$ reaction, 91: inelastic scattering to continuum, 102: (n,γ), 103: (n,p), 107: (n,α), all: all cross sections. 'w' stands for 'with sensitivity computations' and 'Xk npb' stands for 'X thousand neutrons per batch'.

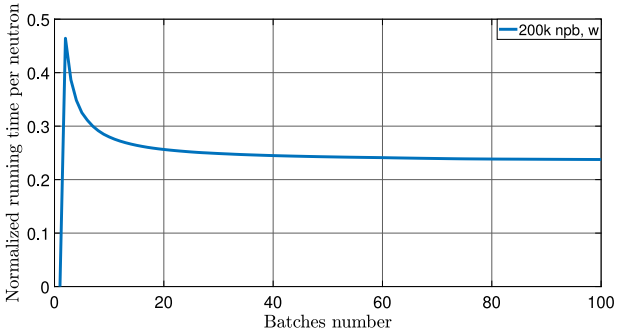


Fig. 7. Running time normalised to the number of neutron as a function of the number of batches during one simulation on 100 batches (use of `set his 1` option in the input file). The peak for low batches numbers is due to the initialisation cost. 'w' stands for 'with sensitivity computations' and 'Xk npb' stands for 'X thousand neutrons per batch'.

the simulations should not increasing the memory consumption when using OPENMP. Focusing on the running time, the '$1/x$' profile which then saturates is also expected. However, several things are surprising and require further work. The first one is that, with sensitivities, the minimum running time is achieved with 25 threads. The second point is that although there is an improvement when going from 2×10^5 neutrons per batch to 4×10^5 neutrons per batch (going from blue to green line), the runtime worsens when increasing again this number to 6×10^5, suggesting a threshold behaviour. Finally, the minimum position does not vary and remains at 25 threads no matter the number of neutrons per batch (even with a test simulation performed with 1×10^6 neutrons per batch).

The influence of the type and number of nuclide/reaction channels for which sensitivity coefficients are determined – chosen for their importance in nuclear reactions – on the running time and memory cost is shown in Fig. 6. The nuclide/reaction pair are represented with their corresponding MT number on the x-axis for isotope W-184. The normalisation is done to the reference case, with 2×10^5 neutrons per batch, 200 batches and with the perturbation of all cross sections. The conclusion that can be drawn from these results is that the choice and the number of cross sections for which sensitivity coefficients are determined do not have a significant impact on the running time or the memory consumption. The difference between computing individual sensitivity vectors or all of them reaches at most 10% for the runtime and barely more than 8% for the memory. The memory cost increases for inelastic scattering ($MT = 4$), which is consistent with the fact that $MT = 4$ is really the sum of various inelastic channels, e.g. more than one reaction for which a sensitivity vector needs to be determined.

Finally, one is interested in the influence of the number of batches on the running time and the memory allocation, keeping the number of neutrons per batch constant. A short series of runs (which is not mentioned here) showed that the memory is constant with the number of batches (slope of the order of 10^{-6}), whereas the running time displayed in Fig. 7 points out that the runtime is overall independent of the number of batches, once the initialisation is done. The initialisation is a fixed cost that is spread amongst all the neutron histories; then each additional batch has a fixed cost, e.g. the runtime per neutron history is constant.

5. Conclusion

The present work addresses the assessment and computational cost evaluation of the recent developments in SERPENT2 to compute sensitivity coefficients in fixed source simulations. The accuracy was evaluated by a direct comparison of the results with the ones of GPT-based deterministic code SUSD3D. The results are generally in good agreement, with a relative gap between SERPENT2 and SUSD3D of 5×10^{-4} for Fig. 2, but high uncertainties are present for low energy range seemingly due to a lack of particles in this energy region, see for instance Fig. 8 where the relative energy reaches 17%.

The computational study has been done with respect to the running time and memory allocation for several parameters (number of neutrons per batch, number of threads, which sensitivity coefficients are computed, number of batches). Putting together the studies presented in Figs. 4–7, one can conclude that an optimal configuration can be reached with the following procedure.

The first step is to set the number of neutrons per batch to the maximum value regarding the available memory and a parameter that has not been studied here, which is the uncertainty that is desired on the sensitivity coefficients. This study is however left for future work. The second step is to increase the number of batches enough to go above the initialisation overshoot and to reach the level of uncertainties that is wanted. Until now, the best parallelisation seems to be obtained with 25 threads.

CRediT authorship contribution statement

P. Griveaux: Investigation, Writing – original draft. **M. Hursin:** Conceptualization, Investigation, Supervision, Writing – review & editing. **I. Kodeli:** Investigation, Supervision. **D. Leichtle:** Funding acquisition, Supervision. **A. Pautz:** Funding acquisition, Supervision.

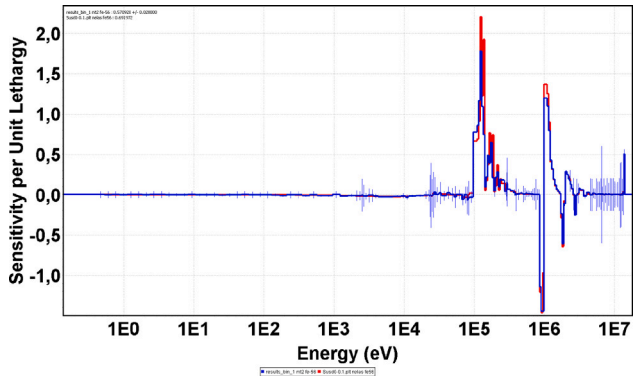


Fig. 8. Sensitivity of the 0 MeV < E < 0.1 MeV neutron flux to elastic scattering of Fe-56. SERPENT2 is in blue, with uncertainty bars, and SUSD3D is in red.

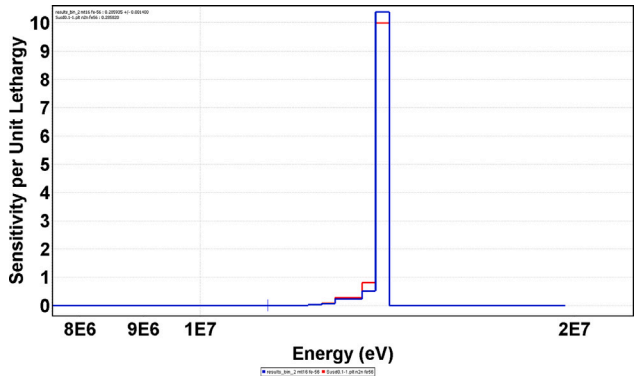


Fig. 9. Sensitivity of the 0.1 MeV < E < 1 MeV neutron flux to $(n, 2n)$ cross section of Fe-56. SERPENT2 is in blue, with uncertainty bars, and SUSD3D is in red.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors would like to thank Thomas Ligonnet, Tom Mager, Edoardo Brunetto and Thomas Guilbaud, all from the Laboratory for Reactor Physics and Systems Behaviour (LRS), EPFL, for fruitful discussions and suggestions during the researches that led to these results. SERPENT2 development team from VTT, is also thanked for the development of sensitivity coefficients computation capabilities in the code. This work has been carried out within the framework of the EUROfusion Consortium. The views and opinions expressed herein do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

Appendix. Serpent2 versus SUSD3D

See Figs. 8–13.

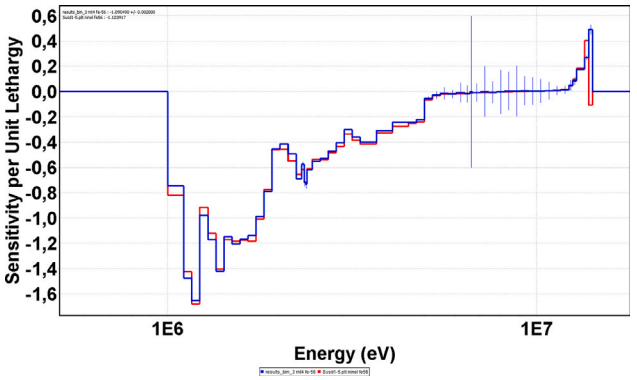


Fig. 10. Sensitivity of the 1 MeV < E < 5 MeV neutron flux to inelastic scattering of Fe-56. SERPENT2 is in blue, with uncertainty bars, and SUSD3D is in red.

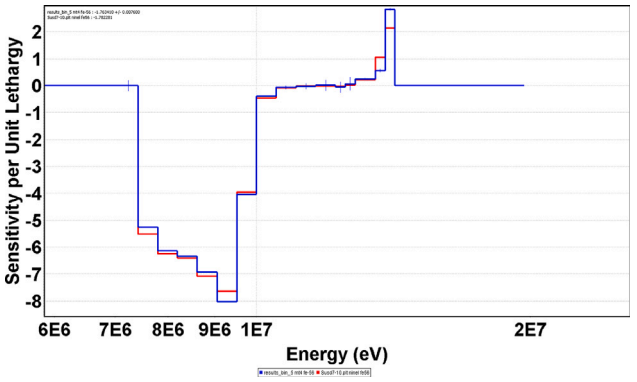


Fig. 11. Sensitivity of the 7 MeV < E < 10 MeV neutron flux to inelastic scattering of Fe-56. SERPENT2 is in blue, with uncertainty bars, and SUSD3D is in red.

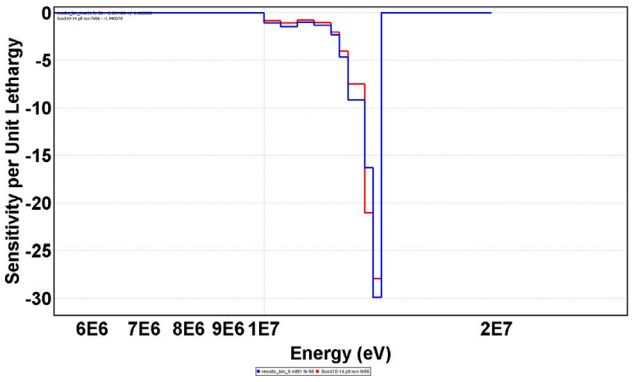


Fig. 12. Sensitivity of the 10 MeV < E < 13.84 MeV neutron flux to inelastic excitation to continuum of Fe-56. SERPENT2 is in blue, with uncertainty bars, and SUSD3D is in red.

Table 2
Values of sensitivities for SERPENT2 (Se-S) and for SUSD3D (SU-S), and of relative uncertainty for SERPENT2 (Se-u) for the different energy ranges. The corresponding figure numbers are present in last column.

Energy (MeV)	Se-S	Se-u	SU-S	Fig
0-0,1	0.570928	0.02	0.691972	8
0,1–1	0.285935	0.0014	0.28582	9
1,0-5,0	−1.09049	0.002	−1.123917	10
5,0-7,0	−1.90507	0.0057	−1.903913	2
7,0-10,0	−1.76341	0.0076	−1.782281	11
10,0-13,84	−2.0344	0.003	−1.940078	12
13,84–14	−1.77084	0.0016	−1.784583	13

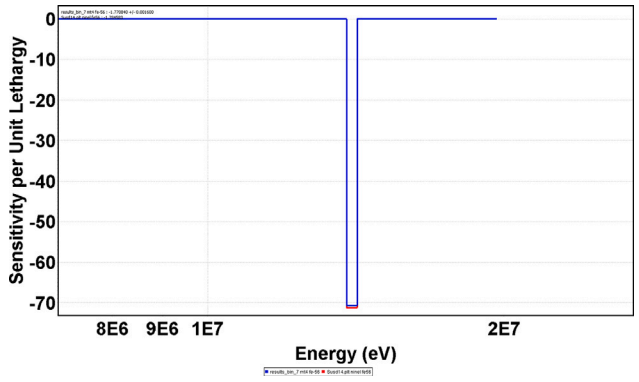


Fig. 13. Sensitivity of the 13.84 MeV $< E <$ 14 MeV neutron flux to inelastic scattering of Fe-56. SERPENT2 is in blue, with uncertainty bars, and SUSD3D is in red.

References

[1] G. Schnabel, H. Sjöstrand, J. Hansson, D. Rochman, A. Koning, R. Capote, Conception and Software Implementation of a Nuclear Data Evaluation Pipeline, Nuclear Data Sheets 173 (2021).

[2] E. Alhassan, D. Rochman, H. Sjöstrand, A. Vasiliev, A.J. Koning, H. Ferroukhi, Bayesian updating for data adjustments and multi-level uncertainty propagation within Total Monte Carlo, Annals of Nuclear Energy 139 (2020).

[3] Manuele Aufiero, Adrien Bidaud, Mathieu Hursin, Jaakko Leppänen, Giuseppe Palmiotti, Sandro Pelloni, Pablo Rubiolo, A collision history-based approach to sensitivity/perturbation calculations in the continuous energy Monte Carlo code SERPENT, Annals of Nuclear Energy 85 (2015) 245–258, https://www.sciencedirect.com/science/article/pii/S0306454915002650.

[4] Brian Kiedrowski, Forrest Brown, Adjoint-based k-eigenvalue sensitivity coefficients to nuclear data using continuous-energy Monte Carlo, Nuclear Science and Engineering 174 (2013) 227–244.

[5] Christopher M. Perfetti, Bradley T. Rearden, Continuous-energy Monte Carlo methods for calculating generalized response sensitivities using tsunami-3D, in: Proceedings of the International Conference on Physics of Reactors, PHYSOR 2014, 2014.

[6] Ho Jin Park, McCARD/MIG stochastic sampling calculations for nuclear cross section sensitivity and uncertainty analysis, Nuclear Engineering and Technology 54 (11) (2022) 4272–4279, https://www.sciencedirect.com/science/article/pii/S1738573322002947.

[7] Guanlin Shi, Yuchuan Guo, Conglong Jia, Zhiyuan Feng, Kan Wang, Shanfang Huang, Quan Cheng, Improvement of sensitivity and uncertainty analysis capabilities of generalized response in Monte Carlo code RMC, Annals of Nuclear Energy 154 (2021) 108099, https://www.sciencedirect.com/science/article/pii/S0306454920307957.

[8] Alejandra Jolodosky, Manuele Aufiero, Massimiliano Fratoni, Exact perturbation theory for fusion blanket design, Annals of Nuclear Energy 131 (2019) 353–366, https://www.sciencedirect.com/science/article/pii/S0306454919301707.

[9] Ivan Kodeli, Multidimensional deterministic nuclear data sensitivity and uncertainty code system: Method and application, Nuclear Science and Engineering 138 (1) (2001) 45–66.

[10] U. Fischer, I. Kodeli, C. Konno, R.L. Perel, Intercomparison of Monte Carlo and SN Sensitivity Calculations for a 14 MeV Neutron Benchmark, in: Andreas Kling, Fernando J.C. Barão, Masayuki Nakagawa, Luis Távora, Pedro Vaz (Eds.), Advanced Monte Carlo for Radiation Physics, Particle Transport Simulation and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN: 978-3-642-18211-2, 2001, pp. 791–796.

[11] U. Fischer, I. Kodeli, R.L. Perel, Benchmark calculations of sensitivities to secondaries' angular distributions, Monte Carlo 2005 Topical Meeting (2005) 1773–1788.

[12] I A Kodeli, Victor Mastrangelo, Evgeny Ivanov, Alexander Aures, Winfried Zwermann, Manuele Aufiero, Yannick Peneliau, Kostadin Ivanov, Enrico Sartori, OECD/NEA intercomparison of deterministic and Monte Carlo cross-section sensitivity codes using SNEAK-7 benchmarks, in: PHYSOR 2016: Unifying Theory and Experiments in the 21st Century, 2016.

[13] Ivan A. Kodeli, XSUN-2022/SUSD3D n/γ, sensitivity-uncertainty code package with recent JEFF-3.3 and ENDF/B-VIII.0 covariance data, in: European Physical Journal Web of Conferences, 281, 2023, p. 00013.

[14] Ivan A. Kodeli, Enrico Sartori, SINBAD – Radiation shielding benchmark experiments, Annals of Nuclear Energy 159 (2021) 108254, https://www.sciencedirect.com/science/article/pii/S0306454921001304.

[15] P Batistoni, M Angelone, L Petrizzi, M Pillon, Neutronics benchmark experiment on tungsten, Journal of Nuclear Materials 329-333 (2004) 683–686, https://www.sciencedirect.com/science/article/pii/S0022311504002776, Proceedings of the 11th International Conference on Fusion Reactor Materials (ICFRM-11).