# Analysis of the fixed-source Option and the weight-window Implementation in Serpent

GRS - 842

Gesellschaft für Anlagen-
und Reaktorsicherheit
(GRS) gGmbH

# Analysis of the fixed-source Option and the weight-window Implementation in Serpent

Anton Travleev

November 2025

**Remark:**

This report refers to the in-house research project 4722E03240 which has been funded by German Federal Ministry for the Environment, Climate Action, Nature Conservation and Nuclear Safety (BMUKN).

The work was conducted by Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH.

The authors are responsible for the content of the report.

**GRS - 842**
**ISBN 978-3-911727-35-8**

**Keywords**

fixed source, neutron flux, radiation transport, Serpent, shielding, variance reduction, weight-window

## Kurzfassung

In dieser Arbeit werden die Optionen des Codes Serpent bewertet, die für Abschirmungsberechnungen relevant sind: der Rechenmodus mit vorgegebener Quelle (*fixed-source*), der Photonentransport und die Anwendung der Methoden zur Varianzreduktion.

Zwei verschiedene Aspekte des Rechenmodus mit vorgegebener Quelle werden getestet. Einer davon ist die Fähigkeit, die Anzahl der verfolgten Teilchen in multiplikativen Medien zu kontrollieren, z. B. um Zerfallsneutronen in einem Lager bestrahlter Brennelemente zu modellieren. Verschiedene in Serpent verfügbare Methoden werden an einem Modell getestet, das eine Anpassung der Neutronenmultiplikation ermöglicht. Die Effizienz der Methoden und deren Einfluss auf das berechnete zeitabhängige Neutronenflussverhalten werden bewertet. Es wird gezeigt, dass die allgemeinste Methode der dynamische Simulationsmodus einer externen Quelle ist (*dynamic external source simulation mode*).

Ein weiterer Aspekt des Rechenmodus mit vorgegebener Quelle ist die Anwendung von Varianzreduktionstechniken mit gewichteten Zellen (*weight-windows*), um die Verteilung der Teilchen im Raum- und Energiebereich zu kontrollieren. Serpent bietet Mittel zur Anwendung eines vom Benutzer vorgegebenen weight-window Parameters, der auf einem Zylinder oder einem rechteckigen räumlichen Gitter sowie auf einem beliebigen Energiegitter definiert ist. Zudem gibt es Optionen zur Generierung solcher weight-window-Gitter global oder für bestimmte Nuklid-Reaktionen. Die iterative Methode zur Generierung von weight-window-Gitter wird an zwei Modellen getestet, die sich im Absorptionsfaktor des Flusses unterscheiden. Einige Empfehlungen zur Wahl des Gitters und der Iterationsparameter können formuliert werden.

Die Modellierung der Physik des Neutronen- und Photonentransports in Serpent wird durch den Vergleich mit experimentellen Ergebnissen und entsprechenden MCNP-Berechnungen validiert. Eines der Abschirmbenchmarks aus dem *International Criticality Safety Benchmark Evaluation Project* (ICSBEP) Handbuch wird mit Serpent berechnet. Die Rechnungen zeigen, dass Serpent vergleichbare Ergebnisse wie MCNP erzielt. Der Effekt einiger Modellierungsoptionen wird im Detail untersucht.

Die durchgeführten Arbeiten zeigen, dass Serpent routinemäßig für Abschirmrechnungen als tragfähige Alternative zu MCNP angewendet werden kann.

## Abstract

In this work, the capabilities of the code Serpent are evaluated, which are relevant for shielding calculations: fixed-source calculation mode, photon transport and application of variance reduction methods.

The fixed-source calculation mode is tested in two different aspects. One is the ability to control population of particle tracks in multiplying media, e. g. to model decay neutrons in an irradiated fuel assembly storage. Several methods available in Serpent are tested on a model that admits adjustment of the neutron multiplication. The efficiency of the methods and their impact on the calculated time-dependent neutron flux behaviour is evaluated. It is shown that the most general method is the dynamic external source simulation mode.

The other aspect of the fixed-source calculation mode is the application of weight-window variance reduction methods to control track populations in space and energy domains. Serpent provides means to apply a user-given weight-window parameter defined on a cylinder or rectangular spatial mesh and on an arbitrary energy grid, as well as means to generate such weight-window meshes globally or for particular responses. The iterative method to generate weight-window meshes is tested on two models that differ by the flux attenuation factor. Some recommendations for the choice of the mesh and the iteration parameters are formulated.

The neutron and photon transport physics modelling in Serpent is validated by comparing against experimental results and correspondent results of MCNP calculations. One of the shielding benchmark problems from the International Criticality Safety Benchmark Evaluation Project (ICSBEP) handbook is calculated with Serpent. The results show that Serpent performs well comparing to MCNP. The effect of some modelling options is investigated in detail.

The performed work shows that Serpent can be applied routinely for shielding calculations as a viable alternative for MCNP.

## Table of content

# 1 Introduction

Serpent simulates particles, as they propagate in space, energy and time. The choice of the initial coordinate, energy and time of each particle is governed by the source definition, either provided by the user or assuming default options by the code itself.

The code Serpent was originally developed as a reactor physics code to model quasi-critical systems in their stationary state. The so-called criticality calculation mode is applied in this case: the initial neutron position is sampled uniformly within the fuel (i.e. spatial regions filled with materials containing fissionable nuclides) if not defined otherwise by the user. Their energy is sampled from the fission spectrum and isotropically. Each neutron history is followed until a fission event. At the fission event, the history is cut, and the fission site and energy of the emitted neutron or neutrons is stored for the next batch. In the next batch the number of histories in the batch is adjusted while the source is sampled from the stored coordinates and energies.

The criticality calculation mode is not suitable for cases, where all particles start from the same predefined source. These cases comprise shielding problems and subcritical systems driven by an external neutron source. This kind of problems can be addressed with the fixed-source (FS) calculation mode in Serpent. In the FS mode, the particle histories in all batches start from a user-defined source distribution, and the particle history is not cut at the fission event.

The class of shielding problems is characterized by a large decrease in the population of particles along the way from the source to detector, which can span many orders of magnitude. In this case the application of variance reduction (VR) techniques is unavoidable to produce statistically relevant results in the regions shielded from the source. In Serpent, the population of tracks close to the region of interest can be controlled by weight-window-meshes (WW-mesh).

When the FS mode is applied for systems with multiplying media, for example to model subcritical systems with an external neutron source, the total amount of tracks to simulate can become impractically large due to new tracks arising from fission neutrons. Serpent provides several methods to limit the number of modelled tracks; they need to be applied carefully, with notion of possible impact of the calculated result.

Photon transport modelling has been added to Serpent since version 2.1.24 /KAL 20/. This feature is important for reactor physics applications, e. g. for the accurate calculation of the energy deposition in fuel and adjacent media, and it is imperative for shielding analyses. The capabilities of Serpent to perform photon transport are tested by benchmarking the performed calculations with experimental data and with previous calculations with MCNP.

## 1.1 Content

The main part of this document provides a general description of the work. Further details are given in appendices.

Specific options of the Serpent code, which are related to the FS calculation mode are tested on a simple geometrical model, which is referred below to as *test model*. The tests cover application of the FS mode in multiplying media, and the generation and application of weight-window-meshes.

Calculations of an international benchmark allow the comparison of Serpent predictions against the evaluated experimental results and previous calculational results obtained with MCNP. The emphasis in this part of this work is on the accurate description of the neutron- and photon spectra of the source and the estimation of the influence of some physics modelling options onto the neutron and photon transport results.

**Main part:**

- Section 2: description of the used test model and tests of the fixed-source calculation mode. The geometry and the main neutron-physics parameters are given.
- Section 3: Fixed-source calculations on the test model with multiplying media. Methods to limit number of tracks are tested and their impact on time-dependent particle transport is analysed.
- Section 4: Generation of weight-window-meshes and performing shielding calculations with the test model. Two model variants are considered: in a less-shielded variant (the neutron flux intensity varies between the source and tallying regions by about 3 orders of magnitude), the methods to generate weight-window-meshes for global variance reduction are tested. In a much more demanding second variant of the model (the neutron flux intensity differs by more than 20 orders

of magnitude), several methods to obtain acceptable weight-window-meshes for global variance reduction are tested. From the findings of these analyses some recommendations are formulated.

- Section 5: Benchmark calculations. Comparison with experimental data and previous MCNP calculations.

**Appendices:**

- Appendix A: Overview of the Serpent input cards, which are relevant for fixed-source shielding calculations.
- Appendix B: Listing of the Serpent input file describing the test model.
- Appendix C: Plots showing all test model shielding calculation results.

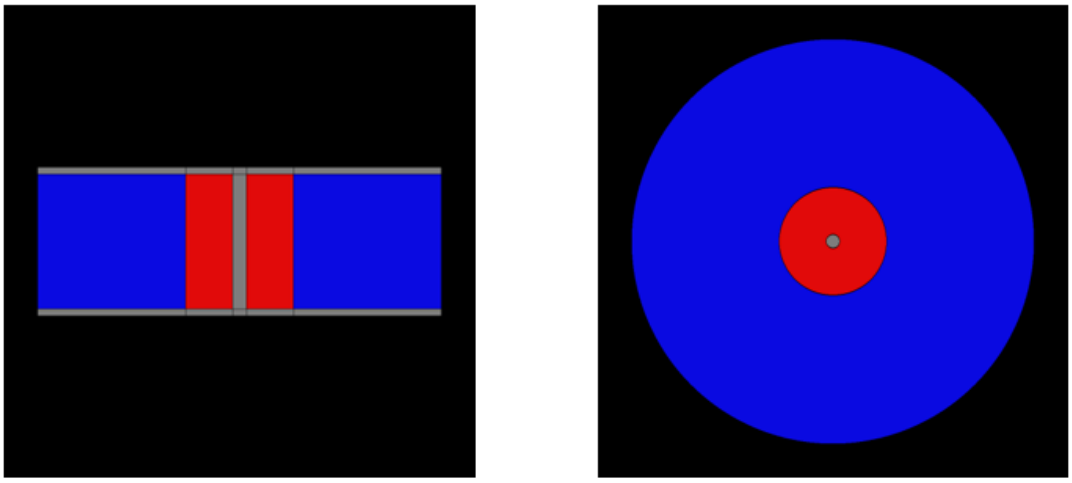
# 2 Test model for fixed-source calculation mode

The model to test the fixed-source (FS) mode related cards represents a vertical metallic uranium cylinder of 8 cm radius and 20 cm height with a central channel along the axis of 1 cm radius. The cylinder is surrounded radially by a 22 cm thick radial layer of water (its outer radius is 30 cm). The uranium enrichment can be adjusted to simulate multiplying media of different multiplication factors.

The choice of the model is governed by the following criteria:

- The model must contain fissionable material to make the criticality calculation mode possible. The multiplication factor can be adjusted by the $^{235}U$ content in metallic uranium.
- The source and detectors are separated by a shielding region, optionally with neutron multiplying media. The source is positioned at the lower face of the cylinder, one of the detectors is positioned at the upper end.
- A channel for neutron streaming is present.
- The vertical cylinder geometry is chosen since the cylinder mesh detector definition is straightforward in this geometry (other directions would require a local coordinate system to define the cylindrical mesh).
- The neutron source is homogeneously distributed within a region of 1 cm height in the cylinder below the bottom face of the uranium cylinder. The initial neutron energy is 1 MeV.

Typical input file describing the model is shown in Listing B.1 in Appendix B. The plots generated by Serpent are shown in Fig. 2.1.



(a) Vertical section (b) Horizontal section

**Fig. 2.1** Test model geometry

The colors represent different materials and regions: red is the metallic uranium cylinder; blue is the water reflector; gray represents almost voided regions, filled with He-4 with negligible density.

## 2.1 Multiplication factor

The multiplication factor of the test model can be adjusted by the content of $^{235}U$ in the uranium cylinder. The values of $k_{eff}$ for different $^{235}U$ content is shown in Tab. 2.1.

The $k_{eff}$ values are obtained in the criticality calculation mode calculations with delayed neutron emission turned on (`set delnu 1`). The following analysis uses the same model for the FS calculation mode. Therefore, it is important to consider how Serpent treats delayed neutrons (DN). In criticality calculation mode, DN emission is enabled by default, so the results include estimates for the total and prompt $k_{eff}$, as well as the DN fraction. In FS calculation mode, the DN emission is set to off by default. We will be interested in the neutron flux time behaviour at time intervals where DN are important, therefore all calculations are done with DN emission turned on.

**Tab. 2.1** Multiplication factor $k_{eff}$ and CPU calculation times as function of $^{235}U$ content

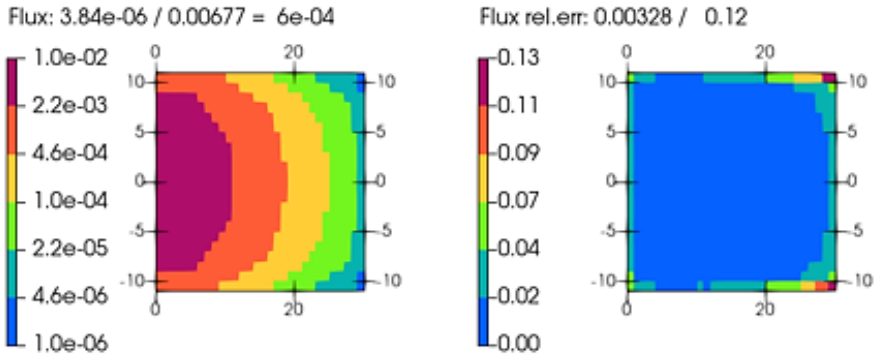
| $^{235}U$ content in wt.% $^{235}U$ | $k_{eff, tot}$ | rel. error | $k_{eff,promt}$ | rel. error | CPU-time, in min |
|---|---|---|---|---|---|
| 0 | 0.18036 | 0.00281 | 0.18016 | 0.00282 | 13.11 |
| 1 | 0.34710 | 0.00187 | 0.34434 | 0.00189 | 12.23 |
| 10 | 0.56029 | 0.00137 | 0.55553 | 0.00137 | 11.06 |
| 20 | 0.68467 | 0.00117 | 0.67861 | 0.00117 | 10.68 |
| 30 | 0.78587 | 0.00109 | 0.77953 | 0.00110 | 12.64 |
| 40 | 0.87555 | 0.00099 | 0.86867 | 0.00098 | 11.29 |
| 50 | 0.95127 | 0.00093 | 0.94426 | 0.00091 | 12.26 |
| 55 | 0.98666 | 0.00091 | 0.97951 | 0.00085 | 11.95 |
| 57 | 1.00235 | 0.00096 | 0.99501 | 0.00094 | 11.48 |
| 59 | 1.01481 | 0.00097 | 1.00681 | 0.00095 | 10.10 |
| 60 | 1.01969 | 0.00097 | 1.01209 | 0.00096 | 11.30 |

From Tab. 2.1 follows, that the case with 57 % of $^{235}U$ content is critical with delayed neutrons. The cases with lower $^{235}U$ content are subcritical. Cases and with an amount of $^{235}U$ larger than or equal to 59 % are prompt supercritical.

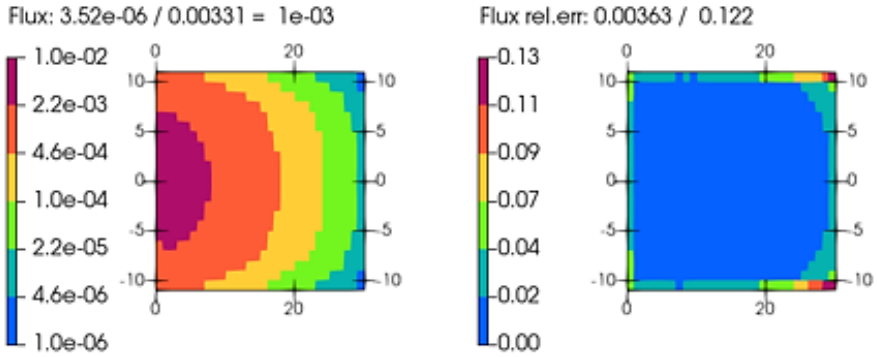
## 2.2 Flux spatial distribution

The difference between criticality and fixed-source calculation modes can be seen in the results, obtained for the most-subcritical (no $^{235}U$) and delayed-critical (57 % of $^{235}U$) cases.

The RZ distribution of the neutron flux and its statistical error, which is obtained from the criticality calculation mode for the most subcritical and the delayed-critical configuration, are shown in Fig. 2.2. Given in the figure are the ratios of the minimum to the maximum flux, $3.84 \times 10^{-6} / 0.00677 = 6 \times 10^{-4}$ for the most subcritical configuration. For the delayed-critical configuration this value is $3.52 \times 10^{-6} / 0.00331 = 1 \times 10^{-3}$.



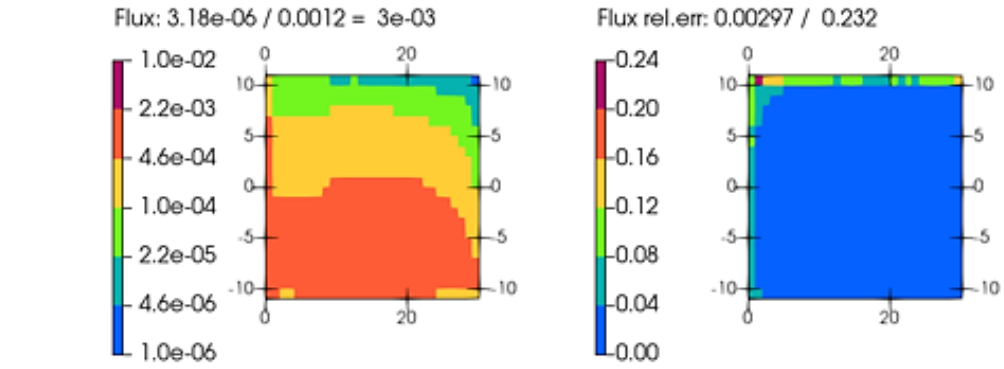

*(a) No U-235 (most subcritical configuration)*



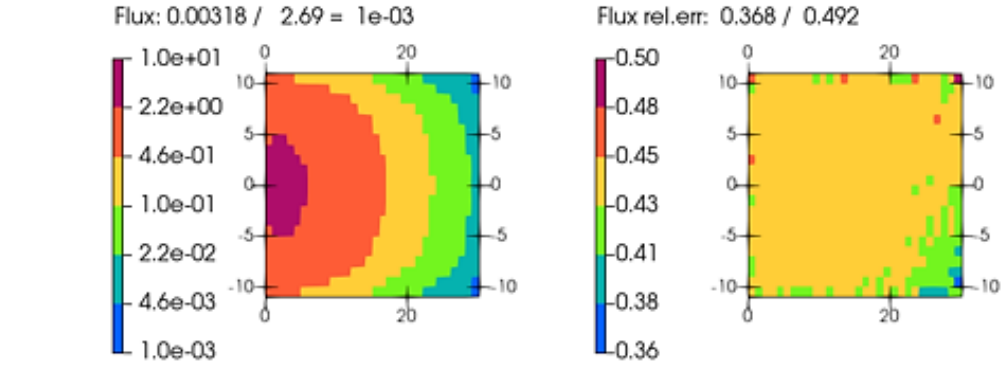

*(b) 57% U-235 (delayed-critical configuration)*

**Fig. 2.2** Neutron flux RZ distribution (left) an its statistical error (right), criticality calculation mode

In these plots, it is shown that the neutron flux and its statistical error have similar spatial distribution for both deeply subcritical and delayed-critical cases. Note that also neutron flux minimum and maximum values are similar in the two cases. The statistical error for neutron flux is below 1 % everywhere, except in regions with low density in the central channel and regions below and above the uranium cylinder and the water reflector.




*(a) No U-235*




*(b) 57% U-235*

**Fig. 2.3** Neutron flux RZ distribution and its error, FS calculation mode

The results from the fixed-source (FS) calculation mode are shown in Fig. 2.3. Here one can see very different behaviour of the neutron flux spatial distribution as well as the statistical error, depending on the neutron multiplication properties of the system. In the deeply subcritical system, Fig. 2.3 (a), the maximal neutron flux is in the vicinity of the neutron source, i. e. in the lower part of model. The statistical error follows- this pattern. In the delayed-critical system, Fig. 2.3 (b), the fission neutrons contribution prevails so one can see the neutron flux distribution like that obtained from the criticality calculation mode. However, the statistical error is above 40 % almost everywhere in the model. This value is considerably larger than that seen in the deeply subcritical case. This behaviour is due to very large (theoretically infinite) dispersion of the number of fission events in the neutron histories, caused by the critical multiplying media.

# 3 Time-dependent flux in fixed-source mode in multiplying media

The large number of fission events in the neutron history affects not only the statistical error (see Fig. 2.3) but also reduces the effectiveness of parallel execution or even may result in a calculation abort.

Serpent stores particles that arise in collisions for later transport in a dedicated memory block. The size of this block is defined at the begin of simulations. Its default size is appropriate for criticality calculations wherein neutron histories are terminated upon occurrence of a fission event.

In FS calculation mode, all fission neutrons are banked to memory for later transport and in case of many fission events the size of the memory block becomes insufficient. Consequently, this results in an abort.

After each batch, Serpent reports the average CPU usage. In a multi-threading calculation, the maximal CPU usage is N × 100 %, where N is the number of threads used by Serpent. This is either defined by the user in the command line as `-omp <N>` or implied from the environment, e. g. when started under an MPI wrapper. It was observed during FS calculations for systems with different multiplication factors that the higher the neutron multiplication, the larger is the dispersion of CPU usage. For example, for the case with no $^{235}U$, the CPU usage was reported to be close to its maximum value throughout all batches. For the delayed-critical case it changed between slightly above 100 % (Serpent has used only 1 CPU from all specified CPUs most of the time of the batch simulation) to the maximum value.

Tab. 3.1 shows the results of the FS mode calculations in the models with different multiplication factors. For the cases with supercritical multiplying media, Serpent did not complete due to insufficient neutron buffer size. Note that in spite of increased CPU time, the relative statistical error of the calculated value increases as the multiplying media becomes closer to critical.

**Tab. 3.1** Time-integrated neutron flux below water reflector, without cut-offs

| $^{235}$U content in wt.% $^{235}$U | neutron flux | rel. error | CPU time, in min |
|---|---|---|---|
| 0 | 1.21782E+00 | 0.00316 | 12.11 |
| 01 | 1.23830E+00 | 0.00279 | 14.47 |
| 10 | 1.27385E+00 | 0.00301 | 15.50 |
| 20 | 1.30895E+00 | 0.00288 | 16.54 |
| 30 | 1.34748E+00 | 0.00278 | 23.33 |
| 40 | 1.42147E+00 | 0.00269 | 29.77 |
| 50 | 1.71795E+00 | 0.00440 | 48.61 |
| 55 | 3.01590E+00 | 0.01642 | 99.57 |
| 57 | 1.46743E+02 | 0.43145 | 343.33 |
| 60 | - | - | - |
| 70 | - | - | - |

## 3.1 Flux time-dependence in fixed-source calculation mode

The problem with FS calculation mode in multiplying media is best understood by analysing the time behaviour of the simulated particle flux. Results of the time-dependent neutron flux tallied in the lower (gray regions on Fig. 2.1) for several multiplication factors are shown in Fig. 3.1 to Fig. 3.3. The results for the upper regions are shown in Fig. 3.4 to Fig. 3.6. In all figures the flux is shown without cut-offs for the models with $^{235}$U content from 0 (most subcritical) up to 57 % (delayed-neutron critical).

In the delayed-critical configuration (57 % of $^{235}$U), the neutron flux remains almost constant in time after delayed neutrons appear in the system. The plots show results for detectors with time bins covering an interval from $1 \times 10^{-9}$ s to 10 s (see time bin `timeBins100` in the input in Listing B.1 in Appendix B), while the neutron transport is modelled by Serpent also outside this range. By default, i. e. without any explicit time-cutting settings provided by the user, there is no history cut-off by time at all. Therefore, the number of histories in the delayed-critical configuration does not die out with time and the value of time-integrated neutron flux can become arbitrarily large and arbitrarily statistically uncertain. This behaviour is seen in Tab. 3.1: the value for the delayed-critical configuration differs by two orders of magnitude and has considerable statistical uncertainty.

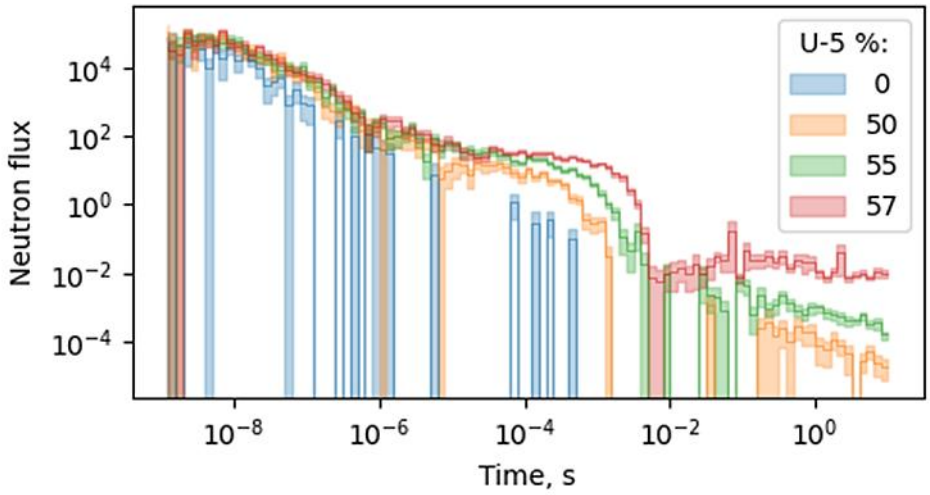



**Fig. 3.1** Time-dependent neutron flux in the lower tally regions: below channel

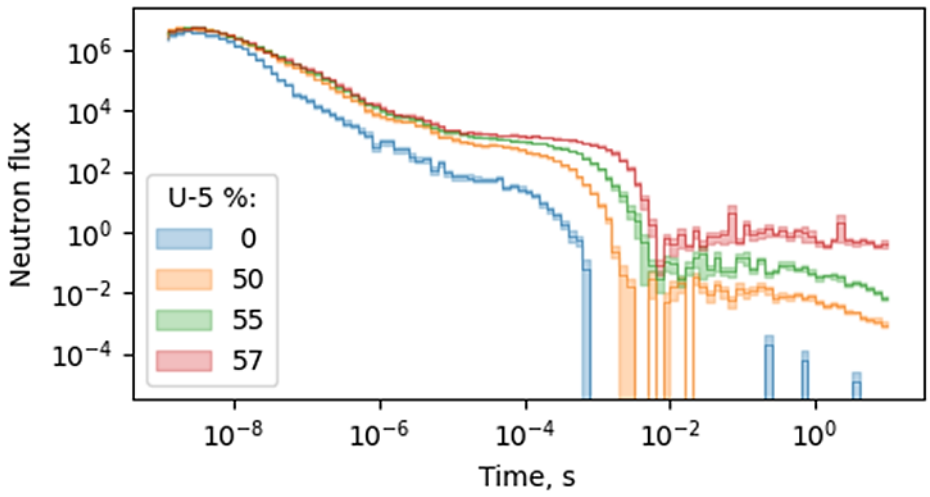



**Fig. 3.2** Time-dependent neutron flux in the lower tally regions: below U cylinder

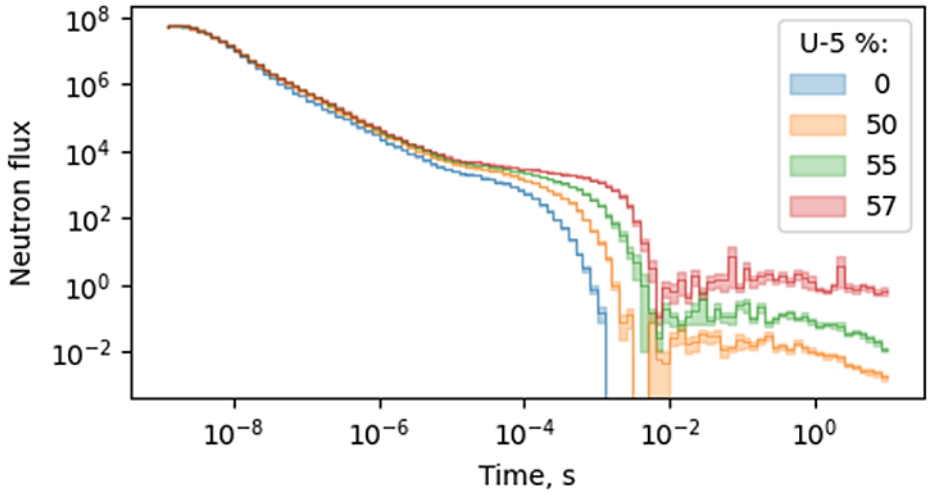



**Fig. 3.3** Time-dependent neutron flux in the lower tally regions: below reflector

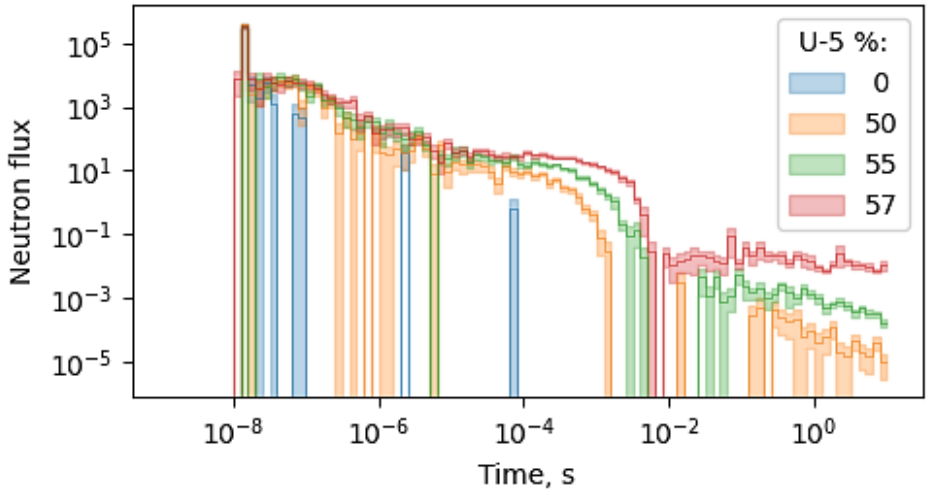



**Fig. 3.4** Time-dependent neutron flux in the upper tally regions: above channel

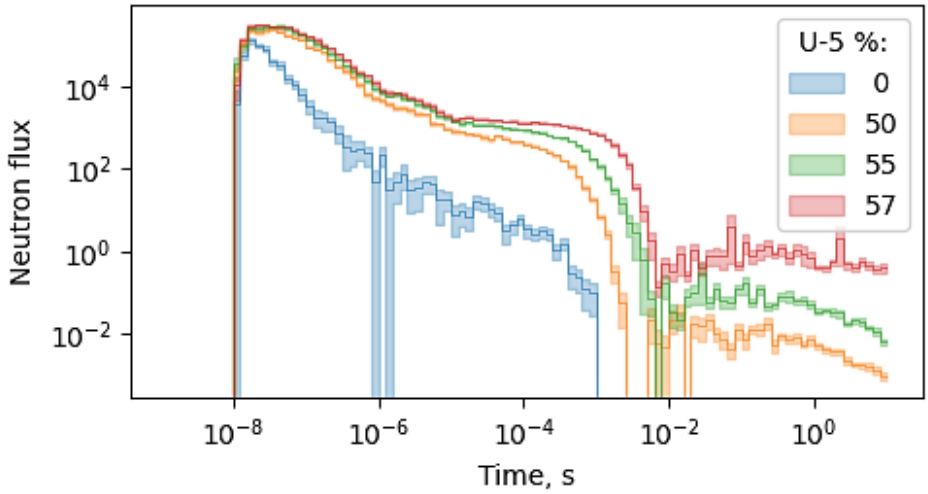



**Fig. 3.5** Time-dependent neutron flux in the upper tally regions: below U cylinder

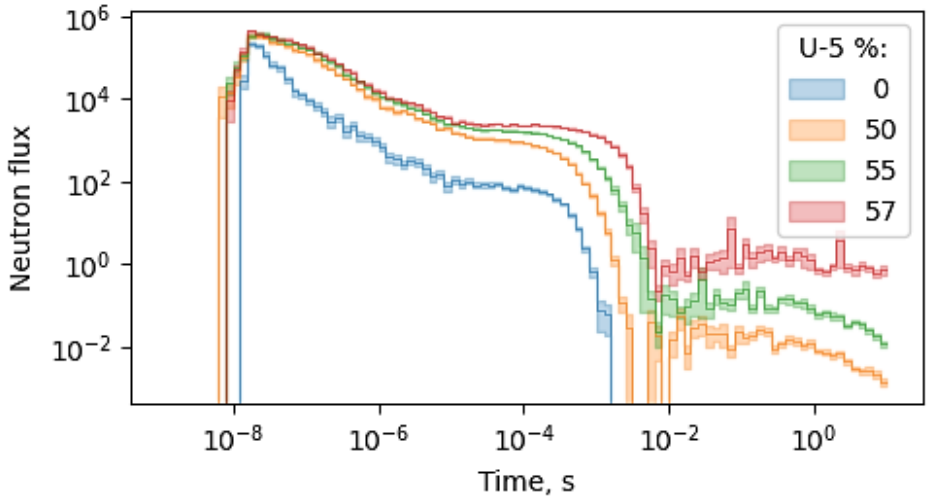



**Fig. 3.6** Time-dependent neutron flux in the upper tally regions: below reflector

The problem with unbound grows of tracks in multiplying media during FS calculation can be addressed in Serpent in several ways. First, one can directly specify to cut-off histories that exceed certain time- or certain amount of fission-events. Second, one can specify times, at which the total amount of tracks will be adjusted. These methods are considered in the sections below.

### 3.1.1 Cut-offs for time and generations

The `set tcut <Tmax>` card sets the time cut-off for particle tracks.

The `set gcut <N>` card sets the generation cut-off: the tracks will be killed after N fission events. This is similar to the time cut-off but does not require the knowledge of the generation time of the modelled system.

The impact of the time and generation cut-offs is analysed for the tally defined in the region below the water reflector for several models, from the least critical (no $^{235}$U) up to the delayed-critical (57 % of $^{235}$U).

#### 3.1.1.1 Time cut-off

Two cases are considered: `set tcut 1e-4` and `set tcut 1e-2`. The time dependence is shown in Fig. 3.7 to Fig. 3.9. The correspondent values of the time-integrated neutron flux, together with CPU times, are shown in Tab. 3.2 and Tab. 3.3.

The effect of applying time cut-off is clearly seen in the figures: there is no flux beyond this time. Respectively, the time-integrated flux values are reduced (compare with values in Tab. 3.1). The effect of cut-off is minimal in the most subcritical case since there are only few tracks reaching the cut-off time, but it is considerable in the delayed-critical case.

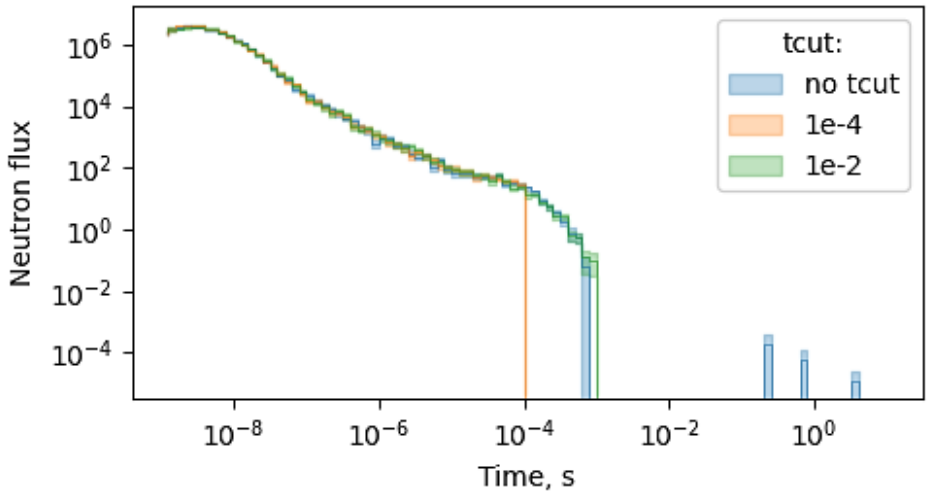



**Fig. 3.7** Time-dependent neutron flux below water reflector with `tcut` card: No-$^{235}$U

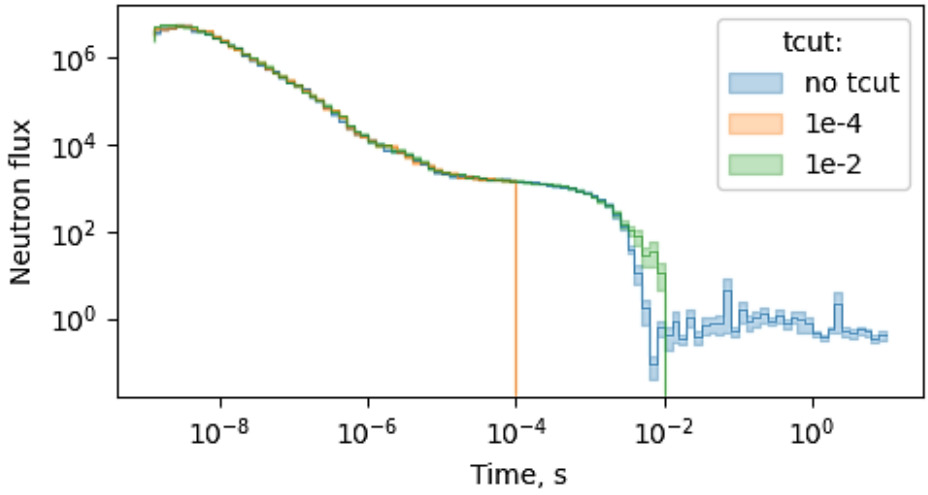



**Fig. 3.8** Time-dependent neutron flux below water reflector with `tcut` card: 57 % $^{235}$U

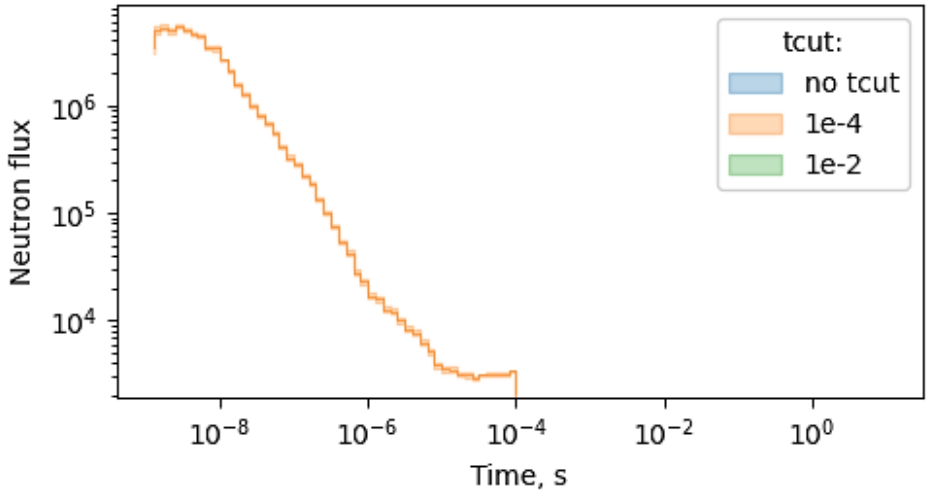



**Fig. 3.9** Time-dependent neutron flux below water reflector with `tcut` card: 60 % $^{235}$U

**Tab. 3.2** Time-integrated neutron flux below water reflector, `set tcut 1e-4`

| $^{235}$U content in wt.% $^{235}$U | neutron flux | Rel. Error | CPU time, in min |
|---|---|---|---|
| 0 | 1.16094E+00 | 0.00300 | 3.85 |
| 01 | 1.17087E+00 | 0.00301 | 4.34 |
| 10 | 1.19772E+00 | 0.00290 | 4.68 |
| 20 | 1.21153E+00 | 0.00276 | 5.51 |
| 30 | 1.23808E+00 | 0.00289 | 7.00 |
| 40 | 1.27637E+00 | 0.00290 | 11.75 |
| 50 | 1.35681E+00 | 0.00319 | 16.70 |
| 55 | 1.46615E+00 | 0.00339 | 21.34 |
| 57 | 1.54623E+00 | 0.00413 | 26.10 |
| 60 | 1.78371E+00 | 0.00541 | 33.95 |
| 70 | - | - | - |

**Tab. 3.3** Time-integrated neutron flux below water reflector, `set tcut 1e-2`

| $^{235}$U content in wt.% $^{235}$U | neutron flux | Rel. Error | CPU time, in min |
|---|---|---|---|
| 0 | 1.21887E+00 | 0.00289 | 10.84 |
| 01 | 1.24514E+00 | 0.00309 | 11.39 |
| 10 | 1.27012E+00 | 0.00281 | 12.82 |
| 20 | 1.30172E+00 | 0.00286 | 15.23 |
| 30 | 1.33855E+00 | 0.00277 | 17.16 |
| 40 | 1.41379E+00 | 0.00298 | 21.64 |
| 50 | 1.67339E+00 | 0.00414 | 37.41 |
| 55 | 2.41494E+00 | 0.01137 | 72.44 |
| 57 | 5.06408E+00 | 0.04496 | 116.58 |
| 60 | - | - | - |
| 70 | - | - | - |

Notes on the use of `set tcut` card:

- The use does not always succeed. For prompt supercritical systems, both time cut-offs set either to $10^{-4}$ or $10^{-2}$ did not prevent from calculation abort for the case with 70 % $^{235}$U (most prompt supercritical). The prompt supercritical case with 60 % $^{235}$U content was completed only with $10^{-4}$ s cut-off time, however, the time-dependent flux calculated for this case, see lower plot in Fig. 3.8, shows decrease in time, while an increase is expected for a prompt-supercritical configuration.
- It provides consistent results for times before cut-off. However, there is some difference for the delayed critical case just before the cut-off time $10^{-2}$, see Fig. 3.8.
- When time cut-off is applied, Serpent divides tally results reported in the output file `<input>_det0.m` by the cut-off time. To compare results of calculations with different time cut-offs, this fact needs to be taken into account: either renormalize at postprocessing, or adjust normalization in the input file, e.g. `set srcrate <1/Tmax>`, where `<Tmax>` is the time cut-off.
- The time-integrated values are affected by the time cut-off, so far there is non-zero flux after the time cut-off. Compare values in Tab. 3.1, Tab. 3.2 and Tab. 3.3.

#### 3.1.1.2 Generation cut-off

The number of simulated neutron tracks can be controlled in Serpent also by setting the maximal number of fission events in the history with the generation cut-off card `gcut`. Two cut-offs were considered, for 10 and 100 generations.

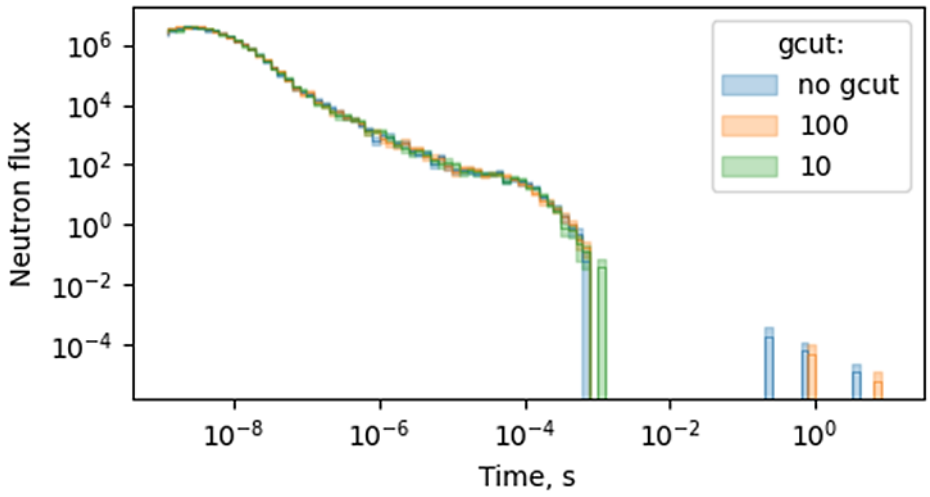



**Fig. 3.10** Time-dependent neutron flux below water reflector with `gcut` card: no $^{235}U$

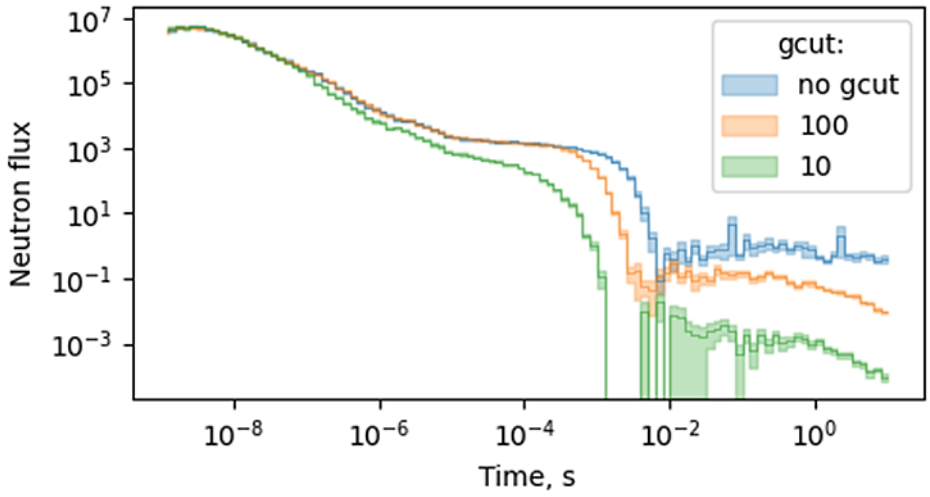



**Fig. 3.11** Time-dependent neutron flux below water reflector with `gcut` card: 57 % $^{235}U$

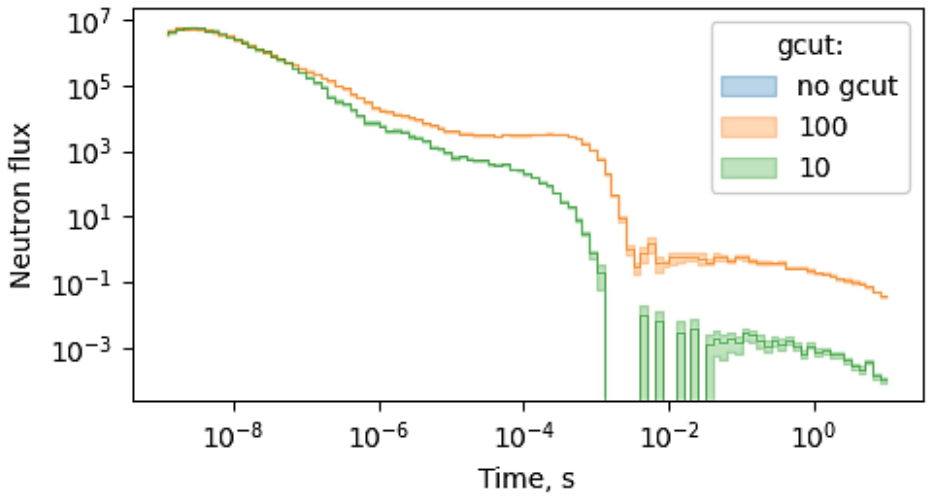



**Fig. 3.12** Time-dependent neutron flux below water reflector with `gcut` card: 60 % $^{235}$U

**Tab. 3.4** Time-integrated neutron flux below water reflector, `gcut 100`

| $^{235}$U content in wt.% $^{235}$U | neutron flux | Rel. Error | CPU time, in min |
|---|---|---|---|
| 0 | 1.22107E+00 | 0.00321 | 6.33 |
| 01 | 1.24080E+00 | 0.00282 | 7.28 |
| 10 | 1.27192E+00 | 0.00290 | 10.00 |
| 20 | 1.30852E+00 | 0.00276 | 13.00 |
| 30 | 1.33801E+00 | 0.00306 | 15.55 |
| 40 | 1.43444E+00 | 0.00302 | 19.02 |
| 50 | 1.73127E+00 | 0.00417 | 34.72 |
| 55 | 2.54439E+00 | 0.00869 | 62.88 |
| 57 | 3.59062E+00 | 0.01089 | 96.56 |
| 60 | 8.72998E+00 | 0.01266 | 271.78 |
| 70 | - | - | - |

**Tab. 3.5** Time-integrated neutron flux below water reflector, `gcut 10`

| $^{235}$U content in wt.% $^{235}$U | neutron flux | Rel. Error | CPU time, in min |
|---|---|---|---|
| 0 | 1.22107E+00 | 0.00321 | 6.33 |
| 01 | 1.24080E+00 | 0.00282 | 7.28 |
| 10 | 1.27192E+00 | 0.00290 | 10.00 |
| 20 | 1.30852E+00 | 0.00276 | 13.00 |
| 30 | 1.33801E+00 | 0.00306 | 15.55 |
| 40 | 1.43444E+00 | 0.00302 | 19.02 |
| 50 | 1.73127E+00 | 0.00417 | 34.72 |
| 55 | 2.54439E+00 | 0.00869 | 62.88 |
| 57 | 3.59062E+00 | 0.01089 | 96.56 |
| 60 | 8.72998E+00 | 0.01266 | 271.78 |
| 70 | - | - | - |

Notes on the use of `set gcut` card:

- Even for the prompt supercritical configuration, the calculation has been completed. The time dependent results, however, are distorted: note the comparison with the case without cut-off in Fig. 3.11 and note the decreasing time behaviour for a supercritical configuration in Fig. 3.12.
- No sharp time boundary as in the case of time cut-off.

### 3.1.2 Time dependent population control

Serpent provides capability to control the number of tracks at certain times. This is called “dynamic external source simulation mode”. At the time points specified by the user the simulated population size is adjusted to match the starting population. At the same time, the total track weight is adjusted to preserve the amount of modelled physical particles.

The mode is activated by providing a time bin to the `set nps` card, see Listing 3.1. For this analysis, the time bin structure spans up to 10 s, see Listing 3.2.

**Listing 3.1** Input cards relevant for the dynamic external source simulation mode

```
set nps <N> <batches>  timeBinsForPopulationControl
tme timeBinsForPopulationControl 1 0.0 <t1> <t2> ... <tN>
```

**Listing 3.2** Time bin structure for dynamic mode

```
tme timeBins010 1
        0.0
        1e-8 3e-8 5e-8
        1e-7 3e-7 5e-7
        1e-6 3e-6 5e-6
        1e-5 3e-5 5e-5
        1e-4 3e-4 5e-4
        1e-3 3e-3 5e-3
        1e-2 3e-2 5e-2
        1e-1 3e-1 5e-1
        1e-0 3e-0 5e-0
        1e+1
```

The time-dependent neutron flux below the water reflector for cases without $^{235}U$ are shown in Fig. 3.13, with 57 % of $^{235}U$ are shown on Fig. 3.14. Time-integrated tally results are listed in Tab. 3.6.

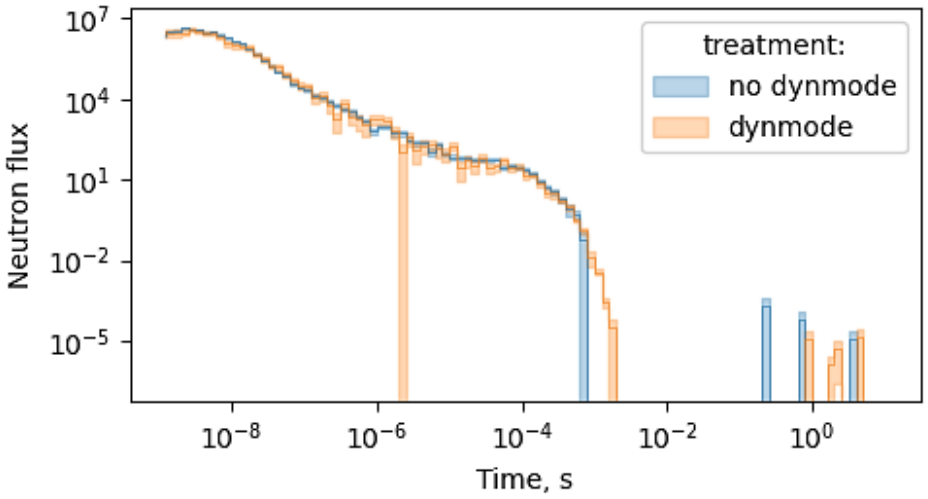



**Fig. 3.13** Time-dependent neutron flux below water reflector, dynmode: No $^{235}U$

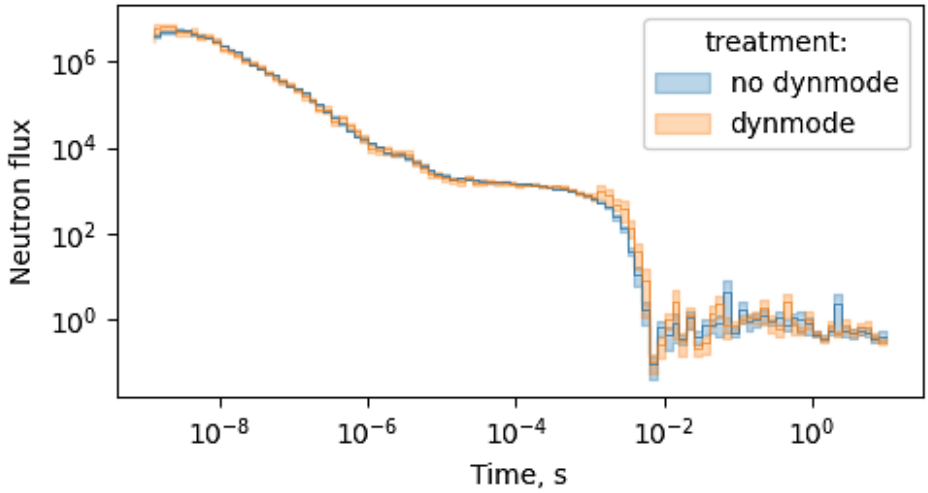



**Fig. 3.14** Time-dependent neutron flux below water reflector, dynmode: 57 % $^{235}$U

**Tab. 3.6** Time-integrated neutron flux below water reflector, dynmode

| $^{235}$U content in wt.% $^{235}$U | neutron flux | Rel. Error | CPU time |
|---|---|---|---|
| 0 | 1.22227E+00 | 0.00835 | 4.38 |
| 01 | 1.23583E+00 | 0.00941 | 6.20 |
| 10 | 1.25361E+00 | 0.00735 | 7.41 |
| 20 | 1.30816E+00 | 0.00801 | 9.84 |
| 30 | 1.33774E+00 | 0.00851 | 13.25 |
| 40 | 1.41408E+00 | 0.00767 | 21.00 |
| 50 | 1.69988E+00 | 0.00972 | 49.60 |
| 55 | 2.71999E+00 | 0.02140 | 91.70 |
| 57 | 1.52864E+01 | 0.12784 | 163.01 |
| 60 | - | - | - |
| 70 | - | - | - |

Notes on the dynamic mode:

- Serpent normalizes results to the first-time interval and cuts all histories at the last time point of the time bin structure used for the dynamic mode. For the time bins the normalization factor is $10^8$ and transport is modelled up to 10 s, see Listing 3.2.
- It provides accurate time-dependent results.
- To specify time bins for time-dependent population control, one needs to know preliminary information about the system time behaviour (that is where preliminary runs with time or generations cut-offs can be applied).
- The chosen time bins did not allow to model prompt super-critical configurations
- For subcritical configurations, time-integrated tallies are in good agreement with the results without dynamic mode treatment, see Tab. 3.1. For the delayed-critical configuration ($^{235}$U content of 57 %), the results still differ by an order of magnitude. The lower value in the dynamic mode is explained by the time cut-off set to 10 s.

# 4 Application of weight-window-mesh

The model used in section 3 to test the time-dependent transport in multiplying media with the fixed-source calculation mode, is used in this section to test application of weight-window-mesh (WW-mesh) for variance reduction (VR).

The Generation of WW-meshes and its application is first tested for the most subcritical configuration (no $^{235}U$), see section 4.1. For this configuration, the neutron flux intensity differs throughout the model geometry by not more than 3 orders of magnitude and the whole spatial domain is sampled already in the calculation without any global variance reduction, therefore the WW-mesh generation for global variance reduction does not require iterations. This model is referred to as less-shielded model.

To test the generation of the WW-mesh iteratively in Serpent, a modified model was set up, that contains Boron with 10 % of B-10 instead of metallic U. This model is referred to below as highly shielded model. This model is characterized by a flux intensity range of more than 20 orders of magnitude, so that only a small spatial region around the source is sampled with tracks in calculations without variance reduction techniques applied. This makes the iterative approach in generating the global variance reduction WW-mesh unavoidable. Several tactics to organize such iterations are considered in section 4.2.

## 4.1 Less-shielded model

For the most-subcritical model considered above, shielding calculations are performed with the neutron source placed at the bottom of the cylinder. The flux is calculated in a mesh detector to obtain spatial distribution over the whole geometry, and in the cell detectors in the regions below and above the model (grey regions in Fig. 2.1).

Four Serpent runs are performed for the model to test several options of the variance reduction (VR) technique:

1. No WW-mesh is applied, to obtain reference results.
2. Several iterations to generate the WW-mesh for global VR and for VR in the upper cell detectors (in the following referred to as local VR).
3. The global VR WW-mesh from run 2 is applied.
4. The local VR WW-mesh from run 2 is applied.

The mesh detector results of run 1 are shown in Fig. 4.1. The flux is sampled everywhere in the geometry. The ratio between the minimum and the maximum flux intensity is about $2 \times 10^{-3}$. The maximum statistical error of about 20 % is located at the cylinder's top.

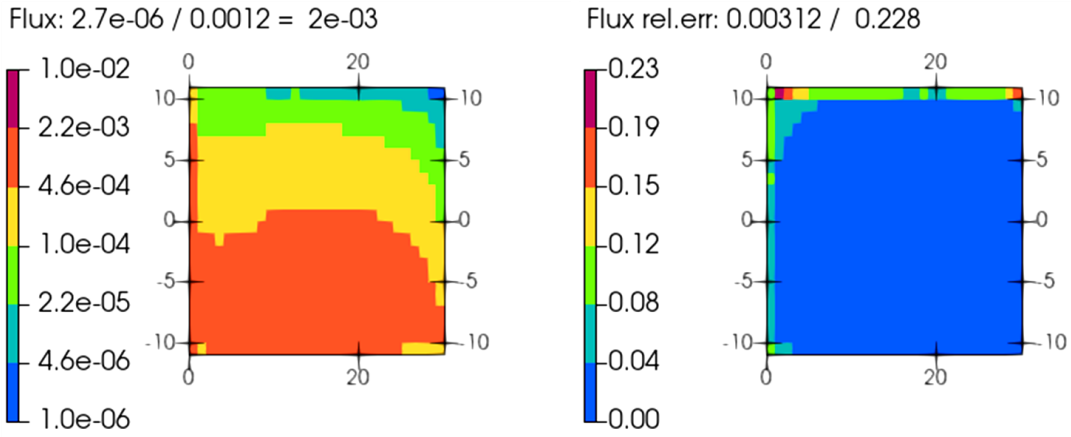



**Fig. 4.1** RZ flux spatial distribution, no VR. Left: neutron flux, right: rel. error

In the next run, the iterative calculations are organized: on each iteration, the WW-mesh generated on the previous iteration is used. Relevant parts of the input file are shown in Listing 4.1. For the first iteration, no WW-mesh is applied. The first three iterations generate a global WW-mesh (see the `wwgen wwGlobal2` card), which should improve track sampling in the whole model. In the last two iterations, the WW-mesh is generated to improve statistics locally in the flux detectors at the cylinder's top (see the `wwgen wwUpper` card).

**Listing 4.1** Input for the iterative generation of WW-mesh. The `wwgen` card(s) define the mesh geometry and target detector. The `wwin` card sets the number of iterations, the mesh to generate and the global density factor

```
% Generate weight-window for global VR
wwgen wwGlobal2
    1e-9 10000
    3 -1
    8 16 1 22
    0 1.0
      2.0
      3.0
      4.0
      5.0
      6.0
      7.0
      8.0
    9.0 10.0 13 16 18 23 28 32
    0 360
    -11.1 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11.1

wwgen wwUpper
```

```
    1e-9 10000
    1 -1
    8 16 1 22
    0 1.0
      2.0
      3.0
      4.0
      5.0
      6.0
      7.0
      8.0
    9.0 10.0 13 16 18 23 28 32
    0 360
    -11.1 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11.1
    detUpper 1.0

% Global VR iterations
wwin wwGlobal
    wi 1 5
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwUpper 1
    wwUpper 1
```

The R-Z maps of the flux (left), its relative error (center) as well as the importance calculated (right) on each iteration, are shown in Fig. 4.2 to Fig. 4.6. The data is organized by cards according to Listing 4.1. As one can see, already after the first iteration the importance distribution for the global VR is settled and is not improving furthermore in the next iterations. In the last two iterations, the importance with respect to the detectors located at the cylinder's top is calculated. Therefore, its distribution in the last two iterations differs from the previous ones.

The application of the global VR WW-mesh helps to achieve statistical errors below 1 % in a larger spatial region (compare errors in Fig. 4.2 and Fig. 4.3). In the last iteration, the application of the local VR WW-mesh helps to improve statistical error in the upper axial layer, i. e. in the region of the targeted cell detector (compare errors in Fig. 4.5 and Fig. 4.6).

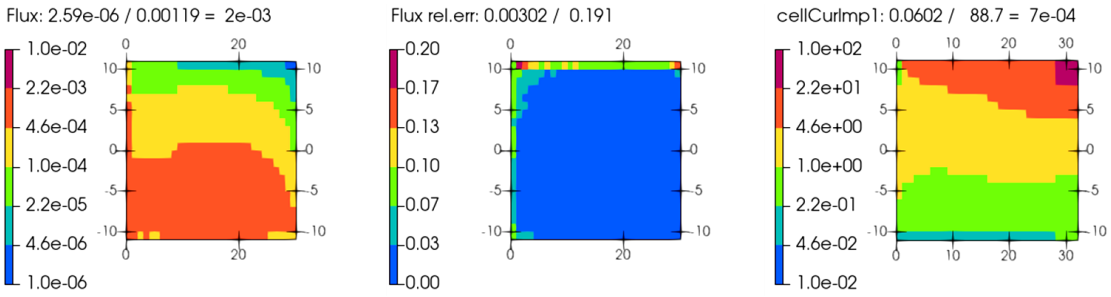



**Fig. 4.2** Flux (left), its rel. error (mid) and importance (right): Iteration 0

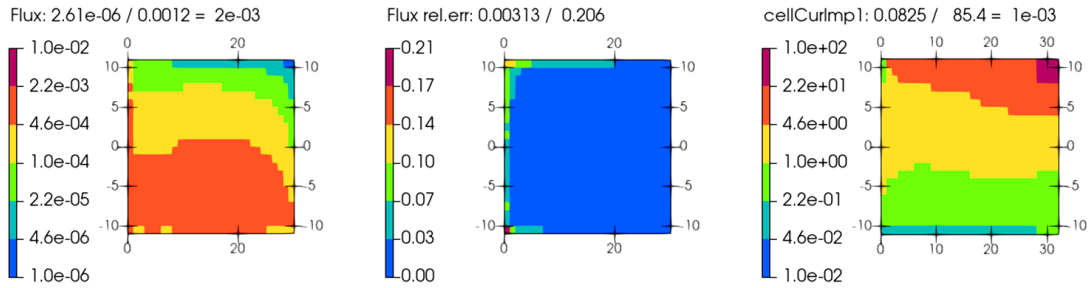



**Fig. 4.3** Flux (left), its rel.error (mid) and importance (right): Iteration 1

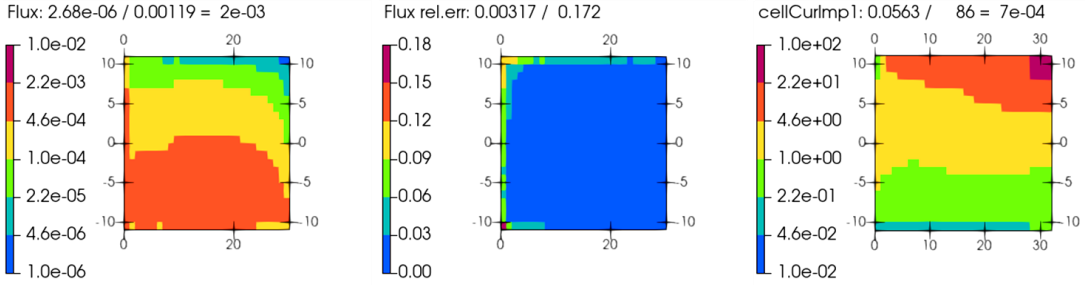



**Fig. 4.4** Flux (left), its rel.error (mid) and importance (right): Iteration 2

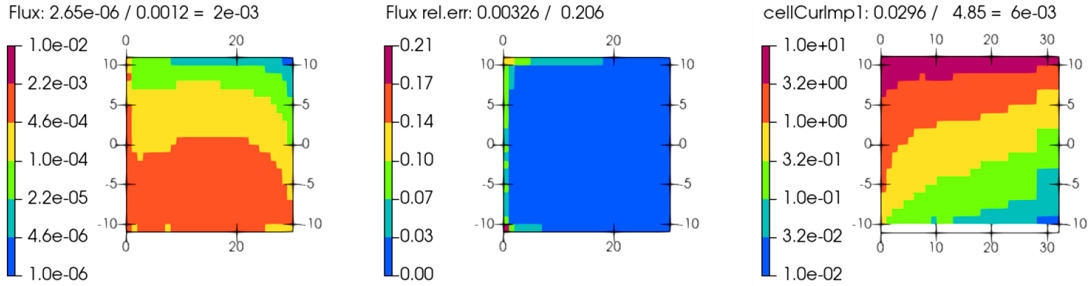



**Fig. 4.5** Flux (left), its rel.error (mid) and importance (right): Iteration 3

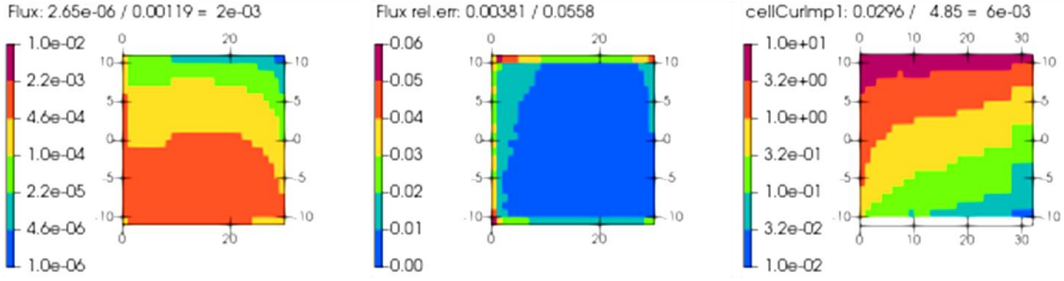



**Fig. 4.6** Flux (left), its rel.error (mid) and importance (right): Iteration 4

After iteration i, the file `<input>.wwd<i>` is written as output. This is a binary file containing WW-mesh parameters that can be used in another Serpent run. In the run above, the latest WW-mesh for global VR is found in a file with suffix `wwd2`, and the local VR WW-mesh is found in a file with suffix `wwd4`. To use a WW-mesh file in another Serpent run, the file should be referenced in the input deck by the `wwin` card, as shown in Listing 4.2.

**Listing 4.2** wwin card to use previous WW-mesh from external file

```
% Use previous WW-mesh
wwin wwGlobal wf "./wd.refModel.inp/refModel.inp.wwd4" 1
```

The Serpent runs with the WW-meshes from iterations 2 (global VR) and 4 (local VR) give flux and error distributions, which are shown in Fig. 4.7 and Fig. 4.8. They are similar to the distribution obtained during the respective iterations in run 2, compare with Fig. 4.5 and Fig. 4.6.

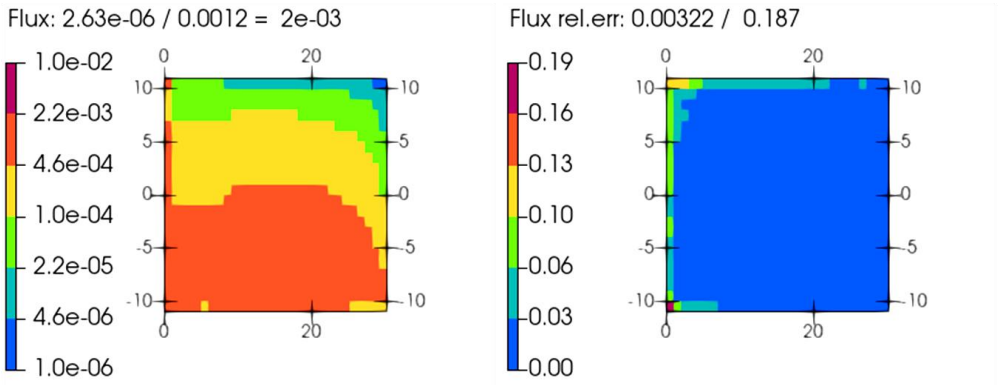



**Fig. 4.7** Flux spatial distribution and its error calculated with WW-mesh: Global VR WW-mesh applied

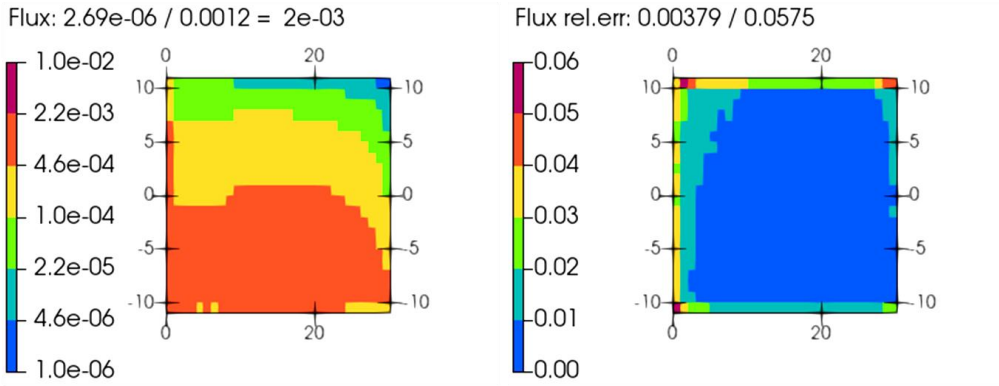



**Fig. 4.8** Flux spatial distribution and its error calculated with WW-mesh: Local VR WW-mesh applied

The cell detector values obtained in the runs without and with WW-meshes and their calculation times are compared in Tab. 4.1. One can see that the application of a WW-mesh increases calculation time but also improves the statistical error of the region above the model, i.e. far from the source. For the considered cell detector values, the local VR WW-mesh shows the best performance. It increased computation time by factor of 1.7, at the same time the statistical error is decreased by factor of 3 for the region above the uranium cylinder and by factor of 2 for the region above the water reflector. Without application of VR, such improvement could be achieved by increasing the number of sampled sources by factors of 9 and 4, respectively. Thus, in this example the application of WW-mesh brings a calculational speed-up at least by factor 2.3.

**Tab. 4.1** Time-integrated neutron flux below and above the model

| **WW-mesh** | **below uranium cylinder** | | **below water reflector** | | **above uranium cylinder** | | **above water reflector** | | **CPU time, in min** |
|---|---|---|---|---|---|---|---|---|---|
| | neutron flux | rel. error | neutron flux | rel. error | neutron flux | rel. error | neutron flux | rel. error | |
| no | 0.0943 | 0.009 | 1.21 | 0.003 | 0.00505 | 0.046 | 0.0321 | 0.018 | 4.45 |
| global | 0.0944 | 0.017 | 1.22 | 0.005 | 0.00518 | 0.028 | 0.0314 | 0.010 | 10.51 |
| local | 0.0926 | 0.009 | 1.22 | 0.004 | 0.00515 | 0.015 | 0.0315 | 0.007 | 7.46 |

### 4.1.1 Concluding notes

The flux intensity in the considered model differs by only three orders of magnitude (the min-to-max ratio is about $2 \times 10^{-3}$). Even calculation without VR gave sufficient track sampling throughout the whole geometry. For such systems VR is not generally needed, however the considered example helped to get familiar with the capabilities of Serpent to generate WW-meshes for global VR or VR in particular detectors, and to use them in subsequent runs.

Serpent also provides adaptive meshes that can be refined during iterations as needed. However, they are applicable only to rectangular mesh geometries.

In the `wi` list of the `wwin` card different WW-mesh generators can be referenced (as shown in Listing 4.1), but all of them must have the same mesh.

## 4.2 Highly shielded model

In the model above, the total flux density differs by only three orders of magnitude, respectively even the direct calculations without VR gave statistically relevant results over the whole geometry. In a next step, we consider now a model where the use of a WW-mesh is unavoidable.

The geometry of the model is the same as above, but instead of metal uranium, boron at density of 2.46 g/cc with a content of B-10 set to 10 % is used, and the region outside, originally willed with water, is voided. Also, the neutron source differs it is placed to the cylinder's axis, 5 cm above the mid-plane. In calculation without VR the tracks are sampled only in a small region around the source and the flux distribution in the other parts of the model geometry remains unknown.

A general approach for this kind of model is to consider models with reduced density. First, the model's density is reduced to a level that makes sampling in the whole geometry domain feasible. The global VR WW-mesh generated for the reduced-density model, although it is not optimal for the model with original density, might help to obtain enough sampling over the whole geometry also in a denser model. The iterations are repeated until the obtained WW-mesh can be applied to the model with original density.

To support this approach, Serpent has the global density factor (GDF) parameter in the `wwin` card. In the previous model it was always set to 1, see entries after each WW-mesh name in the `wwin` card in Listing 4.1.

### 4.2.1 Calculation 1

In the first attempt, the iterations start for density multiplied by $5 \times 10^{-2}$. The relevant parts of the input files and results on each iteration can be found in Appendix C, Listing C.1.

The general structure of the `wwin` card is similar to that used in the previous model, but there are some differences. The mesh size is reduced in the absorber region. Preliminary runs have shown that in a mesh with large elements, the sampled spatial region does not increase with iterations. The next difference is that GDF is applied (floating point values following the name of the WW-mesh generator). Hereby, in the first three iterations, which are done for the model, the densities for all material were multiplied by $5 \times 10^{-2}$. The next six iterations are done for material densities multiplied by $1 \times 10^{-1}$, etc.

The results obtained in each iteration, namely the importance map for the next iteration, the flux distribution and its errors are shown in Appendix C in Fig. C.1. The last successful calculation, iteration 11, is repeated in Fig. 4.9.

For iterations with a GDF of $5 \times 10^{-2}$ and $1 \times 10^{-1}$, all regions were sampled. However, at the first iteration with GDF set to $5 \times 10^{-1}$ (i.e. increased by factor of 5 with respect to the previous iteration), only a small region of the geometry was sampled. Furthermore, this region did not increase with subsequent iterations. The calculation was aborted since the next iteration's time increased by factor of 100. Although this calculation was aborted, some conclusions can be drawn:

- The flux density changes by almost 9 orders of magnitude already for a GDF-value of $1 \times 10^{-1}$ (i.e. when the density is reduced to 1/10 of the original density)
- To continue, either a smaller GDF step should be considered, or means to improve the sampling in the whole region should be implemented, like increasing the whole number of sampled sources in every iteration, and/or increasing the size of the mesh elements.

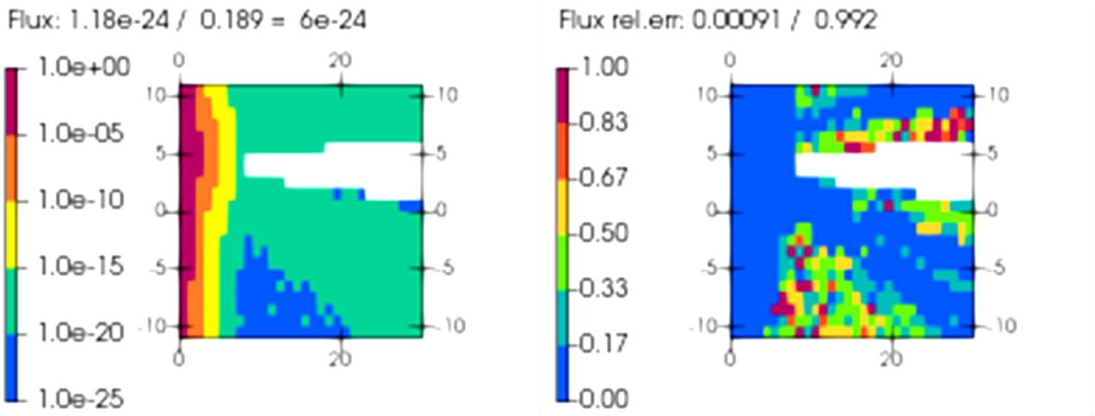



**Fig. 4.9** Calculation 1, iteration 11 (last successful), GDF=0.5

### 4.2.2 Calculation 2

This calculation uses the last successfully generated WW-mesh from the previous run with all regions sampled. The relevant cards are shown in Appendix C, Listing C.2.

In this input file, the mesh geometry description is removed from the `wwgen` card, since it is taken from the WW-mesh file specified in the `wwin` card. In the `wwin` card the GDF

values were changed: instead of jumping from $1 \times 10^{-1}$ to $5 \times 10^{-1}$, additional iterations are introduced for GDF = $2 \times 10^{-1}$.

Despite incomplete importance maps on some iterations, the final result for the model with original material densities was successfully sampled in the whole geometry and shows that the flux density changes by 33 orders of magnitude, see Fig. 4.10. For intermediate steps see Appendix C, Fig. C.2.

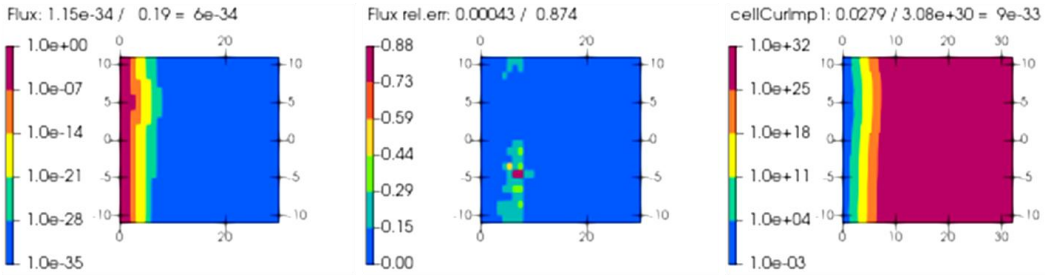



**Fig. 4.10** Calculation 2, iteration 17 (last successful), GDF=1.0

Note: after the importance map and the flux is sampled in the whole model, the subsequent iterations with the same material density do not show improvements in the statistical error.

### 4.2.3 Calculation 3

In a next step, a similar calculation was performed with the same WW-mesh geometry and the same iterations, but as a single run, not using external WW-mesh. It showed similar distributions of the importance, flux and its errors, except in the last iteration, where Serpent stopped with the error “insufficient neutron buffer size”. The last row in Fig. C.1 shows the intermediate results at this iteration. The input file and intermediate results are shown in Appendix C.

### 4.2.4 Calculation 4

The calculation 4 differs from the previous one by two additional radial mesh elements outside of the absorber. Also, two iterations in the beginning are removed: instead of 4 iterations with GDF $5 \times 10^{-2}$, only two iterations are performed. With these modifications, the last iteration show that the whole geometry was sampled, see Fig. 4.11. For intermediate steps see Appendix C, Fig. C.4.

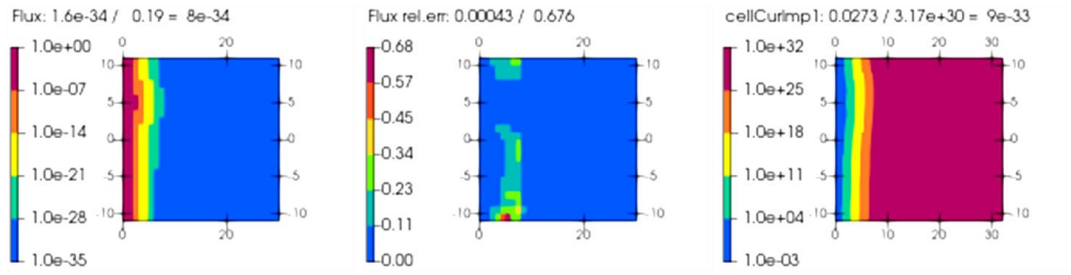



**Fig. 4.11** Calculation 4, iteration 21 (last successful), GDF=1.0

### 4.2.5 Calculation 5

In this calculation, the radial mesh grid is made more coarse in the central channel: some of the 0.2 cm steps, are merged to 0.4 cm steps. This change improved the total calculation time inconsiderably and still results in a completely sampled geometry at the final iteration, as shown in Fig. 4.12. For intermediate steps see Appendix C, Fig. C.5.

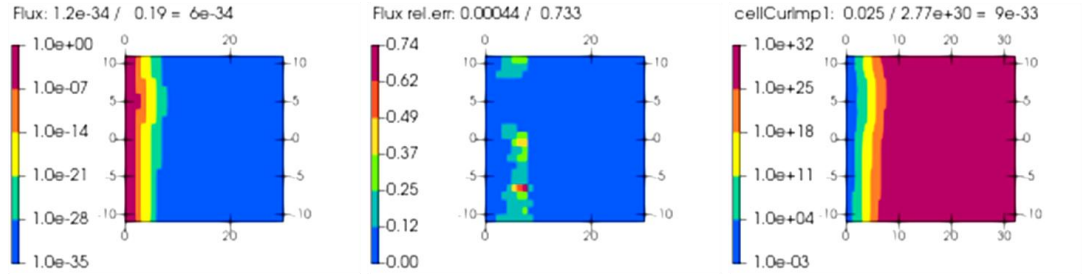



**Fig. 4.12** Calculation 5, iteration 21 (last successful), GDF=1.0

### 4.2.6 Calculation 6

In calculation 6 a coarser radial mesh inside the channel was used, with more radial mesh elements in the absorber. With this WW-mesh geometry tracks did not propagate to the whole geometry. All steps are shown in Appendix C, Fig. C.6.

### 4.2.7 Conclusions

From these calculations the following conclusions can be drawn:

- When global density factors are used, small steps should be considered. They should not be increased by a factor of two.
- If tracks do not propagate to the whole geometry for a given GDF value, iterations with lower GDF should be inserted instead of more iterations with the same GDF.

- The size of adjacent mesh elements should be kept not too different.

If a preliminary estimate of the flux distribution is known, it can be used to define the mesh grid structure: the flux decrease inside mesh element should be less than 1 to 1.5 orders of magnitude. This can be seen in calculation 5, in which 16 mesh grid elements cover the absorber region. In the absorber region, the flux intensity decreases by about 30 orders of magnitude. In calculation 6 the number of mesh grid elements is reduced to 8 and the whole sampled geometry was not reached.

# 5 Neutron and photon leakage benchmark

## 5.1 Calculation of the benchmark

Measures of neutron and photon spectra from a Cf-252 radionuclide source shielded with an iron sphere of different radii were performed in the 1980's. They are described as the benchmark ALARM-CF-FE-SHIELD-001 in Volume VIII of the International Handbook of Evaluated Criticality Safety Benchmark Experiments /NEA 24/, /NEA 25a/. Previously, this benchmark was already used in the GRS to evaluate MCNP predictions /SOM 19/. In this part of the work, the benchmark calculations are repeated with Serpent and compared with benchmark data as well as with previously calculated MCNP results.

The application of Serpent for this benchmark is already presented in /HÄK 22/. In the present report we compare the Serpent and MCNP results in more detail and consider how different aspects of modelling (e.g. source particle spectrum, physics options) affect the Serpent results.

In this series of experiments, a Cf-252 source with known intensity was placed in iron speres of different radii. Measurements of the neutron and photon spectra escaping the sphere's outer surface were performed with different techniques, allowing the measurement of the energy in a range from 0.005 MeV to 17 MeV for neutrons and from 0.407 MeV to 10.3 MeV for photons. The spheres with radius 20, 30, 40, 50, 60 and 70 cm were used. In addition, the measurements were performed for the bare source, i.e. with no shielding.

The Cf-252 nuclide undergoes alpha decay and spontaneous fission and thus emits both neutrons and photons. In the experiment, the total photon spectra induced by both source neutrons and photons is measured, respectively the benchmark contains the total photon spectra results. In the calculations, a distinction is made between the secondary photons leaked from the sphere surface, i.e. caused by the interaction of neutrons with matter, and the direct photons, i.e. caused by the transport of source photons. For this reason, only the neutron spectra calculated with Serpent are compared both with the experimental data and with previous MCNP results, see section 5.6. The calculated secondary and direct photon spectra are compared against the respective MCNP calculations only. Comparison to the MCNP photon spectra with the benchmark are reported in /SOM 19/.

Providing Serpent input decks for the ICSBEP benchmarks is an ongoing current international work /HOL 24/. In this database also the Serpent input files for ALARM-CF-FE-SHIELD-001 can be found /NEA 25b/. In the present work however, the Serpent input decks were manually created based on the MCNP input decks used in /SOM 19/, only the source spectrum definition was partly adopted from /NEA 25b/.

The geometry and material composition conversion are straightforward and exact, since both codes use the same method of geometry description and can represent the geometry as specified in the benchmark. Differences are unavoidable in the description of the neutron source spectrum. This topic is described in section 5.2.

For neutron spectra calculations, several different detectors are defined, following the MCNP model. For comparison with experimental results, the Serpent detector description repeats the one of MCNP, which in turn considers parameters of detectors used in the experimental measurements. The MCNP tally modifiers shown in Listing 5.1 is converted to the Serpent detector description, see Listing 5.2.

**Listing 5.1** Tally modifiers for pulse height gaussian energy broadening in MCNP

```
f2:n 7
f12:n 7
ft2 geb 0 0.002 10000
ft12 geb 0 0.125 0
```

**Listing 5.2** Detector modifiers for pulse height gaussian energy broadening in Serpent

```
det   2 n   dphb phb2
det  12 n   dphb phb12
phb phb2   4 0 0.002 10000
phb phb12  4 0 0.125 0
```

For the calculated spectra, which are not foreseen to be compared with experimental data, e.g. comparison with MCNP in section 5.2 or investigation of the effect of calculational parameters (nuclear data library in section 5.4 or transport modelling options in section 5.5) a separate set of detectors was defined in Serpent, with finer energy structure and without adjustment corresponding to physical detector parameters.

In the following, configurations with different sphere radii are referred to as “Case N”, where N is the index from 1 to 7 shown in Tab. 5.1.

**Tab. 5.1** Description of configuration cases

| Case index | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| sphere diameter in cm | bare source | 20 | 30 | 40 | 50 | 60 | 70 |

## 5.2 Watt source neutron spectrum in Serpent

In MCNP, the source spectrum can be specified either as a set of pairs ($E_i$, $P_i$) that are then interpolated by MCNP, or as a functional dependency number correspondent to one of the predefined distribution functions, followed by the parameters for the chosen function. In Serpent, only the first option is available, while the predefined Watt fission probability distribution equation (5.1) is used in the MCNP calculations in /SOM 19/.

$$p(E) = c\, e^{E/a}\, sinh\sqrt{bE} \tag{5.1}$$

The Parameter $a$ and $b$ can be set by the user. Two sets of parameters for the Watt fission spectrum can be found in the MCNP input decks: the first set is used in the MCNP calculations, the second is commented out, see Tab. 5.2.

**Tab. 5.2** Parameters of the Watt fission spectrum

| Set | $a$ | $b$ |
|---|---|---|
| 1 | 1.175 | 1.04 |
| 2 | 1.025 | 2.926 |

In the serpent input file /NEA 25b/, the neutron source spectrum is defined by linear interpolation of the probabilities given at 281 energy points. The results obtained with this source showed considerable deviation from the MCNP results and this difference might be attributed to the differences in the source spectrum. As one can see in Fig. 5.1, the probability points taken from /NEA 25b/ differ from the probabilities computed by equation (5.1). The deviation from the Watt fission spectrum with the first set of parameters lies within 1 %, except for energies above 10 MeV, where the source spectrum defined in /NEA 25b/ is almost two times higher as calculated with equation (5.1). Comparison with the Watt fission spectrum calculated with the second set of parameters is even worse and, since this set was not actually used in MCNP calculations, it will be not considered further.

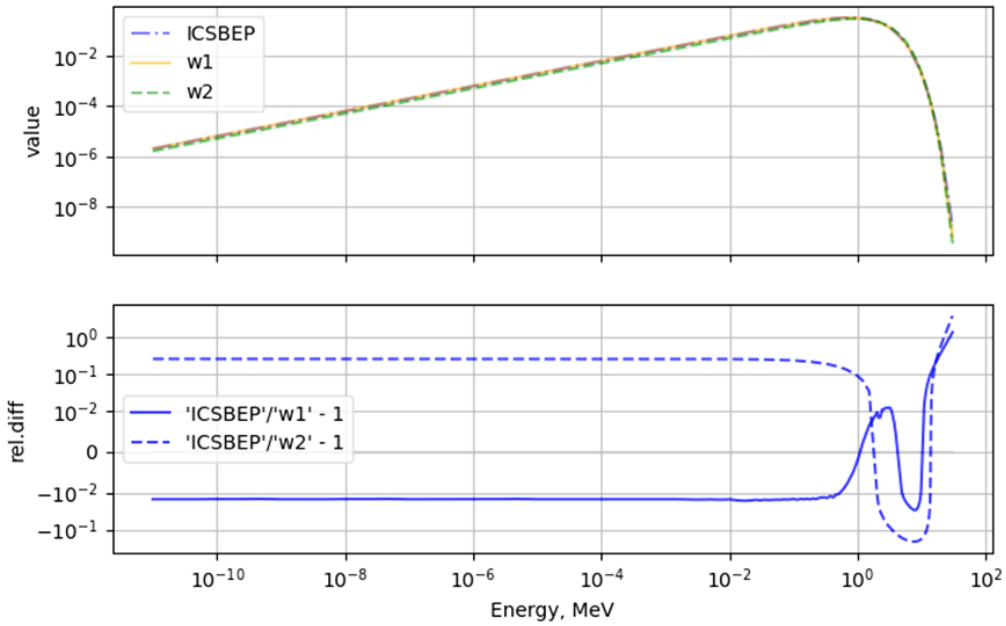



**Fig. 5.1** Neutron source spectrum comparison ICSBEP and MCNP input parameter

"ICSBEP"-values taken from Serpent input /NEA 25b/, "w1" and "w2"-values calculated with the Watt fission spectrum function with the parameter sets given in Tab. 5.2.

Even if the probability points for the source spectrum in Serpent are calculated by equation (5.1), the linear-linear interpolation between the points used by Serpent to define probabilities between the energy points, leads to considerable relative differences, as shown in Fig. 5.2. For energies above 1 MeV, the energy grid with 281 points used by /NEA 25b/ to define the source spectrum has too few points to represent the non-linear behaviour adequately. This can be improved by considering a finer energy grid. The energy grid was generated from the requirement that the linear-linear interpolation should not result in more than 0.1 % absolute and relative deviation. The grid contains 517 energy points and provides the required precision in the whole energy range, see Fig. 5.3.

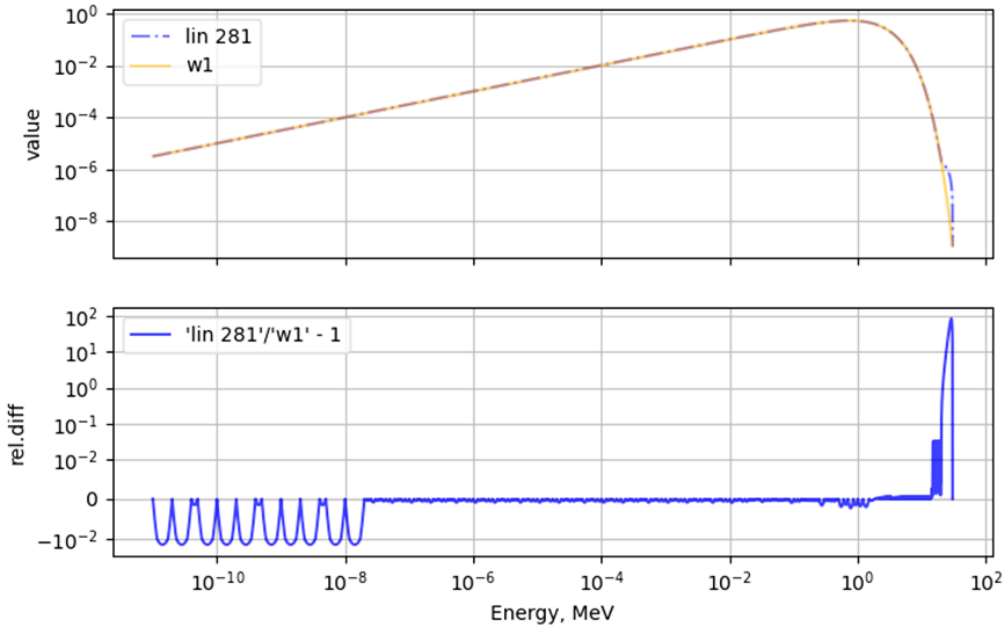



**Fig. 5.2** Neutron source spectrum comparison: Interpolation with 281 points vs. MCNP input parameter w1

"lin 281": lin-lin interpolation on energy grid with 281 points, "w1": Watt fission spectrum.

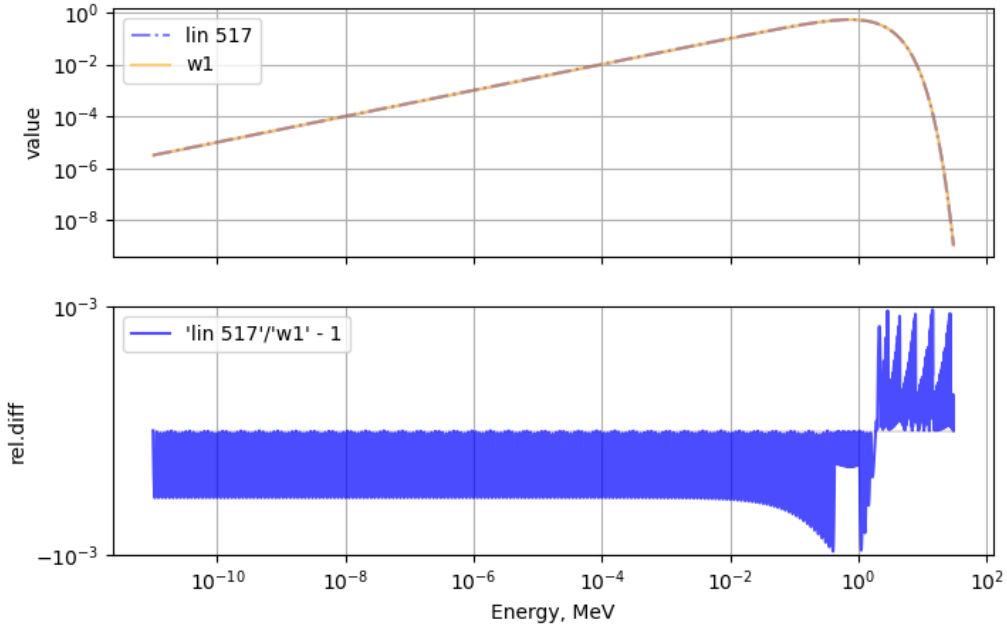



**Fig. 5.3** Neutron source spectrum comparison: Interpolation with 517 points vs. MCNP input parameter w1

"lin 517": lin-lin interpolation on energy grid with 517 points, "w1": Watt fission spectrum.

The impact of the energy grid used to define the neutron source spectrum in the Serpent calculations is shown for all benchmark cases in Fig. 5.4 to Fig. 5.10. The Figures show the calculated neutron (left) and photon (right) spectra for the neutron source spectrum defined by 517 and 281 energy points. The lower plots show the difference between the calculated spectra. The shaded areas in the lower plots show the relative calculation error. For all shielding thicknesses, there is clear difference exceeding the statistical error in the neutron tally energy bin close to 20 MeV, which can be attributed only to the differences in the neutron source spectrum descriptions. The impact on the calculated photon flux spectrum is less pronounced: although one can see for all cases except case 4 that the source with 281 energy points result in a higher photon flux above 10 – 15 MeV, the differences are comparable to the statistical error.

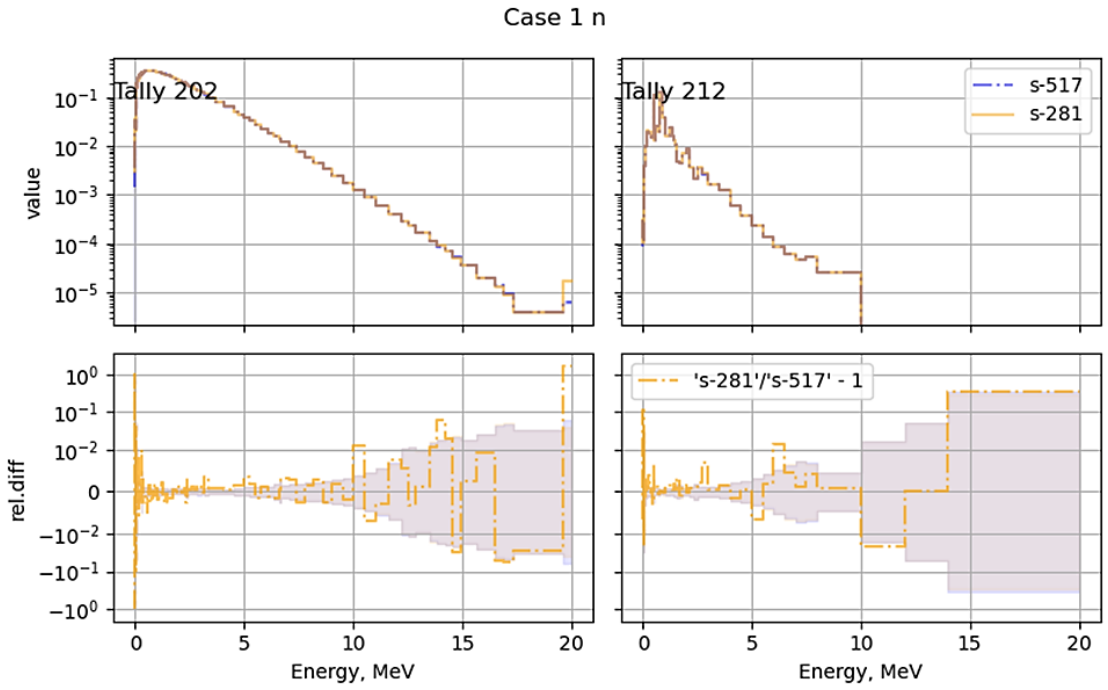



**Fig. 5.4** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 1 (bare source)

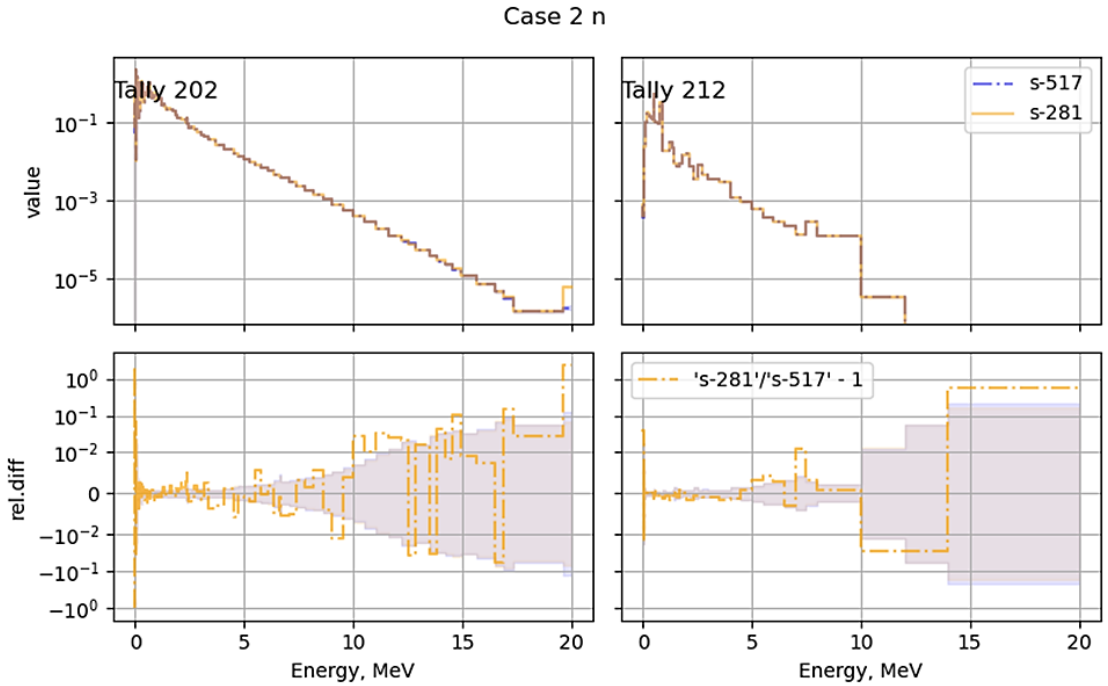



**Fig. 5.5** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 2

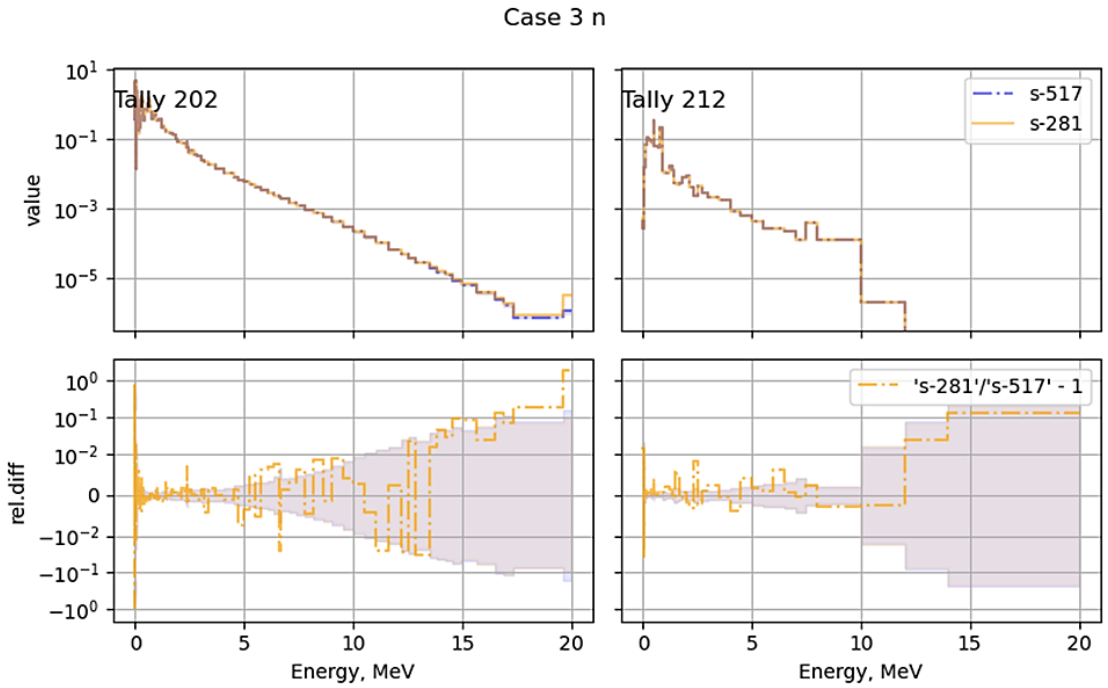



**Fig. 5.6** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 3

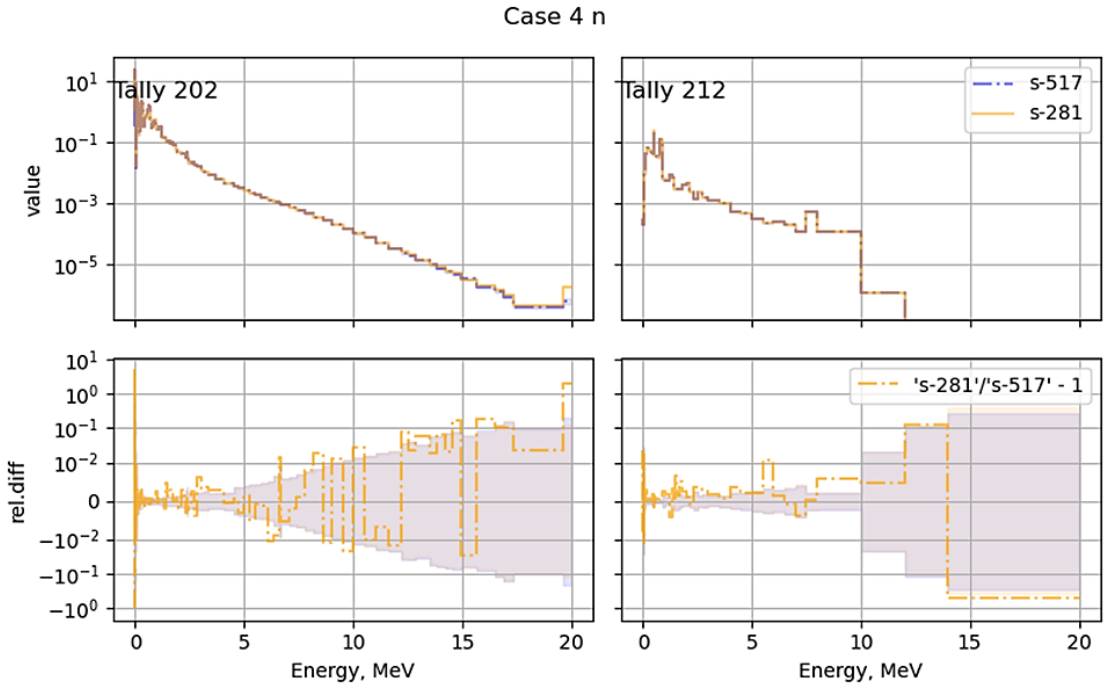



**Fig. 5.7** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 4

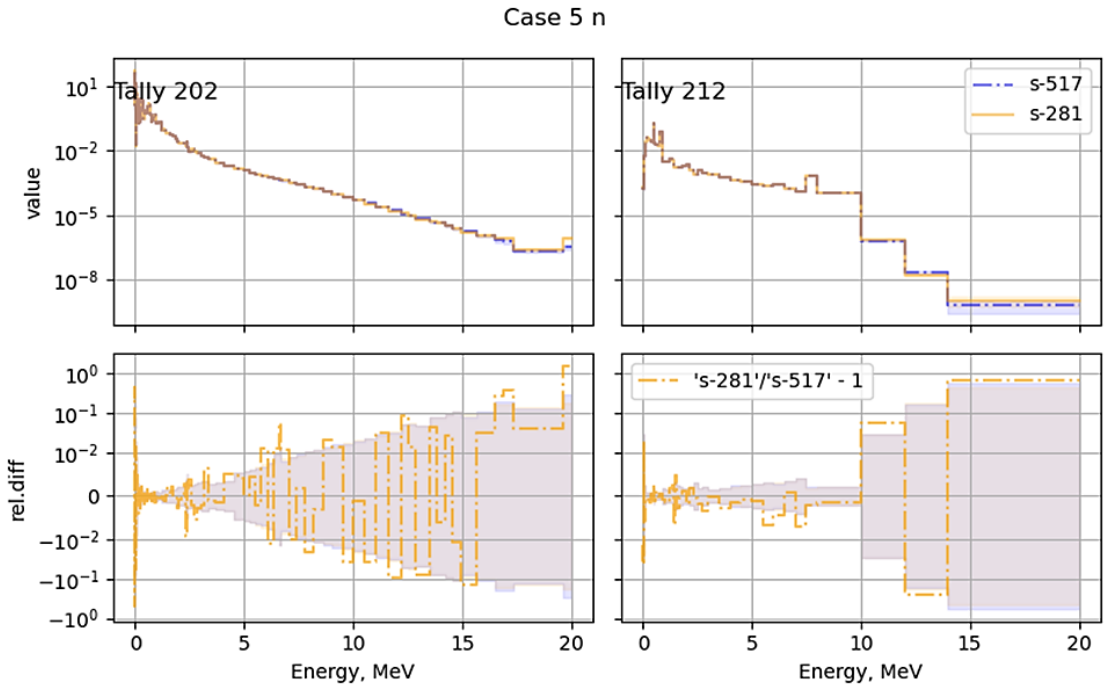



**Fig. 5.8** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 5

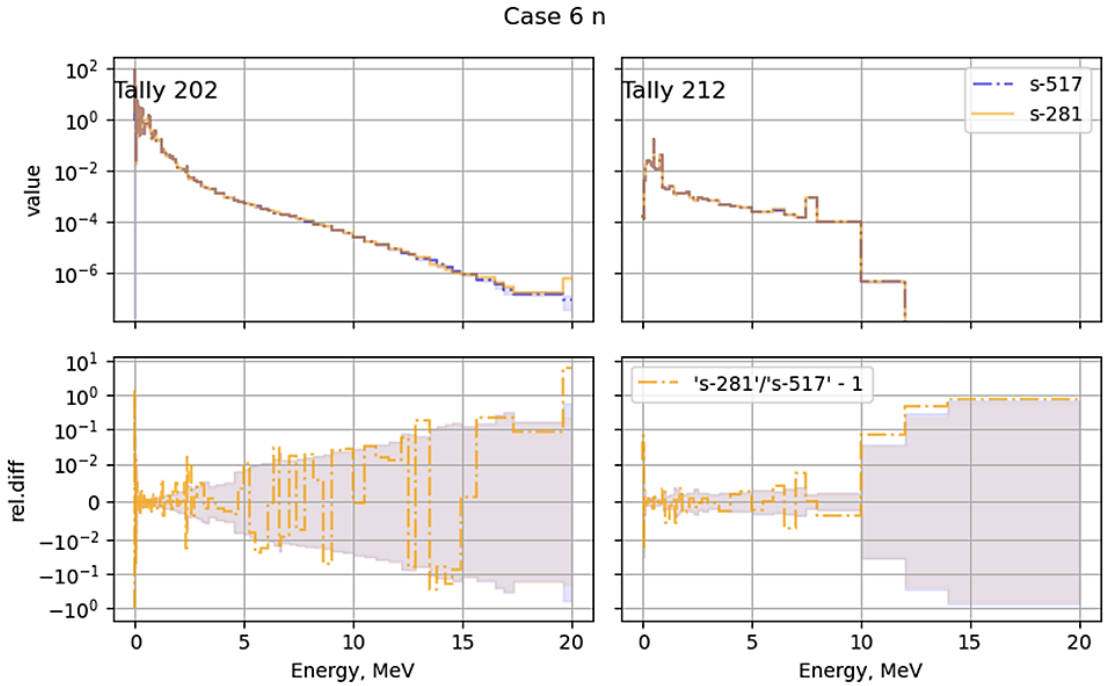



**Fig. 5.9** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 6

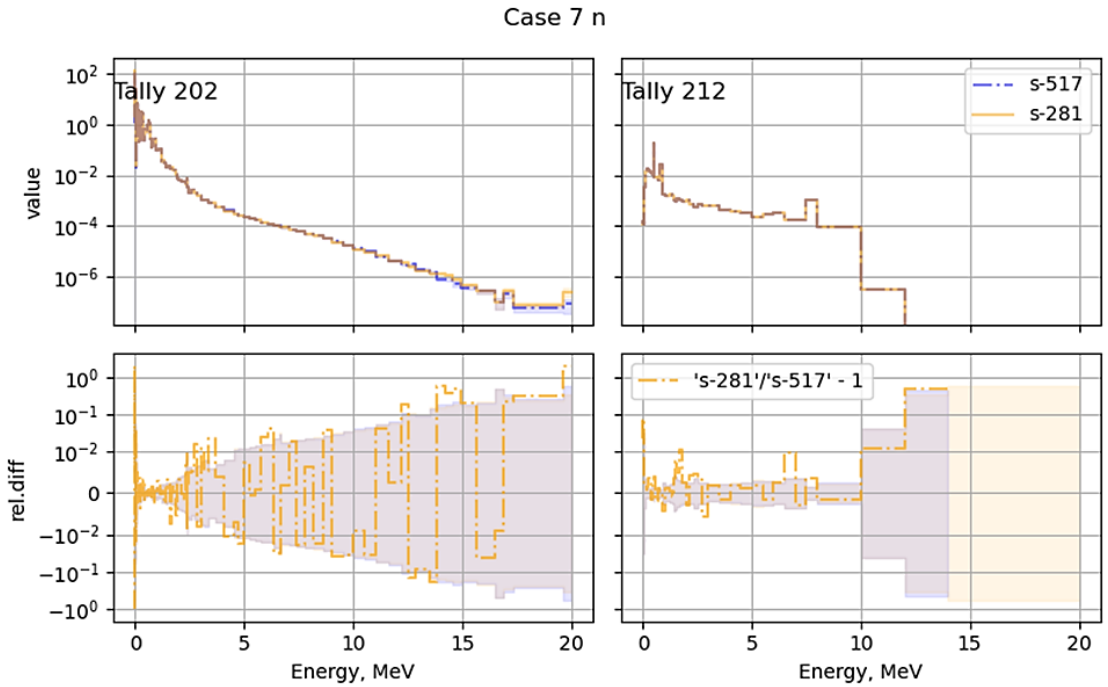



**Fig. 5.10** Neutron (left) and photon (right) spectra comparison for 517 and 281 energy points for case 7

### 5.3 Histogram source spectrum in Serpent

In the MCNP calculations for the direct photon source, the photon source spectrum is given as a histogram function of energy, using the `si` and `sp` cards. This section discusses how this type of source can be represented in the Serpent input file.

Spectrum of the source particles can be specified in Serpent in the `src` card using the `sb` option. Its description in /VTT 25a/ says that when the "histogram" interpolation is chosen, the values given after the `sb` keyword are the weight of the energy bin.

The term weight of the energy bin can be understood in different ways:

1. It is the value proportional to the sampling probability from the energy bin. I. e. for the i-th energy bin, the probability to sample from it is equal to $w_i = \frac{W_i}{\sum W_j}$, where $W_i$ are values specified by the user. Let's consider a small energy range `dE` within the i-th energy bin. The probability to sample from `dE` is equal to the probability to choose the i-th bin, $w_i$, multiplied with the ratio $\frac{dE}{\Delta E_i}$, where $\Delta E_i$ is the bin's width. If this interpretation is correct, the detected flux per unit energy will be proportional to $\frac{w_i}{\Delta E_i}$.

2. It is the weight applied to the source particle after sampling the energy. Let's assume that the energy is sampled uniformly, i. e. the probability to sample within a small energy range $dE$ is $\frac{dE}{\sum \Delta E_j}$. The particles weight is set to $W_i$, therefore its contribution to the detector per unit energy is proportional to $W_i$. The proportionality factor depends on the normalization set elsewhere (e.g. `set srcrate`). For the normalization used in the model (`set srcrate 1` means that the mean source weight is 1), the flux per unit energy will be equal to $\frac{W_i}{\sum W_j \Delta E_j}$.

To check, which of the two interpretations is correct, a Serpent test model is considered. The test Serpent model describes a point isotropic photon source, whose energy spectrum is given with the `sb` option. Photons propagate in a voided region and are tallied with fine energy resolution on a spherical surface centered at the source position.

**Listing 5.3** Serpent input file to test source histogram spectrum definition

```
surf 1 sph    0 0 0   1
surf 2 sph    0 0 0  10
surf 3 sph 9.9 0 0  1e-6  % just to add material to the model

cell 1 0 void 3 -2
cell 2 0 m1     -3
cell 3 0 outside 2

mat m1 -1e-10 tmp=300 1001.03c 1

set nps 10000000 5
src s1 p
        sp 0 0 0
        sw 1.0
        sb 8 1
        1    0
        2    1
        3    2
        3.5  2
        4    1
        6    1
       10    4
       11    0

set srcrate 1.

det 12 p dv 1.0  ds 1 -2 de pe12
det 22 p dv 1.0  ds 1 -2 de pe22
det 32 p dv 1.0  ds 1 -2 de pe32
ene pe12  1   1 2 3 3.5 4 6 10    % arbitrary grid
ene pe22  2  200 1e-3 10          % equal energy width
ene pe32  3  200 1e-3 10          % log-equal energy width
```

The corresponding input file is shown in Listing 5.3. The shaded lines 16 to 23 contain input parameters for `sb` option. There are 8 pairs that represent a histogram (`1` after `8`): the upper energy bin boundary $E_i$ and the correspondent bin weight $W_i$.

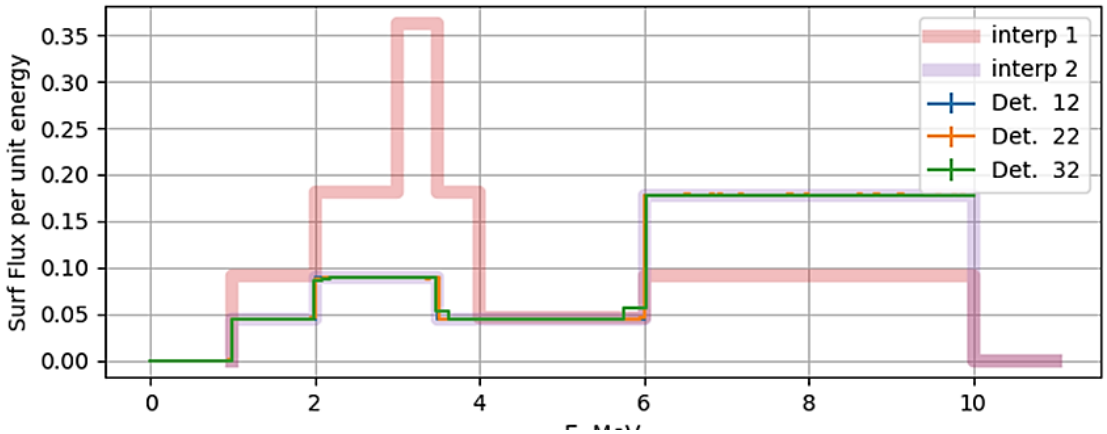



**Fig. 5.11** Flux spectrum calculated in test model Listing 5.3 and predictions based on the two possible interpretations

The calculated results, together with predictions based on both interpretations, are shown on Fig. 5.11. As one can see, the second interpretation is the correct one, i.e. the values in `sb` option are proportional to the probability density.

The source photon spectrum in MCNP is given with the `SI` and `SP` cards, which contain the energy bin boundaries and bin probabilities. Therefore, to convert the spectrum from MCNP representation to Serpent, the values from `SP` card (probability) must be divided by the bin width to get the probability density.

## 5.4 Choice of the nuclear data library

VTT provides data library sets based on different evaluations, see section "*4.1.1 Serpent data libraries*" in /VTT 25b/. The interaction data based on ENDF/B-VII.1 and JEFF-3.2 data libraries were applied for the Serpent calculations of the benchmark, both for the neutron and photon sources. The results for the neutron (left) and secondary photon (right) spectra are shown in Fig. 5.12 to Fig. 5.18. The lower plots show the difference between the results obtained with the ENDF/B-VII.1 and the JEFF-3.2 nuclear data library. The shaded areas in the lower plots show the relative calculation error.

All cases except the case without shielding show similar behaviour:

- Neutron flux spectra:
  - The results of the two libraries are within the statistical error for energies above 1 MeV.
  - Oscillating deviations of ± 10 % and above are present in the energy range from 0.1 MeV up to several MeV. The lower boundary of the energy range exhibiting this behaviour shifts to 0.01 MeV for cases with thicker shield, see Fig. 5.17 and Fig. 5.18.
  - JEFF-3.2 data predicts considerably lower neutron flux for energies below 0.01 MeV.
- Secondary photon flux spectra:
  - JEFF-3.2 predicts lower intensity in the whole energy range, except several peaks (cases 2 to 4) or one single peak (cases 5 to 7) within 1 – 3 MeV, and in the energy bin at 0.01 MeV.

- Considerable difference for energies above 3 MeV of up to one order of magnitude.
- Less pronounced difference for energies between 0.02 MeV and 1 MeV but still exceeding the statistical error.

The direct photon spectra calculated with ENDF/B-VII.1 and JEFF-3.2 data libraries do not show any statistically relevant differences.

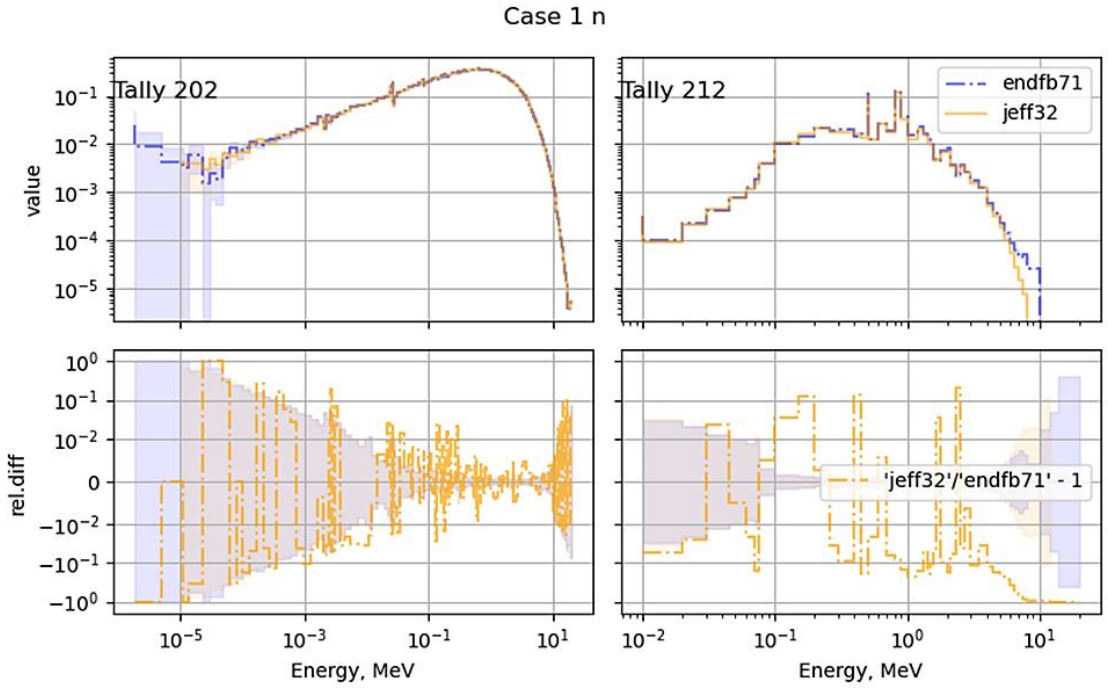



**Fig. 5.12** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 1 (bare source)

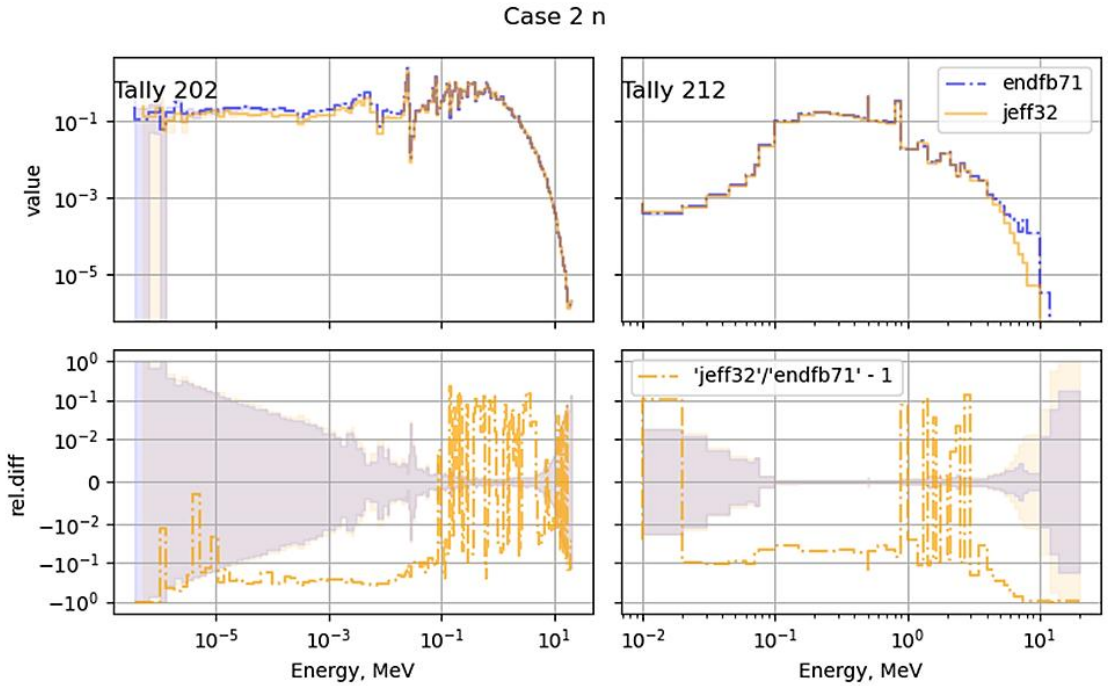



**Fig. 5.13** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 2

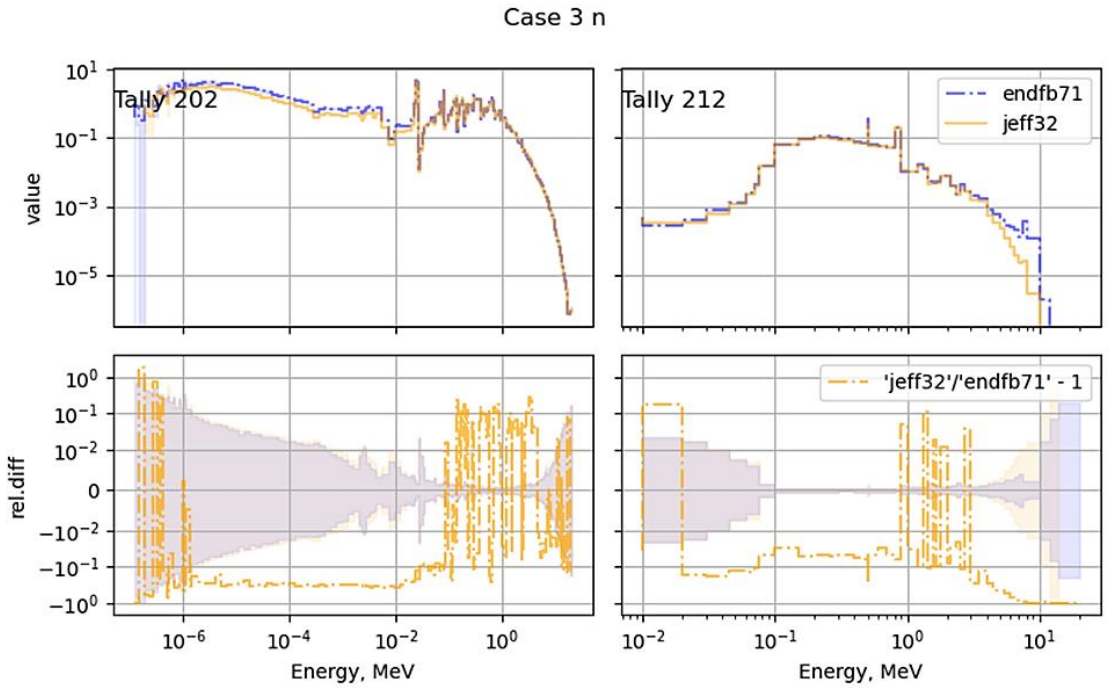



**Fig. 5.14** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 3

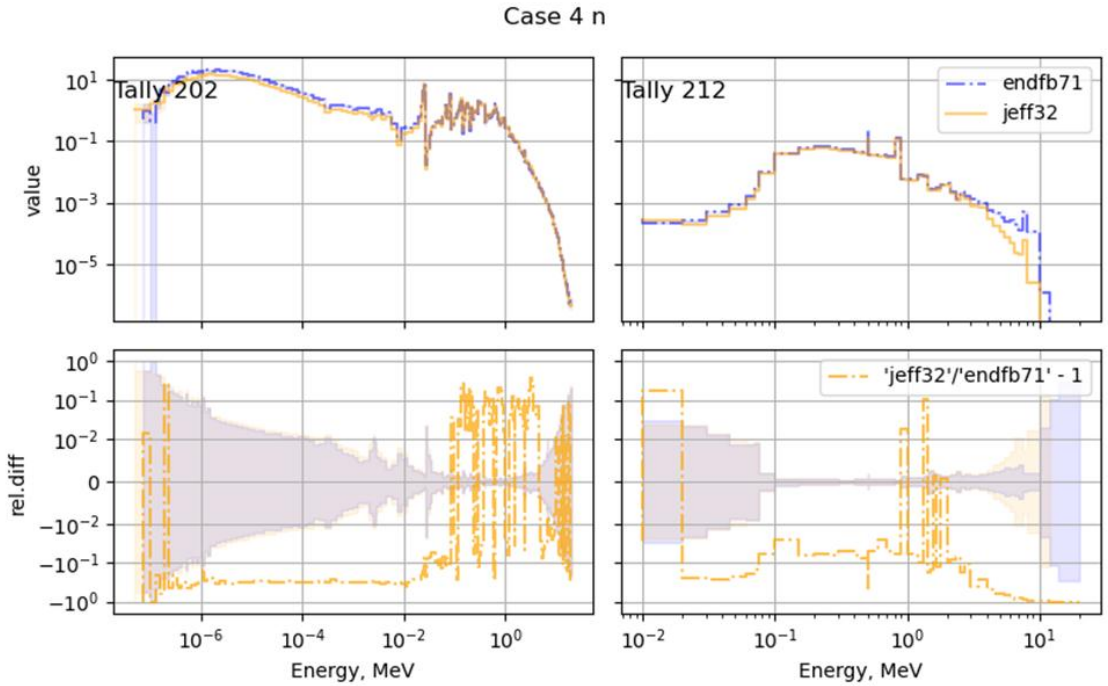



**Fig. 5.15** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 4

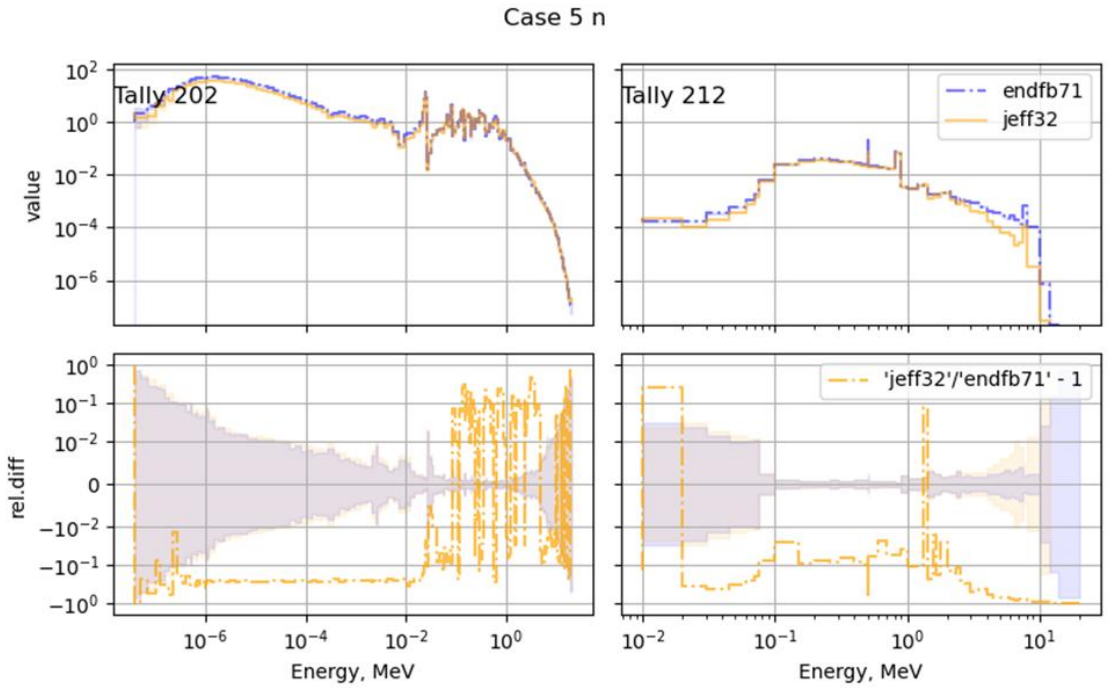



**Fig. 5.16** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 5

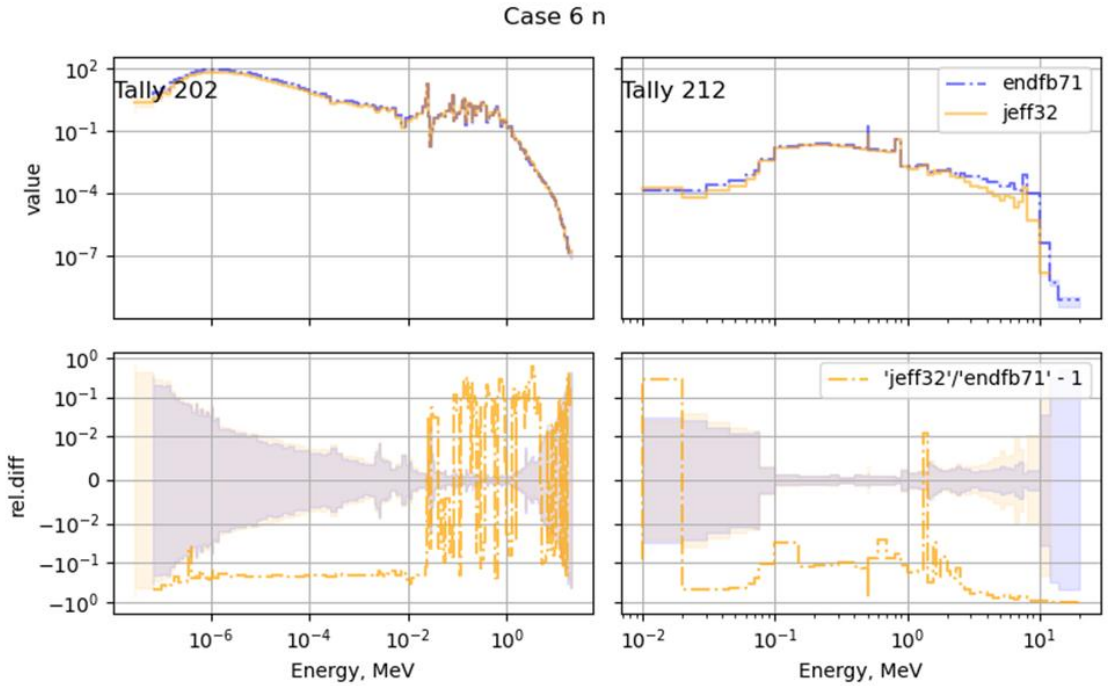



**Fig. 5.17** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 6

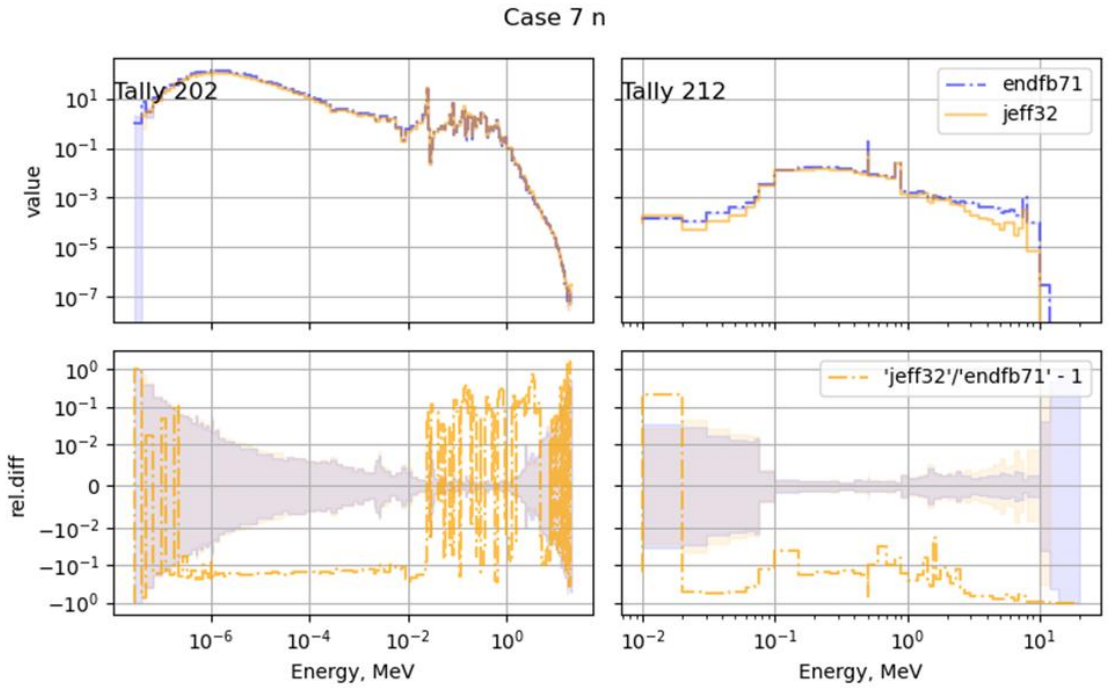



**Fig. 5.18** Neutron (left) and secondary photon (right) spectra comparison for ENDF/B-VII.1 and jeff-3.2 for case 7

## 5.5 Use of the implicit treatment for capture and n, xn reactions and the unresolved resonance probability table sampling options

The `set impl` card is used to turn on and off the implicit treatment for capture and n, xn reactions, and for the number of fission neutrons. By default, the implicit treatment is not active. The `set ures` card controls the unresolved resonance probability table sampling, which is also not active by default.

The impact of these two options on the neutron transport modelling is shown in Fig. 5.19 to Fig. 5.25. They show the neutron (left) and secondary photon (right) spectra, calculated by Serpent with different physics treatment options. Line “ref”: both unresolved resonances and implicit treatment are switched off. Line “ures”: unresolved resonances probability sampling is switched on. Line “impl”: implicit treatment for scattering and n, xn reactions is switched on. The lower plots show the deviations of the “ures” and “impl” results from the reference. The shaded areas in the lower plots show the relative calculation error.

Both options have no considerable impact on the neutron flux spectrum. The secondary photon spectrum shows some dependence on the implicit treatment: it results in underestimation of the photon flux for energies above 1 MeV. This effect grows with the shielding thickness. For cases 5 to 7, the implicit treatment leads to lower photon flux also for lower energies.

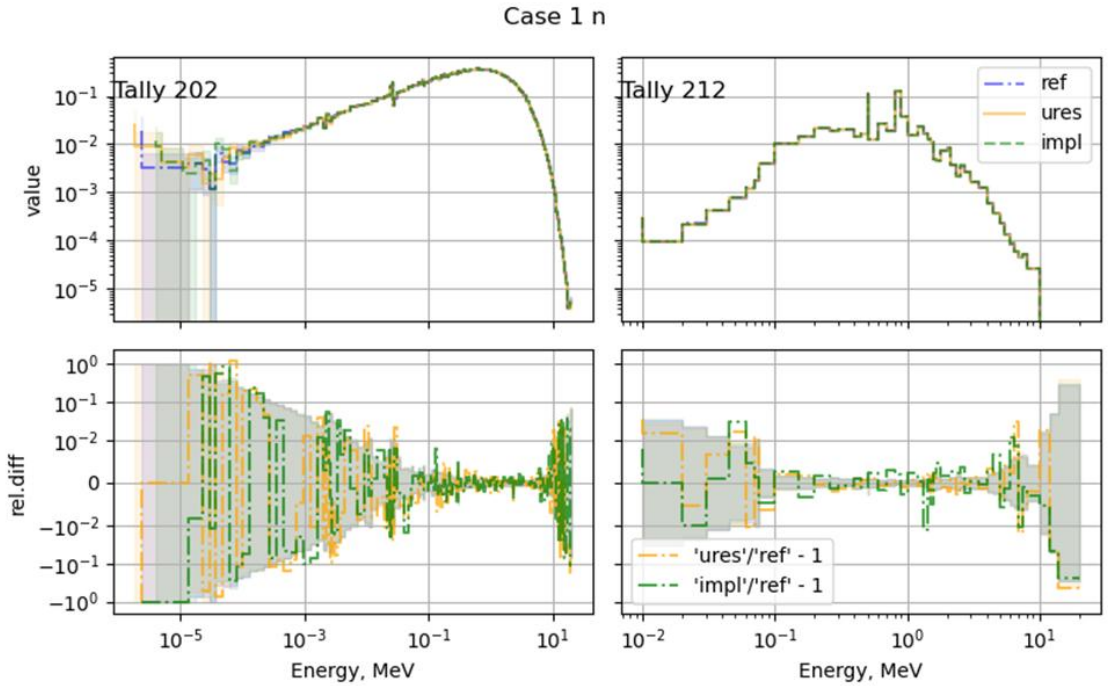



**Fig. 5.19** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 1 (bare source)

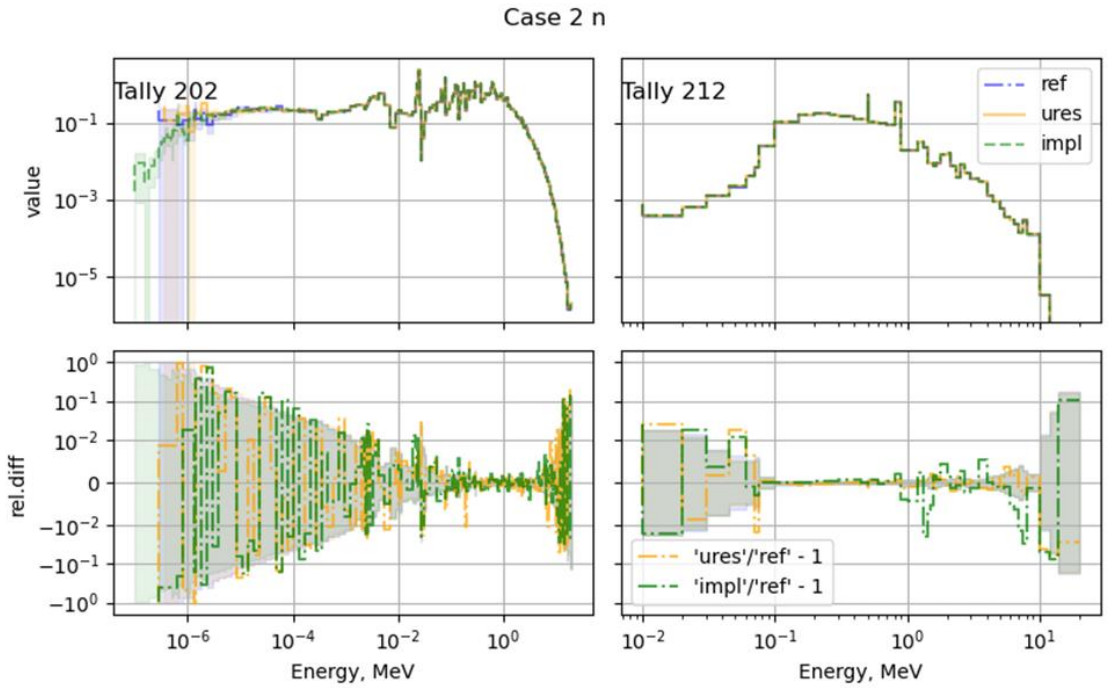



**Fig. 5.20** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 2

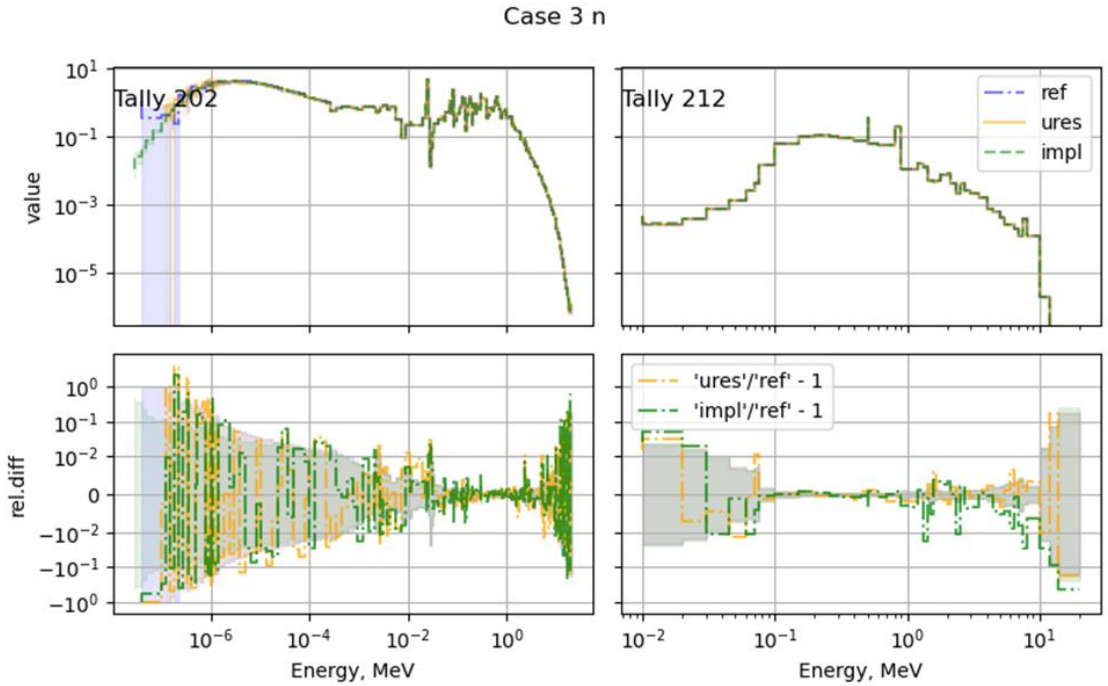



**Fig. 5.21** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 3

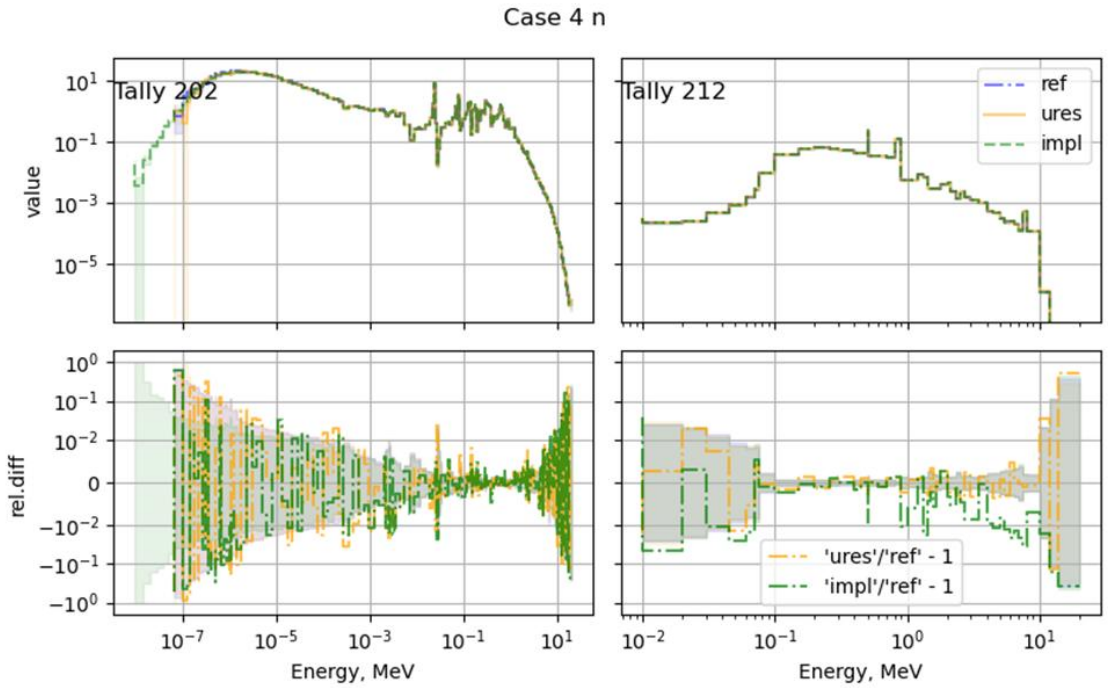



**Fig. 5.22** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 4

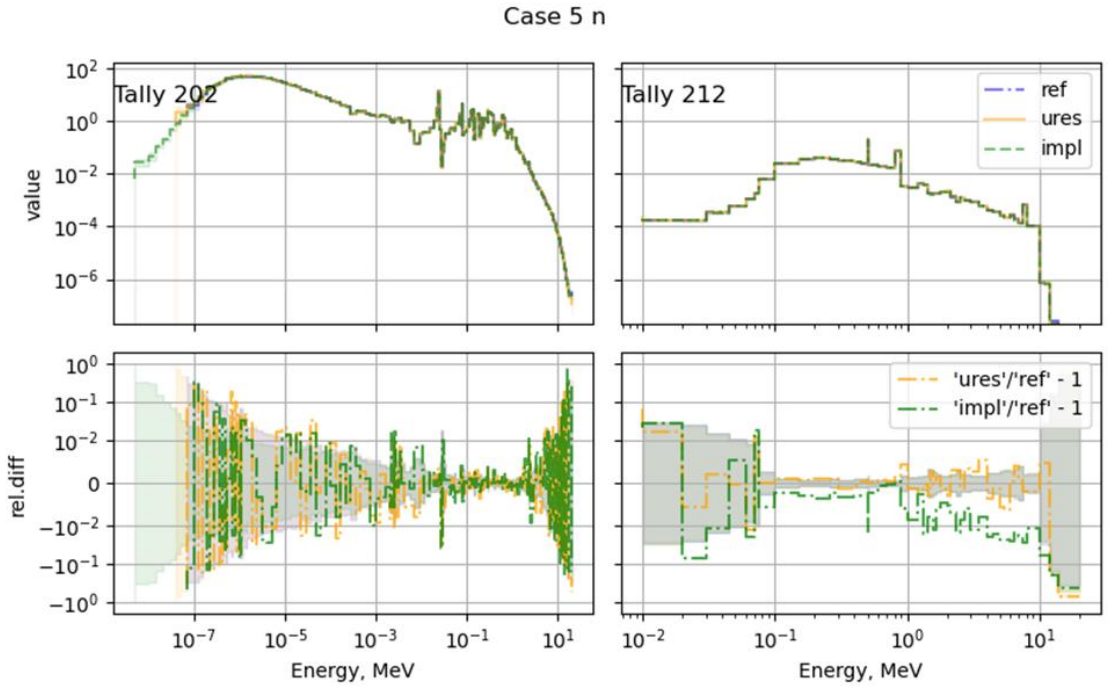



**Fig. 5.23** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 5

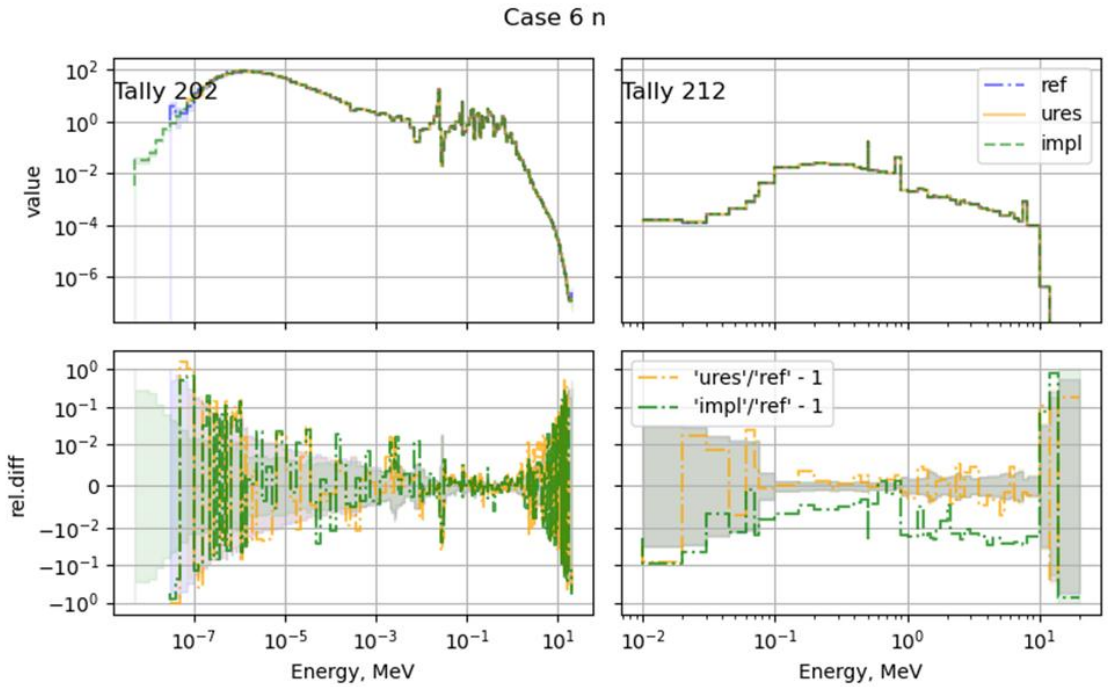



**Fig. 5.24** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 6

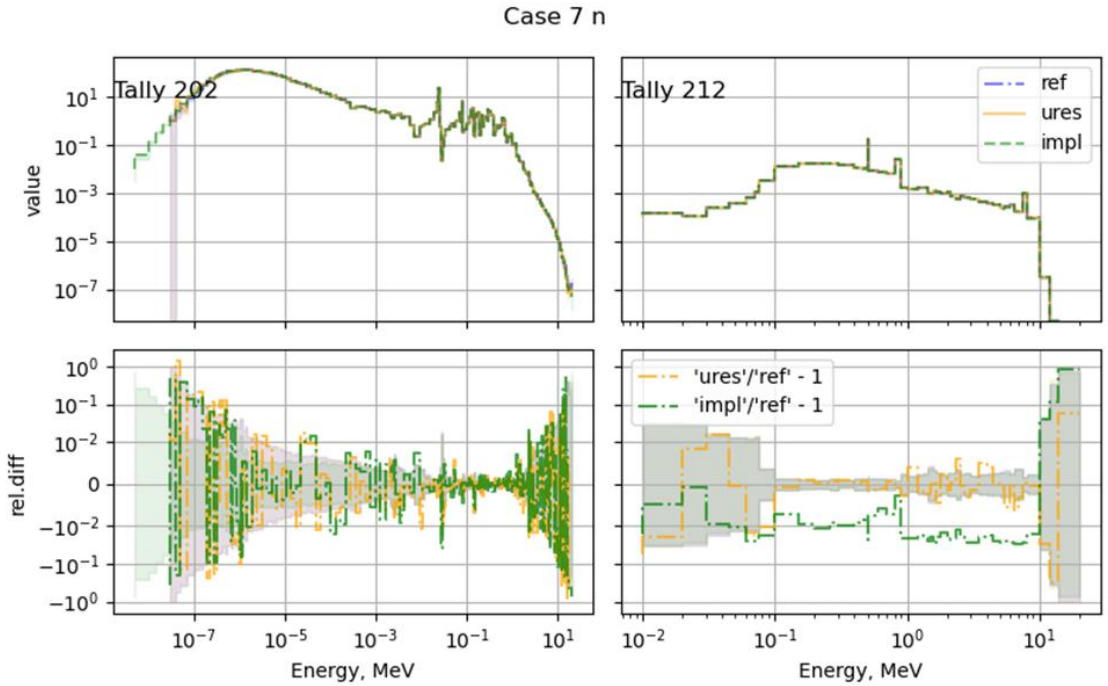



**Fig. 5.25** Neutron (left) and secondary photon (right) spectra with different physics treatment options: Case 7

## 5.6 Comparison with benchmark neutron spectra

The neutron flux spectrum calculated with Serpent and MCNP with additional tally treatment shown in Listing 5.1and Listing 5.2 is compared to the experimental benchmark data in Fig. 5.26 to Fig. 5.32. The figures show the neutron spectra, calculated by MCNP and Serpent, compared with experimental benchmark data. The lower plots show the difference between the two calculations and the experimental data. The shaded areas in the lower plots show the relative experimental and calculation error. One can see that the predictions of MCNP and Serpent both coincide with experimental data within the experimental error. The results of tally 2 (lower energy region, left plots on Fig. 5.26 to Fig. 5.32) as calculated by MCNP and Serpent deviate from the experimental data in a similar way for the two codes and for all cases. The results of tally 12 (upper energy region, right plots) deviate differently from the experimental neutron flux spectrum. Comparison between MCNP and Serpent of the tally without additional treatment shows good agreement, therefore the differences seen for tally 12 for energies above several MeV should be attributed to the differences in the tally treatment.

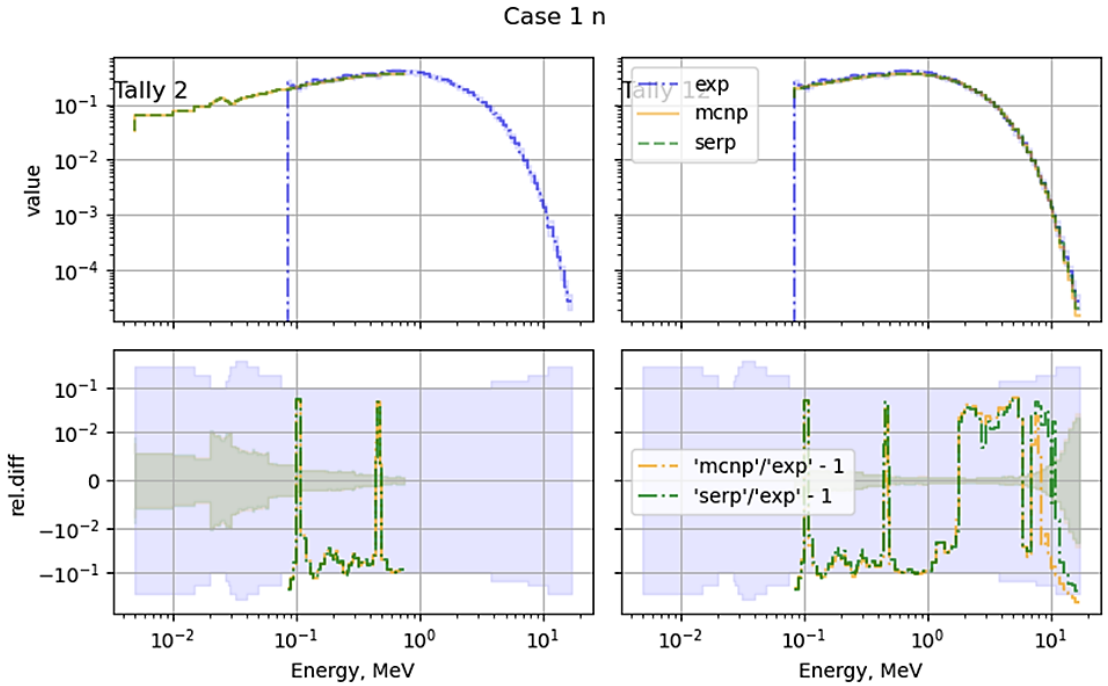



**Fig. 5.26** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 1 (bare source)

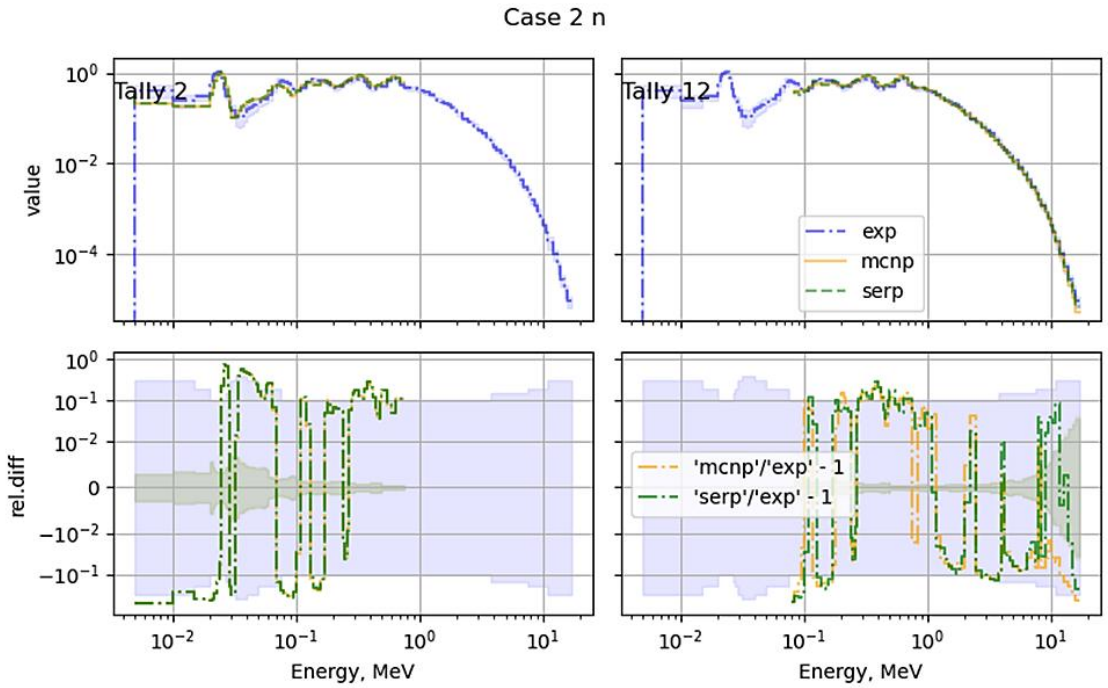



**Fig. 5.27** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 2

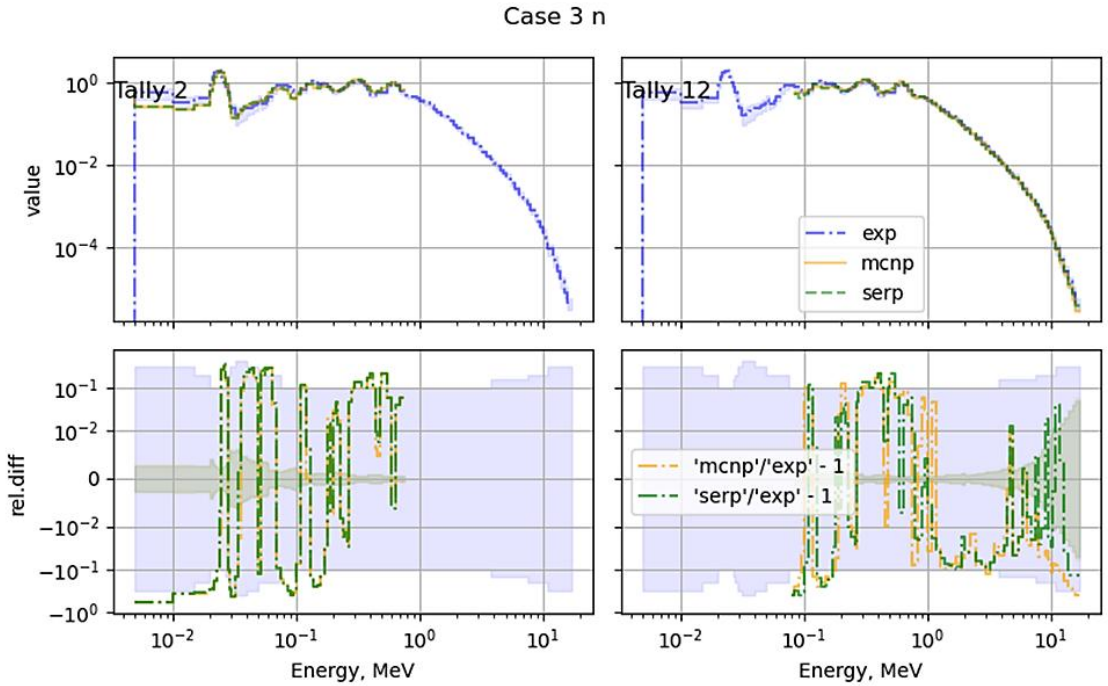



**Fig. 5.28** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 3

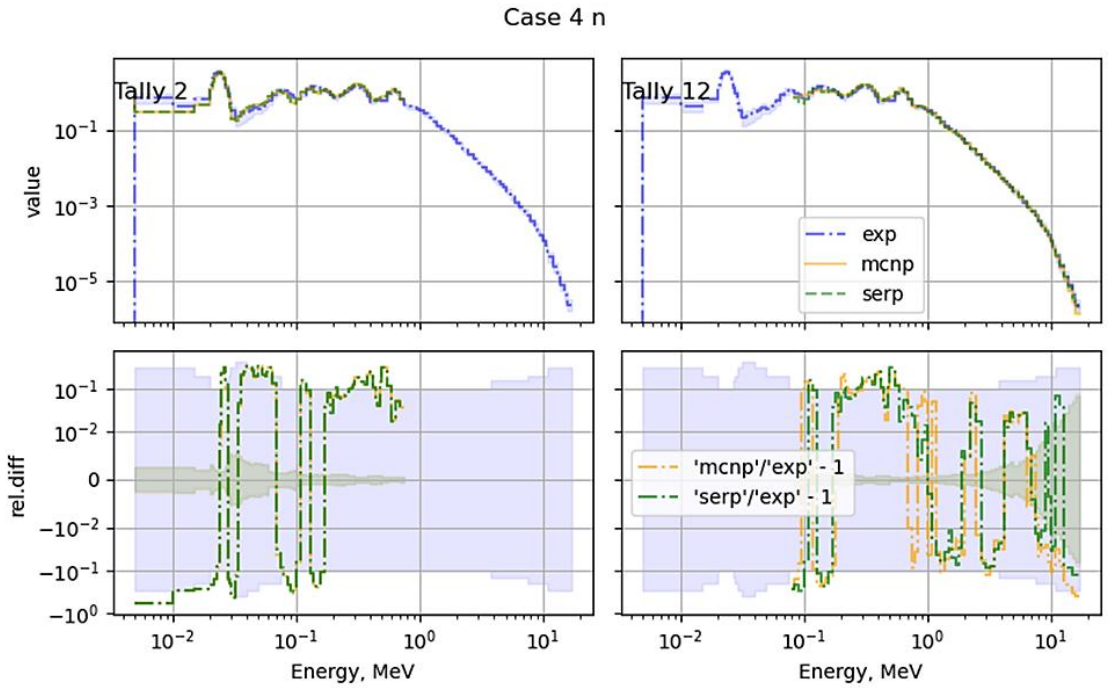



**Fig. 5.29** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 4

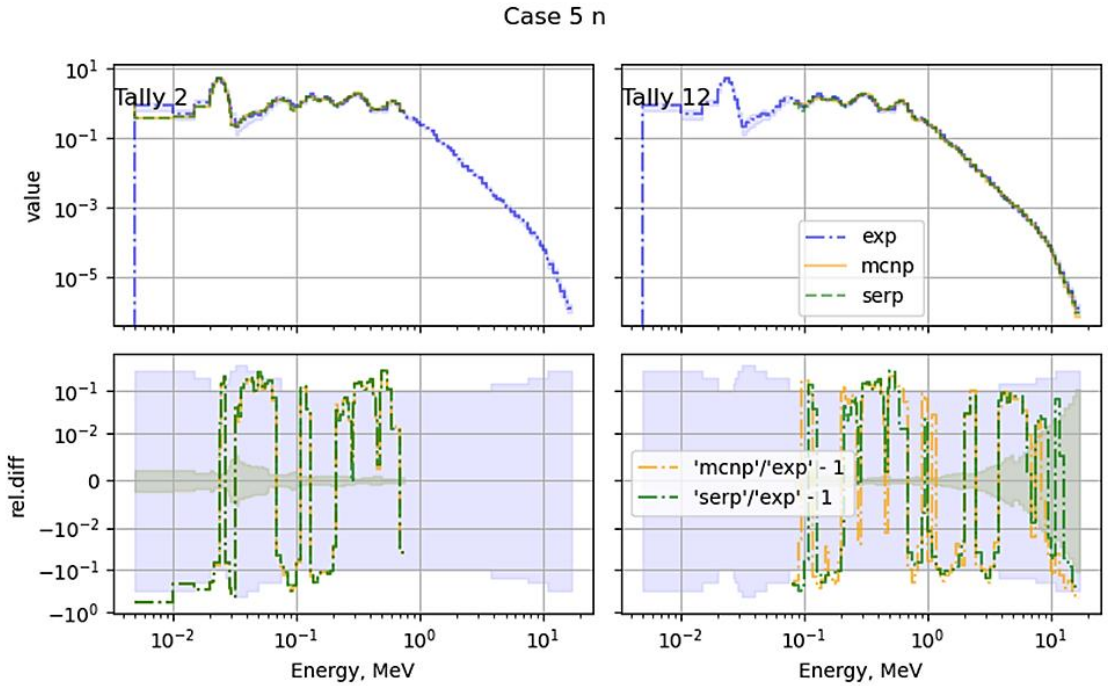



**Fig. 5.30** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 5

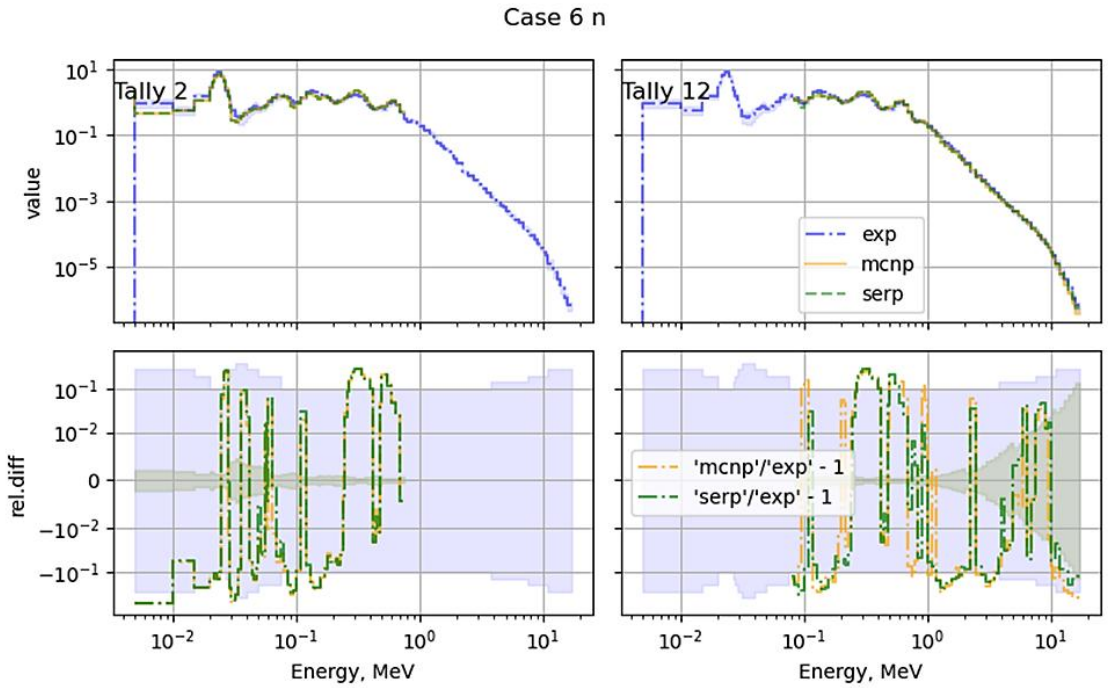



**Fig. 5.31** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 6

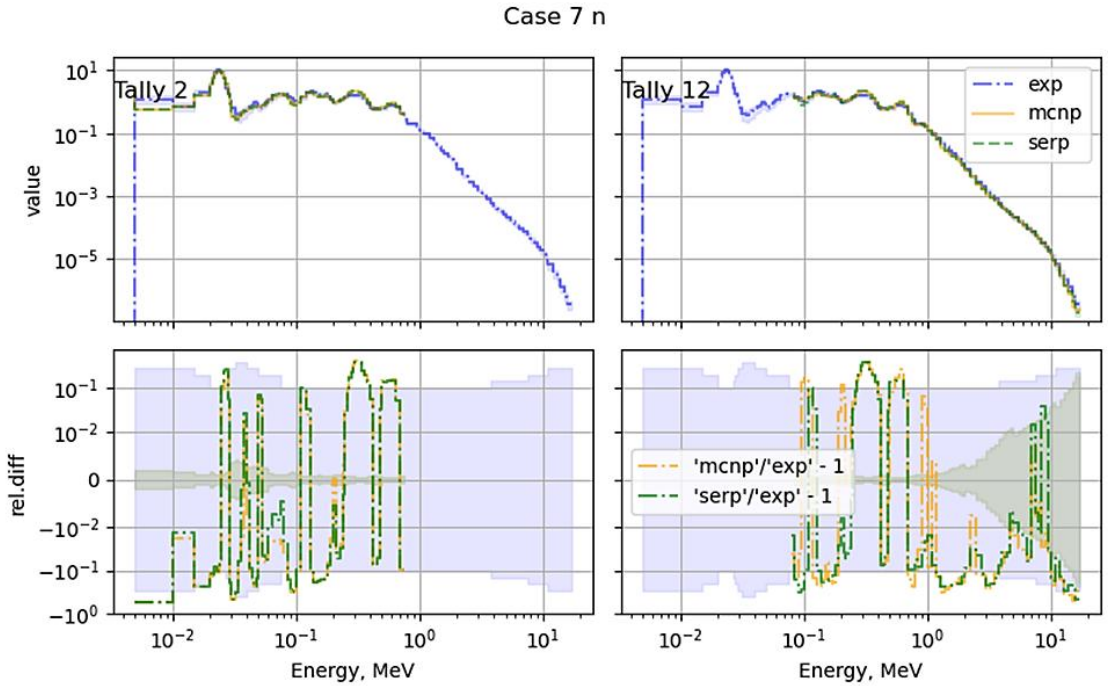



**Fig. 5.32** Neutron spectra of tally 2 and 12, calculated with MCNP and Serpent, compared to experimental benchmark data: Case 7

## 5.7 Conclusion

The calculation of a benchmark problem provided the opportunity to evaluate the integral capabilities of Serpent to model an external source and to perform shielding calculations both for neutrons and photons.

Some dependences of the results on the modelling options were identified:

- The choice of the neutron interaction data library (ENDF/B-VII.1 or JEFF-3.2) affects the neutron spectra in all shielded cases. The data library also affects the spectrum of secondary photons.
- The secondary photon spectrum is found to be dependent on the implicit or analogous treatment of the capture and n, xn reactions, to an extent that depends on the shielding thickness. Similar calculations were performed also with MCNP in /SOM 19/, however no such effect on the results was observed and reported.
- Apart from these observations, which became apparent only when considering the relative differences between results, the neutron flux calculated with Serpent is in a good agreement (up to the same extent as the MCNP-calculated neutron

spectrum) with experimental results. The deviations have the same form as the MCNP-computed results, which points to the accuracy of the experimental results, accuracy of the benchmark models and its implementation in MCNP and Serpent, but not to the accuracy of the physics modelling implemented in Serpent.

# 6 Conclusion

In this work, the computer code Serpent, and specifically the capabilities of its criticality calculation mode to perform shielding calculations, were assessed. The focus was on the fixed-source calculation mode, the photon transport and the application of variance reduction methods via the utilization of weight windows.

Two different aspects of the fixed-source calculation mode were tested. On one side the ability to control the population of particle tracks in multiplying media was analysed, which is needed to model decay neutrons in an irradiated fuel assembly storage. For this, Serpent provides several methods. They were tested on a model that allows an adjustment of the neutron multiplication factor. The efficiency of the methods and their impact on the calculated time-dependent neutron flux behaviour was evaluated. The work shows that the most general method is the dynamic external source simulation mode.

The other aspect of the fixed-source calculation mode is the application of weight-window variance reduction methods to control track populations in space and energy domains. In Serpent, the user-given weight-window parameters can be defined on a spatial mesh in cylinder or rectangular geometry. The subdivision of the energy grid can be done in arbitrary intervals. In addition, Serpent provides means to generate weight-window meshes globally or only for particular responses. The iterative method, which is used to generate weight-window meshes, was tested successfully on two models that differ by the flux attenuation factor.

The capabilities of Serpent to model the neutron and photon transport physics was validated by comparing its results against experimental measurements and former results of MCNP calculations. The shielding benchmark ALARM-CF-FE-SHIELD-001 from the International Criticality Safety Benchmark Evaluation Project (ICSBEP) handbook was chosen for this comparison. The calculations with Serpent result in very similar predictions of the experimental results compared to the MCNP calculations.

Overall, the performed work shows that Serpent can be applied routinely for shielding calculations as a viable alternative for MCNP.

## List of references


/HÄK 22/ Häkkinen, S.: serpent 2 Validation for Radiation Shielding Applications. Journal of Nuclear Engineering and Radiation Science, Bd. 8, Nr. 4, DOI 10.1115/1.4051614, 2022.

/HOL 24/ Holcomb, A. M., van der Marck, S., Trkov, A., Cabellos, O.: Automated Conversion of International Criticality Safety Benchmark Models: Developing a Reproducible Serpent-2 Model Repository from MCNP Inputs. OECD Nuclear Energy Agency (NEA), 2024.

/NEA 25a/ OECD-NEA: International Criticality Safety Benchmark Evaluation Project (ICSBEP) Handbook. Erreichbar unter https://www.oecd-nea.org/jcms/pl_20291/icsbep-handbook.

/NEA 25b/ OECD-NEA: Collection of Serpent input files for ICSBEP. Erreichbar unter https://git.oecd-nea.org/serpent/icsbep/.

/NEA 24/ OECD Nuclear Energy Agency (NEA) (Hrsg.): International Criticality Safety Benchmark Evaluation Project (ICSBEP), Handbook 2022, 2023. NEA/NSC/DOC(95)03: Paris, November 2024.

/SOM 19/ Sommer, F., Bartos, O., Behler, M., Hannstein, V., Kaufholz, P., Kilger, R., Krüger, J., Rowold, F.: Bereitstellung und Weiterentwicklung von Methoden, Daten und Rechenmethoden zu nuklearen Sicherheitsanalysen. Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH (GRS), GRS-Bericht, GRS-542, ISBN 978-3-947685-27-1: Garching, 2019.

/VTT 25a/ VTT: Serpent Wiki, Input syntax manual. Erreichbar unter https://serpent.vtt.fi/mediawiki/index.php?title=Input_syntax_manual, abgerufen am 20. September 2025.

/VTT 25b/ VTT: Serpent Monte Carlo Code online documentation. Erreichbar unter https://serpent.vtt.fi/docs/, abgerufen am 20. September 2025.

## List of figures

## List of figures Appendix C

## List of tables

# A Appendix A: Serpent input file cards relevant for shielding calculations

The cards relevant for fixed-source (FS) calculation mode and application of WW for variance reduction are listed here:

## A.1 Force FS calculation mode

Whether Serpent will follow the criticality calculation mode or the fixed-source calculation mode, depends on the card used to set the number of histories to simulate. When the `set nps` card appears in the input, Serpent activates the fixed-source mode. The `set pop` card activates the criticality calculation mode.

**Listing A.1** `set nps` input card

```
set nps  <total_num>  [<batches>  [<timeBins>]]
```

The `set nps` card in Listing A.1 specifies the total number of histories to be sampled, optionally followed by the number of batches and time bins used in the dynamic mode (see below). In Serpent the notion of batches is used both in the criticality and in the FS calculation modes. In the latter, it affects the Serpent output: the summary statistics is printed out after each batch. Also, detector statistical checks can be printed for every batch (see `dhis` option of `det` card).

The `set nps` and `set pop` are mutually exclusive.

The FS mode also needs a description of the external particle source. In practice this means that together with `set nps`, the `src` card must be given in the input file as well.

## A.2 History cut-offs

As a measure against the population growth in multiplying media, the user can set time and generation cut-offs.

**Listing A.2** `tcut` and `gcut` input card

```
set tcut <Tmax>
set gcut <Gmax>
```

The `set tcut` card sets the time cutoff for neutrons or photons, in seconds. When the particle history reaches `<Tmax>`, it is killed.

The `set gcut` card sets the maximal number of fission events in the particle history. After `<Gmax>` fission events the history is killed.

Both cards affect the time-integrated detector results. In section 3.1.1, their impact is studied in detail. The cards can be applied separately (either time or generation cut-off), or simultaneously.

## A.3 Dynamic external source simulation mode

When the third parameter in set `nps card` is present, it represents the time bins used to control the population of histories in time, see Listing A.3. The size of the population is adjusted in each time bin boundary to match the starting population, compensated by particle weight. With this method one can simulate extended times periods in the sub- and supercritical systems.

**Listing A.3** `tme` and `nps` cards to activate dynamic simulation mode

```
set nps  <total_num>  <batches>  timeBins

% Time bins for time-dependent population control
% fixed-source caluclations in multiplying media
tme timeBins
    1  % 1 arbitrary, 2 lin-uniform, 3 log-uniform
    <t1> <t2> ...
```

## A.4 Cards, related to VR with ww-mesh

The `wwgen` card triggers the generation of the ww-mesh for the geometry and source specified in the input file. It can be detector-specific (i.e. improves VR for the specific detector tally) or global (i.e. improves homogeneously VR in all parts of phase space).

**Listing A.4** **`wwgen`** and **`wwin`** cards

```
% ww-mesh generator
wwgen ...

% ww-mesh parameters
wwin ...
```

In the `wwin` card one can specify directly ww-mesh parameters to be used or refer to an external file to read. For the latter, the files generated with `wwgen` card in the previous run, or the ww-mesh files generated by MCNP can be used.

## A.5 Track buffer size

Not directly related to FS or WW, but the card `nbuf` needs to be set to avoid the error in Listing A.6:

**Listing A.5** `nbuf` card

```
set nbuf <C>
```

**Listing A.6** Serpent error output

```
Input error:

Insufficient neutron buffer size, increase value of
parameter "nbuf" (currently set to 5.0)
```

During history sampling, the new particles appearing in events, for example fission neutrons or photons, are stored in a buffer for later transport. The buffer size is estimated by Serpent at the beginning of the simulation. For some problems characterized by many secondary particles, the initialized buffer becomes insufficient. For this case, the user can adjust the buffer size by specifying a coefficient in the `set nbuf <C>` and `set gbuf <C>` cards, for neutron and photon buffers, respectively.

The coefficient defines the buffer size relative to the simulated batch size. In the multiplying systems, as criticality approaches 1, the variation of the number of secondary particles becomes infinitely large, and it is very probable that the buffer will overflow. To some extent this problem can be solved by increasing the coefficient. It should be noted however, that it affects the total amount of memory used by Serpent.

# B Appendix B: Test model input file

The model input file to test the fixed-source calculation mode is shown in Listing B.1:

**Listing B.1** model input file to test the fixed-source calculation mode

```
set acelib s2v0_jeff32+mcplib.xsdata % relative to $SERPENT_DATA

mat water -1.0  rgb 10 10 225
    8016.03c 1
    1001.03c 2
mat metU -19.0  rgb 225 10 10
    92235.03c __U5  % U-enrichment is used to control Keff of the system.
    92238.03c __U8  % Completely non-fissionable material cannot be used
in Keff calculations.
mat myVoid 1e-15 rgb 125 125 125
    2004.03c 1.0

% Cells with materials
cell channel    0 myVoid              -innerCyl1  planeLower -planeUpper
cell Ucylinder  0 metU      innerCyl1 -innerCyl2  planeLower -planeUpper
cell Wreflector 0 water     innerCyl2 -outerCyl   planeLower -planeUpper
% cells for detectors and for source
cell lowerVoid1 0 myVoid              -innerCyl1 -outerCyl -planeLower
cell lowerVoid2 0 myVoid    innerCyl1 -innerCyl2 -outerCyl -planeLower
cell lowerVoid3 0 myVoid    innerCyl2            -outerCyl -planeLower
cell upperVoid1 0 myVoid              -innerCyl1 -outerCyl planeUpper
cell upperVoid2 0 myVoid    innerCyl1 -innerCyl2 -outerCyl planeUpper
cell upperVoid3 0 myVoid    innerCyl2            -outerCyl planeUpper
% the other world
cell rest       0 outside outerCyl

surf planeLower pz -10
surf planeUpper pz  10
surf innerCyl1 cylz 0 0  1
surf innerCyl2 cylz 0 0  8
surf outerCyl  cylz 0 0 30 -11 11

tme timeBins010 1
        0.0
        1e-8 3e-8 5e-8
        1e-7 3e-7 5e-7
        1e-6 3e-6 5e-6
        1e-5 3e-5 5e-5
        1e-4 3e-4 5e-4
        1e-3 3e-3 5e-3
        1e-2 3e-2 5e-2
        1e-1 3e-1 5e-1
        1e-0 3e-0 5e-0
        1e+1

tme timeBins100 3 100 1e-9 1e1
tme timeBins100Lin 2 100 0 1e-3
% set cfe -1 1e-6
% Detectors at lower and upper layers:
det detLower
    dc lowerVoid1
    dc lowerVoid2
    dc lowerVoid3
    di timeBins100
det detUpper
    dc upperVoid1
    dc upperVoid2
    dc upperVoid3
```

```
    di timeBins100
det detLowerLin
    dc lowerVoid1
    dc lowerVoid2
    dc lowerVoid3
    di timeBins100Lin
det detUpperLin
    dc upperVoid1
    dc upperVoid2
    dc upperVoid3
    di timeBins100Lin
det allGeometry
    dc lowerVoid1
    dc lowerVoid2
    dc lowerVoid3
    dc upperVoid1
    dc upperVoid2
    dc upperVoid3
    dc channel
    dc Ucylinder
    dc Wreflector
det detMesh1
    dn
    1  % even cylindrical mesh
    0  30   30  % Rmin Rmax
    0 360    1  % ThetaMin ThetaMax
   -11 11   22  % Zmin Zmax

% Plot geometry
plot 13          % yz, all boundaries
     1024 1024   % image size, pixels
     0           % plot plane position on x axis
     -35 35      % horizontal axis range
     -35 35      % vertical axis range
plot 33        % xy, all boundaries
    1024 1024  % image size, pixels
    10         % plot plane position on z axis
    -35 35     % horizontal axis range
    -35 35     % vertical axis range

% Plot fast-thermal neutron flux
%(note: orientation 4 works only with detector mesh plots)
mesh
    1          % projection to yz plane
    1024 1024  % plot resolution
    0             % SYM, dummy in serpent-2
    -35 35        % XYZ range. Set to a cube to have a proper-scaled plot
    -35 35
    -35 35
mesh
    3          % projection to xy plane
    1024 1024
    0             % SYM, dummy in serpent-2
    -35 35        % XYZ range. Set to a cube to have a proper-scaled plot
    -35 35
    -35 35
% Plot scores contributing to detector
% This is spatial distribution of scores inside the detector's volume
% Thus, to get an overall picture, one should apply this mesh plot to
% a detector that covers the entire geometry.
mesh
    8              % plot scores contributing to detector
    -13            % blue-red log scale
    allGeometry    % detector name
    1              % projection to yz plane
    1024 1024      % plot resolution
    0              % SYM, dummy in serpent-2
```

```
    -35 35        % XYZ range. Set to a cube to have a proper-scaled plot
    -35 35
    -35 35
mesh
    8             % plot scores contributing to detector
    -13           % blue-red log scale
    allGeometry   % detector name
    3             % projection to xy plane
    1024 1024     % plot resolution
    0             % SYM, dummy in serpent-2
    -35 35        % XYZ range. Set to a cube to have a proper-scaled plot
    -35 35
    -35 35
mesh
    8             % plot scores contributing to detector
    -13           % blue-red log scale
    allGeometry   % detector name
    4             % cyliner mesh to plot
    512 512       % plot resolution. Reduced 2 times to get the
                  % same plot size as in prev meshes
    0
    0 35          % radius
    -17.5 17.5    % z axis range

% By default, Serpent normalizes the results to the unit total loss
% rate, see `set genrate` in wiki. Here, another normalization is
% chosen, to the unit source neutrons. This is (1) comparable to
% MCNP and (2) simplifies comparison between Keff-mode and FS-mode
% calculaitons.
set srcrate 1.

% Model delayed neutrons. By default is is off for FS-mode.
set delnu 1

% Mutually exclusive cards for criticality/fixed-source mode
% set pop   10000 200 10   % criticality calculation mode
% set nps 2000000 200      % fixed-source mode

set nbuf 100

% Definition of the fixed-source
% sp defines the center for radial range
% src lowerSource sp 0 0 0 srad 0 30 sz -11 -10 sd 0 0 1
```

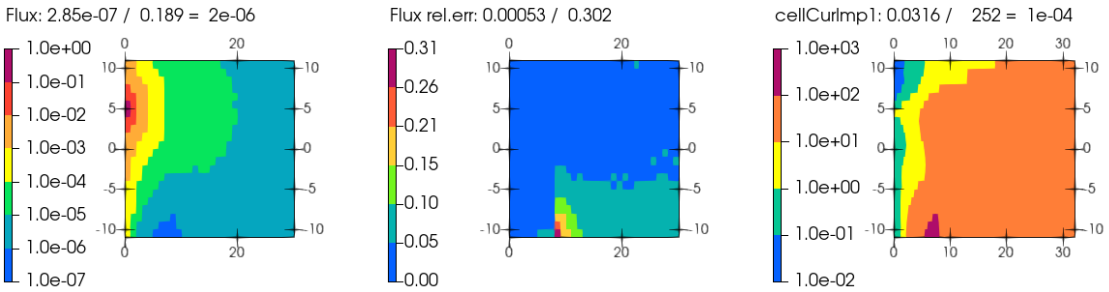
# C Appendix C: Results for highly shielded model

Description of the models and analysis of results see in section 4.2.

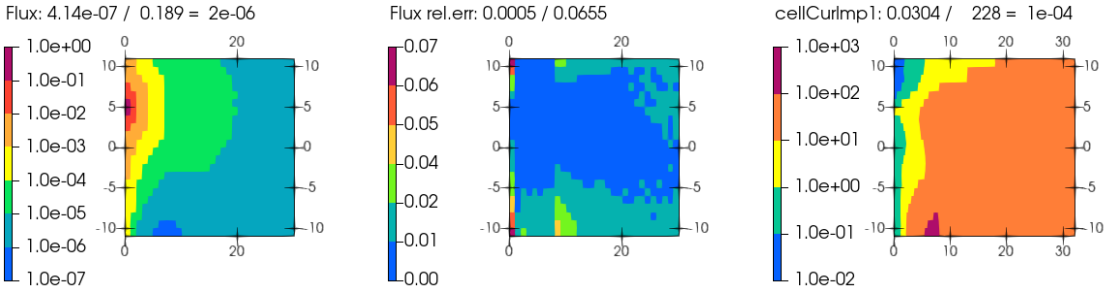
## C.1 Calculation 1

**Listing C.1** `wwin` and `wwgen` cards for calculation 1

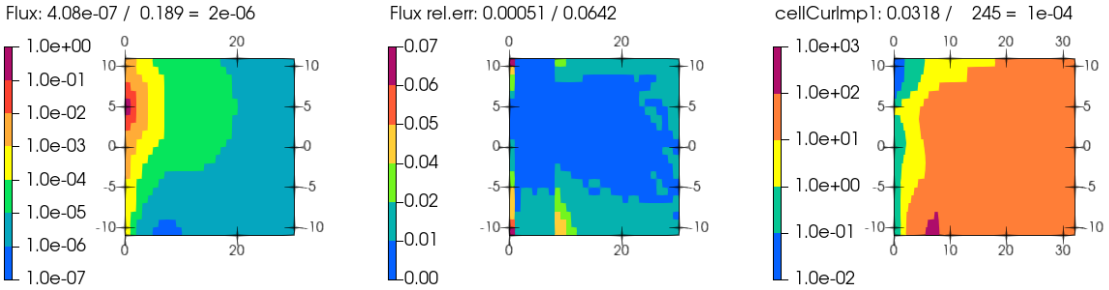
```
wwgen wwGlobal2
    1e-9 10000
    3 -1
    8 42 1 22
    0 0.5 1.0
    1.2 1.4 1.6 1.8 2.0
    2.2 2.4 2.6 2.8 3.0
    3.2 3.4 3.6 3.8 4.0
    4.2 4.4 4.6 4.8 5.0
    5.2 5.4 5.6 5.8 6.0
    6.2 6.4 6.6 6.8 7.0
    7.2 7.4 7.6 7.8 8.0
    13 18 23 28 32
    0 360
    -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11

% Global VR iterations
wwin wwGlobal
    wi 1 27
    wwGlobal2 5e-2
    wwGlobal2 5e-2
    wwGlobal2 5e-2
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
```



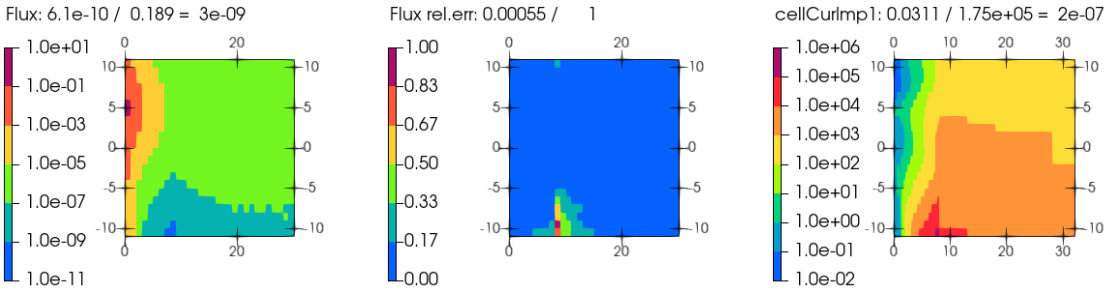

(a) Iteration 0, GDF=0.05



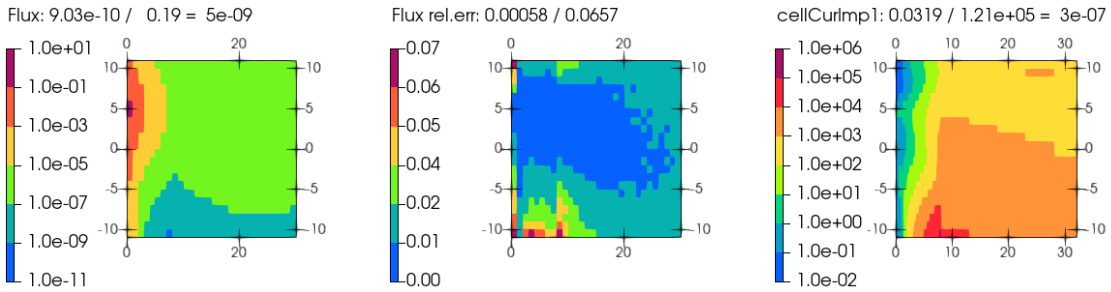

(b) Iteration 1, GDF=0.05



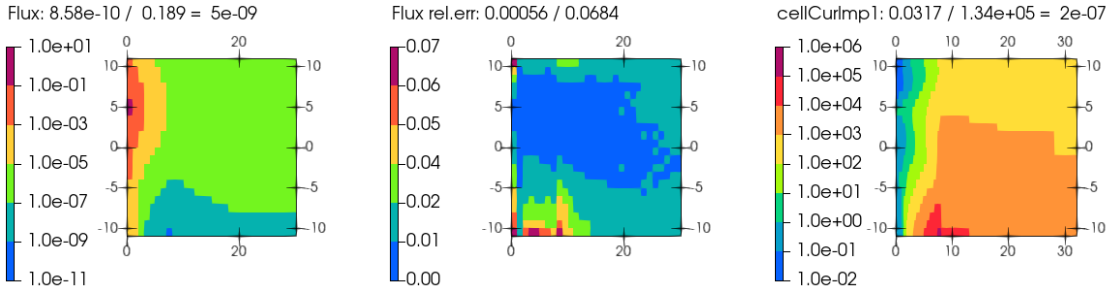

(c) Iteration 2, GDF=0.05



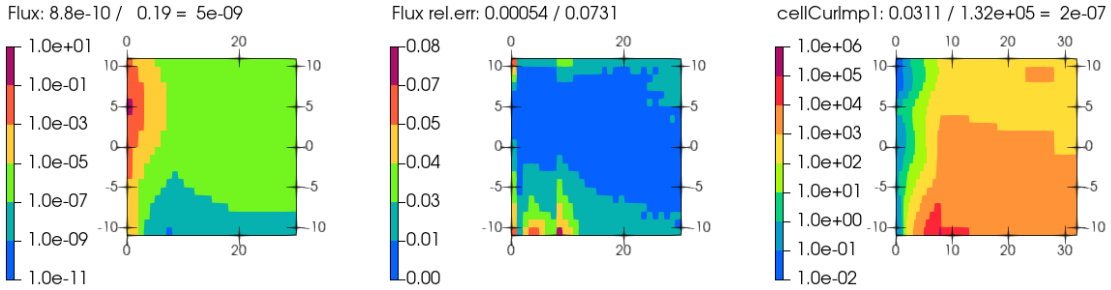

(d) Iteration 3, GDF=0.1



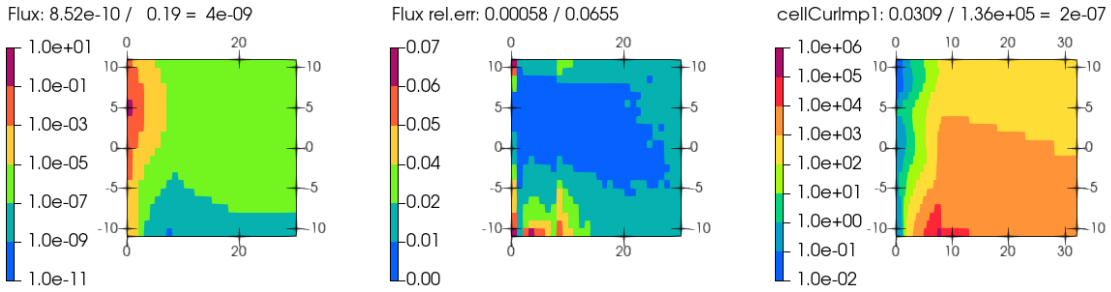

(e) Iteration 4, GDF=0.1



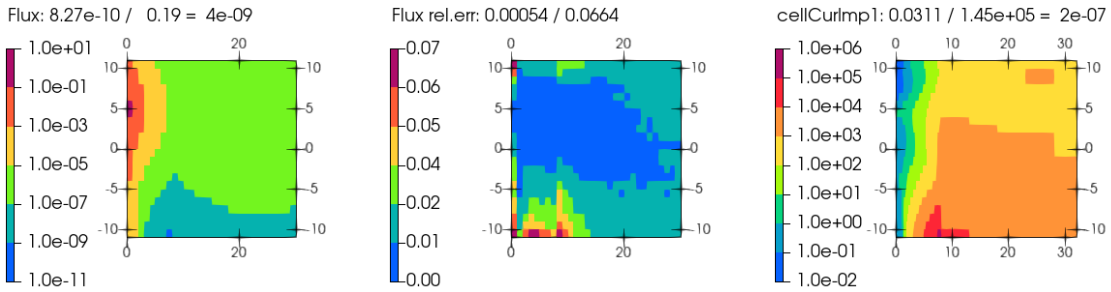

(f) Iteration 5, GDF=0.1



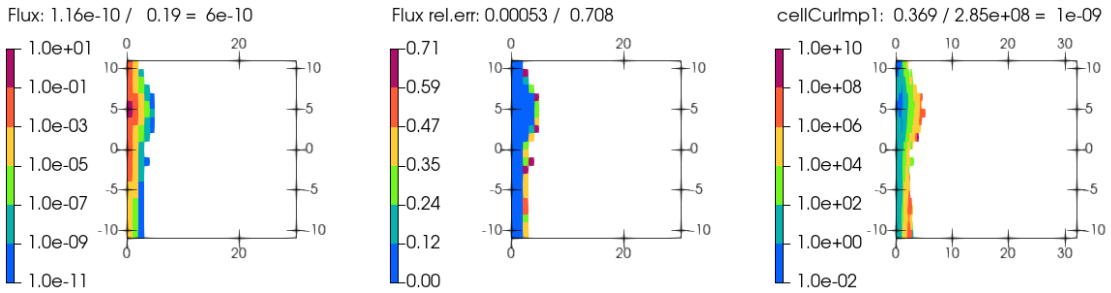

(g) Iteration 6, GDF=0.1



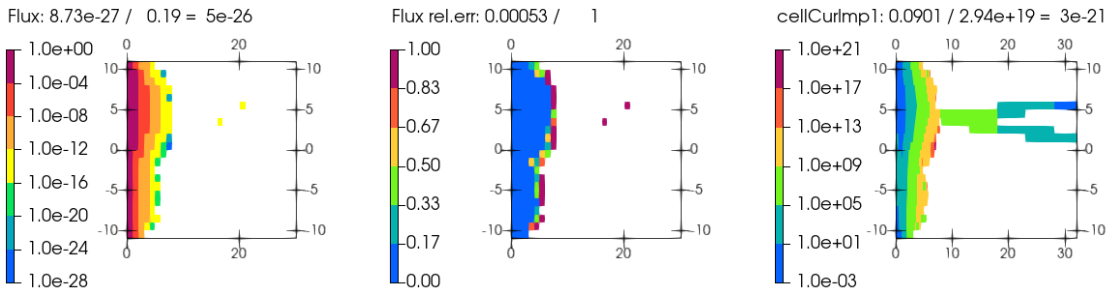

(h) Iteration 7, GDF=0.1



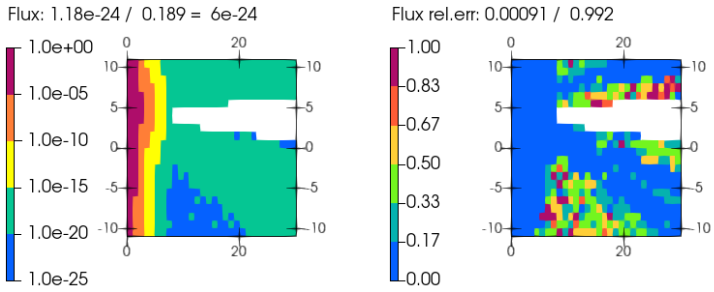

(i) Iteration 8, GDF=0.1




(j) Iteration 9, GDF=0.5




(k) Iteration 10, GDF=0.5



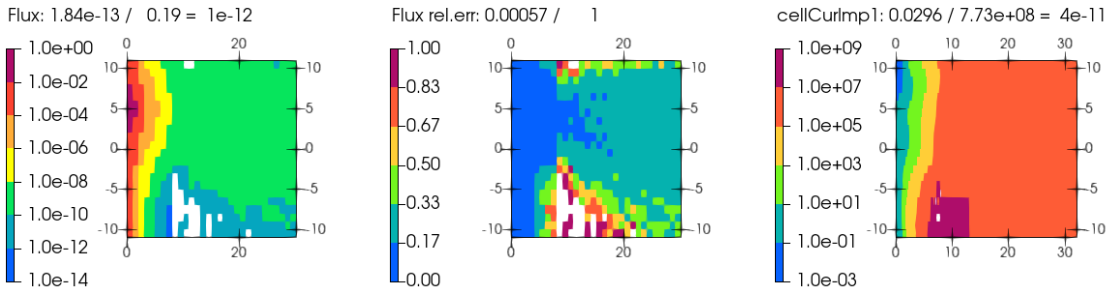

(l) Iteration 11, GDF=0.5

**Fig. C.1** Calculation 1 results: ww-mesh, flux and error on each iteration

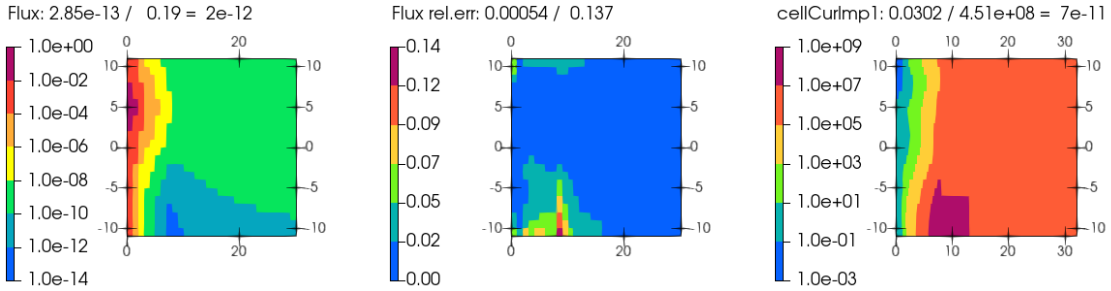
## C.2 Calculation 2

**Listing C.2** `wwin` and `wwgen` cards for calculation 2

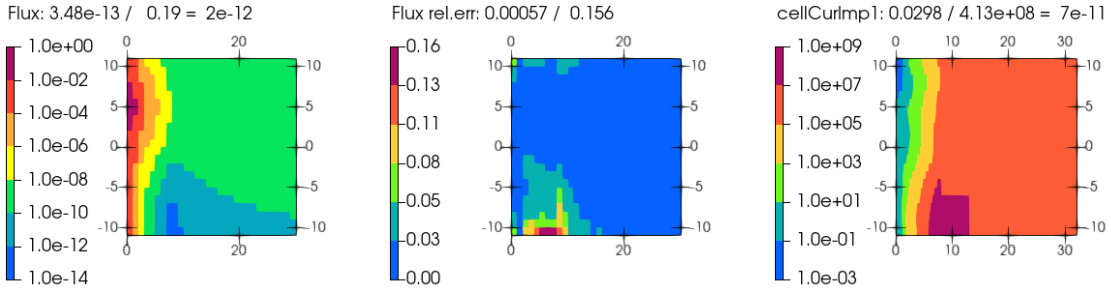
```
wwgen wwGlobal2
    1e-9 10000
    3 -1
    -1

% Global VR iterations
wwin wwGlobal wf "./wd.gvrAll.inp/gvrAll.inp.wwd8" 1
    wi 1 18
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
```



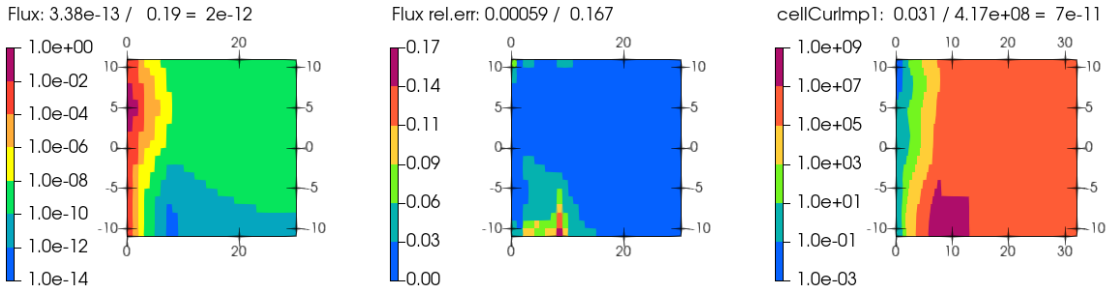

(a) Iteration 0, GDF=0.2



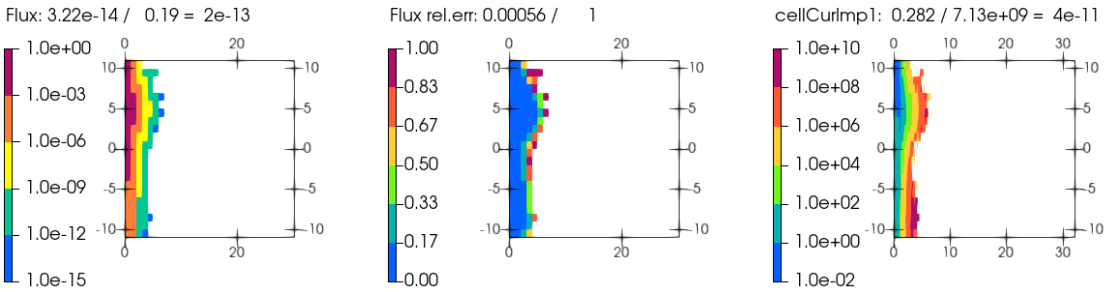

(b) Iteration 1, GDF=0.2



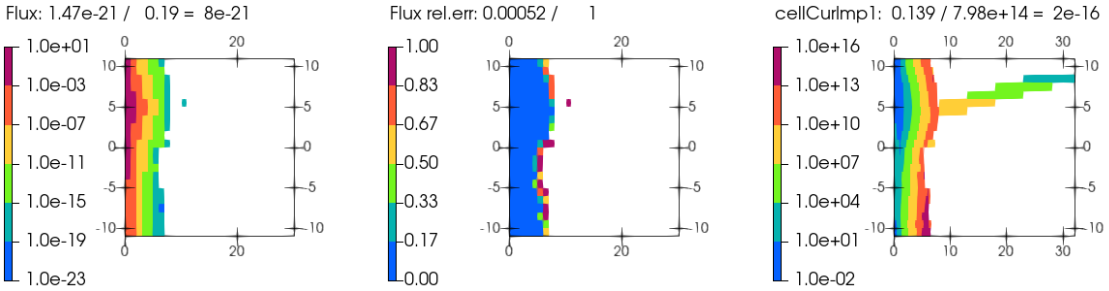

(c) Iteration 2, GDF=0.2



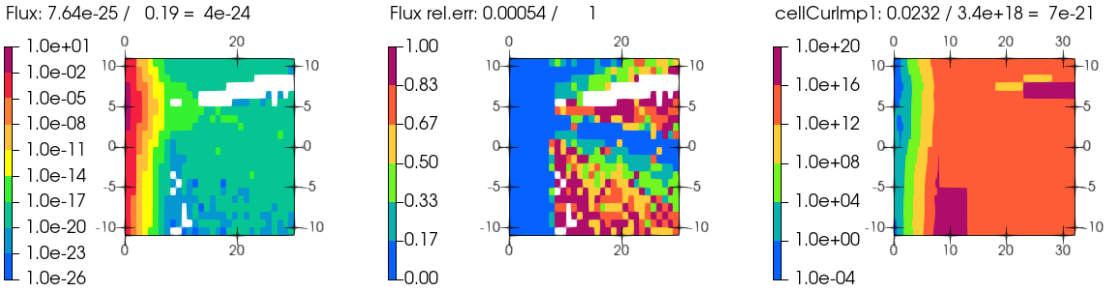

(d) Iteration 3, GDF=0.2



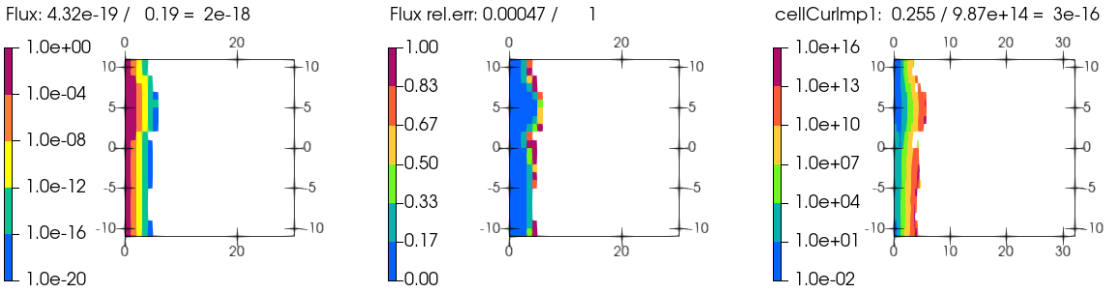

(e) Iteration 4, GDF=0.5



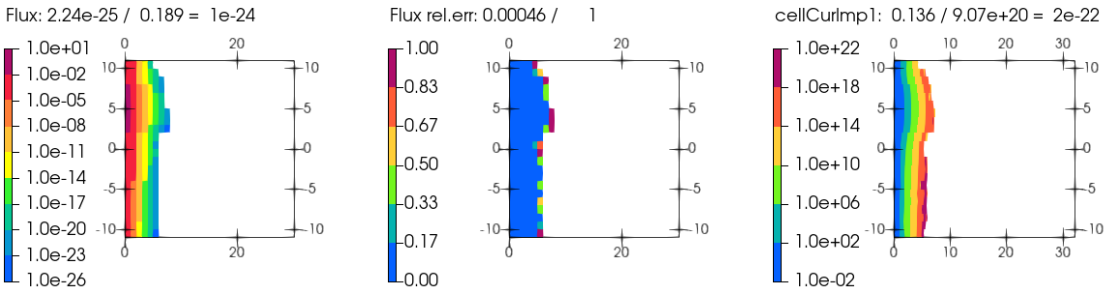

(f) Iteration 5, GDF=0.5



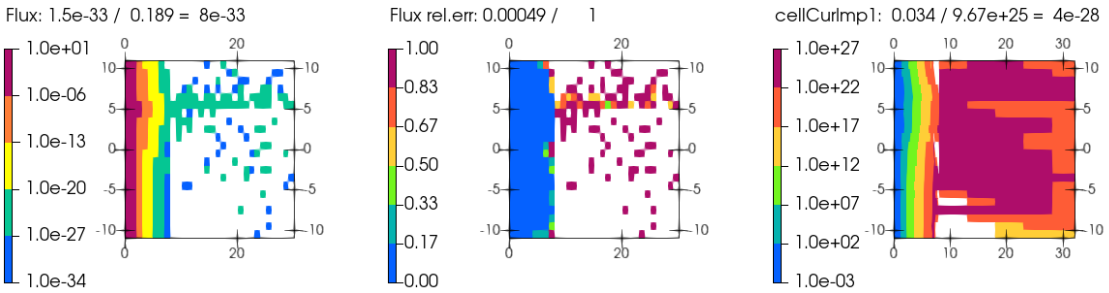

(g) Iteration 6, GDF=0.5



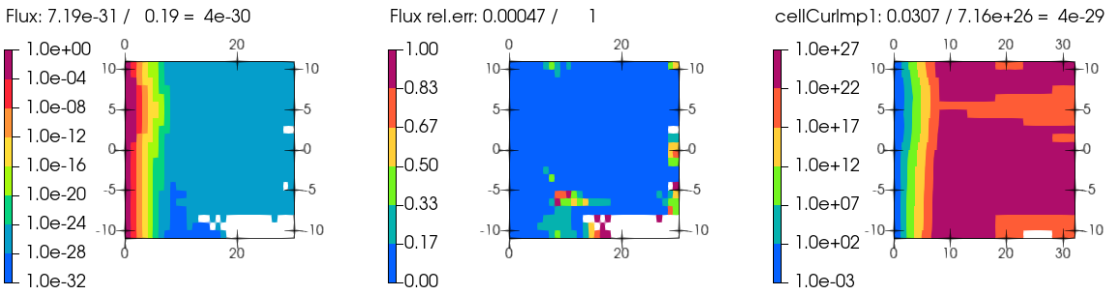

(h) Iteration 7, GDF=0.8



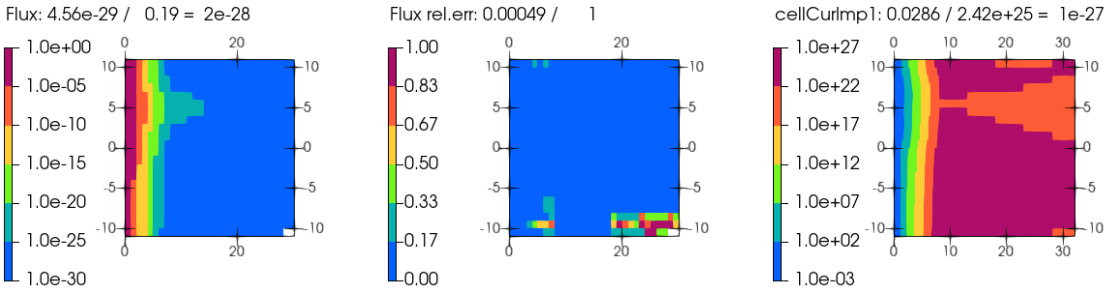

(i) Iteration 8, GDF=0.8



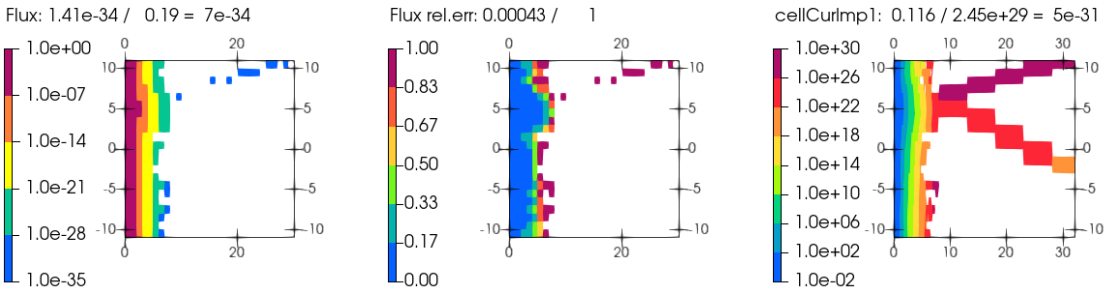

(j) Iteration 9, GDF=0.8



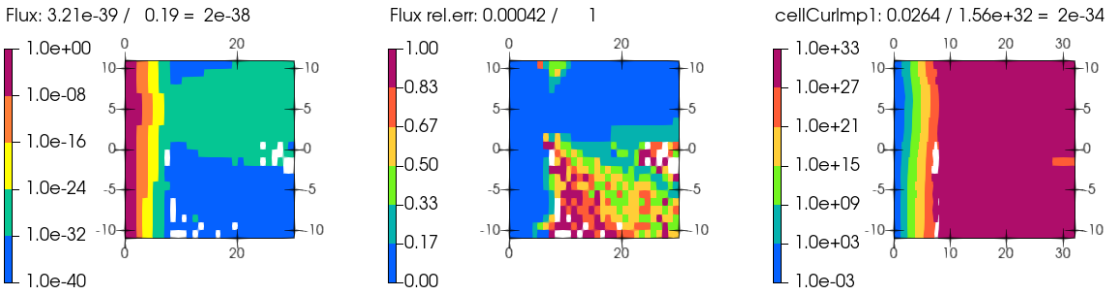

(k) Iteration 10, GDF=0.8



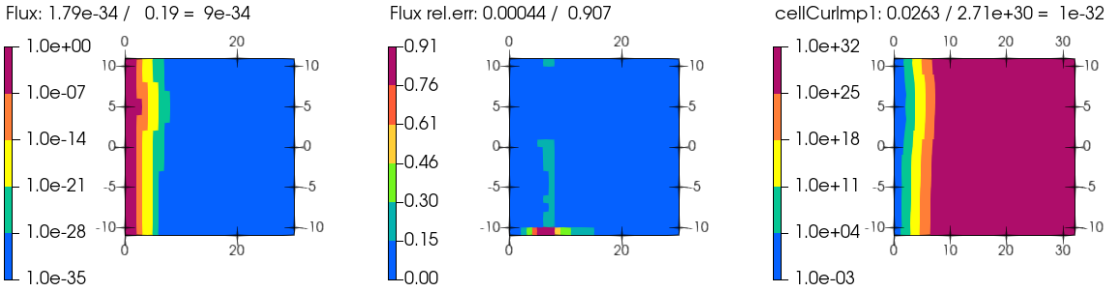

(l) Iteration 11, GDF=0.8



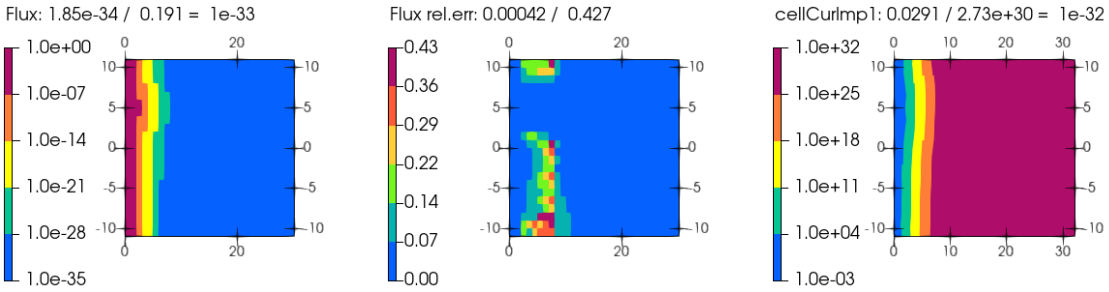

(m) Iteration 12, GDF=1.0



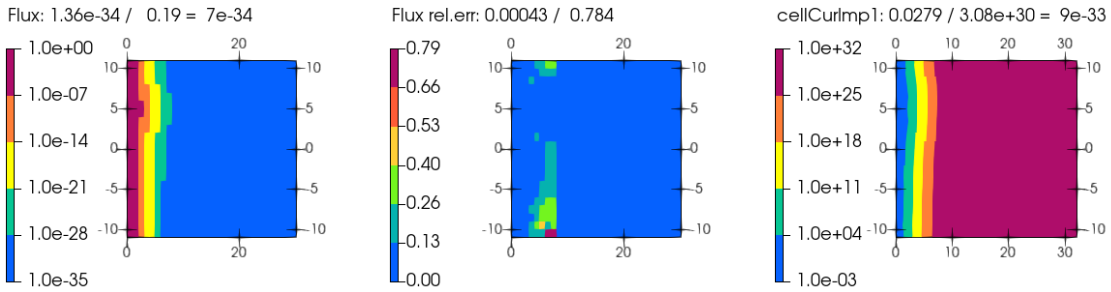

(n) Iteration 13, GDF=1.0



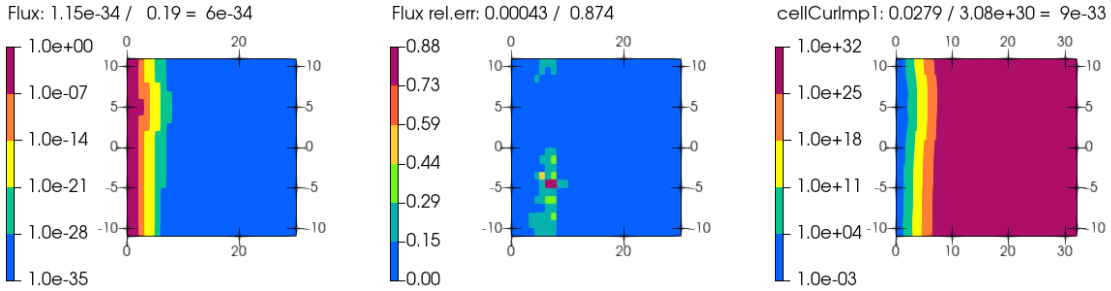

(o) Iteration 14, GDF=1.0




(p) Iteration 15, GDF=1.0




(q) Iteration 16, GDF=1.0




(r) Iteration 17, GDF=1.0

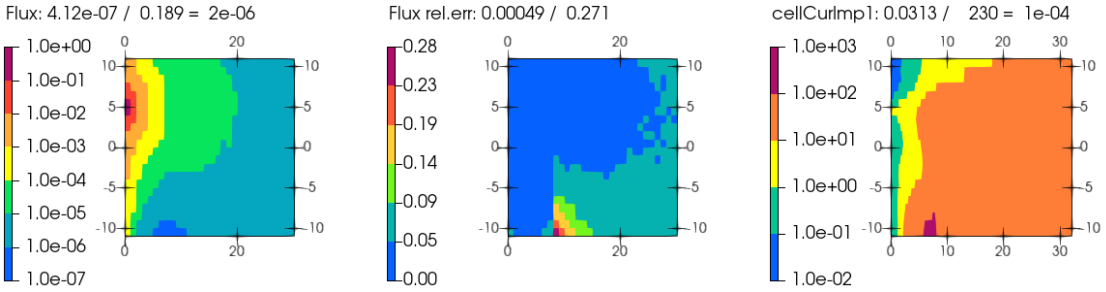
**Fig. C.2** Calculation 2: smaller gdf steps

## C.3 Calculation 3

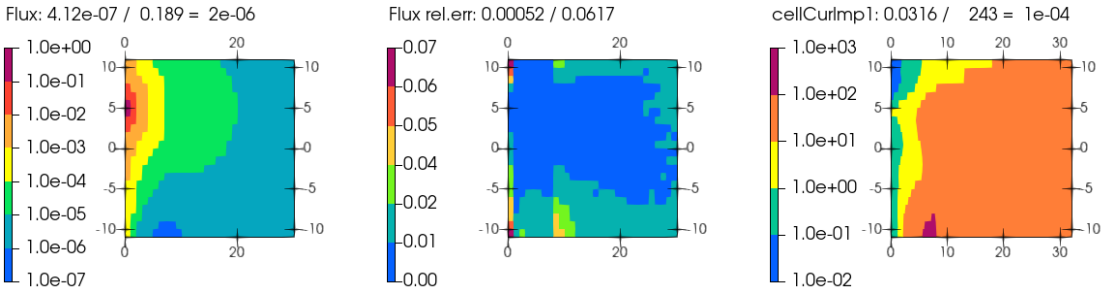
**Listing C.3** `wwin` and `wwgen` cards for calculation 3

```
wwgen wwGlobal2
    1e-9 10000
    3 -1
    8 42 1 22
    0 0.5 1.0
    1.2 1.4 1.6 1.8 2.0
    2.2 2.4 2.6 2.8 3.0
    3.2 3.4 3.6 3.8 4.0
    4.2 4.4 4.6 4.8 5.0
    5.2 5.4 5.6 5.8 6.0
    6.2 6.4 6.6 6.8 7.0
    7.2 7.4 7.6 7.8 8.0
    13 18 23 28 32
    0 360
    -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11

% Global VR iterations
wwin wwGlobal
    wi 1 24
    wwGlobal2 5e-2   % 2.95 minutes.     cell importance: 3.13285E-02
2.30239E+02
    wwGlobal2 5e-2   % 2.63 minutes.     cell importance: 3.15538E-02
2.42973E+02
    wwGlobal2 5e-2   % 2.56 minutes.     cell importance: 3.09912E-02
2.33341E+02
    wwGlobal2 5e-2   % 2.51 minutes.     cell importance: 2.95418E-02
2.40082E+02
    wwGlobal2 1e-1   % 56.9 seconds.     cell importance: 3.15760E-02
1.39142E+05
    wwGlobal2 1e-1   % 3.03 minutes.     cell importance: 3.06918E-02
1.24733E+05
    wwGlobal2 1e-1   % 3.13 minutes.     cell importance: 3.09814E-02
1.17795E+05
    wwGlobal2 1e-1   % 3.09 minutes.     cell importance: 3.18493E-02
1.40188E+05
    wwGlobal2 2e-1   % 46.1 seconds.     cell importance: 2.96482E-02
5.33396E+08
    wwGlobal2 2e-1   % 3.37 minutes.     cell importance: 3.06985E-02
4.14016E+08
    wwGlobal2 2e-1   % 3.3 minutes.      cell importance: 3.03079E-02
4.52364E+08
    wwGlobal2 2e-1   % 3.35 minutes.     cell importance: 2.98639E-02
4.41818E+08
    wwGlobal2 5e-1   % 33.8 seconds.     cell importance: 2.53342E-01
1.71037E+10
    wwGlobal2 5e-1   % 2.14 minutes.     cell importance: 1.44313E-01
1.70382E+15
    wwGlobal2 5e-1   % 3.43 minutes.     cell importance: 3.31153E-02
6.65383E+18
    wwGlobal2 5e-1   % 2.51 hours.       cell importance: 2.66570E-02
1.70496E+17
    wwGlobal2 8e-1   % 29.9 seconds.     cell importance: 2.20256E-01
4.68432E+15
    wwGlobal2 8e-1   % 1.99 minutes.     cell importance: 8.00647E-06
5.46793E+26
    wwGlobal2 8e-1   % 11.9 minutes.     cell importance: 1.19572E-03
7.97121E+27
    wwGlobal2 8e-1   % 8.68 minutes.     cell importance: 1.05614E-03
6.28207E+29
```

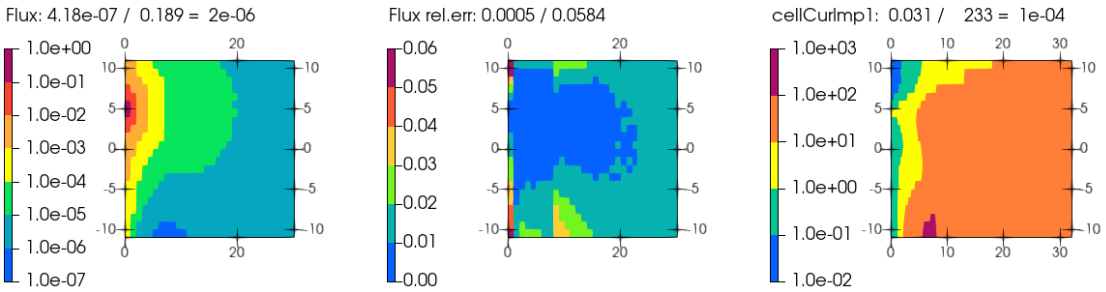
```
    wwGlobal2 1       % 31.4 seconds.      cell importance: 1.65187E-01
1.53760E+24
    wwGlobal2 1       % 9.85 minutes.      cell importance: 1.24088E-01
3.15935E+29
    wwGlobal2 1       % 3.4 minutes.       cell importance: 1.32101E-02
7.67906E+32
    wwGlobal2 1
```



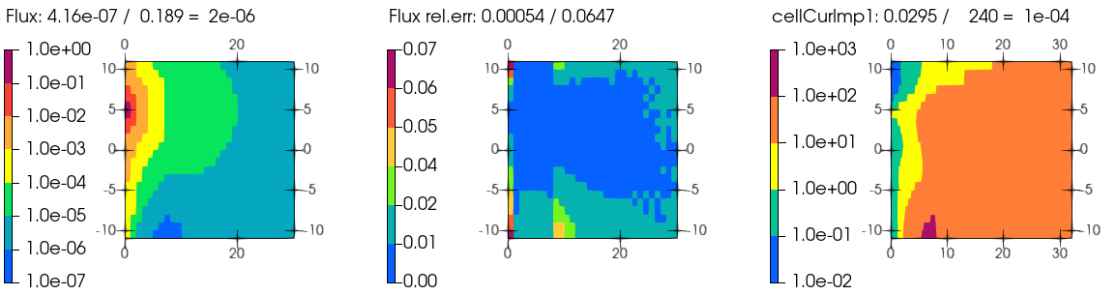

*(a) Iteration 0, GDF=0.05*



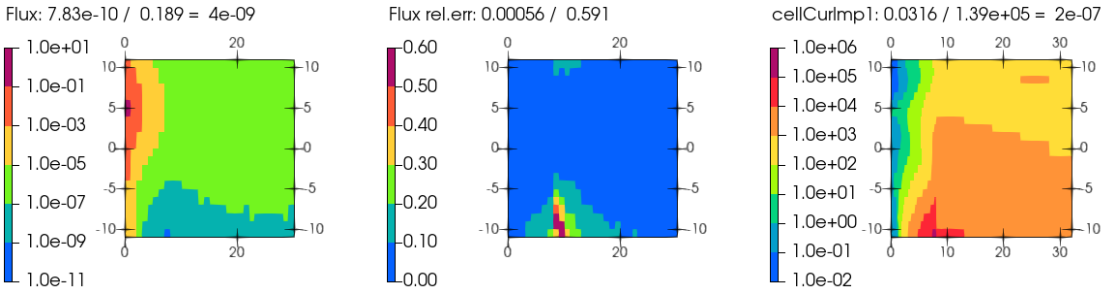

*(b) Iteration 1, GDF=0.05*



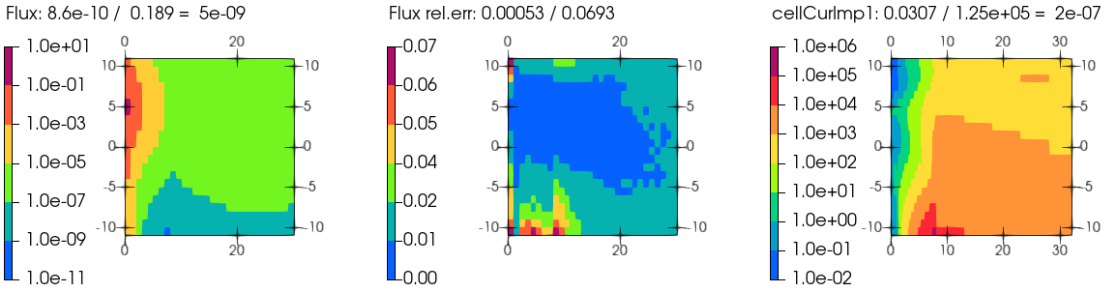

*(c) Iteration 2, GDF=0.05*



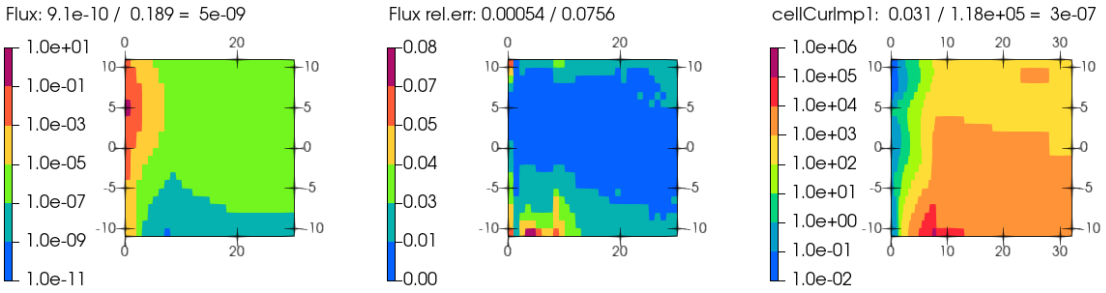

*(d) Iteration 3, GDF=0.05*



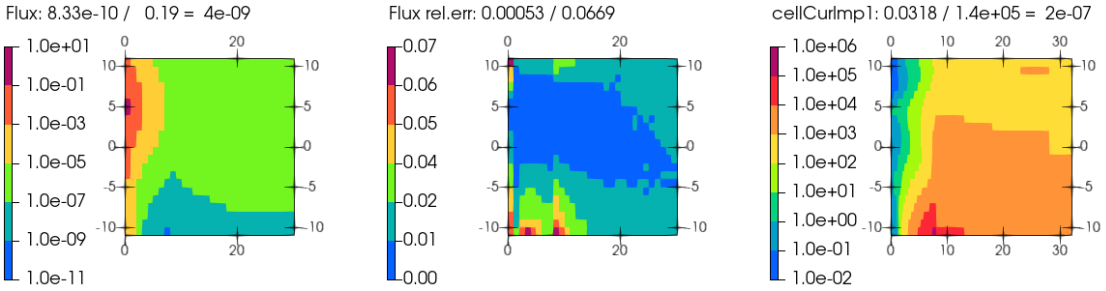

*(e) Iteration 4, GDF=0.1*



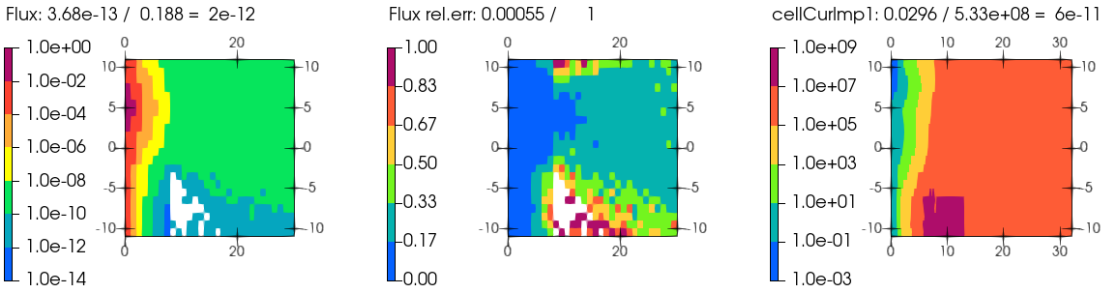

*(f) Iteration 5, GDF=0.1*



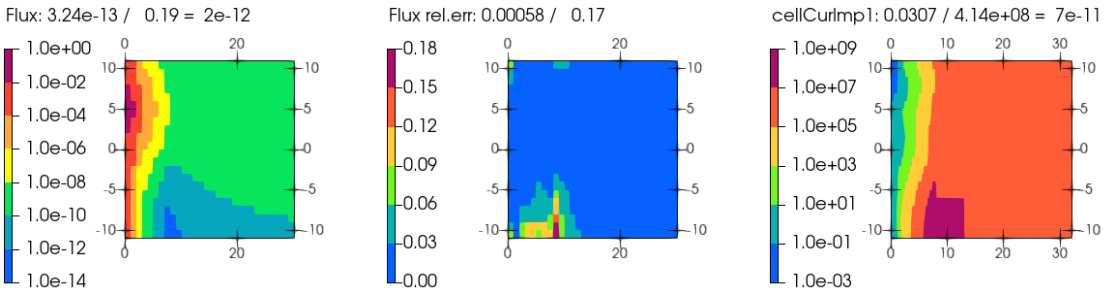

*(g) Iteration 6, GDF=0.1*



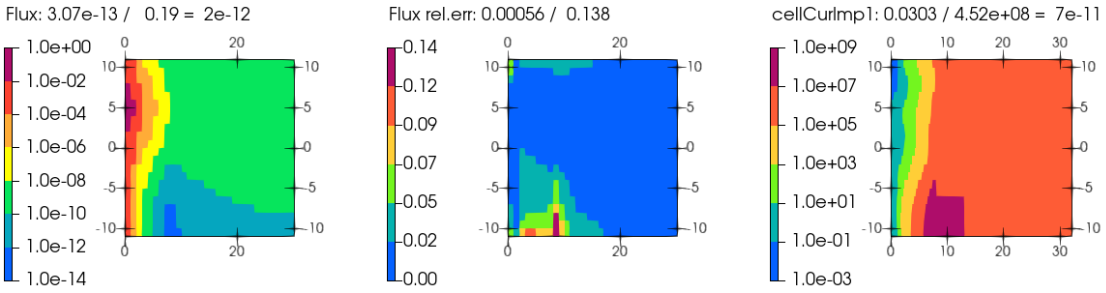

*(h) Iteration 7, GDF=0.1*



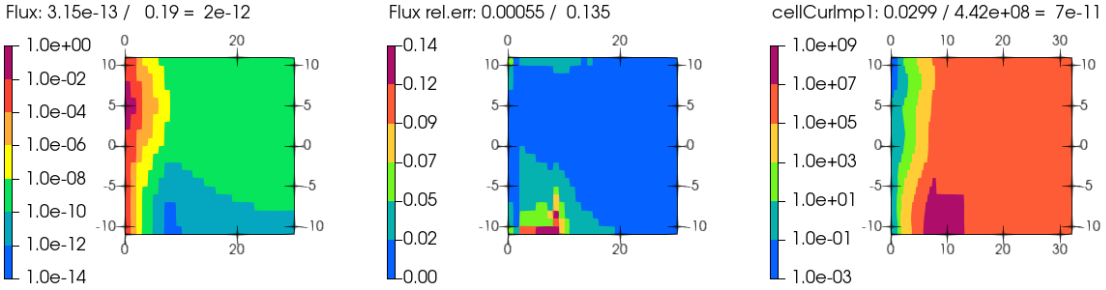
*(i) Iteration 8, GDF=0.2*



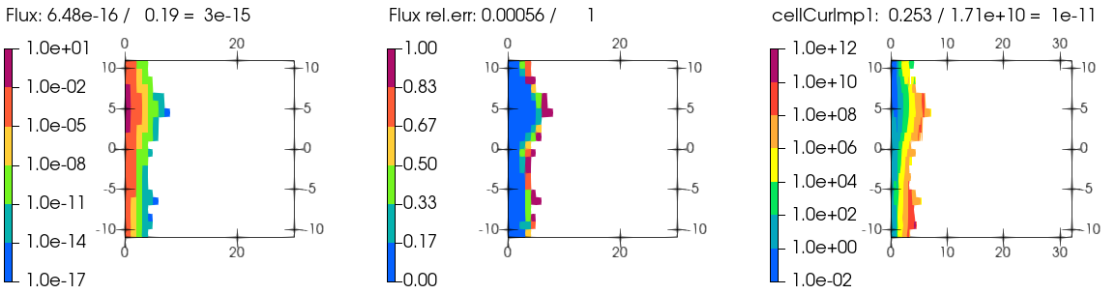

*(j) Iteration 9, GDF=0.2*



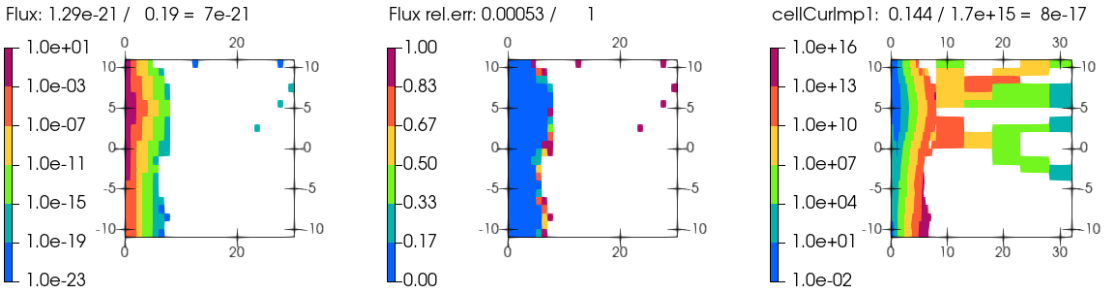

*(k) Iteration 10, GDF=0.2*



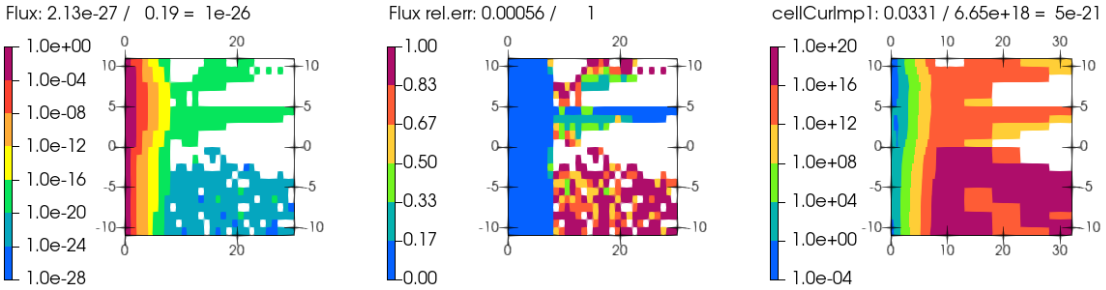

*(l) Iteration 11, GDF=0.2*



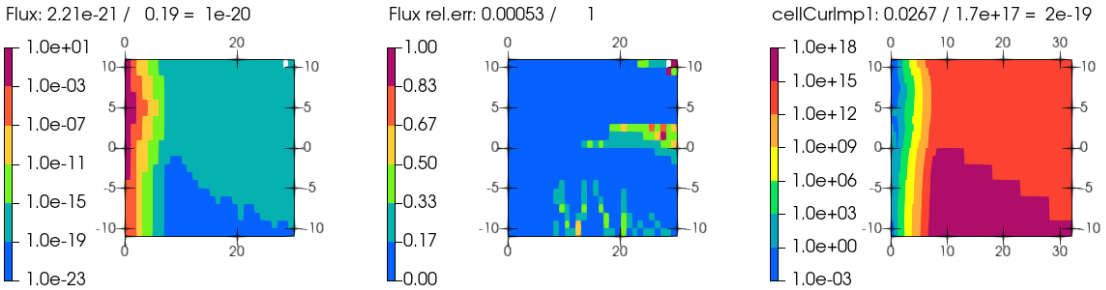

*(m) Iteration 12, GDF=0.5*



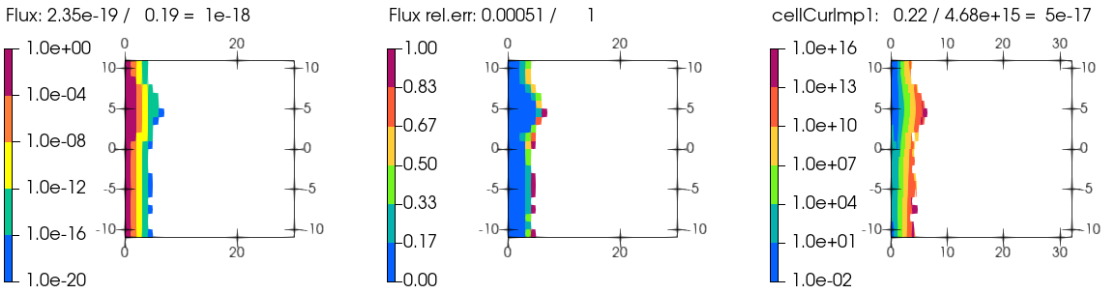

*(n) Iteration 13, GDF=0.5*



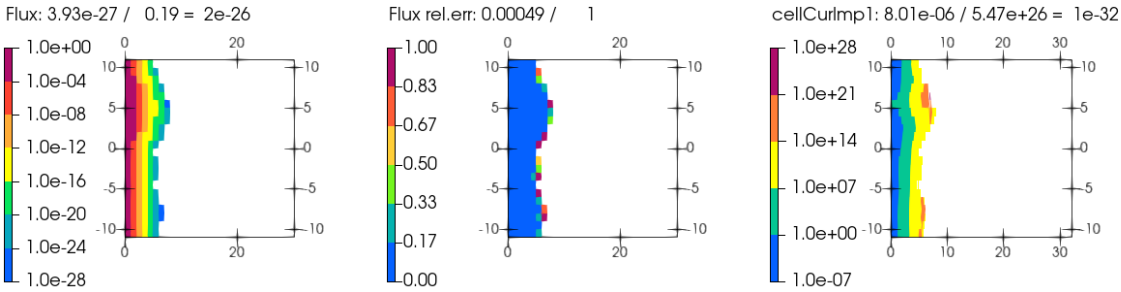

*(o) Iteration 14, GDF=0.5*



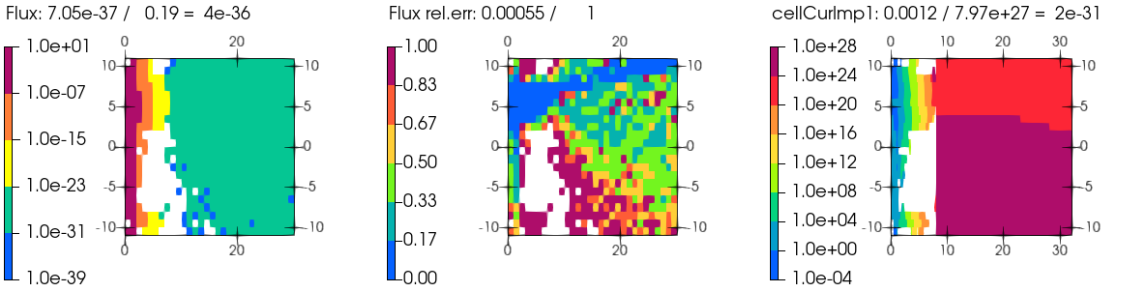

*(p) Iteration 15, GDF=0.5*



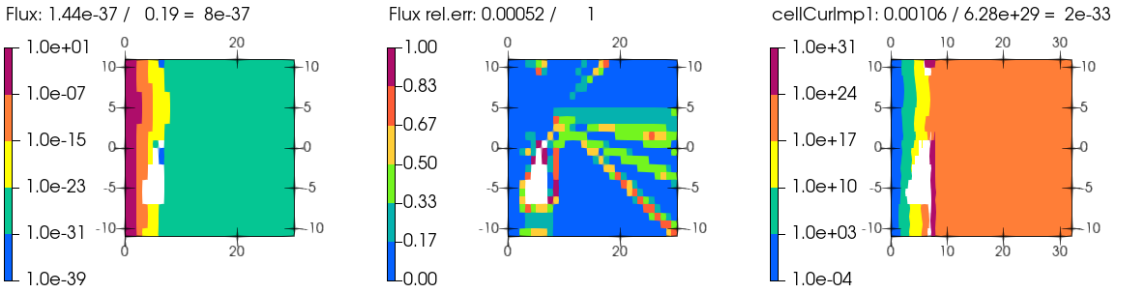

*(q) Iteration 16, GDF=0.8*



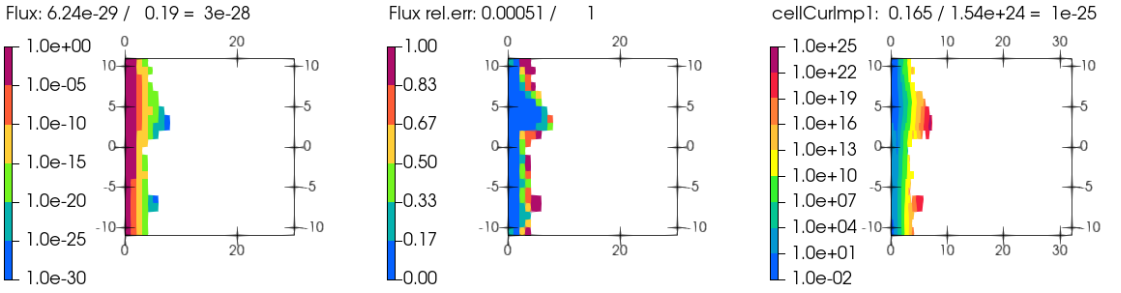

*(r) Iteration 17, GDF=0.8*



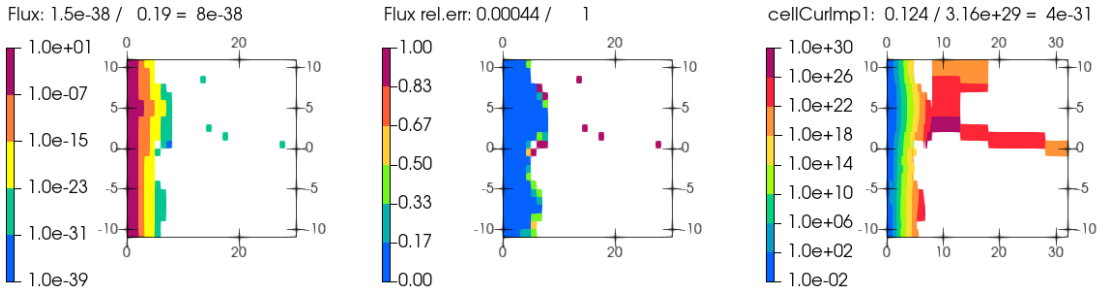

*(s) Iteration 18, GDF=0.8*



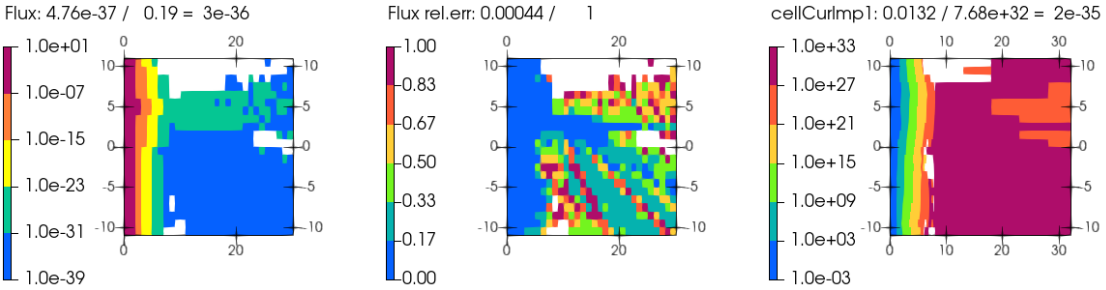

*(t) Iteration 19, GDF=0.8*



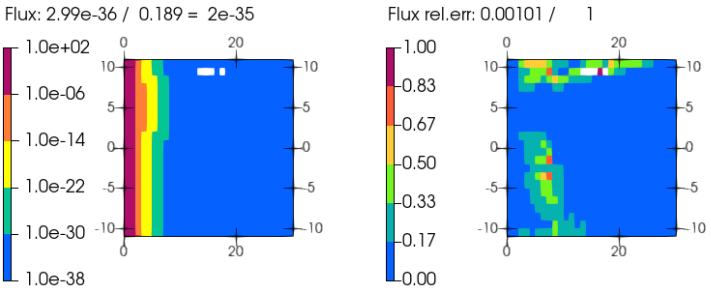

*(u) Iteration 20, GDF=1.0*




*(v) Iteration 21, GDF=1.0*




*(w) Iteration 22, GDF=1.0*



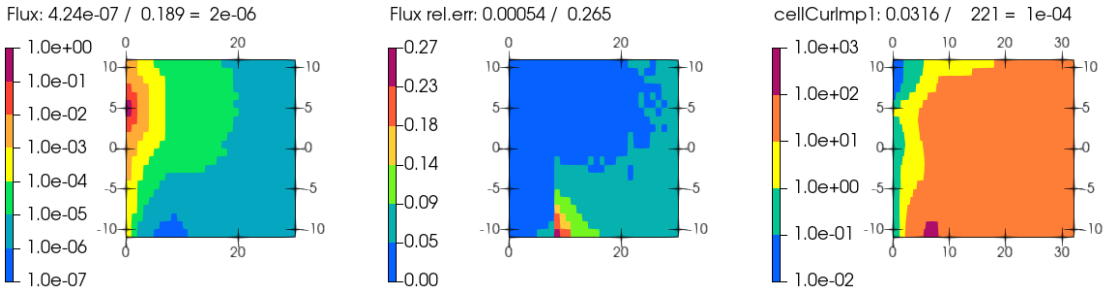

*(x) Iteration 23, GDF=1.0*

**Fig. C.3** Calculation 3: the same as calculation 2, but as single run

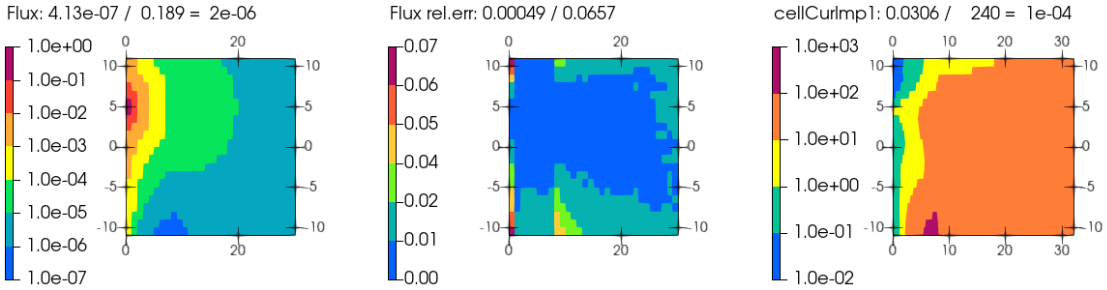
## C.4 Calculation 4

**Listing C.4** `wwin` and `wwgen` cards for calculation 4

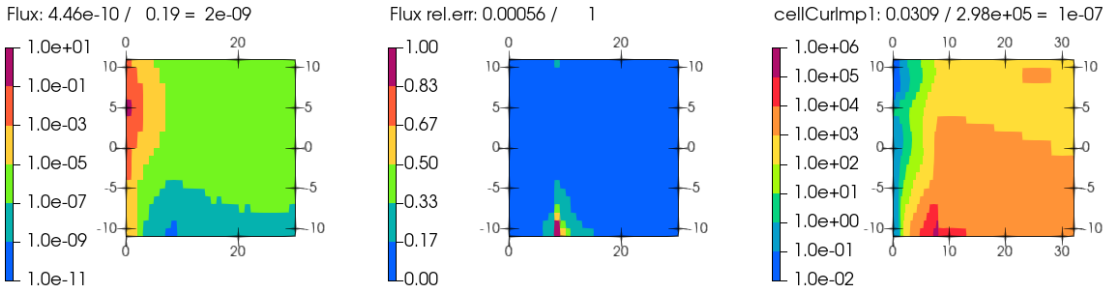
```
% Generate weight window for global VR
wwgen wwGlobal2
    1e-9 10000
    3 -1
    8 44 1 22
    0 0.5 1.0
    1.2 1.4 1.6 1.8 2.0
    2.2 2.4 2.6 2.8 3.0
    3.2 3.4 3.6 3.8 4.0
    4.2 4.4 4.6 4.8 5.0
    5.2 5.4 5.6 5.8 6.0
    6.2 6.4 6.6 6.8 7.0
    7.2 7.4 7.6 7.8 8.0
    13 14 16 18 23 28 32
    0 360
    -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11

% Global VR iterations
wwin wwGlobal
    wi 1 22
    wwGlobal2 5e-2
    wwGlobal2 5e-2
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
```



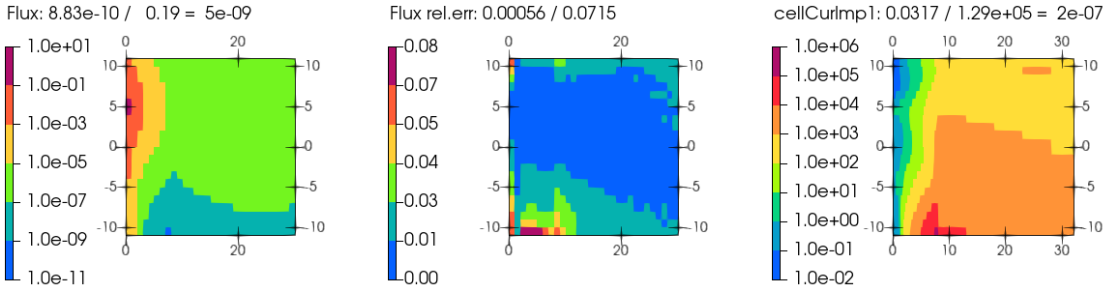

*(a) Iteration 0, GDF=0.05*



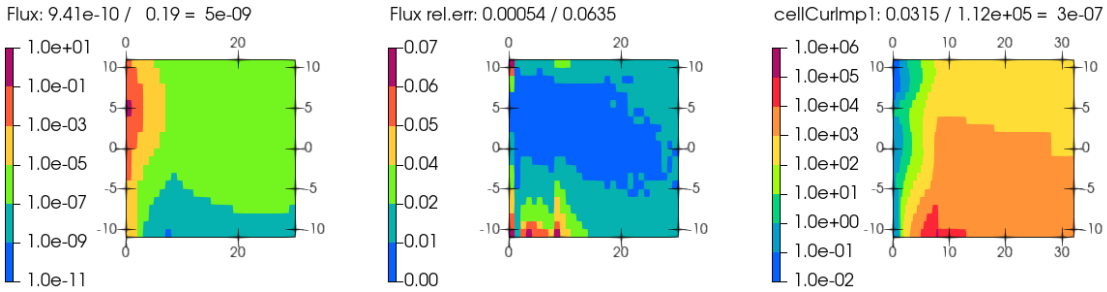

*(b) Iteration 1, GDF=0.05*



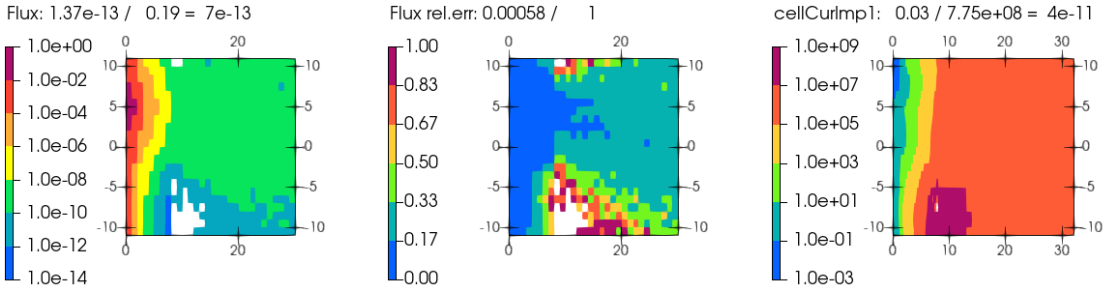

*(c) Iteration 2, GDF=0.1*



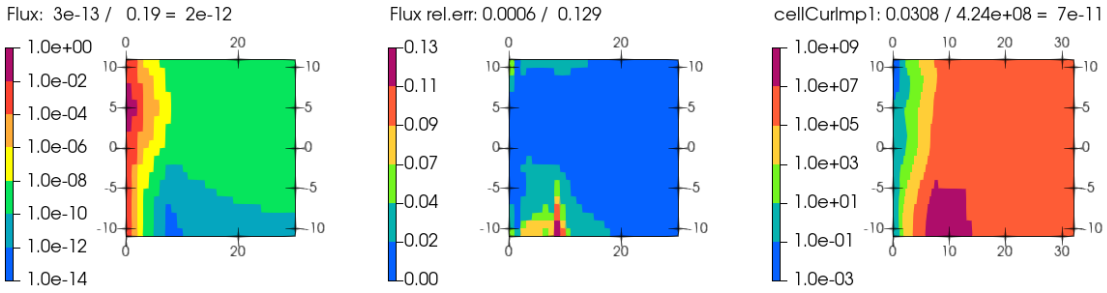

*(d) Iteration 3, GDF=0.1*



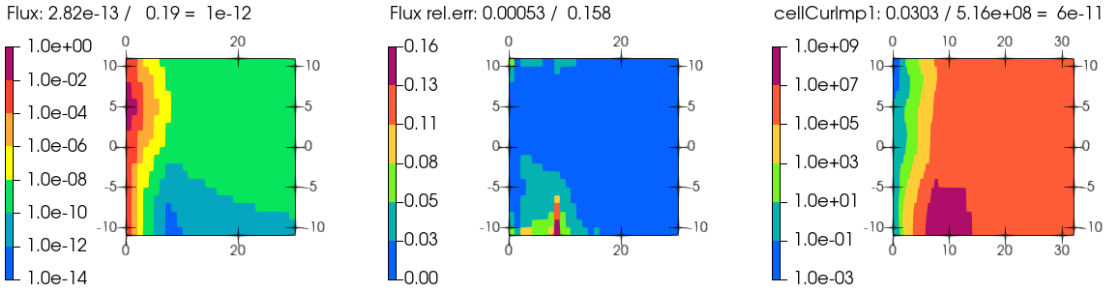

*(e) Iteration 4, GDF=0.1*



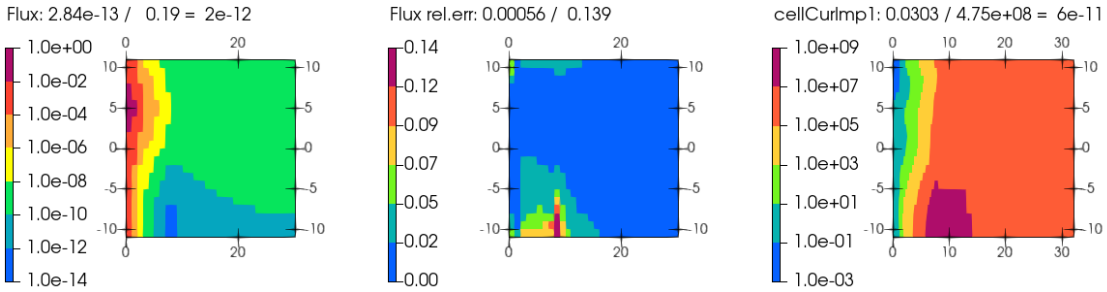

*(f) Iteration 5, GDF=0.2*



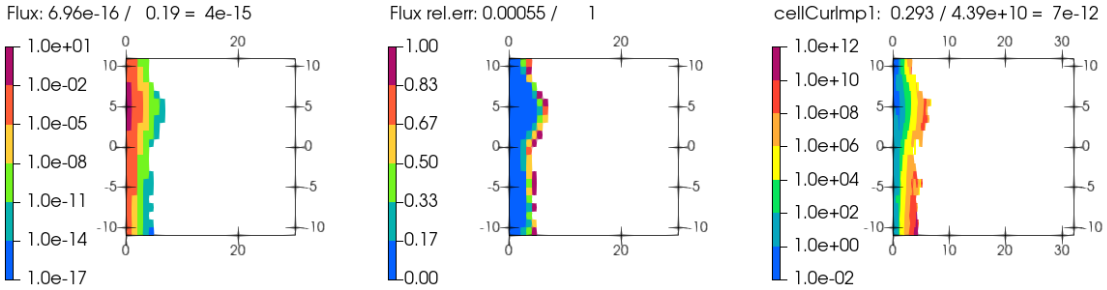

*(g) Iteration 6, GDF=0.2*



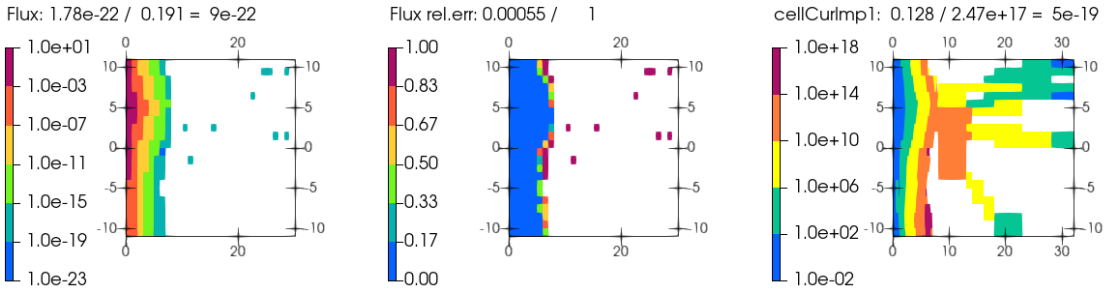

*(h) Iteration 7, GDF=0.2*



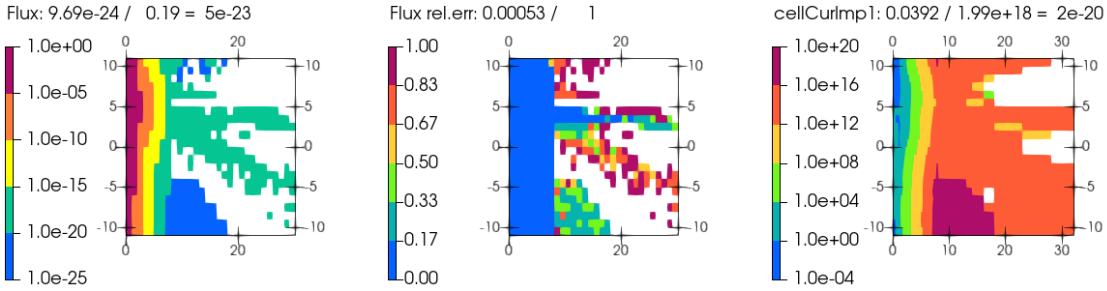

*(i) Iteration 8, GDF=0.2*



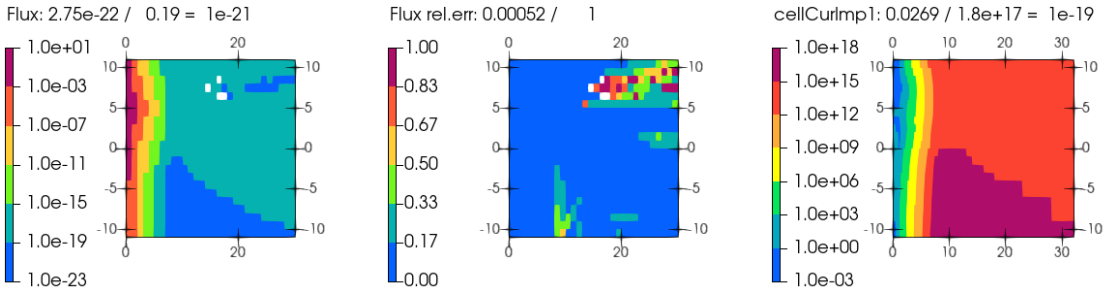

*(j) Iteration 9, GDF=0.5*



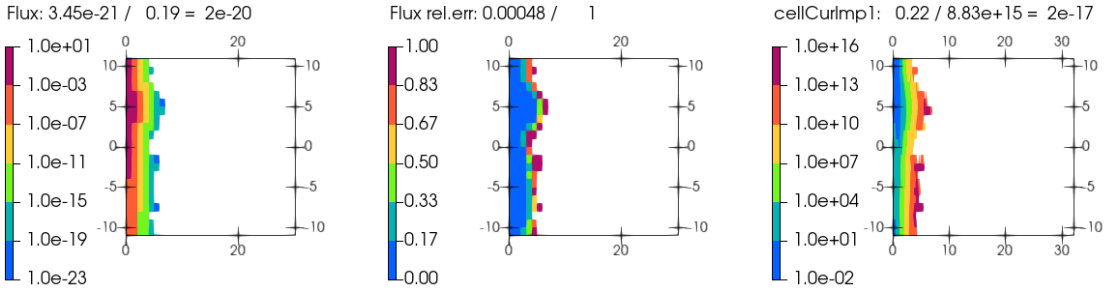

*(k) Iteration 10, GDF=0.5*



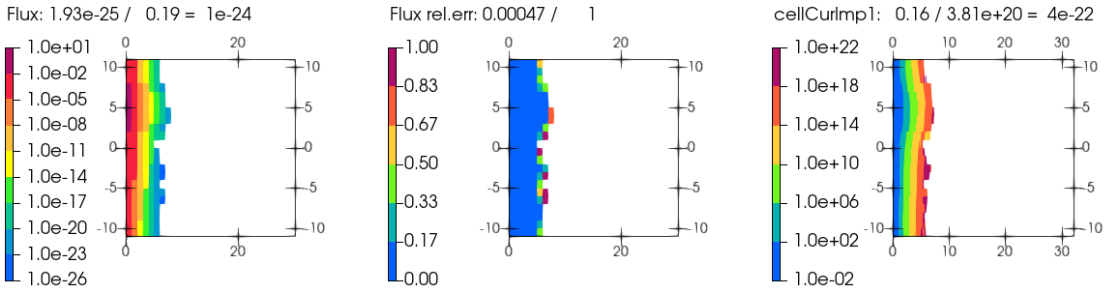

*(l) Iteration 11, GDF=0.5*



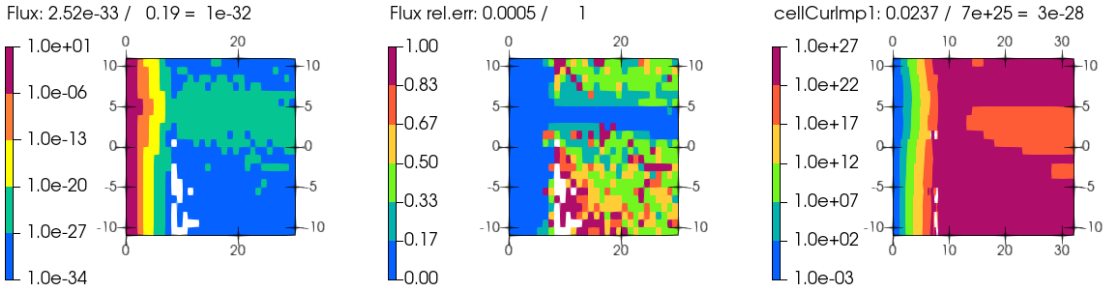

*(m) Iteration 12, GDF=0.5*



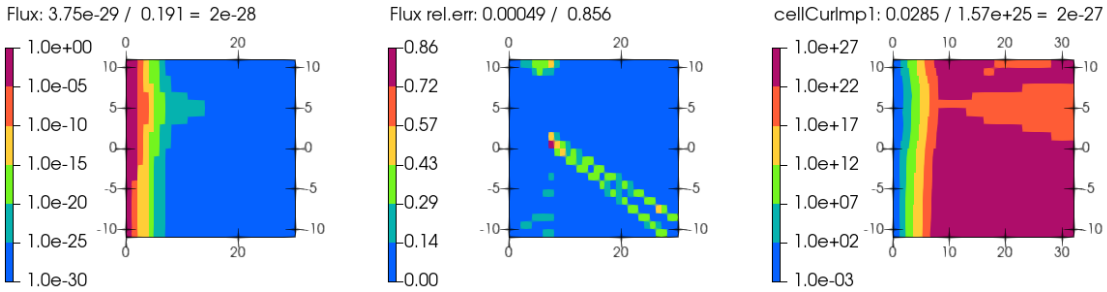

*(n) Iteration 13, GDF=0.8*



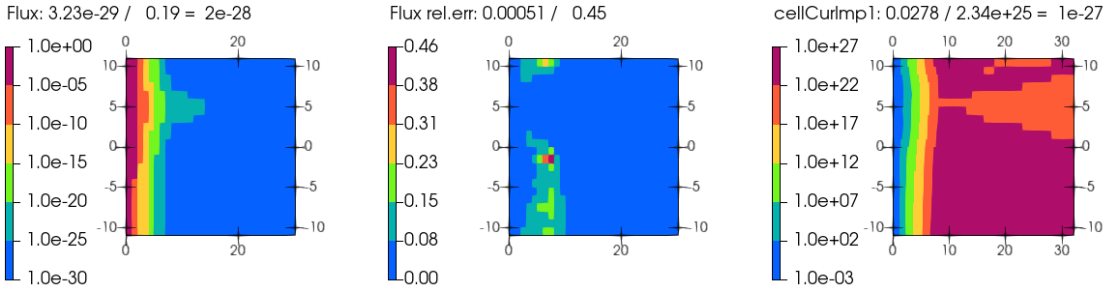

*(o) Iteration 14, GDF=0.8*



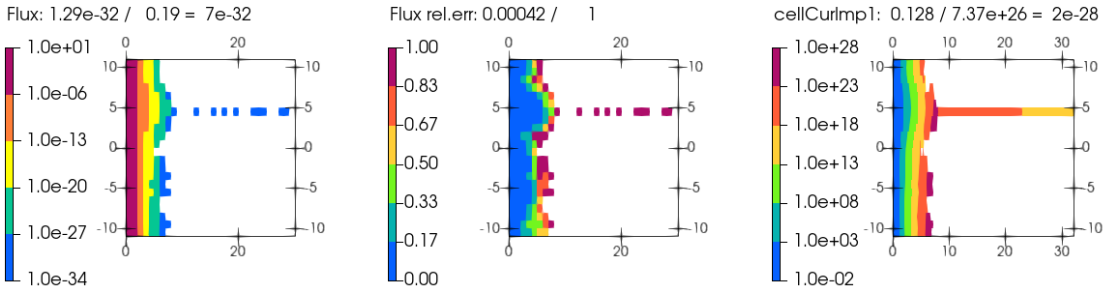

*(p) Iteration 15, GDF=0.8*



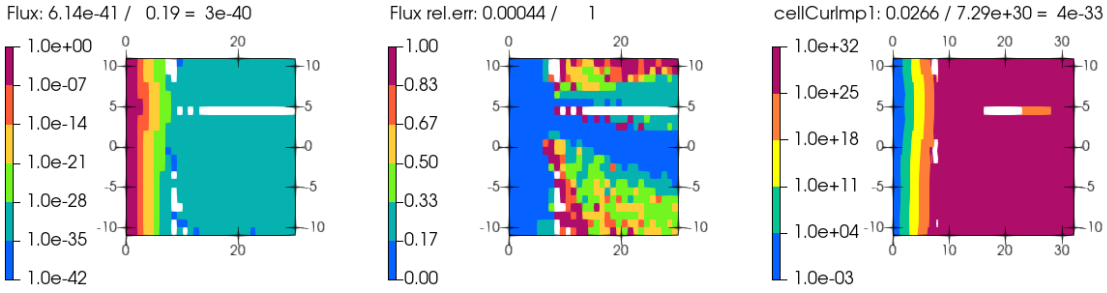

*(q) Iteration 16, GDF=0.8*



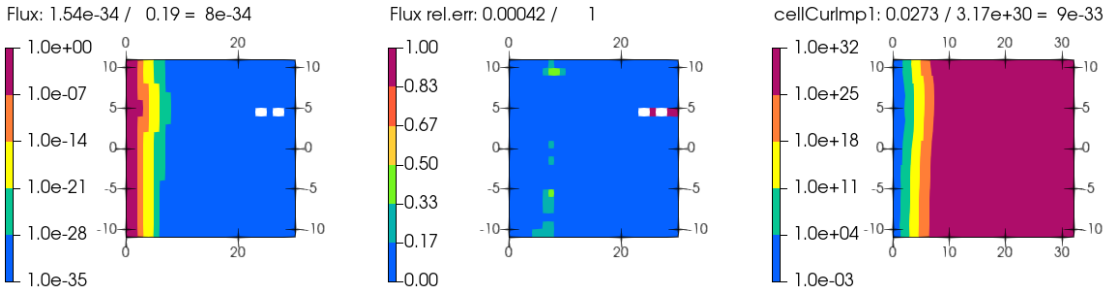

*(r) Iteration 17, GDF=0.8*



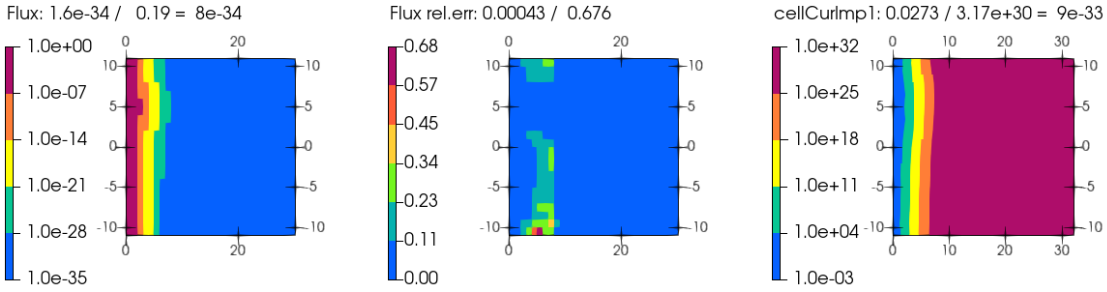

*(s) Iteration 18, GDF=1.0*




*(t) Iteration 19, GDF=1.0*




*(u) Iteration 20, GDF=1.0*



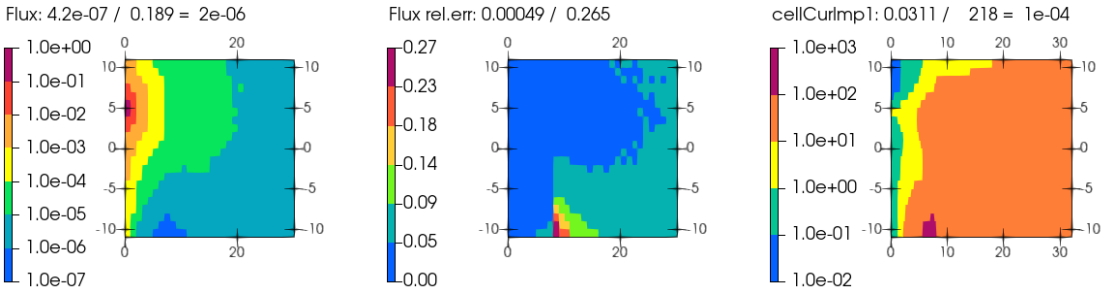

*(v) Iteration 21, GDF=1.0*

**Fig. C.4** Calculation 4: additional radial mesh elements

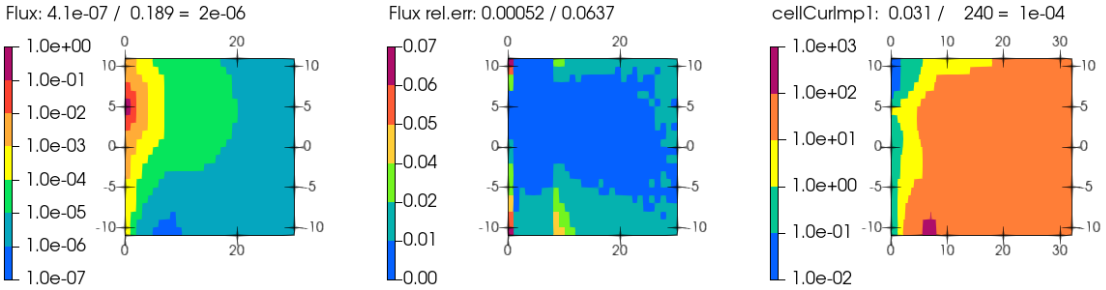
## C.5 Calculation 5

**Listing C.5** `wwin` and `wwgen` cards for calculation 5

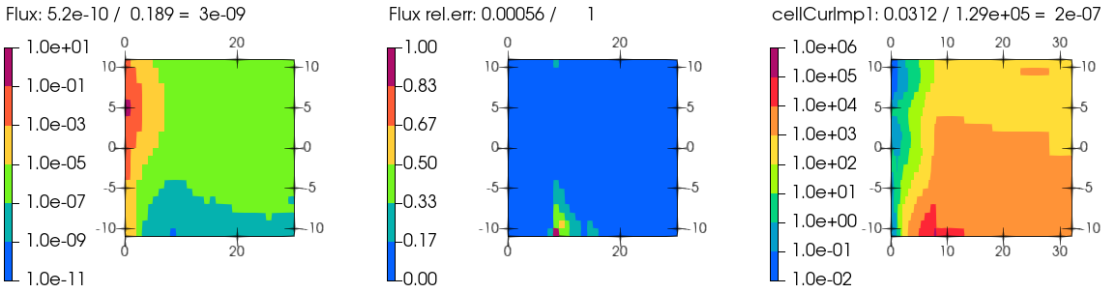
```
% Generate weight window for global VR
wwgen wwGlobal2
    1e-9 10000
    3 -1
    8 30 1 22
    0 0.5 1.0
    1.2 1.6 2.0
    2.2 2.6 3.0
    3.2 3.6 4.0
    4.2 4.6 5.0
    5.2 5.6 6.0
    6.2 6.6 7.0
    7.2 7.6 8.0
    13 14 16 18 23 28 32
    0 360
    -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11

% Global VR iterations
wwin wwGlobal
    wi 1 22
    wwGlobal2 5e-2
    wwGlobal2 5e-2
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
```



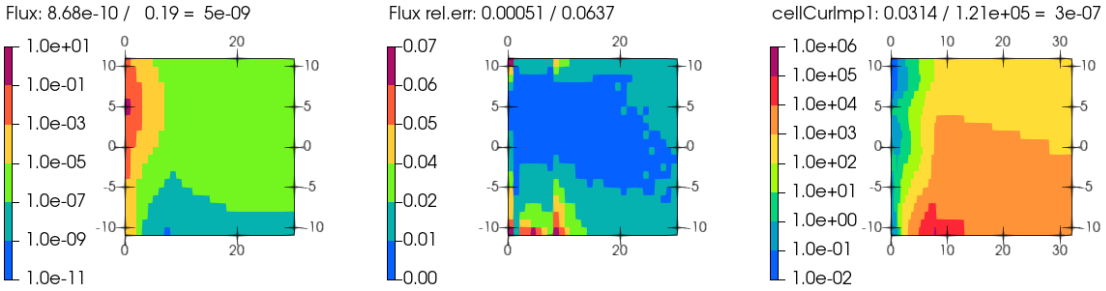

*(a) Iteration 0, GDF=0.05*



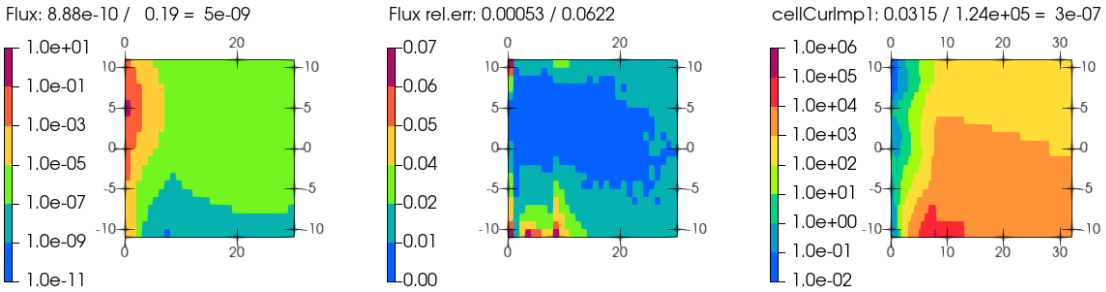

*(b) Iteration 1, GDF=0.05*



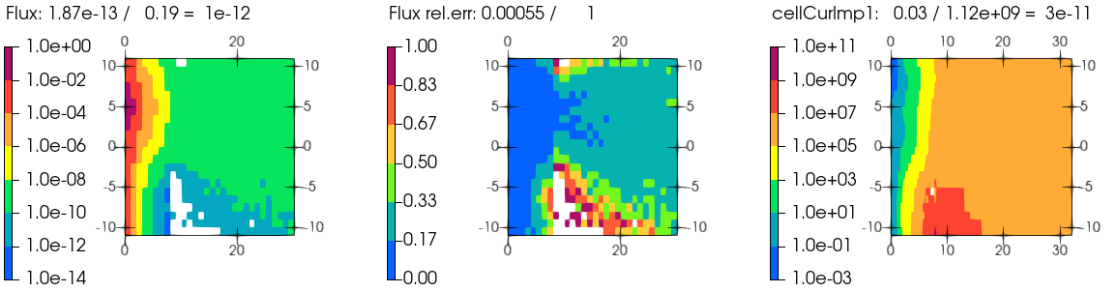

*(c) Iteration 2, GDF=0.1*



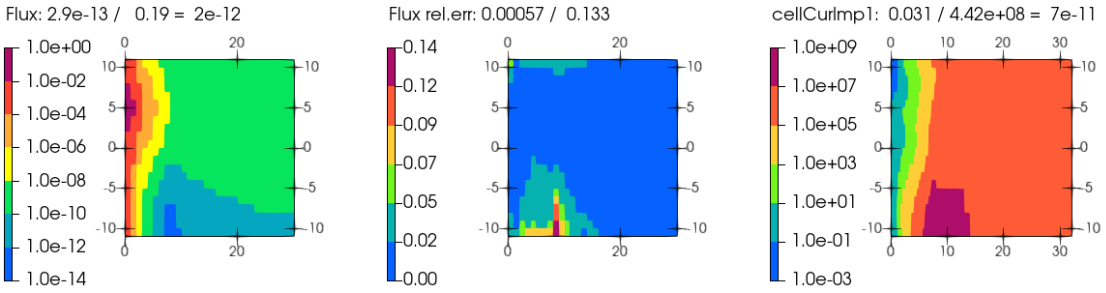

*(d) Iteration 3, GDF=0.1*



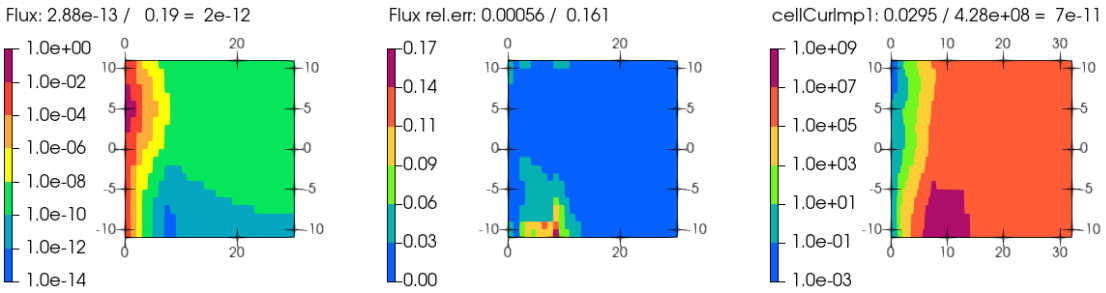

*(e) Iteration 4, GDF=0.1*



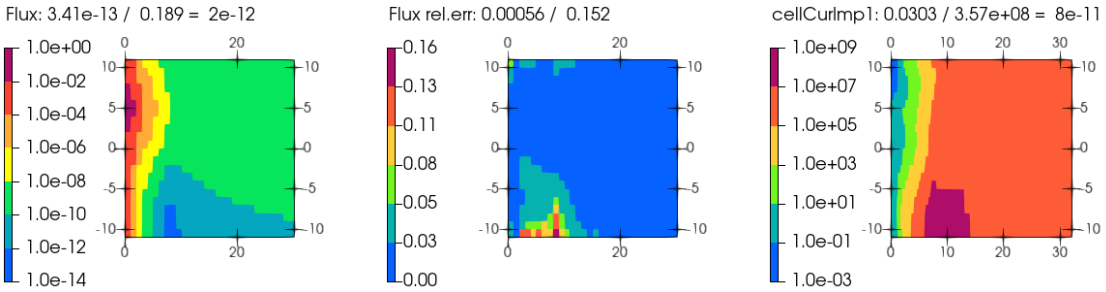

*(f) Iteration 5, GDF=0.2*



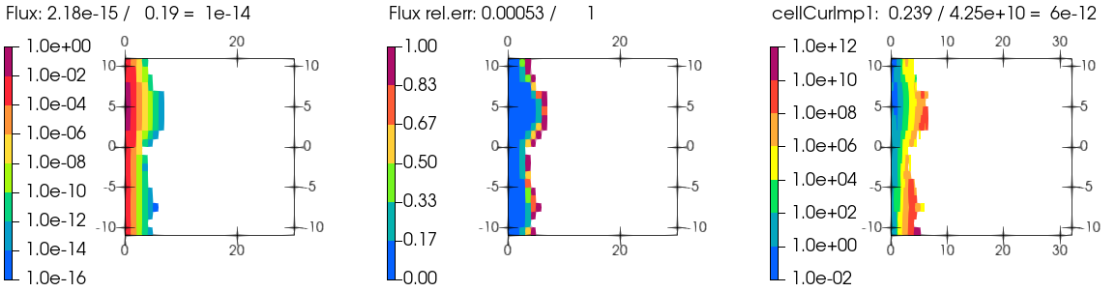

*(g) Iteration 6, GDF=0.2*



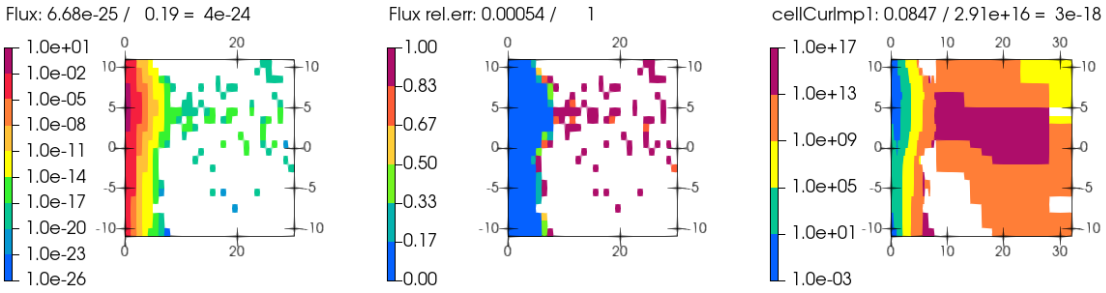

*(h) Iteration 7, GDF=0.2*



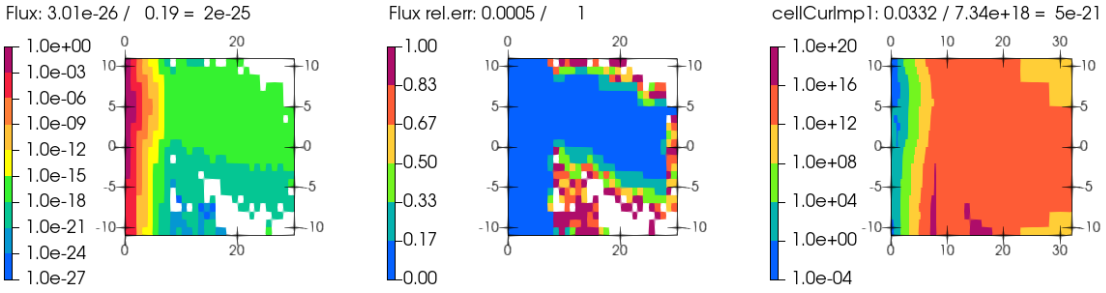

*(i) Iteration 8, GDF=0.2*



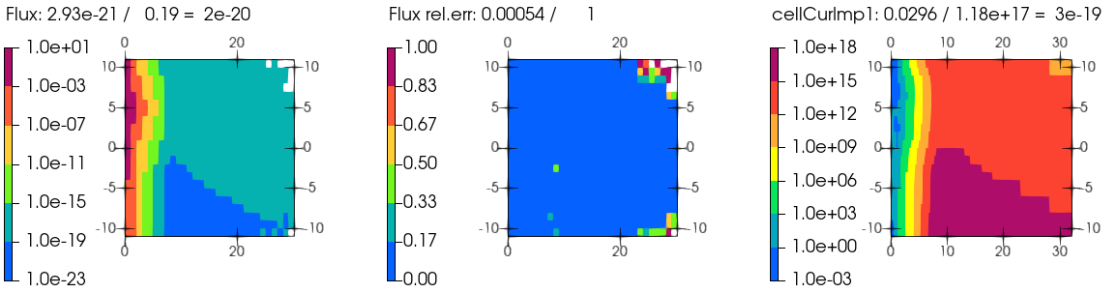

*(j) Iteration 9, GDF=0.5*



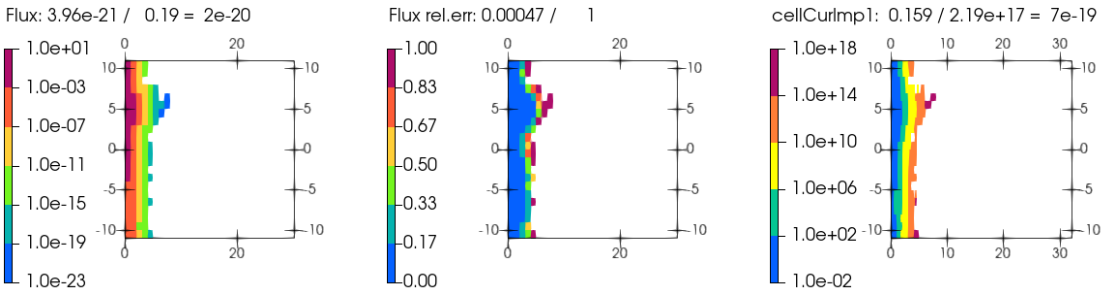

*(k) Iteration 10, GDF=0.5*



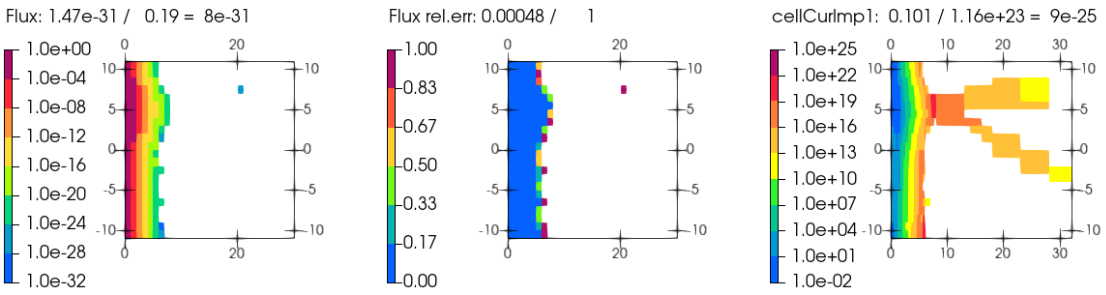

*(l) Iteration 11, GDF=0.5*



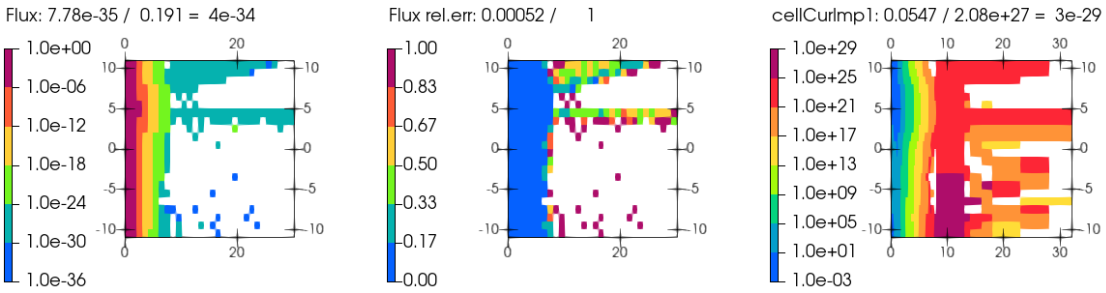
*(m) Iteration 12, GDF=0.5*



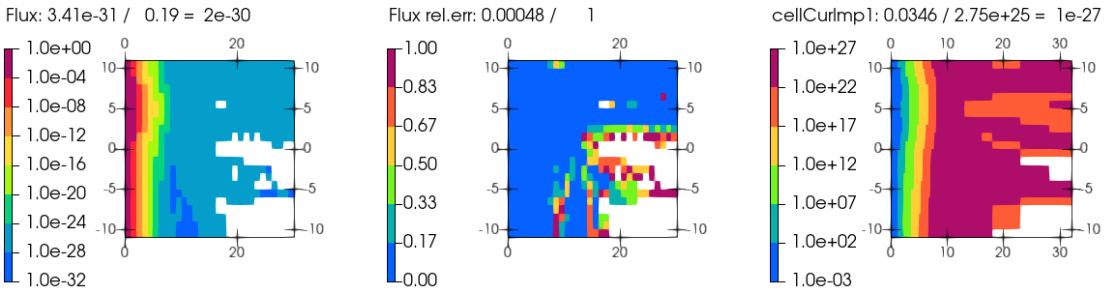

*(n) Iteration 13, GDF=0.8*



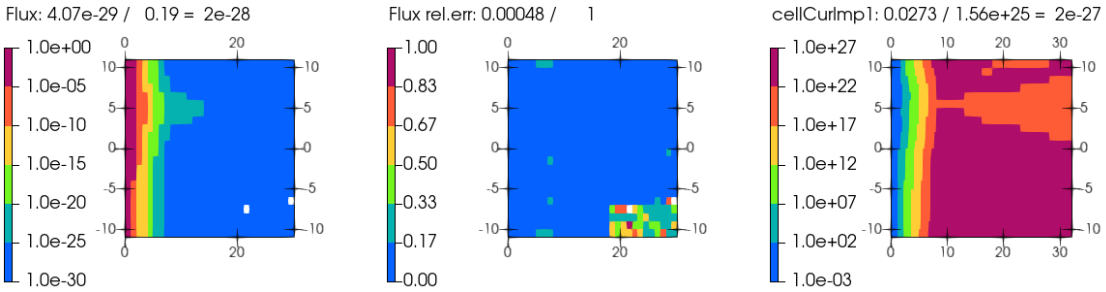

*(o) Iteration 14, GDF=0.8*



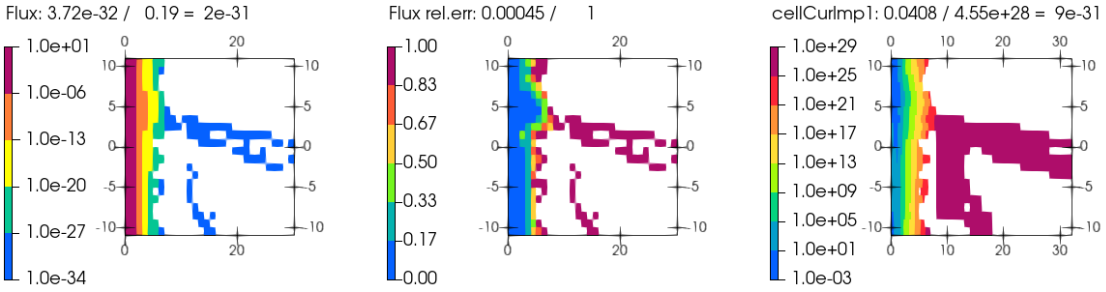

*(p) Iteration 15, GDF=0.8*



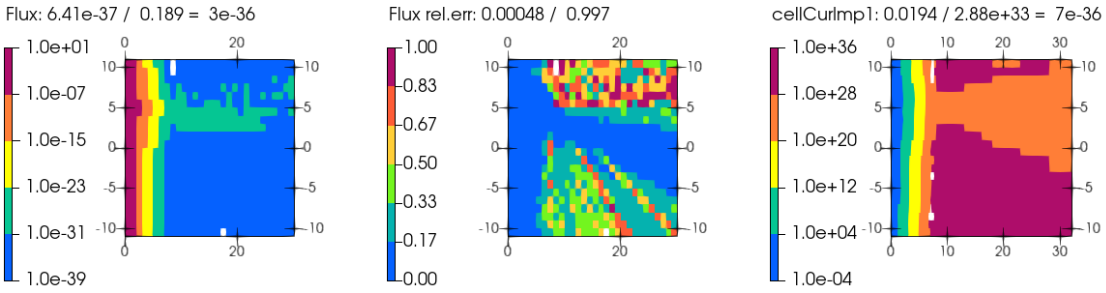

*(q) Iteration 16, GDF=0.8*



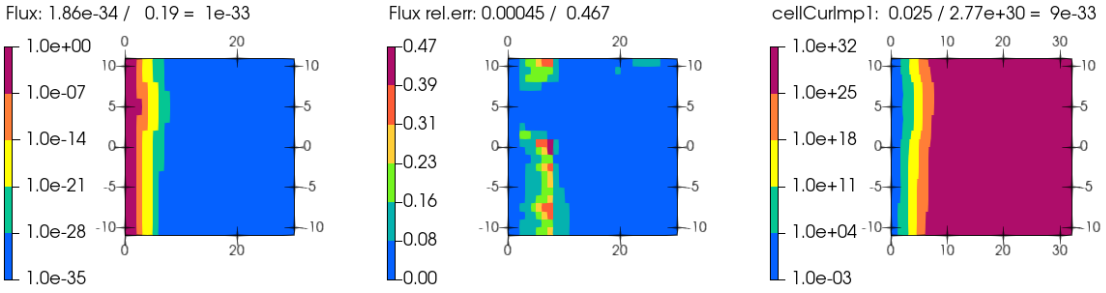

*(r) Iteration 17, GDF=0.8*



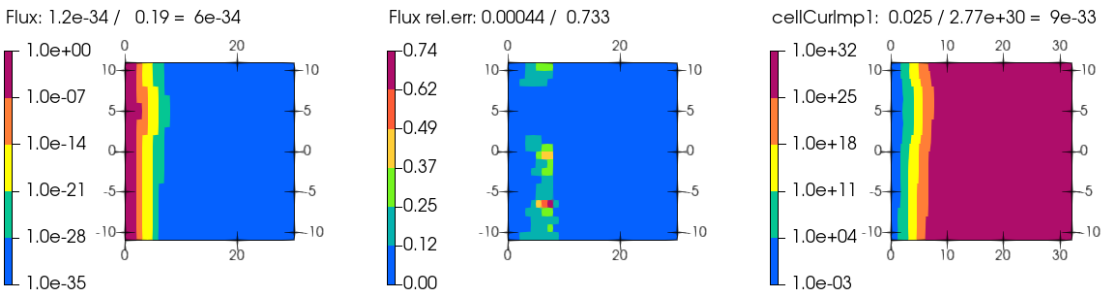

*(s) Iteration 18, GDF=1.0*




*(t) Iteration 19, GDF=1.0*




*(u) Iteration 20, GDF=1.0*



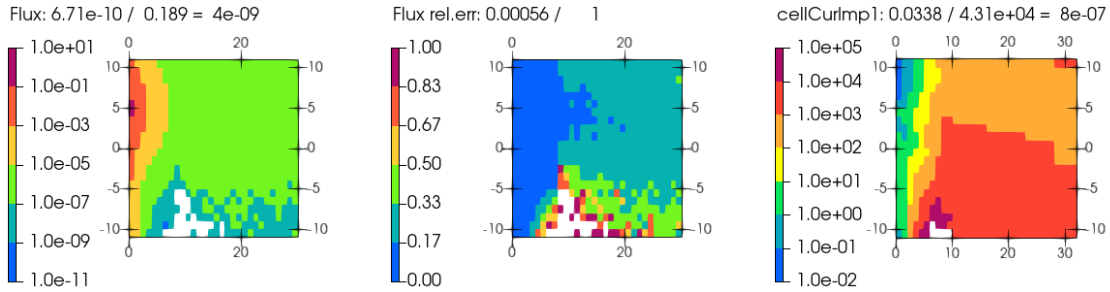

*(v) Iteration 21, GDF=1.0*

**Fig. C.5** Calculation 5: less radial mesh elements in central channel

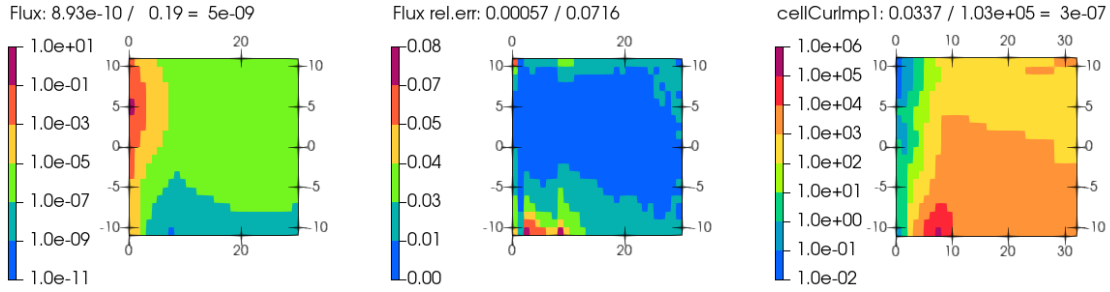
## C.6 Calculation 6

**Listing C.6** `wwin` and `wwgen` cards for calculation 6

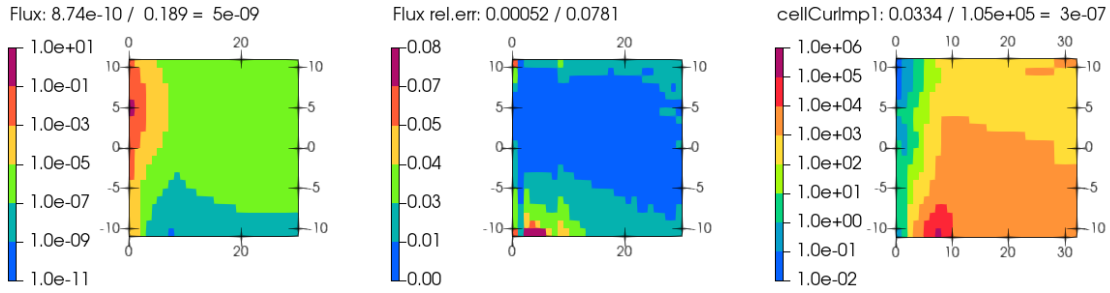
```
% Generate weight window for global VR
wwgen wwGlobal2
    1e-9 10000
    3 -1
    8 16 1 22
    0 1.0
      2.0
      3.0
      4.0
      5.0
      6.0
      7.0
      8.0
    9.0 10.0 13 16 18 23 28 32
    0 360
    -11.1 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1 0
    1 2 3 4 5 6 7 8 9 10 11.1

% Global VR iterations
wwin wwGlobal
    wi 1 20
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 1e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 2e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 5e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 8e-1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1
    wwGlobal2 1

set maxsplit 100 1e-18
```



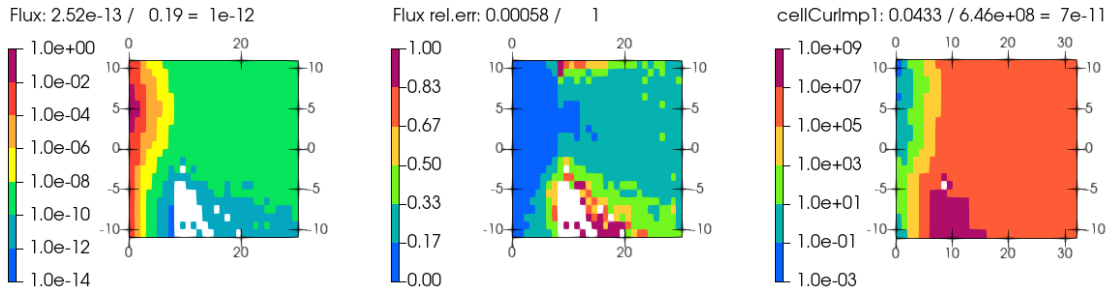

*(a) Iteration 0, GDF=0.1*



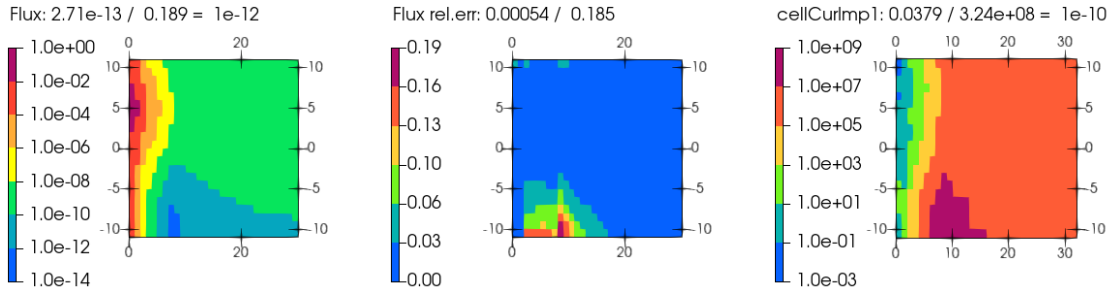

*(b) Iteration 1, GDF=0.1*



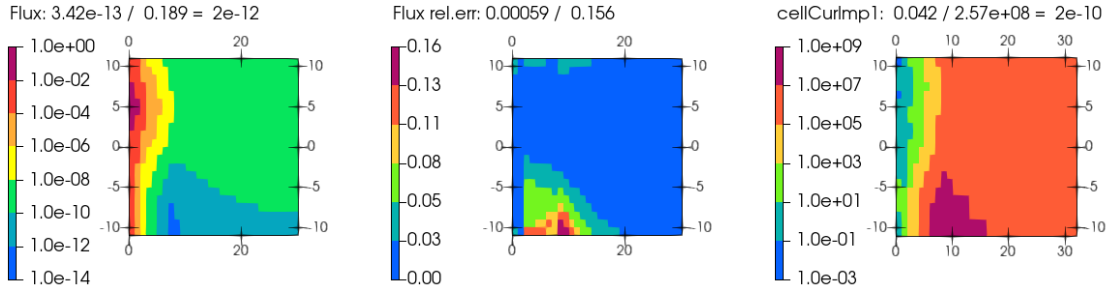

*(c) Iteration 2, GDF=0.1*



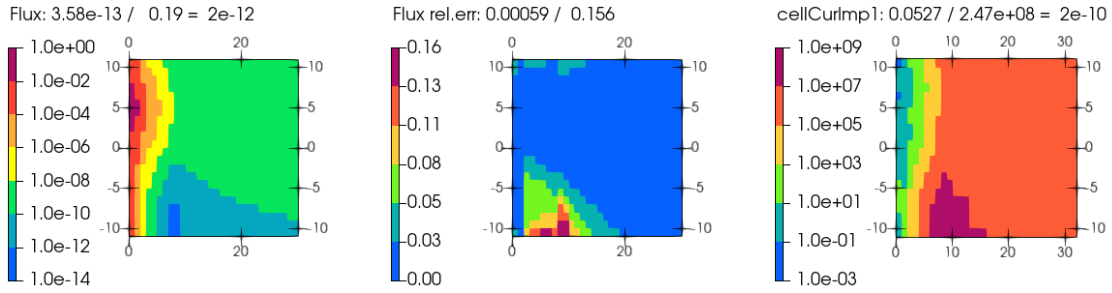

*(d) Iteration 3, GDF=0.2*



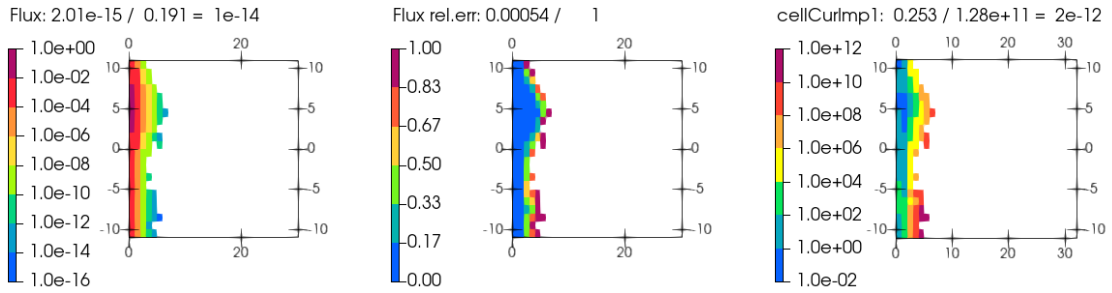

*(e) Iteration 4, GDF=0.2*



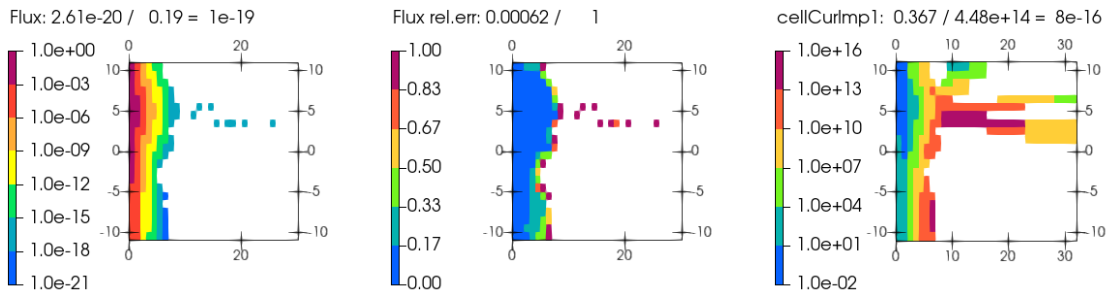

*(f) Iteration 5, GDF=0.2*



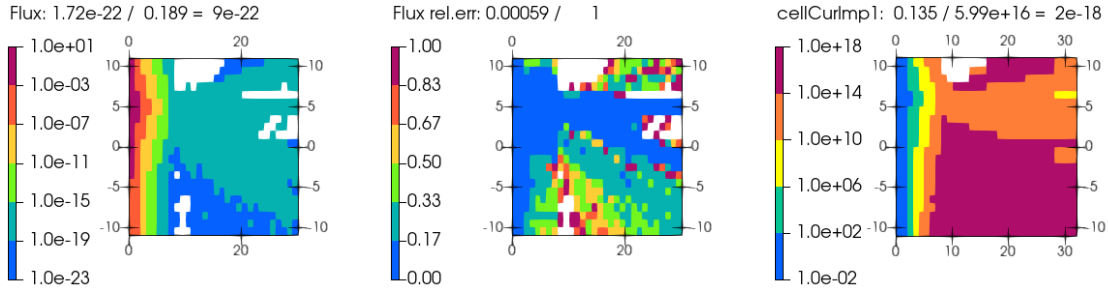

*(g) Iteration 6, GDF=0.2*



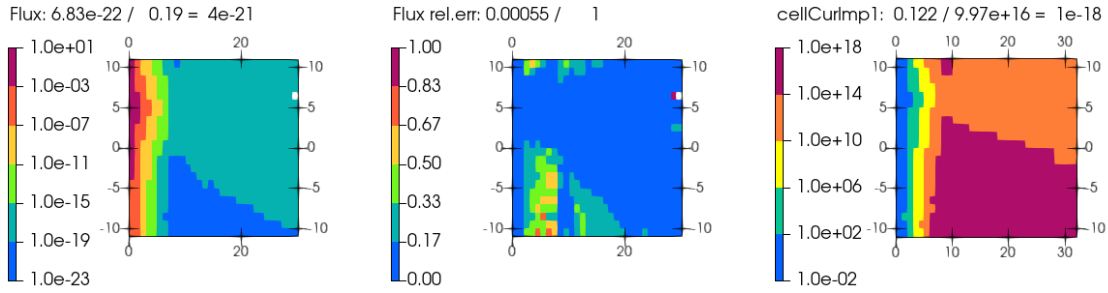

*(h) Iteration 7, GDF=0.5*



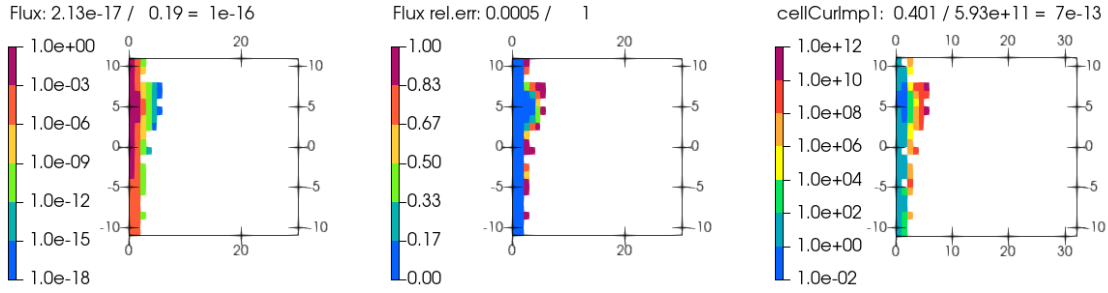

*(i) Iteration 8, GDF=0.5*



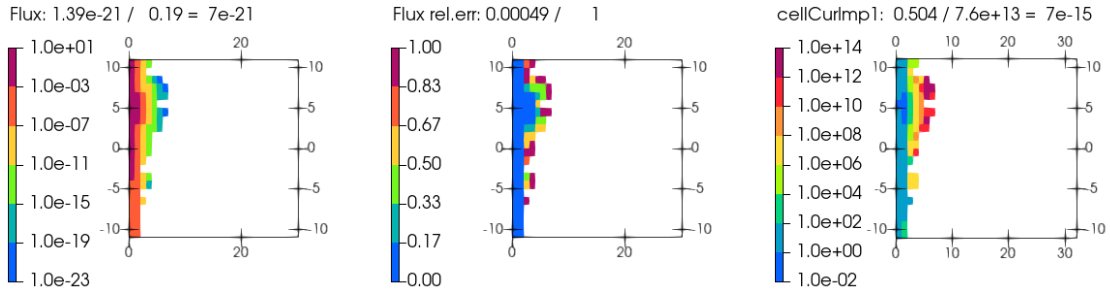

*(j) Iteration 9, GDF=0.5*



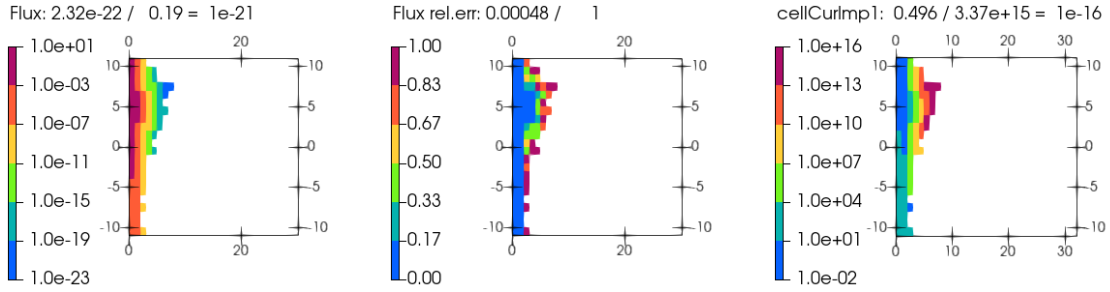

*(k) Iteration 10, GDF=0.5*



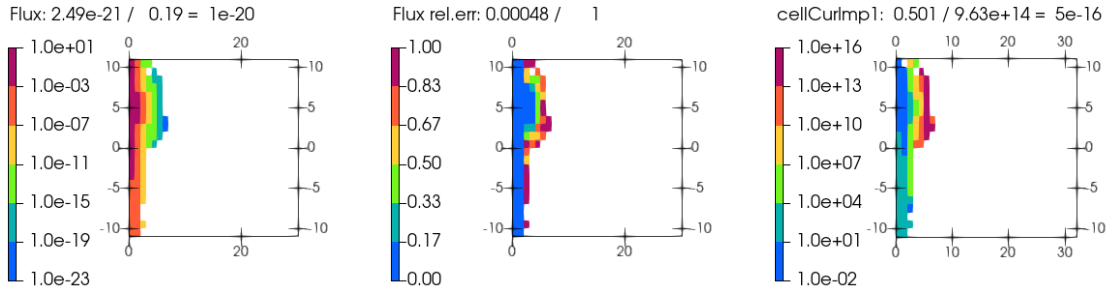

*(l) Iteration 11, GDF=0.8*



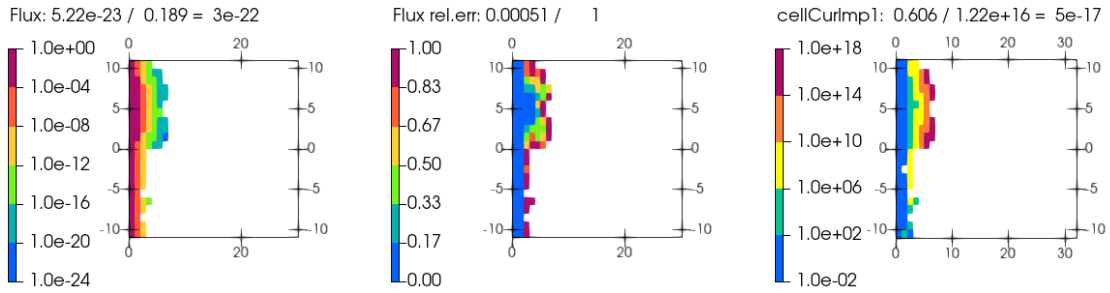

*(m) Iteration 12, GDF=0.8*



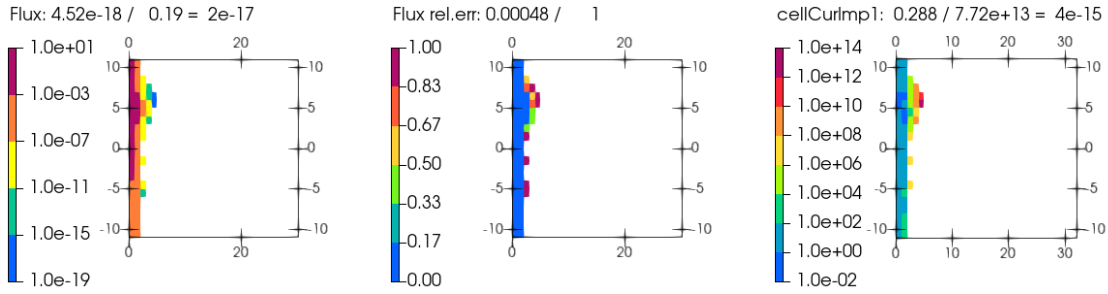

*(n) Iteration 13, GDF=0.8*



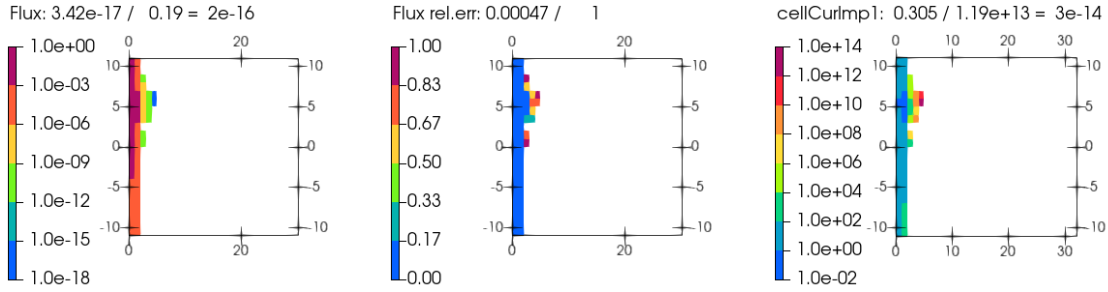

*(o) Iteration 14, GDF=0.8*



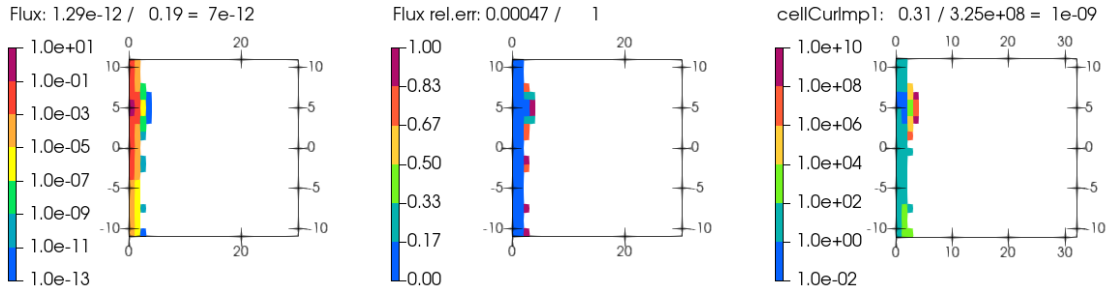

*(p) Iteration 15, GDF=0.8*



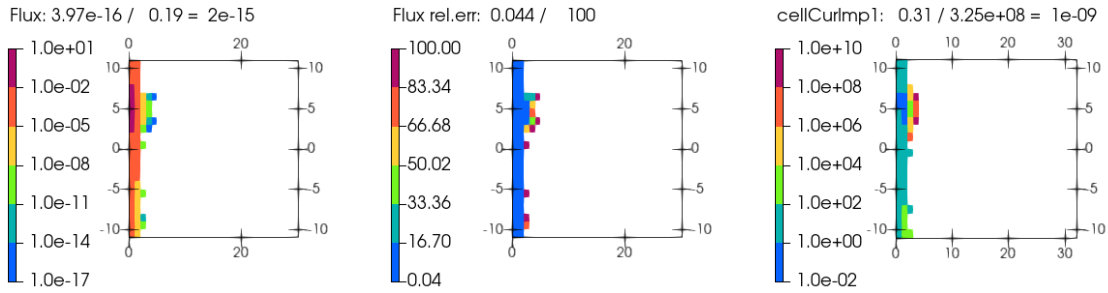

*(q) Iteration 16, GDF=1.0*




*(r) Iteration 17, GDF=1.0*




*(s) Iteration 18, GDF=1.0*




*(t) Iteration 19, GDF=1.0*

**Fig. C.6** Calculation 6: less radial mesh elements in the central channel, more elements in the absorber

**Gesellschaft für Anlagen- und Reaktorsicherheit (GRS) gGmbH**

Schwertnergasse 1
**50667 Köln**
Telefon +49 221 2068-0
Telefax +49 221 2068-888

Boltzmannstraße 14
**85748 Garching b. München**
Telefon +49 89 32004-0
Telefax +49 89 32004-300

Kurfürstendamm 200
**10719 Berlin**
Telefon +49 30 88589-0
Telefax +49 30 88589-111

Theodor-Heuss-Straße 4
**38122 Braunschweig**
Telefon +49 531 8012-0
Telefax +49 531 8012-200

www.grs.de

**ISBN 978-3-911727-35-8**