

## Enhancing the Performance of Fission Source Convergence with Functional Expansion Tallies in Serpent 2 Monte Carlo Code

Jambrina Ana, Leppänen Jaakko, Suikkanen Heikki

This is a Publisher's version version of a publication  
published by PHYSOR2020, Nuclear Energy Group UCAM  
in International Conference on Physics of Reactors 2020 (PHYSOR2020)

DOI:

**Copyright of the original publication:** © PHYSOR2020, Nuclear Energy Group UCAM

**Please cite the publication as follows:**

Jambrina, A., Leppänen, J., Suikkanen, H. (2020). Enhancing the Performance of Fission Source Convergence with Functional Expansion Tallies in Serpent 2 Monte Carlo Code. International Conference on Physics of Reactors 2020 (PHYSOR2020).

**This is a parallel published version of an original publication.  
This version can differ from the original published article.**

## **ENHANCING THE PERFORMANCE OF FISSION SOURCE CONVERGENCE WITH FUNCTIONAL EXPANSION TALLIES IN SERPENT 2 MONTE CARLO CODE**

**Ana Jambrina<sup>1\*</sup>, Jaakko Leppänen<sup>2</sup>, and Heikki Suikkanen<sup>1</sup>**

<sup>1</sup>LUT University

P.O. Box 20, FI-53851 Lappeenranta, Finland

<sup>2</sup>VTT Technical Research Centre of Finland, Ltd.

P.O. Box 1000, FI-02044 VTT, Finland

Ana.Jambrina@lut.fi, Jaakko.Leppanen@vtt.fi, Heikki.Suikkanen@lut.fi

### **ABSTRACT**

This paper presents an upgrade to the built-in response matrix based solver implemented in Serpent 2 Monte Carlo code aiming to improve the fission source convergence when obtaining the forward solution to the  $k$ -eigenvalue criticality source problems. The functional expansion tallies are introduced in an attempt to improve the accuracy of the cell-wise form factors that feed the response matrix solver, replacing the current mesh-based approach. The functional expansion tallies reconstruct the binning surface and collision tallies, by using high-order series expansion to represent the original and continuous spatial distributions. This new feature is implemented to Serpent 2 and tested by single-assembly and full-core PWR calculations (BEAVRS benchmark). The results show enhanced performance of the convergence acceleration methodology based on an improved initial guess of the fission source.

KEYWORDS: Monte Carlo, source convergence, acceleration, Serpent 2, functional expansion tallies

### **1. INTRODUCTION**

Regardless the continuous advances in high performance computing, Monte Carlo methods nevertheless are expensive, not only for large systems, also, for industry basis design-analysis. The goal is to develop strategies that reduce the execution time without sacrificing accuracy.

Monte Carlo neutron transport codes solve  $k$ -eigenvalue problems from an unknown fission source. Simulations begin with a guess of the fission neutron source distribution; and, through a series of so-called inactive cycles, from which results are not collected, it is expected to achieve source convergence. The efficiency of the Monte Carlo calculations would be improved by evolving the current techniques to accelerate the convergence of the fission source, and for extension, reduce the number of inactive cycles.

---

\*Corresponding author.

Monte Carlo fission source error can be broken down into three components: the initial guess distribution, a statistical error and a systematic bias error. The statistical error of order  $\mathcal{O}(1/\sqrt{m})$ , is due to the limited number of neutron histories simulated per cycle,  $m$ . The systematic bias error of order  $\mathcal{O}(1/m)$  is caused by normalization of the source size at each cycle. Unlike the statistical error, the bias does not decay over the cycles. The error in the source sampled in the initial cycle propagates into sources in the successive cycles, decaying over cycles; although, it still represents a major part of the total error in large-scale systems and systems with high dominance ratio.

Several deterministic techniques have been applied to accelerate Monte Carlo source convergence, by reducing the number of inactive cycles used to converge the fission source: e.g., fission matrix-based methods (Kitada and Takeda [1], Carney et al. [2], Pan et al. [3], Dufek and Tuttleberg [4]), Wielandt-based methods (Yamamoto and Miyoshi [5], She et al. [6]), hybrid CMFD/Monte Carlo method (Lee et al. [7]), response matrix-based method (Leppänen [8]).

This study relays on improving the initial guess of the fission source, in an attempt to preserve the other two components of the error in the fission source, i.e. without reducing the neutron histories per cycle. The baseline of the hybrid method proposed is the response matrix-based method developed by Leppänen, [8]. Through a short Monte Carlo simulation, the built-in response matrix-based solver produces a spatial distribution that approximates the fission source to be used by the source sampling as initial guess. The spatial source distribution is reconstructed on a cell-wise form factor basis on a high-resolution mesh. A drawback of the method is that the accuracy of the spatial distribution of sampled source neutrons is limited by mesh resolution. In this work, an alternative to the cell-wise form factors is presented, which applies functional expansion tallies (FET) that are geometrically agnostic and continuous by nature.

The hybrid approach, FET-response matrix-based solver Monte Carlo method implemented in Serpent 2 Monte Carlo Code [9], is presented next, with a subsequent verification test case based on BEAVRS benchmark [10].

## 2. METHODS

The proposed methodology upgrades the built-in response matrix based solver when calculating an improved initial guess for the fission source distribution, by replacing the mesh-based approach (see Algorithm 1, [8]) used to calculate the cell-wise form factors that feed the response matrix solver, by a functional expansion reconstruction (see Algorithm 2) which through a high-order series represents the original and continuous spatial distributions of the form factors.

The reconstructed distribution for each cell  $n$  is calculated once the outer iteration cycle has been converged, by multiplying the source terms and inward currents with the corresponding form factors.

The cell-wise form factors for inward currents (Algorithm 1: line 6) and sources (Algorithm 1: line 7) are defined similar to the response coefficients (Algorithm 3: line 25, and Algorithm 3: line 26), respectively, with additional binning to an intra-cell mesh,  $h$ . The functional expansion form factors for inward currents (Algorithm 2: line 6) and sources (Algorithm 2: line 7) are defined following the 3D-convolution of the surface/collision cell-based tallies (Algorithm 4: line 27).

---

**Algorithm 1: Mesh-based form factors**

---

1: **Iteration scheme:**  
 2: **for**  $k \leftarrow$  outer iterations **do**  
 3:     ...  
 4:     **for**  $n \leftarrow$  mesh cells **do**  
 5:         **for**  $h \leftarrow$  sub-mesh cells **do**  
 6:              $f_{i,h} = \frac{R_{i,h}}{J_i^{in}}$   
 7:              $f_{s,h} = \frac{R_{s,h}}{S}$

---



---

**Algorithm 2: FET-based form factors**

---

1: **Iteration scheme:**  
 2: **for**  $k \leftarrow$  outer iterations **do**  
 3:     ...  
 4:     **for**  $n \leftarrow$  mesh cells **do**  
 5:         **for**  $c \leftarrow$  FE-th coefficients **do**  
 6:              $\hat{f}_i^c = \frac{\hat{R}_i^c}{J_i^{in}}$   
 7:              $\hat{f}_s^c = \frac{\hat{R}_s^c}{S}$

---

Algorithm 3 has been formulated based on Leppänen [8, 11]. Algorithm 4 has been constructed based on Wendt [12], Kerby [13], Cheatham [14] and Griesheimer [15].

---

**Algorithm 3: Response matrix solver**

---

1: **Inputs:**  
 2:  $S_0 \leftarrow S$  uniform source distribution  
 3: **Interface current components:**  
 4: **for**  $n \leftarrow$  mesh cells **do**  
 5:     **for**  $i \leftarrow$  in-boundary **do**  
 6:          $\hat{\mathbf{J}}^{in} = [\mathbf{J}^{in}] \leftarrow \mathbf{J}^{in} = [J_i^{in}]$   
 7:     **for**  $j \leftarrow$  out-boundary **do**  
 8:          $\hat{\mathbf{J}}^{out} = [\mathbf{J}^{out}] \leftarrow \mathbf{J}^{out} = [J_j^{out}]$   
 9: **Mapping:**  
 10: **for**  $n \leftarrow$  mesh cells **do**  
 11:     **for**  $m \leftarrow$  mesh (neighbour) cells **do**  
 12:         **for**  $i \leftarrow$  in-boundary **do**  
 13:             **for**  $j \leftarrow$  out-boundary **do**  
 14:                 **if**  $(J_{m,j}^{out} \equiv J_{n,i}^{in})$  **then**  $T_{m,n}^{j,i} = 1$   
 15:                  $\mathbf{T} = [T_{m,n}]$   
 16:     **for**  $n \leftarrow$  mesh cells **do**  
 17:         **for**  $i \leftarrow$  in-boundary **do**  
 18:             **for**  $j \leftarrow$  out-boundary **do**  
 19:                  $\alpha = [\alpha_{i,j}] \leftarrow \alpha_{i,j} = J_{i,j} / J_i^{in}$   
 20: **Coupling coefficients:**  
 21: **for**  $n \leftarrow$  mesh cells **do**  
 22:     **for**  $j \leftarrow$  out-boundary **do**  
 23:          $\mathbf{s} = [s_j] \leftarrow s_j = J_{s,j} / S$   
 24:     **for**  $i \leftarrow$  in-boundary **do**  
 25:          $\mathbf{r} = [r_i] \leftarrow r_i = R_i / J_i^{in}$   
 26:      $r_s = R_s / S$   
 27: **Initialize:**  
 28: **for**  $n \leftarrow$  mesh cells **do**  
 29:      $[\mathbf{J}_n^{out}]^{(l=0)} = \mathbf{s}_n S_n$  1<sup>st</sup> inner iteration  
 30: **Iteration scheme:**  
 31: **for**  $k \leftarrow$  outer iterations **do**  
 32:     **do**  $l \leftarrow$  inner iteration  
 33:         **for**  $n \leftarrow$  mesh cells **do**  
 34:              $[\hat{\mathbf{J}}^{in}]^l = \mathbf{T} [\hat{\mathbf{J}}^{out}]^{l-1}$   
 35:              $[\mathbf{J}_n^{out}]^l = \alpha_n [\mathbf{J}_n^{in}]^l$   
 36:     **while**  $((\varepsilon > \varepsilon_{\max}) \parallel (l < l_{\max}))$   
 37:         **for**  $n \leftarrow$  mesh cells **do**  
 38:              $R_n = r_{s,n} S_n + \mathbf{r}_n \sum_l [\mathbf{J}_n^{in}]^l$   
 39:              $k_{eff} = \frac{\sum_n R_n}{\sum_n S_n}$

---

---

**Algorithm 4: Functional expansion tallies (FETs)**

---

- |  |  |
|--|--|
| <p>1: <b>Functional expansion (FE):</b></p> <p>2: <math>F(x) \approx \sum_{m=0}^M a_m k_m \psi_m(x) \rho(x)</math></p> <p>3: <b>where</b></p> <p>4: <math>a_m = \int F(x) \psi_m(x) \rho(x) dx</math></p> <p>5: <b>Surface cell-based Tallies (FETs)</b></p> <p>6: <b>for</b> <math>n \leftarrow</math> neutron particles <b>do</b></p> <p>7: <math>\hat{a}_m = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \omega_{n,k} \psi_m(x_{n,k})</math></p> <p>8: <b>Collision cell-based Tallies (FETs)</b></p> <p>9: <b>for</b> <math>n \leftarrow</math> neutron particles <b>do</b></p> <p>10: <math>\hat{a}_m = \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K \frac{\omega_{n,k} \psi_m(x_{n,k})}{\Sigma_t(x_{n,k})}</math></p> <p>11: <b>Legendre FE-basis (1D):</b></p> <p>12: <math>\hat{F}_m(\tilde{x}) = \sum_{m=0}^M \hat{a}_m k'_m P_m(\tilde{x})</math><br/> <math>\rightarrow \tilde{x} \in [-1, 1]</math> Legendre sub-space</p> <p>13: <b>where</b></p> <p>14: <math>\tilde{x} = 2 \frac{x - x_{min}}{x_{max} - x_{min}} - 1</math></p> <p>15: <math>k'_m = \frac{2m + 1}{x_{max} - x_{min}}</math></p> <p>16: <math>P_m(\tilde{x}) = \sum_{i=0}^m \binom{m}{i} \binom{m}{1}^i \left( \frac{\tilde{x} - 1}{2} \right)^i</math></p> | <p>17: <b>Zernike FE-basis (2D):</b></p> <p>18: <math>\hat{F}_n^m(\tilde{r}, \tilde{\phi}) = \sum_{n=0}^N \sum_{m=0}^M \hat{a}_n^m k'_n{}^m Z_n^m(\tilde{r}, \tilde{\phi})</math><br/> <math>\rightarrow \tilde{r} \in [0, 1], \tilde{\phi} \in [0, 2\pi]</math> Zernike sub-space</p> <p>19: <b>where</b></p> <p>20: <math>\tilde{r} = \frac{\sqrt{(x-x_0)^2 + (y-y_0)^2}}{r}, \tilde{\phi} = \phi</math></p> <p>21: <b>for</b> <math>n - m</math> is even, <math>n \geq  m  \geq 0</math> <b>do</b></p> <p>22: <math>Z_n^m(\tilde{r}, \tilde{\phi}) = \begin{cases} R_n^m(\tilde{r}) \Phi_m(\tilde{\phi}) &amp; m &gt; 0 \\ R_n^0(\tilde{r}) &amp; m = 0 \\ R_n^{ m }(\tilde{r}) \Phi_m(\tilde{\phi}) &amp; m &lt; 0 \end{cases}</math></p> <p>23: <math>R_n^{ m }(\tilde{r}) = \sum_{k=0}^{\frac{n- m }{2}} (-1)^k \binom{n-k}{k} \binom{n-2k}{\frac{n- m }{2}-k} \tilde{r}^{n-2k}</math></p> <p>24: <math>\Phi_{ m }(\tilde{\phi}) = \begin{cases} \cos( m \tilde{\phi}) &amp; m &gt; 0 \\ 1 &amp; m = 0 \\ \sin( m \tilde{\phi}) &amp; m &lt; 0 \end{cases}</math></p> <p>25: <math>k'_m = \begin{cases} \frac{n+1}{2\pi} &amp; m = 0 \\ \frac{2(n+1)}{\pi} &amp; \text{otherwise} \end{cases}</math></p> <p>26: <b>Convolution (3D):</b></p> <p>27: <math>\hat{F}_c(\Lambda) = \sum_{c=0}^C \hat{a}_c k'_c \psi_c(\Lambda) \quad \Lambda \in \mathbb{R}^3</math></p> <p>28: <b>where</b></p> <p>29: <math>\hat{F}(\tilde{x}, \tilde{y}, \tilde{z}) = \sum_{I,J,K} \hat{a}_{i,j,k} k'_{i,j,k} P_i(\tilde{x}) P_j(\tilde{y}) P_k(\tilde{z})</math></p> <p>30: <math>\hat{F}(\tilde{r}, \tilde{\phi}, \tilde{z}) = \sum_{N,M,I} \hat{a}_{n,i}^m k'_{n,i}{}^m Z_n^m(\tilde{r}, \tilde{\phi}) P_i(\tilde{z})</math></p> |
|--|--|
- 

### 3. VERIFICATION

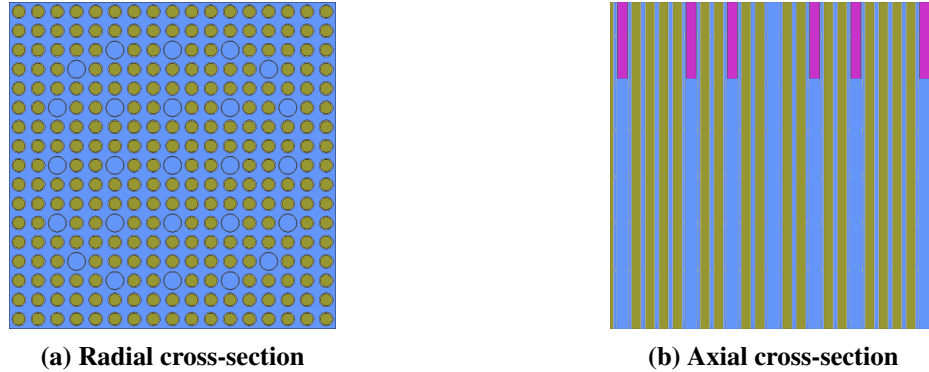
The hybrid approach, FET-response matrix-based solver Monte Carlo method, is implemented in Serpent version 2.1.32 (non-released version at the time of the writing). The calculation of intra-cell form factors relies on subroutines that were originally implemented to perform the coupling of Serpent to the MOOSE Framework [16], and later to be used with Serpent multi-physics interface [12]; while the built-in response matrix solver, which is being used for accelerating the fission source convergence [8], was initially designed as the built-in importance solver to produce weight-window meshes for variance reduction [11].

In order to verify the implemented methodology, a single-assembly and full-core PWR test cases are proposed, based on the MIT BEAVRS benchmark [10], for which the Serpent input model had already been prepared for previous studies [17, 18]. Fission source convergence is monitored using the Shannon entropy [19], for radial ( $x, y$ ) and axial ( $z$ ) dimensions. To reduce random oscillations

in the entropy and assure that the results are not affected by the source bias [20], the simulations were run with an enough large population size (standard deviation of  $\sigma < 0.1\%$ ).

### 3.1. Single-assembly Calculation

The single-assembly test case is defined as a single 2.4% enriched fuel assembly without burnable absorber. The assembly is modelled in an 3D-infinite lattice, with reflective boundary conditions radially and vacuum boundary conditions axially. The fission source convergence is monitored only in the axial dimension. An axially asymmetric fission source distribution is obtained by inserting the control rods, 1/4 of the active fuel height (see Figure 1).

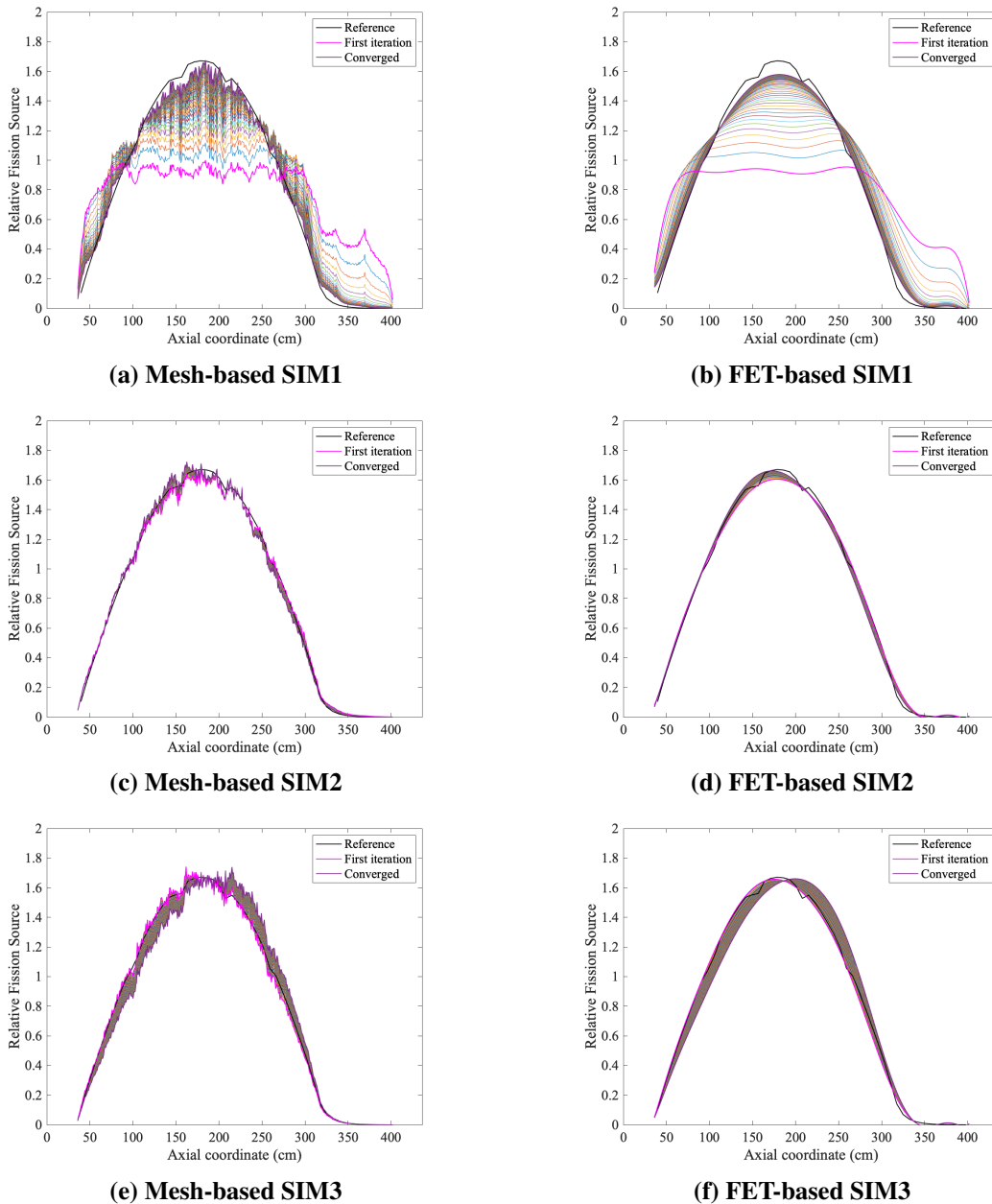


**Figure 1: Radial (1a) and axial (1b) cross-sections of 2.4% wt. U-235 fuel assembly.**

Based on a previous study [8], the response matrix solution is obtained in a 1D-axial Cartesian mesh of 11 bins. The intra-cell source distribution reconstruction is performed in 40 bins for the mesh-based case and with 10-th order Legendre polynomial series expansion for the FET-based case (initial analysis phase). The Monte Carlo simulation samples  $10^6$  source neutrons per cycle.

The evolution of the fission source distribution is captured along three stages (see Figure 2): SIM1, first response-matrix solution initialises from an uniform source distribution, that converges after 79 outer iterations. Side-by-side, it can be observed the behaviour of, both, the mesh-based and FET-based approaches, still far from the converged solution. The functional expansion preserves the distribution continuity and provides a smoother solution. The converged solution is used as initial iteration for the second simulation, which produces the coupling coefficients for the next response matrix solution, SIM2. After 89 outer iterations, the source is converged, being relatively close to the reference solution. The FET-based case shows a good agreement with respect the converged Monte Carlo solution. At SIM3, the response matrix methodology no longer shows improvements, just random variation/noise, intrinsic to the stochastic sampling. This behaviour agrees with the root-mean-square error (RMSE): 0.2582 (SIM1), 0.1331 (SIM2) and 0.1190 (SIM3) - calculated between the reference solution and the FET-based approach.

The new reconstruction method evaluates the fission source in a more congruent and native form to Monte Carlo, given its geometry-free and continuous nature, allowing to improve the convergence performance. Although, the polynomial series estimate, due to its truncation (numerical) error, does not completely reproduce the heterogeneities as a result of the rods insertion and/or the grid-spacers. Additional studies are necessary to establish the parameters that fit the requirements of the case, such i.e., the polynomial expansion order or polynomial basis function.



**Figure 2:** Axial fission source distribution in the single assembly-case. (Coloured curves: outer iterations of response matrix solver; black curve: Monte Carlo converged solution.)

Figure 3 shows the evolution of the inward current response coupling coefficient and direct source to response coupling coefficient, for the two first simulations (SIM1 and SIM2).

The response matrix solver computes a population of  $10^6$  source neutrons 3-times; which given the size of the problem, it is negligible (each simulation takes less than a second in total). There is not a significant difference between the mesh-based and FET-based approach in terms of computational time. The source convergence after the criticality source cycles is presented in Figure 4. The response matrix methodology, for both approaches, calculates the coupling coefficients in agreement; with a better performance than the non-accelerated case since the beginning of the calculation.

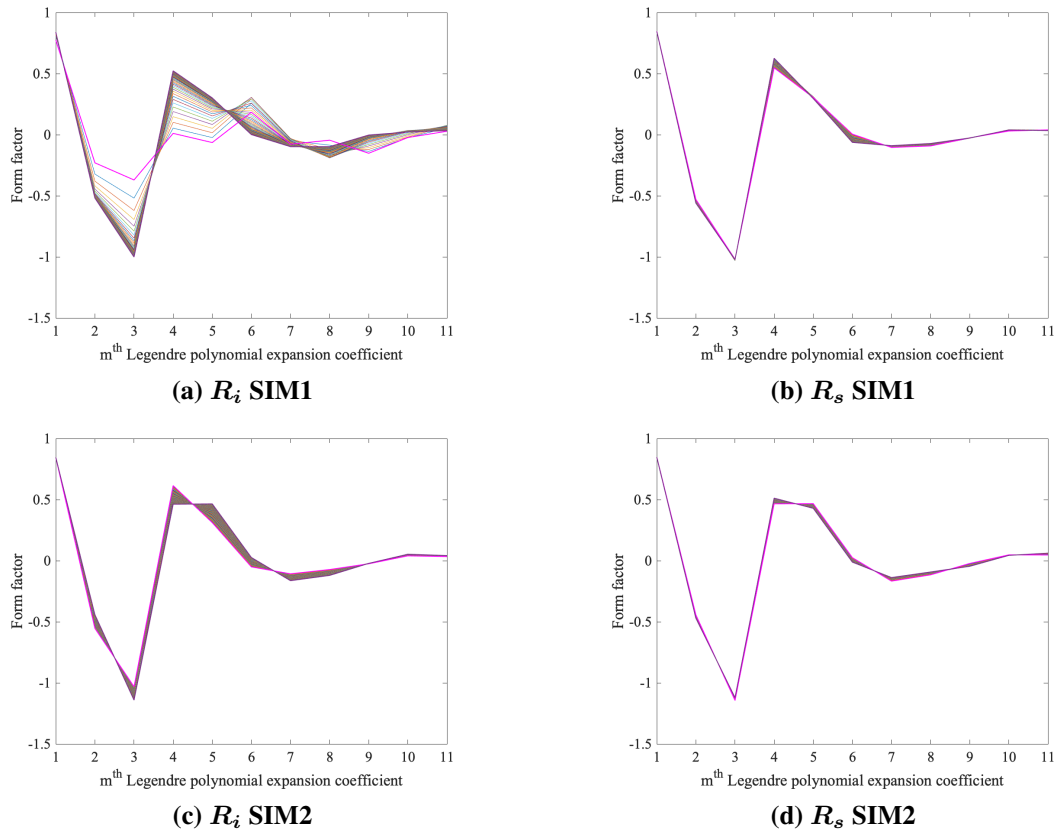


Figure 3: Legendre polynomial expansion of coupling coefficients.

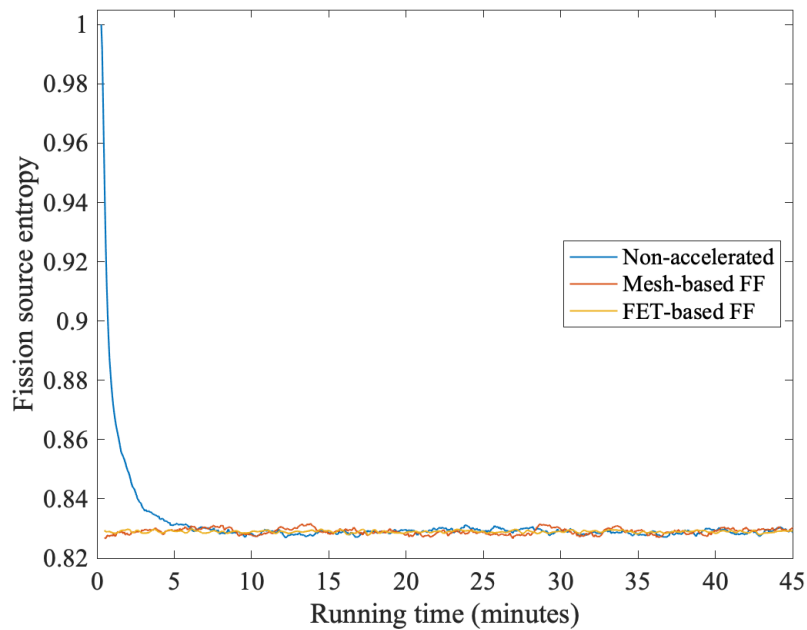
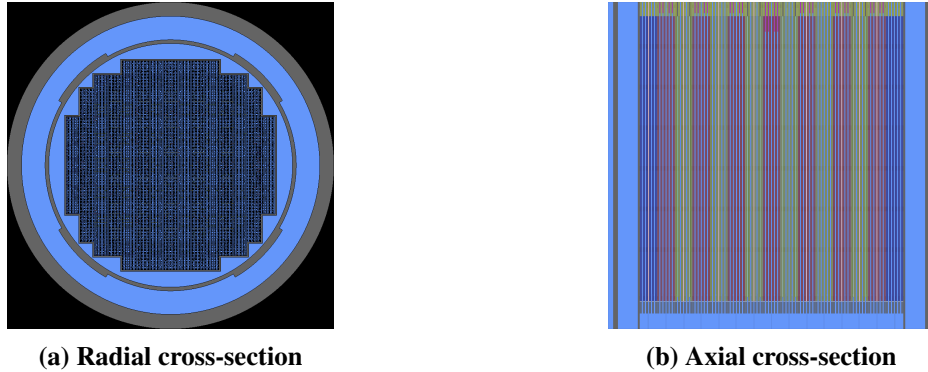


Figure 4: Axial fission source entropy for single-assembly test case.



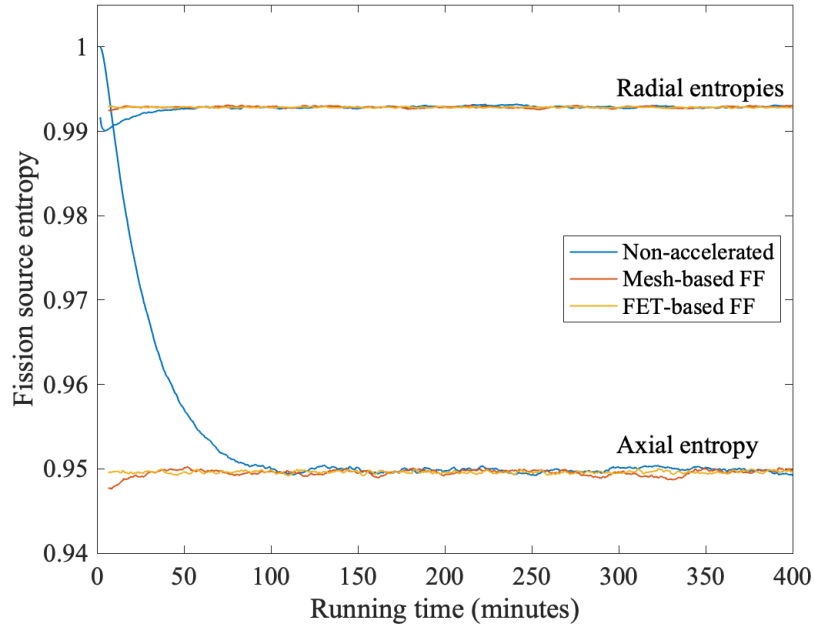
### 3.2. Full-core Calculation

The full-core test case represents a critical core departing from the HZP start-up PWR core specified in the BEAVRS benchmark; where all control rods are fully withdrawn, except bank D with an insertion of  $\sim 6.5\%$  of the active fuel height, and coolant concentration of 975 ppm, (see Figure 5).



**Figure 5:** Radial (5a) and axial (5b) cross-sections of BEAVRS full core.

Based on a previous study [8], the discretisation of the response matrix is  $25 \times 25 \times 22$  Cartesian mesh; where each bin radially represents an assembly. The intra-cell source distributions are reconstructed in a  $4 \times 4 \times 4$  bins for the mesh-based case and in a 10-th order Legendre polynomial series expansion, in each dimension, for the FET-based case. The Monte Carlo simulation samples a population of  $10^7$  source neutrons per cycle.



**Figure 6:** Fission source entropies for critical HZP configuration full-core test case.

Figure 6 shows source convergence, through axial- and radial-Shannon entropy, for the Monte Carlo simulation with and without improved initial fission source guess. Both, mesh-based and

FET-based intra-cell form factors cases are presented; for which, the response matrix solutions takes 30 – 50 seconds in total. Radial entropies converge quite fast, due to a flat profile of the reactor core radial power. It is not the case of the axial entropy, which requires more time to achieve convergence, given a uniform initial source distribution. Nevertheless, the entropy evolution of the accelerated cases is faster than the non-accelerated case given an improved initial guess.

#### **4. CONCLUSIONS AND DISCUSSION**

A new methodology is presented to accelerate the convergence of the fission source on Monte Carlo calculations. The approach is embedded into the response matrix-based method; which, through a short Monte Carlo simulation provides an improved initial guess distribution. The idea behind this upgrade is to reconstruct the intra-cell source distributions by applying functional expansion tallies (FET) instead of a high-resolution mesh, providing a more realistic and accurate solution.

The new implementation reconstructs the fission source in a more natural way: due to its geometry agnostic and continuous nature, which allows to enhance convergence performance. However, the polynomial series estimate, due to its truncation (numerical) error, seems not to fully capture the heterogeneous behaviour caused by the rods insertion or the grid-spacers. Nonetheless, the method shows potential in achieving a nearly converged solution at the initial stages, as the test cases, single-assembly and full-core, show. In terms of performance, the functional expansion tallies as an alternative to a high-resolution mesh to calculate the coupling coefficients involved in the response matrix calculations, do not suppose a major reduction of the computational cost, due to the shortness of the response matrix-based method itself: reconstructing the intra-cell source distribution takes an almost negligible amount of time compared with whole simulation. As well, the response matrix methodology, for both mesh-based and FET-based approaches shows random variations given its stochastic sampling formulation.

The suitability of the functional expansion tallies to accelerate the convergence of the fission source, brings new and improved features to the response matrix method, in addition to, a potential new range of applications. Nevertheless, further testing and analysis are necessary in order to understand those capabilities and performance of the upgraded response matrix approach, due to showing these in practice is not straightforward.

As well, it seems important to provide guidance on the method's use, based on the effect of the free-parameters, e.g., neutron size batch, mesh discretisation, polynomial basis function and expansion order, etc; since, at the current stage, the presented approach is case-based: needing user's inside knowledge on the case itself.

A natural evolution of the method would be to evolve in parallel, both, the response matrix built-in solver and the functional expansion tallies, e.g. additional geometry mapping, methodology automatization, etc. Also, the response matrix method and the functional expansion tallies could be applied to other problems: 1) independently, i.e. response matrix as a reduced order method apply to Monte Carlo calculations, or functional expansion tallies apply to reconstruct Monte Carlo spatial/temporal distributions; and 2) combined, such i.e., time-dependent Monte Carlo methods.

## NOMENCLATURE

### Algorithm 1: ‘Mesh-based form factors’ on page 3

$f_{i,h}$	Inward current cell-wise, $h$ , form factor
$f_{s,h}$	Direct source cell-wise, $h$ , form factor
$R_{i,h}$	Inward current cell-wise, $h$ , from cell boundary $i$ response
$R_{s,h}$	Direct source cell-wise, $h$ , response

### Algorithm 2: ‘FET-based form factors’ on page 3

$f_i^m$	Inward current $m^{th}$ order expansion estimate form factor
$f_s^m$	Source $m^{th}$ order expansion estimate form factor
$\hat{R}_i^m$	Inward current from cell boundary $i$ , $m^{th}$ order estimate response
$\hat{R}_s^m$	Direct source $m^{th}$ order estimate response

### Algorithm 3: ‘Response matrix solver’ on page 3

$\alpha_{i,j}$	Current transfer coefficient from cell boundary $i$ to cell boundary $j$
$J_i^{in}$	Total inward current through cell boundary $i$
$J_j^{out}$	Total outward current through cell boundary $j$
$J_{i,j}$	Partial current from cell boundary $i$ to cell boundary $j$
$J_{s,j}$	Outward current from direct source through cell boundary $j$
$k_{eff}$	Multiplication factor
$R$	Total response
$R_i$	Inward current through cell boundary $i$ response
$r_i$	Inward current through cell boundary $i$ response coefficient
$R_s$	Direct source response
$r_s$	Direct source response coefficient
$S$	Fission source rate
$S_0$	Initial fission source rate
$s_j$	Outward current through cell boundary $j$ response coefficient
$T_{m,n}^{j,i}$	Current topology coefficient, from cell $m$ boundary $j$ to cell $n$ boundary $i$

### Algorithm 4: ‘Functional expansion tallies (FETs)’ on page 4

$\omega_n$	$n^{th}$ particle weight
$\psi_m(x)$	$m^{th}$ basis function expansion
$\psi_m(x_n)$	$n^{th}$ particle contribution to the $m^{th}$ polynomial
$\rho(x)$	Associated weighing function for the basis function
$\Sigma_t(x_{n,k})$	Macroscopic transport cross-section
$a_m$	$m^{th}$ functional expansion coefficient
$F(x)$	Function to be expressed as a series expansion
$k_n^m$	$n^{th}$ , $m^{th}$ orthonormalisation constant transformation to Zernike’s sub-space
$k_m$	$m^{th}$ orthonormalisation constant transformation to Legendre’s sub-space
$k_m$	$m^{th}$ orthonormalization constant
$N$	Total number of particles
$P_m(\tilde{x})$	$m^{th}$ Legendre’s polynomial expansion
$Z_n^m(\tilde{r}, \tilde{\phi})$	$n^{th}$ radial & $m^{th}$ azimuthal Zernike’s polynomial expansion
$\hat{a}_m$	$m^{th}$ functional expansion estimate coefficient
$\hat{F}_c(\Lambda)$	Linear (3D) convolution of functional expansion series
$\tilde{r}, \tilde{\phi}$	Radial & azimuthal coordinate transformation to Zernike’s sub-space
$\tilde{x}$	Coordinate transformation to Legendre’s sub-space

## REFERENCES

- [1] T. Kitada and T. Takeda. “Effective Convergence of Fission Source Distribution in Monte Carlo Simulation.” *J Nucl Sci Technol*, **volume 38**(5), pp. 324–329 (2001).
- [2] S. Carney, F. Brown, and B. Kiedrowski. *Ann Nucl Energy*, pp. 423–431.
- [3] L. Pan, R. Wang, and S. Jiang. “Monte Carlo Fission Matrix Acceleration Method with Limited Inner Iteration.” *Nucl Sci Eng*, **volume 180**(2), pp. 199–208 (2015).
- [4] J. Dufek and K. Tuttleberg. “Monte Carlo Criticality Calculations by a Growing Neutron Population.” *Ann Nucl Energy*, **volume 94**, pp. 16–21 (2016).
- [5] T. Yamamoto and Y. Miyoshi. “Reliable Method for Fission Source Convergence of Monte Carlo Criticality Calculation with Wielandt’s Method.” *J Nucl Sci Technol*, **volume 41**(2), pp. 99–107 (2004).
- [6] D. She, K. Wang, and G. Yu. “Asymptotic Wielandt Method and Superhistory Method for Source Convergence in Monte Carlo Criticality Calculation.” *Nucl Sci Eng*, **volume 172**(2), pp. 127–137 (2012).
- [7] M. J. Lee, H. G. Joo, D. Lee, and K. Smith. “Coarse Mesh Finite Difference Formulation for Accelerated Monte Carlo Eigenvalue Calculation.” *Ann Nucl Energy*, **volume 65**, pp. 101–113 (2013).
- [8] J. Leppänen. “Acceleration of Fission Source Convergence in the Serpent 2 Monte Carlo Code using a Response Matrix Based Solution for the Initial Source Distribution.” *Ann Nucl Energy*, **volume 128**, pp. 63–68 (2019).
- [9] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, and T. Kaltiaisenaho. “The Serpent Monte Carlo code: Status, development and applications in 2013.” *Ann Nucl Energy*, **volume 82**, pp. 142–150 (2015).
- [10] N. Horelik, B. Herman, M. Ellis, S. Kumar, J. Liang, B. Forget, and K. Smith. “Benchmark for Evaluation and Validation of Reactor Simulations (rev. 2.0.2).” (2018). URL <http://crpg.mit.edu/research/beavrs>.
- [11] J. Leppänen, T. Viitanen, and O. Hyvönen. “Development of a Variance Reduction Scheme in the Serpent 2 Monte Carlo Code.” In *Proceedings of M&C 2017*. Jeju, Korea, Apr. 16-20, 2017.
- [12] B. Wendt, L. Kerby, A. Tumulak, and J. Leppänen. “Advancement of Functional Expansion Capabilities: Implementation and Optimization in Serpent 2.” *Nuc Eng Des*, **volume 334**, pp. 138–153 (2018).
- [13] L. Kerby, A. Tumulak, J. Leppänen, and V. Valtavirta. “Preliminary Serpent-MOOSE Coupling and Implementation of Functional Expansion Tallies in Serpent.” In *Proceedings of M&C 2017*. Jeju, Korea, Apr. 16-20, 2017.
- [14] J. Cheatham, J. P. Holloway, and W. R. Martin. “Improving Monte Carlo Source Convergence with the Functional Expansion Technique.” *Trans Am Nucl Soc*, **volume 96**, pp. 838–840 (2007).
- [15] D. P. Griesheimer. *Functional Expansion Tallies for Monte Carlo Simulations*. Ph.D. thesis, University of Michigan, Ann Arbor, MI, USA (2005).
- [16] D. Gaston, C. Newman, G. Hansen, and D. Lebrun-Grandié. “MOOSE: A Parallel Computational Framework for Coupled Systems of Nonlinear Equations.” *Nucl Eng Des*, **volume 239**, pp. 1768–1778 (2009).

- [17] J. Leppänen, R. Mattila, and M. Pusa. “Validation of the Serpent-ARES code sequence using MIT BEAVRS benchmark - Initial core at HZP conditions.” *Ann Nucl Energy*, **volume 69**, pp. 212–225 (2014).
- [18] J. Leppänen and R. Mattila. “Validation of the Serpent-ARES code sequence using the MIT BEAVRS benchmark - HFP conditions and fuel cycle 1 simulations.” *Ann Nucl Energy*, **volume 96**, pp. 324–331 (2016).
- [19] F. Brown. “On the Use of Shannon Entropy of the Fission Distribution for Assessing Convergence of Monte Carlo Criticality Problems.” In *Proceedings of PHYSOR 2006*. Vancouver, BC, Canada, Sept. 10-14, 2006.
- [20] F. Brown, W. R. Martin, J. Leppänen, W. Haeck, and B. Cochet. “Reactor Physics Calculations with Monte Carlo.” Technical report, Los Alamos National Laboratory, LA-UR-10-02762 (2010).