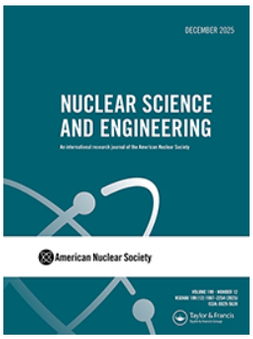
Monte Carlo and Deterministic Implementations of the Zeta-Eigenvalue Equation for Efficient Criticality Searches



Nuclear Science and Engineering

ISSN: 0029-5639 (Print) 1943-748X (Online) Journal homepage: www.tandfonline.com/journals/unse20

Monte Carlo and Deterministic Implementations of the Zeta-Eigenvalue Equation for Efficient Criticality Searches

Valeria Raffuzzi, Nicolò Abrate & Sandra Dulla

To cite this article: Valeria Raffuzzi, Nicolò Abrate & Sandra Dulla (05 Nov 2025): Monte Carlo and Deterministic Implementations of the Zeta-Eigenvalue Equation for Efficient Criticality Searches, Nuclear Science and Engineering, DOI: 10.1080/00295639.2025.2568341

To link to this article: https://doi.org/10.1080/00295639.2025.2568341

NUCLEAR SCIENCE AND ENGINEERING

DOI: https://doi.org/10.1080/00295639.2025.2568341

Monte Carlo and Deterministic Implementations of the Zeta-Eigenvalue Equation for Efficient Criticality Searches

Valeria Raffuzzi [a], Nicolò Abrate [b,c], and Sandra Dulla [b,c]

[a]*Department of Engineering, University of Cambridge, Cambridge, United Kingdom;* [b]*Dipartimento Energia, Politecnico di Torino, NEMO Group, Torino, Italy;* [c]*Sezione di Genova, Istituto Nazionale di Fisica Nucleare (I.N.F.N.), Genova, Italy*

Received July 30, 2025
Accepted for Publication September 24, 2025

Abstract — *The zeta (ζ)-eigenvalue equation is a formulation of the neutron transport equation that ensures criticality by scaling the density of the nuclides or materials of choice with the parameter ζ. As a consequence, the value of ζ provides quantitative information about what material density or nuclide concentration can make a system critical. This paper proposes a Monte Carlo algorithm to solve the ζ-eigenvalue equation. Additionally, it tackles practical challenges in the deterministic implementation of the method. The Monte Carlo algorithm, based on a fixed-point iteration scheme, is implemented in the code SCONE and tested on some cases of practical utility. Examples include the search of critical gadolinium concentration in a pressurized water reactor assembly and the search of critical boron in BEAVRS. The Monte Carlo implementation is successful at finding the critical densities requested and is faster than state-of-the art iterative searches based on k-eigenvalue calculations. The second part of the paper illustrates the impact of the density scaling on both the generation of multigroup constants and spatial self-shielding effects, relevant to the deterministic implementation. A computational scheme exploiting the ζ formulation to get reliable results in a deterministic framework is also suggested.*

Keywords — *Criticality search, zeta eigenvalue, Monte Carlo transport, deterministic transport.*

I. INTRODUCTION

Several eigenvalue formulations exist that allow one to solve the neutron transport equation for a fundamental mode. The most common is the k-eigenvalue formulation, where the neutron multiplication factor k_{eff} is used to scale the fission source. Alternatively, the α-eigenvalue, where α is the fundamental time-decay constant of the system, is used to capture time-dependent effects [1]. Davison [2] proposed the c-eigenvalue (also indicated as γ), used as a scaling factor of the fission and scattering source. Ronen et al. [3] introduced the δ-eigenvalue, which acts as a density scaling for all the materials in the system. While all of these formulations are useful for some applications, when applied to reactor design, they all fall short of providing the designer with quantitative information on how to make the system critical.

More recently, Abrate et al. [4] proposed the zeta (ζ)-eigenvalue formulation. Similarly to δ, ζ is used as a density scaling parameter. However, instead of acting on all the materials in the system, it acts on only the materials or nuclides of choice: This allows for devising applications of practical interest for nuclear engineers. For example, the concentration of boron dissolved in a pressurized water reactor (PWR) coolant that makes the reactor critical can be obtained from solving a ζ-eigenvalue equation. The same applies to the concentration of cadmium to be added to the control rods of a reactor, or burnable poisons in the fuel. This is in contrast to using an iterative search method, which normally consists of repeating a k-eigenvalue calculation multiple times and might fail to converge [5]. In this respect, perturbation theory [6] could provide similar information while being computationally cheaper than the iterative k-eigenvalue calculations, despite its accuracy

CONTACT Valeria Raffuzzi vr339@cam.ac.uk

strongly depending on the kind of parameter perturbed and the magnitude of the perturbation. The ζ-eigenvalue approach does not suffer from these limitations, although the existence of a physical solution depends both on the isotope of interest and on the features of the off-critical system considered in the calculation.

In [4], the ζ-eigenvalue formulation was implemented in a deterministic test platform and applied to deal with simple one-dimensional (1D) problems. In this work, based on [7], a Monte Carlo algorithm to solve the ζ-eigenvalue equation is devised and implemented in the Monte Carlo code SCONE [8]. Then, the method is tested on some realistic numerical examples. It must be noticed that a somewhat similar method is implemented in the Monte Carlo code Serpent [9]. Serpent allows the option to perform a critical density iteration on the nuclides of choice, which produces a density scaling factor. This can be applied only to reasonably strong absorbers. Similarly, inline critical searches were developed in the past for both deterministic [10] and Monte Carlo [11] methods. Despite the similarities, the method proposed here is more general and assumption free. Additionally, this paper expands on the work performed in [4] by discussing practical aspects of the deterministic implementation of the ζ-eigenvalue equation and its challenges, different from those of the Monte Carlo framework.

The rest of the paper proceeds as follows. Sec. II describes the theory of the ζ-eigenvalue equation and the Monte Carlo and deterministic implementations. Sec. III presents the results of the numerical tests. Sec. V provides a summary and outlines avenues of future work.

II. THEORY AND METHODOLOGY

II.A. Theory

A generic, compact formulation of the neutron transport equation is given by

$$\frac{1}{v}\frac{d\psi}{dt} + (L+T)\psi = (F+S)\psi \quad . \tag{1}$$

where ψ = angular neutron flux; t = time; v = velocity; L, T, F, and S = streaming, collision, fission, and scattering operators, respectively. In order to solve this equation for the fundamental mode of the angular flux, it is normally posed as an eigenvalue equation. The most common formulation is the k-eigenvalue equation, where the neutron multiplication factor k is introduced to scale the fission source in order to maintain the system critical:

$$(L+T-S)\psi_k = \frac{1}{k}F\psi_k \quad . \tag{2}$$

As mentioned in Sec. I, the alternative ζ-eigenvalue equation ensures criticality by scaling the density of a nuclide, ensemble of nuclides, or material m chosen by the user. Naturally, the contributions of m and everything else, referred to as $\overline{m}$, have to be explicitly split:

$$(L + T_{\overline{m}} + \frac{1}{\zeta}T_m)\psi_\zeta = (F_{\overline{m}} + \frac{1}{\zeta}F_m + S_{\overline{m}} + \frac{1}{\zeta}S_m)\psi_\zeta \ . \tag{3}$$

Note that in Eq. (3), k was fixed to a value of 1. When solving for the fundamental mode ψ_ζ, this is expected to be that of the critical system, i.e., the system with atomic densities (or concentrations) of material m being scaled by ζ. Rearranging the terms to write Eq. (3) as an eigenvalue equation, one gets

$$(L + T_{\overline{m}} - F_{\overline{m}} - S_{\overline{m}})\psi_\zeta = \frac{1}{\zeta}(F_m + S_m - T_m)\psi_\zeta \quad . \tag{4}$$

If one were to solve this problem with a Monte Carlo code by power iteration as is commonly done, the term in parentheses on the right hand side of Eq. (4) (for brevity, B_m) should be the source to be produced each iteration for the next; the operator in parentheses on the left hand side (for brevity, $B_{\overline{m}}$) should be inverted to calculate the estimate of the flux for the following iteration. This is described by

$$\psi_\zeta^{(n+1)} = \frac{1}{\zeta^{(n)}} B_{\overline{m}}^{-1} B_m \psi_\zeta^{(n)} \quad . \tag{5}$$

However, there is no guarantee that the operators B_m and $B_{\overline{m}}$ are positive. This depends on the properties of the system modeled, such as the cross sections, geometry, and boundary conditions. For example, if m is a nonfissile nuclide, B_m will always be negative. Negative operators are difficult to interpret in a Monte Carlo framework, and they might lead to high variance results. While recently some work has been done to tackle negative weight particles in Monte Carlo [12], cases with a net negative source were not tested. Despite this maybe being possible, using the conventional power iteration to solve the ζ-eigenvalue equation in Monte Carlo was not pursued in this work. To overcome this issue, Eq. (3) can be solved as a fixed-point iteration, as shown in Eq. (6), with initial guesses $\psi_\zeta^{(0)} = \psi_{\zeta,0}$ and $\zeta^{(0)} = \zeta_0$. By integrating Eq. (6), we obtain Eq. (7), which is solved to update ζ:

$$\psi_{\zeta}^{(n+1)} = (L + T_{\overline{m}} + \frac{1}{\zeta^{(n)}} T_m - S_{\overline{m}} - \frac{1}{\zeta^{(n)}} S_m)^{-1} (F_{\overline{m}} + \frac{1}{\zeta^{(n-1)}} F_m) \psi_{\zeta}^{(n)} \tag{6}$$

$$\zeta^{(n+1)} = \frac{\int_{V_m,\Omega,E} (F_m + S_m - T_m) \psi_{\zeta}^{(n+1)} \mathrm{dV d\Omega dE}}{\int_{V,\Omega,E} L \psi_{\zeta}^{(n+1)} \mathrm{dV d\Omega dE} + \int_{\mathrm{V}_{\overline{\mathrm{m}}},\Omega,\mathrm{E}} (T_{\overline{m}} - F_{\overline{m}} - S_{\overline{m}}) \psi_{\zeta}^{(\mathrm{n}+1)} \mathrm{dV d\Omega dE}.} \tag{7}$$

Here, the integrals are tallied over the relevant parts of the phase-space. The domain V is divided into V_m and $V_{\overline{m}}$, which are the regions where m is contained and everywhere else, respectively. It must be noticed that in this process, the scattering operators $S_{\overline{m}}$ and S_m have been moved to the left-hand side of the equation and are now subtracted to the total collision operators. This was done to maximize similarities between the Monte Carlo implementation of Eq. (6) and the classical k-eigenvalue power iteration algorithm. In practice, this makes the method equivalent to a critical inline search, where ζ converges in parallel with the fission source. The details of the Monte Carlo algorithm adopted to solve this equation are explained in the next section.

Conversely, in a deterministic framework, negative operators are not an issue numerically, allowing one to estimate the fundamental eigenpair with the classical power iteration method available in most codes. Nevertheless, as will be discussed in the following, the adoption of a deterministic approach poses some challenges concerning the generation of the multigroup constants and the treatment of spatial self-shielding.

II.B. Monte Carlo Implementation

The Monte Carlo code used for this work is SCONE [8]. SCONE has the capability of solving fixed-source problems, the k-eigenvalue equation, the $\alpha - k$ or α-static equation [13], and the c-eigenvalue equation [14]. For this work, the fixed-point iteration scheme described in Eqs. (6) and (7) was implemented.

When writing the fixed-point iteration scheme needed to solve the ζ-eigenvalue equation as in Eq. (6), the Monte Carlo algorithm is almost identical to the power iteration used to solve the k-eigenvalue equation. In fact, the neutron source to be used in each iteration is simply the fission source calculated in the previous iteration. This approach requires minimal code modifications.

The main steps of the ζ calculation are described as follows:

1. Define the problem by specifying which nuclides that one wants to rescale the density of. These can be one or more nuclides belonging to one or more materials. However, the scaling factor is common to them all since it is the global factor 1ζ.

2. Provide initial guesses for $\psi_{\zeta,0}$ and ζ_0. In SCONE, by default, the initial guess for ζ_0 is 1.

3. Perform transport as usual, by saving the fission neutrons produced as the source of the successive iteration (n + 1). Note that in this case, the number of fission neutrons produced is not scaled by k_{eff}. Every time macroscopic cross sections are looked up, the atomic densities of the nuclides that belong to m are scaled by a factor of $1\zeta^{(n)}$.

4. During transport, tally the separate reaction rate contributions of m and $\overline{m}$ and leakage in order to estimate ζ as shown in Eq. (7). In SCONE, the collision flux estimator is used for this purpose, and an analog estimator is used for leakage.

5. The calculation should converge to the fundamental ζ and fission source during the inactive cycles and conclude at the end of the active cycles, while ζ converges to its critical value k_{eff}, which can be estimated in parallel with the usual implicit estimator given by

$$k_{\mathrm{eff}} = \frac{\int_{V_m,\Omega,E} F\psi \mathrm{dV d\Omega dE}}{\int_{V,\Omega,E} (L + T - S)\psi \mathrm{dV d\Omega dE}}. \tag{8}$$

which should converge to 1. It should be noticed that k_{eff} can be calculated during a ζ simulation to obtain additional information, but it is not used.

While the Monte Carlo algorithm is easy to adapt from k to ζ calculations, there are some subtleties that must be reported. For a start, the problem must be well posed in order to have a solution: ζ must have a positive value to be physically meaningful. It is clear from the estimator definition in Eq. (7) that this might not always be the case. Physically, ζ represents the ratio between the net neutron production in m over the net neutron losses due

to leakage and $\overline{m}$. In the case, for example, of a boron search in a PWR, the numerator of Eq. (7) would be negative since boron is a net absorber. If the neutron losses in the rest of the system were more prominent than neutron production, the denominator would be positive and ζ negative. This example represents the case of a reactor that is subcritical in the absence of boron. There is no amount of boron that could make the reactor critical unless the sign of the balance equation is reversed. Care must be taken when defining the problem to ensure that the search is physically meaningful.

From an implementation point of view, it must be noticed that certain standard acceleration techniques, such as precalculating the majorant cross section at the beginning of the calculation when using delta tracking, cannot be used naively. This is the case because the material cross sections depend on ζ and change at each iteration. This implies that the majorant cross section should be recalculated at the beginning of each iteration. To circumvent this issue, all the models tested in this work were run with surface tracking except for the three-dimensional (3D) PWR BEAVRS [15]. Fortunately, when simulating a large number of particles, recalculating the majorant at the beginning of each cycle is a negligible overhead.

Despite the similarities with the k-eigenvalue power iteration, the two algorithms do not necessarily converge at the same rate. In the following numerical examples, some cases were slow to converge. In SCONE, during transport, it is conventional to use the cumulative estimate of the eigenvalue at the end of the previous iteration rather than the cyclewise one. This is the case also during the inactive cycles, i.e., during convergence. Clearly, the cumulative estimate of the eigenvalue is affected by the initial guess, and it takes longer to converge to the fundamental eigenvalue than the cyclewise one. In order to accelerate convergence, tally flushing was implemented. This consists of resetting the cumulative estimate of the eigenvalue every fixed number of cycles, chosen by the user.

II.C. Deterministic Implementation

The deterministic code used to solve the ζ eigenproblem in this work is TEST [4], which implements the Cartesian, one-dimensional (1D) finite-difference P_N and S_N multigroup methods. Contrarily to matrix-free codes, e.g., those employing the transport sweep, TEST assembles the discretized operators in Eq. (1), constructs the eigenvalue problem of interest, and solves for the eigenpairs.

Since the ζ scaling can be introduced in any region of the system, possibly even in a material initially absent (e.g., boron in the coolant regions), its associated eigenvalue problem is constructed by first assembling the discretized operators for the regions $\overline{m}$ and then by creating a new material and the corresponding operators, which will be scaled by ζ, for the regions m. Then, the eigenpairs can be estimated using either the power iteration method, if the user is interested in the fundamental eigenvalue only, or Krylov-based methods, if the user needs the higher-order eigenvalues.

Contrarily to Monte Carlo, which builds its continuous-energy data on the fly, deterministic codes should know in advance the multigroup constants featuring m and $\overline{m}$, which are usually precomputed using either lattice physics or Monte Carlo itself. In either case, the spatial and energy self-shielding calculations are carried out for a reference m density, disregarding the spectral variations induced by the ζ scaling. Depending on the magnitude of these spectral effects, this might make the naïve deterministic implementation of ζ (i.e., scaling multigroup constants generated for a -reference m density) inaccurate. This nonlinear behavior is similar to the thermal feedback that couples the fission power produced in the system with its temperature field through the Doppler effect. Therefore, as is commonly done to treat the Doppler effect in multiphysics codes, this issue is usually tackled by tabulating cross sections over a range of the parameters of interest. An example iterative procedure for a critical search is as follows:

1. *Off-line calculations*: Compute the group constants (GCs) parameterized with respect to the atomic density featuring the isotopes of interest.

2. O*n-line calculations*:

   a. Solve the k-eigenvalue problem with a starting set of GCs, evaluated at a certain density of the isotope of interest in m, say, $N_{m,0}$.

   b. Update the value of the density according to a root-finding algorithm, such as the bisection method or the Newton-Raphson algorithm.

   c. Interpolate the GCs at the new density $N_m^{(n)}$.

   d. Solve the k-eigenvalue problem with the interpolated constants relative to density $N_m^{(n)}$.

e. Iterate steps b, c, and d until a convergence criterion is satisfied, for instance, $|k-1|<\varepsilon$.

Similarly, one could employ an iterative procedure involving ζ rather than k-eigenvalue calculations. In this case, ζ can be used to guide the iterative process until no density scaling is further needed, as follows:

1. *Off-line calculations*: Compute the GCs parameterized with respect to the atomic density featuring the isotopes of interest, as above.

2. *On-line calculations*:

a. Solve the ζ-eigenvalue problem with a starting set of GCs, evaluated at a certain density of the isotope of interest in m, say, $N_{m,0}$.

b. Interpolate the GCs at the density $N_m^{(n)}=\frac{N_m^{(n-1)}}{\zeta^{(n-1)}}$.

c. Solve the ζ-eigenvalue problem with the interpolated constants relative to density $N_m^{(n)}$.

d. Iterate steps b, c, and d until a convergence criterion is satisfied, for instance, $|\zeta^{(n)}-1|<\varepsilon$.

A numerical comparison between the convergence rates of the two approaches is shown in Sec. IV.

A more sophisticated strategy would be embedding the interpolation process directly into the power iteration method, thus reducing the number of calculations. This approach is very similar to the fixed-point algorithm discussed in Sec. II.A and implemented in Monte Carlo, and it could be also coupled with a Krylov solver, employing a less stringent tolerance on the inner iterations. These approaches are more invasive to implement and as a consequence were not developed as part of this paper. However, they will be treated in future work.

Another challenging aspect affecting the solution of the ζ equation with a deterministic solver is related to spatial self-shielding. While scaling the atomic density, ζ modifies the mean free path (MFP) of the particles streaming in the ζ regions containing m. If the spatial mesh is not fine enough, spatial self-shielding effects may not be sufficiently described, potentially leading to unphysical solutions. A straightforward expedient for this problem involves employing a very fine spatial mesh in regions containing isotope m, though this approach may be computationally inefficient should the isotope m be present in numerous regions of the system, as in the case of a moderator.

III. MONTE CARLO NUMERICAL RESULTS

The methodology presented in Sec. II was applied to some test cases. All the calculations presented in the following used the JEFF-3.1.1 cross-section library and were run on 100 threads of an AMD Ryzen Threadripper PRO 5995WX node.

III.A. The POPSY Benchmark

The first test case was the critical benchmark POPSY. This consists of a sphere of plutonium surrounded by a uranium reflector (the exact benchmark specifications can be found in [16]). Here, the atom density of ^{239}Pu was initially set to 0.03 atom/(b·cm), and the ζ algorithm was used to find the critical density of ^{239}Pu. It must be emphasized that the model should converge to the correct critical density regardless of the initial density specified; however, the choice of initial density might affect the number of iterations required to converge. The initial configuration had a k_{eff} of 0.87964 ± 6 pcm. For the ζ calculation, a neutron population of 50 000 neutrons/cycle was run, with 200 inactive cycles and 5000 active cycles. While the problem converged to the correct eigenvalue and fission source relatively quickly, such a large number of active cycles were necessary to obtain an acceptable uncertainty on the value of ζ. The ζ calculation took around 2.5 min to complete, and the results are shown in Table 1. The uncertainty in the ^{239}Pu critical atomic density was propagated from the uncertainty in ζ; k_{eff} was also estimated during the ζ simulation. In this case, the standard deviation on ζ is larger than that on k_{eff}. This is the case because k_{eff} is an integral parameter, while ζ includes local contributions tallied over m and $\overline{m}$.

TABLE 1

ζ Calculation results for a critical ^{239}Pu search in POPSY

ζ	k_{eff}	Critical Density [atom/(b·cm)]
$0.82225\pm1.2\times10^{-4}$	0.99994 ± 8 pcm	$3.64853\times10^{-2}\pm5.3\times10^{-6}$

If, for example, m was localized and had a relatively low reaction rate, the uncertainty of ζ could be large.

When performing a k-eigenvalue calculation with the density found by the ζ calculation, the system resulted critical with a k_{eff} of 0.99998 ± 7 pcm. It was confirmed that the energy spectra in the fuel and reflector regions also match within uncertainties between the two calculation types. The reference k-eigenvalue calculation was run with the same neutron population and settings, and it took the same time as the ζ calculation to run (about 2.5 min). This confirms that the ζ implementation does not add a significant run-time overhead.

III.B. Two-Dimensional PWR Pincell

A standard two-dimensional (2D) PWR pincell with 3.5 wt% enriched UO_2 and initial water density of 1.0 g/cm^3 [corresponding to atomic densities of 6.68×10^{-3} and 3.34×10^{-3} atoms/(b·cm), respectively, for ^{1}H and ^{16}O] was also simulated. In this case, the water density was adjusted to obtain criticality.

As known from reactor physics, this problem should have two solutions: A PWR lattice can be critical in an undermoderated state and in an overmoderated state. However, the Monte Carlo ζ algorithm can converge only to one solution, equivalent in this case to the overmoderated scenario ($\zeta = 8.757 \times 10^{-3} \pm 4.7 \times 10^{-6}$). This is the case almost regardless of the initial guess ζ_0 and of the initial water density. The only exception is when the choice of the initial water density, scaled by $1\zeta_0$, results in a system that is subcritical and undermoderated. In that situation, the estimate of ζ results in a negative value because both water and fuel are net absorbers. This results in Eq. (7) returning a negative value. Then, the Monte Carlo calculation is interrupted. It must be noticed that this does not happen when the system is subcritical and overmoderated. In that case, water is a net absorber, but the fuel is a net producer; hence, Eq. (7) returns a positive value. At the time of writing, analysis is underway to characterize the convergence features of the numerical scheme implemented. Future work will focus on alternative formulations of the ζ-eigenvalue equation, which might allow converging to the undermoderated solution.

k_{inf} as a function of the moderator-to-fuel ratio [or hydrogen to heavy metal ratio (H/HM)] is shown in Fig. 1, together with the stable ζ solution. The curve was produced by performing a sweep of k-eigenvalue calculations with varied water density.

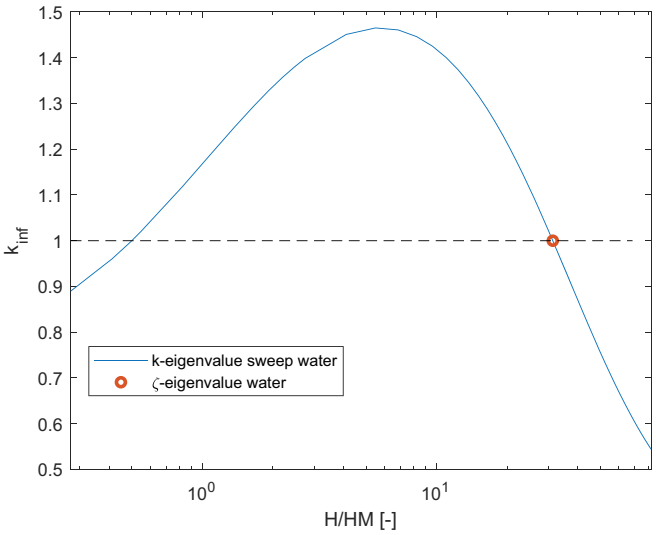


Fig. 1. k_{inf} as a function of the moderator-to-fuel ratio, calculated as number of hydrogen atoms (H) over number of heavy metal atoms (HM) in the system, for the 2D PWR pincell.

III.C. Three-Dimensional PWR Assembly

A 3D PWR assembly with radially reflective boundary conditions was also modeled. Its radial cross section is shown in Fig. 2. It is a standard 17×17 lattice with a pitch of 1.265 cm, fuel rod radius of 0.475 cm, and guide tube radius of 0.613 cm. The coolant material is water at 600 K, with a density of 0.6939 g/cm^3. The fuel is enriched 3.5% enriched UO_2 at 900 K, except for the gadolinium pins (in bright green in the picture) where the enrichment is 2.7%. Two different types of searches were tested on this model: a critical gadolinium search and a critical boron search where the boron concentration had a variable axial distribution.

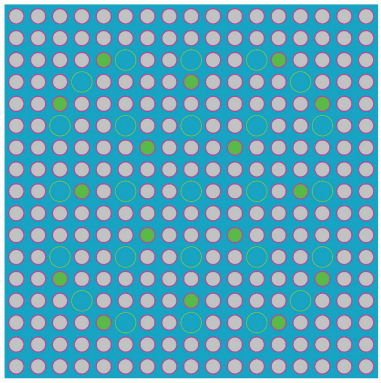


Fig. 2. Radial geometry cross section of the 3D PWR assembly model. The fuel pins containing gadolinium are in bright green.

III.C.1. Critical Gadolinium Search

For this case, the assembly was limited to a 100-cm height for computational efficiency. For this model, the search consisted in finding the critical densities of the six gadolinium isotopes listed in Table 2. Their corresponding initial atomic density values are also reported. The k_{eff} of the model with the starting gadolinium densities is 1.08718 ± 8 pcm.

The ζ calculation was performed using 1 million particles/cycle, 100 inactive cycles, and 100 active cycles. It produced a ζ value of $9.407 \times 10^{-2} \pm 4.4 \times 10^{-5}$ and a k_{eff} of 0.99999 ± 9 pcm. The resulting gadolinium critical densities are listed in Table 2 with uncertainties propagated from the ζ value.

The cyclewise values of ζ, k_{eff}, and Shannon entropy produced in the ζ calculation during the inactive cycles are plotted in Fig. 3. The Shannon entropy was calculated in two separate ways: using a purely axial discretization with 20 axial bins, and a radial pin-by-pin mesh. Here, tally flushing was applied to ζ every 10 cycles in order to facilitate convergence. The effect of this feature can be observed by the steep drops in ζ and k_{eff} that occur as soon as the cumulative value of ζ, used to scale the densities during transport is reset.

This ζ calculation converges after ~60 cycles. In this case, a frequency of 10 cycles was chosen for tally flushing to clearly show its effect on the convergence behavior. However, it is obvious that flushing the eigenvalue tally every inactive cycle would result in the fastest convergence. When this is done, convergence occurs after ~30 inactive cycles. On the other hand, an equivalent k-eigenvalue calculation (using the cumulative estimator during the inactive cycles, i.e., without flushing) needs ~20 cycles to converge. Clearly, the convergence properties of ζ calculations can be extremely different from those of k-eigenvalue calculations.

This is confirmed by the Shannon entropy plot in Fig. 3. For instance, in previous k-eigenvalue calculations of similar 3D PWR assembly problems, the axial Shannon entropy was found to converge more slowly than the radial entropy, which converges almost immediately [17]. This can also be seen in Fig. 5 is in Sec. III.C.2 per author. This is simply the case because the model is much longer in its axial dimension than in the radial one. Additionally, in k-eigenvalue calculations, k_{eff} is used to scale the fission source uniformly in all fissile materials, so it does not affect the fission source distribution. In this case, ζ is scaling local densities, thus impacting the fission source. Because ζ and the fission source are coupled and converge simultaneously, their overall convergence rate is determined by the slower of the two. In this problem, the axial fission source converges as fast as an equivalent k-eigenvalue calculation since the gadolinium concentration is axially uniform. On the other hand, the radial fission source converges at the same rate as ζ and k_{eff}, as it is strongly affected by the local variations in gadolinium density.

TABLE 2

Gadolinium isotopes used for the critical search in the 3D PWR model, and their starting and critical Densities obtained by the ζ calculation*

Isotope	Starting Density	Critical Density
^{154}Gd	4.6173×10^{-5}	$4.9084 \times 10^{-4} \pm 1.7 \times 10^{-9}$
^{155}Gd	2.9711×10^{-4}	$3.1584 \times 10^{-3} \pm 1.3 \times 10^{-8}$
^{156}Gd	4.1355×10^{-4}	$4.3962 \times 10^{-3} \pm 1.8 \times 10^{-8}$
^{157}Gd	3.1518×10^{-4}	$3.3505 \times 10^{-3} \pm 1.4 \times 10^{-8}$
^{158}Gd	4.9786×10^{-4}	$5.2924 \times 10^{-3} \pm 2.2 \times 10^{-8}$
^{160}Gd	4.3764×10^{-4}	$4.6399 \times 10^{-3} \pm 1.9 \times 10^{-8}$

*All densities are in atoms/(b·cm).

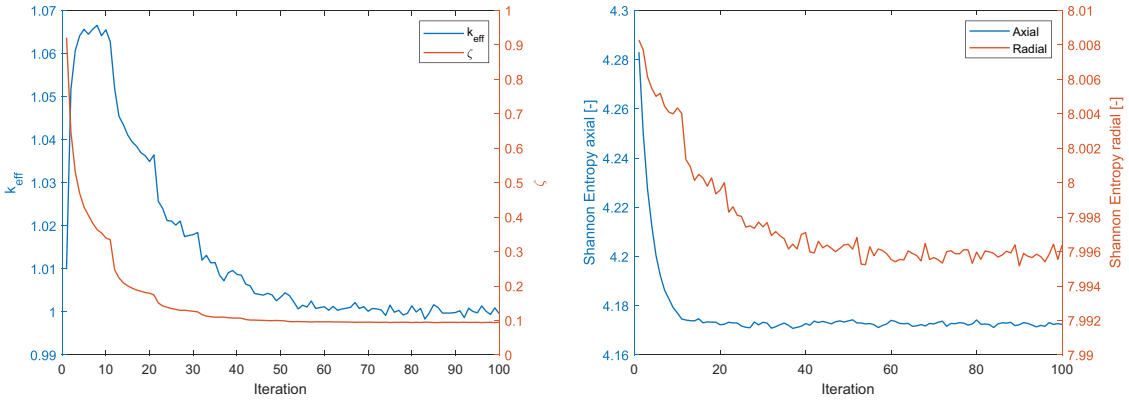


Fig. 3. Convergence of the gadolinium search in the 3D PWR model: k_{eff} and ζ (left) and Shannon entropy (right) as a function of the inactive cycle number.

Similarly to what is observed in the other cases, the final relative error in ζ is larger than that in k_{eff}. However, during convergence, k_{eff} seems to be subject to large cycle-to-cycle fluctuations. Since ζ is scaling the density of gadolinium, which is a strong neutron poison, the reason might be that a small fluctuation in ζ produces a large fluctuation in k_{eff}. Thoroughly studying the convergence properties of ζ is outside the scope of this paper but will be the subject of future work.

For this model, a conventional iterative search based on k-eigenvalue calculations was performed to compare performances. The search algorithm applied was the *regula falsi* method detailed in [18], with the convergence criterion given by

$$|k_{\text{eff}} \pm 2\sigma| < \epsilon\,. \tag{9}$$

This method, which is a hybrid between the bisection and secant methods, requires two starting points that bracket the searched optimum. These must be user defined.

In [18], some short cuts to accelerate the search were implemented. For example, the converged fission source from one iteration was used as the initial guess for the successive one to accelerate convergence. Additionally, a calculation was automatically interrupted if Eq. (9) was definitely not satisfied. In this application, fission source convergence was not a limiting factor since it was shown that 20 inactive cycles were sufficient to converge the fission source in a k-eigenvalue calculation. However, the number of active cycles was adjusted according to how far the system was from criticality. The search started with a relatively low number of active cycles to scope the search space with quick calculations. Then, if in iteration i the standard deviation σ_i was larger than $\frac{|k_{\text{eff},i}-1|}{10}$, the number of active cycles N_{i+1} to be used in the next iteration was adjusted. The scaling $\sigma \propto 1\sqrt{(N)}$ was applied using as a target $\sigma_{target,i+1} = \frac{|k_{\text{eff},i}-1|}{10}$. The factor 10 was chosen arbitrarily, and an optimization of the iterative search method was not performed in this work.

Here, the convergence criterion adopted was $\epsilon = 20$ pcm. The initial densities were those reported in Table 2. The initial bracketing scaling values were set to 1.0 and 0.05. A summary of the search results is presented in Table 3. The search converged in five iterations, totaling 140 inactive cycles and 417 active cycles. On the other hand, a ζ calculation could obtain the same result using 30 inactive cycles and 100 active cycles. The final ζ value was 9.375×10^{-2}, which is in good agreement with the ζ calculation.

TABLE 3

Iterative criticality search on the gadolinium density, where each iteration used 1 million particles/cycle and 20 inactive cycles

Iteration	Active Cycles	ζ	k_{eff}
0	10	1.0	1.08722 ± 24 pcm
	10	0.05	0.96759 ± 39 pcm
1	10	0.0673	0.98396 ± 26 pcm
2	10	0.1019	1.00367 ± 17 pcm
3	10	0.0922	0.99902 ± 16 pcm
4	27	0.0947	1.00036 ± 13 pcm
5	340	0.0937	0.99993 ± 4 pcm

III.C.2. Critical Boron Search

A similar PWR assembly model was tested to perform a critical boron search. Here, the assembly was 3.6 m tall and divided into 10 axial nodes, each with a different coolant density. The initial ^{10}B concentration axial profile is shown in Fig. 4; ^{11}B had the same axial profile but was 4.02 times more abundant. In the ζ calculation, the densities of ^{10}B and ^{11}B at each axial node were scaled while preserving the axial profile.

A ζ calculation was performed with 1 million particles/cycle, 200 inactive cycles, and 100 active cycles. The convergence of the eigenvalue and fission source during the inactive cycles is shown in Fig. 5. Contrarily to the previous case, tally flushing was not used as the convergence rate was observed to be similar to that of the equivalent k-eigenvalue calculation. Here, the convergence bottleneck was clearly the axial fission source rather than ζ. The radial entropy,

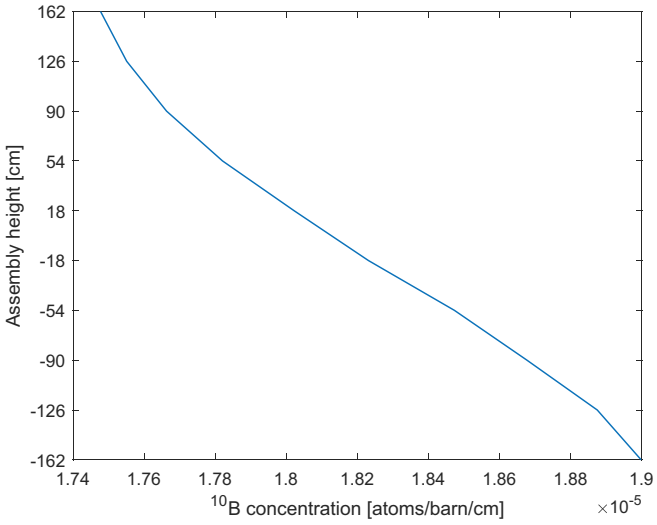


Fig. 4. Initial ^{10}B concentration axial profile.

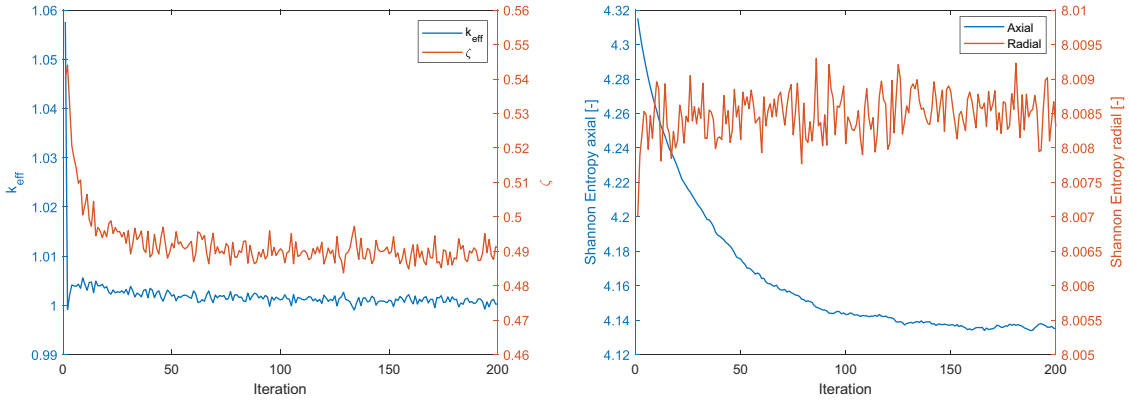


Fig. 5. Convergence of the boron search in the 3D PWR model: k_{eff} and ζ (left) and Shannon entropy (right) as a function of the inactive cycle number.

unaffected by the axial boron density variations, converges as quickly as it would be expected for a k-eigenvalue calculation. The convergence behavior of ζ is in stark contrast with what was observed for the critical gadolinium search. The differences between the two cases will guide the focus of the convergence analysis subject of future work. In this case the starting boron density was closer to the critical one, boron was homogeneously distributed in the coolant rather than being localized, and boron is not as strong an absorber as gadolinium. At the moment of writing, it is thought that these parameters affect the convergence rate of ζ. The calculated value of ζ was $0.4906 \pm 2.2 \times 10^{-4}$, corresponding to a k_{eff} equal to 0.99996 ± 6 pcm.

Once again, an iterative search was performed as a performance comparison. The same methodology as before was applied. The only exception is that in this case, in order to reduce the number of inactive cycles needed, the converged fission source for the critical case was provided as an initial guess for all calculations, and 30 inactive cycles were used to allow for small adjustments. Each cycle was run with 1 million particles, and the number of active cycles was varied as before. The initial boron densities were those used in the ζ calculation, the bracketing values of ζ were chosen to be 20 and 0.1, and ϵ was 20 pcm. This time the iterative search took longer to converge: 18 iterations (for a total of 20 calculations) counting 600 inactive cycles and 806 active cycles. This is generous accounting since in a real application, the first calculation would have used 200 inactive cycles rather than 30. In comparison, the ζ calculation took 200 inactive cycles and 100 active cycles. The ζ found by the iterative search was 0.4908, corresponding to a k_{eff} of 1.00010 ± 4 pcm.

III.D. BEAVRS

The last Monte Carlo model investigated in this work is the PWR BEAVRS [15], for which the critical boron concentration was searched for. A radial cross section of the 3D geometry is shown in Fig. 6, together with the fission rate radial profile calculated with the critical boron concentration.

For this model, a ζ calculation was run to find the critical densities of ^{10}B and ^{11}B dissolved in the reactor coolant. The k_{eff} obtained for the starting densities, i.e., 7.9714×10^{-6} and 3.2247×10^{-5} atoms/(b·cm) for ^{10}B and ^{11}B, respectively, is 0.99838 ± 5 pcm. ζ was found to be $1.01356 \pm 2.3 \times 10^{-4}$, which corresponds to boron densities of $7.86475 \times 10^{-6} \pm 1.8 \times 10^{-9}$ and $3.18156 \times 10^{-5} \pm 7.1 \times 10^{-9}$ atoms/(b·cm). An iterative criticality search was not performed for this case since this would have been too computationally expensive. A reference k-eigenvalue calculation was run with the critical boron densities, obtaining a value of k_{eff} equal to 1.00004 ± 3 pcm. These simulations were run on 100 parallel threads with 1 million particles/cycle, 200 inactive cycles, and 50 active cycles.

Fig. 7 shows how the model converges during the 200 inactive cycles. Because they were shown to be enough to converge a k-eigenvalue calculation of 3D BEAVRS, 200 cycles were chosen [19]. From Fig. 7 it can be seen that consistently with the previous case, the eigenvalues and the fission source converge at the same rate for the two calculation types

IV. DETERMINISTIC NUMERICAL RESULTS

In this section, the numerical results obtained using the TEST deterministic code are presented, focusing on the specific challenges arising in the deterministic framework.

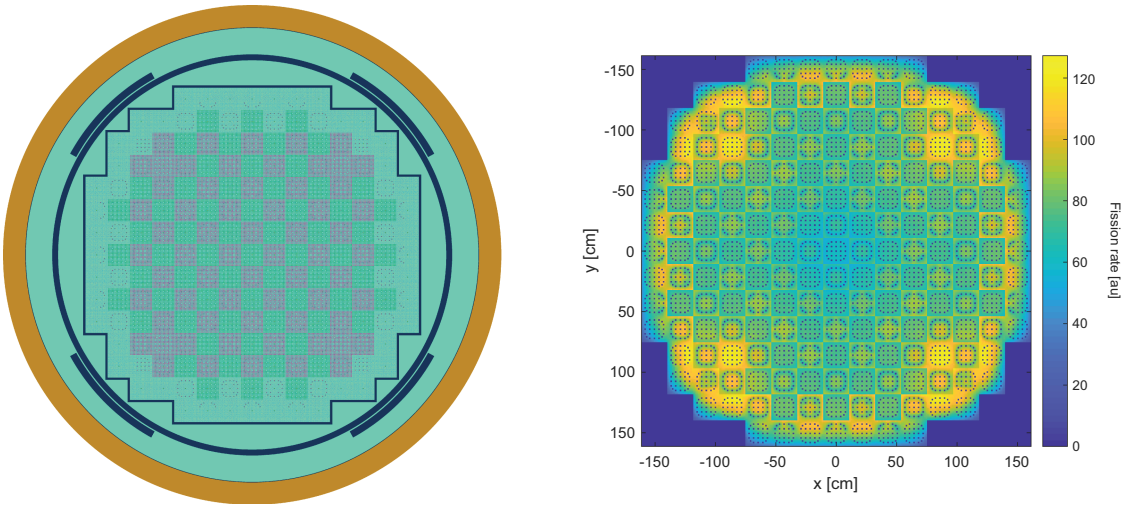


Fig. 6. Radial geometry cross section (left) and fission rate profile (right) of the BEAVRS model.

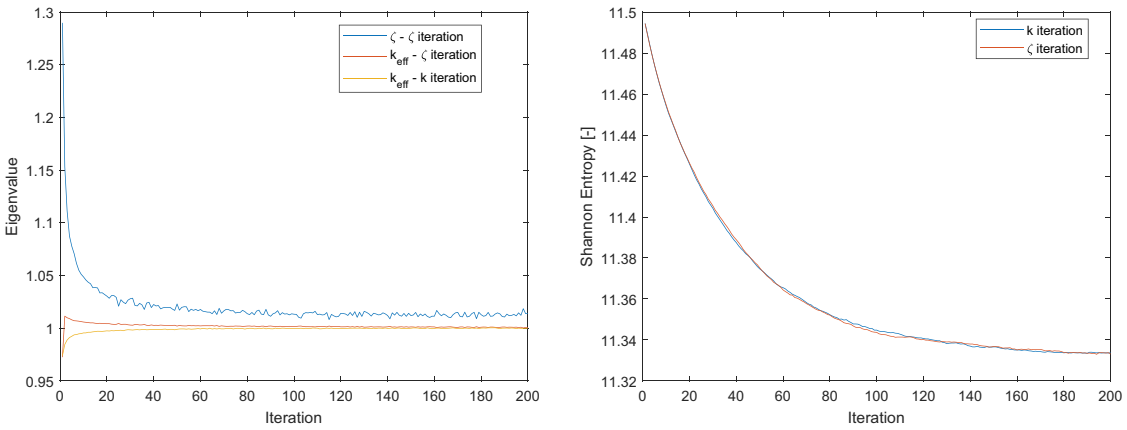
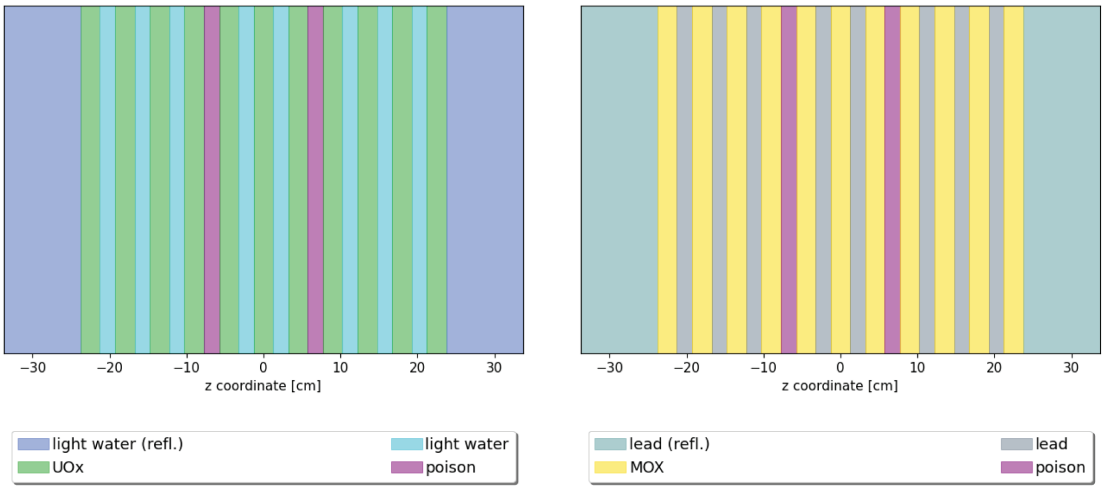


Fig. 7. Convergence of the BEAVRS model: k_{eff} and ζ (left) and Shannon entropy (right) as a function of the inactive cycle number. The k_{eff} and the Shannon entropy are plotted both for the ζ calculation and for the k-eigenvalue calculation with the critical boron densities.

IV.A. The Effect of GC Parameterization

As already pointed out in [4], the direct solution of the ζ-eigenvalue equation in a deterministic frame does not take into account the second-order effects on the GCs due to the density scaling. In the following, the impact of this approximation is investigated with numerical examples, focusing on scaling the absorber density in the thermal (left) and fast (right) slab systems sketched in Fig. 8. Both systems were 67.5 cm wide, with the reflector being 10 cm on each side and alternating layers of fuel (2.5 cm thick) with coolant and poison layers (all 2.0 cm thick). In the thermal system, designed to be like a light water reactor (LWR), the coolant and the reflector are water with nominal density 1.0 g/cm^3, and the fuel is 3% enriched UO_2. The fast system, designed to be like a lead fast reactor (LFR), has the same geometry of the thermal one, but water is replaced with lead, and the fuel is ~13.84% enriched mixed oxide. The absorbers used in the two systems are boron and boron carbide, respectively. All the calculations presented in this paragraph were performed using a six-group P_5 model for the LFR-like system and an eight-group P_5 model for the LWR-like system. The energy structure for the LWR model consisted of eight groups equally spaced in lethargy between 1.0 $\times 10^{-11}$ and 20.0 MeV The six-group grid for the LFR was obtained by collapsing the three thermal groups of the eight-group grid. This was done to avoid high statistical uncertainties in the GSs, generated with Monte Carlo, at unimportant energies for the fast reactor



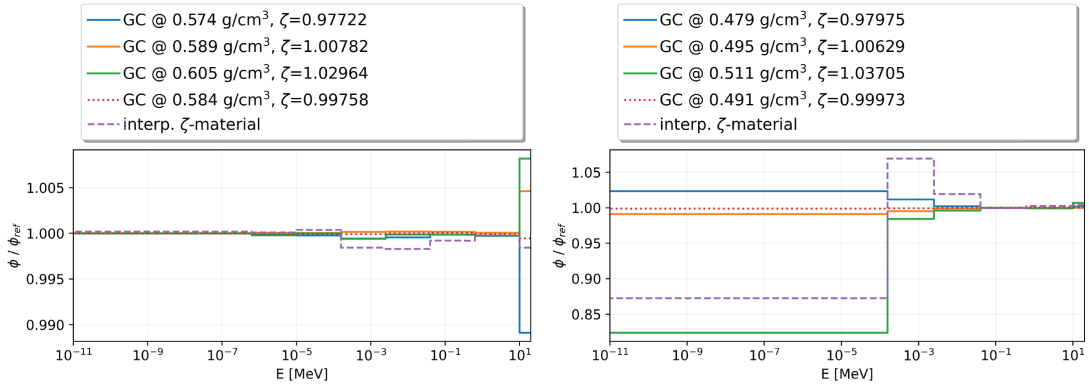
(a) *LWR-like system.* (b) *LFR-like slab.*

Fig. 8. Sketch of the 1D heterogeneous systems considered for the deterministic ζ calculation of the critical poison concentration. The ζ regions (m) are indicated in purple.

model. A distinct set of GSs was generated for each material, with the only exceptions being the reflector and coolant, which were treated separately despite sharing the same composition.

As discussed in Sec. II.C, the most natural strategy to take into account the density spectral effects in a ζ calculation consists of constructing a parametric set of GCs, interpolating the GCs at the density suggested by the ζ calculation, and iterating until a convergence criterion is satisfied. For physically accurate results, the interpolation should be carried out for all the GCs (describing both m and $\overline{m}$) in the model.

Fig. 9 shows the impact of different GC treatments as the ratio between the flux spectra of the critical systems obtained with and without GC interpolation. Here, the reference neutron spectra were obtained by interpolating GCs within the ζ iterative procedure described above. The naïve ζ calculations were carried out scaling a fixed set of GCs. Different cases, with GCs calculated at different starting densities and producing different ζ eigenvalues, were tested. The effect of interpolating only the GCs of the ζ-material is also investigated. For the LWR-like case, the spectral ratio is very close to unity for each set of GCs, except in the fast group, where the discrepancies are slightly larger. The largest deviation is the dashed line, which represents the case where only the GCs of the ζ-material are interpolated. For the LFR-like system, the discrepancies are larger, probably due to the



(a) *Spectral variation for the LWR-like slab.* (b) *Spectral variation for the LFR-like slab.*

Fig. 9. Ratio of the flux spectra obtained with ζ calculations that scale a fixed set of GCs (calculated at different starting densities) and the reference spectrum, obtained by interpolating the GCs for a thermal (left) and a fast (right) system.

moderation provided by the boron carbide poison in a fast spectrum system.

The importance of the spectral effects introduced by the use of consistent GCs clearly depends on several factors, such as the model simulated, the choice of ζ-material (m) and the choice of reference density. The simple numerical examples provided show that these effects can be significant, even when the reference density is extremely close to the critical one. As a consequence, using a numerical scheme interpolating GCs is advised, unless the user was confident that the ζ-material does not strongly affect the spectrum of the system, or was looking for rough estimates rather than accurate results. While one might be tempted to only interpolate the GCs of the ζ-material, this can also lead to errors.

IV.B. Spatial Discretization Effects

As mentioned previously, the spatial self-shielding effects due to the ζ density scaling require the adoption of a sufficiently fine mesh. This can be observed from the curves shown in Fig. 10, generated with a k-eigenvalue sweep for different moderator densities in a LWR-like slab. This model is similar to the one sketched in Fig. 8 (left), but replacing the absorbers with the moderator and increasing the number of fuel layers up to 17. Here, reference results were produced with continuous-energy Monte Carlo. The deterministic curves were realized with a seven-group, P_5 model with different mesh sizes, expressed in centimeters.

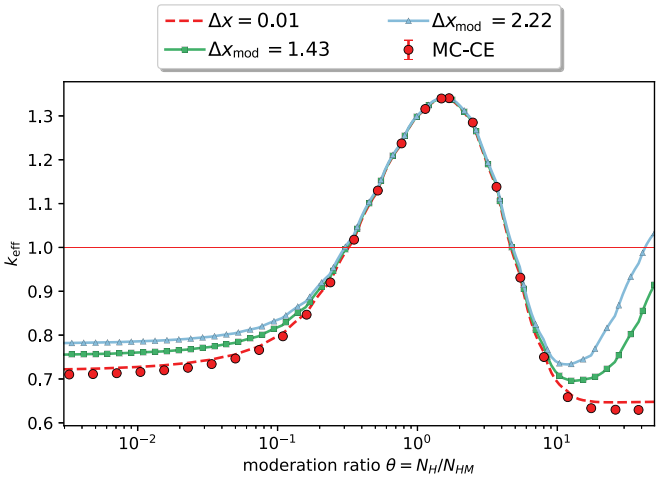


Fig. 10. k_{eff} as a function of the moderator-to-fuel ratio, calculated as number of hydrogen atoms (H) over number of HM in the system, for a LWR-like slab similar to the one sketched in Fig. 8 (left) for different mesh sizes in centimeters.

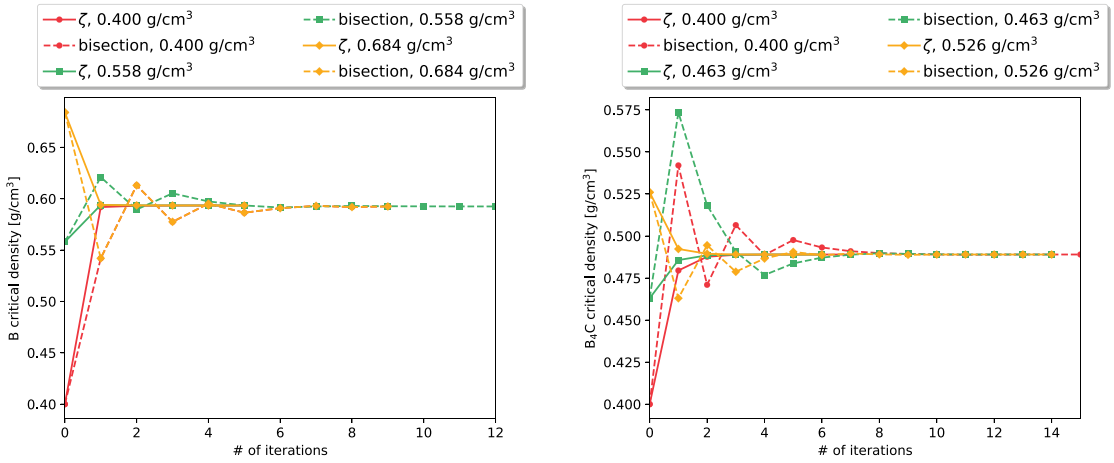
The calculations were carried out considering either a uniformly spaced mesh (case $\Delta x = 0.01$ in the figure) or an unstructured mesh having different mesh sizes according to the mfp featured in the region. In this last case, Δx_{mod} indicates the mesh size in the moderator. When the mesh size in the moderator is not sufficiently small, the curve exhibits a third intersection with the axis $k_{\text{eff}} = 1$, which is a spurious critical configuration leading to unphysical moderator densities and caused by numerical effects. If the corresponding deterministic ζ-eigenvalue problem was solved, e.g., with a Krylov solver, three fundamental eigenvalues would appear, as shown in [4]. When ensuring that the mesh size is smaller than the minimum mfp featured in the system for moderator density, the curve follows the physical trend of the reference Monte Carlo results.

IV.C. Performances of the ζ-Based Interpolation Scheme

This section presents a comparison between the computational performances of the ζ-based interpolation scheme described in Sec. II.C and a simple bisection scheme based on k-eigenvalue calculations. Figs. 11 and 12, respectively, show the convergence of the critical density and of the relative error in critical density for the LWR-like (left) and LFR-like (right) slabs previously described. As before, the poison density is being scaled. Several tests, with different starting densities, were performed. In all cases, the ζ-based interpolation scheme converges after two to four iterations, offering a faster convergence compared to the bisection method regardless of the density value used to initialize the iterative process. This last observation suggests that although the spectral effects may not be represented consistently with the GCs used in the first ζ iteration, the density correction $1/\zeta^{(0)}$ retrieved solving the first ζ eigenproblem accounts for most of the difference with the critical model.

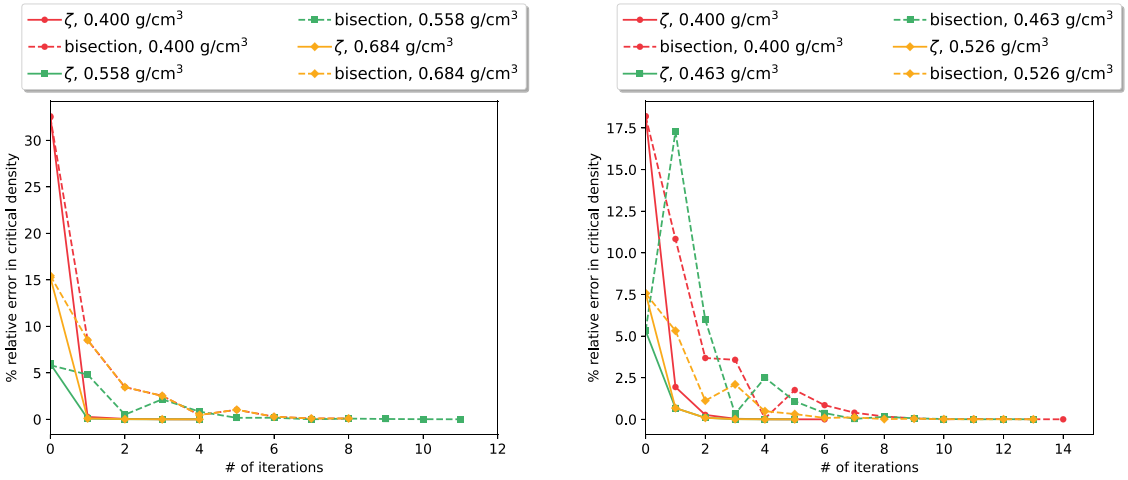
The scenario where more than one physical solution is present, i.e., when scaling the density of a moderator, was not tackled in this section. In that case, both the k- and ζ-eigenvalue iterative schemes might struggle to converge. While solving the ζ equation with a Krylov solver would give the advantage of finding all the real ζ eigenvalues, setting a criterion to choose which ζ to converge to while interpolating GCs each iteration is not trivial. This will be the subject of future investigations.

Another subtlety worth mentioning concerns the choice of an appropriate convergence criterion for ζ. While the choice of the convergence criterion is driven by physical



(a) *LWR-like system.* (b) *LFR-like system.*

Fig. 11. Convergence of the critical density for the ζ-driven interpolation scheme and the bisection-driven interpolation scheme.



(a) *LWR-like slab.* (b) *LFR-like slab.*

Fig. 12. Convergence of the relative error for the ζ-driven interpolation scheme and the bisection-driven interpolation scheme.

intuition in the case of k_{eff}, such intuition does not exist for ζ. An acceptable error in ζ is such that it leads to an acceptable error in k_{eff}. However, quantifying it would require knowing the sensitivity of k_{eff} to ζ, which could be calculated knowing the sensitivity of k_{eff} to the density of the ζ-material m.

V. CONCLUSIONS AND FUTURE PERSPECTIVES

The ζ-eigenvalue equation is a formulation of the neutron transport equation that ensures criticality by scaling the density of the nuclides or materials of choice with the parameter ζ. In this paper, Monte Carlo and deterministic implementations of the ζ-eigenvalue equation are proposed, focusing on the main challenges posed by this new approach and proposing possible solutions.

The Monte Carlo algorithm consists of a fixed-point iteration scheme, where the particle source at each iteration is the fission source of the previous one. This allows the implementation to be trivial in Monte Carlo codes that are able to solve k-eigenvalue problems. Additionally, practical aspects of the deterministic implementation of the ζ formulation are tackled.

The Monte Carlo algorithm method was tested on some realistic numerical cases, like a critical gadolinium search in a 3D PWR assembly and a critical boron search in the BEAVRS reactor. The ζ algorithm could find the critical densities requested by the user in almost all cases.

The only exception was that of a poorly posed problem, for which the Monte Carlo ζ estimators return an unphysical negative value. This could happen, according to Eq. (7), when the material on which one is performing the search is a net absorber and there are net neutron losses in the rest of the system. This scenario occurred when we tried to search the critical water density in a 2D PWR pincell, and the starting water density was such that the system was subcritical and undermoderated. In this model, the ζ algorithm always converged to the overmoderated critical water density over the undermoderated one. Future work will focus on alternative formulations of the ζ equation, which might allow converging on the undermoderated solution.

From the other numerical tests, which all converged smoothly, some general observations can be drawn. The relative uncertainty of the calculated ζ, at the end of the active cycles, is always larger than that of k_{eff} from the same calculation. This is the case since estimating ζ requires obtaining some local reaction rates, while k_{eff} is an integral parameter. In the case where the material whose density is scaled by ζ is very localized, or has low cross sections, the ζ estimator can have substantial statistical noise.

The numerical convergence of some models was monitored during the inactive cycles. Despite the similarities in algorithms, there is no reason to assume that the ζ calculations will converge at the same speed as k-eigenvalue calculations for the same model. This was shown in the case of the 3D PWR assemblies with gadolinium. When the density search is performed on a localized material with extremely large cross sections, like gadolinium-bearing pins, convergence might be slow. In other cases, like the critical boron search in BEAVRS, ζ calculations and k-eigenvalue calculations converged exactly at the same rate. Overall, the run times of the ζ calculations and k-eigenvalue calculations are comparable if the same population settings are used.

A performance comparison with an iterative criticality search method was performed for the 3D PWR assembly cases with gadolinium and boron. While the iterative search method used could be further optimized, some reasonable starting conditions and acceleration techniques were used for the comparison. In all cases, the ζ calculation converged to the critical density of interest noticeably faster.

The second part of the paper was devoted to discussing the specific challenges pertaining to the deterministic implementation of the ζ equation. The first aspect analyzed in this work is the spectral effects induced by the density scaling, which requires the parameterization of the multigroup constants with respect to the density of the ζ-material in order to yield a consistent critical density adjustment. The other challenge is the necessity of a sufficiently fine spatial mesh to accurately represent spatial self-shielding effects. An algorithm combining ζ-eigenvalue calculations and the GC interpolation was also proposed and tested in the deterministic frame, showing better performances than a conventional iterative approach based on k-eigenvalue calculations. These findings suggest that this algorithm should be studied extensively in future works, analyzing different systems and the impact of the choice of the ζ-material on their performances. Tackling cases where multiple critical densities exist is also of interest. Moreover, the performance of the ζ-based iterative search could be further improved by embedding the GC interpolation directly into the power iteration method rather than applying it between successive calculations.

As a concluding remark, one could fairly argue that when a reactor is designed, there are several parameters to tune at once for criticality (e.g., burnable absorbers heterogeneously distributed across the core). However, the ζ-eigenvalue method investigated in this paper works only by scaling the density of one or several nuclides by the same factor. If one were to try to tune several parameters at once with different scaling factors, they would likely incur more severe convergence issues. As a consequence, in the context of reactor design, the method could mostly be useful in some specific situations. Examples include critical boron or xenon searches, e.g., in coupled transport-depletion calculations, and the determination of the critical concentration of fissile isotopes in molten salt reactors, where a continuous adjustment of the fuel salt composition is required. On the other hand, criticality safety problems tend to be conservatively simplified, to the point that finding criticality by scaling one parameter (such as isotopic density) could be of practical interest.

Acknowledgments

This work has been carried out under the auspices of the Italian National Group of Mathematical Physics (Gruppo Nazionale di Fisica Matematica, GNFM) of the National Institute of High Mathematics (Instituto Nazionale di Alta Matematica, INDAM).

Author Contributions

CRediT: **Valeria Raffuzzi:** Conceptualization, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing; **Nicolò Abrate:** Conceptualization, Methodology, Software, Visualization,

Writing – original draft, Writing – review & editing; **Sandra Dulla:** Conceptualization, Methodology, Writing – original draft, Writing – review & editing.

Disclosure Statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the EPSRC grant MaThRad [EP/W026899/1]. Peter Benie assisted through maintaining the Cambridge Nuclear Energy Group workstations. Valeria Raffuzzi acknowledges Paul Cosgrove for useful discussions. Italian National Group of Mathematical Physics.

ORCID

Valeria Raffuzzi http://orcid.org/0000-0001-7859-5620

References

1. D. E. Cullen et al., "Static and Dynamic Criticality: Are They Different?" UCRL-TR-201506, Lawrence Livermore National Laboratory (Nov. 2003).
2. B. Davison, *Neutron Transport Theory*, Clarendon Press, Oxford (1957).
3. Y. Ronen, D. Shalitin, and J. J. Wagschal, "A Useful Different Eigenvalue for the Transport Equation," *Trans. Am. Nucl. Soc.*, **24**, 474 (1976).
4. N. Abrate et al., "A Generalized Eigenvalue Formulation for Core-Design Applications," *Nucl. Sci. Eng.*, **197**, *8*, 2047 (cited on pp. 3–4, 8, 20, 23) (2023); https://doi.org/10.1080/00295639.2022.2134685.
5. D. Mancusi and A. Zoia, "Chaos in Eigenvalue Search Methods," *Ann. Nucl. Energy*, **112**, 354 (2018); https://doi.org/10.1016/j.anucene.2017.10.022.
6. A. Gandini, "Generalized Perturbation Theory (GPT) Methods. A Heuristic Approach," *Advances in Nuclear Science and Technology*, Vol. 19, J. Lewins and M. Becker, Eds. Springer, Boston (1987).
7. V. Raffuzzi, N. Abrate, and S. Dulla, "A Monte Carlo Implementation of the ζ-Eigenvalue Equation," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2025)*, Denver, Colorado, April 27-30, 2025, p. 1198, American Nuclear Society (Apr. 2025).
8. V. Raffuzzi, P. Cosgrove, and M. A. Kowalski, "Status of the SCONE Monte Carlo Neutron Transport Code," *Ann. Nucl. Energy* (2025) (submitted for publication).
9. J. Leppänen et al., "The Serpent Monte Carlo Code: Status, Development and Applications in 2013," *Ann. Nucl. Energy*, **82**, 142 (2015); https://doi.org/10.1016/j.anucene.2014.08.024.
10. A. Dall'Osso et al., "A Neutron Balance Approach in Critical Parameter Determination," *Ann. Nucl. Energy*, **35**, 9, 1686–1694 (2008); https://doi.org/10.1016/j.anucene.2008.01.019.
11. Y. Jo. & Z. Cho, "Inline critical boron concentration search with p-{CMFD} feedback in whole-core continuous-energy {Monte} {Carlo} simulation," *Ann. Nucl. Energy*, **120**, 402–409 (Oct. 2008); https://doi.org/10.1016/j.anucene.2018.05.050.
12. H. Belanger, D. Mancusi, and A. Zoia, "Unbiasedness and Optimization of Regional Weight Cancellation," *Phys. Rev. E*, **106**, *2*, 025302 (Aug. 2, 2022); https://doi.org/10.1103/PhysRevE.106.025302.
13. A. Zoia, E. Brun, and F. Malvagi, "Alpha Eigenvalue Calculations with TRIPOLI-4," *Ann. Nucl. Energy*, **63**, 276 (2014); https://doi.org/10.1016/j.anucene.2013.07.018.
14. B. C. Kiedrowski, "Evaluation of Computing C-Eigenvalues with Monte Carlo," *Trans. Am. Nucl. Soc.*, **106**, 512 (2012).
15. N. Horelik et al., "Benchmark for Evaluation and Validation of Reactor Simulations (BEAVRS), V1.0.1," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013)*,Sun Valley, Idaho, May 5-9, 2013, American Nuclear Society (2013).
16. International Criticality Safety Benchmark Evaluation Project (ICSBEP) website, Organisation for Economic Co-operation and Development, Nuclear Energy Agency (2023); https://www.oecd-nea.org/science/wpncs/icsbep/.
17. V. Raffuzzi et al., "Monte Carlo Source Convergence Acceleration by Hybrid Multigroup and Continuous Energy Neutron Transport," *Nucl. Sci. Eng.*, **197**, *3*, 364 (2023); https://doi.org/10.1080/00295639.2022.2107262.
18. D. F. Gill, B. R. Nease, and D. P. Griesheimer, "Movable Geometry and Eigenvalue Search Capability in the MC21 Monte Carlo Code," *Proc. Int. Conf. Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), Sun Valley, Idaho, May 5-9, 2013, American Nuclear Society (2013)*.
19. B. R. Herman, "Monte Carlo and Thermal Hydraulic Coupling Using Low-Order Nonlinear Diffusion Acceleration," Doctor of Science Thesis, Massachusetts Institute of Technology (Sep. 2014).