On the stability of algorithms for enforcing xenon equilibrium

P. Cosgrove, E. Shwageraus

Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, United Kingdom

Abstract

Enforcing xenon equilibrium is used in both Monte Carlo and deterministic neutron transport codes to aid in the simulation of burn-up. This paper shows that the 'inline' algorithm common to enforcing xenon equilibrium in Monte Carlo is unconditionally stable while fully converging neutronics before updating the xenon and iodine densities is not. This analysis includes different versions of the inline algorithm, accounting for different tallying strategies. The inline algorithm is also found to modestly accelerate the convergence of the neutronic solution relative to the standard source iteration algorithm without inline feedback. The theoretical predictions are supported by numerical results from both a simple diffusion solver and a Monte Carlo code. The analysis devised may be relevant to other inline feedback algorithms.

Keywords: Neutronics, Xenon, Stability, Burn-up, Monte Carlo

1. Introduction

Xenon-135 and its main precursor, iodine-135, strongly affect the behaviour and design of thermal nuclear reactors operating at any substantial power. This is due to xenon's extremely large thermal capture cross-section and its relative abundance as a fission product. Reactor designers and operators seek to know the steady-state spatial distribution of these two isotopes (or their equilibrium values) in order to determine their imposed reactivity penalty and the reactor dynamics which they may induce.

Enforcing xenon equilibrium is a common capability in neutron transport solvers used for reactor physics. The justification for introducing it has evolved over time: initially, it was introduced to allow Monte Carlo neutronics solvers to calculate xenon and iodine equilibrium concentrations without having to perform several expensive depletion steps to determine them (Griesheimer, 2010; Isotalo, 2013). It has since been used to avoid numerical instabilities occurring in spatially-large coupled neutronics/depletion calculations (Isotalo et al., 2013).

The naïve means of enforcing xenon equilibrium is to obtain a converged neutronics solution, using the reaction rate information to set the xenon and iodine densities, and

Email address: pmc55@cam.ac.uk (P. Cosgrove)

repeating the process until some degree of convergence is achieved. Although this method might be applied in some lattice physics codes, more sophisticated algorithms are applied in contemporary Monte Carlo and deterministic transport solvers.

The most common algorithm deployed in Monte Carlo codes for enforcing xenon equilibrium is known as the 'inline' algorithm (Griesheimer, 2010; Isotalo et al., 2013). Here the simulation accumulate fission and xenon absorption reaction rates in specified volumes (which tend to be homogeneous and correspond to the burnable regions used in depletion) during batches of the inactive cycles, while the fission source is still converging. After a specified number of batches, the xenon and iodine isotopic densities are updated and the next batch begins, with this continual update typically (although not necessarily) concluding alongside the inactive cycles.

One approach in deterministic methods, on the other hand, is to iterate between the Coarse-Mesh Finite Difference (CMFD) solver (which accelerates the transport calculation) (Smith, 1983; Smith and Rhodes III, 2002) and the xenon/iodine update. Once converged, a transport sweep is performed and the algorithm again iterates between CMFD and xenon/iodine updates (Kochunas et al., 2017a). This algorithm is not directly considered in the current paper, but is discussed in a more general case in Kochunas et al. (2017b). Another approach, used by STREAM (Choi and Lee, 2021), for example, performs the CMFD calculation, followed by transport, followed by multi-physics feedback and a cross-section update before repeating until convergence – this approach is also known to be only conditionally stable, requiring under-relaxation (with STREAM using a relaxation factor chosen empirically based on the feedback strength) (Choi et al., 2021).

While each of these algorithms are performed to stabilise depletion, they themselves are not necessarily stable. In particular, the latter iteration between CMFD and xenon/iodine densities is known to be unstable (Kochunas et al., 2017b), with the MPACT user manual highlighting that a relaxation between the two may be necessary to stabilise the iteration (Kochunas et al., 2013; MPACT Team, 2016, 2019). There has recently been some further analysis given to the naïve algorithm (and others) by Shen et al. (Shen and Kochunas, 2021; Shen et al., 2021). Meanwhile several other iteration schemes (albeit, for neutronics and thermal-hydraulics) have been examined by Hamilton et al. (2016) and Facchini et al. (2021), including Anderson acceleration, which appears to hold some promise as a stable multi-physics convergence acceleration scheme. However, to the authors' knowledge, there has not been further analysis of the Monte Carlo inline algorithm in the literature. The properties of the algorithm for enforcing xenon equilibrium may also have significant effect on the stability of isotopic depletion calculations; extending the analysis in Cosgrove et al. (2020) to cover such cases was the original motivation for the work described in this paper.

Through performing Fourier stability analysis and numerical experiments, this paper will demonstrate that the inline xenon algorithm and its variants are, indeed, stable, whereas the outer iteration between neutronics and xenon density can excite spatial instabilities.

2. Xenon equilibrium algorithms

The equations for xenon and iodine at their equilibrium values in a homogeneous medium are simple and standard in reactor physics (Duderstadt and Hamilton, 1976). It is normally assumed that iodine has a negligible neutron cross-section. At equilibrium, this gives:

$$I = \frac{\langle \gamma_{\mathrm{I}} \Sigma_{\mathrm{f}} \phi \rangle}{\lambda_{\mathrm{I}}} \,, \tag{1a}$$

$$X = \frac{\langle (\gamma_{\mathrm{I}} + \gamma_{\mathrm{X}}) \Sigma_{\mathrm{f}} \phi \rangle}{\lambda_{\mathrm{X}} + \langle \sigma_{\mathrm{X}} \phi \rangle} \,. \tag{1b}$$

Here I and X are the iodine and xenon densities, respectively, γ_{I} is the fission yield of iodine (commonly including that of its short-lived precursor, technetium-135), Σ_{f} is the macroscopic fission cross-section, ϕ is the scalar neutron flux, λ_{I} is iodine's decay constant, γ_{X} is xenon's direct fission yield, λ_{X} is xenon's decay constant, and σ_{X} is xenon's capture cross-section. The angular brackets denote an integration over phase space, typically over all neutron energies in a specified volume of fuel, divided by the the fuel's volume.

As discussed in the introduction, Eq. (1) can be coupled to a neutronics solver in several different ways when enforcing xenon equilibrium. Here we are concerned with a 'fully-converged' fixed-point iteration or by an 'inline' coupling scheme. Both of these will now be described more fully, as will common practice in whole-core deterministic methods which are not analysed in this paper.

The naïve, fully-converged fixed-point algorithm is a conventional scheme for converging different physics: a neutronics solution is solved to convergence, after which Eq. (1) uses the calculated reaction rates to update the xenon and iodine densities. Neutronics is solved to convergence once again and the process repeats. If unstable, this iteration can be augmented with an under-relaxation, using some relaxation factor α. This is shown in Algorithm 1. Although such an approach is relatively convenient for deterministic solvers which can easily re-converge given a previous fission source, the computational cost of iterating Monte Carlo calculations renders this algorithm impractical in the stochastic domain.

Inline coupling of neutronics and xenon implies converging the fission source and the xenon and iodine densities simultaneously: during one or several source iterations (or inactive cycles for Monte Carlo), the fission and xenon capture reaction rates are calculated in each region where it is desired to enforce xenon equilibrium. These reaction rates are used in Eq. (1) to update the local xenon and iodine densities and the source iteration continues with these periodic nuclide density updates.

Different strategies may be employed in accumulating the reaction rates which are used to perform the xenon update. This work follows a similar naming convention used by Kreher et al. (2019) when performing inline thermal-hydraulic updates with Monte Carlo. The two strategies which can also be applied to inline xenon updates are: multi-batch update with tally-flushing, and single-batch update with moving window tally-flushing.

The multi-batch update with tally-flushing strategy performs J source iterations (or simulates J generations of particles), all while accumulating reaction rates. Following these J iterations, the xenon and iodine densities are updated and the reaction rates are reset to

zero in advance of the next J iterations. This strategy is used by MC21 (Griesheimer, 2010; Griesheimer et al., 2015) and Serpent (Leppänen et al., 2015). The strategy is shown in Algorithm 2

The single-batch update with moving window tally-flushing strategy updates xenon during each iteration of the transport solver. However, the reaction rates used for this update are accumulated over the previous J iterations such that, at each subsequent transport iteration, the oldest iteration-wise reaction rates are discarded and replaced by those newly obtained. To the authors' knowledge, this strategy has not been applied to xenon equilibrium before, although it has been to thermal-hydraulic feedback by Kreher et al. (2019), displaying an improved convergence behaviour against the multi-batch strategy. The strategy is shown in Algorithm 3. Both strategies should become identical when $J = 1$.

Although it is feasible, the authors are not familiar with the inline algorithm in this form being used alongside deterministic transport codes when accounting for xenon – this is unsurprising given it was devised with Monte Carlo solvers in mind, where the computational expense of performing several transport solutions (either for depletion or for a fixed-point iteration) may be prohibitive (Griesheimer, 2010).

It should be emphasised that this inline algorithm is nearly identical to that suggested by Gill et al. (2017) and further investigated by Kreher et al. (2019) for coupling neutronics and thermal-hydraulics. Indeed, Gill (2020) has since considered the inline algorithm applied to xenon equilibrium and thermal-hydraulics simultaneously, highlighting that no analysis presently exists to justify the stabilising behaviour of inline methods.

That being said, this may too quickly dismiss work in the deterministic realm. The standard practice in whole-core deterministic transport is to iterate between neutronics and other physics during the CMFD update – this normally proceeds as a standard fixed-point iteration where CMFD is solved to some convergence criterion (which may or may not converge the eigenvalue) before using the resulting fluxes to update the nuclide densities and macroscopic cross-sections, before repeating the entire process until the CMFD neutronics and nuclide densities stop changing within some tolerance. Such a scheme is not necessarily stable, requiring some degree of relaxation between the iterations Kochunas et al. (2017a,b). However, the recently proposed X-CMFD method avoids this need for relaxation by only partially-converging the CMFD solution before updating the other physics fields while still ensuring stability (Shen et al., 2019a,b; Senecal and Ji, 2017). This is somewhat analogous with the inline methods in Monte Carlo.

3. Stability analysis

Fourier or Von Neumann stability analysis is performed to estimate the numerical stability of many algorithms throughout computational physics. In nuclear reactor physics, it has been applied to various algorithms for converging a neutronics solution (Adams and Larsen, 2002), coupling neutron transport and CMFD (Cho and Park, 2003; Jarrett et al., 2016), multi-physics feedback to neutronics (Kochunas et al., 2017b; Shen et al., 2019a,b), and burn-up (Densmore et al., 2013; Cosgrove et al., 2020), among many other areas. The results of

Input: $\mathbf{N}, n_{\text{it.}}, \alpha$
`/* Construct transport and fission operators from nuclide densities */`
$[\mathcal{L}^{(0)}, \nu\Sigma_{\text{f}}] \leftarrow \mathbf{N}$
`/* Iterate between neutronics and xenon/iodine updates */`
for $n = 0, ..., n_{\text{it.}} - 1$ **do**
  `/* Solve neutronics */`
  $\phi^{(n)} \leftarrow \left(\mathcal{L}^{(n)} - \frac{1}{k_{\text{eff}}^{(n)}}\nu\Sigma_{\text{f}}\right)\phi^{(n)} = 0$
  `/* Update densities */`
  $[X, I] \leftarrow \langle\sigma\phi^{(n)}\rangle$
  $X^{(n+1)} \leftarrow \alpha X + (1-\alpha)X^{(n)}$
  $I^{(n+1)} \leftarrow \alpha I + (1-\alpha)I^{(n)}$
  $\mathcal{L}^{(n+1)} \leftarrow X^{(n+1)}$
end

Algorithm 1: Fully-converged fixed-point xenon equilibrium algorithm.

the analysis are scalars or matrices whose eigenvalues determine whether non-physical behaviour will grow or dampen under given conditions, and, if so, provide an indication of how rapidly.

The analysis proceeds in several steps, the first of which is to represent solutions of the relevant system of equations as a true, spatially-uniform solution plus perturbation terms. The perturbation terms are subsequently decomposed into a linear combination of spatial Fourier modes across the domain, each of which can be analysed individually (due to orthogonality) to determine their behaviour when subject to the algorithm of interest. The combination of a spatially-flat exact solution and the need to decompose perturbation terms into orthogonal harmonics necessitates that the domain of interest is quite simple – in this paper it is given as $x \in [0, L]$, where L is the length of the domain.

Although the authors are concerned with how neutron transport solvers have enforced xenon equilibrium, the systems in which one enforces xenon equilibrium tend to be spatially-large thermal systems where the diffusion equation is a reasonable representation of the neutronics – this is partially justified (with respect to Monte Carlo neutron transport) in Appendix A. The mono-energetic eigenvalue form of the diffusion equation is considered:

$$-\frac{\partial}{\partial x}D\frac{\partial\phi}{\partial x} + \Sigma_{\text{a}}\phi = \frac{1}{k}\nu\Sigma_{\text{f}}\phi \ . \tag{2}$$

Here D is the diffusion coefficient, ϕ is the scalar neutron flux as before, Σ_{a} and Σ_{f} are the absorption and fission cross-sections, respectively, ν is the average number of neutrons per fission, and k is the criticality eigenvalue. The scalar flux amplitude is determined by a specified reactor power per unit area, P, such that:

$$P = \int_0^L \kappa\Sigma_{\text{f}}\phi \mathrm{d}x \ . \tag{3}$$

Input: $\mathbf{N}, n_{\text{cycles}}, J$
/* Construct transport operator, $\mathcal{L}$, and fission operator, $\nu\Sigma_{\text{f}}$, from nuclide densities, $\mathbf{N}$ */
$[\mathcal{L}^{(0)}, \nu\Sigma_{\text{f}}] \leftarrow \mathbf{N}$
/* Generate an initial guess fission source */
$\phi^{(0)} \leftarrow \Phi$
$F^{(0)} \leftarrow \nu\Sigma_{\text{f}}\phi^{(0)}$
$k_{\text{eff}}^{(0)} \leftarrow 1$
/* Initialise reaction rate estimates */
$\langle\sigma\phi\rangle \leftarrow 0$
/* Perform source iterations */
for $n = 0, ..., n_{\text{cycles}} - 1$ **do**
  /* Invert the transport operator/perform the random walk */
  $\phi^{(n+1)} \leftarrow \frac{1}{k_{\text{eff}}^{(n)}} \left(\mathcal{L}^{(n)}\right)^{-1} F^{(n)}$
  $F^{(n+1)} \leftarrow \nu\Sigma_{\text{f}}\phi^{(n+1)}$
  $k_{\text{eff}}^{(n+1)} = k_{\text{eff}}^{(n)} \frac{\langle F^{(n+1)}\rangle}{\langle F^{(n)}\rangle}$
  /* Increment reaction rate estimates */
  $\langle\sigma\phi\rangle \leftarrow \langle\sigma\phi\rangle + \frac{1}{J}\langle\sigma\phi^{(n+1)}\rangle$
  /* Update densities and flush reaction rates */
  if $\mod(n+1, J) = 0$ **then**
    $[X, I] \leftarrow \langle\sigma\phi\rangle$
    $\langle\sigma\phi\rangle \leftarrow 0$
    $\mathcal{L}^{(n+1)} \leftarrow [\mathbf{N}, X]$
  else
    $\mathcal{L}^{(n+1)} \leftarrow \mathcal{L}^{(n)}$
  end
end

Algorithm 2: Multi-batch update with tally-flushing inline xenon equilibrium algorithm.

Input: $\mathbf{N}, n_{\text{cycles}}, J$
`/* Construct transport operator,` $\mathcal{L}$`, and fission operator,` $\nu\Sigma_{\text{f}}$`, from nuclide densities,` $\mathbf{N}$ `*/`
$[\mathcal{L}^{(0)}, \nu\Sigma_{\text{f}}] \leftarrow \mathbf{N}$
`/* Generate an initial guess fission source */`
$\phi^{(0)} \leftarrow \Phi$
$F^{(0)} \leftarrow \nu\Sigma_{\text{f}}\phi^{(0)}$
$k_{\text{eff}}^{(0)} \leftarrow 1$
`/* Perform source iterations */`
for $n = 0, ..., n_{\text{cycles}} - 1$ **do**
  `/* Invert the transport operator/perform the random walk */`
  $\phi^{(n+1)} \leftarrow \frac{1}{k_{\text{eff}}^{(n)}} \left(\mathcal{L}^{(n)}\right)^{-1} F^{(n)}$
  $F^{(n+1)} \leftarrow \nu\Sigma_{\text{f}}\phi^{(n+1)}$
  $k_{\text{eff}}^{(n+1)} = k_{\text{eff}}^{(n)} \frac{\langle F^{(n+1)} \rangle}{\langle F^{(n)} \rangle}$
  `/* Update densities */`
  if $n + 1 \geq J$ **then**
    $\langle \sigma\phi \rangle \leftarrow \frac{1}{J} \sum_{j=0}^{J-1} \langle \sigma\phi^{(n+1-j)} \rangle$
    $[X, I] \leftarrow \langle \sigma\phi \rangle$
  else
    $\langle \sigma\phi \rangle \leftarrow \frac{1}{n+1} \sum_{j=1}^{n+1} \langle \sigma\phi^{(j)} \rangle$
    $[X, I] \leftarrow \langle \sigma\phi \rangle$
  end
  $\mathcal{L}^{(n+1)} \leftarrow [\mathbf{N}, X]$
end

Algorithm 3: Single-batch update with moving window tally-flushing inline xenon equilibrium algorithm.

Here κ is the average energy release per fission. Although it is certainly possible to generalise the analysis which follows to the transport equation, the algebra becomes more involved while the results limit to the those obtained from using a diffusion equation in relevant cases. Boundary conditions are also chosen to facilitate Fourier decomposition, although these only become relevant at a later stage to determine the allowable Fourier frequencies; here, reflective boundaries are applied such that:

$$\left.\frac{\partial\phi}{\partial x}\right|_{x=0} = \left.\frac{\partial\phi}{\partial x}\right|_{x=L} = 0 \,. \tag{4}$$

For the xenon and iodine densities, given that neutrons are assumed monoenergetic, Eq. (1) becomes:

$$I = \frac{\gamma_\mathrm{I}\Sigma_\mathrm{f}\phi}{\lambda_\mathrm{I}} \,, \tag{5a}$$

$$X = \frac{(\gamma_\mathrm{I}+\gamma_\mathrm{X})\Sigma_\mathrm{f}\phi}{\lambda_\mathrm{X}+\sigma_\mathrm{X}\phi} \,. \tag{5b}$$

Note that the scalar flux, iodine densities, and xenon densities are allowed to vary continuously with x. Hence, the macroscopic cross-sections and diffusion coefficient in Eq. (2) vary continuously in x and Eq. (5) is used to update the iodine and xenon densities without any further modification. This is as opposed to, say, updating the densities in relatively large, independent spatial regions of homogeneous composition with a region-averaged flux, as is done in most practical reactor physics tools.

The coupling from the nuclide densities to neutronics is through the absorption cross-section and diffusion coefficient featuring in Eq. (2), such that:

$$\Sigma_\mathrm{a} = \Sigma_\mathrm{a,i} + \sigma_\mathrm{X} X \,, \tag{6a}$$

$$D = \frac{1}{3(\Sigma_\mathrm{t,i} + \sigma_\mathrm{X} X - \bar{\mu}\Sigma_\mathrm{s,i})} \,. \tag{6b}$$

Here Σ_t is the total cross-section, Σ_s is the scattering cross-section, $\bar{\mu}$ is the average scattering cosine, and the i subscript denotes the initial 'fresh' cross-sections when no xenon is present. It is assumed that xenon's contribution to neutron scattering is negligible.

For this system of equations, the converged solutions are known:

$$\phi_0 = \frac{P}{\kappa\Sigma_{\mathrm{f},0}L} \,, \tag{7a}$$

$$k_0 = \frac{\nu\Sigma_{\mathrm{f},0}}{\Sigma_{\mathrm{a},0}} \,, \tag{7b}$$

$$X_0 = \frac{(\gamma_\mathrm{I}+\gamma_\mathrm{X})\Sigma_\mathrm{f}\phi_0}{\lambda_\mathrm{X}+\sigma_\mathrm{X}\phi_0} \,, \tag{7c}$$

$$I_0 = \frac{\gamma_\mathrm{I}\Sigma_\mathrm{f}\phi_0}{\lambda_\mathrm{I}} \,, \tag{7d}$$

where the macroscopic cross-sections and diffusion coefficient under these conditions are:

$$\Sigma_{\mathrm{a},0} = \Sigma_{\mathrm{a,i}} + \sigma_{\mathrm{X}} X_0 \ , \tag{8a}$$

$$\Sigma_{\mathrm{f},0} = \Sigma_{\mathrm{f,i}} \ , \tag{8b}$$

$$\Sigma_{\mathrm{s},0} = \Sigma_{\mathrm{s,i}} \ , \tag{8c}$$

$$D_0 = \frac{1}{3(\Sigma_{\mathrm{t,i}} + \sigma_{\mathrm{X}} X_0 - \bar{\mu}\Sigma_{\mathrm{s},0})} \ . \tag{8d}$$

Eq. (7a) comes from the uniformity of the problem and the power normalisation in Eq. (3). Eq. (7b) also results from the uniformity of the system and the lack of neutron leakage. Eqs. (7c) and (7d) result from applying the uniform flux solution to Eq. (5).

By Taylor expansion about the converged solutions, all other solutions at some iteration number, n, may be written in terms of the converged solution plus some perturbation terms:

$$\phi^{(n)} = \phi_0 + \epsilon\phi_1^{(n)} + \mathcal{O}(\epsilon^2) \ , \tag{9a}$$

$$k^{(n)} = k_0 + \epsilon k_1^{(n)} + \mathcal{O}(\epsilon^2) \ , \tag{9b}$$

$$X^{(n)} = X_0 + \epsilon X_1^{(n)} + \mathcal{O}(\epsilon^2) \ , \tag{9c}$$

$$I^{(n)} = I_0 + \epsilon I_1^{(n)} + \mathcal{O}(\epsilon^2) \ . \tag{9d}$$

Here the 1 subscript denotes a first-order perturbation, and ϵ is a small parameter. Higher-order components of each expression are contained in the $\mathcal{O}(\epsilon^2)$ term. Each perturbation term can then be represented as a Fourier series:

$$\phi_1 = \sum_{\omega=-\infty}^{\infty} \phi_\omega e^{i\omega x} \ , \tag{10a}$$

$$k_1 = \sum_{\omega=-\infty}^{\infty} k_\omega e^{i\omega x} \ , \tag{10b}$$

$$X_1 = \sum_{\omega=-\infty}^{\infty} X_\omega e^{i\omega x} \ , \tag{10c}$$

$$I_1 = \sum_{\omega=-\infty}^{\infty} I_\omega e^{i\omega x} \ . \tag{10d}$$

Here i is the imaginary unit and ω is the Fourier frequency, allowable values of which are determined by the boundary conditions; for the reflective boundary conditions specified, $\omega = \frac{m\pi}{L} \ \forall \ m \in \mathbb{Z}$. The terms with a subscripted ω are the corresponding Fourier amplitudes. The expression further simplifies for Eq. (10b): not only does the eigenvalue possess a single, flat Fourier mode (as it is uniform across the problem), but it can also be shown that the first order perturbation to the eigenvalue is zero after a single iteration, i.e., $k_1^{(n)} = 0$.

These expressions are ultimately inserted into the equations corresponding to the coupling scheme of interest, with the aim of determining the behaviour of the Fourier amplitudes

subject to the given coupling scheme. As the inline and fully-converged fixed-point coupling schemes are quite different and necessitate treating neutronics slightly differently, they will be detailed in separate sections. As the simpler scheme, the fully-converged fixed-point iteration will be analysed first.

It should be emphasised that, in the Monte Carlo context, the expressions obtained here are only exact where statistical error does not occur, i.e., where an infinite number of particles are simulated. This may adversely affect the accuracy of the stability estimates. A similar disclaimer has been made by Keady and Larsen (2016) for a stability analysis applied to CMFD-accelerated source convergence in Monte Carlo simulations – they highlighted cases in which a noiseless simulation was predicted to be stable, but nevertheless instability was induced by Monte Carlo noise.

3.1. Fully-converged fixed-point iteration

The fully-converged fixed-point scheme consists of solving neutronics to convergence for a given xenon density field before recalculating the xenon and iodine concentrations (perhaps including relaxation) and repeating. This gives the following system of equations describing the coupling scheme:

$$-\frac{\partial}{\partial x}D^{(n)}\frac{\partial\phi^{(n)}}{\partial x}+\Sigma_{\mathrm{a}}^{(n)}\phi^{(n)}=\frac{1}{k^{(n)}}\nu\Sigma_{\mathrm{f}}\phi^{(n)}\ , \tag{11a}$$

$$X^{(n+1)}=\alpha\frac{(\gamma_{\mathrm{I}}+\gamma_{\mathrm{X}})\Sigma_{\mathrm{f}}\phi^{(n)}}{\lambda_{\mathrm{X}}+\sigma_{\mathrm{X}}\phi^{(n)}}+(1-\alpha)X^{(n)}\ , \tag{11b}$$

$$I^{(n+1)}=\alpha\frac{\gamma_{\mathrm{I}}\Sigma_{\mathrm{f}}\phi^{(n)}}{\lambda_{\mathrm{I}}}+(1-\alpha)I^{(n)}\ . \tag{11c}$$

Here the (n) superscript denotes the solution after n iterations of the algorithm. Although Eq. (11c) is included here, it has no direct relevance to stability given that iodine's cross-section is often neglected and so it has no feedback to neutronics; hence, iodine may be safely neglected in the following stability analysis.

The next step of the analysis involves the insertion of Eq. (9) into Eq. (11), linearising, and neglecting terms which are second-order-or-higher in ϵ. This gives:

$$-D_0\frac{\partial^2\phi_1^{(n)}}{\partial x^2}+\sigma_{\mathrm{X}}\phi_0X_1^{(n)}=0\ , \tag{12a}$$

$$X_1^{(n+1)}=\alpha\frac{(\gamma_{\mathrm{I}}+\gamma_{\mathrm{X}})\lambda_{\mathrm{X}}\Sigma_{\mathrm{f},0}}{\left(\lambda_{\mathrm{X}}+\sigma_{\mathrm{X}}\phi_0\right)^2}\phi_1^{(n)}+(1-\alpha)X_1^{(n)}\ . \tag{12b}$$

Inserting Eq. (10), Eq. (12) simplifies, and one may consider individual Fourier amplitudes due to orthogonality. Using this and re-arranging gives:

$$\phi_\omega^{(n)}=-\frac{\sigma_{\mathrm{X}}\phi_0}{\omega^2D_0}X_\omega^{(n)}\ , \tag{13a}$$

$$X_\omega^{(n+1)}=\alpha\frac{(\gamma_{\mathrm{I}}+\gamma_{\mathrm{X}})\lambda_{\mathrm{X}}\Sigma_{\mathrm{f},0}}{\left(\lambda_{\mathrm{X}}+\sigma_{\mathrm{X}}\phi_0\right)^2}\phi_\omega^{(n)}+(1-\alpha)X_\omega^{(n)}\ . \tag{13b}$$

Thus, inserting Eq. (13a) into Eq. (13b), the amplitude of the xenon density's ω Fourier mode grows according to:

$$X_\omega^{(n+1)} = \left((1-\alpha) - \alpha \frac{\sigma_\mathrm{X} X_0}{\omega^2 D_0} \frac{\lambda_\mathrm{X}}{\lambda_\mathrm{X} + \sigma_\mathrm{X} \phi_0} \right) X_\omega^{(n)} \ . \tag{14}$$

From this expression one can see that small values of ω may induce an amplification factor with magnitude greater than 1. Smaller values of ω, corresponding to the flattest Fourier modes, can only be sustained in large geometries (given the definition of ω above as inversely proportional to the problem length). Hence numerical instabilities with this fixed-point scheme are more likely to manifest in spatially-large reactor problems.

3.2. Inline coupling

Both inline coupling algorithms add an additional sophistication to the analysis: neutronics is not presumed to be converged. That is, the equivalent of Eq. (11a) must account for the source iteration algorithm. A simplification in the inline algorithms – relative to fully-converged fixed-point iteration – is that relaxation is not performed. Hence, $\alpha = 1$, simplifying the xenon update equation. The two inline algorithms are considered separately in this section.

3.2.1. Multi-batch update with tally-flushing

Given that the reaction rates used to update the xenon and iodine densities are accumulated/averaged over several source iterations while the nuclide densities remain fixed during those iterations, this necessitates a staggered indexing to distinguish whether xenon is updated following a source iteration. Hence, to make this clear, two new iteration variables, j and m, are defined, such that:

$$n = j + mJ \ . \tag{15}$$

Here j corresponds to the source iteration index while m is the xenon update iteration index, both beginning at 0 and j reaching a maximum of $J-1$, after which j is reset to 0 and m is incremented.

Given the above, the system of equations describing the multi-batch inline coupling scheme (neglecting iodine) can be written:

$$\left(-\frac{\partial}{\partial x} D^{(m)} \frac{\partial}{\partial x} + \Sigma_\mathrm{a}^{(m)} \right) \phi^{(0,m)} = \frac{1}{k^{(j,m-1)}} \nu\Sigma_\mathrm{f} \phi^{(j,m-1)} \ , \quad \text{if } j = J-1 \ , \tag{16a}$$

$$\left(-\frac{\partial}{\partial x} D^{(m)} \frac{\partial}{\partial x} + \Sigma_\mathrm{a}^{(m)} \right) \phi^{(j+1,m)} = \frac{1}{k^{(j,m)}} \nu\Sigma_\mathrm{f} \phi^{(j,m)} \ , \quad \text{if } j < J-1 \ , \tag{16b}$$

$$X^{(m+1)} = \frac{\frac{1}{J} \sum_{j=0}^{J-1} (\gamma_\mathrm{I} + \gamma_\mathrm{X}) \Sigma_\mathrm{f} \phi^{(j,m)}}{\lambda_\mathrm{X} + \frac{1}{J} \sum_{j=0}^{J-1} \sigma_\mathrm{X} \phi^{(j,m)}} \ . \tag{16c}$$

Once again, introducing Eq. (9) into Eq. (16), linearising, and neglecting higher-order terms gives:

$$\left(-D_0\frac{\partial^2}{\partial x^2}+\Sigma_{\mathrm{a},0}\right)\phi_1^{(0,m)}+\sigma_{\mathrm{X}}\phi_0 X_1^{(m)}=\Sigma_{\mathrm{a},0}\phi_1^{(j,m-1)}\ ,\quad \text{if } j=J-1\ , \tag{17a}$$

$$\left(-D_0\frac{\partial^2}{\partial x^2}+\Sigma_{\mathrm{a},0}\right)\phi_1^{(j+1,m)}+\sigma_{\mathrm{X}}\phi_0 X_1^{(m)}=\Sigma_{\mathrm{a},0}\phi_1^{(j,m)}\ ,\quad \text{if } j<J-1\ , \tag{17b}$$

$$X_1^{(m+1)}=\frac{(\gamma_{\mathrm{I}}+\gamma_{\mathrm{X}})\Sigma_{\mathrm{f},0}}{\left(\lambda_{\mathrm{X}}+\sigma_{\mathrm{X}}\phi_0\right)^2}\frac{\lambda_{\mathrm{X}}}{J}\sum_{j=0}^{J-1}\phi_1^{(j,m)}\ . \tag{17c}$$

This equation simplifies by inserting Eq. (10), giving for each Fourier mode:

$$\phi_\omega^{(0,m)}=\frac{\Sigma_{\mathrm{a},0}\phi_\omega^{(j,m-1)}-\sigma_{\mathrm{X}}\phi_0 X_\omega^{(m)}}{\Sigma_{\mathrm{a},0}+\omega^2 D_0}\ ,\quad \text{if } j=J-1\ , \tag{18a}$$

$$\phi_\omega^{(j+1,m)}=\frac{\Sigma_{\mathrm{a},0}\phi_\omega^{(j,m)}-\sigma_{\mathrm{X}}\phi_0 X_\omega^{(m)}}{\Sigma_{\mathrm{a},0}+\omega^2 D_0}\ ,\quad \text{if } j<J-1\ , \tag{18b}$$

$$X_\omega^{(m+1)}=\frac{(\gamma_{\mathrm{I}}+\gamma_{\mathrm{X}})\Sigma_{\mathrm{f},0}}{\left(\lambda_{\mathrm{X}}+\sigma_{\mathrm{X}}\phi_0\right)^2}\frac{\lambda_{\mathrm{X}}}{J}\sum_{j=0}^{J-1}\phi_\omega^{(j,m)}\ . \tag{18c}$$

Eq. (18) can be used to produce a matrix relating $\phi_\omega^{(J-1,m-1)}$ and $X_\omega^{(m)}$ to $\phi_\omega^{(J-1,m)}$ and $X_\omega^{(m+1)}$. This is:

$$\begin{bmatrix}\phi_\omega^{(J-1,m)}\\ X_\omega^{(m+1)}\end{bmatrix}=\mathbf{A}\begin{bmatrix}\phi_\omega^{(J-1,m-1)}\\ X_\omega^{(m)}\end{bmatrix}\ , \tag{19}$$

where:

$$\mathbf{A}_{1,1}=\rho_\omega^J\ , \tag{20a}$$

$$\mathbf{A}_{1,2}=-\frac{\sigma_{\mathrm{X}}\phi_0}{\Sigma_{\mathrm{a},0}}\sum_{j=1}^{J}\rho_\omega^j\ , \tag{20b}$$

$$\mathbf{A}_{2,1}=\frac{\eta}{J}\sum_{j=1}^{J}\rho_\omega^J\ , \tag{20c}$$

$$\mathbf{A}_{2,2}=-\frac{\sigma_{\mathrm{X}}\phi_0\eta}{J\Sigma_{\mathrm{a},0}}\sum_{i=1}^{J}\sum_{j=1}^{i}\rho_\omega^j\ , \tag{20d}$$

with:

$$\rho_\omega=\frac{1}{1+\omega^2 L_D^2}\ , \tag{21}$$

where the diffusion length is used:

$$L_D = \sqrt{\frac{D_0}{\Sigma_{\mathrm{a},0}}} \,, \tag{22}$$

and:

$$\eta = \frac{\lambda_{\mathrm{X}}(\gamma_{\mathrm{I}} + \gamma_{\mathrm{X}})\Sigma_{\mathrm{f},0}}{(\lambda_{\mathrm{X}} + \sigma_{\mathrm{X}}\phi_0)^2} \,. \tag{23}$$

The spectral radius of $\mathbf{A}$ can be used to determine the behaviour of the inline coupling scheme.

In the case of $J = 1$, simpler expressions can be obtained. For example, in terms of the growth of the flux Fourier modes, one would obtain:

$$\phi_\omega^{(n+1)} = \frac{1-\delta}{1+\omega^2 L_D^2}\phi_\omega^{(n)} \,, \tag{24}$$

where $\delta = \frac{\sigma_{\mathrm{X}} X_0}{\Sigma_{\mathrm{a},0}} \frac{\lambda_{\mathrm{X}}}{\lambda_{\mathrm{X}}+\sigma_{\mathrm{X}}\phi_0}$. As ω is varied from $-\infty$ to ∞, the amplification factor will remain bounded by $1-\delta$, implying the unconditional stability of the inline algorithm. This expression is similar to that of the source iteration algorithm, which, for this problem with cross-sections at their xenon equilibrium values, would be:

$$\phi_\omega^{(n+1)} = \frac{1}{1+\omega^2 L_D^2}\phi_\omega^{(n)} \,. \tag{25}$$

The amplification factor here is simply the dominance ratio of the problem. Given that δ is positive and less than 1, these two expressions imply that the fission source converges more rapidly when simultaneously enforcing xenon equilibrium – similar behaviour was found by Kochunas et al. (2017b) for general flux-dependent feedback on cross-sections and by Shen et al. (2019b) when performing non-linear updates alongside partially-converged CMFD. That being said, as $\delta \sim 10^{-3}$ for realistic reactor systems, the practical convergence acceleration is minute.

3.2.2. Single-batch update with moving window tally-flushing

With this algorithm, given that xenon is updated every iteration, the double indexing is no longer necessary for accounting and the (n) superscript can be restored. The equations describing the algorithm are:

$$\left(-\frac{\partial}{\partial x}D^{(n)}\frac{\partial}{\partial x} + \Sigma_{\mathrm{a}}^{(n)}\right)\phi^{(n+1)} = \frac{1}{k^{(n)}}\nu\Sigma_{\mathrm{f}}\phi^{(n)} \,, \tag{26a}$$

$$X^{(n)} = \frac{\frac{1}{J}\sum_{j=0}^{J-1}(\gamma_{\mathrm{I}} + \gamma_{\mathrm{X}})\Sigma_{\mathrm{f}}\phi^{(n-j)}}{\lambda_{\mathrm{X}} + \frac{1}{J}\sum_{j=0}^{J-1}\sigma_{\mathrm{X}}\phi^{(n-j)}} \,. \tag{26b}$$

These equations neglect the initialisation period necessary for the algorithm; in the first $J-1$ iterations an alternative xenon update equation must be employed, although this period is

assumed to be brief and to not affect the longer-term stability behaviour. Expressing the variables in terms of first-order perturbations, linearising, and neglecting higher-order terms gives:

$$\left(-D_0\frac{\partial^2}{\partial x^2}+\Sigma_{\mathrm{a},0}\right)\phi_1^{(n+1)}+\sigma_X\phi_0X_1^{(n)}=\Sigma_{\mathrm{a},0}\phi_1^{(n)}\ , \tag{27a}$$

$$X_1^{(n)}=\frac{(\gamma_\mathrm{I}+\gamma_\mathrm{X})\Sigma_{\mathrm{f},0}}{\left(\lambda_\mathrm{X}+\sigma_\mathrm{X}\phi_0\right)^2}\frac{\lambda_\mathrm{X}}{J}\sum_{j=0}^{J-1}\phi_1^{(n-j)}\ . \tag{27b}$$

Considering the orthogonal Fourier frequencies gives:

$$\phi_\omega^{(n+1)}=\frac{\Sigma_{\mathrm{a},0}\phi_\omega^{(n)}-\sigma_\mathrm{X}\phi_0X_\omega^{(n)}}{\Sigma_{\mathrm{a},0}+\omega^2D_0}\ , \tag{28a}$$

$$X_\omega^{(n)}=\frac{(\gamma_\mathrm{I}+\gamma_\mathrm{X})\Sigma_{\mathrm{f},0}}{\left(\lambda_\mathrm{X}+\sigma_\mathrm{X}\phi_0\right)^2}\frac{\lambda_\mathrm{X}}{J}\sum_{j=0}^{J-1}\phi_\omega^{(n-j)}\ . \tag{28b}$$

Inserting Eq. (28b) into Eq. (28a) gives:

$$\phi_\omega^{(n+1)}=\frac{1-\delta/J}{1+\omega^2L_D^2}\phi_\omega^{(n)}-\frac{\delta/J}{1+\omega^2L_D^2}\sum_{j=1}^{J-1}\phi_\omega^{(n-j)}\ , \tag{29}$$

where δ and L_D remain as previously defined. This can be rewritten as a system of equations:

$$\Phi_\omega^{(n+1)}=\mathbf{A}\Phi_\omega^{(n)}\ , \tag{30}$$

with:

$$\Phi_\omega^{(n+1)}=\begin{bmatrix}\phi_\omega^{(n+1)}\\ \phi_\omega^{(n)}\\ \vdots\\ \phi_\omega^{(n-J+2)}\end{bmatrix}\ , \tag{31}$$

$$\Phi_\omega^{(n)}=\begin{bmatrix}\phi_\omega^{(n)}\\ \phi_\omega^{(n-1)}\\ \vdots\\ \phi_\omega^{(n-J+1)}\end{bmatrix}\ , \tag{32}$$

and:

$$\mathbf{A}=\begin{bmatrix}\frac{1-\delta/J}{1+\omega^2L_D^2} & -\frac{\delta/J}{1+\omega^2L_D^2} & \cdots & -\frac{\delta/J}{1+\omega^2L_D^2}\\ 1 & 0 & \cdots & 0\\ 0 & \ddots & \ddots & \vdots\\ \vdots & & & \end{bmatrix}\ . \tag{33}$$

The spectral radius of $\mathbf{A}$ determines the stability of the single-batch update with moving window tally-flushing. Setting $J=1$ reduces Eq. (30) to Eq. (24).

4. Numerical investigations

The validity of Eqs. (14), (19) and (30) have been tested against simulation. This will use a simple diffusion solver written in MATLAB as well as the Serpent Monte Carlo code (Leppänen et al., 2015). Numerical estimates of the spectral radius, ρ, can be made using the well-known approximation (Keady and Larsen, 2016):

$$\rho \approx \frac{||u^{(n+1)} - u^{(n)}||}{||u^{(n)} - u^{(n-1)}||} \,. \tag{34}$$

Here u is the appropriate vector, corresponding to the equation of interest. This approximation is valid for large values of n and tends to provide a more reliable estimate when ρ approaches 1. When ρ is small, the solution may converge too rapidly and floating-point error may thwart its numerical estimation; when ρ is large and greater than 1, non-linear behaviour may cause the growth of higher modes to saturate quickly, also negating the use of Eq. (34). The flux and xenon vectors for the estimate can be obtained directly from the diffusion solver.

One-group cross-sections and iodine and xenon fission yields are generated from Serpent using a reflected homogeneous slab geometry with only a single burnable region in which the xenon equilibrium density is obtained. This geometry is made by homogenising a pressurised water reactor (PWR) pin geometry with a pitch of 1.26 cm, 5% enriched UO_2 fuel with a radius 0.4095 cm and a density of 10.4 g/cm^3, natural zirconium cladding with thickness 0.0655 cm, and water density of 0.7 g/cm^3. The cladding is neglected in the homogenised composition. The average neutron scattering cosine is about 0.643 while the average energy per fission is about 203 MeV. The half-life of xenon is taken as 9.14 hours. Unless stated otherwise, all results are generated using a constant power density of 104 W/cm^3 (rather than a constant power per unit area) – this is standard for a PWR. The diffusion solver uses 150 discrete regions which is sufficiently fine for the continuous treatment of the diffusion equation in the stability analysis to remain valid.

Theoretical and numerical spectral radii for the inline scheme (for $J = 1$) and the fully-converged fixed-point iteration with several relaxation factors are shown in Fig. 1. The numerical spectral radii shown here are produced by the diffusion solver. Recalling that the lowest spurious frequency that the reflected geometry can sustain is $\omega = \frac{\pi}{L}$, the numerical results were obtained by varying the length of the geometry and calculating the spectral radius using Eq. (34). As predicted from Eq. (14), low frequencies (which occur in large geometries) can be excited – these are oscillatory, given the sign of the amplification factor in Eq. (14). Likewise, as seen from Eq. (19), the inline algorithm is unconditionally stable. All numerical spectral radii match the theoretical results well, except, as one might expect, where the radii are small and convergence is rapid.

Fig. 2(a) shows theoretical and numerical spectral radii of the multi-batch update inline scheme with tally-flushing where the update batch size is varied. The accuracy of numerical spectral radii deteriorates for higher frequencies – otherwise, the theoretical and numerical results agree well. Although accumulating reaction rates across more cycles apparently reduces the spectral radius, this spectral radius corresponds to the xenon update iterations,

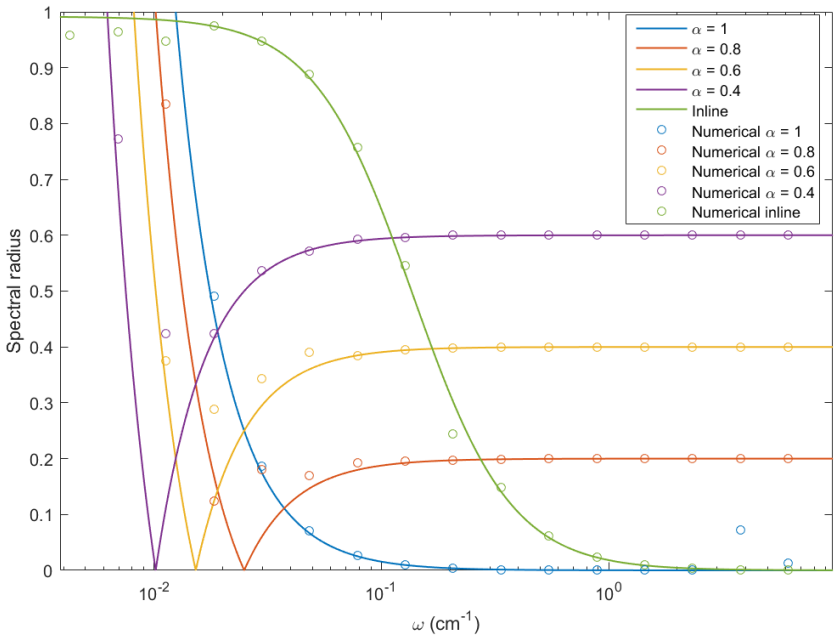
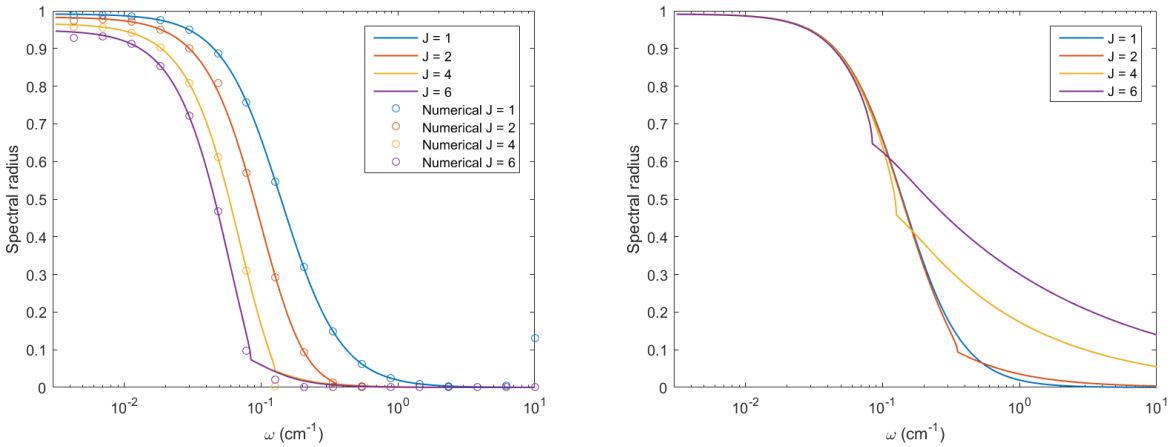


Fig. 1. Theoretical and numerical spectral radii with Fourier frequency of the fully-converged fixed-point algorithm with relaxation (from Eq. (14)) and the inline algorithm with $J = 1$ (from Eq. (19)).

which become less frequent as the number of cycles to accumulate reaction rates increases. As an approximation to the 'per source iteration' convergence, the J-th root of these spectral radii are shown in Fig. 2(b) (where, as before, J is the number of iterations between xenon updates). The figure shows that accumulating over more batches does not practically accelerate convergence of low frequencies; the value of this strategy is predominantly in mitigating statistical effects from a Monte Carlo solver. Indeed, accumulating reaction rates across multiple batches will cause high-frequency modes to persist longer than when using only a single batch.

The stability behaviour of the single batch update strategy with moving window tally-flushing is shown in Fig. 3. For high frequencies and multiple batches, the accuracy of the numerical estimates are severely degraded due to rapid convergence for low spectral radii and the many source iterations required to evaluate the spectral radii when multiple batches are used. Otherwise, for fewer batches or lower frequencies (which are of most interest for evaluating stability), the numerical spectral radii are in good agreement with the theoretical predictions.

The effect of varying power density as well is also shown from the stability analysis. Fig. 4 shows how spectral radius varies with power density and length when using the fully-converged fixed-point scheme, with and without relaxation. It is unsurprising that the stability of enforcing xenon equilibrium varies with the power normalisation – at PWR operating power densities, the problem is nearly at its least stable, though relaxation remains effective across variations in power. Fig. 5 shows the same investigation when using the inline



(a) Theoretical and numerical radii.

(b) J-th root of theoretical radii.

Fig. 2. Spectral radii of the multi-batch update with tally-flushing strategy when varying the number of batches, with theoretical results from Eq. (19).

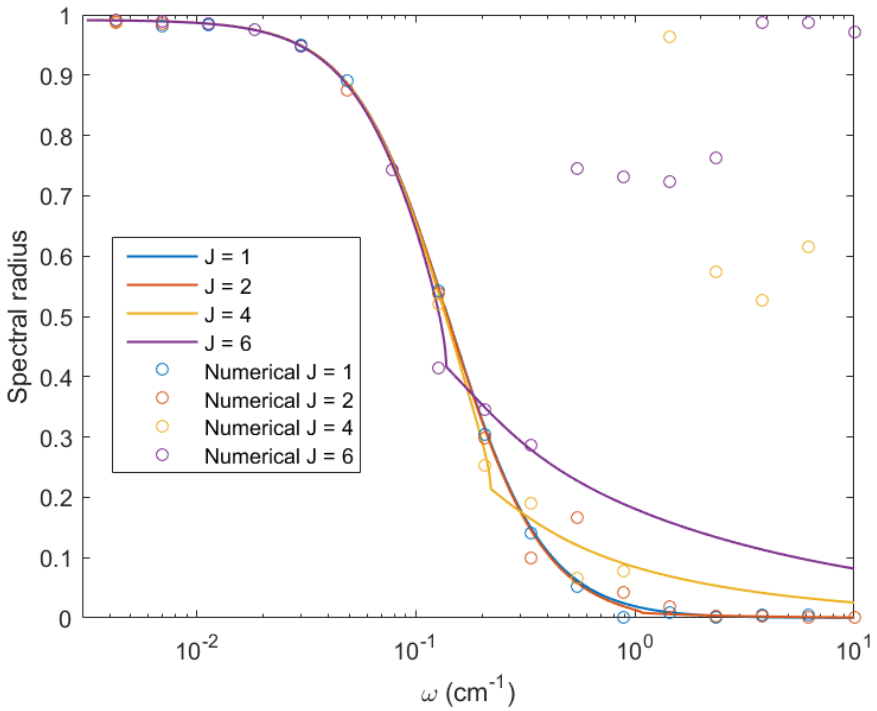
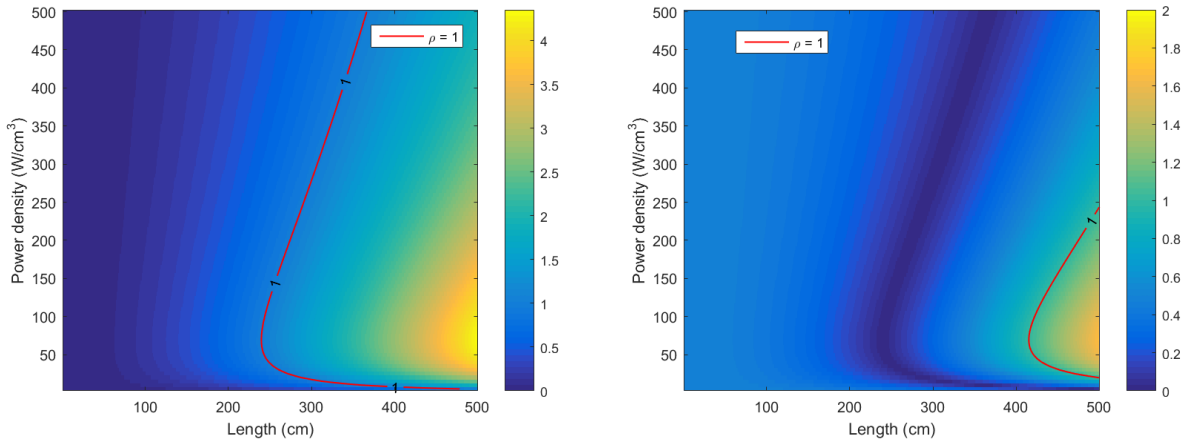


Fig. 3. Spectral radii of the single batch update moving window tally-flushing inline algorithm using various window sizes, with theoretical results from Eq. (30).



(a) $\alpha = 1$ **(b)** $\alpha = 0.5$

Fig. 4. Spectral radii of two fully-converged fixed-point iteration schemes for problems of different lengths and power densities, evaluated using Eq. (14).

scheme with only a single batch update – as one would anticipate, the algorithm is robust to variations in the power normalisation, as well as the size of the geometry.

Finally, Serpent is used to examine the validity of the theoretical predictions in a higher-fidelity problem, subject to stochastic noise. When using Serpent with the inline coupling scheme, the source code was modified to return the flux and calculated xenon densities at the end of each inactive cycle. Each inline MC solution simulated 10,000,000 particles for 10 inactive cycles and 1 active cycle – these settings were chosen to attempt to minimise stochastic phenomena affecting inactive cycle estimates of the spectral radius. It is also unnecessary to perform many inactive cycles in these investigations because stochastic noise dominates the estimation of the spectral radius after very few cycles.

Although not done natively by Serpent, the fully-converged fixed-point algorithm is easily implemented by running Serpent to completion, using estimated reaction rates to calculate the xenon density, and updating the input file appropriately – this was handled by an external script. Each fully-converged fixed-point Monte Carlo solution simulated 100,000 particles for 200 inactive cycles and 100 active cycles – in this case, ensuring good source convergence is necessary, while it is no longer important to minimise noise in cycle-wise reaction rate estimates.

The geometry in Serpent is divided into 40 equal-length regions across which the xenon density is calculated. Each coupling scheme was applied to a system of length varying from 1 m to 4 m, in increments of 1 m. Furthermore, each problem of interest was run 25 times, allowing statistical errors to be calculated.

The two algorithms start with different conditions in order to start them reasonably far from a converged condition, improving the ease with which the spectral radius might be estimated. For the inline algorithm, the initial xenon density was set to zero in all regions across the problem. However, the initial fission source used was for a converged problem with vacuum axial boundary conditions, i.e., a cosine-like fission source distribution. Using

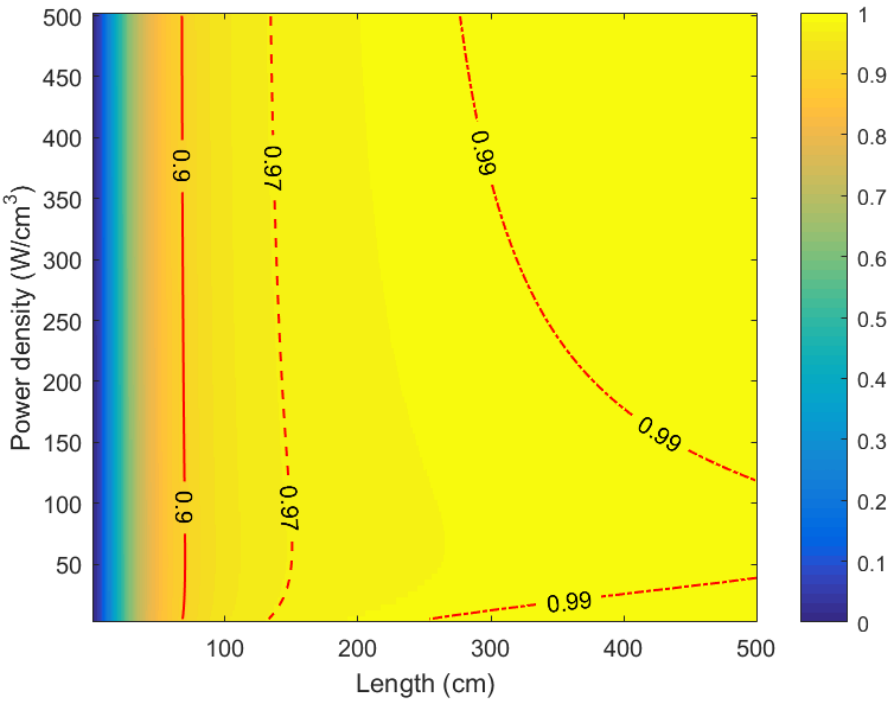
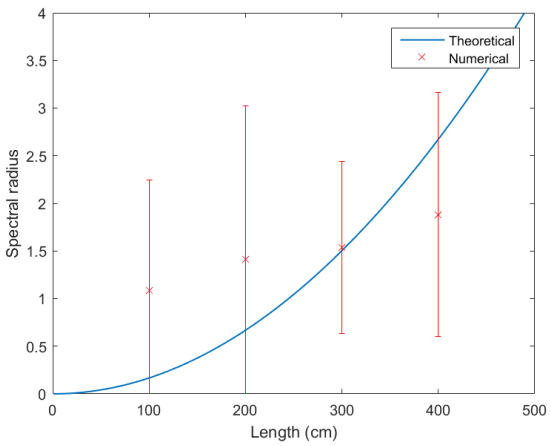
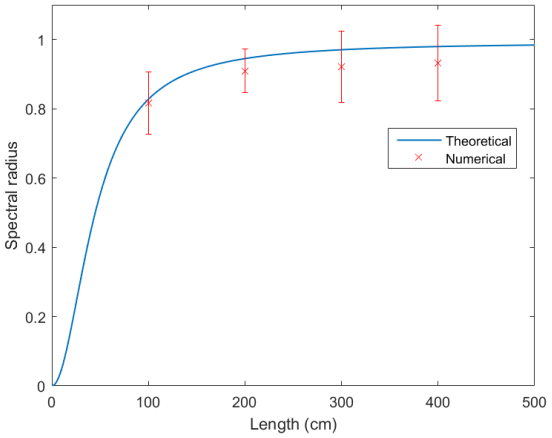


Fig. 5. Spectral radii of the inline algorithm with $J = 1$ for problems of different lengths and power densities, evaluated using Eq. (24).



(a) Fully-converged fixed-point algorithm with theoretical results from Eq. (14)



(b) Inline algorithm with theoretical results from Eq. (24)

Fig. 6. Comparison of theoretical spectral radii with Monte Carlo estimates for different xenon equilibrium algorithms.

this boundary condition on a uniform problem should have largely removed asymmetric fission source modes – therefore, the inline simulation is performed using periodic boundary conditions, where the first mode (with frequency $\omega = 2\pi/L$) should be dominant to begin. For the fully-converged fixed-point algorithm, the xenon density in each region is sampled uniformly between 2-4E-9 atoms/b/cm, while the initial fission source distribution, for all iterations, is sampled from a uniform distribution across the length of the problem.

These results are shown in Fig. 6(a) for the fully-converged fixed-point scheme without relaxation and in Fig. 6(b) for a one-batch inline scheme. The mean value of the spectral radius is plotted with error bars corresponding to two standard deviations.

In spite of running many simulations, the variance for the fully-converged fixed-point algorithm is substantial. This is partly justified in that, for Monte Carlo solvers generally, the use of Eq. (34) is more problematic as statistical noise can greatly erode its accuracy. Eq. (34) is also most accurate where the spectral radius is close to one – in Fig. 6(a), the mean values are more reasonable for the 200 cm and 300 cm problem, where neither convergence nor divergence occurs rapidly. On the other hand, for 100 cm and 400 cm, the spectral radii are higher and lower than theory would predict – this may be partly attributable to both rapidly converging, either to a flat solution for the 100 cm system or to an oscillatory solution for the 400 cm system. Although the spectral radii estimates are noisy, the qualitative instability behaviour is as predicted, at least after 5 iterations: the 100 cm and 200 cm problems are relatively spatially flat, while the 300 cm and 400 cm problems fall into oscillatory solutions. There is some variation across the solutions, although this mainly concerns the extent to which any unstable modes have developed, with some growing or depressing earlier than others depending on the initial guess xenon distribution.

Much better agreement is obtained between theory and numerical estimates for the inline algorithm, supporting the correctness of the analysis.

5. Conclusions

The stability properties of two classes of xenon equilibrium algorithm were analysed: the fully-converged fixed-point approach and the inline feedback approach. The fixed-point algorithm was found to be only conditionally stable while the inline algorithm is unconditionally stable. The inline algorithm was shown to accelerate the source iteration procedure, although not substantially, in agreement with results from Kochunas et al. (2017b). Two more sophisticated approaches to the inline algorithm were considered, following the same lines as multi-batch strategies used alongside inline Monte Carlo feedback. For lower frequencies, both schemes partially accelerate the convergence of the flux/xenon system and remain unconditionally stable. For higher frequencies (which are not so concerning for stability) they exhibit slow convergence, although the spectral radius remains low. Ultimately, in idealised problems where an infinite number of particles are simulated, there is little practical difference between the various inline coupling schemes.

Future work will require accounting for stochastic phenomena in the treatment of the two multi-batch strategies, as these were designed to account for deleterious effects of Monte Carlo noise and correlation (Griesheimer, 2010; Kreher et al., 2019). It remains to be stated definitively whether these algorithms possess additional advantages or disadvantages when subject to stochastic phenomena. However, the current work should facilitate more sophisticated analyses in future.

6. Data availability statement

To the best of the authors' knowledge, this paper and references herein contain all the data needed to reproduce and validate the results presented.

Declarations

Declarations of interest: none.

Appendix A. validity of the diffusion approximation

Throughout the paper, neutronics was handled using the diffusion approximation. This may appear particularly egregious when comparing the results of the stability analysis to Monte Carlo neutron transport, where the source iteration (and therefore the inline algorithm) will not necessarily behave similarly to that given by neutron diffusion. This can be justified by considering the behaviour of the transport equation as applied to Monte Carlo in a relatively diffusive regime. Such an analysis is the concern of this appendix.

First, the mono-energetic k-eigenvalue form of the transport equation with isotropic scattering in 1D slab geometry is:

$$\mu \frac{\partial}{\partial x}\psi(x,\mu) + \Sigma_{\mathrm{t}}\psi(x,\mu) = \frac{\nu\Sigma_{\mathrm{f}}}{2k}\int_{-1}^{1}\mathrm{d}\mu'\psi(x,\mu') + \frac{\Sigma_{\mathrm{s}}}{2}\int_{-1}^{1}\mathrm{d}\mu'\psi(x,\mu')\,, \tag{A.1}$$

where μ is the cosine of the neutron angle of flight relative to the positive x-axis, ψ is the angular neutron flux, and the other symbols retain the same meanings as used throughout the rest of the paper, with cross-sections treated as constant in space. Note that anisotropic scattering can be accounted for by applying a transport correction to both Σ_t and Σ_s when comparing with diffusion. For the analysis to follow, this equation can be supplemented with either reflective or periodic boundary conditions. The reflective boundary conditions are:

$$\psi(0,\mu) = \psi(0,-\mu)\ ,\ 0 < \mu \leq 1\ , \tag{A.2a}$$
$$\psi(L,\mu) = \psi(L,-\mu)\ ,\ -1 \leq \mu < 0\ , \tag{A.2b}$$

while the periodic boundary conditions are:

$$\psi(0,\mu) = \psi(L,\mu)\ ,\ -1 \leq \mu < 0\ , \tag{A.3a}$$
$$\psi(L,\mu) = \psi(0,\mu)\ ,\ 0 < \mu \leq 1\ . \tag{A.3b}$$

From the solution of the transport equation, the scalar flux can be calculated as:

$$\phi(x) = \int_{-1}^{1} \mathrm{d}\mu \psi(x,\mu)\ . \tag{A.4}$$

The amplitude of the scalar flux is given by some normalisation condition. The eigenvalue is calculated as:

$$k = \frac{\int_0^L \mathrm{d}x \nu\Sigma_\mathrm{f}\phi(x)}{\int_0^L \mathrm{d}x \Sigma_\mathrm{a}\phi(x)}\ . \tag{A.5}$$

Subject to either of these boundary conditions, the fundamental angular flux solution should be isotropic and homogeneous across the domain, while the corresponding eigenvalue should be $k = \nu\Sigma_\mathrm{f}/\Sigma_\mathrm{a}$.

The Monte Carlo source iteration procedure for the transport equation is given as:

$$\mu\frac{\partial}{\partial x}\psi^{(n+1)}(x,\mu) + \Sigma_\mathrm{t}\psi^{(n+1)}(x,\mu) - \frac{\Sigma_\mathrm{s}}{2}\int_{-1}^{1}\mathrm{d}\mu'\psi^{(n+1)}(x,\mu') = \frac{\nu\Sigma_\mathrm{f}}{2k^{(n)}}\int_{-1}^{1}\mathrm{d}\mu'\psi^{(n)}(x,\mu')\ , \tag{A.6}$$

where n is the iteration index. Note that this differs from the standard deterministic source iteration procedure (where scattering is on the right-hand side and fixed at the n-th iteration), as the scattering source is fully converged at each iteration during Monte Carlo calculations. This treatment also differs from deterministic methods in that no account of spatial or angular discretisation is taken, as Monte Carlo treats both as continuous.

The angular flux at iteration n can be expressed as the converged solution plus perturbation terms:

$$\psi^{(n)}(x,\mu) = \psi_0 + \epsilon\psi_1^{(n)}(x,\mu) + \mathcal{O}(\epsilon^2)\ , \tag{A.7}$$

where ψ_0 is the converged solution, ϵ is a small parameter, $\psi_1^{(n)}$ is a first-order perturbation to the angular flux, and $\mathcal{O}(\epsilon^2)$ contains all higher-order terms in ϵ. The standard Fourier

series representation for perturbations to the angular flux is:

$$\psi_1^{(n)}(x,\mu) = \sum_{\omega=-\infty}^{\infty} \psi_\omega^{(n)}(\mu) e^{i\omega x} \ , \tag{A.8}$$

where $\psi_\omega(\mu)$ is the cosine-dependent Fourier coefficient for Fourier frequency ω, allowable values of which are determined by the chosen boundary conditions. As for diffusion, it is straightforward to show that first-order perturbations to the eigenvalue are zero after a single iteration.

Inserting Eq. (A.8) into Eq. (A.6), and considering orthogonal Fourier modes, one obtains:

$$(i\omega\mu + \Sigma_\mathrm{t})\, \psi_\omega^{(n+1)}(\mu) - \frac{\Sigma_\mathrm{s}}{2} \int_{-1}^{1} \mathrm{d}\mu' \psi_\omega^{(n+1)}(\mu') = \frac{\Sigma_\mathrm{a}}{2} \int_{-1}^{1} \mathrm{d}\mu' \psi_\omega^{(n)}(\mu') \ . \tag{A.9}$$

Dividing through by $i\omega\mu + \Sigma_\mathrm{t}$ and integrating with respect to μ between -1 and 1, one obtains:

$$\int_{-1}^{1} \mathrm{d}\mu \psi_\omega^{(n+1)}(\mu) - \Sigma_\mathrm{s} \frac{\arctan(\frac{\omega}{\Sigma_\mathrm{t}})}{\omega} \int_{-1}^{1} \mathrm{d}\mu' \psi_\omega^{(n+1)}(\mu') = \Sigma_\mathrm{a} \frac{\arctan(\frac{\omega}{\Sigma_\mathrm{t}})}{\omega} \int_{-1}^{1} \mathrm{d}\mu' \psi_\omega^{(n)}(\mu') \ . \tag{A.10}$$

The integrations over μ and μ' are identical and so the dummy variable of μ' in the latter can be replaced with μ. Hence, one obtains:

$$\int_{-1}^{1} \mathrm{d}\mu \psi_\omega^{(n+1)}(\mu) = \frac{\Sigma_\mathrm{a}}{\frac{\omega}{\arctan(\frac{\omega}{\Sigma_\mathrm{t}})} - \Sigma_\mathrm{s}} \int_{-1}^{1} \mathrm{d}\mu \psi_\omega^{(n)}(\mu) \ , \tag{A.11}$$

or, expressing this in terms of scalar flux Fourier coefficients, given Eq. (A.4):

$$\phi_\omega^{(n+1)} = \frac{\Sigma_\mathrm{a}}{\frac{\omega}{\arctan(\frac{\omega}{\Sigma_\mathrm{t}})} - \Sigma_\mathrm{s}} \phi_\omega^{(n)} \ , \tag{A.12}$$

or, by making the common substitution $\omega = \Sigma_\mathrm{t}\lambda$, where λ is a non-dimensionalised Fourier frequency, this can be written in terms of the scattering ratio, $c = \Sigma_\mathrm{s}/\Sigma_\mathrm{t}$, as:

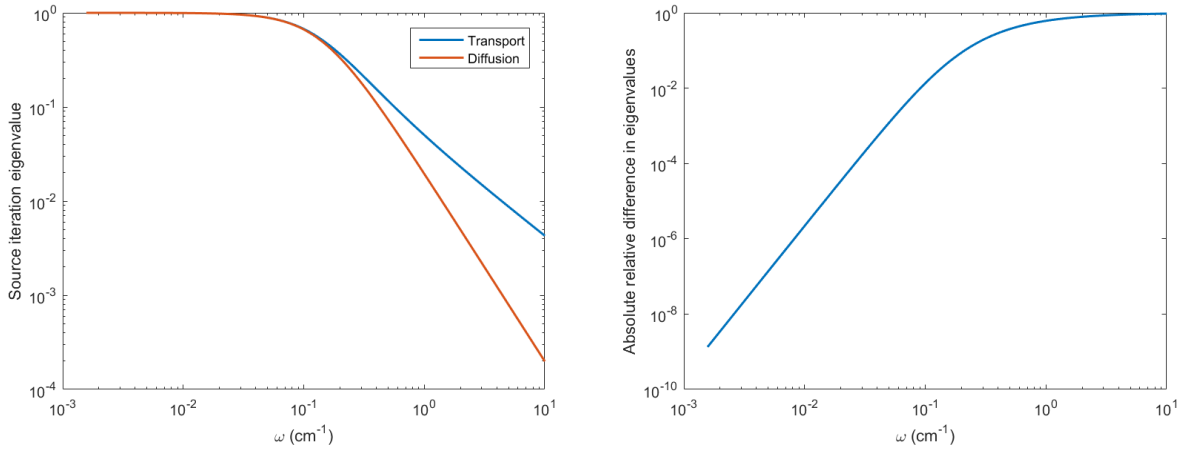
$$\phi_\omega^{(n+1)} = \frac{1-c}{\frac{\lambda}{\arctan(\lambda)} - c} \phi_\omega^{(n)} \ . \tag{A.13}$$

These expressions are exact, but not identical to the diffusion iteration eigenvalue given by Eq. (21). However, about $\omega/\Sigma_\mathrm{t} \approx 0$, one can expand the arctangent as:

$$\arctan\left(\frac{\omega}{\Sigma_\mathrm{t}}\right) = \frac{\omega}{\Sigma_\mathrm{t}} - \frac{1}{3}\left(\frac{\omega}{\Sigma_\mathrm{t}}\right)^3 + \frac{1}{5}\left(\frac{\omega}{\Sigma_\mathrm{t}}\right)^5 - \frac{1}{7}\left(\frac{\omega}{\Sigma_\mathrm{t}}\right)^7 + \dots \ . \tag{A.14}$$

Truncating this expression to the third power in ω/Σ_t, inserting it into Eq. (A.12), cancelling ω's and expanding the denominator up to $(\omega/\Sigma_\mathrm{t})^2$, one obtains:

$$\phi_\omega^{(n+1)} = \frac{\Sigma_\mathrm{a}}{\Sigma_\mathrm{a} + \frac{\omega^2}{3\Sigma_\mathrm{t}}} \phi_\omega^{(n)} \ . \tag{A.15}$$



(a) Iteration eigenvalues of transport and diffusion

(b) Relative difference in iteration eigenvalues

Fig. A.7. Source iteration eigenvalues for transport and diffusion (from Eqs. (A.12) and (A.16)) and their relative difference.

Simplifying further, and taking the total cross-section to be the transport-corrected cross-section, one obtains the diffusion source iteration eigenvalue seen previously:

$$\phi_\omega^{(n+1)} = \frac{1}{1+\omega^2 L_D^2}\phi_\omega^{(n)} \ , \tag{A.16}$$

where $L_D = \sqrt{\frac{D}{\Sigma_\mathrm{a}}}$ is the diffusion length as defined previously. Hence, for low frequencies, Monte Carlo transport should be reasonably approximated by diffusion. Additionally, replacing ω with λ as was done for the transport eigenvalue, one can obtain the result in terms of the scattering ratio:

$$\phi_\omega^{(n+1)} = \frac{1}{1+\frac{\lambda^2}{3(1-c)}}\phi_\omega^{(n)} \ . \tag{A.17}$$

The validity of the diffusion approximation across a range of frequencies can be numerically investigated for the system that was considered in Section 4. Using the same cross-sections, the difference between the transport and diffusion iteration eigenvalues are shown in Fig. A.7. Although there is substantial difference between these iteration eigenvalues for high frequencies, the eigenvalues are small in this regime and thus the Fourier amplitudes associated with these frequencies quickly diminish regardless. On the other hand, transport and diffusion agree well for the low frequencies which are much more problematic in the stability analyses with which this work is concerned. In the numerical Monte Carlo examples considered in Section 4, the 100 cm problem sustains the highest first-mode frequency at about 0.06 cm^{-1} where the relative difference between the transport and diffusion eigenvalues is less than 1%. Hence, the use of the diffusion approximation for this work when discussing Monte Carlo neutron transport is justifiable.

References

Adams, M.L., Larsen, E.W., 2002. Fast iterative methods for discrete-ordinates particle transport calculations. Progress in Nuclear Energy 40, 3–159. doi:10.1016/S0149-1970(01)00023-3.

Cho, N., Park, C., 2003. A Comparison of Coarse Mesh Rebalance and Coarse Mesh Finite Difference Acceleration for the Neutron Transport Calculations, in: Proc. M&C 2003, Gatlinburg, Tennessee.

Choi, S., Kim, W., Choe, J., Lee, W., Kim, H., Ebiwonjumi, B., Jeong, E., Kim, K., Yun, D., Lee, H., Lee, D., 2021. Development of high-fidelity neutron transport code STREAM. Computer Physics Communications 264, 107915. doi:10.1016/J.CPC.2021.107915.

Choi, S., Lee, D., 2021. Three-dimensional method of characteristics/diamond-difference transport analysis method in STREAM for whole-core neutron transport calculation. Computer Physics Communications 260, 107332. doi:10.1016/J.CPC.2020.107332.

Cosgrove, P., Shwageraus, E., Parks, G.T., 2020. Stability analysis of predictor-corrector schemes for coupling neutronics and depletion. Annals of Nuclear Energy 149. doi:10.1016/j.anucene.2020.107781.

Densmore, J.D., Gill, D.F., Griesheimer, D.P., 2013. Stability analysis of burnup calculations. Transactions of the American Nuclear Society 109, 695–698. doi:10.1007/s10967-012-2210-3.2.

Duderstadt, J.J., Hamilton, L.J., 1976. Nuclear Reactor Analysis. John Wiley and Sons, Inc., New York.

Facchini, A., Lee, J., Joo, H.G., 2021. Investigation of Anderson acceleration in neutronics-thermal hydraulics coupled direct whole core calculation. Annals of Nuclear Energy 153, 108042. doi:10.1016/j.anucene.2020.108042.

Gill, D., 2020. Inline thermal and xenon feedback iterations in Monte Carlo reactor calculations, in: Proc. PHYSOR 2020, Cambridge, UK.

Gill, D.F., Griesheimer, D.P., Aumiller, D.L., 2017. Numerical Methods in Coupled Monte Carlo and Thermal-Hydraulic Calculations. Nuclear Science and Engineering 185, 194–205. doi:10.13182/NSE16-3.

Griesheimer, D., Gill, D., Nease, B., Sutton, T., Stedry, M., Dobreff, P., Carpenter, D., Trumbull, T., Caro, E., Joo, H., Millman, D., 2015. MC21 v.6.0 – A continuous-energy Monte Carlo particle transport code with integrated reactor feedback capabilities. Annals of Nuclear Energy 82, 29–40. doi:10.1016/j.anucene.2014.08.020.

Griesheimer, D.P., 2010. In-line xenon convergence algorithm for Monte Carlo reactor calculations, in: Proc. PHYSOR 2010, Pittsburgh, Pennsylvania.

Hamilton, S., Berrill, M., Clarno, K., Pawlowski, R., Toth, A., Kelley, C.T., Evans, T., Philip, B., 2016. An assessment of coupling algorithms for nuclear reactor core physics simulations. Journal of Computational Physics 311, 241–257. doi:10.1016/j.jcp.2016.02.012.

Isotalo, A., 2013. Computational Methods for Burnup Calculations with Monte Carlo Neutronics. Ph.D. thesis. Aalto University.

Isotalo, A.E., Leppänen, J., Dufek, J., 2013. Preventing xenon oscillations in Monte Carlo burnup calculations by enforcing equilibrium xenon distribution. Annals of Nuclear Energy 60, 78–85. doi:10.1016/j.anucene.2013.04.031.

Jarrett, M., Kochunas, B., Zhu, A., Downar, T., 2016. Analysis of stabilization techniques for CMFD acceleration of neutron transport problems. Nuclear Science and Engineering 184, 208–227. doi:10.13182/NSE16-51.

Keady, K.P., Larsen, E.W., 2016. Stability of Monte Carlo k-eigenvalue simulations with CMFD feedback. Journal of Computational Physics 321, 947–964. doi:10.1016/j.jcp.2016.06.002.

Kochunas, B., Collins, B., Downar, T., Martin, W., 2013. MPACT: Michigan Parallel Advanced Characteristics Transport, in: Proc. M&C 2013, Sun Valley, Idaho.

Kochunas, B., Collins, B., Stimpson, S., Salko, R., Jabaay, D., Graham, A., Liu, Y., Kim, K.S., Wieselquist, W., Godfrey, A., Clarno, K., Palmtag, S., Downar, T., Gehin, J., 2017a. VERA core simulator methodology for pressurized water reactor cycle depletion. Nuclear Science and Engineering 185, 217–231. doi:10.13182/NSE16-39.

Kochunas, B., Fitzgerald, A., Larsen, E., 2017b. Fourier analysis of iteration schemes for k-eigenvalue transport problems with flux-dependent cross sections. Journal of Computational Physics 345, 294–307. doi:10.1016/j.jcp.2017.05.028.

Kreher, M., Forget, B., Smith, K., 2019. Single-Batch Monte Carlo Multiphysics Coupling, in: Proc. M&C 2019, Portland, Oregon.

Leppänen, J., Pusa, M., Viitanen, T., Valtavirta, V., Kaltiaisenaho, T., 2015. The Serpent Monte Carlo code: Status, development and applications in 2013. Annals of Nuclear Energy 82, 142–150. doi:10.1016/j.anucene.2014.08.024.

MPACT Team, 2016. MPACT Standard Input User's Manual Version 2.2.0. Technical Report. Consortium for the Advanced Simulation of Light Water Reactors.

MPACT Team, 2019. MPACT Theory Manual Version 4.1. Technical Report. Consortium for Advanced Simultion of Light Water Reactors.

Senecal, J.P., Ji, W., 2017. Approaches for mitigating over-solving in multiphysics simulations. International Journal for Numerical Methods in Engineering 112, 503–528. doi:10.1002/nme.5516.

Shen, Q., Adamowicz, N., Kochunas, B., 2019a. Relationship Between Relaxation and Partial Convergence of Nonlinear Diffusion Acceleration for Problems with Feedback, in: Proc. M&C 2019, Portland, Oregon.

Shen, Q., Adamowicz, N., Kochunas, B., Larsen, E.W., 2019b. X-CMFD: A Robust Iteration Scheme for CMFD-Based Acceleration of Neutron Transport Problems with Nonlinear Feedback, in: Proc. M&C 2019, Portland, Oregon.

Shen, Q., Choi, S., Kochunas, B., 2021. A Robust Relaxation-free Multiphysics Iteration Scheme for CMFD–Accelerated Neutron Transport k–eigenvalue Calculation–II: Numerical Results. Nuclear Science and Engineering doi:10.1080/00295639.2021.1906586.

Shen, Q., Kochunas, B., 2021. A Robust Relaxation-Free Multiphysics Iteration Scheme for CMFD-Accelerated Neutron Transport k-Eigenvalue Calculations — I: Theory. Nuclear Science and Engineering doi:10.1080/00295639.2021.1906585.

Smith, K., 1983. Nodal Method Storage Reduction by Nonlinear Iteration. Transactions of the American Nuclear Society 44, 265–266.

Smith, K., Rhodes III, J., 2002. Full-Core, 2-D, LWR Core Calculations with CASMO-4E, in: Proc. PHYSOR 2002, Seoul, Korea.