LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-JRNL-864001

The Transient Multi-Level Method for Monte Carlo Reactor Statics Calculations

E. S. Gonzalez, B. C. Kiedrowski, G. G. Davidson

May 6, 2024

Annals of Nuclear Energy

Disclaimer

The Transient Multi-Level Method for Monte Carlo Reactor Statics Calculations

Evan S. Gonzalez[a,*], Brian C. Kiedrowski[b], Gregory G. Davidson[c]

[a]*Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94551*
[b]*Nuclear Engineering and Radiological Sciences, University of Michigan, 2355 Bonisteel Blvd., Ann Arbor, MI 48109, USA*
[c]*Oak Ridge National Laboratory, 1 Bethel Valley Rd., Oak Ridge, TN 37830, USA*

Abstract

The Transient Multi-Level (TML) method is applied to a time-dependent Monte Carlo transport solver to offload some of the computational burden of the expensive Monte Carlo solve to lower-order Coarse Mesh Finite Difference (CMFD) and Exact Point Kinetics Equations (EPKE) solvers via factorization of the neutron flux at the transport and CMFD levels using the Predictor Corrector Quasi-Static Method (PCQM). The Monte Carlo transient is solved by a modified fission source iteration scheme that introduces a single transient source bank. The method is implemented in the production-level Monte Carlo code, Shift, and verified with prescribed reactivity ramps from the two-dimensional version of the C5G7-TD reactor benchmark. The results show that, as compared to other quasi-static methods, the TML reduces the stochastic noise inherent to the transient Monte Carlo solver by factors of $\sim$2 to 6 for various norm comparisons of the reactor power ampli-

*Corresponding author. Work completed at University of Michigan. Transcript prepared at Lawrence Livermore National Laboratory.

Email addresses: gonzalez121@llnl.gov (Evan S. Gonzalez), bckiedro@umich.edu (Brian C. Kiedrowski), davidsongg@ornl.gov (Gregory G. Davidson)

tude. The TML additionally reduces the number of Monte Carlo evaluations needed to simulate the transient, leading to roughly an order of magnitude improvement in CPU time relative to the standard PCQM for the problems tested.
Keywords: transient, quasi-static, Monte Carlo

1. Introduction

Until the mid-2000s reactor transient calculations were predominantly limited to point kinetics, where the spatial, energy, and angular dependence are integrated out of the underlying neutron transport and delayed neutron precursor density equations to reduce the problem to a single "point" reactor model. In the 1950s, Henry and a series of colleagues observed that loosely-coupled reactor physics problems could be solved with sufficient accuracy by factorizing the neutron flux into the product of a time-dependent amplitude function solved by point kinetics on a fine time scale and a shape function with weak time-dependence solved by neutron diffusion on a coarse time scale [1, 2, 3]. In the late 60s, Ott and Meneley improved the accuracy of the method by introducing updates to the shape function with an iterative procedure to develop the Improved Quasi-Static Method (IQM) [4].

Gehin's 1992 dissertation introduced the Predictor-Corrector Quasi-static Method (PCQM), which avoided the costly IQM iterative procedure to determine the shape function by updating the entire neutron flux on the coarse time scale as a "predictor" and correcting the solution with the fine time-resolution amplitude function [5]. The method received little attention with respect to computational efficiency at the time of publication but was re-

investigated by Dulla et al. in the mid-2000's [6, 7]. In their analysis, Dulla et al. showed the PCQM to be well-suited for implementations requiring high-fidelity solvers because it typically requires fewer evaluations on the coarse time scale than the IQM. On the other hand, several studies over the last few years have suggested that the IQM might actually be better-suited for multiphysics calculations because it allows for iterative communication between the neutronics and thermal hydraulics solvers [8, 9, 10]. Between the two, the clearly-superior technique has yet to be established, as both methods have started to see transport-level applications in recent years.

Zhu improved the speed and accuracy of the PCQM in the deterministic Michigan Parallel Characteristics Transport (MPACT) code by introducing an intermediate time-discretization [11]. In what Zhu calls the Transient Multi-Level (TML) method, the Coarse Mesh Finite Difference (CMFD) equations are used to propagate the amplitude function forward in time at the intermediate time scale to preserve some spatial and energy dependence in between Method of Characteristics (MOC) transport solutions. Shen showed further improvement by introducing a fourth, energy-integrated, CMFD time discretization level to the TML between the CMFD and point kinetics levels [12]. Shen additionally characterized Lie and Strang operator splitting for multiphysics schemes that might be relevant for future applications of this work.

With huge strides in computational capacity, the previous decade has also seen several reactor kinetics implementations that make use of Monte Carlo (MC) transport solvers. MC solvers offer numerous advantages over deterministic solvers (at the cost of computational expense), including not re-

quiring spatial, angular, or energy discretization. Sjenitzer and Hoogenboom showed that explicitly tracking the inventory of delayed neutron precursors produces very accurate results, but is prohibitively expensive for practical applications [13, 14]. Hackemack et al. implemented the PCQM with fixed-source Monte Carlo transport at the coarse time step but found that the performance was limited by long fission chains [15]. Kooreman and Griesheimer expanded on the work of Hackemack et al. by introducing the MC-PCQM algorithm into a multiphysics solver [10].

Jo et al. solved the issue of non-terminating fission chains by solving the Monte Carlo step with fission source iteration [16]. Lee et al. improved fission source convergence rates by introducing the CMFD equations to a MC solver [17, 18]. Shaner developed a frequency transform method in 2018 (based on a deterministic technique developed by Ban et al. [19]) that uses time-dependent CMFD equations (in a distinct formulation from those developed by Zhu) to propagate the space- and energy-dependent frequencies forward in time so that they can inform time-dependent updates to the Monte Carlo solution [20]. Kreher et al. incorporated thermal-hydraulic feedback into Shaner's frequency transform method [21]. Recently, He et al. and Mascolino and Haghighat have had success in reducing simulation times of hybrid MC-deterministic transient calculations with methods based on the Transient Fission Matrix technique [22, 23].

This work applies a MC solver at the transport level of Zhu's TML method [11]. The transient MC criticality solver is largely modeled after Jo's transient criticality solver implementation [16] with a collapsed number of particle source banks (from five to two). The remainder of this paper is organized

as follows. Section 2 gives a high-level overview of the TML and describes the detailed methodologies for the three solver levels. Section 3 describes the implementation details specific to the transient MC criticality, CMFD, and Exact Point Kinetics Equations (EPKE) solver levels of the TML and how they are coupled together. Section 4 describes the results of the method applied to a 2D multigroup reactor benchmark. Finally, conclusions are given in Section 5.

2. Transient Methodologies

This section begins with a high-level overview of the TML method being described in this paper. Then, the equations being solved by the three transient subsolvers are defined: the transport-level Transient Fixed Source Problem (TFSP), the diffusion-level TFSP, and the EPKEs. Figure 1 shows the temporal discretization being solved by the TML, where a high-fidelity (but computationally expensive) MC solver is used with a coarse time step, a time-dependent CMFD solver that applies the diffusion approximation is used with an intermediate time step, and an EPKEs solver that has integrated out the spatial and energy dependence of the transient is used with a fine time step.

2.1. Transient Multi-Level Method

The TML method consists of two nested levels of PCQM temporal discretizations [5] [7], where each of the discretizations is formulated by a separation of variables of the neutron flux into the product of an amplitude function, and a shape function. The amplitude function is characterized by reduced phase space dimensionality and strong temporal dependence. The

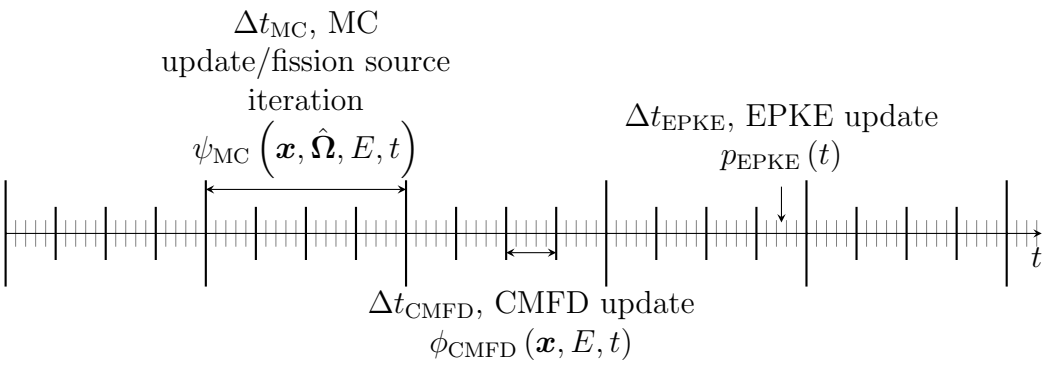

Figure 1: TML time stepping scheme [11].

shape function, which typically retains full phase space dimensionality, is assumed to have weak temporal dependence and is generally more computationally expensive than the amplitude function. Quasi-static methods take advantage of these characteristics to update the two functions on separate time scales, where $\Delta t_{\mathrm{shape}} > \Delta t_{\mathrm{amplitude}}$. In this way, much of the computational work in capturing time-dependent features of the neutron flux is off-loaded to lower-order approximate amplitude solvers.

The PCQM updates the entire neutron flux on the macro time scale, as opposed to other quasi-static methods that only update the shape function on the macro time scale. This initial updated flux solution is the "predictor" flux and typically has errors associated with integrating over a large time step. The algorithm proceeds by updating the amplitude function on the micro time scale to "correct" the flux estimate. The main benefit of the PCQM algorithm is that it eliminates an expensive inner iteration that would be required by other quasi-static methods.

The TML method [11] extends the predictor-corrector quasi-static method by introducing an intermediate nested time discretization level, shown in Fig. 1. As with the PCQM, the angular flux is computed with a high-fidelity solver and factorized into the product of amplitude and shape functions. The amplitude function at the transport level is now the solution of the CMFD equations discussed in Sec. 2.3 and is used to correct the transport-level predictor flux. The CMFD solution is further factorized into the product of another shape and amplitude function, where instead the amplitude is the solution of the EPKEs discussed in Sec. 2.4 and used to update the CMFD-level predictor flux.

The main benefit of adding the extra CMFD level is that it allows the spatial and energy dependence of the shape function to be updated in between MC time steps that would not otherwise be captured by the globally-integrated point kinetics equations. It will be shown that the addition of the CMFD level also allows for a larger transport-level macro time step, improving the overall speed of the transient calculation.

2.2. Transient Fixed Source Problem

At the highest-fidelity level, the MC method is used to solve the underlying time-dependent neutron transport problem. The time-dependent Boltzmann neutron transport equation and delayed neutron precursor equations

are

$$\frac{1}{v(E)}\frac{\partial\psi}{\partial t}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)+\boldsymbol{M}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)=\frac{\chi_{\mathrm{p}}(\boldsymbol{x},E,t)}{4\pi k_{\mathrm{eff}}^{0}}\boldsymbol{F}_{\mathrm{p}}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)+\sum_{j=1}^{J}\frac{\chi_{\mathrm{d},j}(\boldsymbol{x},E,t)}{4\pi}S_{\mathrm{d},j}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)\,, \tag{1a}$$

$$\frac{\partial C_j}{\partial t}(\boldsymbol{x},t)=\frac{1}{k_{\mathrm{eff}}^{0}}\boldsymbol{F}_{\mathrm{d},j}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)-\lambda_j(\boldsymbol{x},t)C_j(\boldsymbol{x},t)\,, \tag{1b}$$

where the basic symbols that appear in Eq. (1) are defined in Table A.3 and the operators are defined such that $\boldsymbol{M}$ corresponds to particle migration (streaming, collisions, and scattering), $\boldsymbol{F}_{\mathrm{p}}$ corresponds to prompt fission, $\boldsymbol{F}_{\mathrm{d},j}$ corresponds to delayed fission and $S_{\mathrm{d},j}$ corresponds to an accumulated delayed neutron source term,

$$\begin{aligned}&\boldsymbol{M}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)\\&=\boldsymbol{\nabla}\cdot\hat{\boldsymbol{\Omega}}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)+\Sigma_t(\boldsymbol{x},E,t)\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)\\&-\int_0^\infty\int_{4\pi}\Sigma_{\mathrm{s}}(\boldsymbol{x},\hat{\boldsymbol{\Omega}}\cdot\hat{\boldsymbol{\Omega}}',E'\to E,t)\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}}',E',t)\,\mathrm{d}\Omega'\,\mathrm{d}E'\,,\end{aligned} \tag{2a}$$

$$\begin{aligned}&\boldsymbol{F}_{\mathrm{p}}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)\\&=\int_0^\infty\int_{4\pi}\left[1-\beta(\boldsymbol{x},E',t)\right]\nu\Sigma_{\mathrm{f}}(\boldsymbol{x},E',t)\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}}',E',t)\,\mathrm{d}\Omega'\,\mathrm{d}E'\,,\end{aligned} \tag{2b}$$

$$\begin{aligned}&\boldsymbol{F}_{\mathrm{d},j}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)\\&=\int_0^\infty\int_{4\pi}\beta_j(\boldsymbol{x},E',t)\nu\Sigma_{\mathrm{f}}(\boldsymbol{x},E',t)\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}}',E',t)\,\mathrm{d}\Omega'\,\mathrm{d}E'\,,\end{aligned} \tag{2c}$$

$$S_{\mathrm{d},j}(\boldsymbol{x},t)=\lambda_j(\boldsymbol{x},t)C_j(\boldsymbol{x},t)\,. \tag{2d}$$

In Eqs. (1), the appearance of the initial k-eigenvalue, k^0_{eff}, serves to normalize the terms that contain the fission source integral and ensures criticality of the initial state. For this work, an implicit Euler scheme is selected to discretize the temporal domain of Eq. (1) with the superscript index, n, denoting discretized points in time. Integrating over the range, $t^{n-1} \leq t \leq t^n$, Eq. (1) becomes

$$\begin{aligned}
&\frac{1}{v(E)\Delta t^n}\left[\psi^n(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E) - \psi^{n-1}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)\right] + \boldsymbol{M}\psi^n(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E) \\
&\qquad = \frac{\chi^n_{\text{p}}(\boldsymbol{x},t)}{4\pi k^0_{\text{eff}}}\boldsymbol{F}_{\text{p}}\psi^n(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E) + \sum_{j=1}^{J}\frac{\chi^n_{\text{d},j}(\boldsymbol{x},t)}{4\pi}S^n_{\text{d},j}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)\,, \qquad \text{(3a)} \\
&C^n_j(\boldsymbol{x}) = C^{n-1}_j(\boldsymbol{x})f_{1,j} + \frac{1}{k^0_{\text{eff}}}\boldsymbol{F}_{\text{d},j}\psi^{n-1}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)f_{2,j} \\
&\qquad + \frac{1}{k^0_{\text{eff}}}\boldsymbol{F}_{\text{d},j}\psi^n(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)f_{3,j}\,. \qquad \text{(3b)}
\end{aligned}$$

The time-interpolated weighting factors are derived by approximating the fission source as varying linearly in time and integrating with Lagrange polynomial interpolation. These weighting factors are

$$f_{1,j} = e^{-\lambda_j\Delta t^n}\,, \qquad \text{(4a)}$$

$$f_{2,j} = \frac{1 - e^{-\lambda_j\Delta t^n} - \lambda_j\Delta t^n e^{-\lambda_j\Delta t^n}}{\lambda_j\Delta t^n}\,, \qquad \text{(4b)}$$

$$f_{3,j} = \frac{e^{-\lambda_j\Delta t^n}\left(1 - e^{\lambda_j\Delta t^n} + \lambda_j\Delta t^n e^{\lambda_j\Delta t^n}\right)}{\lambda_j\Delta t^n}\,, \qquad \text{(4c)}$$

and a detailed integration can be found in Appendix A of [24].

To integrate Eq. (1b), the delayed fission term $\frac{1}{k^0_{\text{eff}}}\boldsymbol{F}_{\text{d},j}\psi(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E,t)$ is approximated as varying linearly in time. Higher-order approximations are

possible, but can lead to negative weighting factors analogous to Eqs. (4). These negative weighting factors can lead to negative MC particle weights, which are generally avoided as they have no physical interpretation, can lead to increased variance, and can complicate other variance reduction schemes that might be implemented in a production code. After integrating Eq. (1b) and substituting into Eq. (3a), the TFSP is

$$
\begin{aligned}
(\boldsymbol{M} + \frac{1}{v(E)\Delta t^n})\psi^n(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E) = & \frac{\psi^{n-1}(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E)}{v(E)\Delta t^n} \\
& + \frac{1}{4\pi k^0_{\text{eff}}} \Big[\chi^n_{\text{p}}(\boldsymbol{x}, E) \boldsymbol{F}_{\text{p}} \psi^n(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E) \\
& + \sum_{j=1}^{J} \chi^{n-1}_{\text{d,j}}(\boldsymbol{x}, E) S^{n-1}_{\text{d},j}(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E) f_{1,j} \\
& + \sum_{j=1}^{J} \chi^{n-1}_{\text{d,j}}(\boldsymbol{x}, E) \boldsymbol{F}_{\text{d},j} \psi^{n-1}(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E) f_{2,j} \\
& + \sum_{j=1}^{J} \chi^{n}_{\text{d,j}}(\boldsymbol{x}, E) \ \boldsymbol{F}_{\text{d},j} \psi^{n}(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E) f_{3,j} \Big] . \qquad (5)
\end{aligned}
$$

2.3. Time-Dependent Coarse Mesh Finite Difference Equations

To formulate the time-dependent CMFD equations, the TFSP is derived from the continuous-energy diffusion equation:

$$\begin{aligned}\frac{1}{v\left(E\right)}\frac{\partial\phi\left(\boldsymbol{x},E,t\right)}{\partial t} =&\boldsymbol{\nabla}\cdot D\left(\boldsymbol{x},E,t\right)\boldsymbol{\nabla}\phi\left(\boldsymbol{x},E,t\right)-\Sigma_{\mathrm{t}}\left(\boldsymbol{x},E,t\right)\phi\left(\boldsymbol{x},E,t\right)\\ &+\int_{0}^{\infty}\Sigma_{\mathrm{s0}}\left(\boldsymbol{x},E'\rightarrow E,t\right)\phi\left(\boldsymbol{x},E',t\right)\,\mathrm{d}E'\\ &+\chi_{\mathrm{p}}\left(\boldsymbol{x},E,t\right)\left(1-\beta\left(\boldsymbol{x},t\right)\right)\frac{1}{k_{\mathrm{eff}}^{0}}\boldsymbol{F}\phi\left(\boldsymbol{x},E,t\right)\\ &+\chi_{\mathrm{d}}\left(\boldsymbol{x},E,t\right)S_{\mathrm{d}}\left(\boldsymbol{x},E,t\right)\,, \qquad (6a)\\ \frac{\partial C_{j}}{\partial t}\left(\boldsymbol{x},t\right) =&\beta_{j}\left(\boldsymbol{x},t\right)\frac{1}{k_{\mathrm{eff}}^{0}}\boldsymbol{F}\phi\left(\boldsymbol{x},E,t\right)-\lambda_{j}\left(\boldsymbol{x},t\right)C_{j}\left(\boldsymbol{x},t\right)\,, \qquad (6b)\end{aligned}$$

where the delayed neutron fractions have been averaged over energy, linear anisotropy has been assumed, and

$$\boldsymbol{F}\phi(\boldsymbol{x},E,t)\equiv\int_{0}^{\infty}\nu\Sigma_{\mathrm{f}}(\boldsymbol{x},E',t)\phi(\boldsymbol{x},E',t)\,\mathrm{d}E'\,. \qquad (7)$$

Zhu [11] derives the diffusion-level TFSP by applying a backward Euler time discretization to Eqs. (6) and integrating over the range $t^{n-1}\leq t\leq t^{n}$ to arrive at:

$$\begin{aligned}\frac{\phi^{n}(\boldsymbol{x},E)-\phi^{n-1}(\boldsymbol{x},E)}{v(E)\Delta t^{n}} =&\boldsymbol{\nabla}\cdot D^{n}(\boldsymbol{x},E)\boldsymbol{\nabla}\phi^{n}(\boldsymbol{x},E)-\Sigma_{\mathrm{t}}^{n}(\boldsymbol{x},E)\phi^{n}(\boldsymbol{x},E)\\ &+\int_{0}^{\infty}\Sigma_{\mathrm{s0}}^{n}(\boldsymbol{x},E'\rightarrow E)\phi^{n}(\boldsymbol{x},E')\,\mathrm{d}E'\\ &+\chi_{\mathrm{p}}^{n}(\boldsymbol{x},E)(1-\beta^{n}(\boldsymbol{x}))\frac{1}{k_{\mathrm{eff}}^{0}}\boldsymbol{F}\phi^{n}(\boldsymbol{x},E)\\ &+\chi_{\mathrm{d}}^{n}(\boldsymbol{x},E)S_{\mathrm{d}}^{n}(\boldsymbol{x},E)\,. \qquad (8)\end{aligned}$$

Since negative particle weights are not of concern in deterministic solves, the fission source can be approximated as having higher-order (quadratic) temporal dependence. The resulting interpolated delayed neutron and fission source terms are

$$\boldsymbol{F}\phi^n(\boldsymbol{x},E) \equiv \int_0^\infty \nu\Sigma_{\mathrm{f}}^n(\boldsymbol{x},E')\phi^n(\boldsymbol{x},E')\,\mathrm{d}E'\,. \tag{9a}$$

$$S_{\mathrm{d}}^n(\boldsymbol{x},E) = \omega^n(\boldsymbol{x})\frac{1}{k_{\mathrm{eff}}^0}\boldsymbol{F}\phi^n(\boldsymbol{x},E) + \tilde{S}_{\mathrm{d}}^{n-1}(\boldsymbol{x},E)\ , \tag{9b}$$

$$\omega^n(\boldsymbol{x}) = \sum_{j=1}^J \beta_j^n(\boldsymbol{x})\Omega_j^n(\lambda_j^n(\boldsymbol{x}))\ , \tag{9c}$$

$$\begin{aligned}\tilde{S}_d^{n-1}(\boldsymbol{x},E) = &\sum_{j=1}^J \lambda_j^n(\boldsymbol{x})\Omega_j^0(\lambda_j^n(\boldsymbol{x}))C_j^{n-1}(\boldsymbol{x}) \\ &+ \frac{1}{k_{\mathrm{eff}}^0}\boldsymbol{F}\phi^{n-1}(\boldsymbol{x},E)\sum_{j=1}^J \beta_j^{n-1}(\boldsymbol{x})\Omega_j^{n-1}(\lambda_j^n(\boldsymbol{x})) \\ &+ \frac{1}{k_{\mathrm{eff}}^0}\boldsymbol{F}\phi^{n-2}(\boldsymbol{x},E)\sum_{j=1}^J \beta_j^{n-2}(\boldsymbol{x})\Omega_j^{n-2}(\lambda_j^n(\boldsymbol{x}))\,.\end{aligned} \tag{9d}$$

The time-interpolated weighting factors here are derived in a similar fashion to those defined in Eq. (4). As noted above, the fission source is now approximated as having quadratic time dependence, which yields the following weighting factors using Lagrange polynomial interpolation and integrating

over the time step $t^{n-1} \leq t \leq t^n$:

$$\Omega_j^0(\tilde{\lambda}_j^n(\boldsymbol{x})) = e^{-\tilde{\lambda}_j^n(\boldsymbol{x})}\,, \tag{10a}$$

$$\Omega_j^n(\tilde{\lambda}_j^n(\boldsymbol{x})) = \frac{\kappa_2(\tilde{\lambda}_j^n(\boldsymbol{x})) + \gamma\kappa_1(\tilde{\lambda}_j^n(\boldsymbol{x}))}{(1+\gamma)}\,, \tag{10b}$$

$$\Omega_j^{n-1}(\tilde{\lambda}_j^n(\boldsymbol{x})) = \left[\Omega_j^0(\tilde{\lambda}_j^n(\boldsymbol{x})) - \frac{\kappa_2(\tilde{\lambda}_j^n(\boldsymbol{x})) + (\gamma-1)\kappa_1(\tilde{\lambda}_j^n(\boldsymbol{x}))}{\gamma}\right]\,, \tag{10c}$$

$$\Omega_j^{n-2}(\tilde{\lambda}_j^n(\boldsymbol{x})) = \frac{\kappa_2(\tilde{\lambda}_j^n(\boldsymbol{x})) - \kappa_1(\tilde{\lambda}_j^n(\boldsymbol{x}))}{(1+\gamma)\gamma}\,, \tag{10d}$$

$$\kappa_0(x) = 1 - e^{-x}\,, \tag{10e}$$

$$\kappa_1(x) = 1 - \frac{\kappa_0(x)}{x}\,, \tag{10f}$$

$$\kappa_2(x) = 1 - \frac{2\kappa_1(x)}{x}\,, \tag{10g}$$

$$\tilde{\lambda}_j^n(\boldsymbol{x}) = \lambda_j^n(\boldsymbol{x})\Delta t^n\,. \tag{10h}$$

A detailed derivation is given in Appendix A of [24].

A time-dependent balance equation can then be derived by introducing a cumulative fission spectrum,

$$\chi^n(\boldsymbol{x}, E) = \chi_p^n(\boldsymbol{x}, E)(1-\beta^n(\boldsymbol{x})) + \chi_d^n(\boldsymbol{x}, E)\beta^n(\boldsymbol{x})\,, \tag{11}$$

into Eq. (9a) to arrive at

$$\begin{aligned}
&-\boldsymbol{\nabla}\cdot D^n(\boldsymbol{x}, E)\boldsymbol{\nabla}\phi^n(\boldsymbol{x}, E) + \Sigma_{\mathrm{t}}^n(\boldsymbol{x}, E)\phi^n(\boldsymbol{x}, E) \\
&\quad = \int_0^\infty \Sigma_{\mathrm{s0}}^n(\boldsymbol{x}, E' \to E)\phi^n(\boldsymbol{x}, E')\,\mathrm{d}E' + \chi^n(\boldsymbol{x}, E)\frac{1}{k_{\mathrm{eff}}^0}\boldsymbol{F}\phi^n(\boldsymbol{x}, E) + S_{\mathrm{tr}}^n(\boldsymbol{x}, E)\,.
\end{aligned} \tag{12}$$

The transient source term is

$$S_{\mathrm{tr}}^{n}(\boldsymbol{x},E) = A^{n}(\boldsymbol{x},E)\phi^{n}(\boldsymbol{x},E) + B^{n}(\boldsymbol{x},E)\frac{1}{k_{\mathrm{eff}}^{0}}\boldsymbol{F}\phi^{n}(\boldsymbol{x},E) + C^{n-1}(\boldsymbol{x},E)\,, \quad (13)$$

where

$$A^{n}(\boldsymbol{x},E) = -\frac{1}{v(E)\Delta t^{n}}\ , \quad (14a)$$

$$B^{n}(\boldsymbol{x},E) = \chi_{\mathrm{d}}^{n}(\boldsymbol{x},E)(\omega^{n}(\boldsymbol{x}) - \beta^{n}(\boldsymbol{x}))\ , \quad (14b)$$

$$C^{n-1}(\boldsymbol{x},E) = \chi_{\mathrm{d}}^{n}(\boldsymbol{x},E)\tilde{S}_{\mathrm{d}}^{n-1}(\boldsymbol{x},E) + \frac{\phi^{n-1}(\boldsymbol{x},E)}{v(E)\Delta t^{n}}\,. \quad (14c)$$

In this Multigroup Matrix (MGM) formulation [11], A and B are flux- and fission source-dependent coefficients, and C is a constant coefficient depending on quantities defined at the previous point in time t^{n-1}. The MGM allows the entire transient diffusion problem to be written and solved as a standard linear system.

The transient CMFD equations themselves are derived by discretizing the energy and spatial domains. Discretizing the energy domain is straightforward by applying the multigroup approximation. The spatial domain is discretized by integrating the resulting equation over a coarse mesh volume,

typically spanning a fuel pin cell or assembly, to arrive at

$$\begin{aligned}
&\sum_{u\in x,y,z}\frac{1}{\Delta_k^u}\Big[\Big(-\tilde{D}^{u,n}_{k-\frac{1}{2},l,m,g}-\hat{D}^{u,n}_{k-\frac{1}{2},l,m,g}\Big)\phi^n_{k-1,l,m,g}\\
&\qquad+\Big(\tilde{D}^{u,n}_{k-\frac{1}{2},l,m,g}+\tilde{D}^{u,n}_{k+\frac{1}{2},l,m,g}-\hat{D}^{u,n}_{k-\frac{1}{2},l,m,g}+\hat{D}^{u,n}_{k+\frac{1}{2},l,m,g}\Big)\phi^n_{k,l,m,g}\\
&\qquad+\Big(-\tilde{D}^{u,n}_{k+\frac{1}{2},l,m,g}+\hat{D}^{u,n}_{k+\frac{1}{2},l,m,g}\Big)\phi^n_{k+1,l,m,g}\Big]-\Sigma^n_{\mathrm{t},k,l,m,g}\phi^n_{k,l,m,g}\\
&=\sum_{g'=1}^{G}\Sigma^n_{\mathrm{s0},k,l,m,g'\to g}\phi^n_{k,l,m,g'}+\frac{\chi^n_{k,l,m,g}}{k^0_{\mathrm{eff}}}\sum_{g'=1}^{G}\nu\Sigma^n_{\mathrm{f},k,l,m,g'\to g}\phi^n_{k,l,m,g'}+S^n_{\mathrm{tr},k,l,m,g}\,.
\end{aligned}\tag{15}$$

The subscript g corresponds to the energy group index and the subscripts (k,l,m) are cell indices corresponding to an arbitrary direction set (u,v,w). This notation is borrowed from OpenMC documentation [25] and is used to condense the summation across spatial dimensions. The Δ terms describe the respective length of a CMFD cell. For example, $\Delta^u_k = u_{k+\frac{1}{2}} - u_{k-\frac{1}{2}}$.

The $\hat{D}$ diffusion coefficient is a non-linear coupling term used to relate the neutron current and scalar flux. This non-linear coupling term is defined based on whether the CMFD cell is coupled to another CMFD cell or a problem boundary. For cell-to-cell coupling, the neutron current and scalar flux are related by

$$J^{u,n}_{k\pm\frac{1}{2},l,m,g}=-\tilde{D}^{u,n,\mathrm{cell}}_{k,l,m,g}\left(\pm\phi^n_{k\pm1,l,m,g}\mp\phi^n_{k,l,m,g}\right)+\hat{D}^{u,n,\mathrm{cell}}_{k,l,m,g}\left(\phi^n_{k\pm1,l,m,g}+\phi^n_{k,l,m,g}\right)\,,\tag{16a}$$

$$\tilde{D}^{u,n,\mathrm{cell}}_{k,l,m,g}=\frac{2D^n_{k\pm1,l,m,g}D^n_{k,l,m,g}}{D^n_{k\pm1,l,m,g}\Delta^u_k+D^n_{k,l,m,g}\Delta^u_{k\pm1}}\,.\tag{16b}$$

For cell-to-boundary coupling, the neutron current and scalar flux are related by

$$J^{u,n}_{k\pm\frac{1}{2},l,m,g} = \pm\tilde{D}^{u,n,\text{bound}}_{k,l,m,g}\phi^{n}_{k,l,m,g} + \hat{D}^{u,n,\text{bound}}_{k,l,m,g}\phi^{n}_{k,l,m,g}\,, \tag{16c}$$

$$\tilde{D}^{u,n,\text{bound}}_{k,l,m,g} = \frac{2D^{n}_{k,l,m,g}\left(1-\alpha^{u,n}_{k\pm\frac{1}{2},l,m,g}\right)}{4D^{n}_{k,l,m,g}\left(1+\alpha^{u,n}_{k\pm\frac{1}{2},l,m,g}\right)+\left(1-\alpha^{u,n}_{k\pm\frac{1}{2},l,m,g}\right)\Delta^{u}_{k}}\,, \tag{16d}$$

where α is an albedo boundary condition

$$\alpha^{u,n}_{k\pm\frac{1}{2},l,m,g} = \frac{J^{u,n-}_{k\pm\frac{1}{2},l,m,g}}{J^{u,n+}_{k\pm\frac{1}{2},l,m,g}}\,. \tag{17}$$

2.4. Exact Point Kinetics Equations

The EPKEs are

$$\frac{\mathrm{d}p\left(t\right)}{\mathrm{d}t} = \frac{\rho\left(t\right)-\beta\left(t\right)}{\Lambda\left(t\right)} + \frac{1}{\Lambda\left(0\right)}\sum_{j}\lambda_{j}\left(t\right)\zeta_{j}\left(t\right), \tag{18a}$$

$$\frac{\mathrm{d}\zeta_{j}\left(t\right)}{\mathrm{d}t} = \frac{\Lambda\left(0\right)}{\Lambda\left(t\right)}\beta^{\text{eff}}_{j}\left(t\right)p\left(t\right) - \lambda_{j}\left(t\right)\zeta_{j}\left(t\right)\,, \tag{18b}$$

where the quantities of interest are the core-wise amplitude function p, the time-dependent "dynamic" reactivity ρ, and the adjoint-weighted delayed neutron precursor number densities ζ_j. The adjoint-weighted point reactor kinetics parameters are given in Table A.3. The definitions of these param-

eters are:

$$\zeta_j(t) = \frac{\left\langle \psi_0^\dagger \chi_{\mathrm{d},j} C_j \right\rangle}{\left\langle \psi_0^\dagger \boldsymbol{F}_0 \Psi_0 \right\rangle}, \tag{19a}$$

$$\rho(t) = \frac{\left\langle \psi_0^\dagger (\boldsymbol{F} - \boldsymbol{M}) \Psi \right\rangle}{\left\langle \psi_0^\dagger \boldsymbol{F} \Psi \right\rangle}, \tag{19b}$$

$$\Lambda(t) = \frac{\left\langle \psi_0^\dagger \frac{1}{v} \Psi \right\rangle}{\left\langle \psi_0^\dagger \boldsymbol{F} \Psi \right\rangle}, \tag{19c}$$

$$\beta_j^{\mathrm{eff}}(t) = \frac{\left\langle \psi_0^\dagger \boldsymbol{F}_{\mathrm{d},j} \Psi \right\rangle}{\left\langle \psi_0^\dagger \boldsymbol{F} \Psi \right\rangle}, \tag{19d}$$

$$\lambda_j^{\mathrm{eff}}(t) = \frac{\left\langle \psi_0^\dagger \lambda_j \right\rangle}{\left\langle \psi_0^\dagger \boldsymbol{F} \Psi \right\rangle}. \tag{19e}$$

The angle bracket notation indicates integration over the space, direction, and energy domains. More detailed derivations of the EPKEs can be found in [2, 6, 7]. In this work, the EPKEs are also solved by applying a backward Euler temporal discretization with the fission source approximated as behaving quadratically in time [26] (as opposed to the linear behavior in the MC approximation).

3. Monte Carlo Transient Criticality Solver and Subsolver Couplings

In this section, the individual source terms and particle banks in the transient MC solver are discussed to outline how the time-dependent source terms from the TFSP can be integrated into a static MC criticality solver to

extend its functionality to support transient criticality calculations. Then, to accelerate these transient MC calculations, the equations and procedures for coupling the various subsolver levels of the TML method (MC to CMFD and CMFD to EPKE) are discussed.

3.1. Transient Monte Carlo Solver

Hackemack et al. showed that MC implementations of the TFSP given by Eq. (5) can result in prohibitively long individual particle histories for large positive reactivity insertions [15]. This is because fixed-source algorithms typically cycle through the entire fission chain until an individual history terminates. One way to address the issue of long non-terminating fission chains is to implement particle rouletting for low-weight particles.

Alternatively, Jo et al. proposed that Eq. (5) be solved as a modified fission source iteration problem [16]. The fission source iteration algorithm only banks the subsequent generation of fission particles at each iteration, effectively guaranteeing the termination of each history. Additionally, a fission source iteration could help avoid numerical instability by allowing a greater chance for the neutron flux distribution to become balanced before advancing to the next time step. For these reasons, the same methodology is also used in this work. To accommodate these changes, a fission source iteration index

is applied to all operators applied at time t^n in Eq. (5) to arrive at

$$\left(\boldsymbol{M}+\frac{1}{v(E)\Delta t^n}\right)\psi^{l+1,n}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)=\underbrace{\frac{\psi^{n-1}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)}{v(E)\Delta t^n}}_{①}$$
$$+\frac{1}{4\pi k_{\text{eff}}^0}\Big[\underbrace{\chi_{\text{p}}^n(\boldsymbol{x},E)\boldsymbol{F}_{\text{p}}\psi^{l,n}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)}_{②}$$
$$+\sum_{j=1}^{J}\underbrace{\chi_{\text{d},j}^{n-1}(\boldsymbol{x},E)\boldsymbol{S}_{\text{d},j}\psi^{n-1}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)}_{③}f_{1,j}$$
$$+\sum_{j=1}^{J}\underbrace{\chi_{\text{d},j}^{n-1}(\boldsymbol{x},E)\boldsymbol{F}_{\text{d},j}\psi^{n-1}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)}_{④}f_{2,j}$$
$$+\sum_{j=1}^{J}\underbrace{\chi_{\text{d},j}^{n}(\boldsymbol{x},E)\boldsymbol{F}_{\text{d},j}\psi^{l,n}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)}_{⑤}f_{3,j}\Big]. \tag{20}$$

A total of $L = L_{\text{inactive}} + L_{\text{active}}$ fission source iterations are computed for each time step, where L_{inactive} and L_{active} are the number of inactive and active cycles, respectively, and set by the user. During the inactive cycles, tallies are turned off as the fission source is allowed to converge. During the active cycles, tallies are turned on, and an estimate of k_{eff} and the fission source distribution are recorded.

Traditional criticality solvers solve the steady-state equation,

$$\boldsymbol{M}\psi^{l+1,n}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)=\boldsymbol{F}\psi^{l,n}(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E)\ , \tag{21}$$

where all operators are defined for a single point in time, t^n, and the right-hand side contains only a single source term communicated between fission source iterations that corresponds to both prompt and delayed fission neutrons. The transient criticality formulation has five distinct source terms, some of which are defined at t^n, and some of which are defined at t^{n-1}. For this work, a solver of Eq. (20) was implemented on a static research branch of Shift, a MC solver developed and maintained by Oak Ridge National Laboratory (ORNL) [27].

The left-hand side of Eq. (20) is implemented by augmenting the total cross section at the beginning of a random walk by the factor $\frac{1}{v\Delta t^n}$. This augmentation accounts for the sampling of an additional "time census" pseudo-absorption event when performing the distance-to-collision calculation. When interpreted this way as an additional absorption interaction, the distance to the nearest collision is sampled with uniform random number $\xi \sim U([0,1])$ as

$$d_{\text{collision}} = -\frac{1}{\Sigma_{\text{t}} + \frac{1}{v\Delta t^n}} \ln(\xi)\,. \tag{22}$$

The implementation of the source terms on the right-hand side of Eq. (20) uses two separate particle source banks: one passed between fission source iterations (terms (2) and (5)), and one passed between time indices (terms (1), (3), and (4)). To decrease computational time, all five source terms are sampled implicitly at all collisions by pushing a copy of the incident particle to its respective source bank with an appropriate weighting factor. The individual source terms and their corresponding weight modifications are:

(1) Neutrons from time t^{n-1} that are still in flight by the time t^n. These

"time census" neutrons are implicitly captured at every collision during active cycles. This is done by passing a copy of the incident particle's state information (position, direction, energy, and weight) at every collision to the transient time-stepping bank. In order to ensure an unbiased result, the banked particle state requires a weight modification that corresponds to the probability that a time-absorption event is sampled at the collision site as

$$w_{\text{banked}} = w_{\text{incident}} \left(\frac{\frac{1}{v(E)\Delta t^n}}{\Sigma_{\text{t}}^n (\boldsymbol{x}, E) + \frac{1}{v(E)\Delta t^n}} \right) , \tag{23a}$$

where w_{incident} is the incident weight of the particle prior to experiencing a collision, and w_{banked} is the weight that is passed to the time stepping bank data structure. The incident particle will then proceed through the MC algorithm and collision mechanics will be processed with a weight adjustment factor corresponding to the compliment of Eq. (23a) as

$$w_{\text{collision}} = w_{\text{incident}} \left(\frac{\Sigma_{\text{t}}^n (\boldsymbol{x}, E)}{\Sigma_{\text{t}}^n (\boldsymbol{x}, E) + \frac{1}{v(E)\Delta t^n}} \right) . \tag{23b}$$

For very small time steps the weight of the particles exiting a time-absorption event could be very small if $\frac{1}{v\Delta t} \ll \Sigma_{\text{t}}$. Therefore, a particle rouletting procedure is performed upon exiting a collision if the weight falls below a user-specified universal threshold. In this procedure, a uniform random number between zero and one is sampled and the particle history is terminated with user-defined probability p_{roulette}. Otherwise the particle history continues with a multiplicative adjustment

of $1/(1 - p_{\text{roulette}})$.

(2) Neutrons resulting from prompt fission. This bank is already handled in steady-state k-eigenvalue calculations and therefore requires no modification to the criticality solver. If a prompt neutron is sampled, its state information is banked at the fission site to be carried over the next fission source iteration and used to source a new particle history. Because these neutrons are emitted on very small time scales, their state information is not passed to subsequent time steps.

(3) This is a special term that accounts for the accumulation of delayed neutron precursors when integrating over previous time steps. These accumulated delayed neutron precursors will emit delayed neutrons at times that might exceed $t^n + \Delta t^n$. In the Shift implementation, copies of all of the particle state information in the time step iteration bank at time index $n - 1$ are injected into the time step iteration bank at n with weight of

$$w_{t^n} = w_{t^{n-1}} f_{1,j} \,. \tag{23c}$$

(4) Neutrons resulting from delayed neutron precursors that were accumulated over Δt^n and numerically weighted as being sourced from the previous time index, $n - 1$, with a weight modification of

$$w_{\text{banked}} = w_{\text{incident}} f_{2,j} \,. \tag{23d}$$

(5) Neutrons resulting from delayed neutron precursors that were accumulated over Δt^n and numerically weighted as being sourced from the

current time index, n, with a weight modification of

$$w_{\text{banked}} = w_{\text{incident}} f_{3,j} \,. \tag{23e}$$

The particle state information is then copied and pushed to the time-stepping source bank.

A simplified branching process that neglects energy-dependence is shown in Fig. 2 to depict the implicit banking of source terms at each collision site.

Figure 3 depicts the transfer of source terms between fission source iterations and time steps. Fission source terms (2 and 5) are passed between fission source iterations. During the active fission cycles, time step source terms (1 and 4) are tallied and read in as source terms at the start of the next time step. Finally, the accumulated delayed neutron term (3) is read from the previous time step iteration bank at $n-1$.

The recursive process of transferring delayed neutron source terms from $n-1$ causes the time step source bank to accumulate terms exponentially. To prevent this exponential growth, a population control technique known as "combing" [28] is performed in between time steps that keeps the size of the bank fixed. Say, for example, that the size of the transient will be fixed to contain the state information of N_{trans} source particles. Before combing, the weight of all of the particles in the transient source bank will be added together. Next, the order of all of the particles in the source bank will then be randomly shuffled and only the state information for the first N_{trans} will be retained (the remaining particle state information will be discarded). The

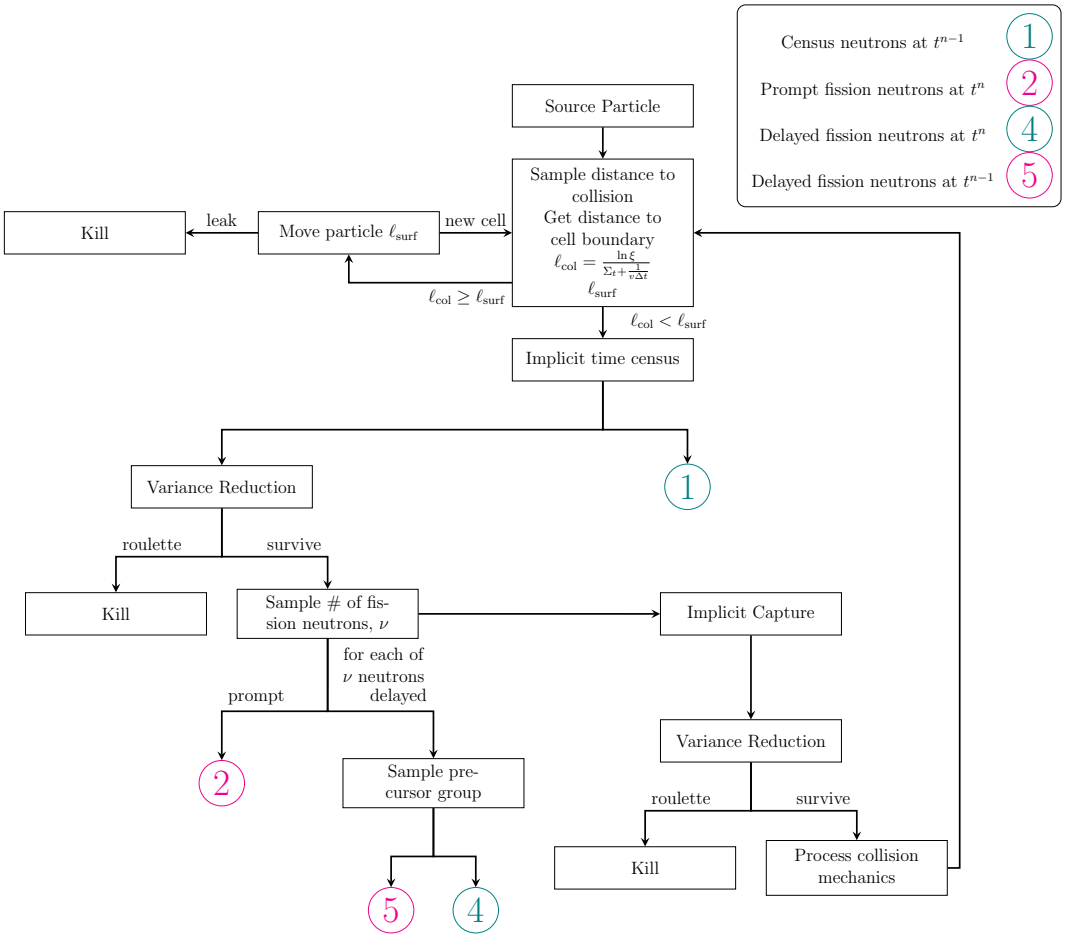

Figure 2: Simplified neutron history branching in transient fission source iteration scheme.

weight of all of the remaining N_{trans} source particles will be added together so that the weight of the remaining particles can be updated as

$$w_{j,\text{combed}} = w_j \frac{\sum_j^{N_{\text{trans}}} w_j}{\sum_i w_i}, \tag{24}$$

where i corresponds to the index of the source particle before shuffling and j corresponds to the index of the same source particle after shuffling.

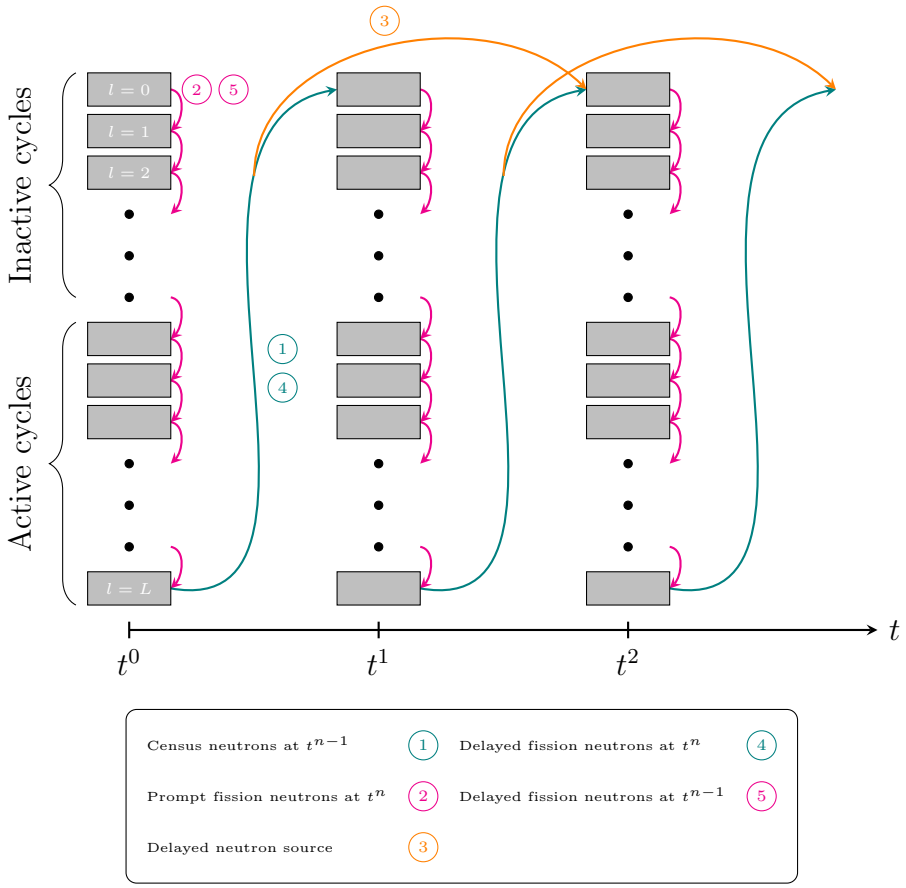

Figure 3: Transfer of source terms between fission source iterations and time steps.

3.2. Monte Carlo Transport and CMFD Coupling

The PCQM formulation between the MC and CMFD solvers factorizes the angular flux as

$$\psi_{\mathrm{MC}}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t\right) = p_{\mathrm{MC}}\left(\boldsymbol{x}, E, t\right) \Psi_{\mathrm{MC}}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t\right) \ , \tag{25}$$

where p_{MC} is the transport-level amplitude function and Ψ_{MC} is the transport-level shape function. The shape function is computed by

$$\Psi_{\text{MC}}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t\right) = \frac{\psi_{\text{MC}}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t\right)}{p_{\text{MC}}\left(\boldsymbol{x}, E, t\right)} \approx \frac{\psi_{\text{MC}}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t\right)}{\phi_{\text{CMFD}}\left(\boldsymbol{x}, E, t\right)} \,, \tag{26}$$

where ϕ_{CMFD} is the CMFD scalar flux and is the solution to Eq. (15). The time-dependent CMFD solution can be obtained with any standard linear algebra solver after setting up the system of equations as a matrix described by Zhu's MGM method (defined in Section 2.3) [11]. The CMFD solver in Shift uses the Generalized Minimal Residual (GMRES) method to solve the matrix formulation of Eq. (15) where the solution vector consists of the cell-centered CMFD scalar fluxes for each energy group. Uniqueness with the factorization given by Eq. (25) is ensured with a normalization constraint for each CMFD cell:

$$\frac{1}{V_{\text{CMFD}}} \left\langle \psi_{\text{MC}}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E, t\right) \right\rangle = 1 \,. \tag{27}$$

The MC and CMFD coupling algorithm is explicitly defined in Algorithm 1. To begin, the forward and adjoint steady state fluxes are evaluated at $t = 0$. In this work, the adjoint solution is generated using an SP_3 k-eigenvalue solver in the Denovo solver [29].

The algorithm proceeds by advancing the MC solution on the coarse time step Δt_{MC} by solving Eq. (20). This flux solution is denoted as the "predictor" flux with superscript p. The CMFD parameters that appear in Eq. (15) at the beginning and end of the Δt_{MC} time step are calculated from the MC tallies. While quantities such as multigroup macroscopic cross sections and scalar fluxes are straightforward to compute using path-length

or collision tallies, special care must be taken when computing the diffusion coefficients, $\tilde{D}$ and $\hat{D}$.

The $\tilde{D}$ diffusion coefficient is defined by Eq. (16b) or Eq. (16d) with the "standard" diffusion coefficients defined as

$$D^n_{k,l,m,g} \approx \frac{1}{3(\Sigma^n_{\mathrm{t},k,l,m,g} - \mu_0 \sum_{g'}^{G} \Sigma^n_{\mathrm{s0},k,l,m,g' \to g})} \,. \tag{28}$$

This approximation assumes that the in-scatter rate of neutrons from E' to E is approximately equal to the out-scatter rate of neutrons from E to all other energies E', i.e., the medium is weakly-absorbing. Moreover, this approximation requires group-to-group scattering matrices for every coarse mesh volume for every time step.

In the Shift implementation, these matrices are computed using analog tallies that are scored every time a neutron transitions from group g' to group g. These analog tallies can be subject to poor statistics in parts of the problem with highly-absorbing media or low neutron populations and result in ill-conditioned matrix systems when performing the CMFD solve. These limitations make the MC implementation of the TML particularly challenging when compared to deterministic implementations. More accurate diffusion coefficients can be calculated with migration area tallies [30] but have not been explored in this work. Once the $\tilde{D}$ are known, the $\hat{D}$ diffusion coefficients can computed by Eq. (16a) or Eq. (16c).

After all CMFD parameters have been computed at the beginning and end of the Δt_{MC} time step, they are interpolated on the Δt_{CMFD} time scale so that the CMFD scalar flux amplitude function (and corresponding EPKE

substeps) can be propagated forward in time. Finally, the CMFD shape function is used to correct the MC tallies at t^n and precursor concentrations are computed. The superscript c is used to denote the corrected flux solution.

Algorithm 1 PCQM coupling between MC and CMFD levels in the TML.

1: Solve Eq. (21) for forward steady-state forward flux ψ^0_{MC}
2: Solve for steady-state adjoint flux $\psi^{\dagger,0}$;
3: **for all** $1 \leq n \leq N_{\text{MC}}$ **do**
4: $\quad \psi^{\text{p},n}_{\text{MC}} \leftarrow$ Eq. (20)
5: $\quad$ Linearly interpolate CMFD volume- and surface-integrated MC tallies between ψ^{n-1} and ψ^n
6: $\quad$ Propagate through Algorithm 2 to compute $\phi^{\text{p},n}_{\text{CMFD}}$ and $\phi^{\text{c},n}_{\text{CMFD}}$
7: $\quad \psi^{\text{c},n}_{\text{MC}} \leftarrow \psi^{\text{p},n}_{\text{MC}} \frac{\phi^{\text{c},n}_{\text{CMFD}}}{\phi^{\text{p},n}_{\text{CMFD}}}$
8: $\quad$ Update precursor concentrations using corrected MC flux distribution.
9: **end for**

3.3. CMFD and EPKE Coupling

The PCQM formulation between the CMFD and EPKE solvers factorizes the scalar flux as

$$\phi_{\text{CMFD}}\left(\boldsymbol{x}, E, t\right) = p_{\text{CMFD}}\left(t\right) \Psi_{\text{CMFD}}\left(\boldsymbol{x}, E, t\right) \,, \tag{29}$$

where $p_{\text{CMFD}}\left(t\right) = p_{\text{EPKE}}\left(t\right)$. The normalization constraint used to ensure uniqueness after the factorization in Eq. (29) is defined as

$$\begin{aligned} c &= \left\langle \psi^{\dagger,0}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E\right), \frac{1}{v\left(E\right)} \Psi^0_{\text{CMFD}}\left(\boldsymbol{x}, E\right) \right\rangle \\ &= \left\langle \psi^{\dagger,0}\left(\boldsymbol{x}, \hat{\boldsymbol{\Omega}}, E\right), \frac{1}{v\left(E\right)} \Psi_{\text{CMFD}}\left(\boldsymbol{x}, E, t\right) \right\rangle \,. \end{aligned} \tag{30}$$

The amplitude and shape functions are then computed as

$$\phi_{\mathrm{CMFD}}^{\mathrm{c},n}\left(\boldsymbol{x},E\right)=\frac{c}{\left\langle \psi^{\dagger,0}\left(\boldsymbol{x},\hat{\boldsymbol{\Omega}},E\right),\frac{1}{v(E)}\phi_{\mathrm{CMFD}}^{\mathrm{p},n}\left(\boldsymbol{x},E\right)\right\rangle }p_{\mathrm{EPKE},n}\phi_{\mathrm{CMFD}}^{\mathrm{p},n}\left(\boldsymbol{x},E\right)\,. \tag{31}$$

The PCQM coupling between the CMFD/EPKE levels is described by Algorithm 2. To begin, the multigroup cross sections needed for the CMFD solver are post-processed from the direct MC tallies and interpolated on the CMFD time scale. Next, the normalization constant c is computed with Eq. (30) to ensure that the shape/amplitude factorization is unique. Then, the CMFD predictor solution is propagated forward in time. The EPKE parameters are then interpolated on the Δt_{EPKE} time scale and the EPKE amplitude function is propagated forward in time to the end of the CMFD time step. Finally, the CMFD predictor flux is corrected using Eq. (31) and the precursor concentrations are computed.

Algorithm 2 PCQM coupling between CMFD and EPKE levels in the TML.

1: Linearly interpolate multigroup cross sections needed for CMFD solver on Δt_{CMFD} time step
2: $c \leftarrow$ Eq. (30)
3: **for all** $1 \leq n \leq \Delta t_{\mathrm{MC}}/\Delta t_{\mathrm{CMFD}}$ **do**
4: $\quad \phi_{\mathrm{CMFD}}^{\mathrm{p},n} \leftarrow$ Eq. (15)
5: $\quad$ Linearly interpolate globally-integrated EPKE parameters at Δt_{EPKE} resolution
6: $\quad p_{\mathrm{EPKE}}^{n} \leftarrow$ Eq. (18)
7: $\quad \phi_{\mathrm{CMFD}}^{\mathrm{c},n} \leftarrow$ Eq. (31)
8: $\quad$ Update precursor concentrations using corrected CMFD flux distribution.
9: **end for**

Figure 4 shows the flow of information between the various TML levels.

At the beginning and end of each MC time step, multigroup cross sections are passed down to the CMFD solver to be interpolated and propagated forward through the CMFD time steps. At end of the CMFD propagations, a space- and energy-dependent amplitude function p_{CMFD} is passed back to the MC solver to update the weights of the particles in the time-stepping source bank as a function of their current position and energy

$$w^{\text{c},n-1} = w^{\text{p},n-1} \frac{\phi^{\text{c},n-1}_{\text{CMFD}}\left(\boldsymbol{x}, E\right)}{\left\langle \phi^{\text{c},n-1}_{\text{CMFD}}\left(\boldsymbol{x}, E\right)\right\rangle} . \tag{32}$$

Between the CMFD and EPKE levels, point kinetics parameters are passed from the CMFD calculation. At the end of each EPKE propagation, a globally-integrated amplitude function is passed back up to the CMFD solver to update the scalar flux prediction.

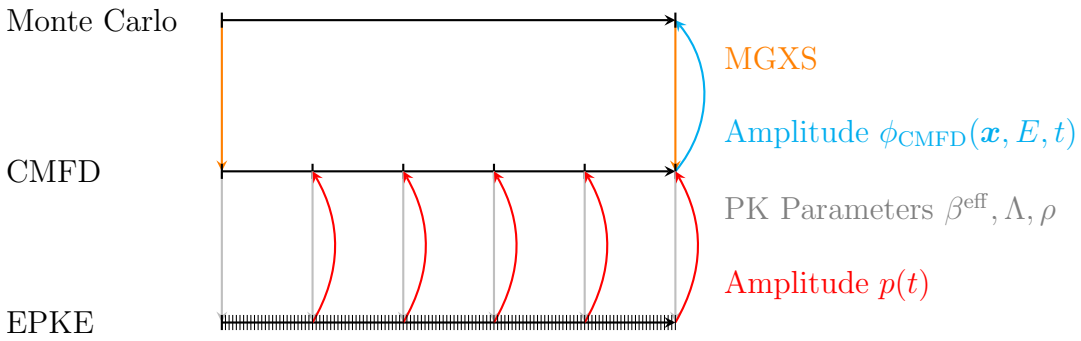

Figure 4: Flow of information for coupling between the MC, CMFD, and EPKE levels.

3.4. Implementation Into Shift

Several changes to Shift's source code were implemented on a frozen research branch to enable the TML algorithm. The burden of implementing

these changes was lessened by leveraging several of Shift's existing features. For example, introducing time-stepping capabilities into Shift was accomplished by repurposing the existing depletion infrastructure to set up a series of MC k-eigenvalue solves. By repurposing Shift's surface census tally capabilities, census particle state information could be written to HDF5 files at the end of a MC solve so that it could be read in at the start of the subsequent MC solve. Passing information between Denovo's deterministic adjoint SP_3 and CMFD solvers was enabled by repurposing Shift's hybrid CADIS and FW-CADIS capabilities [31]. Point kinetics tallies were enabled by repurposing Shift's nodal tally capabilities and summing over the spatial and energy domains. Additional infrastructure was developed for specifying time-dependent multipliers to material densities to initiate transient events and performing particle combs via shuffling for population control [28].

4. C5G7 TD3 Benchmark Results

The TML implementation in Shift was tested using the C5G7 deterministic transient benchmark [32]. The C5G7 reactor benchmark models a small Pressurized Water Reactor (PWR) core with quarter-core radial symmetry. There are sixteen fuel assemblies: eight Uranium Dioxide (UO2) assemblies, and eight Mixed Oxide (MOX) assemblies (with varying levels of enrichment), surrounded by a water reflector. Each assembly consists of a 17×17 array with 264 fuel pins, 24 guide tubes for control rods to be placed, and one instrument tube in the center of the grid for a fission chamber. A diagram of the southeast quadrant homogenized at the pincell level can be seen in Fig. 5.

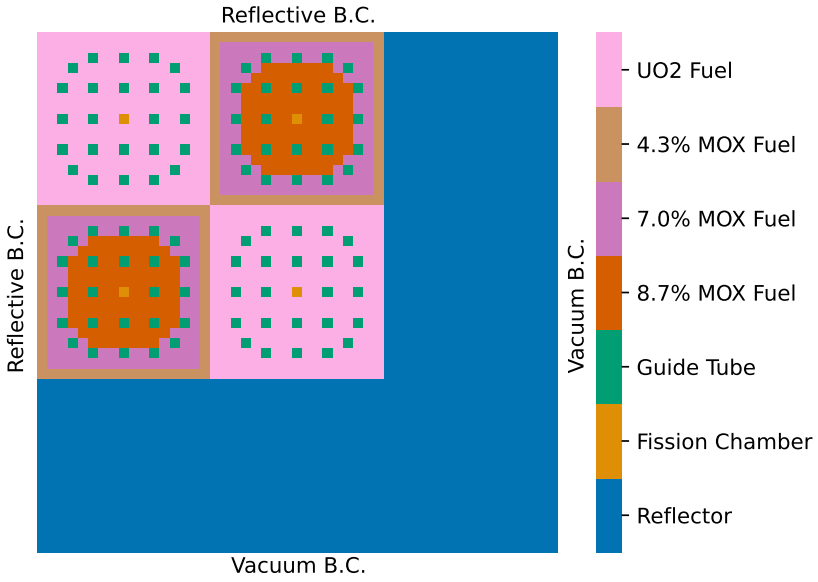

Figure 5: Southeast quadrant of the C5G7 reactor core configuration homogenized at the pincell level.

Each fuel cell has a pitch of 1.26 cm and has been homogenized into two regions: a fuel zone homogenized over the ceramic fuel, gap, and cladding with radius of 0.54 cm, and a surrounding moderator zone. Transport-corrected macroscopic cross sections and scattering matrices are given in the benchmark specifications [32].

While several prescribed transient exercises are specified in the C5G7 benchmark, this work presents the results of simulating the third prescribed exercise, "TD3". Exercise TD3 simulates four transient events by linearly varying the core moderator density for a 2D version of the reactor geometry with the time-dependent profile seen in Fig. 6, where ω is the minimum moderator density used throughout the transient at $t = 1\,\mathrm{s}$.

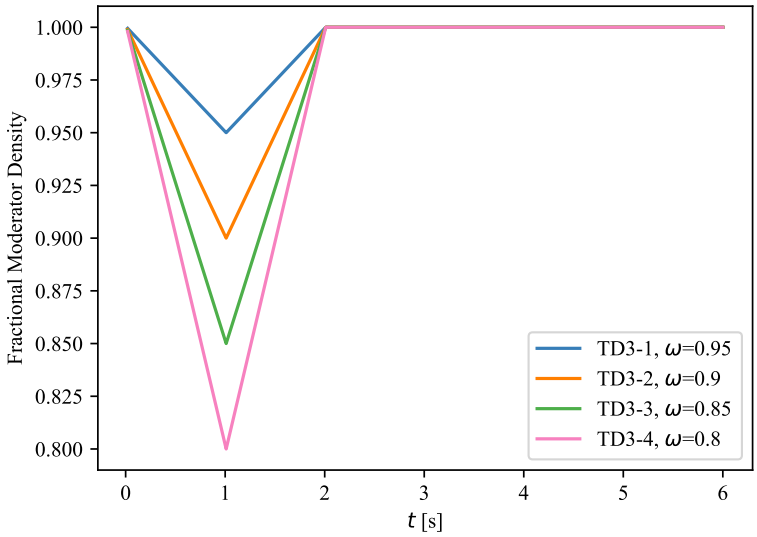

Figure 6: Time-dependent changes in the average moderator density specified for the C5G7-TD3 exercises.

4.1. PCQM Results

Before implementing the full TML in Shift, a PCQM implementation (without the CMFD intermediate level) was first implemented to test the transient fission source iteration and provide a benchmark to assess performance improvements upon introducing the CMFD level. The PCQM results are shown in Fig. 7 for a macro time step of $\Delta t_{\mathrm{MC}} = 0.1\,\mathrm{s}$, which is the same transport-level time step used for a deterministic PCQM simulation of the same transient simulated by DeHart et al. [33]. Each MC step was simulated with 10^5 neutron histories per cycle for 150 inactive cycles and 250 active cycles. The micro time step was selected to be $\Delta t_{\mathrm{EPKE}} = 1\,\mathrm{ms}$.

The results show that stochastic noise in the dynamic reactivity after $t > 2\,\mathrm{s}$ is amplified by the exponential transform of the solution that the

EPKE solver assumes. This same phenomenon was also observed by Guo et al. for a MC-PCQM implementation in the RMC code [34] and Shaner for a frequency transform method [20] that also assumes exponential behavior of the amplitude function. Shaner showed that the stochastic noise for reactivities near zero is resolved as the number of particle histories is increased, while Guo et al. use a polynomial fitting technique of the dynamic reactivity for noise reduction.

Figure 8 presents the same simulations with stochastic noise in the dynamic reactivity artificially removed for $t > 2\,\mathrm{s}$ by forcing the dynamic reactivity in the simulation to be zero after $t = 2\,\mathrm{s}$ instead of being derived from MC tallies (i.e. $\rho\,(t > 2\,\mathrm{s}) \equiv 0$). To accomplish this, the dynamic reactivity computed by the CMFD solver is hardcoded to be zero for $t \geq 2\,\mathrm{s}$ for the C5G7-TD3 transient. Removing noise in the dynamic reactivity also results in a smoothing of the fractional core fission rate profile. From these results it is also apparent that initial bias immediately after $t = 2\,\mathrm{s}$ is propagated throughout the transient, resulting in positive feedback loop as time progresses. In future work, adaptive schemes for devoting more computational resources to resolving statistical uncertainties for dynamic reactivities near zero should be considered.

4.2. TML Results

Figure 9 shows the results of the TD3 simulations using the TML implementation, now with the CMFD level. Each TD3 case simulated was run to $t = 6\,\mathrm{s}$ with 10^5 particle histories per cycle, 150 inactive cycles, and 250 active cycles. The CMFD mesh uses a pincell resolution (51×51 mesh elements) over the active region of the reactor. Partial current tallies are used

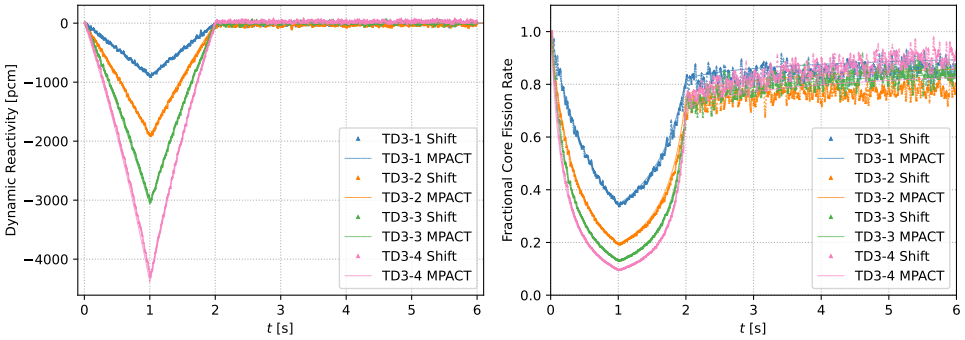

Figure 7: Dynamic reactivity and relative fission rate of the PCQM implementation.

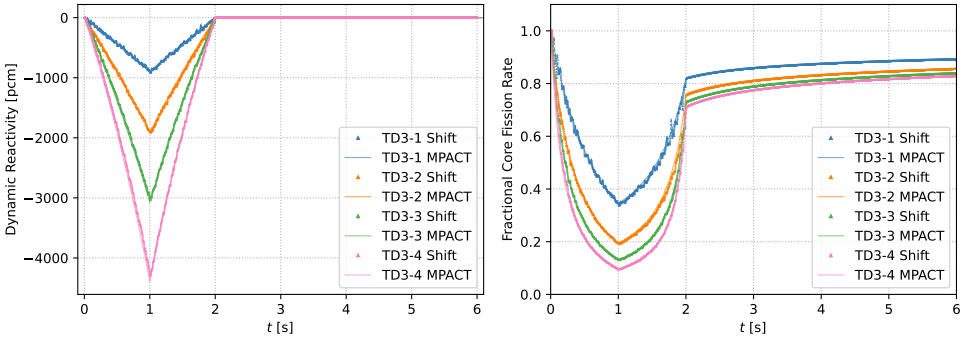

Figure 8: Dynamic reactivity and relative fission rate of the PCQM implementation with stochastic noise removed for $t > 2\,\mathrm{s}$.

to define the CMFD boundary sources, but the CMFD calculation did not directly include the reflector regions. All of the reaction rate, surface current and point kinetics tallies were accumulated on this mesh during the MC active cycles to be used as input for the CMFD solver. The time steps at the various TML levels are selected to be $\Delta t_{\mathrm{MC}} = 0.25\,\mathrm{s}$, $\Delta t_{\mathrm{CMFD}} = 0.05\,\mathrm{s}$, and $\Delta t_{\mathrm{EPKE}} = 0.001\,\mathrm{s}$.

The TML method noticeably reduces the stochastic noise in both the dynamic reactivity and corresponding fractional core fission rate profiles. This is because (1) there are fewer MC steps being computed throughout the

transient and (2) MC tallies (and their accompanying stochastic noise) are not directly driving the EPKE solution. Instead, point kinetics parameters are derived from the CMFD solutions that use linearly-interpolated MC-generated cross sections to provide spatial and energy resolution updates to the shape function throughout the transient. The importance of these intermediate updates is illustrated by the slight curvature in the dynamic reactivity (left) for both the descending ($0 \leq t < 1\,\mathrm{s}$) and ascending ($1 \leq t < 2\,\mathrm{s}$) portions of the curve. Therefore, linearly interpolating EPKE parameters between MC solutions at $t = 0$, $t = 1$, and $t = 2$, is not sufficient to capture this feature. Error bars are not presented in Fig. 9 as the propagation of error in the MC tallies through the CMFD and EPKE subsolvers is not straightforward. Instead, batch statistics are used to assess the resolution of stochastic noise later in Fig. 10.

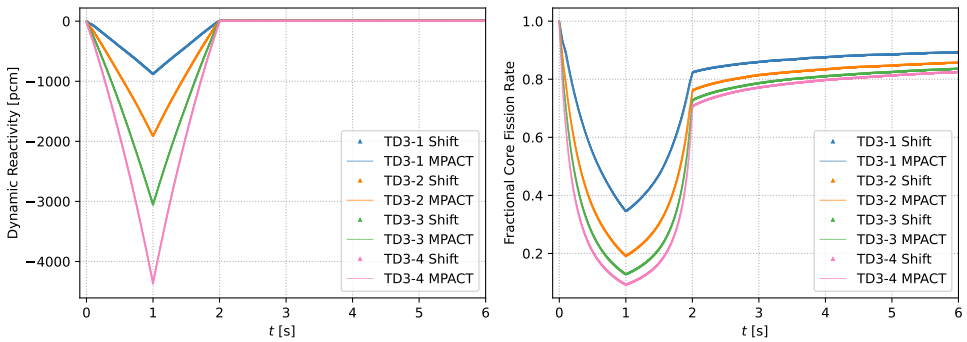

Figure 9: Dynamic reactivity and relative fission rate of the TML implementation with stochastic noise removed for $t > 2\,\mathrm{s}$.

Several norm comparisons are given in Table 1 to assess TML performance improvements over the PCQM implementation. The MPACT solutions are used as the reference solution. The L_1 and L_∞ norm evaluations for the TML decrease by roughly an order of magnitude for every TD3 case when

compared to the PCQM. The L_2 norm improves by a factor of 2-4 for all TD3 cases.

Table 1: Norm comparisons (given in %) to MPACT solution for both PCQM and TML implementations.

Norm		TD3-1	TD3-2	TD3-3	TD3-4
$L_\infty = \max_n \left[\frac{p_{n,\text{Shift}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}}\right]$	PCQM	8.42	13.48	10.12	12.38
	TML	2.40	3.65	2.83	2.23
$L_2 = \frac{1}{N}\left[\sum_n \left(\frac{p_{n,\text{Shift}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}}\right)^2\right]^{\frac{1}{2}}$	PCQM	0.28	0.47	0.28	0.46
	TML	0.09	0.12	0.13	0.13
$L_1 = \frac{1}{N}\sum_n \left\lvert\frac{p_{n,\text{Shift}} - p_{n,\text{MPACT}}}{p_{n,\text{MPACT}}}\right\rvert$	PCQM	2.17	3.70	2.18	3.42
	TML	0.62	0.91	1.07	1.06

Ideally, the use of a larger MC time step would bring down the total simulation time as the MC solve is much costlier than both the CMFD and EPKE solves. To show this, a rough timing comparison between the various implementations is shown in Table (2). All of the Shift calculations were executed on an ORNL compute cluster featuring dual-processor AMD Opteron 6378 CPUs with 16 cores per CPU, for a total of 32 cores per node, and 128 GB RAM per node. All Shift simulations of the C5G7 reactor geometry were run on eight nodes (256 cores). The MPACT simulations were carried out for the same transient specifications up until $t = 10\,\text{s}$. Additionally, the MPACT benchmark only reported wall time using nine processors. The wall times reported by MPACT were then multiplied by a factor of nine to give an estimate for the number of core hours used. While these numbers are approximate and not directly comparable, it can be seen that the Shift PCQM and TML implementations are roughly one and two orders of magnitude slower, respectively, than the MPACT implementation.

Table 2: CPU time comparison (hours) between the TD3 solution methods.

TD3 Case	Shift-PCQM (6 s)	Shift-TML (6 s)	MPACT (10 s)
1	2510.1	395.0	30.6
2	3067.1	395.1	29.7
3	2533.1	398.0	29.7
4	2039.8	351.1	29.7

As stated previously, Shaner showed that the stochastic noise in the MC frequency transform method was resolved with an increasing number of particle histories [20]. For the MC TML implementation, small perturbations in the dynamic reactivity are amplified by the EPKE solver. A batch analysis was performed to assess whether this is also the case with the MC TML implementation. Three values for the number of histories to simulate per fission source iteration cycle: $N_{\text{hpc}} = 10^4$, 10^5, and 10^6. The C5G7-TD3-4 transient was modeled based on the magnitude of its change in reactivity. Other than varying N_{hpc}, all Shift inputs used to generate each individual simulation of the batch analysis were identical to those used to generate the results presented in Fig. 9. All values of N_{hpc} were simulated with ten batches, where each batch began with a unique random seed. The average over the two batches for each point in time are shown as the solid lines in Fig. 10. The shaded regions show the standard deviation at each point in time. It is clearly shown that the noise is resolved with increasing particle histories. The quality of the solution is fairly consistent while the reactivity is changing, i.e. $t < 2\,\text{s}$. The solution is much more sensitive to stochastic noise for dynamic reactivities near zero. In future implementations, more computational resources (i.e., larger values of N_{hpc}) should be devoted to

temporal regions where the imposed reactivity should be zero.

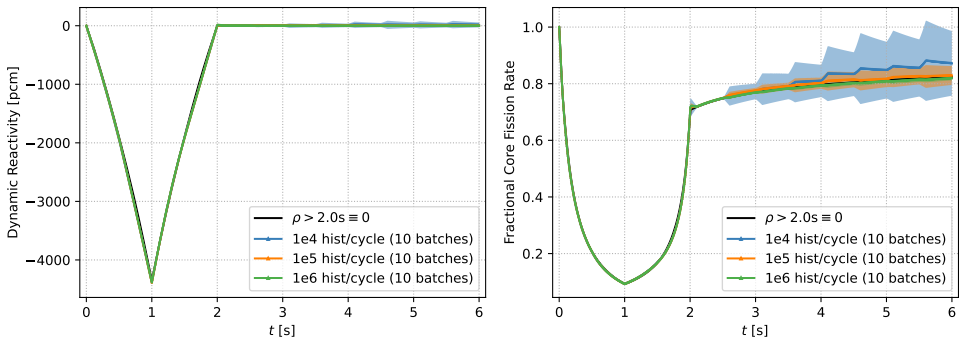

Figure 10: Uncertainties from stochastic noise associated with dynamic reactivity and relative fission rate for TD3-4.

A refinement of the MC and CMFD time scales was also analyzed to see how the ratio $\Delta t_{\mathrm{MC}}/\Delta t_{\mathrm{CMFD}}$ affects the accuracy of the solution. The L_2 norm comparison for each of the simulations for various values of Δt_{MC} is shown in Fig. 11 (left), where the reference solution used time steps of $\Delta t_{\mathrm{MC}} = 0.1\,\mathrm{s}$, $\Delta t_{\mathrm{CMFD}} = 2\,\mathrm{ms}$, and $\Delta t_{\mathrm{EPKE}} = 80\,\mu\mathrm{s}$. Accuracy gains appear to level off for values of $\Delta t_{\mathrm{MC}}/\Delta t_{\mathrm{CMFD}} > 10$. The right plot in Fig. 11 shows the percentage of the total simulation time spent computing the MC time propagations. The percentage of time spent on the MC solves appear to decrease linearly with an increasing number of CMFD steps per MC step. Since the most significant gains in solution accuracy occur for $\Delta t_{\mathrm{MC}}/\Delta t_{\mathrm{CMFD}} < 10$, the MC solver takes upwards of 95% of the total simulation time in the TML implementation. Therefore, acceleration efforts should ideally focus on reducing the number of MC steps required to solve a transient while retaining the accuracy of the solution.

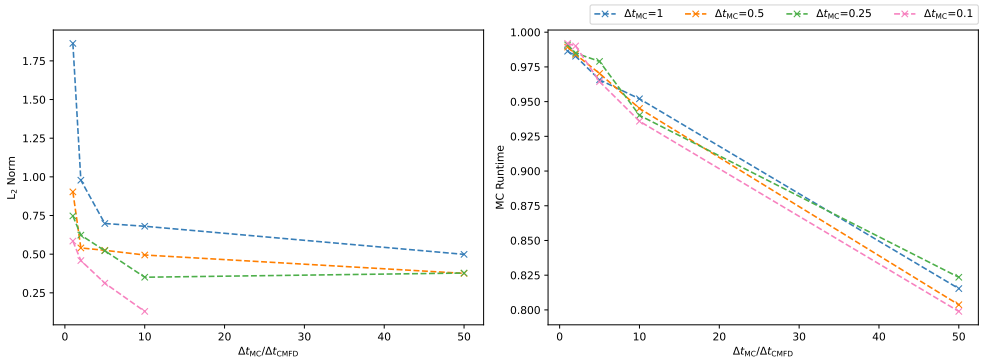

Figure 11: Time step convergence analysis for TD3-4 exercise.

5. Conclusions

In this paper, the TML method was implemented using a modified MC fission source iteration solution at the transport level. The MC TML implementation showed marked reductions in both stochastic noise and overall simulation time when compared against the standard MC PCQM technique for the C5G7-TD3 transient benchmark problem. It was shown that stochastic noise where the dynamic reactivity is near zero is amplified in the transient power profiles but that this stochastic noise is resolved as the number of histories simulated approaches infinity. Similar stochastic noise resolution was observed with Shaner's frequency transform method, indicating that the observation is likely generalizable to other transients using the TML.

It was also shown that, for the C5G7-TD3 problem, performance gains diminish significantly after the number of CMFD per MC steps of around ten, where $\geq 95\%$ of the total simulation time is spent on the MC calculations. Zhu's initial deterministic implementation of the TML found that the MOC transport solver accounted for roughly 50% of the overall simulation time and

the CMFD solver accounted for 31% of the overall simulation time. It should be noted though that Zhu's calculations also included thermal hydraulics solvers that accounted for 17% of the overall simulation time [11]. Though it was expected that the MC solver would dominate the overall simulation time, it draws attention to the fact that further acceleration techniques should be oriented towards limiting the number of MC steps needed to simulate the transient without losing accuracy. While the MC TML shows speedup when compared to the PCQM, the method is still roughly an order of magnitude slower than the MOC TML for the C5G7-TD3 transient.

Areas of future work include investigating models that further stress the MC TML implementation. In particular, models that stress the limitation of the implementation's dependence on group-to-group scattering matrices through analog tallies. Problems with regions of low statistics for these matrices may lead to ill-conditioned CMFD systems and instability of the algorithm. It may be necessary to implement global variance reduction techniques to ensure better MC statistics [35, 36, 37]. Additionally, the TML implementation in Shift only supports multigroup physics. Monte Carlo solvers using continuous-energy physics will lead to even more computationally expensive transients. Acceleration schemes are therefore necessary to make the algorithm more tractable for practical applications. Other areas of future work include investigating alternative formulations of the CMFD equations (such as pCMFD, odCMFD, and lpCMFD [38, 39]) to be used in the TML.

Acknowledgments

This work was funded by the Spent Fuel and Waste Disposition program from the Department of Energy – Office of Nuclear Energy (DOE-NE). This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] A. F. Henry, The application of reactor kinetics to the analysis of experiments, Nuclear Science and Engineering 3 (1) (1958) 52–70. arXiv:https://doi.org/10.13182/NSE58-1, doi:10.13182/NSE58-1.
URL https://doi.org/10.13182/NSE58-1

[2] A. F. Henry, N. J. Curlee, Verification of a method for treating neutron space-time problems, Nuclear Science and Engineering 4 (6) (1958) 727–744. arXiv:https://doi.org/10.13182/NSE4-727, doi:10.13182/NSE4-727.
URL https://doi.org/10.13182/NSE4-727

[3] J. B. Yasinsky, A. F. Henry, Some numerical experiments concerning space-time reactor kinetics behavior, Nuclear Science and Engineering 22 (2) (1965) 171–181. arXiv:https://doi.org/10.13182/NSE65-A20236, doi:10.13182/NSE65-A20236.
URL https://doi.org/10.13182/NSE65-A20236

[4] K. O. Ott, D. A. Meneley, Accuracy of the quasistatic treatment of

spatial reactor kinetics, Nuclear Science and Engineering 36 (3) (1969) 402–411.

[5] J. C. Gehin, A quasi-static polynomial nodal method for nuclear reactor analysis, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA (1992).

[6] S. Dulla, E. H. Mund, P. Ravetto, Accuracy of a predictor-corrector quasi-static method for space-time reactor dynamics, in: International Conference on the Physics of Reactors, PHYSOR, 2006.

[7] S. Dulla, E. Mund, P. Ravetto, The quasi-static method revisited, Progress in Nuclear Energy 50 (2008) 908–920. doi:10.1016/j.pnucene.2008.04.009.

[8] C. Patricot, A.-M. Baudron, O. Fandeur, The multi-physics improved quasi-static method - application to a neutronics-thermomechanics coupling, Progress in Nuclear Energy 101 (2017) 352–359, special Issue on the Physics of Reactors International Conference PHYSOR 2016: Unifying Theory and Experiments in the 21st Century. doi:https://doi.org/10.1016/j.pnucene.2017.03.021.
URL https://www.sciencedirect.com/science/article/pii/S0149197017300665

[9] Z. M. Prince, J. C. Ragusa, Multiphysics reactor-core simulations using the improved quasi-static method, Annals of Nuclear Energy 125 (2019) 186–200. doi:https://doi.org/10.1016/j.anucene.2018.10.056.

URL https://www.sciencedirect.com/science/article/pii/S030645491830584X

[10] G. Kooreman, D. P. Griesheimer, Predictor-corrector quasi-static method for tightly-coupled reactor multiphysics transient calculations, EPJ Web Conf. 247 (2021) 07011. doi:10.1051/epjconf/202124707011.
URL https://doi.org/10.1051/epjconf/202124707011

[11] Y. X. Ang Zhu, T. Downar, A multilevel quasi-static kinetics method for pin-resolved transport transient reactor analysis, Nuclear Science and Engineering 182 (4) (2016) 435–451. arXiv:https://doi.org/10.13182/NSE15-39, doi:10.13182/NSE15-39.
URL https://doi.org/10.13182/NSE15-39

[12] Q. Shen, B. Kochunas, Y. Xu, S. Choi, T. Downar, Transient multilevel scheme with one-group CMFD acceleration, Nuclear Science and Engineering 195 (7) (2021) 741–765. arXiv:https://doi.org/10.1080/00295639.2020.1866388, doi:10.1080/00295639.2020.1866388.
URL https://doi.org/10.1080/00295639.2020.1866388

[13] B. L. Sjenitzer, J. E. Hoogenboom, Dynamic monte carlo for nuclear reactor kinetics calculations, Nuclear Science and Engineering 175 (2013) 94–107.

[14] B. L. Sjenitzer, J. E. Hoogenboom, J. J. Escalante, V. S. Espinoza, Coupling of dynamic monte carlo with thermal-hydraulic feedback, Ann. of Nucl. Energy 76 (2015) 27–39.

[15] M. W. Hackemack, J. C. Ragusa, D. P. Griesheimer, J. M. Pounders, A Monte Carlo implementation of the predictor-corrector quasi-static method., in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), American Nuclear Society, 2013.

[16] Y. Jo, B. Cho, N. Z. Cho, Nuclear reactor transient analysis by continuous-energy Monte Carlo calculation based on predictor-corrector quasi-static method, Nuclear Science and Engineering 183 (2016) 229–246.

[17] M. J. Lee, H. Joo, D. Lee, K. Smith, Investigation of CMFD accelerated Monte Carlo eigenvalue calculation with simplified low dimensional multigroup formulation, in: International Conference on the Physics of Reactors, PHYSOR, 2010.

[18] M. J. Lee, H. G. Joo, D. Lee, K. Smith, Multiset CMFD acceleration of source convergence for three-dimensional Monte Carlo reactor calculations, in: Transactions of the American Nuclear Society, Vol. 107, 2012.

[19] Y. Ban, T. Endo, A. Yamamoto, A unified approach for numerical calculation of space-dependent kinetic equation, Journal of Nuclear Science and Technology 49 (5) (2012) 496–515. arXiv:https://doi.org/10.1080/00223131.2012.677126, doi:10.1080/00223131.2012.677126. URL https://doi.org/10.1080/00223131.2012.677126

[20] S. C. Shaner, Development of high fidelity methods for 3D Monte Carlo

transient analysis of nuclear reactors, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA (2018).

[21] B. F. Miriam A. Kreher, Samuel Shaner, K. Smith, Frequency transform method for transient analysis of nuclear reactors in monte carlo, Nuclear Science and Engineering 197 (2) (2023) 279–290. arXiv:https://doi.org/10.1080/00295639.2022.2067739, doi:10.1080/00295639.2022.2067739.
URL https://doi.org/10.1080/00295639.2022.2067739

[22] D. He, Y. Li, Q. Pan, X. Liu, Development of a predictor-corrector quasi-static transient fission matrix combination method for the neutronics transient simulation, Nuclear Engineering and Design 424 (2024) 113301. doi:https://doi.org/10.1016/j.nucengdes.2024.113301.
URL https://www.sciencedirect.com/science/article/pii/S0029549324004011

[23] V. Mascolino, A. Haghighat, A novel hybrid deterministic and monte carlo neutron transport formulation and algorithm (trapid) for accurate and fast 3-d reactor kinetics, Nuclear Science and Engineering 198 (3) (2024) 592–627. doi:10.1080/00295639.2023.2197844.

[24] E. S. Gonzalez, Multi-level methods for monte carlo reactor kinetics, Ph.D. thesis, University of Michigan, Ann Arbor, MI (2023).

[25] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget, OpenMC: A state-of-the-art Monte Carlo code for research and development, Ann. of Nucl. Energy 82 (2015) 90–97.

[26] T. Downar, Y. Xu, V. Seker, PARCS v3.0 User Manual, Tech. Rep. 20, U.S. NRC (2010).

[27] T. M. Pandya, S. R. Johnson, G. G. Davidson, T. M. Evans, S. P. Hamilton, Shift: A massively parallel Monte Carlo radiation transport package, in: Joint International Conference on Mathematics and Computation, Supercomputing in Nuclear Applications and the Monte Carlo Method, American Nuclear Society, Nashville, TN, 2015.
URL https://www.osti.gov/biblio/1185847

[28] T. E. Booth, A weight (charge) conserving importance-weighted comb for Monte Carlo.
URL https://www.osti.gov/biblio/204256

[29] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, K. T. Clarno, Denovo: A new three-dimensional parallel discrete ordinates code in scale, Nuclear Technology 171 (2) (2010) 171–200. arXiv:https://doi.org/10.13182/NT171-171, doi:10.13182/NT171-171.
URL https://doi.org/10.13182/NT171-171

[30] B. R. Herman, B. Forget, K. Smith, B. N. Aviles, Improved diffusion coefficients generated from Monte Carlo codes., in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2013), American Nuclear Society, 2013.

[31] T. M. Pandya, et al., Implementation, capabilities, and benchmarking of shift, a massively parallel Monte Carlo radiation transport code, Journal of Computational Physics 308 (2016) 239–272.

[32] E. E. Lewis, M. A. Smith, N. Tsoulfanidis, G. Palmiotti, T. A. Taiwo, R. N. Blomquist, Benchmark specification for deterministic 2-D/3-D MOX fuel assembly transport calculations without spatial homogenisation (C5G7 MOX), Tech. rep., Nuclear Energy Agency, Nuclear Science Committee, Expert Group on 3-D Radiation Transport Benchmarks Report (2001).

[33] M. D. DeHart, Z. Mausolff, Z. Weems, D. Popp, K. Smith, F. Shriver, S. Goluoglu, Z. Prince, J. Ragusa, Preliminary results for the OECD/NEA time dependent benchmark using rattlesnake, rattlesnake-iqs and TDKENO, Tech. Rep. INL/EXT-16-39723; TRN: US1702267, Idaho National Laboratory (8 2016). doi:10.2172/1364489.
URL https://www.osti.gov/biblio/1364489

[34] X. Guo, X. Shang, J. Song, G. Shi, S. Huang, K. Wang, Kinetic methods in monte carlo code rmc and its implementation to c5g7-td benchmark, Annals of Nuclear Energy 151 (2021) 107864. doi:https://doi.org/10.1016/j.anucene.2020.107864.
URL https://www.sciencedirect.com/science/article/pii/S0306454920305624

[35] J. C. Wagner, D. E. Peplow, S. W. Mosher, Fw-cadis method for global and regional variance reduction of monte carlo radiation transport calculations, Nuclear Science and Engineering 176 (2014) 37–57.

[36] A. Davis, A. Turner, Comparison of global variance reduction techniques for monte carlo radiation transport simulations of ITER,

Fusion Engineering and Design 86 (9) (2011) 2698–2700, proceedings of the 26th Symposium of Fusion Technology (SOFT-26). doi:https://doi.org/10.1016/j.fusengdes.2011.01.059.
URL https://www.sciencedirect.com/science/article/pii/S0920379611000718

[37] Q. Pan, H. Lv, S. Tang, J. Xiong, X. Liu, Pointing probability driven semi-analytic monte carlo method (pdmc) – part i: Global variance reduction for large-scale radiation transport analysis, Computer Physics Communications 291 (2023) 108850. doi:https://doi.org/10.1016/j.cpc.2023.108850.
URL https://www.sciencedirect.com/science/article/pii/S0010465523001959

[38] B. Kochunas, Theoretical convergence rate analysis of a unified CMFD formulation with various diffusion coefficients, in: International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2019), American Nuclear Society, 2019, pp. 978–990.

[39] D. Wang, S. Xiao, A linear prolongation approach to stabilizing cmfd, Nuclear Science and Engineering 190 (1) (2018) 45–55. arXiv:https://doi.org/10.1080/00295639.2017.1417347, doi:10.1080/00295639.2017.1417347.
URL https://doi.org/10.1080/00295639.2017.1417347

Appendix A. Nomenclature

Table A.3: Symbols used throughout document.

$\boldsymbol{x}$	position	E	energy
$\hat{\boldsymbol{\Omega}}$	direction	t	time
ψ	angular flux	C	delayed neutron precursor concentration
λ	decay constant	j	delayed neutron precursor group
k^0_{eff}	multiplication factor at	v	particle speed
	initial steady state	β	delayed neutron fraction
χ_{p}	prompt neutron emission spectrum	χ_{d}	delayed neutron emission spectrum
Σ_{t}	macroscopic total cross section	Σ_{s}	macroscopic scatter cross section
Σ_{f}	macroscopic fission cross section	ν	fission neutron multiplicity
$\phi\left(\boldsymbol{x}, E, t\right)$	scalar flux	Σ_{s0}	zeroth Legendre moment
D	diffusion coefficient		of the scattering cross section
ρ	dynamic reactivity	p	reactor power amplitude
β^{eff}	effective delayed neutron fraction	Λ	effective mean neutron generation time
ζ	adjoint-weighted delayed	λ^{eff}	effective decay constant
	neutron precursor concentration	ξ	uniform random number
l	fission source iteration index	$\psi^{\dagger}$	adjoint angular flux
w	particle weight		