공학박사학위논문

Enhancement of Direct Whole Core Transient Calculation Capability

직접 전노심 과도해석능 고도화

2022년 2월

서울대학교 대학원

에너지시스템공학부

강 준 수

Enhancement of Direct Whole Core Transient Calculation Capability

직접 전노심 과도해석능 고도화

지도 교수 주 한 규

이 논문을 공학박사 학위논문으로 제출함
2022년 1월

서울대학교 대학원
에너지시스템공학부
강 준 수

강준수의 공학박사 학위논문을 인준함
2022년 1월

위 원 장 심 형 진 (인)
부위원장 주 한 규 (인)
위 원 이 은 기 (인)
위 원 이 현 철 (인)
위 원 김 응 수 (인)

Abstract

As regulations for nuclear reactors have been strengthened, and demand for high-fidelity multi-physics simulation has increased, high-fidelity direct whole core transient calculation is required. However, due to its heavy computational burden, application of direct whole core transient calculation to realistic core problems requires too long computing time or leadership class computing facilities containing thousands of CPU cores. In this work, the efficiency of direct whole core transient capability of nTRACER is enhanced through applying GPU computing technique and improvement of methodology.

The 3D direct whole core transport calculation of nTRACER is composed of 2D planar method of characteristics (MOC), 3D coarse mesh finite difference (CMFD), and 1D axial MOC calculations. In this work, quasi-static method is employed which solve the point kinetics equation (PKE) to simulate the temporal variation of overall amplitude of neutron flux. Consequently, 3-level method composed of MOC/CMFD/PKE which uses different time step size for each level is implemented in nTRACER. Relatively large time step sizes are used for MOC and CMFD calculations, whose computational burden is heavy and results vary slowly. On the other hand, small time step size is used for PKE calculation. As a consequence, the calculation burden of 3D direct whole core calculation can be alleviated without significant loss of accuracy.

Since the temporal change rate of each variation vary over time, adaptive solution method is implemented to avoid unnecessary MOC and CMFD calculation and minimize the required cost to the accuracy. For MOC calculation, adaptive solution is implemented by performing MOC calculation only when the core conditions are change significantly. In this work, rather than using

simple cross section change, fine mesh residual norm is used as a MOC invoking criterion, which can evaluate the effect of MOC update more precisely. The time step size for CMFD calculation is controlled by the algorithm that is designed to maintain the error occur at each step below prescribed tolerance. To implement the adaptive time step control algorithm, new error model is derived. By using this model, error occur at each time step is calculated during transient calculation and the calculated error is used to determine the time step size. 5×5 fuel assembly problem is used to evaluate the new adaptive solution methods. The new MOC invoking criteria reduces the relative peak power error by 80 % level when compared to previous criteria. The adaptive time step size control algorithm effectively controls the error below prescribed tolerance for the interval where the core power level changes rapidly. When compared to the fixed time step size case which has similar number of time step, the adaptive time step size control algorithm decreases the relative peak power error by 80 % level.

GPU computing technique is applied for computationally intensive components of nTRACER such as MOC or CMFD. Since the characteristics of GPU are different from CPU, optimization is made accordingly. Especially, optimization of transient CMFD linear system solution is performed intensively in this work. First, rather than the group-major ordered linear system solution which is inappropriate for GPU computing, multi-group direct solution method is employed to enhance the stability and speed of the convergence. And, instead of ILU (Incomplete LU) preconditioner, SPAI (Sparse approximate inverse) preconditioner is used to utilize the massive parallelism of GPU. SPAI preconditioner is constructed by approximating inverse of the CMFD matrix

using prescribed sparsity structure. Since the sparsity structure used for the construction determines the construction cost and the quality of preconditioning, it is important to find the optimal sparsity structure. In this work, SPAI prediction algorithm is devised for optimization of sparsity structure. By using the SPAI prediction algorithm with 2 % drop criteria, the required iteration number is reduced by 13 % when compared to existing fixed sparsity structure.

The effectiveness of new direct whole core transient capability is verified through various realistic core problems. Verification of neutron kinetics, kinetics data treatment, and thermal feedback model is performed using SPERT III E-core RIA (Reactivity initiated accident) experiments. For 5 representative tests, the calculated values for experimental data such as peak power level, reactor period, and released energy show good agreement within uncertainty range. When using conventional two step method to analyze same experiment, large difference between calculated value and experimental data occurs. All calculations are run on the cluster containing 20 commercial GPUs and are finished within 7 hours. It is 7 times faster than the calculation using CPU version of nTRACER on the cluster containing 320 GPUs. Considering the price of computing facilities for each calculation, the new transient capability is about 13 times cost effective than the previous CPU version. To check the possibility of actual use of direct whole core transient calculation for realistic core problem, hypothetical RIA in APR1400 core is used. The RIA is simulated up to 1 s using 125 time steps. The calculation is run on the cluster containing 24 GPUs and is finished within 19 hours.

Keywords: Direct whole core transient calculation

Multi-level method

GPU acceleration

Adaptive time step control

High fidelity reactor analysis

Student ID: 2016-21282

Contents

List of Figures

List of Tables

1 Introduction

1.1 Study Background and Motivation

An accurate prediction of neutron behavior under transient conditions is one of the main concerns in the nuclear reactor physics field. The time-dependent Boltzmann transport equation is the fundamental equation that describes the neutron behavior under transient conditions. The solution of time-dependent Boltzmann transport equation is the angular flux which has degree of freedom of seven (three for spatial, two for angular, energy, and time). Therefore, directly solving the time-dependent Boltzmann transport equation requires enormous computing power and long time.

Due to the large computational burden of direct whole core transport calculation, so-called "two-steps" procedure has been employed in both steady-state and transient neutronics analyses. The two-steps procedure involves i) pre-generation of assembly-wise homogenized and group-condensed cross sections and ii) few-group diffusion theory based core calculation. Unavoidable error is introduced in the two-steps procedure by the pre-generation of few-group assembly-wise homogenized cross section. Those approximations are acceptable for homogeneous core problems, but introduce non-trivial error in heterogeneous problems. Furthermore, in the transient calculation, additional

error is caused by the loose coupling with thermal-hydraulics (T/H) calculation and the lower average energy of the delayed neutrons.

It was only in the early 2000s that research on direct whole core transport calculations began to be intensively conducted due to the growth of computing power. Among the various numerical methods, the method of characteristics (MOC) has received a considerable attention due to its capability for heterogeneous geometry handling, anisotropic scattering treatment, and scalability. Especially 2D/1D approach, that couples 2D planar MOC and fine-mesh 1D axial solution within 3D coarse mesh finite difference (CMFD) framework, is considered as the most practical approach for direct whole core transport calculation (Joo et al. 2004; Jung et al. 2013; Kochunas et al. 2013). This approach significantly reduced the heavy computational burden of 3D transport calculation while retaining the accuracy. As a consequence, direct whole core calculation codes employing 2D/1D approach have been successfully analyzed commercial light water reactors (LWRs). Especially, the use of thousands CPU cores in leadership-class computing facilities enables to complete steady-state calculation of commercial LWR core within minutes.

The success for steady state analysis has led to considerable interest in the application of direct whole core calculation in transient reactor analysis (Cho et al. 2005; Zhu et al. 2015). Much heavier computational burden than the steady state calculation has emerged as a new challenge for direct whole core transient calculation. In general, dozens to hundreds of time points are needed to analyze a single transient condition. Because of the short neutron generation time, time-dependent neutron transport equation is a stiff differential equation which requires an implicit solution method. When using an implicit method, the computational burden is greatly increased because the time-dependent

neutron transport calculation should be solved at each time points. Therefore, several researches have been conducted to alleviate the heavy computational burden of direct whole core transport transient calculation.

The transient capability of DeCART code (Cho et al. 2005) was the first attempt of the MOC based direct whole core transient calculation. In DeCART, the MOC calculation was done intermittently based on the rationale that the sub-pin level regional flux shape determined by MOC calculation wouldn't vary much unless there is a significant change in the material composition of the cell. The MOC calculation only updates the sub-pin level flux shape and radial current correction factor used in 3D CMFD kernel. The most transient calculations are carried by the 3D CMFD kernel. To avoid the unnecessary MOC update, the MOC update is performed only when the condition of the core is changed significantly. This intermittent MOC update based on the monitoring of the core condition is called conditional MOC update scheme.

After then, there were several approaches to alleviate the computational burden of direct whole core transport transient calculation by adding additional level of neutronics solution other than 2D planar MOC and 3D CMFD kernel. In MPACT code (Zhu et al. 2015), 3 level multi-level methodology was employed which is composed of 2D MOC, 3D CMFD, and exact point kinetics equation(PKE). In this multi-level method, the neutron flux is factorized into fast varying amplitude and slow varying shape. Amplitude represents the overall magnitude of the neutron flux of whole core, so it doesn't have spatial and energy dependence. Therefore, the amplitude is calculated as a PKE solution. By using micro time step size, that is much smaller than macro time step size used for CMFD calculation, for PKE calculation amplitude can be calculated accurately without posing heavy computational burden. This kind of approach

that factorizes the neutron flux into one amplitude and shape is called quasi-static method. The coupling between different level of solutions (MOC and CMFD, CMFD and PKE) was carried out using predictor-corrector approach in MPACT. This approach has an advantage that the nonlinear coupling between different level solutions is resolved without significant loss of accuracy and without using iterative methods. The multigrid amplitude function (MAF) method is employed in OpenMOC code (Shaner 2014). Unlike quasi-static method, coarse-mesh based amplitude is introduced rather than whole-core amplitude in MAF.

The transient methodology of DeCART involving conditional MOC update scheme was employed during the early development stage of nTRACER. In nTRACER, not only the MOC calculation but also the subgroup fixed source problem (SGFSP), which is performed to treat spatial self-shielding on-the-fly, is also performed conditionally (Jung and Joo 2015). The maximum fuel temperature change of cell is used as SGFSP invoking criterion because escape cross section is depend on the core temperature.

The direct whole core transport transient capabilities have been validated using various problems. In spite of improvement in transient methodology, the heavy computational burden of transport transient calculation has still been an obstacle to the application to realistic core problems. Most of the validation were limited to homogenized problem with simplified thermal feedback model such as 2D LRA benchmark problem (Tsujita et al. 2013), or the problems without thermal feedback like C5G7-TD benchmark (Boyarinov et al. 2016). There have been a few attempts to apply the direct whole core transient calculation to realistic core problems, but there have been limitations caused by the calculation burden. It required too long computing time or strongly

relied on high performance computing facilities. The computing time required for analysis of RIA in one-eighth core using DeCART was approximately 5 days (Hursin et al. 2012). MPACT achieved hours of computing time for RIA in quarter core through utilization of thousands of cores in leadership-class computing facilities (Kochunas et al. 2017). However, so far, it is not common in the industry to use such high performance computing facilities due to the limited budget.

In order to reduce the strong reliance on high performance computing facilities, graphics processing units (GPUs) computing technology was adopted in steady-state calculation module of nTRACER (Choi et al. 2018). Utilization of GPU in computations other than computer graphics is referred as general-purpose computing on GPUs (GPGPU). GPGPU can achieve better performance with smaller cost than CPU computing in terms of energy, heat production, and cost, when used in adequate applications. The application which is suitable for GPGPU can employ data parallel, in which operations are executed on different data elements simultaneously, and involves lots of data elements to process in parallel. The ray tracing calculation in MOC and linear system solution in CMFD have these properties. Therefore, GPGPU was adopted for these routines and great performance improvement was achieved. These solvers are newly coded to reflect the processor architecture of GPU, because GPU has a very different processor architecture from that of CPU. As a result, single consumer-grade GPU card, which is affordable, can produce similar performance of hundreds of CPU cores when used in the direct whole core transport calculation.

1.2 Objectives and Scopes

The direct whole core transient calculation capability of nTRACER has been advanced to meat the following purposes:

- Establish a direct whole core transient calculation methodology with enhanced efficiency and stability,
- Accomplish the applicability of direct whole core transient calculation using affordable small computing facility to a realistic core problem.

The multi-level methods used in direct whole core transient calculation codes has successfully alleviated the heavy computational burden from frequent transport calculations. Similar to MPACT, 3-level solution structure consisting of MOC, CMFD and PKE was adopted in nTRACER. However, nTRACER has its own coupling method between different solution models. The nonlinear coupling between CMFD and PKE is resolved through fixed point iteration which is referred as improved quasi-static (IQS) method. It is less preferred than predictor-corrector quasi-static (PCQS) method, which is used in MPACT, because of the cost from nonlinear iteration. But there are already many components in direct whole core transient calculation whose solutions should be converged through fixed point iteration such as thermal-hydraulics, MOC, CMFD. Therefore, adding one more components, PKE, may not increase the cost from nonlinear iteration and could get better accuracy from tight coupling. Comparison with various quasi-static coupling methods and some legacy methods including exponential transform (ET) and stiffness confinement method (SCM) was performed to confirm the performance of IQS method. For the coupling between CMFD and MOC, the sub-pin level flux shape and current

correction factor is remained constant between intermittent MOC calculations because of the slow change of these variables.

Using the multi-level method is based on the fact that each solution models has different temporal resolution requirements. The solution model with higher spatial resolution requires lower temporal resolution while solutions with lower spatial resolution such as PKE requires higher temporal resolution. By using adequate time step size for each component, more efficient calculation is possible without unnecessary waste of calculation costs. However, the temporal resolution requirements for each solution models are problem dependent and remains unknown before running the actual calculation. To guarantee the accuracy of transient calculation, small time step size should be used, but this can significantly increase the computational cost in the case of MOC and CMFD. Therefore, adaptive algorithm that can control the time step size on-the-fly is required for efficient yet accurate transient calculation. The conditional transport update scheme originally adopted in DeCART can be considered as one of these adaptive algorithm for MOC solution. The MOC invoking criteria is improved by devising more sophisticated model that can quantify the effect of MOC update more precisely than previous cross section change criterion. The normalized fine mesh residual norm is estimated at each macro time steps and be used as a new MOC invoking criterion. Also, adaptive time step size control (ATC) algorithm for CMFD calculation is developed. ATC controls the time step size to maintain the magnitude of local error below the prescribed tolerance. Previous approaches to adaptively select the time step size used for shape calculation of quasi-static method (Prince and Ragusa 2019; Caron et al. 2017) only consider the truncation error in shape calculation and only applied in diffusion calculation solver. In multi-physics

calculation involving thermal feedback, the local amplitude error can occur due to various causes other than truncation error of shape calculation. New error model incorporating possible errors was derived and the time step size for CMFD calculation is adjusted based on this model.

Since MOC calculation is performed intermittently, the proportion of CMFD calculation time increases in transient calculation. Therefore, optimization of transient CMFD calculation is essential for efficient transient calculation. In steady-state calculation, CMFD is an eigenvalue problem which is solved by power iterative method, whereas in transient calculation, CMFD is a fixed-source problem. Therefore, the method of solving the transient CMFD calculation should also be optimized accordingly. Through investigation, it was found to be inadequate to use conventional group major order CMFD formulation for transient CMFD calculation. As a substitution for group major order CMFD formulation, whole CMFD linear system is solved directly using preconditioned BiCGSTAB solver. The convergence of preconditioned BiCGSTAB has strong dependence on the quality of preconditioner. Sparse approximate inverse (SPAI) preconditioner (Grote and Huckle 1997) which is appropriate for GPGPU is employed in nTRACER. SPAI preconditioner is generated using prescribed sparsity structure and the quality of preconditioning is dependent to sparsity structure. Efficient a priori sparsity structure determination algorithm using the characteristics of CMFD matrix is devised, and it is presented in this thesis.

In this thesis, all improvements made to the transient calculation capabilities of nTRACER is covered. In Chapter 2, the direct whole core transient calculation methodology is overviewed. The transport transient solution based on 2D/1D method consisting of 2D planar MOC within 3D CMFD framework is described. The effective cross section generation procedure required for

precise consideration of thermal feedback effect is also detailed. The adjoint flux calculation capabilities and control rod decusping method which are required in transient calculation are also described in this chapter. The more detailed description of the multi-level method implemented in nTRACER is followed in Chapter 3. This chapter describes how the different solution models are coupled and the examination results of various flux factorization methods. In Chapter 4, two adaptive algorithms are addressed: adaptive macro time step control algorithm and conditional transport update scheme. In the first section of this chapter, how the local error model used to determine the macro time step size is derived and the evaluation results of error model are detailed. In the second section of this chapter, the conditional transport update scheme, in particular, newly devised MOC invoking criterion is described. Chapter 5 presents the optimization of transient CMFD solution using GPGPU. CMFD formulation is determined through comparison of efficiency and sparsity pattern determination algorithm is devised to generate more precise SPAI preconditioner. The accuracy of direct whole core calculation methodology and applicability to a realistic core problem of new direct whole core transient calculation driver is validated through SPERT III E-core analysis(McCardell et al. 1969) and APR1400 analysis. The results of realistic core analyses are described in Chapter 6. Chapter 7 concludes this thesis.

2 Direct Whole Core Transient Calculation Methodology

For high fidelity transient calculation, accurate calculation of sub-pin level distribution of the neutron flux and its change during transient are required. Since 3D transport calculation for sub-pin level spatial resolution is very complicated, an efficient yet accurate methodology is required. In addition to the calculation method for neutron flux, it is also important to accurately produce the effective cross section and kinetics data used for it. Significant changes in effective cross section occur during transient events. Accordingly, this changes should be considered through appropriate methods.

In nTRACER, the sub-pin level flux distribution is calculated through MOC calculation. However, utilization of MOC to a 3D transport calculation of reactor is not feasible without any approximation, in spite of the advanced computing capabilities of the present day. As a practical option to perform MOC based direct whole core transport calculation, 2D/1D method was developed which utilizes a planar 2D MOC solution and 1D axial solution within 3D CMFD framework. The Fig. 2.1 shows the schematic diagram of 2D/1D transport method. The whole method is composed of two main components: 2D planar MOC and 3D CMFD kernel. The 3D CMFD kernel is consistent to the whole core transport problem. It is considered as a main solver so that the

pin level flux distribution is determined by its solution. 2D planar MOC solver just determine the sub-pin level shape of the flux and the surface current with given pin-wise fission source distribution from 3D CMFD kernel. The 2D MOC calculations are performed on thick planes and 3D CMFD solver has axially tall nodes. The axial variation of flux within the tall CMFD node through the 1D axial calculation. The 1D axial calculation which has higher order or finer mesh structure than 3D CMFD solver is performed for each pin. With given radial leakage information from the 3D CMFD solution, it determines diffusion correction factor for axial direction in 3D CMFD calculation. In nTRACER, augmented 1D axial MOC solver is employed as an axial solver for enhancement of stability. Despite the thick plane structure for MOC and tall nodes in 3D CMFD calculation, this method shows sufficient accuracy thanks to the fact that the axial material heterogeneity is small when compared to the radial material heterogeneity in LWRs. This efficient transport calculation method can be extended to transient problem. In the following sections, the detailed derivation of the time-dependent transport solution will be discussed.

The Doppler broadening has a significant effect on the neutronics results such as power distribution and reactivity of the core. Especially, the fuel temperature changes rapidly when the power of the core rises sharply during transient event, and accordingly, the Doppler broadening is a major feedback source. Therefore, the Doppler broadening and its effect in the resonance self-shielding should be considered. The subgroup method is used to generate the effective cross section which incorporate the effect of Doppler broadening.

The kinetic parameters such as delayed neutron data and neutron velocities also affects the results of transient calculation significantly. Several methods are

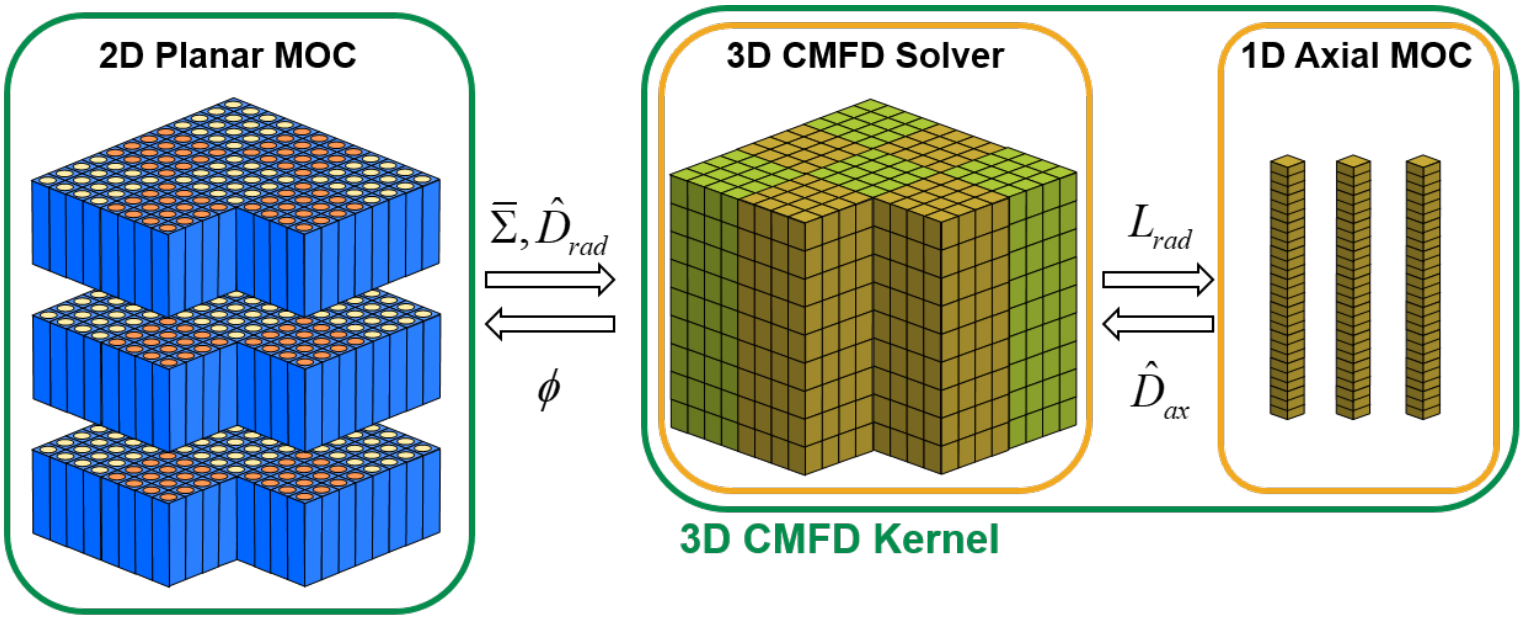


Figure 2.1 Schematic diagram of nTRACER 3D transport calculation framework

applied for simplified treatment of these kinetic parameters without significant loss of accuracy.

2.1 Time-dependent Neutron Transport Solutions

2.1.1 Time-dependent Planar MOC Solution

The sub-pin level transport transient solution is obtained by solving the following 2D time-dependent multi-group planar transport problem for Angle m and Plane k:

$$\begin{aligned}\frac{1}{v_g}\frac{\partial}{\partial t}\varphi_g^{m,k}(x,y,t) = Q_g^{m,k}(x,y,t) - L_g^{m,k}(x,y,t) \\ - \left(\Omega_x^m \frac{\partial}{\partial x} + \Omega_y^m \frac{\partial}{\partial y} + \Sigma_g^k(x,y,t)\right),\end{aligned} \tag{2.1}$$

where φ is angular flux, L_z is axial leakage, and Q is the neutron source. The neutron source includes both prompt and delayed fission sources and defines as:

$$\begin{aligned}Q_g^{m,k}(x,y,t) = \frac{1}{4\pi}\sum_{g'=1}^{G}\Sigma_{s,g'g}^k(x,y,t)\phi_{g'}^k(x,y,t) \\ + \frac{\chi_{p,g}^k(x,y,t)\left(1-\beta_{tot}^k(x,y,t)\right)}{4\pi}\psi^k(x,y,t) \\ + \frac{\chi_{p,g}^k(x,y,t)}{4\pi}Q_d^k(x,y,t),\end{aligned} \tag{2.2}$$

where $\chi_{p,g}$ is prompt fission spectra, $chi_{d,g}$ is delayed fission spectra, ϕ is scalar flux, β_{tot} is total delayed neutron fraction, and ψ and Q_d are the total fission source and the delayed neutron source which are defined as:

$$\varphi^k(x,y,t) = \sum_{g'=1}^{G}\nu\Sigma_{f,g'}^k(x,y,t)\phi_{g'}^k(x,y,t), \tag{2.3}$$

$$Q_d(x,y,t) = \sum_{j=1}^{n_{prec}} \lambda_j^k(x,y,t) C_j^k(x,y,t), \tag{2.4}$$

where λ_j is delayed neutron precursor decay constant, and C_j is the delayed neutron precursor density of precursor group j. The delayed neutron precursor density is obtained by solving the precursor equation which is given as:

$$\begin{aligned} \frac{dC_j^k(x,y,t)}{dt} = \beta_j^k(x,y,t)\psi^k(x,y,t) - \lambda_j^k(x,y,t)C_j^k(x,y,t), \\ j = 1, \ldots, n_{prec}. \end{aligned} \tag{2.5}$$

Due to its stability and large memory requirements of high-order methods (e.g. high-order backward differentiation formula), the Backward Euler method is used for temporal discretization of Eq. (2.1). At n^{th} time point t_n with given time step size Δt_n, Eq. (2.1) is discretized as:

$$\begin{aligned} &\left(\Omega_x^m \frac{\partial}{\partial x} + \Omega_y^m \frac{\partial}{\partial y} + \Sigma_g^{k,n}(x,y) + \frac{1}{v_g \Delta t_n}\right) \varphi_g^{m,k,n}(x,y) \\ &\qquad = Q_g^{m,k,n}(x,y) - L_{z,g}^{m,k,n}(x,y) + \frac{\varphi_g^{m,k,n-1}(x,y)}{v_g \Delta t_n}. \end{aligned} \tag{2.6}$$

When consider the right-hand side of it is as a fixed source, Eq. (2.6) can be considered as steady-state transport problem. However, solving Eq. (2.6) directly is not practical due to the memory limitation. Enormous computational memory is required to store all the angular flux of previous time step for each ray segment whose length is about few mm. For a typical whole core problem, the number of ray segments per plane is about 170 million and about 90 GB memory is required per plane to save angular flux data. The isotropic approximation is applied to angular flux difference term in order to reduce the memory requirement (Cho et al. 2005). Although this approximation may

result in nontrivial errors if angular flux is very large in certain directions as noted in the large void regions of boiling water reactors (BWRs), it would be acceptable in most transients of pressurized water reactors (PWRs) to which nTRACER is targeted. The resulted equation is written as:

$$\begin{aligned}&\left(\Omega_x^m\frac{\partial}{\partial x}+\Omega_y^m\frac{\partial}{\partial y}+\Sigma_g^{k,n}(x,y)\right)\varphi_g^{m,k,n}(x,y)\\&=Q_g^{m,k,n}(x,y)-L_{z,g}^{m,k,n}(x,y)-\frac{\phi_g^{k,n}(x,y)-\phi_g^{k,n-1}(x,y)}{4\pi v_g\Delta t_n}.\end{aligned}\tag{2.7}$$

Note that the scalar fluxes resulted from the isotropic approximation are in the difference form. With this difference form, influence of the error posed by the isotropic approximation would be reduced (Cho et al. 2005).

The partial differential equation (PDE) problem in Eq. (2.7) can be solved using MOC. MOC is a numerical method to solve a PDE by converting the PKE into a family of ordinary differential equations (ODEs). For MOC solution, characteristic curves are defined each of which has a fixed direction. The characteristic curve which has an Angle m is defined as:

$$\overrightarrow{r}=s\Omega.\tag{2.8}$$

By substitute Eq. (2.8) into Eq. (2.7), following ODE can be obtained:

$$\frac{d}{ds}\Phi_g^{m,k,n}(s)+\Sigma_g^{k,n}(s)\varphi_g^{m,k,n}(s)=\tilde{Q}_g^{m,k,n}(s),\tag{2.9}$$

where

$$\tilde{Q}_g^{m,k,n}(s)=Q_g^{m,k,n}(s)-L_{z,g}^{m,k,n}(s)-\frac{\phi_g^{k,n}(s)-\phi_g^{k,n-1}(s)}{4\pi v_g\Delta t_n}.\tag{2.10}$$

To solve Eq. (2.9), the spatial regions are discretized into sub-pin level regions called *flat source region*(FSR). In each FSR it is assumed that the source term $\tilde{Q}$ is constant. Also, there are set of FSRs called *flat cross section region*(FXR) where the cross section is assumed to be constant. With flat source and flat cross section assumption, Eq. (2.9) can be solved for a characteristic curve which has a direction of Angle m and passing FSR i using the method of integrating factor as:

$$\varphi_{g,i}^{m,k,n}(s) = \varphi_{g,i}^{m,k,n}(s_0)\exp(-\Sigma_{g,i}^{k,n}s) + \frac{\tilde{Q}_{g,i}^{m,k,n}}{\Sigma_{g,i}^{k,n}}\left(1-\exp(-\Sigma_{g,i}^{k,n}s)\right). \tag{2.11}$$

By taking average of Eq. (2.11) over the segment of the characteristic curve having direction of Angle m, divided by FSR i, the averaged angular flux can be obtained as:

$$\bar{\varphi}_{g,i}^{m,k,n} = \frac{1}{l}\int_0^l \varphi_{g,i}^{m,k,n}(s)ds = \frac{\varphi_{g,i}^{m,k,n}(0) - \varphi_{g,i}^{m,k,n}(l)}{\Sigma_{g,i}^{k,n}l} = \frac{\tilde{Q}_{g,i}^{m,k,n}}{\Sigma_{g,i}^{k,n}l}. \tag{2.12}$$

where l is the length of the segment. The characteristic curves are defined for a set of angles and spacing given by the user as an input parameter. This characteristic curve can be viewed as a *ray* that travels over the core in a specific direction. By tracking the set of rays and calculating the flux for each segment, average scalar flux for each FSR can be obtained. Therefore, this solution scheme is called *ray tracing calculation*. The scalar flux of the FSR i can be calculated as the weighted sum of angular flux as:

$$\phi_{g,i} = \sum_{m=1}^{M}\omega_m\frac{\delta}{A_i}\sum_{r\in i} l_r\bar{\varphi}_{g,i,r}^{m,k,n}, \tag{2.13}$$

where r is an index of rays, δ is a ray spacing, A_i is the area of FSR i, and ω_m is the weighting factor for angle m.

As shown in Eq. (2.4), the delayed neutron source is coupled to the precursor equation in Eq. (2.5). Instead of solving the precursor equation through temporal discretization and solving the coupled system of neutron and precursor equations by the iterative method, analytic solution for the precursor equation is possible if the temporal variation of the total fission source ψ is approximated as a polynomial over time. To consider the variation of fission source which changes more rapidly than the neutron flux, it is approximated as a quadratic polynomial. Based on the assumption of quadratic fission source variation, the fission source can be expressed as a function of time using the Lagrange interpolation as:

$$\psi^k(x,y,t) = \sum_{l=0}^{2} \psi^{k,n-l}(x,y) \prod_{\substack{0\le m\le 2 \\ m\neq l}} \frac{t - t_{n-m}}{t_{n-l} - t_{n-m}} \tag{2.14}$$

By applying Eq. (2.14) into Eq. (2.5), Eq. (2.5) can be directly solved for the precursor density by applying integrating factor and the resulted solution for the case of time step size Δt_n and Δt_{n-1} is obtained as a function of the fission source at n^{th} time step ψ^n:

$$C_j^{k,n}(x,y) = \kappa_j C_j^{k,n-1}(x,y) + \frac{\beta_j^{k,n}(x,y)}{\lambda_j} \left(\sum_{l=0}^{2} \Omega_j^{n-l} \psi^{k,n-l} \right), \tag{2.15}$$

where

$$\gamma = \frac{\Delta t_n}{\Delta t_{n-1}}, \quad \kappa_j = e^{-\lambda_j \Delta t_n}, \quad \bar{\kappa}_j = 1 - \kappa_j,$$

$$\Omega_j^{n-2} = \frac{1}{\lambda_j \Delta t_{n-1}(\gamma+1)} \left(\frac{2\bar{\kappa}_j}{\lambda_j \Delta t_{n-1}} - \gamma(\kappa_j + 1) \right),$$

$$\Omega_j^{n-1} = \frac{1}{\lambda_j \Delta t_{n-1}} \left(\kappa_j + 1 + \frac{\bar{\kappa}_j}{\gamma} \left(1 - \frac{2}{\lambda_j \Delta t_{n-1}} \right) \right),$$

$$\Omega_j^{n} = 1 - \frac{2}{(\gamma+1)\lambda_j \Delta t_{n-1}} + \frac{\bar{\kappa}_j}{\gamma(\gamma+1)\lambda_j \Delta t_{n-1}} \left(\frac{2}{\lambda_j \Delta t_{n-1} - 1} \right).$$

Since the total fission source which is determined by the neutron flux is the only unknown variable in the right-hand side of Eq. (2.15), the precursor density at n^{th} time step can be obtained by solving Eq. (2.1) without any other iterative procedure.

2.1.2 Time-dependent CMFD Solution

The CMFD accelerates the fission source convergence by rebalancing the fission source distribution in coarse mesh, and tightly couple two high-order solutions (2D planar MOC and 1D axial MOC). The coarse mesh problem equivalent to high-order calculation is generated through the homogenization. The macroscopic cross sections, neutron velocities, and neutron flux are pin-wise homogenized by following relations to preserve the reaction rates of high-order solutions:

$$\Sigma_i = \frac{\sum\limits_{l \in i} \Sigma_l \phi_l V_l}{\sum\limits_{l \in i} \phi_l V_l} \tag{2.16}$$

$$v_i = \frac{\sum\limits_{l \in i} \phi_l V_l}{\sum\limits_{l \in i} \frac{1}{v_l} \phi_l V_l} \tag{2.17}$$

$$\phi_i = \frac{\sum_{l \in i} \phi_l V_l}{\sum_{l \in i} V_l} \tag{2.18}$$

where i is the coarse Mesh index, and l is the sub-pin level fine Mesh index.

The time-dependent CMFD problem is given as the following fixed source problem for homogenized computational Mesh i at n^{th} time point, t_n:

$$\sum_{u=x,y,z} \frac{1}{h_i^u} \left(J_{i,g}^{u+} - J_{i,g}^{u-} \right) + \left(\Sigma_{r,g}^i + \frac{1}{v_g^i \Delta t_n} \right) \phi_g^{i,n} = Q_i^{n-1} + \frac{\phi_g^{i,n-1}}{v_g^i \Delta t_n}, \tag{2.19}$$

where $J_i^{u\pm}$ represents the surface current to the positive and negative u directions. The surface currents are determined by the following CMFD relations:

$$J_{i,g}^{u\pm,n} = -\tilde{D}_{i,g}^{u\pm,n} \left(\phi_{i,g}^{i\pm1,n} - \phi_{i,g}^{i,n} \right) + \hat{D}_{i,g}^{u\pm,n} \left(\phi_{i,g}^{i\pm1,n} + \phi_{i,g}^{i,n} \right), \tag{2.20}$$

where $\tilde{D}$ is the flux difference to current factor commonly used in the finite difference method applied to the Fick's law, and $\hat{D}$ is the current correction factor which is defined as:

$$\hat{D}_{i,g}^{u\pm n} = \frac{J_{i,g,HO}^{u\pm n} + \tilde{D}_{i,g}^{u\pm n} \left(\phi_g^{i\pm1,n} - \phi_g^{i,n} \right)}{\left(\phi_g^{i\pm1,n} + \phi_g^{i,n} \right)}, \tag{2.21}$$

where the subscript HO indicates the high-order solution, i.e. 2D planar MOC and 1D axial MOC.

For simplicity, Eq. (2.19) can be written in matrix operation form as:

$$\mathbf{R}\Phi^n + \mathbf{T}\Phi^n = \mathbf{S}\Phi^n + (\mathbf{F} - \mathbf{F_d})\Phi^n + \mathbf{T}\Phi^{n-1} + Q_d^n, \tag{2.22}$$

where

$$\mathbf{R}\Phi^n\ |_g^i = \sum_{u=x,y,z} \frac{1}{h_i^u}\left(J_{i,g}^{u+} - J_{i,g}^{u-}\right) + \Sigma_{r,g}^i \phi_g^{i,n},$$

$$\mathbf{T}\Phi^n\ |_g^i = \frac{1}{v_g^i \Delta t_n}\phi_g^{i,n},$$

$$\mathbf{F}\Phi^n\ |_g^i = \chi_g^{i,n} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}^{i,n}\phi_{g'}^{i,n},$$

$$\mathbf{F_d}\Phi^n\ |_g^i = \sum_{j=1}^{n_{prec}} \chi_{dg,j}^{i,g}\beta_j^{i,n} \sum_{g'=1}^{G} \nu\Sigma_{f,g'}^{i,n}\phi_{g'}^{i,n},$$

$$Q_d^n\ |_g^i = \sum_{j=1}^{n_{prec}} \chi_{dg,j}^{i,n}\lambda_j C_j^{i,n}.$$

The CMFD linear system can be obtained by moving all terms that contain the unknowns, i.e. scalar flux vector, to the left side:

$$\left(\mathbf{R} + \mathbf{T} - \mathbf{S} - \mathbf{F} + \mathbf{F_d}\right)\Phi^n = \mathbf{T}\Phi^{n-1} + Q_d^i. \tag{2.23}$$

The precursor equation in pin-wise coarse mesh equivalent to Eq. (2.5) for homogenized computational Mesh i is given as:

$$\frac{\partial C_j^i(t)}{\partial t} = \beta_j^i(t)\psi^i(t) - \lambda_j C_j^i(t), \quad j = 1, \ldots, n_{prec}. \tag{2.24}$$

Similar to high-order solution, Eq. (2.24) can be solved analytically using quadratic approximation for the fission source and the result can be written as:

$$C_j^{i,n} = \kappa_j C_j^{i,n-1} + \frac{\beta_j^{i,n}}{\lambda_j}\left(\Omega_j^{n-2}\psi^{i,n-2} + \Omega_j^{n-1}\psi^{i,n-1} + \Omega_j^n\psi^{i,n}\right). \tag{2.25}$$

By inserting Eq. (2.25) into Eq. (2.23), the finalized linear system, that can determine all the unknowns at n^{th} time step including precursor density not only the neutron flux, is constructed as:

$$\left(\mathbf{R}+\mathbf{T}-\mathbf{S}-\mathbf{F}-\tilde{\mathbf{F}}_{\mathbf{d}}\right)\Phi^{n}=\mathbf{T}\Phi^{n-1}+\tilde{Q}_{d}^{i}. \tag{2.26}$$

where

$$\tilde{\mathbf{F}}_{\mathbf{d}}\Phi^{n}\,|_{g}^{i}=\sum_{j=1}^{n_{prec}}\chi_{dg,j}^{i,g}\beta_{j}^{i,n}\left(1+\frac{\Omega_{j}^{n}}{\lambda_{j}}\right)\sum_{g'=1}^{G}\nu\Sigma_{f,g'}^{i,n}\phi_{g'}^{i,n},$$

$$\tilde{Q}_{d}^{n}\,|_{g}^{i}=\sum_{j=1}^{n_{prec}}\chi_{dg,j}^{i,n}\left(\lambda_{j}\kappa_{j}C_{j}^{i,n-1}+\beta_{j}^{i,n}\left(\Omega_{j}^{n-2}\psi^{i,n-2}+\Omega_{j}^{n-1}\psi^{i,n-1}\right)\right).$$

2.2 Effective Cross Section Generation

The Doppler broadening determines the core behavior in transient conditions, e.g. reactivity initiated accident (RIA) in hot zero power (HZP) condition. Therefore, reflecting the resonance self-shielding effect is important for high-fidelity transient calculation. The effective multi-group cross sections of the resonant nuclides are generated on-the-fly during the calculation, and adjusted for change of material composition or temperature condition. The subgroup method and the resonance interference factor library method (RIFL) (Choi et al. 2016) are used to generate the effective cross sections. For the subgroup method, following subgroup fixed source problem (SGFSP) for n^{th} subgroup level is solved:

$$\Omega\cdot\Delta\varphi_{n}(\mathbf{r},\Omega)+\left(N_{R}(\mathbf{r})\sigma_{n}+\lambda\Sigma_{p}(\mathbf{r})\right)\varphi(\mathbf{r},\Omega)=\frac{1}{4\pi}\lambda\Sigma_{p}(\mathbf{r}). \tag{2.27}$$

The resulted scalar flux is converted to the escape cross section, which reflects the heterogeneity effect through leakage, as the following relation:

$$\Sigma_n^{esc}(\mathbf{r}) = \frac{\phi(\mathbf{r})}{1-\phi(\mathbf{r})} N_R(\mathbf{r})\sigma_n - \lambda\Sigma_p(\mathbf{r}). \tag{2.28}$$

The SGFSP of Eq. (2.27) is solved prior to the main transport equation. When determine the self-shielded cross section, the following inverse relation derived from Eq. (2.28) is used to obtain the flux:

$$\phi(\mathbf{r}) = \frac{\lambda\Sigma_p(\mathbf{r}) + \Sigma_n^{esc}(\mathbf{r})}{N_R(\mathbf{r})\sigma_n + \lambda\Sigma_p(\mathbf{r}) + \Sigma_n^{esc}(\mathbf{r})}, \tag{2.29}$$

and the self-shielded cross section is determined by following subgroup formula:

$$\sigma_x(\mathbf{r}, T) = \frac{\sum_{n=1}^{N_s} w_n^x(T)\phi_n(\mathbf{r})\sigma_n}{\sum_{n=1}^{N_s} w_n^x(T)\phi_n(\mathbf{r})}. \tag{2.30}$$

The temperature dependency is considered with the interpolation of subgroup weights and additional parameters, i.e. temperature correction factor (TCF), which are also determined from the SGFSP calculation. To improve the Doppler feedback prediction, the neutron scattering in the U^{238} resonance is accurately modeled in the subgroup parameter generation procedure. The resonance interference factor and the cross sections of non-resonant nuclides are interpolated from pre-tabulated nuclear data. In general, the nuclear data are tabulated with 300 K interval, except for the nuclides used in the moderator.

2.3 Kinetics Parameters Treatment

In the transient formulation, following kinetics parameter data are required: delayed neutron fraction, delayed fission spectra, delayed neutron precursor decay constants, and group-wise neutron velocities. These data are obtained from the ENDF/B-VII.1 nuclear data. The six delay group structure is used in this nuclear data library. The delayed neutron fraction and delayed neutron precursor decay constant are not affected by core condition such as temperature, but only by fissioning isotopes and delayed neutron groups. Because of the isotope dependence of the decay constants, the delayed neutron source should be calculated for each fission isotopes individually. However, this approach in spatial dynamics calculation is too complicated, because usage of macroscopic fission cross section of material mixture is unavailable for separated delayed neutron calculation. To avoid this complexity, the decay constant set of only one representative isotope (e.g., U^{235} for PWR problem) is used for all isotopes.

The fitting procedure is employed for consistent result with single isotope-independent decay constant set (Cahalan and Ott 1973). It preserves following integral parameters from the original data for each isotope:

$$\beta_i = \sum_{j=1}^{6} \beta_{ij} = \sum_{j=1}^{6} \tilde{\beta}_{ij}, \tag{2.31}$$

$$\bar{\lambda}_i = \frac{1}{\beta_i} \sum_{j=1}^{6} \lambda_{ij}\beta_{ij} = \frac{1}{\beta_i} \sum_{j=1}^{6} \tilde{\lambda}_j\tilde{\beta}_{ij}, \tag{2.32}$$

$$\overline{\left(\frac{1}{\lambda_i}\right)} = \frac{1}{\beta_i} \sum_{j=1}^{6} \frac{\beta_{ij}}{\lambda_{ij}} = \frac{1}{\beta_i} \sum_{j=1}^{6} \frac{\tilde{\beta}_{ij}}{\tilde{\lambda}_j}, \tag{2.33}$$

where β_{ij}, λ_{ij} are the delayed neutron fraction and the decay constant from original data, $\tilde{\beta}_{ij}$ is the fitted delayed neutron fraction, and $\tilde{\lambda}_j$ is isotope-independent decay constant for isotope i and delayed neutron group j. Each of above equations represents the total delayed neutron fraction, the average decay constant, and the mean decay time, respectively. This preserving equation can be written in general expression:

$$\sum_{j=1}^{6} \left(\lambda_{ij}\right)^{-\gamma} \beta_{ij} = \sum_{j=1}^{6} \left(\tilde{\lambda}_j\right)^{-\gamma} \tilde{\beta}_{ij}, \qquad \gamma = -1, 0, 1. \tag{2.34}$$

The original data in the left-hand side of Eq. (2.34) can be represented as a linear combination of isotope-independent decay constants:

$$\left(\lambda_{ij}\right)^{-\gamma} \beta_{ij} = \sum_{k=j-1}^{j+1} \left(\tilde{\lambda}_k\right)^{-\gamma} Y_k^{ij}, \qquad \gamma = -1, 0, 1. \tag{2.35}$$

The coefficients Y_k^{ij} are determined by solving the following linear system:

$$\begin{pmatrix} 1 & 1 & 1 \\ \tilde{\lambda}_{j-1} & \tilde{\lambda}_j & \tilde{\lambda}_{j+1} \\ 1/\tilde{\lambda}_{j-1} & 1/\tilde{\lambda}_j & 1/\tilde{\lambda}_{j+1} \end{pmatrix} \begin{pmatrix} Y_{j-1}^{ij} \\ Y_j^{ij} \\ Y_{j+1}^{ij} \end{pmatrix} = \begin{pmatrix} \beta_{ij} \\ \beta_{ij}\lambda_{ij} \\ \beta_{ij}/\lambda_{ij} \end{pmatrix}. \tag{2.36}$$

The fitted delayed neutron fraction can be determined by applying Eq. (2.35) into the left-hand side of Eq. (2.34) and reordering it:

$$\sum_{j=1}^{6} \sum_{k=j-1}^{j+1} \left(\tilde{\lambda}_k\right)^{-\gamma} Y_k^{ij} = \sum_{j=1}^{6} \left(\tilde{\lambda}_j\right)^{-\gamma} \sum_{k=j-1}^{j+1} Y_j^{ik}, \qquad \gamma = -1, 0, 1. \tag{2.37}$$

By comparing the right-hand side of Eq. (2.34) and Eq. (2.37), the fitted delayed neutron fraction is defined as:

$$\tilde{\beta}_{ij} = \sum_{k=j-1}^{j+1} Y_j^{ik}. \tag{2.38}$$

The fitted delayed neutron fraction $\tilde{\beta}$ is denoted by β later on for simplicity. Because of isotope dependence of the delayed neutron fraction, the average delayed neutron fraction is calculated for delayed neutron group j for each FXR:

$$\beta_j = \frac{\sum_{i=1}^{I} \beta_{ij} \left(\sum_{g=1}^{G} \nu\sigma_{fg}^{i} N_i \phi_g \right)}{\sum_{i=1}^{I} \sum_{g=1}^{G} \nu\sigma_{fg}^{i} N_i \phi_g}, \tag{2.39}$$

where $\nu\sigma_{fg}^{i}$ is the microscopic nu-fission cross section of isotope i for group g, and N_i is the number density of isotope i.

The delayed emission spectrum is much softer than the prompt fission spectrum. Therefore appropriate treatment is required for delayed emission spectrum. It has dependence on both isotope and the delayed neutron group. Fortunately the dependence on isotope is relatively small as shown in Fig. 2.2. Therefore, the isotope dependence is neglected and only one spectrum that doesn't change during transient situation is used for whole core problem. The spectrum is obtained by averaging over whole core problem after steady-state calculation. On the other hand, the delayed emission varies significantly with different delayed neutron groups as shown in Fig. 2.3. The delayed neutron groups are considered by calculating the averaged delayed neutron spectrum

for each delayed neutron group as:

$$\bar{\chi}_{dk,g} = \frac{\sum_j V_j \sum_i \chi^i_{dk,g,j} \beta^i_{k,j} \psi^i_j}{\sum_g \left[\sum_j V_j \sum_i \chi^i_{dk,g,j} \beta^i_{k,j} \psi^i_j \right]}, \tag{2.40}$$

where i is isotope, j is FXR index, k is delayed neutron group, and ψ represents a fission source. The delayed emission spectra of each delayed neutron precursor group doesn't have dependence on node because the isotope dependence is neglected. But the total delayed emission spectrum has the node dependence because of the node dependence of delayed neutron fraction:

$$\chi^j_{d,g} = \frac{\sum_k \beta^j_k \chi_{dk,g}}{\sum_g \sum_k \beta^j_k \chi_{dk,g}}, \tag{2.41}$$

where j is a node index. The prompt emission spectrum is calculated during transient calculation, because nTRACER multi-group cross section library contains only averaged fission spectrum which is used for steady-state calculation:

$$\chi_g = (1-\beta)\chi_{p,g} + \beta\chi_{d,g}. \tag{2.42}$$

By using the delayed emission spectrum, prompt emission spectrum is computed in transient calculation as following equation:

$$\chi_{p,g} = \frac{\chi_g - \beta\chi_{d,g}}{1-\beta}. \tag{2.43}$$

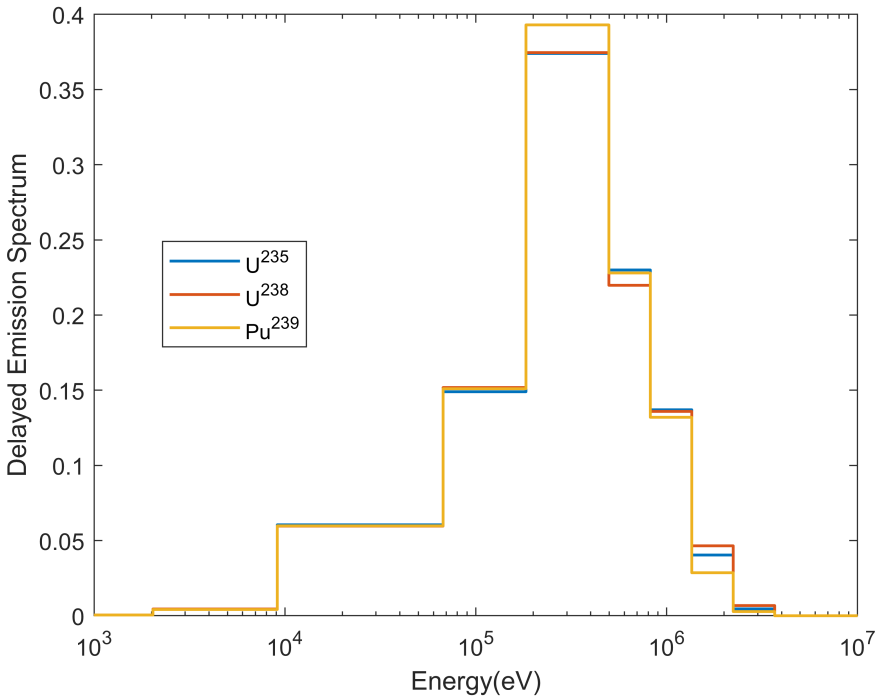


Figure 2.2 Averaged delayed emission spectrum for various isotopes

The multi-group averaged neutron velocities can be obtained by spectrum weighting which preserves the change rate of the neutron density:

$$\frac{1}{v_g} = \frac{\int_g \frac{1}{v(E)}\phi(E)dE}{\int_g \phi(E)dE}. \tag{2.44}$$

Note that the velocity can be expressed as a function of energy as follows:

$$v(E) = v_{th}\sqrt{\frac{E}{E_{th}}}, \tag{2.45}$$

where the subscript th indicates the thermal neutron, whose energy is 0.025 eV and velocity is 2,200 m/s. It is calculated during nTRACER multi-group

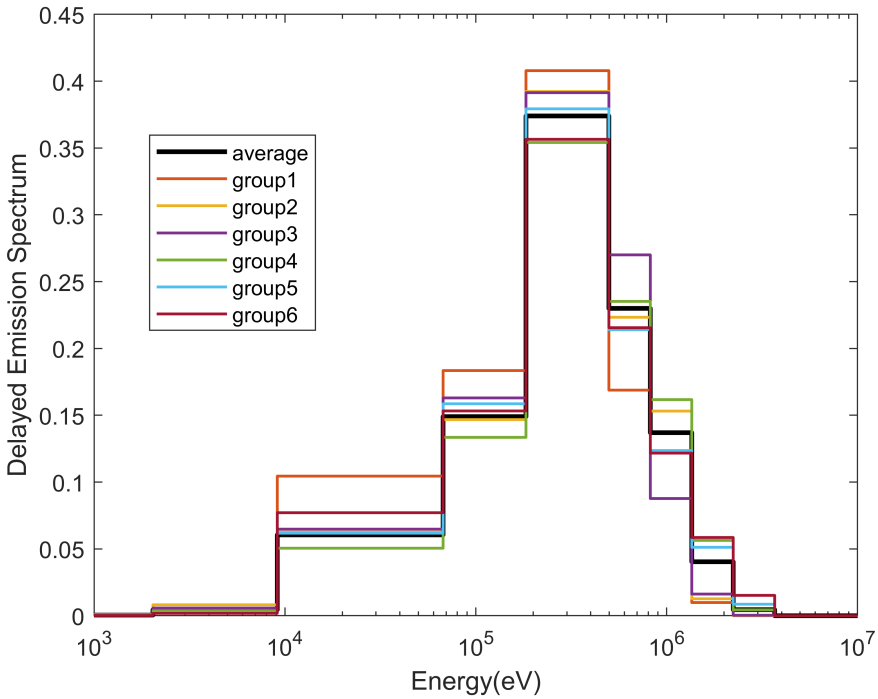


Figure 2.3 U^{235} delayed emission spectrum for various delayed neutron groups

cross section library generation procedure, using typical neutron specturm in LWRs for each material.

2.4 CMFD-based Adjoint Capability

The fundamental mode adjoint flux has been widely used in reactor physics field to estimate the perturbed reactivity without solving eigenvalue problem of the perturbed system. It is also used to estimate the core kinetics parameter such as effective delayed neutron fraction. Therefore, adjoint flux is useful tool to estimate dynamic reactivity and kinetic parameters during the transient calculation. To avoid the complexity of MOC based adjoint capability, CMFD-based adjoint capability was implemented in nTRACER.

The standard CMFD linear system can be written as:

$$\mathbf{M}\Phi = \frac{1}{k_{eff}}\mathbf{F}\Phi, \tag{2.46}$$

where $\mathbf{M}$ denotes the migration matrix which involves leakage, absorption, and scattering to other energy groups, and $\mathbf{F}$ denotes the total fission production matrix. The CMFD adjoint equation can be obtained by the transpose of matrices in Eq. (2.46) as:

$$\mathbf{M}^*\Phi^* = \frac{1}{k^*_{eff}}\mathbf{F}^*\Phi^*, \tag{2.47}$$

where Φ^* is the adjoint flux vector. In nTRACER steady-state calculation, the matrix $\mathbf{M}$ is explicitly constructed and stored in a Compressed Sparse Row (CSR) format. Therefore, it can be directly transposed for adjoint flux calculation. On the other hand, the matrix $\mathbf{F}$ is stored as decomposed into two matrices, fission spectra and fission cross section:

$$\mathbf{F} = \boldsymbol{\chi}\boldsymbol{\nu}\boldsymbol{\Sigma}_f. \tag{2.48}$$

By transpose both divided matrices and switch the order of multiplying, the transposed matrix F^* can also be simply obtained:

$$\mathbf{F}^* = \boldsymbol{\nu}\boldsymbol{\Sigma}^*_f\boldsymbol{\chi}^*. \tag{2.49}$$

The power iteration which is used to solve the CMFD eigenvalue problem in steady-state calculation can also be used to solve Eq. (2.47). The dynamic

reactivity and essential kinetic parameters can be obtained during the transient calculation through weighting with initial adjoint flux. When define the factor $F(t)$ as the following:

$$F(t) = \left\langle \Phi_0^*, \mathbf{F}(t)\Phi(t) \right\rangle, \tag{2.50}$$

the dynamic reactivity and kinetic parameters as a function of time can be written as:

$$\rho(t) = \frac{\left\langle \Phi_0^*, \left[\mathbf{F}(t) - \mathbf{M}\right] \Phi(t) \right\rangle}{F(t)}, \tag{2.51}$$

$$\beta(t) = \frac{\left\langle \Phi_0^*, \mathbf{F}_d(t)\Phi(t) \right\rangle}{F(t)}, \tag{2.52}$$

$$\Lambda(t) = \frac{\left\langle \Phi_0^*, \frac{1}{v(t)}(t)\Phi(t) \right\rangle}{F(t)}, \tag{2.53}$$

$$\xi_j(t) = \frac{\left\langle \Phi_0^*, \boldsymbol{\chi}_{dj} C_j(t) \right\rangle}{F(0)}, \tag{2.54}$$

where

$\rho(t)$ = dynamic reactivity,

$\beta(t)$ = effective delayed neutron fraction,

$\Gamma(t)$ = prompt neutron lifetime,

$\xi_j(t)$ = adjoint-flux weighted precursor density for delayed group j.

2.5 Approximate Flux Weighting Method

In 3D transient calculation involving movement of control rod, the partial insertion of control rod in a specific node can be induced. Fig. 2.4 shows a schematic configuration of the node involving partial insertion of control rod. In 2D/1D MOC method, it is not feasible to change the axial mesh structure adaptively according to the location of control rod tip that changes frequently. Therefore, the partial rodded node is treated as a homogenized node. However, if the simple volume weighting method is used for homogenization, significant error is induced in the control rod worth. This issue is so-called control rod cusping effect problem.

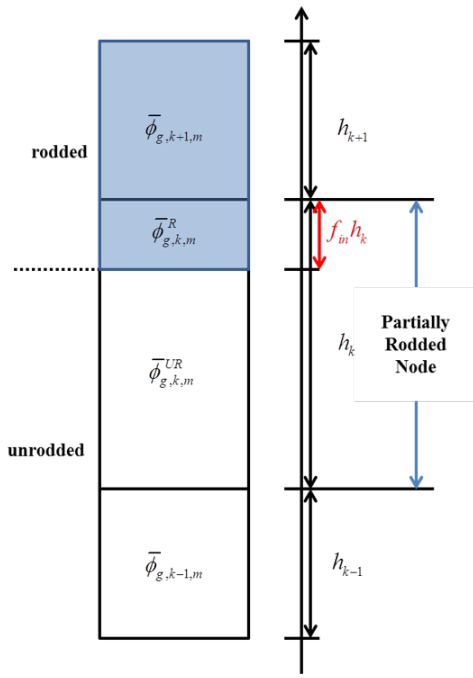


Figure 2.4 Configuration of partial rodded node

The approximate flux weighting (AFW) method is employed to mitigate the cusping effect in the problem involving partial rodded nodes (Gehin 1992).

This method approximate the fluxes of rodded region and unrodded region of partial rodded node as a volume weighted flux using the flux using the fluxes of the neighbor nodes in axial direction. In the core where the control rod is inserted from the top, the flux of rodded region, $\bar{\phi}^R$, and the flux of unrodded region, $\bar{\phi}^{UR}$, are defined as:

$$\bar{\phi}^R_{g,k,m} = \frac{fh_k\bar{\phi}_{g,k,m} + h_{k+1}\bar{\phi}_{g,k+1,m}}{fh_k + h_{k+1}}, \tag{2.55}$$

$$\bar{\phi}^{UR}_{g,k,m} = \frac{(1-f)h_k\bar{\phi}_{g,k,m} + h_{k-1}\bar{\phi}_{g,k-1,m}}{(1-f)h_k + h_{k-1}}, \tag{2.56}$$

where k is index of plane which contains a partial rodded node, m is a radial index of partial rodded node, f is a fraction of rodded region, h is a height of planes. Using the approximate fluxes, the homogenized XS for partially rodded node is obtained as:

$$\bar{\Sigma}_{g,k,m} = \frac{\Sigma^R_{g,k,m} f\bar{\phi}^R_{g,k,m} + \Sigma^{UR}_{g,k,m}(1-f)\bar{\phi}^{UR}_{g,k,m}}{f\bar{\phi}^R_{g,k,m} + (1-f)\bar{\phi}^{UR}_{g,k,m}}. \tag{2.57}$$

3 The Multi-level Method

In this chapter, the multi-level method is detailed that improve the efficiency of transport transient calculation method explained in chapter 2. There are several neutronics models that describes a target system in direct whole core calculation solution of nTRACER: 2D planar MOC, 1D axial MOC, and 3D CMFD. These models have different computational costs and fidelity, and can be classified into computationally expensive high-fidelity model and low-fidelity model with cheaper computational cost. 2D planar MOC corresponds to the former and 3D CMFD kernel, which is composed of 3D CMFD and 1D axial MOC, corresponds to the latter. Two models are combined by outer iteration. In outer iteration, neutron flux level of each pin-cell is determined by 3D CMFD kernel, while intra-cell neutron flux distribution is determined by 2D planar MOC.

In transient situation, intra-cell neutron flux distribution changes slowly compared to flux level of pin-cell. Therefore, using different time scales for each model can be an efficient option. Though the truncation error of the 2D MOC increases as the time-scale becomes coarser, but may be smaller than the truncation error of the 3D CMFD kernel. So using appropriate time step size of 2D MOC calculation can reduce the heavy calculation burden from ray tracing while maintaining accuracy.

Furthermore, the neutron flux level of pin-cell can be factorized into slow varying 'shape' and rapid varying 'amplitude' for more efficient solution. Several methods utilizing this concept of factorization have been developed. Among these methods, the quasi-static method is one of the most popular methods. In the quasi-static method, the amplitude represents the overall amplitude of the neutron flux of the whole-core problem. This amplitude can be calculated using the exact point kinetics equation (PKE) which is derived through integration of time dependent neutron balance equation with known shape distribution. Quasi-static methods can be also categorized into several methods depending on how the amplitude of CMFD solution is determined or coupling method between CMFD and PKE. Improved quasi-static method (IQS), which resolves the nonlinear coupling between PKE and CMFD in an iterative manner, is implemented in nTRACER due to its accuracy in T/H coupled simulation.

As a result of applying neutron flux factorization, 3-level multi-level solution structure which is composed of 2D MOC, 3D CMFD and PKE is constructed. Fig. 3.1 shows the different time scales of multi-level method.

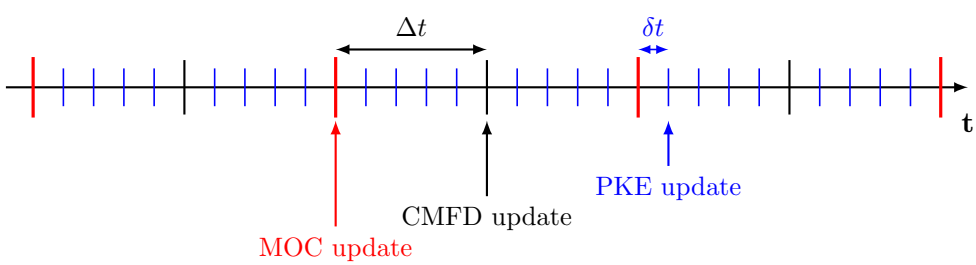


Figure 3.1 Schematic illustration of time scales in multi-level method

In the following sections, the intermittent transport update scheme is explained first. Then the neutron flux factorization methods including IQS is detailed. The examination of flux factorization methods is followed, and the

delayed neutron precursor treatment combined with IQS is explained in final section.

3.1 Intermittent Transport Update

The concept of intermittent transport update was first employed by DeCART (Cho et al. 2005). This scheme is employed to alleviate the heavy computational burden of transport update without significant loss of accuracy. There are two transport calculations that have heavy computational burden: 2D planar MOC and SGFSP. In spite of their heavy computational burden, the variables determined as their solution vary slowly. The intra-cell flux shape and radial current correction factor determined by 2D planar MOC, and escape cross section determined by SGFSP correspond to these variables. Based on this fact, kind of loose coupling method is used for transport calculations and coarse mesh calculations. For the time steps without transport update, aforementioned variables remain constant from the latest time step with transport update. In this way, iteration which can be too expensive for direct whole core calculation can be avoided. To prevent the significant error from this loose coupling, the effect of transport update should be expected precisely, and transport update should be done when the expected effect of transport update is large enough. Detailed description about the monitoring strategy for the effect of transport update will be presented in Section 4.1.

3.2 Neutron Flux Factorization Methods

3.2.1 Improved Quasi-Static Method

The improved quasi-static method (IQS) is based on a factorization of neutron flux into point-wise amplitude and shape as:

$$\phi_g^i(t) = p(t)\tilde{\phi}_g^i(t), \tag{3.1}$$

where p is the amplitude, $\tilde{\phi}$ is the shape, and i is the index of the computational Mesh. The amplitude represents overall amplitude changes of neutron flux, and it is only dependent on time. The shape still has temporal dependence, but it has comparatively smaller time variation than the neutron flux. Since the factorization in Eq. (3.1) is not an approximation, the constraint condition is required for unique factorization. The normalization condition weighted with initial adjoint flux Φ_0^* is used as the constraint condition:

$$\left\langle \Phi_0^*, \frac{1}{v(t)}\tilde{\Phi}(t) \right\rangle = K_0, \tag{3.2}$$

where the initial value $K_0 = \left\langle \Phi_0^*, \frac{1}{v(t)}\tilde{\Phi}_0 \right\rangle$.

With the factorization Eq. (3.1), the derivative of the neutron flux can be represented as:

$$\frac{\partial \Phi_g^i(t)}{\partial t} = \tilde{\Phi}_g^i(t)\frac{dp(t)}{dt} + p(t)\frac{\partial \tilde{\Phi}_g^i(t)}{\partial t}. \tag{3.3}$$

Applying Eq. (3.3) into the time-dependent CMFD problem in Eq. (2.26) yields:

$$\left(\mathbf{R} + \mathbf{T} - \mathbf{S} - \mathbf{F} + \tilde{\mathbf{F}}_\mathbf{d}\right)\tilde{\Phi}^n = \mathbf{T}\tilde{\Phi}^{n-1} + \frac{1}{p^n}\left[\tilde{Q}_d^i - \frac{1}{v}\frac{dp^n}{dt}\tilde{\Phi}^n\right]. \tag{3.4}$$

The modified CMFD linear system can be obtained by multiplying the amplitude at n^{th} time step p^n to Eq. (3.4), inserting Eq. (2.26), and moving all the terms involving unknowns into the left side:

$$\left(\mathbf{R}+\mathbf{T}-\mathbf{S}-\mathbf{F}+\tilde{\mathbf{F}}_{\mathbf{d}}+\frac{1}{\mathbf{v}}\frac{dp^n}{dt}\right)\Phi^n=\frac{p^n}{p^{n-1}}\mathbf{T}\Phi^{n-1}+\tilde{Q}_d^n. \tag{3.5}$$

With knowledge of the amplitude p^n and its temporal derivative term, the linear system in Eq. (3.5) can be directly solved for pin-wise neutron fluxes by Krylov linear system solver such as preconditioned BiCGSTAB.

Integrating Eq. (2.19) and Eq. (2.24) over space and energy with weighting function Φ_0^* yields point kinetics equations:

$$\frac{dp(t)}{dt}=\frac{\rho(t)-\beta(t)}{\Lambda(t)}p(t)+\frac{1}{\Lambda}\sum_{j=1}^{n_{prec}}\lambda_j\xi_j(t), \tag{3.6}$$

$$\frac{d\xi_j(t)}{dt}=-\lambda_j\xi_j(t)+\frac{F(t)}{F_0}\beta_j(t)p(t),\quad j=1,\ldots,n_{prec}, \tag{3.7}$$

where the point kinetics parameters $\rho(t)$, $\beta(t)$, $\Lambda(t)$ and $\xi_j(t)$ are calculated using the definitions in Eq.(2.51) through Eq. (2.54). The amplitude can be evaluated by integrating Eq. (3.6) over time with knowledge of point kinetics parameters.

Since integration of PKE equation doesn't require heavy computational cost, temporal discretization can be done with micro time step δt which must be much smaller than macro time step Δt used for discretization of CMFD equation. However, point kinetics parameters are dependent to the shape. Therefore, estimation of point kinetics parameters should be accompanied

by accurate estimation of shape. As a result, the modified CMFD equation in Eq. (3.5) and the PKE in EQ. (3.6) are nonlinear coupled. IQS directly solve this nonlinear coupled system. Nonlinear systems should be solved with iterative methods. Though there are some complicated nonlinear iterative methods based on Newton's method, only fixed point iteration is considered in this work. Algorithm 1 shows typical algorithm of IQS involving fixed point iteration.

Algorithm 1 IQS algorithm

1: **while** $e > \epsilon$ **do** ▷ outer iteration
2: Evaluate $\tilde{\Phi}_i$ by solving Eq. (3.5) with Δt ▷ shape update
3: Generate PKE parameters in Eqs. (2.51)-(2.54)
4: Update p_i by solving Eq. (3.6) with δt ▷ amplitude update
5: Get corrected Φ_i with updated p_i
6: **end while**
7: $t = t + \Delta t$

Note that the shape is determined by solving Eq. (3.5). But the solution calculated by solving Eq. (3.5) is the neutron flux, not a shape. There are two possible ways to determine the shape from the neutron shape. The first one factorizes calculated neutron flux using known p^n which is calculated as previous PKE solution. It is based on a rationale that Eq. (3.5) is solved for the shape using given p^n. However, the shape determined in this way may not satisfy the uniqueness condition of flux factorization in Eq. (3.2) until the solution converges enough. On the contrary, the second one uses Eq. (3.2) to determine the shape as:

$$\tilde{\Phi} = \frac{K_0}{\left\langle \Phi_0^*, \frac{1}{v(t)}\Phi(t) \right\rangle}\Phi. \tag{3.8}$$

The former is referred as 'IQS' and the latter is referred as 'IQS-rescale' later on.

3.2.2 Predictor Corrector Quasi-Static Method

The predictor-corrector quasi-static method (PCQS) also based on the separation of neutron flux into point-wise amplitude and shape in Eq. (3.1). However, unlike IQS, the shape is not estimated directly. Rather, the original CMFD problem in Eq. (2.26) is solved directly in PCQS. The neutron flux solution obtained as the solution of CMFD problem is considered as the predictor. The constraint condition in Eq. (3.2) is assumed to be preserved, the shape and amplitude are extracted from predictor solution using the inverse relation as the following equations:

$$p_p^n = \frac{1}{K_0} \left\langle \Phi_0^*, \frac{1}{v} \Phi_p^n \right\rangle \tag{3.9}$$

$$\tilde{\Phi}^n = \frac{K_0}{\left\langle \Phi_0^*, \frac{1}{v} \Phi_p^n \right\rangle} \Phi_p^n \tag{3.10}$$

where the subscript p indicates the predictor. The reason that the predictor indicator is omitted for the shape in Eq. (3.10) is that it is assumed to be sufficiently accurate. The algorithm of PCQS can be presented as Algorithm 2.

3.2.3 Exponential Transform Method

The exponential transform method (ET) was designed to enhance accuracy of the conventional theta method. This method anticipates exponential variation

Algorithm 2 PCQS algorithm

1: Predict Φ_p by solving Eq. (2.26) with Δt ▷ predictor
2: Factorize Φ_p into p_p and $\tilde{\Phi}$ based on Eq.(3.9) and Eq.(3.10)
3: Generate PKE parameters in Eqs. (2.51)-(2.54)
4: Update corrected amplitude p_c by solving Eq. (3.6) with δt ▷ corrector
5: Get corrected $\Phi_i = p_c \tilde{\Phi}$
6: $t = t + \Delta t$

of the neutron flux since typical transient event in PWR such as the hot-zero-power control rod ejection shows exponential variation of power. Similar with quasi-static method, the regional flux is factorized into two components in ET method:

$$\phi_g^i(t) = e^{\alpha_g^i t} \tilde{\phi}_g^i(t). \tag{3.11}$$

The exponential function $e^{\alpha_g^i t}$ can be seen as the amplitude function in Eq. (3.1), since it can catch rapid variation of flux variable and make the transformed flux function $\tilde{\phi}_g^i(t)$ change comparatively slowly. By introducing factorization in Eq. (3.11) into EQ. (2.26) and moving the unknown terms into left-hand side, modified CMFD problem similar with Eq. (3.5) is constructed as:

$$\left(\mathbf{R} + \mathbf{T} - \mathbf{S} - \mathbf{F} + \tilde{\mathbf{F}}_\mathbf{d} + \frac{\boldsymbol{\alpha}}{\mathbf{v}}\right) \Phi^n = \mathbf{e}^{\alpha \Delta t_n} \mathbf{T} \Phi^{n-1} + \tilde{Q}_d^n, \tag{3.12}$$

where $\boldsymbol{\alpha}$ and $\mathbf{e}^{\alpha \Delta t_n}$ represents the diagonal matrix that contains α and $e^{\alpha \Delta t_n}$, respectively. Similar with IQS, the transformed flux can be evaluated by solving Eq. (3.12). The inverse period α_g^i can be determined in various way, it is determined in nTRACER with regional neutron fluxes at the current time step and the previous time step and converged during the fixed point iteration. The difference of this method from IQS is PKE equations and uses the exponential function which has regional and energy dependency rather than the amplitude

function which has only time dependency. The main focus of this method is to make the transformed flux function less variable over time for small truncation error while IQS focused to find out the unique factorization of shape and amplitude on exact perturbation theory.

3.3 Delayed Neutron Precursor Treatment

As described in Chapter 2, the delayed neutron precursor density is analytically calculated using approximation of temporal variation of fission source as a polynomial over time. The analytic solution can be expressed as:

$$C_j^{i,n} = C_j^{i,n-1} e^{-\lambda_j \Delta t_n} + \int_{t_{n-1}}^{t_n} \beta_j^i \psi^i(t') e^{-\lambda_j (t_n - t')} dt'. \tag{3.13}$$

Quadratic approximation of temporal variation of fission source is preferred than linear approximation, in spite of heavier memory requirement and longer computing time. Since the local error from quadratic approximation is proportional to cubic of the time step size, it is higher order technique than BE method which is used to calculate the shape in 3D CMFD kernel. The reason for applying higher order technique is that the variation of cross section is also related to the variation fission source, so it is difficult to capture the temporal change of the fission source with the technique that has same or lower order.

Rather than using higher order technique for fission source approximation, the variation of flux amplitude which is calculated for micro time-scale through PKE calculation can be used for accurate calculation of fission source variation. The fission source can be factorized using the amplitude as:

$$\psi^i(t) = p(t)\tilde{\psi}^i(t). \tag{3.14}$$

The factorized fission source $\tilde{\psi}$ varies slowly so that it is possible to use linear approximation to capture the temporal variation of it. For the time interval $[t_{n-1}, t_n]$, $\tilde{\psi}^i$ is linearly interpolated as:

$$\tilde{\psi}^i(t) = -\frac{t - t_n}{\Delta t_n}\tilde{\psi}^i_{n-1} + \frac{t - t_{n-1}}{\Delta t_n}\tilde{\psi}^i_n. \tag{3.15}$$

By plugging Eq. (3.15), the integral term in the right hand side of Eq. (3.13) can be expressed as:

$$\begin{aligned}
&\int_{t_{n-1}}^{t_n} \beta^i_j(t')\psi^i(t')e^{-\lambda_j(t_n - t')}dt' \\
&\qquad = -\frac{\beta^i_j\tilde{\psi}^i_{n-1}}{\Delta t_n}\int_{t_{n-1}}^{t_n} p(t')(t' - t_n)e^{-\lambda_j(t_n - t')}dt' \\
&\qquad\quad + \frac{\beta^i_j\tilde{\psi}^i_n}{\Delta t_n}\int_{t_{n-1}}^{t_n} p(t')(t' - t_{n-1})e^{-\lambda_j(t_n - t')}dt'.
\end{aligned} \tag{3.16}$$

The integral terms in Eq. (3.16) can be calculated by trapezoidal rule using the amplitude results calculated by PKE calculation at micro time steps. By using the calculated integral values as coefficients, the precursor density can be expressed as linear combination of the latest two fission sources:

$$C^{i,n}_j = C^{i,n-1}_j e^{-\lambda_j \Delta t_n} + \omega^i_{j,n-1}\tilde{\psi}^i_{n-1} + \omega^i_{j,n}\tilde{\psi}^i_n, \tag{3.17}$$

where

$$\begin{aligned}
\omega^i_{j,n-1} &= -\frac{\beta^i_j}{\Delta t_n}\int_{t_{n-1}}^{t_n} p(t')(t' - t_n)e^{-\lambda_j(t_n - t')}dt', \\
\omega^i_{j,n} &= \frac{\beta^i_j}{\Delta t_n}\int_{t_{n-1}}^{t_n} p(t')(t' - t_{n-1})e^{-\lambda_j(t_n - t')}dt'.
\end{aligned}$$

3.4 Examination of Flux Factorization Methods

In this section, different flux factorization methods are evaluated. Two different problems are used for this evaluation. One is C5G7-TD problem which is pure neutronics problem, and the other is SPERT III E-core problem involving super-prompt RIA in which T/H feedback has a big influence on the results.

3.4.1 C5G7-TD Results

The characteristics and effectiveness of flux factorization methods as pure neutronics solver are investigated with C5G7-TD benchmark problem (Boyarinov et al. 2016). This tests calculation capability of modern transport codes for transient problems without spatial homogenization and T/H feedback. This benchmark problem consists of six problem sets each of which involves transient events such as control rod movement or moderator density change. Each problem sets consists of sub-problems defined with different parameters. The radial and axial configuration of C5G7-TD core is shown in Fig. 3.2. The radial configuration of the problem is the same as that of well-known C5G7 MOX benchmark (Lewis et al. 2001), but there are minor differences in the axial geometry for modeling control rod insertion from the top. Since this benchmark is designed to verify transport codes, the pin-cell has two region geometry without homogenization. To capture the sub-pin level neutron flux variation, the fuel and moderator regions are divided into 4 annular regions with 8 azimuthal sectors respectively. The square pin-cell used for moderator reflector modeling is divided into 7×7 array of FSRs. The height of the fuel assembly is 128.52 cm and there are top and bottom reflector whose height are 21.42 cm. For axial modeling, the core is divided into 32 planes with equal

height of 5.355 cm. The active fuel is divided into 24 planes, and top and bottom reflector is divided into 4 planes each. The ray sets with 0.05 cm spacing which generated for 16 azimuthal angles and 4 polar angles per octant of the solid angle sphere were used for planar MOC calculation.

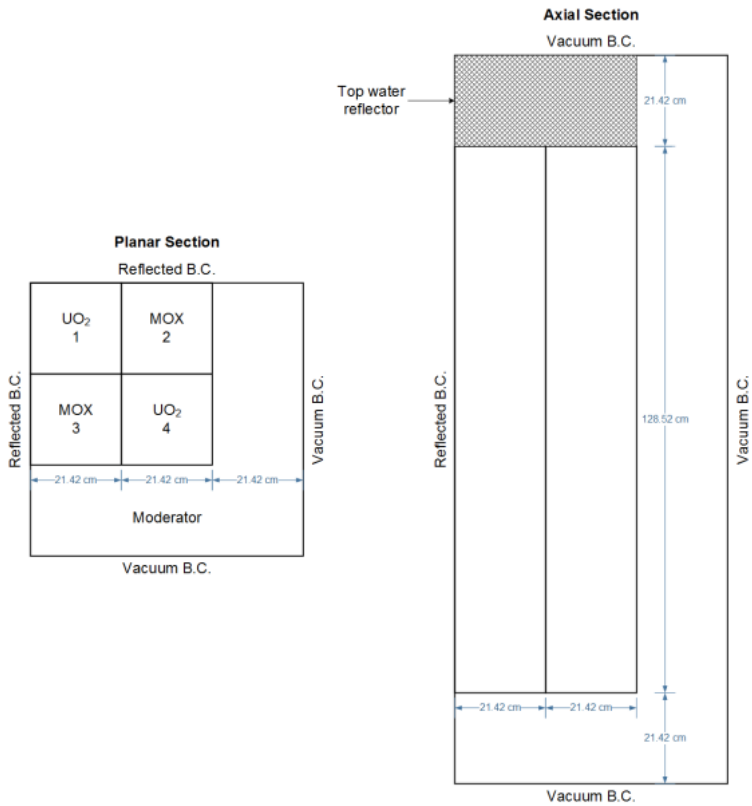


Figure 3.2 Core configuration of C5G7-TD

Out of various problems, TD2-1 was used as for investigation. It is a 2D problem which approximates the control rod transient as a ramp change of the material composition. Six different cases including reference solution were performed. The details of these cases are listed in Table 3.1. The reference solution was generated by the Backward Euler calculation with a fine time step size of 1 ms. The other cases used a macro time step size of 50 ms. For the

cases using quasi-static methods, PKE was solved with a micro time step size of 1 ms. The performance of each case is evaluated by comparing the results with the reference solution.

Table 3.1 Options for different cases in SPERT III E-core calculation

NAME	Δt	δt	Remark
Ref.	1 ms	-	Solution reference (Backward Euler)
BE	50ms	-	Backward Euler
ET		-	Exponential Transform
PCQS		1 ms	Predictor-corrector quasi-static
IQS			Improved quasi-static
IQS-rescale			Improved quasi-static with rescaling

The results of examination of flux factorization methods using C5G7-TD problem are shown in Table 3.2. It summarizes the relative core power error at different time points. When compared to BE solution, four flux factorization methods improves the accuracy. The error reduction depends on time point and factorization method. At the time points involving continuous movement of control rod (0.5 and 1.5 s), ET achieves error reduction of about 20 % and three quasi-static methods (PCQS, IQS, and IQS-rescale) achieve error reduction of up to 30∼35 %. At 1 s, when the control rod is stopped and the withdrawal begins, error reduction is about 20 % for all cases. The error reduction is about 10 % at 4 s, 2 seconds after the cross section perturbation. Though the magnitude of relative power error is smaller than 0.05 % at other time points, it is large at 2 s. Especially, large error occurs when using BE that the error is larger than 0.5 % at 2 s. Flux factorization method reduces the error significantly at 2 s. Error reduction is about 40 % when using ET, and it is up to 80∼90 % when using quasi-static methods.

The average number of outer iteration and inner iteration per time step are also shown in Table 3.2. The outer iteration represents the entire procedure of the transient calculation which contains the transport solution update. The inner iteration means the number of iteration required by the linear system solver (BiCGSTAB) in CMFD solution. When see the average number of outer iteration, all methods showed similar values except IQS showed a slightly higher value. The inner iteration number shows relatively large discrepancy between methods. Since PCQS just corrects the calculated flux from BE, it does not affect the number of iteration. Though ET updates the exponents using fixed point iterations, it also shows similar number of iterations with BE. Two IQS methods results relatively large discrepancy of iteration number with other method. IQS requires more iterations for both inner and outer iterations. On the contrary, the number of required iterations are reduced when using IQS-rescale method.

Figure 3.3 shows the behavior of relative core power error over time with three different quasi-static methods. The error behavior of PCQS and IQS-rescale resembles each other, because they determine the neutron flux shape in same way which uses the uniqueness condition of flux factorization.

3.4.2 SPERT III E-core Results

SPERT III E-core is used for evaluation of flux factorization methods coupled with T/H calculation. The detailed description of the core modeling can be found in Appendix A. Imaginary super-prompt RIA starting from HZP condition in SPERT III E-core up to 0.3 s is calculated with different options. Similar to C5G7-TD calculation, six different cases listed in Table 3.1 are also performed for SPERT III E-core. The reference solution is generated by the

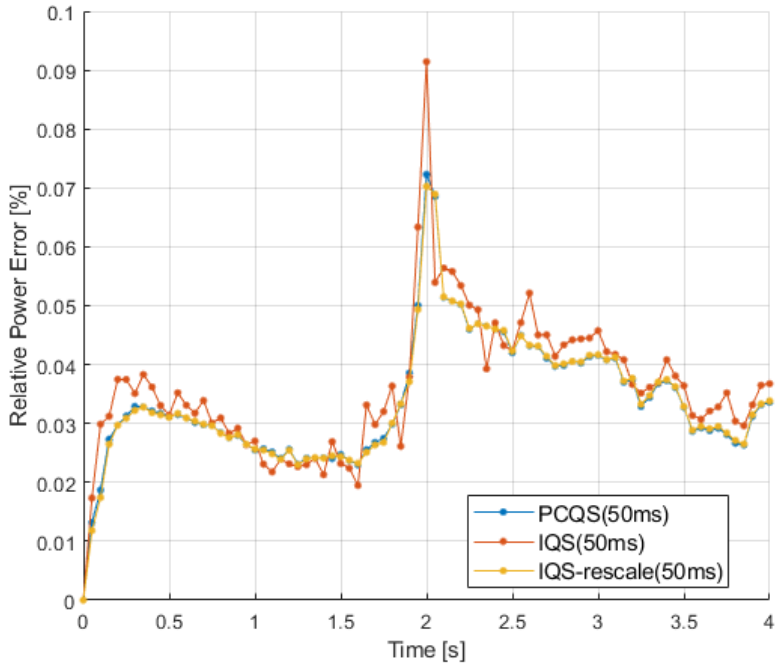


Figure 3.3 Comparison of power relative error

Backward Euler calculation with a fine time step size of 1 ms. The other cases use a macro time step size of 5 ms. For the cases using quasi-static methods, PKE is solved with a micro time step size of 1 ms. The performance of each case is evaluated by comparing the results with the reference solution.

Fig. 3.4 shows the calculated power histories and summarized results are shown in Table 3.3. Cases using quasi-static methods(PCQS, IQS, IQS-rescale) shows better accuracy than other methods. The power histories of quasi-static methods are well matched with the reference, while other methods shows large discrepancy. BE case shows forward shifted solution. It has 34 ms earlier and slightly higher power peak when compared with the reference solution. ET case generated solution that is less shifted than Backward Euler method, but peak

Table 3.2 Summary of examination of flux factorization methods using C5G7 TD2-1 problem

		BE	ET	PCQS	IQS	IQS -rescale
Relative power error (%)	0.5 s	0.044	0.038	0.031	0.031	0.031
	1.0 s	0.033	0.027	0.026	0.027	0.026
	1.5 s	0.037	0.028	0.024	0.023	0.024
	2.0 s	0.553	0.332	0.072	0.091	0.073
	4.0 s	0.039	0.034	0.034	0.037	0.034
Avg. outer iteration		2.49	2.50	2.50	2.69	2.49
Avg. inner iteration		44.6	44.0	44.2	58.3	33.3
Time (s)		93.9	96.1	100.4	108.6	98.5

power level is 60 MW lower than the reference. Table 3.3 indicates quasi-static methods can reduce the error of integrated power by 93 %.

A close up view of power history of quasi-static methods at power peak where large error occurs is shown in Fig. 3.5. Before the power peak all quasi-static methods showed similar behavior. However, at the power peak, PCQS solution is underestimated about 12 MW than other methods. Only the predicted solution, not PKE solution, is tightly coupled with T/H solution in PCQS. Therefore, the resulted T/H solution would be not accurate enough if the discrepancy between the predictor and the corrector are significant. Especially this discrepancy is large at the power peak. This phenomenon is naturally resolved with IQS methods which correct the neutron flux during the nonlinear iteration.

The computing time of each case is also summarized in Table 3.3. The reference case takes 11 hours 50 minutes. BE case, which uses same method as

Table 3.3 Summary of examination of flux factorization methods for RIA in SPERT III E-core

	Ref.	BE	ET	PCQS	IQS	IQS -rescale
Max. Power (MW)	531	536	471	515	527	527
Diff. (MW)	-	5	-60	-16	-4	-4
Max. time (s)	0.209	0.175	0.215	0.205	0.210	0.210
Diff (s)	-	-0.034	0.006	-0.004	0.001	0.001
Integrated Power (MWs)	28.1	29.5	26.6	28.2	28.2	28.2
Diff. (MWs)	-	1.4	-1.5	0.1	0.1	0.1
Computing time (min)	710	179	171	166	287	181

reference case but 5 times larger time step size, requires about 3 hours. The computing time reduction was about 75 %. Compared to other and BE cases, there are no significant difference caused from using other method, except for IQS case. IQS requires about 2 times more outer iterations for each time step.

Performance examination results above shows the efficiency of quasi-static methods compared to legacy methods. For pure neutronics problem, there is no significant difference between quasi-static methods(IQS, IQS-rescale, PCQS). However, in the multi-physics solution where neutronics is coupled with T/H solution, IQS-rescale seems an ideal flux factorization method when judging both efficiency and accuracy. Therefore, IQS-rescale is used as a default option.

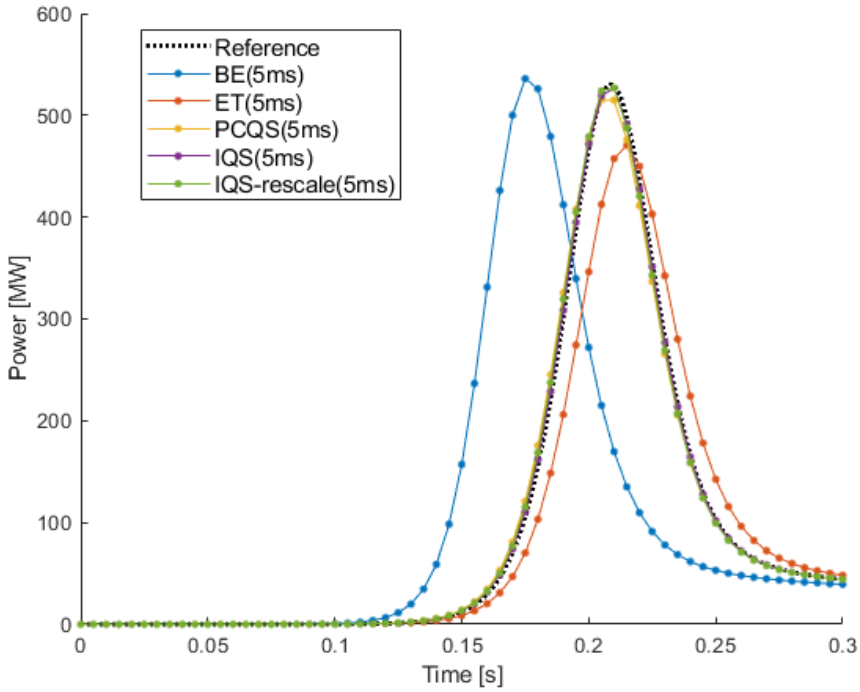


Figure 3.4 Comparison of power history

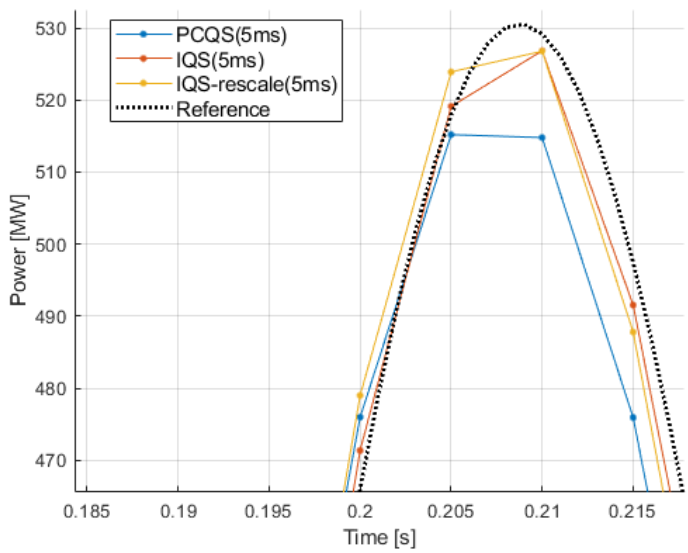


Figure 3.5 Close up view of power pulse peak

4 Adaptive time step Control

4.1 Conditional Transport Update

As described in Chapter 3, by using intermittent transport update, the overwhelming computational burden and memory demands caused by 2D MOC calculation was alleviated while the accuracy is retained. To guarantee the accuracy of intermittent planar MOC calculation scheme, kind of adaptive algorithm that conditionally invoke the MOC update by monitoring the cell homogenized cross section was also used in DeCART(Cho et al. 2005). However, exact definition of the invoking criterion was not documented and rigorous verification of the conditional MOC update scheme was not carried out. The concept of the conditional MOC update scheme was adopted in nTRACER and improved (Jung and Joo 2015). Not only 2D MOC calculation, the subgroup fixed source problem(SGFSP) was also conditionally performed by monitoring the maximum fuel temperature change. The conditional MOC update scheme that monitoring the change of 1 group condensed absorption cross section was devised for nTRACER. The invoking criteria for conditional MOC and SGFSP were determined from the sensitivity study using 3×3 3D assembly rod ejection problem.

The overall calculation flow of conditional transport update scheme is illustrated in Fig. 4.1. For accurate estimation of invoking parameters such as fuel

temperature change or cross section change without posing heavy calculation cost, solution procedures are coordinated properly. The pilot T/H calculation is performed before determining the conditional SGFSP update. Though the pilot T/H calculation uses the power distribution of the previous time step, the results has sufficient accuracy for monitoring the fuel temperature change. After determination of the conditional SGFSP, microscopic cross section of each FXR is determined based on the initial guess of T/H condition. Then the pilot 3D transient CMFD calculation is performed before determining the conditional MOC update. By using the pilot 3D CMFD calculation results, the change of neutron flux spectrum at new time step can be considered.

4.1.1 MOC Invoking Criteria

The conditional SGFSP update scheme monitoring the change of fuel temperature is effective. The escape cross sections are dependent to the core temperature distribution and core composition only, it is straight forward to monitor the temperature distribution. On the other hand, the exact estimation of sub-pin level flux shape change requires a MOC calculation which is not practical. Therefore, 1 group condensed absorption cross section was selected as an approximation. The 1 group absorption cross section of FXR i is calculated as:

$$\Sigma_a^i = \frac{\sum_{g=1}^{G} \Sigma_{a,g}^i \phi_g^i}{\sum_{g=1}^{G} \phi_g^i}. \tag{4.1}$$

This approximation is based on the rationale that the change of 1 group absorption cross section can capture the effect of thermal feedback and change of global flux distribution. Both thermal feedback and global flux distribution

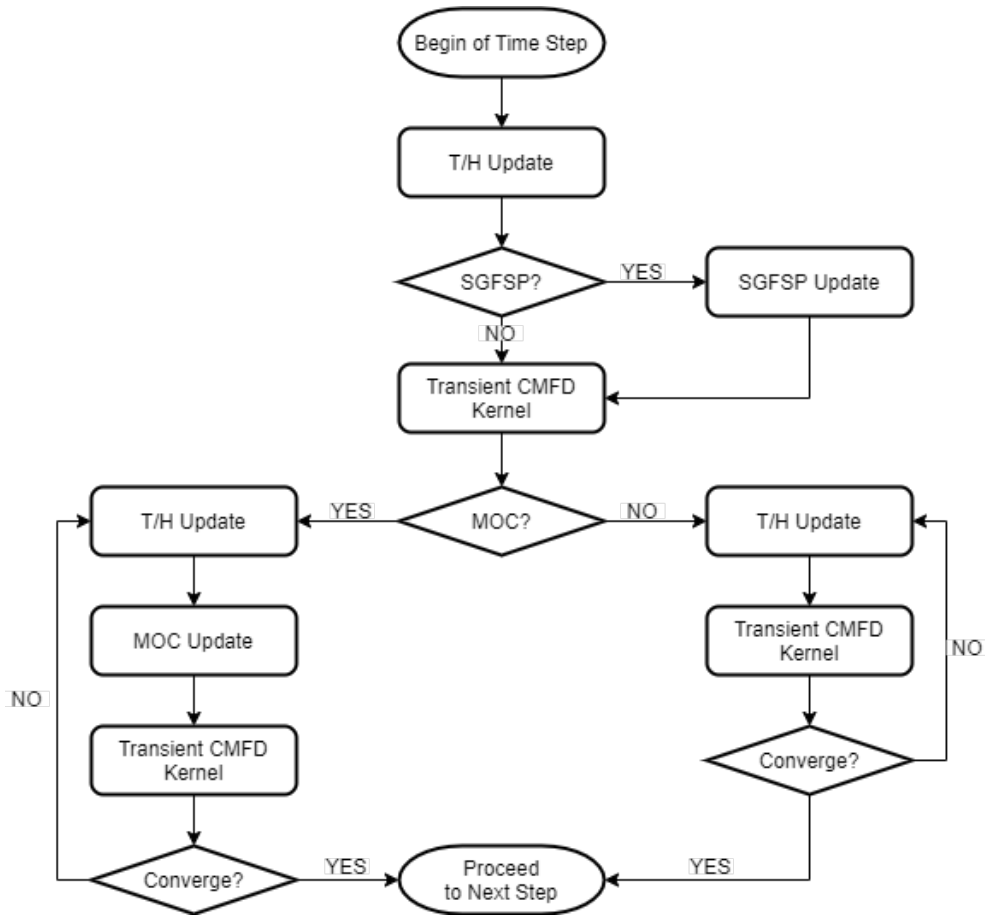


Figure 4.1 Calculation flow of conditional transport update algorithm

affects the sub-pin level flux shape. Certainly there is a correlation between the two variables, 1 group absorption cross section and sub-pin level flux shape, but it is difficult to say that the magnitude of change of each is proportional.

The new conditional MOC update scheme which is based on the estimation of residual norm of sub-pin level shape problem is devised for more precise quantitative estimation of sub-pin level flux shape change. The MOC calculation fine-mesh residual integrated over Ω for FSR k and group g at n^{th} time

step can be defined as:

$$\begin{aligned}\mathbb{F}_{k,g}^{n} = &\left(\frac{1}{v_g^n \Delta t_n} + \frac{1}{v_g^n}\frac{1}{p^n}\frac{dp^n}{dt}\right)\phi_{k,g}^{n} + \Sigma_{t,k,g}\phi_{k,g}^{n} \\ &- Q_{k,g}^{n} + L_{z,k,g}^{n} + L_{r,k,g}^{n} - \frac{p^n}{p^{n-1}}\frac{1}{v_g^n \Delta t_n}\phi_{k,g}^{n-1},\end{aligned} \tag{4.2}$$

where $\mathbb{F}$ is a residual, L_z and L_r is a net leakage in axial and radial direction, respectively. Direct calculation of this residual also requires a ray tracing calculation because radial leakage L_r is cannot be calculated with scalar flux distribution. However, when the solution of the transient fixed source problem involving the MOC update is fully converged, the residual $\mathbb{F}$ can be assumed as zero. Then L_r can be calculated using Eq. (4.2). The fine mesh radial leakage normalized by pin-wise flux level calculated at the latest time step involving the MOC update is assumed to be constant and used for prediction of fine mesh residual for conditional MOC update. The normalized fine mesh residual $\tilde{L}_r$ for FSR k and group g is written as:

$$\begin{aligned}\tilde{L}_{r,k,g}^{n_0} = &-\left(\frac{1}{v_g^{n_0} \Delta t_{n_0}} + \frac{1}{v_g^{n_0}}\frac{1}{p^{n_0}}\frac{dp^{n_0}}{dt}\right)S_{k,g}^{n_0} - \Sigma_{t,k,g}S_{k,g}^{n_0} \\ &+ \tilde{Q}_{k,g}^{n_0} - \tilde{L}_{z,k,g}^{n_0} - \frac{1}{v_g^n \Delta t_n}\frac{\tilde{\phi}_{j,g}^{n_0-1}}{\tilde{\phi}_{j,g}^{n_0}}S_{k,g}^{n_0-1},\end{aligned} \tag{4.3}$$

where $S_{k,g}^{n}$ is a fine-mesh flux shape of FSR k which is in pin j and group g at n^{th} time step and $\tilde{\phi}_{j,g}^{n}$ is a coarse-mesh pin-wise flux shape of pin j and group g at n^{th} time step. The n_0^{th} time step is base time step which is the latest time

step involving the MOC update. The $S^n_{k,g}$ is defined as follows:

$$S^n_{k,g} = \frac{\phi^n_{k,g}}{\phi^n_{j,g}} = \frac{\phi^n_{k,g}}{\frac{\sum_{k\in j} \phi^n_{k,g} V_k}{\sum_{k\in j} V_k}}. \tag{4.4}$$

Then the fine-mesh residual at time step n can be calculated by approximate the radial leakage in Eq. (4.2) using the normalized radial leakage from base time step and updated pin-wise flux level as follows:

$$L^n_{z,k,g} \approx \phi^n_{j,g} \tilde{L}^{n_0}_{r,k,g}. \tag{4.5}$$

4.1.2 Evaluation of Flux Shape Change Estimator

The two MOC update invoking criteria, monitoring 1 group condensed absorption cross section (denoted by XS) and monitoring approximated fine-mesh residual norm (denoted as RES), was evaluated by comparison with the actual change caused by involving MOC update at each time step. To obtain the actual difference, two different calculation was performed for each time step. For a certain time step, transient fixed source problem was solved without MOC update. After the solution is converged, solution is started over again with MOC update and then proceed to next step. The actual change is estimated as an amplitude relative difference between two solutions:

$$\delta p^n = \frac{\|p^n_{CMFD} - p^n_{MOC}\|}{p^n_{MOC}}. \tag{4.6}$$

Note that this amplitude difference can quantify the affect of sub-pin level flux shape change from MOC update. The evaluation was performed with a 5×5

assembly rod ejection problem. This problem consists of asymmetric 5×5 fuel assembly array in which three types of 17x17 pin-cell assemblies are loaded (Hursin et al. 2016). The radial configurations of the core and assembly are illustrated in Fig. 4.2. Three types of fuel assembly have different enrichment. Reflective boundary conditions are used both radial and axial directions. The transient is initiated by ejecting the control rod in the assembly marked in Fig. 4.2 within 0.1 second. The composition of control rod material is adjusted to have realistic core transient behavior. The transient is simulated up to 0.6 s with 5 ms time step size.

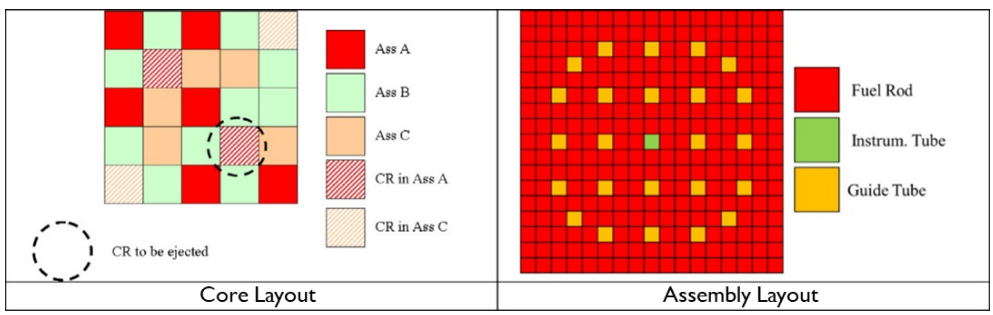


Figure 4.2 Configuration of 5×5 assembly problem

Fig. 4.3 and Fig. 4.4 show the estimated invoking parameters and its comparison with actual change of XS and RES criterion respectively. The affect of sub-pin level flux shape change from MOC update was negligible at the most of the time steps. But it has a significant impact right after the control rod was removed or near the power peak at 0.33 s. AS shown in Fig. 4.3, the XS invoking criterion shows similar behavior with actual change before the power peak, but after the peak power it overestimates the impact of shape change from MOC update. On the other hand, the fine-mesh residual norm shows a similar shape with actual change as shown in Fig. 4.4. These results indicate

that using RES invoking criteria will enhance the efficiency of conditional MOC update scheme.

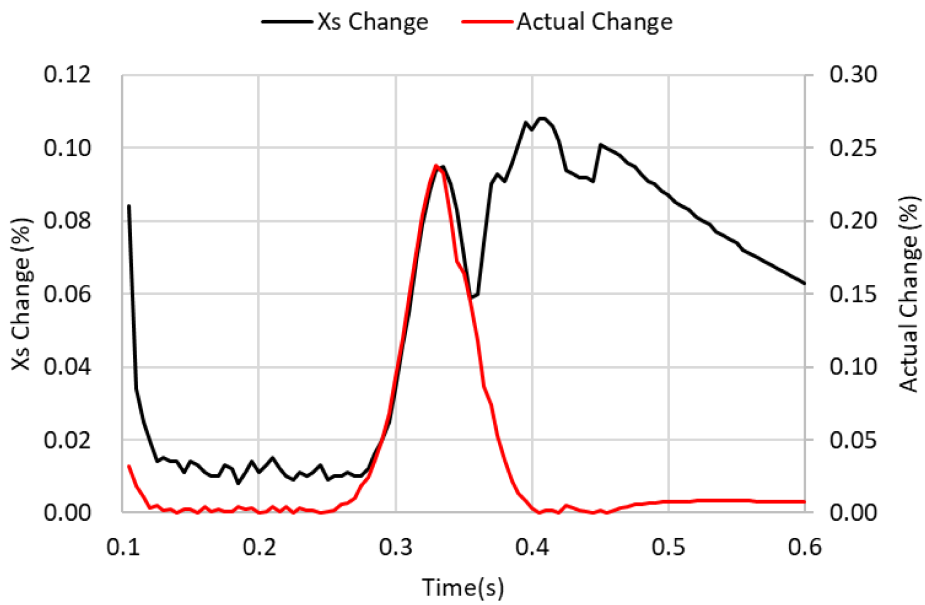


Figure 4.3 Estimated cross section change compared to actual change

The efficiency of RES invoking criteria is validated through the comparison with the reference solution. The reference solution was obtained through the nTRACER run with 5ms time step size without conditional transport update scheme. The XS invoking criteria is also tested to confirm the superiority of RES invoking criteria. The calculations employing conditional transport update used the same time step size with the reference solution, 5 ms. Following values are used as XS and RES invoking criteria: 0.1% for XS, 5×10^{-5} for RES. These values are determined to make both cases involve a similar number of MOC updates. For a same time step, estimated cross section change shows about 20 to 30 times larger than the fine-mesh residual norm.

The examination results are summarized in Table 4.1. As expected two cases have similar level of conditional MOC update. The MOC update is performed

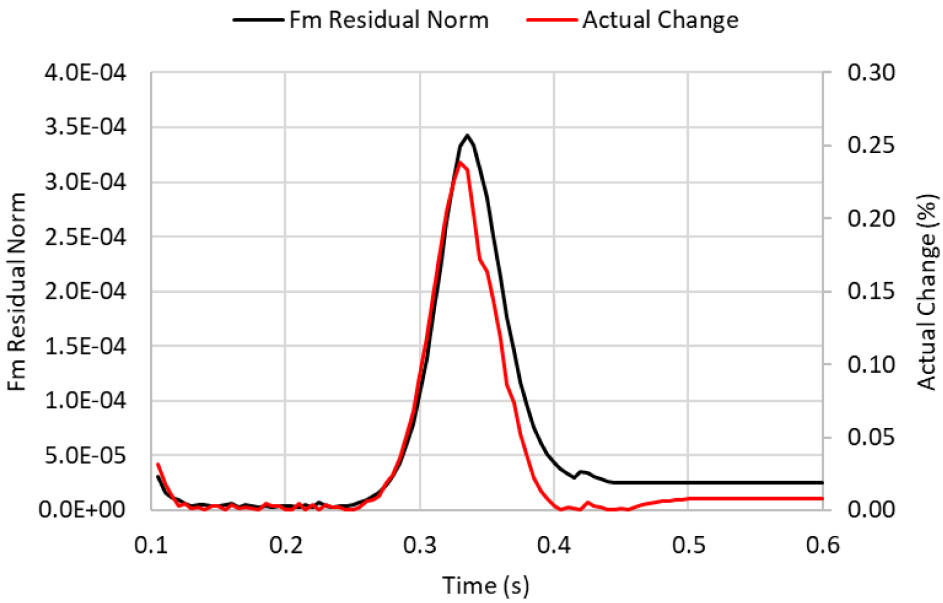


Figure 4.4 Estimated fine-mesh residual norm compared to actual change

for 60 time steps in RES case and for 54 time steps in XS case out of total 120 time steps. As the number of MOC update decreased, total computing time is decreased by about 30%. Fig. 4.5 shows the power history of the reference solution and the power level at the time steps involving MOC update of two cases. Though the number of MOC update is similar, MOC update is performed more frequently during the period from 0.4 to 0.6 s, after power peak, in XS case. On the other hand, RES case performs more MOC update before the power peak at 0.33 s. This tendency is consistent to that analysis results presented above that the impact of MOC update is overestimated after the power peak when using XS invoking criteria. Fig. 4.6 shows the relative power error of two cases. In spite of the similar number of MOC update, XS case shows about 2 times larger relative error than RES case.

Table 4.1 Examination results of conditional MOC invoking criteria

	ref.	XS	RES
# of MOC Update	120	54	60
Computing time (min)	89	61	66
Max. Power Rel. Error (%)	-	0.93	0.20
Released Energy Rel. Error (%)	-	0.40	0.00

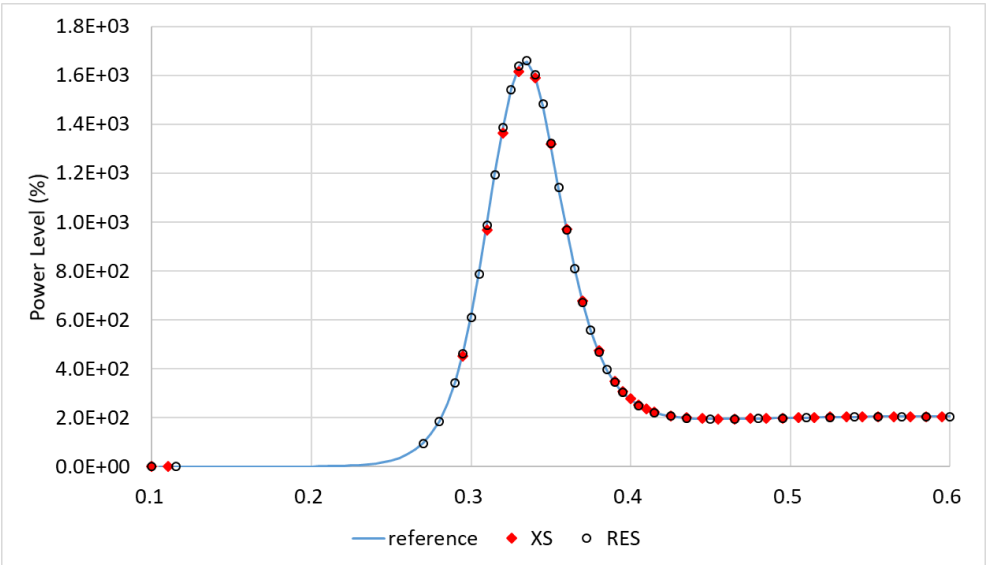


Figure 4.5 Power history comparison between MOC invoking criterion

4.2 Time Step Control of CMFD

4.2.1 Error Analysis of Multi-level Method

The PKE solution works as a main solver in the multi-level method because it determines the overall amplitude of the neutron flux. Therefore, a certain level of accuracy should be maintained for PKE solution during transient calculation. For time interval $[t_{n-1}, t_n]$, the PKE problem in Eq. (3.6) is solved by using

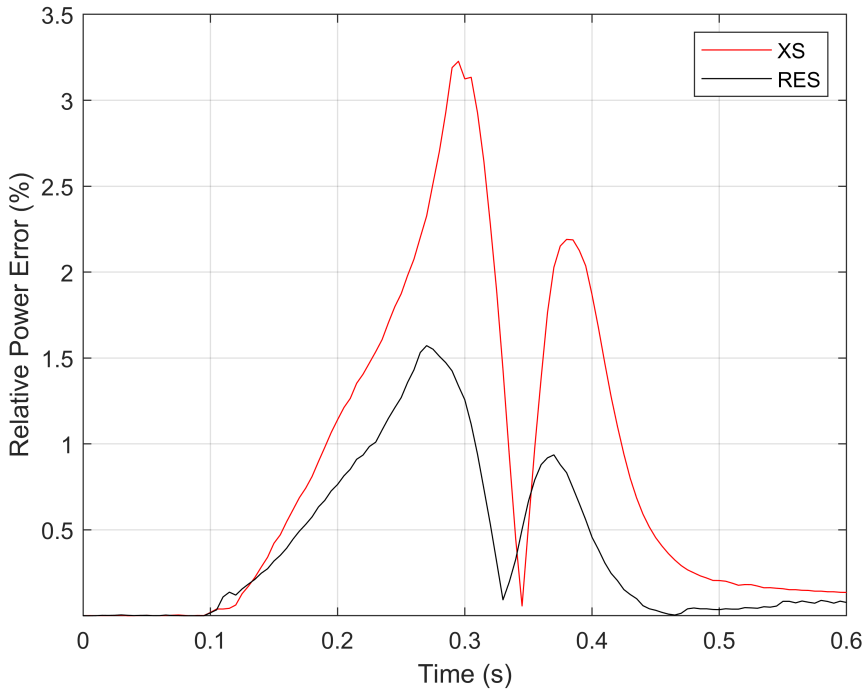


Figure 4.6 Power Error comparison between MOC invoking criterion

Backward Euler method as:

$$p^m = \frac{p^{m-1} + \delta t q_d^m}{1 - \delta t \alpha^m}, \qquad m = 1, \ldots, M \tag{4.7}$$

where

$$\alpha^m = \frac{\rho^m - \beta^m}{\Lambda^m},$$
$$q_d^m = \frac{1}{\Lambda_0} \sum_{j=1}^{6} \lambda_j \xi_j,$$
$$M = \frac{t^n - t^{n-1}}{\delta t} = \frac{\Delta t}{\delta t}.$$

The Backward Euler itself can be completely correct when using small enough micro time step size δt. Fortunately, since the computational cost of PKE calculation is negligible, it is possible to use small enough δt without noticeable increase of computing time. However, error can be occurred by the incorrect α. When considering the definition of α, the error of α has dependence on macro time step size. Therefore, to find appropriate macro time step size, the definition of α error should be derived. The error of α is denoted by ϵ later on, and it is defined as a difference between numerical solution α and exact solution α^*:

$$\epsilon^m = \alpha^m - \alpha^{*,m}. \tag{4.8}$$

The local error of amplitude caused by the incorrect α in macro time interval $[t_{n-1}, t_n]$ can be expressed as a function of ϵ and macro time step size Δt. By introducing Eq. (4.8) into Eq. (4.7), it can be written as:

$$p^m = \frac{p^{m-1} + \delta t q_d^m}{1 - \delta t \left(\alpha^{*,m} + \epsilon^m\right)}, \qquad m = 1, \ldots, M. \tag{4.9}$$

Then the local error of amplitude at intermediate time step m, which is denoted by δp_{loc}^m, is defined as:

$$\begin{aligned} \delta p_{loc}^m =& \frac{p^{m-1} + \delta t q_d^m}{1 - \delta t \left(\alpha^{*,m} + \epsilon^m\right)} - \frac{p^{m-1} + \delta t q_d^m}{1 - \delta t \alpha^{*,m}} \\ =& \frac{\delta t \epsilon^m \left(p^{m-1} + \delta t q_d^m\right)}{\left(1 - \delta t \alpha^{*,m}\right)\left(1 - \delta t \left(\alpha^{*,m} + \epsilon^m\right)\right)}. \end{aligned} \tag{4.10}$$

The global error of amplitude at intermediate time step m, δp_{glob}^m, is defined by a following recursive relation:

$$\delta p_{glob}^m = \frac{\delta p_{glob}^{m-1}}{1 - \delta t \alpha} + \delta p_{loc}^m \tag{4.11}$$

Note that it is assumed that the results at the macro time step t_{n-1} is exact, which indicates that δp^1_{glob} is equal to δp^1_{loc}. Then δp^m_{glob} can be expressed as follows:

$$\delta p^m_{glob} = \sum_{i=1}^{m} \delta p^i_{loc} \prod_{j=1}^{m-i} \frac{1}{1-\delta t \alpha^j}. \tag{4.12}$$

The relative local error of amplitude at n$^{\text{th}}$ macro time step, ε, is equivalent to the accumulated global error over M intermediate time steps:

$$\varepsilon^n = \frac{\delta p^M_{glob}}{p^M} = \sum_{i=1}^{M} \frac{\delta p^i_{loc}}{p^M} \prod_{j=1}^{M-i} \frac{1}{1-\delta t \alpha^j}. \tag{4.13}$$

When assume that the micro time step size is small enough so that $\delta t \alpha \ll 1$ is satisfied, the production term can be bounded as follows:

$$\prod_{j=1}^{M-i} \frac{1}{1-\delta t \alpha^j} \leq \left(\frac{1}{1-\delta \alpha^k} \right)^{M-i} = \exp\left(\frac{M-i}{M} \alpha^k \Delta t \right) \tag{4.14}$$

where $\alpha^k = \max\limits_{1 \leq j \leq M} \alpha^j$. Using the bound of production term, $|\varepsilon^n|$ can be bounded as follows:

$$|\varepsilon^n| \leq \frac{1}{p^M} \max_{1 \leq i \leq M} |\delta p^i_{loc}| \sum_{i=1}^{M} \exp\left(\frac{M-i}{M} \alpha^k \Delta t \right). \tag{4.15}$$

The sum of exponential terms in Eq. (4.15) can be approximated as an integral as:

$$\begin{aligned} &\sum_{i=1}^{M} \exp\left(\frac{M-i}{M} \alpha^k \Delta t \right) \\ &\quad \approx \int_0^M \exp\left(\frac{\alpha^k \Delta t}{M} x \right) dx = \frac{M}{\alpha^k \Delta t} \left(\exp(\alpha^k \Delta t) - 1 \right). \end{aligned} \tag{4.16}$$

By plugging Eq. (4.16) into Eq. (4.15), the bound of $|\varepsilon^n|$ can be written as:

$$|\varepsilon^n| \leq \frac{1}{p^M} \max_{1 \leq i \leq M} |M\delta p^i_{loc}| \frac{\left(\exp(\alpha^k \Delta t) - 1\right)}{\alpha^k \Delta t}. \tag{4.17}$$

The definition of $M\delta p^i_{loc}$ can be simplified by neglecting the terms containing δt in its definition:

$$M\delta p^i_{loc} = \frac{\Delta t \epsilon^i \left(p^{i-1} + \delta t q^i_d\right)}{\left(1 - \delta t \alpha^{*,i}\right)\left(1 - \delta t \left(\alpha^{*,i} + \epsilon^i\right)\right)} \approx \Delta t \epsilon^i p^{i-1}. \tag{4.18}$$

By employing approximation in Eq. (4.18) in Eq. (4.17), the bound of $|\varepsilon^n|$ can be expressed as:

$$|\varepsilon^n| \leq \frac{\max\limits_{1 \leq i \leq M} |p_i|}{p^M} \max_{1 \leq i \leq M} |\epsilon^i| \frac{\left(\exp(\alpha^k \Delta t) - 1\right)}{\alpha^k \Delta t} \Delta t. \tag{4.19}$$

Fig. 4.7 shows the variation of the fractional term that contains $\alpha^k \Delta t$ in Eq. (4.19) over the range of $-1.5 \leq \alpha^k \Delta t \leq 1.5$. When $\alpha^k \Delta t$ has small magnitude, the fractional term is close to 1. When using adaptive control time step size control scheme, $\alpha^k \Delta t$ will be controlled under certain level and consequently the fractional term will be close to 1 and doesn't change rapidly. Therefore, the local amplitude error is mainly determined by ϵ and macro time step size. Eq. (4.19) can be written in more simplified form as:

$$|\varepsilon^n| \leq f^n \max_{1 \leq i \leq M} |\epsilon^i| \Delta t, \tag{4.20}$$

where

$$f^n = \frac{\max\limits_{1\le i\le M} |p_i|}{p^M} \frac{\left(\exp(\alpha^k \Delta t) - 1\right)}{\alpha^k \Delta t}.$$

With appropriate definition for ϵ, it is possible to evaluate the local error bound and optimal macro time step size.

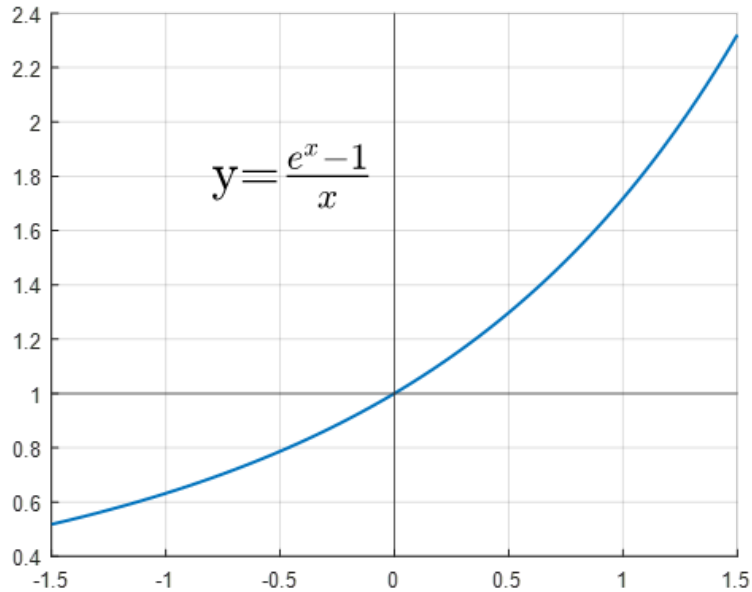


Figure 4.7 Variation of fractional term in local amplitude error bound

The error of α is caused by two causes: linear interpolation error and bias. Since the α is numerically calculated only at macro time steps, it is not calculated directly at the intermediate steps for PKE that exists between CMFD time steps. Therefore, α^m in the time interval $[t_{n-1}, t_n]$ is determined by linear interpolation of two points, (t_{n-1}, α^{n-1}) and (t_n, α^n). Fig. 4.8 shows how the local error is caused in time interval $[t_{n-1}, t_n]$ from the exact α at t_{n-1}. The green line represents an exact variation of α over the time interval. When

use a linear interpolation like blue line, interpolation error is introduced. Also, bias can be introduced in α at t_n. This bias can be introduced by various cause, such as the shape error from transient CMFD or cross section error from the transient heat conduction calculation. Later on, the linear interpolation error is denoted by ϵ_{lin} and the bias error is denoted by ϵ_{bias}.

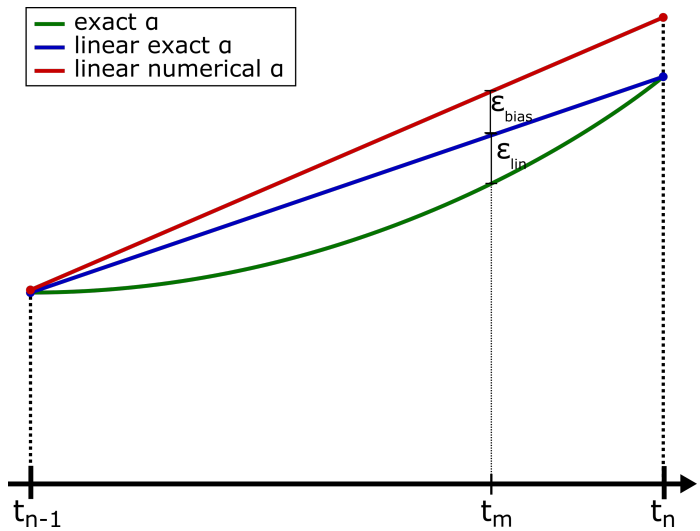


Figure 4.8 Schematic illustration of PKE parameter error

The error introduced from linear interpolation can be derived analytically. When the actual function $\alpha(t)$ has a second derivative, the linear interpolation error in the time interval $[t_{n-1}, t_n]$ is bounded by

$$|\epsilon_{lin}| \leq \frac{\Delta t_n^2}{8} \max_{t_{n-1} \leq t \leq t_n} |\alpha''(t)|. \tag{4.21}$$

It can be proven using Rolle's theorem. Unlike the linear interpolation error, it is much complicated to derive the bias error ϵ_{bias} rigorously especially for multi-physics calculation. $\alpha(t)$ can be defined using operators and shape function as:

$$\alpha(t) = \frac{\rho(t) - \beta(t)}{\Lambda(t)} = \frac{\left\langle \Phi_0^*, \left[\mathbf{F}(t) - \mathbf{F}_d(t) - \mathbf{M}(t)\right] \tilde{\Phi}(t) \right\rangle}{K_0}, \tag{4.22}$$

where K_0 is the constant defined as a unique factorization criteria for quasi-static method. In multi-physics calculation the local truncation error occurs for both shape and temperature which affects the operators. Also, each error affects each other. The effect of two errors on each other is neglected in the error model because of the small magnitude of the error. As shown in Fig. 4.8, the bias error from accumulated error of shape and temperature at t_{n-1} is not considered in adaptive time step control. First, accurate tracking of the accumulated error of shape or temperature is not available. Also, as shown in Eq. (4.20), the local amplitude error from the α bias from accumulated error can be seen as proportional to time step size. Consequently, controlling time step size doesn't affect the global error from accumulated α bias. Therefore only the α bias from local truncation error of shape and temperature are considred for adaptive time step control. The local truncation error of shape $\delta\tilde{\phi}$ for a specific pin i and group g at time step n is defined as:

$$\delta\tilde{\phi}_{i,g}^{n} = \frac{\Delta t_n^2}{2}\tilde{\phi}_{i,g}^{\prime\prime n}. \tag{4.23}$$

The bias of α caused by shape truncation error, ϵ_{shp}, can be directly obtained as:

$$\epsilon_{shp}^{n}(t) = \frac{\Delta t_n^2}{2}\frac{\left\langle \Phi_0^*, \left[\mathbf{F}^n - \mathbf{F}_d^n - \mathbf{M}^n\right]\delta\tilde{\Phi}^{\prime\prime n}\right\rangle}{K_0}. \tag{4.24}$$

On the other hand, explicit calculation of affect of temperature error on shape is complicated and not practical because it is hard to exactly quantify the affect of cross section change by the temperature error on α. Therefore, effect of temperature error on α is quantified using constant temperature coefficients as:

$$\rho = \rho_0 + \gamma_f \Delta T_f + \gamma_m \Delta T_m, \tag{4.25}$$

where γ_f and γ_m are fuel temperature coefficient and moderator coefficient and T_f and T_m are core averaged fuel temperature and moderator temperature, respectively. The core averaged temperatures are obtained through weighted average with importance factor as a weight function. The importance factor for each pin is defined as a sum of initial adjoint flux for all groups. The local truncation error of Crank-Nicolson method, which is used to solve heat conduction calculation, is defined as a third order function of time step size as:

$$\delta T \approx \frac{T^{(3)}}{12}\Delta t^3. \tag{4.26}$$

By using the simplified reactivity model, the affect of δT on PKE parameter α can be estimated as:

$$\begin{aligned} |\epsilon^n_{temp}| =& \frac{1}{\Lambda^n}\left|\gamma_f \delta T_f + \gamma_m \delta T_m\right| \\ \leq& \frac{\Delta t_n^3}{12\Lambda^n} \max_{t_{n-1}\leq t\leq t_n}\left|\gamma_f T_f^{(3)}(t) + \gamma_m T_m^{(3)}(t)\right|. \end{aligned} \tag{4.27}$$

As described above, the local amplitude errors are derived with assumption of exact shape and temperature at previous time step. However, the accumulated error of amplitude, ε^n_{glob}, can cause a significant local error in temperature when the power level is too high. Especially error can be caused for the fuel temperature which is directly affected by the power of the core. Following adiabatic model can be used to quantify the temperature error from global amplitude error, because the increase or decrease in heat in fuel pellet caused by the global amplitude error is not immediately transferred to the moderator:

$$\frac{dT}{dt} = \frac{1}{CM}p, \tag{4.28}$$

where C is a heat capacity of fuel and M is a total mass of fuel. The power integral error caused by ε^n_{glob} in time interval $[t_{n-1}, t_n]$ when using Crank-Nicolson method to solve Eq. (4.28) is defined using trapezoidal rule as:

$$\frac{1}{2}\left(\varepsilon^{n-1}_{glob}p^{n-1} + \varepsilon^{n}_{glob}p^{n}\right)\Delta t_n. \tag{4.29}$$

By using Eq. (4.25) and Eq. (4.20), the relative local power error caused from accumulated error can be approximated as it bound:

$$\varepsilon^n_{acc} \approx \frac{-\gamma_f f^n \Delta t_n^2}{2CM\Lambda^n}\left(\varepsilon^{n-1}_{glob}p^{n-1} + \varepsilon^{n}_{glob}p^{n}\right). \tag{4.30}$$

Unlike the local amplitude error from accumulated error of shape or temperature, the local amplitude error from accumulated amplitude error is proportional to square of time step size. Therefore, this error should be included in the local error model for adaptive time step control.

By combining the errors derived earlier, the final local amplitude error model for adpative time step control is defined as:

$$\varepsilon_{loc} = \varepsilon_{lin} + \varepsilon_{shp} + \varepsilon_{temp} + \varepsilon_{acc}. \tag{4.31}$$

Estimation of ε_{acc} requires the information of global error,ε^n_{glob}. Using Eq. (4.7), the growth factor of accumulated error at macro time step n can be defined as:

$$\frac{\delta p^n}{\delta p^{n-1}} = \prod_{m=1}^{M}\frac{1}{1-\delta t\alpha^m} \leq \left(\frac{1}{1-\delta t\alpha^j}\right)^M \tag{4.32}$$

where

$$\alpha^j = \max_{1 \le m \le M} \alpha^m.$$

With assumption of $\delta t \alpha^j \ll 1$, Eq. (4.32) can be approximated as a exponential as:

$$\left(\frac{1}{1-\delta t\alpha^j}\right)^M = \left(1+\frac{\delta t\alpha^j}{1-\delta t\alpha^j}\right)^M = \exp\left(M\frac{\delta t\alpha^j}{1-\delta t\alpha^j}\right) \approx \exp\left(\alpha^j \Delta t\right) \tag{4.33}$$

Then ε^n_{glob} can be defined using recurrence relation as follows:

$$\begin{aligned}
\varepsilon^n_{glob} &= \exp\left(\alpha^j \Delta t\right) \frac{p^{n-1}}{p^n} \varepsilon^{n-1}_{glob} + \varepsilon^n_{loc} \\
&= \exp\left(\alpha^j \Delta t\right) \frac{p^{n-1}}{p^n} \varepsilon^{n-1}_{glob} + (\varepsilon^n_{loc} - \varepsilon^n_{acc}) \\
&+ \frac{-\gamma_f f^n \Delta t_n^2}{2CM\Lambda^n} \left(\varepsilon^{n-1}_{glob} p^{n-1} + \varepsilon^n_{glob} p^n\right),
\end{aligned} \tag{4.34}$$

with initial condition $\varepsilon^0_{glob} = 0$. When reordering Eq. (4.34) by putting the terms containing ε^n_{glob} in the left hand side, ε^n_{glob} can be expressed as:

$$\varepsilon^n_{glob} = \frac{1}{(1-p^n g^n)} \left[\left(g^n + \frac{\exp\left(\alpha^j \Delta t\right)}{p^n}\right) p^{n-1} \varepsilon^{n-1}_{glob} + \varepsilon^n_{loc} - \varepsilon^n_{acc}\right], \tag{4.35}$$

where

$$g^n = \frac{-\gamma_f f^n \Delta t_n^2}{2CM\Lambda^n}.$$

The bound of local amplitude error can be defined as:

$$\begin{aligned}|\varepsilon_{loc}^n| &\leq |\varepsilon_{lin}^n| + |\varepsilon_{shp}^n| + |\varepsilon_{temp}^n| + |\varepsilon_{acc}^n| \\ &\leq f^n \left(C_{lin}^n \Delta t_n^3 + C_{shp}^n \Delta t_n^3 + C_{temp}^n \Delta t_n^4 + g^n \left(\varepsilon_{glob}^{n-1} p^{n-1} + \varepsilon_{glob}^n p^n \right) \right),\end{aligned} \tag{4.36}$$

where the coefficients are defined as:

$$\begin{aligned}C_{lin}^n &= \frac{1}{8} \max_{t_{n-1} \leq t \leq t_n} |\alpha''(t)|, \\ C_{shp}^n &= \frac{\left\langle \Phi_0^*, \left[\mathbf{F}^n - \mathbf{F}_d^n - \mathbf{M}^n \right] \delta \tilde{\Phi}''^n \right\rangle}{2K_0}, \\ C_{temp}^n &= \frac{1}{12\Lambda^n} \max_{t_{n-1} \leq t \leq t_n} \left| \gamma_f T_f^{(3)}(t) + \gamma_m T_m^{(3)}(t) \right|.\end{aligned}$$

4.2.2 Estimation of Error

For successful adaptive control of the time step size, the error bound derived in Eq. (4.36) should be estimated accurately during the calculation without posing heavy additional computational burden. The PKE calculation using micro time step size enables efficient and accurate estimation of the errors from linear interpolation. It can be approximated efficiently by running an additional PKE calculation using quadratic interpolation of α. The linear interpolation error at time step n is approximated as an amplitude difference between two PKE solution:

$$|\varepsilon_{lin}^n| \approx \frac{|p_{lin}^n - p_{quad}^n|}{p_{quad}^n}, \tag{4.37}$$

where p_{lin} is the amplitude using linear interpolation and p_{quad} is the amplitude using quadratic interpolation.

The bias of α from shape truncation error is directly calculated using Eq. (4.24). The second derivative of the shape of pin i and group g is calculated using the shape vector calculated in the latest 3 steps as:

$$\tilde{\phi}''^{n}_{i,g} \approx \frac{\tilde{\phi}^{n}_{i,g} - \tilde{\phi}^{n-1}_{i,g}}{\Delta t_n^2} - \frac{\tilde{\phi}^{n-1}_{i,g} - \tilde{\phi}^{n-2}_{i,g}}{\Delta t_n \Delta t_{n-1}}. \tag{4.38}$$

The temperature coefficients are required for estimation of bias of α from incorrect temperature. Both fuel temperature coefficient and moderator temperature coefficient are calculated by performing additional steady-state calculations with perturbed fuel temperature and moderator temperature respectively, after the steady state calculation converges. The temperature coefficient is calculated using reactivity difference between two calculations:

$$\gamma = \frac{1}{\Delta T}\left(\frac{1}{k_{eff}(0)} - \frac{1}{k_{eff}(\Delta T)}\right), \tag{4.39}$$

where ΔT is the perturbation magnitude, $k_{eff}(\Delta T)$ and $k_{eff}(0)$ are the eigenvalue calculated with and without perturbation, respectively. In this work, $5K$ increment is used as a perturbation for both fuel and moderator temperature.

The temperature truncation error is estimated using a time step doubling strategy. When using time step doubling method, the truncation error is estimated as a difference between two T/H calculations. At every macro time step, an additional T/H calculation is performed for two time steps with half time step size. The pin-wise linear heat generation rate is required as an input for T/H calculation. By using the PKE solution, intermediate T/H calculation can be performed without explicit calculation of the pin-wise linear heat generation rate through 3D CMFD calculation. The input for intermediate

step is estimated by correcting the linear interpolated pin-wise heat generation rates calculated at two macro time steps using the results of PKE solution as:

$$q_i^{n-\frac{1}{2}} = \frac{1}{2}\left(q_i^{n-1} + q_i^n\right)\frac{2p_{pke}^{\frac{M}{2}}}{p^{n-1}+p^n}. \tag{4.40}$$

Since the global truncation error of Crank-Nicolson is proportional to square of the time step size, the local truncation error of temperature can be calculated as:

$$\delta\bar{T}^n = \frac{4}{3}\left(\bar{T}^n_{\Delta t} - \bar{T}^n_{0.5\Delta t}\right), \tag{4.41}$$

where $\bar{T}$ is core averaged temperature weighted with importance factor. The error of fuel temperature and moderator temperature are calculated respectively and the local amplitude error from temperature truncation error is estimated as:

$$|\varepsilon^n_{temp}| = f^n\frac{\Delta t_n}{\Lambda^n}\left|\gamma_f\delta\bar{T}^n_f + \gamma_m\delta\bar{T}^n_m\right|. \tag{4.42}$$

The global amplitude error at each time steps using Eq. (4.35) and it is used to estimate the local amplitude error using Eq. (4.30). The specific heat of fuel is determined using the core average fuel temperature, $\bar{T}_f$. Note that there is a limit to accuracy of estimated global amplitude error because the global error of shape and temperature are not taken into account when calculating local amplitude error. However, if the time step size is large in a transient situation where the core power level is increasing very high, the error due to the global amplitude error can occur significantly. So, the effect of global error must be considered to guarantee certain accuracy, even if the estimation is somewhat inaccurate.

4.2.3 Determination of Time Step

The objective of the adaptive time step control is to maintain the local amplitude error ε_{loc} at each time step below a prescribed level. The prescribed error is denoted by ε_{pre}. Since the definition of local amplitude error bound in Eq. (4.36) is too complicated, it is not practical to find exact Δt that satisfies $\epsilon_{loc} = \epsilon_{pre}$ by solving Eq. (4.36) analytically. Rather than find exact Δt, simplified time step determination scheme that uses imaginary order of local amplitude error is employed. With given ε_{pre}, the time step can be determined as:

$$h^n = \left(\frac{\varepsilon_{pre}}{\varepsilon^n} \right)^{\frac{1}{l}} h' \tag{4.43}$$

where

h^n = time step size at n^{th} time step,

ε^n = relative local error of amplitude at n^{th} time step obtained with h',

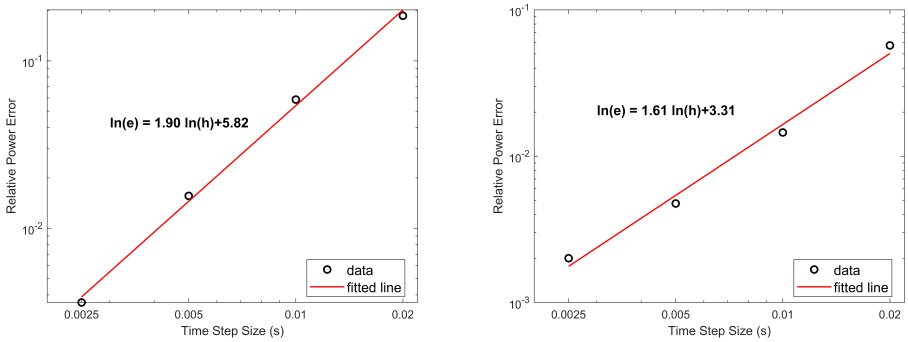
h' = trial time step size,

l = order of the local error.

As shown in Eq. (4.36), ε_{lin} and ε_{shp} is proportional to Δt^3 and ε_{temp} is proportional to Δt^4. The error from the accumulated error can be approximated as a function of time step size that has $(l+1)^{th}$ order. Therefore, it is possible to expect that ε_{lin} and ε_{shp} will work as a main contribution of error when Δt is small enough. Based on this expectation, the order of error l can be approximated as 3.

Fig. 4.9 shows the relative global error of power level in 5×5 fuel assembly problem (Hursin et al. 2016) calculated with various time step sizes. Fig. 4.9-

(a) shows the results at 0.1 s, immediately after the ejection of control rod is finished, and 4.9-(b) shows at 0.4 s, after the power peaking. When plotted in logarithmic scale, the slope of global amplitude error is close to 2 in both cases, which means it is proportional to the square of time step size. Because the global error has one less order than the local error, the numerical results are consistent to the approximated order of local amplitude error.



(a) Errors at 0.1 s **(b)** Errors at 0.4 s

Figure 4.9 Relative global error of power level in 5×5 problem

The estimation of local amplitude error ε^n is required to determine Δt_n. However, it is inevitable to estimate ε^n without knowledge of Δt_n. Also, estimation of ε^n requires additional calculation. Therefore, ε^n is approximated by extrapolation using two previous estimated errors, and treating previous time step size as trial time step size:

$$\varepsilon^n \approx \left(2\tilde{\varepsilon}^{n-1} - \tilde{\varepsilon}^{n-2}\right) \Delta t_n^3, \tag{4.44}$$

where $\tilde{\varepsilon}^n = \dfrac{\varepsilon^n}{\Delta t_n^3}$.

Because Eq. (4.44) involves various approximations, the estimated error can be not accurate enough. If the actual error is much larger than the

estimated error, the accuracy of the solution can be deteriorated. Consequently, deteriorated solution makes the error estimation less accurate. To prevent this situation, various stabilization techniques are applied during time step size determination. First, safety margin is applied to prescribed tolerance to make determined time step size smaller. The default value safety margin is set to 0.3. With safety margin θ, the formula of the time step size control is expressed as:

$$h^n = \left(\frac{(1-\theta)\,\varepsilon_{pre}}{\varepsilon^n} \right)^{\frac{1}{l}} h'. \tag{4.45}$$

On the other hand, if the time step size increases too rapidly, it is better to avoid it because the error can be larger than the prescribed tolerance. Therefore, maximum increase factor is introduced to prevent too rapid increase of time step size. The default value of maximum increase factor, 0.3, is determined empirically.

Although various stabilization techniques are applied, the estimated error can be larger than the specified error tolerance at several time steps. That means that the current time step size is too large at that step, and the solution is not accurate enough. In this case, the calculation doesn't proceed to the next step, and the calculation of current time step is performed again with smaller time step size. This scheme is denoted as restart scheme later on. The new time step size is determined by using Eq. (4.45). The overall time step size determination scheme is depicted in Fig. 4.10

4.2.4 Evaluation of Adaptive Time Step Size Control

The effectiveness of adaptive time step control algorithm is examined for the 5×5 fuel assembly problem (Hursin et al. 2016). The effect of incorrect

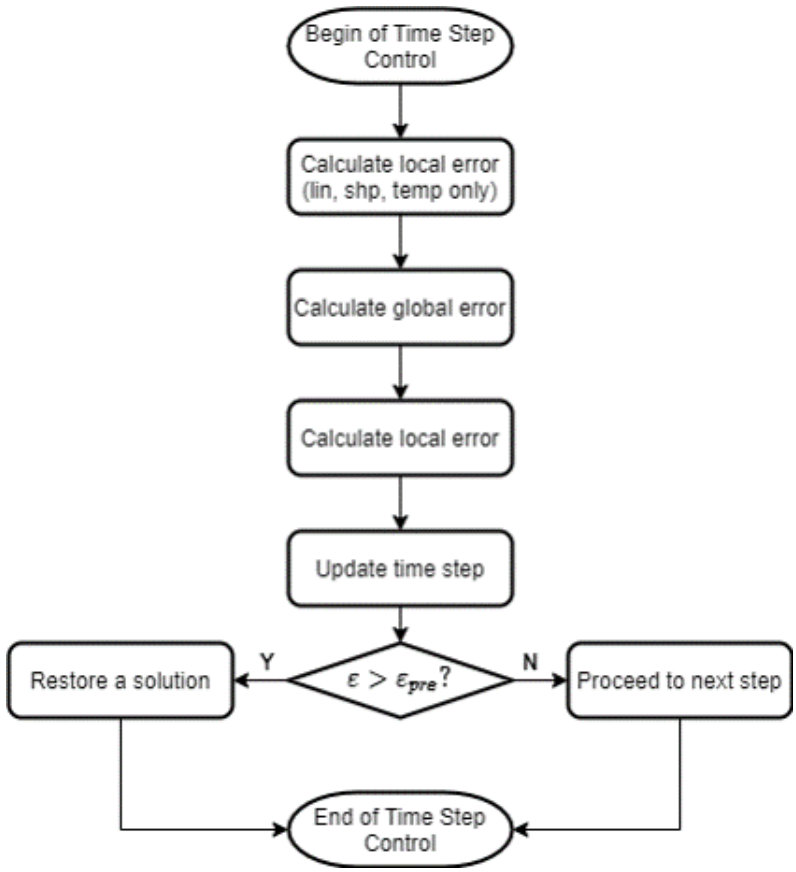


Figure 4.10 Adaptive time step size control algorithm

temperature on local amplitude error is incorporated using rather simple reactivity model with constant temperature coefficients in Eq. (4.25). Therefore, to confirm that the inaccuracy from the simple reacitivity is not significant, it should be verified first. There are two types of error related to temperature, ε_{temp} and ε_{acc}. Each error type is evaluated through comparison with actual errors.

The evaluation of ε_{acc} is performed by generate additional solution at each time step with perturbation of core power level. The actual $\epsilon^{n,*}$ is calculated

as the difference of α of two solutions as:

$$\epsilon^{n,*} = \alpha_{perturb} - \alpha. \tag{4.46}$$

On the other hand, the estimated error is calculated using Eq. (4.30). The calculation for actual error calculation is performed using time step size of 5 ms and power increment of 1 % is given as a perturbation at each time step. The comparison result of estimated error and actual error is shown in Fig. 4.11. The estimated error is well matched with the actual error. This indicates that the fuel temperature coefficient and adiabatic model used to estimate the effect of accumulated error on local amplitude error is accurate enough.

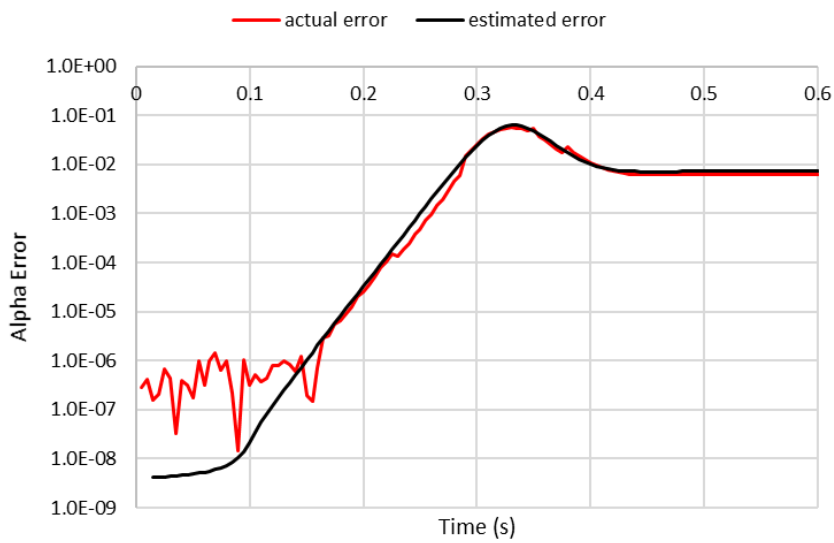


Figure 4.11 Estimated local error from accumulated error compared to actual error

The evaluation of ε_{temp} is conducted by performing additional T/H calculation using coarse time step size during transient calculation using fine time step size. The fine time step size was set as 1 ms and coarse time step size

was set as 5 ms. At every fifth fine time step, additional T/H calculation was performed which is updated from the T/H condition of 5 time steps before. The actual $\epsilon^{n,*}$ is calculated at every fifth time step as the difference of α using two different T/H conditions:

$$\epsilon^{n,*} = \alpha_{coarse} - \alpha_{fine}. \tag{4.47}$$

The estimated error is calculated by time step doubling method using PKE as Eq. (4.41). The comparison results are depicted in Fig. 4.12. Except for several dips in actual error behavior, the estimated error is well matched with the actual error. This indicates the validity of time step doubling strategy which uses PKE to estimate the heat generation rate at the intermediate time step.

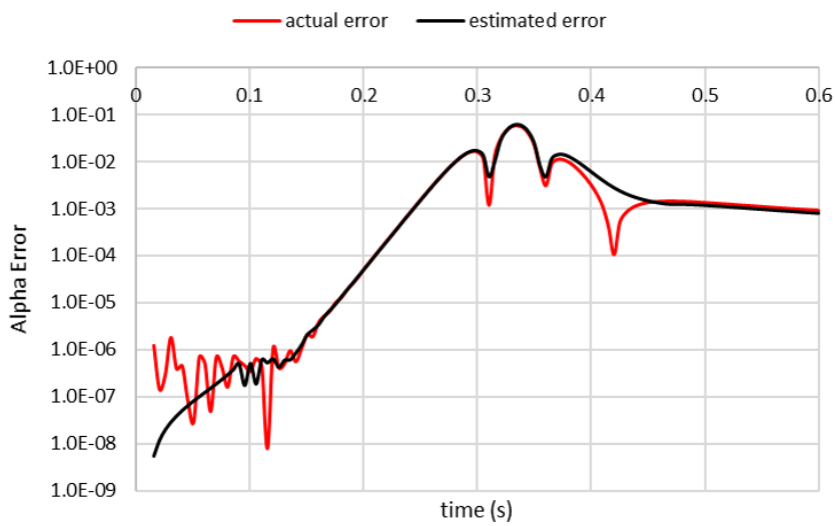


Figure 4.12 Estimated local error from temperature truncation error compared to actual error

Fig. 4.13 shows the variation of time step sizes with different prescribed tolerances. ATC refines the time step size with tight tolerance. Also, it automat-

ically controls time step size during the interval that contains significant change of core condition, such as control rod movement or power burst. Otherwise, ATC coarsening the time step size.

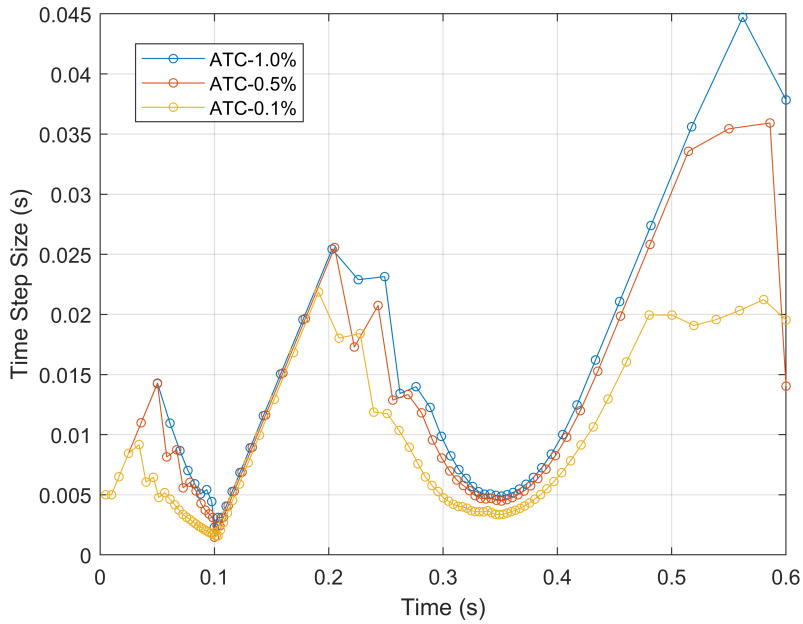


Figure 4.13 Time step size variation with different tolerances

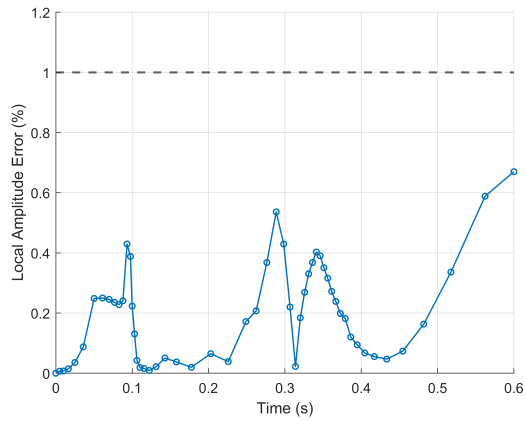
To evaluate the local amplitude error model, the local amplitude error is calculated by comparing with the reference solution as:

$$\varepsilon_{loc}^{n} = \frac{\left(p^{n} - p_{ref}^{n}\right) - \left(p^{n-1} - p_{ref}^{n-1}\right) exp(\alpha \Delta t_{n})}{p_{ref}^{n}}. \tag{4.48}$$

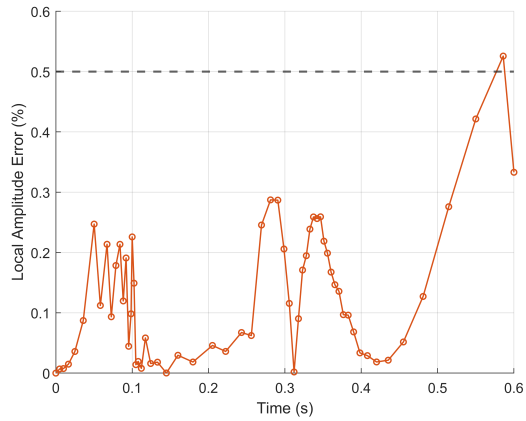
The solution calculated with fixed fine time step size of 1 ms is used as a reference solution. Fig. 4.14 shows the calculated local amplitude error behaviors with various prescribed tolerances. The dotted lines represent the given tolerance for local amplitude error. Fig. 4.14 error shows the local amplitude errors of ATC solutions with different tolerances. As shown in Fig.

4.14-(a) and (b), the calculated local amplitude error is controlled well below the prescribed error tolerance, when using relatively large tolerance. But as shown in Fig. 4.14-(c), the error over the tolerance after power burst though it is controlled well until 0.45 s. This phenomenon is caused by the global α error which is neglected for adaptive time step control. As described above, local amplitude error from the global α error is proportional to time step size, so the accumulated global error from this kind of error is not affected by time step size. Therefore, the large local amplitude error at the late stage of calculation is unavoidable. But the effectiveness of derived local amplitude error model can be validated in that it controls amplitude error well in intervals where rapid changes occur.

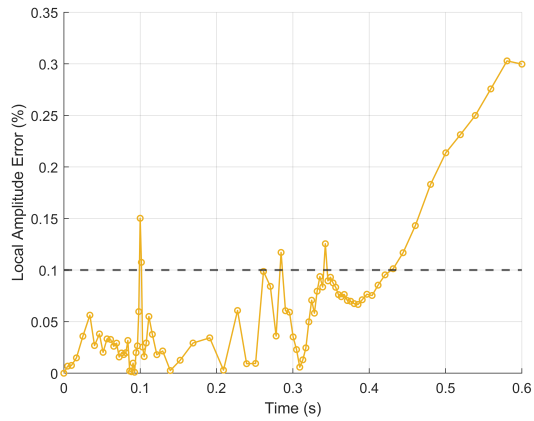
To examine the accuracy and efficiency of adaptive time step size control (ATC) algorithm, it is compared with the fixed time step size (FTS) results. The summarized results of performance examination are shown in Table. 4.2. The global errors at different time steps are calculated by comparing with the reference solution calculated with 1 ms fine fixed time step size. When compare the two ATC results with 1.0 % and 0.5 % tolerances, two cases requires same computing time because of more frequent invoking of restart when using 1.0 % tolerance. This indicates that using tight enough condition is necessary to optimize the computational efficiency of ATC. The comparison between FTS results with 10 ms time step size and ATC case with 0.5 % tolerance indicates that ATC can significantly improve the accuracy when similar number of time steps are used. The global amplitude error after the control rod ejection is reduced by about 90 %, the peak power error is reduced by about 80 % and the released energy error is reduced by about 30 %.



(a) $\varepsilon_{pre} = 1.0$ %



(b) $\varepsilon_{pre} = 0.5$ %



(c) $\varepsilon_{pre} = 0.1$ %

Figure 4.14 Local amplitude error with different tolerances

Table 4.2 Performance examination results of adaptive time step control algorithm

	Fixed time step size		Adaptive time step size		
	10 ms	5 ms	1.0 %	0.5 %	0.1 %
# of time steps	60	120	54	61	85
# of restart	-	-	7	4	4
Computing time (m)	55	89	56	56	64
Power rel. error at 0.1 s (%)	6.4	1.64	1.25	0.72	0.20
Peak power rel. error (%)	2.3	0.49	1.08	0.41	0.40
Total released energy rel. error (%)	0.43	0.14	0.33	0.31	0.23

4.3 Employment of Multi T/H Steps

As a transient situation progresses, the contribution of each error can vary significantly. Fig. 4.15 shows evolution of contribution of each type of error when using ATC to solve the 5×5 fuel assembly control rod ejection problem up to 2 s. In this case, the local amplitude error tolerance was given as 0.1 %. As the neutron flux shape changes significantly during the control rod ejection, shape truncation error contributes a lot to macro time step size determination. The local amplitude error from linear interpolation of α is the main source of error during exponential growth of core power level after the control rod ejection. Around the time of reaching the maximum power level, the macro time step size is determined by the local amplitude error from global amplitude error. After the power burst, temperature truncation error becomes a main source of local amplitude error. Though the core power level doesn't change rapidly in this interval, the moderator temperature changes relatively rapidly due to the heat transfer from fuel surface to moderator increases continuously after the power burst caused by control rod ejection.

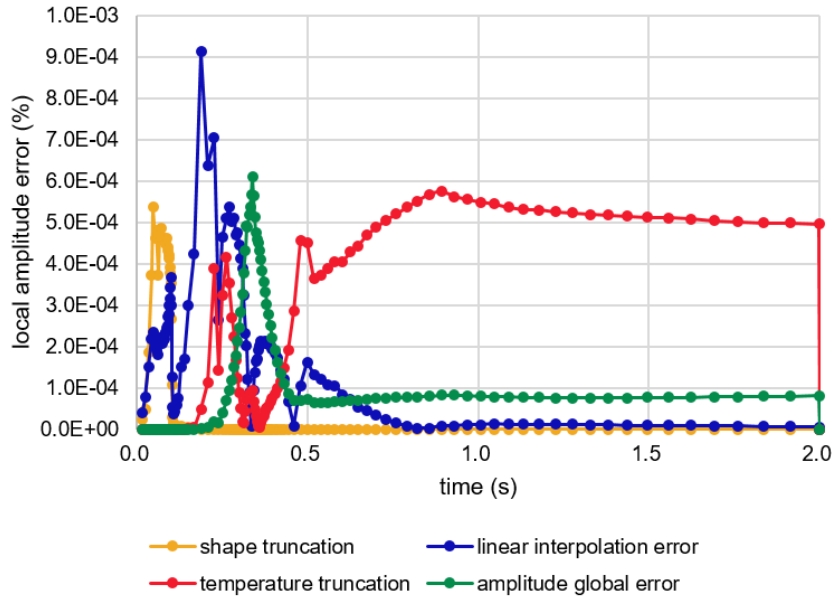


Figure 4.15 Contribution of each error type

When considering that the computing time for neutronics calculation is longer than T/H calculation, it is not preferable that the macro time step size is determined by truncation error of T/H calculation. Therefore, Multi TH step scheme is employed for ATC. It additionally calculates the independent time step size for T/H calculation that can regulates the contribution of temperature truncation error in macro time step size determination procedure. The time step size for T/H calculation in n$^{\text{th}}$ macro time interval is denoted as Δt_n^{TH}. When the tolerance for contribution of temperature truncation error is given as Θ, T/H time step size is determined based on the time step determination algorithm in Eq. (4.43):

$$\Delta t_n^{TH} = \left(\frac{\Theta |\varepsilon_{pre}|}{|\varepsilon_{temp}^{n-1}|} \right)^{\frac{1}{2}} \Delta t_{n-1}^{TH}. \tag{4.49}$$

Note the accumulated amplitude error over macro time interval $[t_{n-1}, t_n]$ due to temperature truncation error is proportional to square of TH time step size. The number of T/H time steps in n$^{\text{th}}$ macro time interval is denoted as L. For intermdeiate T/H calculations, pin-wise heat generation rates are calculated by correcting the linear interpolated heat generation rate using PKE solutions as:

$$q_i(t_l) = \left(w_l q_i^{n-1} + (1 - w_l)\, q_i^n \right) \frac{2p_{pke}^l}{w_l p^{n-1} + (1 - w_l)\, p^n}, \qquad l = 1, \ldots, L, \tag{4.50}$$

where $w_l = \dfrac{t_l - t_{n-1}}{\Delta t_n}$.

The amplitude error caused by temperature truncation error over macro time interval $[t_{n-1}, t_n]$ is calculated as a difference between two T/H calculation results as before applying multi T/H step scheme. When $L = 1$, the temperature truncation error is calculated as EQ. (4.41). On the other hand, when L is larger than 1, additional T/H calculation using macro time step size, and temperature truncation error is calculated as:

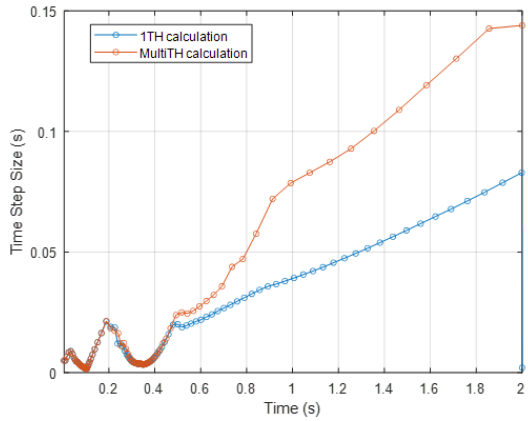
$$\delta \bar{T}^n = \frac{L^2}{\left(L^2 - 1\right)} \left(\bar{T}_{\Delta t}^n - \bar{T}_{\Delta t^{TH}}^n \right). \tag{4.51}$$

Table 4.3 and Fig. 4.16 shows the calculation results of multi T/H step scheme for the 5×5 fuel assembly control rod ejection problem up to 2 s, and its comparison with the results of ATC without multi T/H step scheme. The local amplitude error tolerance was given as 0.1 % for both cases. As expected, by using multi T/H step scheme, larger macro time step can be used for later interval (0.5-2 s) of the calculation. As a result, the computing time is reduced by about 10 %. Since the pin-wise heat generation rate used for

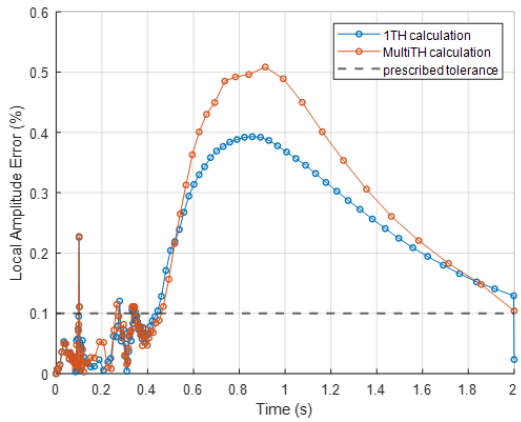
intermediate T/H calculations obtained by correction using PKE solutions rather than explicit calculation, the quality of error control is slightly lowered when compared to the case without multi T/H scheme. Consequently, larger error was observed when using multi T/H scheme as shown in Fig. 4.16-(b), (c). However, as the calculation progresses, both local and global error decreases and the accuracy of the two cases becomes similar. Considering that the local amplitude error is somewhat proportional to the cubic of macro time step size, multi T/H step scheme can be seen as efficient, because the local amplitude error ratio between two case is smaller than macro time step size ratio. Furthermore, multi T/H can be more efficient when analyzing changes up to a longer time than this case.

Table 4.3 Results with multi T/H step scheme

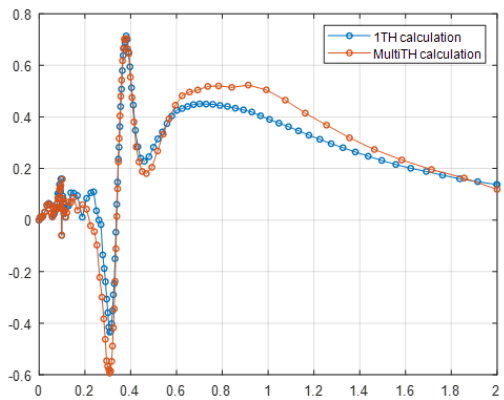
	1 T/H	Multi T/H
# of time step	116	99
Computing time (m)	84	77
Power rel. error at 0.1 s (%)	0.16	0.16
Peak power rel. error (%)	0.32	0.46
Power integral error (%)	0.28	0.31



(a) Time step size variation



(b) Local amplitude error



(c) Global amplitude error

Figure 4.16 The results of ATC employing multi T/H step scheme

5 Enhancement of CMFD Solution

5.1 Formulations of Transient CMFD

The source iteration which adopts so-called *group major* ordering has been widely used for CMFD solution. With group major ordering, the CMFD equation, whose source term is updated in Gauss-Seidel manner, can be solved group by group. If this source iteration converges fast, heavy computational cost from solve the entire CMFD linear system directly. However, group major ordered CMFD showed poor convergence and instability for transient problems under certain conditions. Also, it is not appropriate for the GPU application because of heavy overhead communication cost from source iteration. As an alternative, multi-group direct solution of CMFD linear system was also implemented in nTRACER. In multi-group direct solution, the entire CMFD linear system in Eq. (2.26) is solved directly.

5.1.1 Group Major Ordering

In group major ordering, the linear system variable is first arranged according to the group index and then arranged according to the spatial index. As the result, the transient CMFD linear system $\mathbf{A}\Phi = Q$, which is formulated by applying group major ordering to Eq. (2.26) become a block matrix formed for energy groups as:

$$\begin{pmatrix} \tilde{\mathbf{R}}_1 & -\tilde{\mathbf{S}}_{2,1} & \dots & -\tilde{\mathbf{S}}_{G-1,1} & -\tilde{\mathbf{S}}_{G,1} \\ -\tilde{\mathbf{S}}_{1,2} & \tilde{\mathbf{R}}_2 & \ddots & -\tilde{\mathbf{S}}_{G-1,2} & -\tilde{\mathbf{S}}_{G,2} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -\tilde{\mathbf{S}}_{1,G-1} & -\tilde{\mathbf{S}}_{2,G-1} & \ddots & \tilde{\mathbf{R}}_{G-1} & -\tilde{\mathbf{S}}_{G,G-1} \\ -\tilde{\mathbf{S}}_{1,G} & -\tilde{\mathbf{S}}_{2,G} & \dots & -\tilde{\mathbf{S}}_{G-1,G} & \tilde{\mathbf{R}}_G \end{pmatrix} \begin{pmatrix} \Phi_1 \\ \Phi_2 \\ \vdots \\ \Phi_{G-1} \\ \Phi_G \end{pmatrix} = \begin{pmatrix} Q_1 \\ Q_2 \\ \vdots \\ Q_{G-1} \\ Q_G \end{pmatrix} \tag{5.1}$$

where $\tilde{\mathbf{R}}_g$ is a septa-diagonal matrix which contains removal terms, a $1/v\Delta t$ term, and fission source from group g in group g, while $\tilde{\mathbf{S}}_{g'g}$ is a diagonal matrix which contains a scattering source and fission source in group g from another group g'.

Algorithm 3 shows how the CMFD linear system is solved when using group major ordering. The whole process can be interpreted as a block Gauss-Seidel iteration. For each group, 1G linear system is constructed and solved using BiCGSTAB solver. Because of small, septa-diagoanl coefficient matrix, this linear system doesn't require a lot of iterations. Therefore, if the block Gauss-Seidel converges well, a reduction in total FLOPs and computing time can be anticipated.

However, block Gauss-Seidel method doesn't always guarantee fast and stable convergence. The Gauss-Siedel method can be seen as the iterative solution of the linear system $\mathbf{A}x = b$ using matrix splitting. The splitting of coefficient matrix $\mathbf{A}$ can be written as:

$$\mathbf{A} = \mathbf{M} - \mathbf{N}, \tag{5.2}$$

Algorithm 3 Group major ordered CMFD algorithm

Compute residual: $r^0 = Q - \mathbf{A}\Phi^i$
$i = 1$
while $\|r^i\| > \epsilon\|r^0\|$ **do** ▷ Block Gauss-Seidel iteration
  for $g = 1$ to G **do** ▷ Group sweep
    Update source: $\tilde{Q}_g^i = Q_g + \sum_{g'=1}^{g-1} \tilde{\mathbf{S}}_{g',g}\Phi_{g'}^{i-1} + \sum_{g'=g+1}^{G} \tilde{\mathbf{S}}_{g',g}\Phi_{g'}^{i}$
    Solve 1G system: $\tilde{\mathbf{R}}_g\Phi_g^i = \tilde{Q}_g^i$ ▷ Using *BiCGSTAB*
  end for
  Compute residual: $r^i = Q - \mathbf{A}\Phi^i$
  $i = i + 1$
end while

where $\mathbf{M} \in \mathbb{R}^{n\times n}$ is a nonsingular matrix. The iterative methods can be formulated to generate the approximate solution $x^{(i)}$ as:

$$x^{(i)} = \mathbf{M}^{-1}\mathbf{N}x^{(i-1)} + \mathbf{M}^{-1}b, \quad i \geq 1, \tag{5.3}$$

where the starting vector $x^{(0)}$ is given. The error of the approximate solution $x^{(i)}$ is defined as:

$$e^{(i)} = x^* - x^{(i)}. \tag{5.4}$$

If the iterative method is convergent, following condition should be satisfied:

$$\lim_{i\to\infty} e^{(i)} = \lim_{i\to\infty} \left(\mathbf{M}^{-1}\mathbf{N}\right)^i e^{(0)} = 0. \tag{5.5}$$

If $e^{(0)} \neq 0$, this condition is equivalent to following condition:

$$\lim_{i\to\infty} \left(\mathbf{M}^{-1}\mathbf{N}\right)^i = \lim_{i\to\infty} \left[\rho\left(\mathbf{M}^{-1}\mathbf{N}\right)\right]^i = 0. \tag{5.6}$$

Therefore, this iterative method is convergent to the unique solution x^* if and only if $\rho(\mathbf{M}^{-1}\mathbf{N}) < 1$.

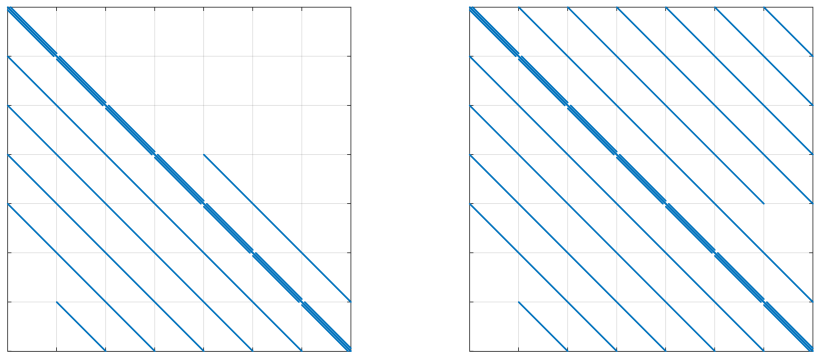
The block Gauss-Seidel method used in group major ordered CMFD split the coefficient matrix $\mathbf{A}$ as:

$$\mathbf{M} = \mathbf{D} - \mathbf{L}, \qquad \mathbf{N} = \mathbf{U}, \tag{5.7}$$

where

$$\mathbf{D} = \left\{ \tilde{\mathbf{R}}_g \in \mathbb{R}^{N\times N} | g = 1, \ldots, G \right\},$$
$$\mathbf{L} = \left\{ \tilde{\mathbf{S}}_{i,j} \in \mathbb{R}^{N\times N} | i < j, \quad i = 1, \ldots, G, \quad j = 1, \ldots, G \right\},$$
$$\mathbf{U} = \left\{ \tilde{\mathbf{S}}_{i,j} \in \mathbb{R}^{N\times N} | i > j, \quad i = 1, \ldots, G, \quad j = 1, \ldots, G \right\},$$

where N is the total number of spatial node, G is the total number of energy group. Faster convergence can be expected for small upper matrix U which represents the neutron source from lower energy to higher energy. The sparsity structure of CMFD matrix of steady-state and transient problem are shown in Fig. 5.1. It is part of CMFD matrices generated for small problem using 7 group macro cross section of C5G7-TD benchmark (Boyarinov et al. 2016) to show the difference between steady-state and transient matrix schematically. Unlike steady-state problem which is an eigenvalue problem, the transient problem is fixed source problem which contains the fission source terms. Therefore, the upper matrix of transient CMFD matrix contains lots of elements having large magnitude. Considering this aspect, transient CMFD linear system is disadvantageous to use block Gauss-Seidel method using group major ordering compared to steady-state CMFD linear system.



(a) Steady-state CMFD matrix **(b)** Transient CMFD matrix

Figure 5.1 Comparison of sparsity patterns of CMFD matrix

5.1.2 Multi-group Direct Solution

Contrary to group major ordering, the linear system variable is first arranged according to the node index and then arranged according to the group index in node major ordering. Rather than solving the CMFD linear system group-by-group, the whole linear system $\mathbf{A}\Phi = Q$ is solved directly using the iterative solver such as BiCGSTAB. When using this approach, the reduction of calculation cost from excluding scattering terms in iterative solver like group major ordering CMFD, but stable convergence is possible if well constructed preconditioner is prepared.

5.1.3 Numerical Calculation Results

The quadrant of 5×5 2D problem which is based on APR1400 benchmark(Yuk and Cho 2019) is used for investigation of CMFD solution schemes. The configuration of the problem is depicted in Fig. 5.2.

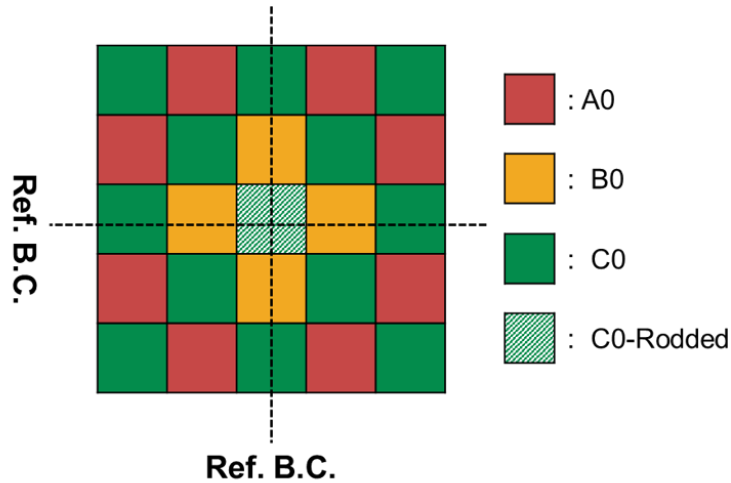


Figure 5.2 Configuration of 5×5 quadrant problem

The spectral radius of iteration matrix $\mathbf{M}^{-1}\mathbf{N}$ is calculated for CMFD matrices generated in steady-state problem and three transient problems with different dynamic reactivities. In addition to conventional forward Gauss-Seidel method, Jacobi and backward Gauss-Seidel method were also investigated. The results are summarized in Table 5.1. For steady-state problem, Gauss-Seidel method shows lower spectral radius than Jacobi method. As expected, the spectral radii for transient cases are close to 1 which means the convergence speed is very slow. The higher the dynamic reactivity, the larger the spectral radius. Even divergent case was observed when using the Jacobi method for the transient problem with high dynamic reactivity.

Though the poor convergence behavior of group major ordered Gauss-Seidel solution for transient problem is identified, it is compared with multi-group direct solution scheme to clearly compare the efficiency of two solution schemes. Same problem is solved using two solutions schemes. The calculation is performed until the residual error reduction rate reach the criterion of 0.001. The comparison results are shown in Table 5.2. Due to the poor convergence

Table 5.1 Spectral radii of group major ordered CMFD solution

	Steady-state	Transient		
ρ_d (pcm)	-	0	436	1049
Jacobi	0.91456	0.99930	0.99941	1.00693
Forward GS	0.83944	0.99686	0.99751	0.99803
Backward GS	0.83805	0.99910	0.99925	0.99941

of group major CMFD scheme for transient problem, the number of required outer iteration for transient problem is more than 20 times larger than the required iterations for steady-state problem. More BiCGSTAB iteration is also required for transient solution when using the multi group direct solution, but the increase ratio was about 3∼4. Regardless of these characteristics of group major CMFD solution, the computing time of group major CMFD solution is much larger than multi group direct solution for both steady-state and transient problems. This is because group major CMFD solution is not efficient for the GPU-base CMFD calculation. To get an optimized performance of GPU calculation, there should be enough data to exploit enough parallelism of GPUs. However, block matrices used in group major CMFD solution is too small to exploit massive parallelism of GPUs. Also, there latency is occurred when launching a GPGPU kernel to run the computationally intensive portion of the code on GPU. Since group major CMFD solution requires more frequent launch of GPGPU kernel than multi group direct solution, more latency is occurred. Therefore, multi-group direct CMFD solution is applied for the transient capability of nTRACER.

Table 5.2 Comparison of two CMFD solution schemes

	Steady-state		Transient					
	GM	Direct	GM	Direct	GM	Direct	GM	Direct
ρ_d (pcm)	-		0		436		1049	
total computing time (s)	1.25	0.0105	22.4	0.0309	28.3	0.0420	37.8	0.0387
time ratio (GM / Direct)	119		724		673		976	
BiCGSTAB time (s)	0.924	0.0096	16.3	0.0302	20.7	0.0413	27.3	0.0379
# of BiCGSTAB	3,829	19	75,928	55	93,853	80	124,743	73
# of outer	26	-	520	-	643	-	855	-

5.2 Preconditioner for Node Major Transient CMFD

5.2.1 Sparse Approximate Inverse Preconditioner

In multi group direct transient CMFD scheme the entire CMFD linear system $\mathbf{A}\Phi = Q$ is solved directly, where $\mathbf{A}$ is a large, sparse, and asymmetric matrix. Because of the size and asymmetry of $\mathbf{A}$, BiCGSTAB iterative solver is applied in nTRACER. The efficiency of Krylov iterative solver such as BiCGSTAB is determined by the quality of preconditioner. The preconditioner based on an incomplete LU factorization (ILU) has been widely used due to its high quality of preconditioning. Especially, the Block ILU preconditioner (BILU)(Joo and Downar 1996) which is specialized for single-group CMFD system from group major ordered CMFD has been used for CPU-based CMFD solver of nTRACER and enables the efficient solution. However, BILU is not applicable in node major CMFD linear system because of the complex structure of the coefficient matrix. Therefore, other types of ILU preconditioner, such as ILU(0), which can be applied to general matrices, should be used for node major CMFD. More significant disadvantage of applying ILU preconditioner is that it is not well-parallelized. The application of the preconditioner requires the solution of triangular system at each step, which is not inherently parallel. This disadvantage can significantly deteriorate efficiency in massive parallel environment such as GPU.

A preconditioner that approximates the inverse of $\mathbf{A}$ is advantageous in parallelization aspect. The application of the preconditioner only involves matrix-vector multiplication which is inherently parallel. These type of preconditioners are called *Sparse Approximate Inverse* (SPAI) preconditioner. Among the several methods to approximate the inverse matrix, and the method based on

Frobenius norm minimization (Grote and Huckle 1997) is implemented in this study. The method computes a sparse matrix $\mathbf{M} \approx \mathbf{A}^{-1}$ for a given sparsity structure as the solution of the following minimization problem:

$$\min_{\mathbf{M} \in \mathcal{S}} \|\mathbf{I} - \mathbf{AM}\|_F, \tag{5.8}$$

where $\mathcal{S}$ is a set of sparse matrices that has a given sparsity structure, $\mathbf{I}$ is the identity matrix, and $\|\cdot\|_F$ is the Frobenius norm. Because square of the Frobenius norm is equivalent to square sum of Euclide norm of all column as:

$$\|\mathbf{I} - \mathbf{AM}\|_F^2 = \sum_{k=1}^{n} \|e_k - \mathbf{A}m_k\|_2^2, \tag{5.9}$$

Eq. (5.8) can be separated into independent Euclide norm minimization problems for each column as:

$$\min_{m_k} \|e_k - \mathbf{A}m_k\|_2, \quad k = 1, \ldots, n, \tag{5.10}$$

where e_k and m_k is the k^{th} column of $\mathbf{I}$ and $\mathbf{M}$ respectively. Therefore, this method enables parallel construction of SPAI preconditioner and that is why this method was selected.

Solving Eq. (5.10) for a specific column index k is equivalent to solving

$$\min_{\hat{m}_k} \|\hat{e}_k - \hat{\mathbf{A}}_k \hat{m}_k\|_2, \tag{5.11}$$

where

$$\hat{m}_k = \left\{ m_k(j) | j \in \mathcal{J} \right\}, \qquad \mathcal{J} = \left\{ j | m_k(j) \neq 0 \right\},$$

$$\hat{\mathbf{A}}_k = \left\{ \mathbf{A}(i,j) | i \in \mathcal{I}, \quad j \in \mathcal{J} \right\}, \qquad \mathcal{I} = \left\{ i | \mathbf{A}(i,\mathcal{J}) \neq 0 \right\},$$

$$\hat{e}_k = \left\{ e_k(i) | i \in \mathcal{I} \right\}.$$

The submatrix $\hat{\mathbf{A}}_k$ has full rank if $\mathbf{A}$ is nonsingular. Then $\hat{\mathbf{A}}_k$ can be QR decomposed as:

$$\hat{\mathbf{A}}_k = \mathbf{Q}_k \begin{pmatrix} \mathbf{R}_k \\ 0 \end{pmatrix}, \tag{5.12}$$

and the solution of Eq. (5.11) can be obtained as:

$$\hat{m}_k = \mathbf{R}_k^{-1} \mathbf{Q}^T \hat{e}_k. \tag{5.13}$$

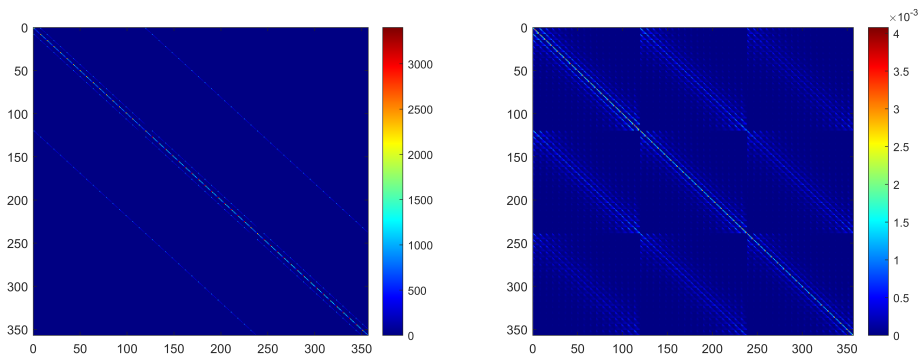
5.2.2 A Priori Sparsity Structure for SPAI preconditioner

Though the construction procedure is inherent parallel, the computational cost of QR decomposition can be very expensive if the number of nonzero elements of given sparsity structure is too large, because the required FLOPs of the submatrix $\hat{\mathbf{A}}_k \in \mathbb{R}^{m \times n}$ is $2n^2 \left(m - n/3\right)$. Therefore, it is essential to find efficient sparsity structure to approximate the inverse matrix well even with small number of nonzero elements. The adaptive procedure was developed to obtain applicability to general problems (Grote and Huckle 1997). In this procedure following numerical test is performed to judge whether to add or not a nonzero at j^{th} row at k^{th} column:

$$\frac{\left(r_k^T \mathbf{A} e_j\right)^2}{\|\mathbf{A} e_j\|_2^2} > tolerance, \tag{5.14}$$

where $r_k = e_k - \mathbf{A}m_k$ is the residual for a given sparsity pattern. This process is repeated until the residual norm becomes lower than a given threshold. For node major ordered CMFD, this approach would be very expensive. Therefore, *a priori* sparsity structure was selected for steady-state CMFD solver of nTRACER.

The easiest way to determine a sparsity structure is using the sparsity structure of coefficient matrix without modification. Fig. 5.3 shows scaled images of CMFD matrix and its inverse matrix respectively. It is clear that inverse matrix has entries having large magnitude at sparsity structure of CMFD coefficient matrix. However, using the sparsity pattern involving all entries of the coefficient matrix required heavy cost compared to the quality of resulted preconditioner. Therefore, *dropped out* version of sparsity structure was also tried for steady-state CMFD solution of nTRACER. It dropped out the sparsity structure corresponding to scattering terms. Though most of the nonzero elements of the coefficient matrix are scattering terms, their magnitudes are relatively small.



(a) CMFD matrix **(b)** Inverse matrix

Figure 5.3 Scaled images of matrices from steady-state problem

The dropped out sparsity pattern generates preconditioner with quite good quality without posing heavy construction cost, but it has the disadvantage of limiting the quality of preconditioning below a certain level. In steady-state calculation, reduction of preconditioner construction cost by using dropped out sparsity structure can be an optimal choice because preconditioner is used for one steady-state only. However, the preconditioner constructed at initial time step can be used for many subsequent time steps in transient calculation. Also, computing time proportion of CMFD is also increased in transient calculation due to the conditional transport update scheme. Therefore, the benefits of improving the quality of the preconditioner can be greater than the resulting increase of construction time.

For improved quality of SPAI preconditioner, prediction algorithm for a priori sparsity pattern determination is devised. The prediction algorithm for k^{th} column of SPAI preconditioner is shown in Algorithm 4. As shown in Eq. (5.10), SPAI algorithm finds a vector m_k that minimizes the norm of the residual vector r_k which is defined as:

$$r_k = e_k - \mathbf{A}m_k. \tag{5.15}$$

The second term of the right hand side of Eq. (5.15) can be represented as a linear combination of column vectors of CMFD matrix $\mathbf{A}$:

$$\mathbf{A}m_k = \sum_j m_{j,k} a_j, \tag{5.16}$$

where a_j is j^{th} column of CMFD matrix. Note that the (j,k) element of SPAI matrix is used as a linear combination coefficient. If j^{th} element of the residual

vector, $r_{j,k}$, is not zero, adding the (j,k) element can reduce the residual norm. It is based on the characteristic of CMFD matrix that the diagonal element has larger magnitude than other elements in the same column. By add $m_{j,k} = -r_{j,k}/a_{j,j}$, $r_{j,k}$ becomes 0 and the new residual elements caused by $m_{j,k}$ can be estimated as follows by neglecting the influence from other element:

$$r_{i,k} = m_{j,k} a_{i,j}, \quad i \in \left\{ i : a_{i,j} \neq 0, i \neq j \right\}. \tag{5.17}$$

The expected reduction of residual norm by adding $m_{j,k}$ is equivalent to the ratio of root of squared sum of diagonal element and other elements of column a_j:

$$\sqrt{\frac{\sum\limits_{i \neq j} (r_{i,k})^2}{(r_{j,k})^2}} = \sqrt{\frac{\sum\limits_{i \neq j} (a_{i,j})^2}{(a_{j,j})^2}} \tag{5.18}$$

This procedure is done iteratively until the estimated residual elements become smaller than given criterion.

The performance of SPAI preconditioners using various sparsity structures are examined for CMFD matrix generated for transient case with dynamic reacitivity of 1049 pcm of 5×5 quadrant problem. The required construction time, Frobenius norm, maximum column-wise residual norm, and required BiCGSTAB iterations to reach given convergence criteria are compared to evaluate the quality of SPAI preconditioner. The convergence criterion is given as residual error reduction of 10^{-6}. The comparison of fixed-full and fixed-dropped clarifies the efficiency of dropped out strategy and inefficiency of fixed-full strategy, respectively. Though 10 times more nonzeros are used, the Frobenius norm only reduced by only 13 %. 0.1 drop criteria for prediction algorithm induces similar number of nonzeros with dropped out strategy. Consequently,

Algorithm 4 SPAI prediction algorithm for k^{th} column

1: $m_k = 0$
2: $r_k^{(0)} = e_k$
3: **for** iLv = 1 to nLvMax **do**
4: **for** j$\in \left\{ j : r_{j,k}^{(iLv)} \neq 0 \right\}$ **do**
5: $m_{j,k} = -\dfrac{r_{j,k}^{(iLv)}}{a_{j,j}}$
6: **for** i$\in \left\{ i : a_{i,j} \neq 0, i \neq j \right\}$ **do**
7: $r_{i,k}^{(iLv)} = m_{j,k} a_{i,j}$
8: **if** $|r_{i,k}^{(iLv)}| < \epsilon$ **then** ▷ Drop small elements
9: $r_{i,k}^{(iLv)} = 0$
10: **end if**
11: **end for**
12: **end for**
13: **if** $\|r_k^{(iLv)}\| = 0$ **then**
14: Exit
15: **end if**
16: **end for**

construction time, and the quality of resulted SPAI preconditioner of Fixed-dropped case and Predict-0.1 are similar, though there are slight differences. The noticeable difference was only observed for maximum column-wise residual norm. Using prediction algorithm induces smaller maximum column-wise residual norm which means the residual is more uniform distribution when compared to fixed sparsity case. By using more tight drop criteria for prediction algorithm, SPAI preconditioners which has better quality than dropped case can be generated. Unlike Fixed-full case, using more nonzeros enhances the quality of preconditioner which indicates that the prediction algorithm could find the efficient sparsity pattern as intended. Due to the increased number of nonzeros the construction time is increased when using more tight drop criteria. In this

case, only one matrix is solved using the preconditioner and only about hundred iterations are required. Since the performance of GPU-based BiCGSTAB solver is too great, that the required computing time for hundreds iterations is less than 0.1 s which is far less than the increased computing time. But in transient calculation, myriads of BiCGSTAB iterations are required for hundreds of time steps. Therefore, the gain from reduced BiCGSTAB iteration can exceed the loss from increased construction time.

Table 5.3 Quality of SPAI preconditioner with various sparsity structures

CASE	Drop criteria	NNZ	Construction time (s)	Frobenius Norm	Max. Residual norm	# of iteration
Fixed-full	-	3,704,555	55.1	54.9	0.412	116
Fixed-dropped	-	368,480	1.01	62.7	0.424	124
Predict	0.1	378,244	1.03	60.2	0.319	135
	0.03	967,330	3.71	41.6	0.272	90
	0.01	1,880,959	9.99	33.0	0.224	81

The performance of SPAI preconditioner using new sparsity structure determination algorithm for realistic core transient calculation is examined through CEA ejection analysis in quarter-core problem in the benchmark based on APR1400(Yuk and Cho 2019). The transient calculation was performed up to 1 s using 110 time steps. The SPAI preconditioner generated at the first time step is used for whole transient calculation. The examination results are shown in Table. 5.4. When using drop tolerance of 2 %, the number of nonzeros becomes about 3.3 times larger than that of dropped case of fixed sparsity pattern. As a result of increase of nonzeros, required construction time increases about 5 times and the application time increases about 2 times. But required iteration number of BiCGSTAB is reduced by 13 % and consequently the total computing time for linear system solution is also decreased by 10 %.

This indicates that the disadvantage from increase of the number of nonzeros can be compensated by better quality of preconditioner when it is used for enough number of time steps.

Table 5.4 Performance examination of SPAI sparsity structure determination algorithm for APR1400 quarter-core problem

	Fixed-dropped	Predict-0.02
SPAI nnz	4,678,944	15,480,350
BiCGSTAB iteration per step	358.1	311.1
SPAI construction time (s)	2.2	11.6
SPAI application time (s)	20.8	39.4
total BiCGSTAB time (s)	851.1	769.0

6 Numerical Analyses

6.1 SPERT III E-core RIA Experiments

6.1.1 Calculation Options and Basic Information

For all the calculations for the E-core, the MOC calculations are carried out with a ray parameter set identified by 0.05 cm ray spacing, and 16 azimuthal and 4 polar angles for the octant of the solid angle sphere. The scattering anisotropy is explicitly incorporated with the P2 scattering option of nTRACER. For the heat conduction calculation, each fuel pellet is divided into 15 equi-distance intervals and the cladding is divided into 2 intervals. The axial geometry of the neutronics calculation, which consists of 20 planes for the core, is used also in the T/H calculation. All the RIA tests are simulated for 0.3 sec. The default time step size for CMFD is set as 1 msec and the conditional update option is turned on. For the time section where the transient rod is ejected, 5 msec time steps are used with the conditional update option turned off. For the PKE solution, micro time steps of size 0.01 msec are used.

All calculations were performed on a heterogeneous computing cluster, which was equipped both CPUs and GPUs. For each calculation reported here, 100 CPU cores and 20 commodity GPUs mounted on 5 computing nodes were used. The detailed specifications of the computing cluster are given in Table 6.1.

Table 6.1 Computing cluster specifications

# of Nodes	5
CPU / Node	2 × Intel Xeon E5-2630 v4 (20 Cores, 2.4 GHz)
GPU / Node	4 × NVIDIA GeForce RTX 2080 Ti
Memory / Node	8 × 16 GB DDR4 RAM
Interconnect	Mellanox Infiniband (56 Gbps)

6.1.2 Core Property Calculation at Zero Power Conditions

The transient simulation results for the rod ejection tests of the E-core cannot match well with the measured data unless the steady-state results are reasonable. Thus, the reliability of the nTRACER model of E-core should be examined first for the initial steady state conditions. In this regard, the simulation results for the E-core are presented starting from the steady-state conditions in this section. The nuclear characteristics parameters of the E-core at zero power conditions were measured and reported. The critical CRA position and control rod worth were estimated during core loading procedure in 1966. The measurement was performed at both cold zero power (CZP, 21.11 °C) and hot zero power (HZP, 287.78 °C). In 1967, the reduced prompt neutron generation time was estimated, and the control rod worth was refined at CZP. Those core parameters were generated by the nTRACER steady-state calculations and compared with the measured data. The results are summarized in Table 6.2.

The calculated multiplication factor at the measured critical CRA position at both conditions are quite close to unity. The reactivity error is less than 130 pcm. Accordingly, the calculated critical CRA positions are within 1 cm of the measured positions. Note that the measured value of the effective delayed neutron fraction and prompt neutron generation time are not available because

Table 6.2 Nuclear characteristics parameters of the SPERT III E-core at zero power conditions

nuclear parameters	CZP		HZP	
	nTRACER	**experimental**	**nTRACER**	**experimental**
k_eff at critical CRA position	0.99870	1.00000	1.00118	1.00000
calculated critical CRA position (cm)	37.25	36.96	70.79	71.76
effective delayed neutron fraction (pcm)	744	N/A	740	N/A
prompt neutron generation time (μs)	17.3	N/A	17.2	N/A
reduced prompt neutron generation time (ms)	2.32	2.15	2.32	N/A
total excess reactivity ($)	13.4	13.1 $\pm$ 0.7	3.3	2.5
Differential CRA worth near critical ($/cm)	0.69	0.61	0.16	0.16

only the reduced neutron generation time are measurable. The calculated reduced prompt neutron generation time agrees with the measure one within 7.9 % which is smaller than the evaluated uncertainty 12 %. The calculated total excess reactivity at CZP also agrees well with the experimental data within the uncertainty range. At HZP, the difference between the calculated and measured data for the total excess reactivity is 0.8 $. Although the measurement is not provided, it can be judged that the difference would be within the uncertainty because the nTRACER value is quite close to the value of the Tripoli-4® Monte Carlo code which was 3.1 $ (Zoia and Brun 2016). The calculated differential CRA worth near the critical position is about 12 % higher than measurement at CZP while the agreement is very good at HZP.

The comparison of the calculated and measured values of the integral control rod worth at CZP is shown in Fig. 6.1. The agreement between the two is quite good over the entire CRA operation range. In 1969, the rod worth curves of the transient rod were generated at both CZP (21.11 °C) and HZP (260.0 °C) based on the experimental data for the static and low initial power transient tests (McCardell et al. 1969). The rod worth of the transient rod at insertion depth x was calculated as the reactivity difference from the zero insertion state as:

$$\frac{1}{\beta_{eff}}\left(\frac{1}{k_{eff}(x)}-\frac{1}{k_{eff}(0)}\right). \tag{6.1}$$

The effective delayed neutron used for reactivity normalization is 744 pcm for CZP and 739 pcm for HZP. The calculated transient rod worth agrees quite well with the experimental data for insertion less than 10 cm at both states as shown in Fig. 6.2. It is slightly overestimated for more insertion. This much difference can be caused by the transient rod worth calculation method and the minor assumptions introduced in the core geometry modeling.

6.1.3 Analysis of the RIA simulation results

The faithfulness of nTRACER modeling of E-core was confirmed by the calculated results for the zero power states. The five representative tests were selected out of dozens of RIA tests performed in E-core for validation of nTRACER. Each test is characterized by its own set of core inlet temperature, initial power level, and control rod conditions. The initial conditions and inserted reactivity are listed in Table 6.3. The precise initial power of the low initial power tests (Test 43, Test 60, Test 70) were not documented. 50 W was assigned to these tests as the initial power because this value was indicated as the nominal zero

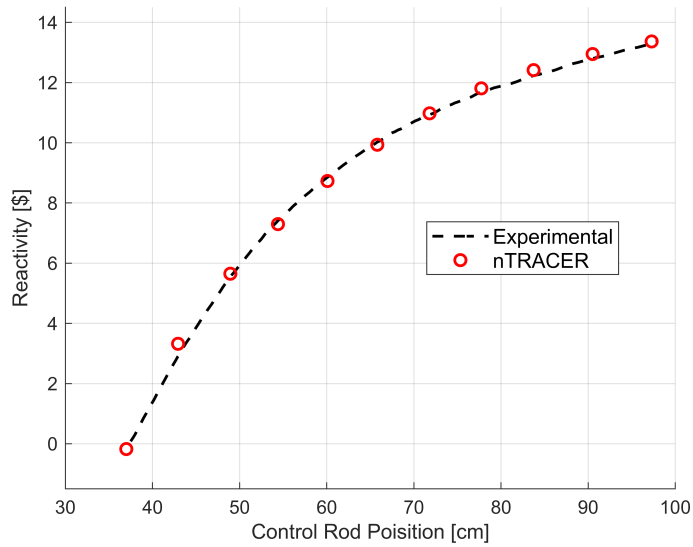


Figure 6.1 Control rod reactivity curve at cold zero power

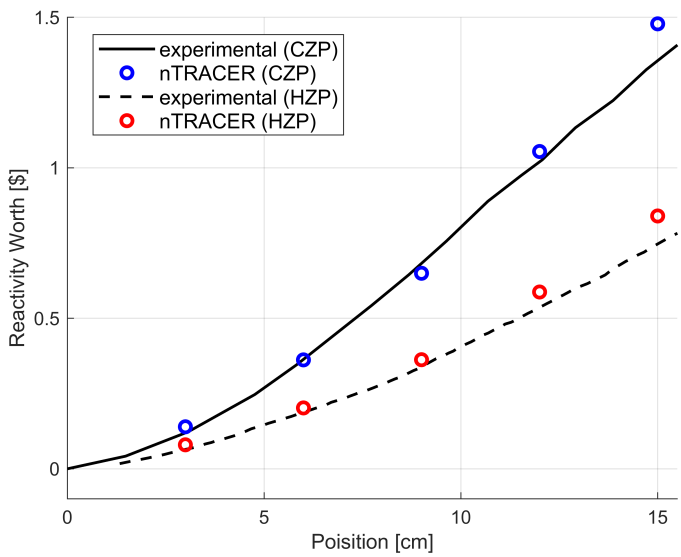


Figure 6.2 Transient rod reactivity curves

Table 6.3 Initial conditions of the E-core experiments

Test	Initial power (MW)	Inserted reactivity ($)	Inlet temperature (°C)	Flow rate (m/s)	Pressure (MPa)
43	∼0.05	1.21 ± 0.05	25.6	0	0.10
60	∼0.05	1.23 ± 0.05	260.0	4.3	10.34
70	∼0.05	1.21 ± 0.05	121.7	4.3	10.34
81	0.9 ± 0.1	1.17 ± 0.05	262.2	4.3	10.34
86	19 ± 1	1.17 ± 0.05	261.1	4.3	10.34

power level in the documentation(McCardell et al. 1969). Excluding the zero flow rate case of Test 43, the core inlet temperature, flow rate, and pressure specified in Table 6.3 were used as input data for simple T/H calculation. Because simple T/H solve cannot simulate zero mass flow, a small value (1 kg/s core mass flow) was used for Test 43.

All the transient events are initiated by displacing the transient rod. According to the reference (McCardell et al. 1969), the transient rod displacement is calculated assuming the constant design acceleration of 50.8 m/s 2 and the initial velocity of 0 m/s. The problem is, however, that the precise position of the CRA and the transient rod of the tests are not indicated in the documentations. To obtain the inserted reactivity, Fujita who carried out the SPERT analysis with the CASMO5/PARCS code system introduced the following three-step procedure for determination of the axial positions of the CRA and the transient rod (Fujita and Sakai 2019):

(1) First, search the CRA position which satisfies critical state with fully withdrawn transient rod. The effective delayed neutron ratio estimated

in this step is used to estimate the normalized reactivity in Steps (2) and (3)

(2) Search another CRA position at which the excess reactivity of the core is the same as the inserted reactivity of the target test with the transient rod kept withdrawn.

(3) Search the transient rod position that makes the core critical while the axial position of the CRA of Step 2 is retained.

This three-step procedure (denoted as '3-step') was taken in the nTRACER simulations of the low-initial power tests. In this procedure, the amount of reactivity insertion is assumed to be the excess reactivity calculated in Step (2). This assumption is appropriate for low-initial power tests. In the high-initial power tests (e.g. Test 86), however, the temperature distribution changes due to the difference in the axial position of the transient rod between Steps (2) and step (3) and this causes different feedback effect. This leads to an over-estimation of inserted reactivity. To mitigate this problem, the third step is modified, and the fourth step is added as follows (denoted as '4-step') for Test 81 and 86:

(3) Search the transient rod position to make the core critical maintaining the temperature distribution as well as the axial position of the CRA as those determined at Step (2).

(4) Search the critical position of the control rod with T/H feedback on keeping the transient rod position determined at step (3).

Since the T/H states of Steps (2) and (4) are different, the calculated transient rod worth cannot be exactly the same as the target value unless several iterations for rod position search are performed. However, it turned out

that the iteration was not necessary because the calculated rod worth obtained in one 4-step calculation match the target within 0.004 $.

The initial kinetic parameters and the axial positions of control rods for each transient calculation determined by the aforementioned procedure are listed in Table 6.4. The effect of different temperature distributions at different rodded states was negligible for Test 81 since the initial power itself is still low. On the other hand, for Test 86, the transient rod determined with the 3-step procedure is inserted 0.22 cm more than the 4-step procedure case. To examine this difference the transient simulation for Test 86 was performed with the two initial conditions.

Table 6.4 Initial kinetic parameters and axial positions of control rods of the E-core tests

	Effective delayed neutron fraction (pcm)	Prompt neutron generation time (μs)	Axial position (cm)	
			control rod assembly	transient rod
Test 43	741.7	17.1	26.57	12.02
Test 60	739.1	17.0	52.62	18.77
Test 70	741.5	17.0	30.10	12.67
Test 81 (3-step)	739.0	17.1	53.51	18.60
Test 81 (4-step)	739.0	17.1	53.51	18.60
Test 86 (3-step)	737.0	17.1	60.74	21.73
Test 86 (4-step)	737.0	17.1	60.67	21.51

The magnitude of time of peak power, the energy release upto the peak time, and the reactivity compensation at peak power are well documented in Reference (McCardell et al. 1969). Especially, the reactor period is also documented for cold-startup tests and hot-startup tests. The nTRACER simulation results are compared with the experimental data in Table 6.5. The calculated

reactivity compensation at peak power is defined as the reactivity difference between initial reactivity insertion and the reactivity at the calculated peak power. The calculated reactor period is determined by an exponential curve fitting of the core power history data sampled for 0.02 seconds immediately after the reactivity insertion was finished. By comparing the two cases of Test 86, the suitability of the 4-step procedure, which incorporates the feedback effect on the transient rod position, is confirmed. The calculated peak power of the 3-step procedure is out of the uncertainty range while that of the 4-step procedure is well-matched with the experimental data. Except for the Test 86 3-step case, the calculated peak power, energy release, reactivity compensation, and reactor period match well with the experimental data within the uncertainty ranges.

The calculated peak time for several tests are, however, out of the uncertainty range. The calculated peak time tends to be slightly later than the measured peak time. As mentioned earlier, the displacement of the transient rod was calculated with a constant acceleration. However, the actual acceleration could not be constant so that the actual time for ejection might be slightly different from the calculation one with the constant acceleration. This would lead to a slight difference in the peak time between the measured and calculated cases. Also, for the low initial power cases, the uncertain initial power of the experiment which was set to 50 W in the calculation would cause the uncertainty in peak time. If the measured data are shifted such that the measured peak time coincides with the calculated one, excellent overall matches of the measured and calculated values are observed as presented in Fig. 6.3 through Fig. 6.5. These plots demonstrate the simulations are indeed quite good as long as the ejection time is matched.

Table 6.5 Summary of the nTRACER calculations for E-core RIA tests

	Test 43		Test 60	
	Experiment	Calculation	Experiment	Calculation
Peak power (MW)	280 ± 42	251	410 ± 41	443
Peak time (sec)	0.230 ± 0.006	0.219	0.227 ± 0.005	0.229
Released energy to peak time (MW s)	6.0 ± 1.0	5.5	8.5 ± 1.1	9.3
Compensated reactivity at peak time ($)	0.22 ± 0.02	0.22	0.24 ± 0.03	0.24
Reactor period (ms)	10.0 ± 0.2	9.8	9.7 ± 0.19	9.7
	Test 70		Test 81	
	Experiment	Calculation	Experiment	Calculation
Peak power (MW)	280 ± 42	299	410 ± 41	443
Peak time (sec)	0.20 ± 0.01	0.223	0.135 ± 0.003	0.139
Released energy to peak time (MW s)	6.3 ± 1.1	6.6	7.8 ± 1	7.6
Compensated reactivity at peak time ($)	0.22 ± 0.02	0.22	0.18	0.19
Reactor period (ms)	10.3 ± 0.2	10.1	-	-
	Test 86 (3-step)		Test 86 (4-step)	
	Experiment	Calculation	Experiment	Calculation
Peak power (MW)	610 ± 60	708	610 ± 60	633
Peak time (sec)	0.110 ± 0.005	0.113	0.110 ± 0.005	0.113
Released energy to peak time (MW s)	17 ± 2	16.68	17 ± 2	15.4
Compensated reactivity at peak time ($)	0.22	0.21	0.22	0.21

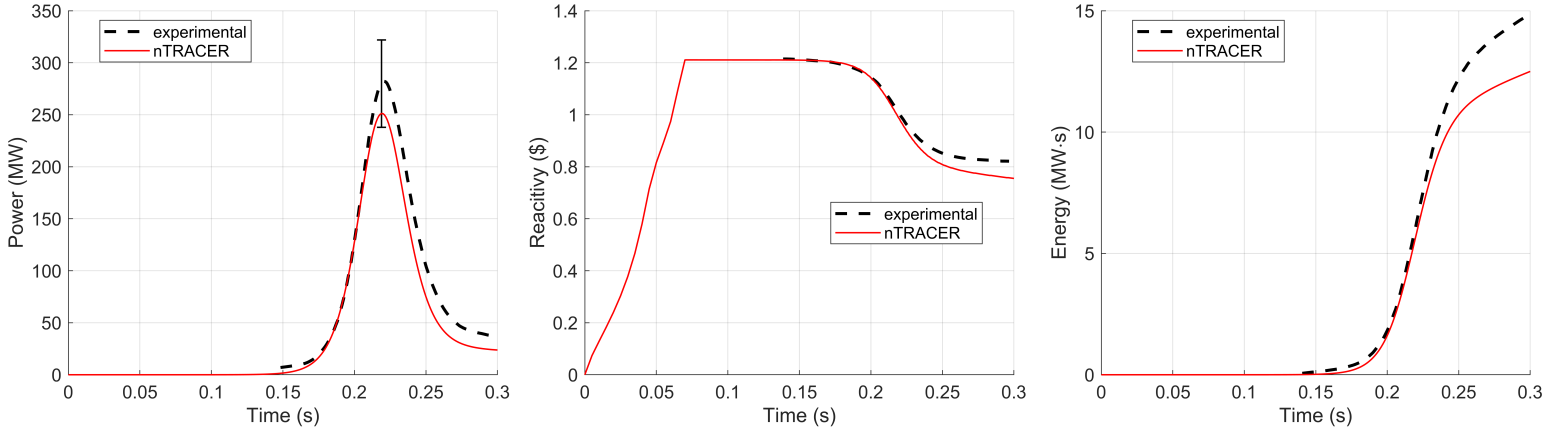


Figure 6.3 Evolution of power, reactivity, and energy for Test 43 compared with shifted experimental data

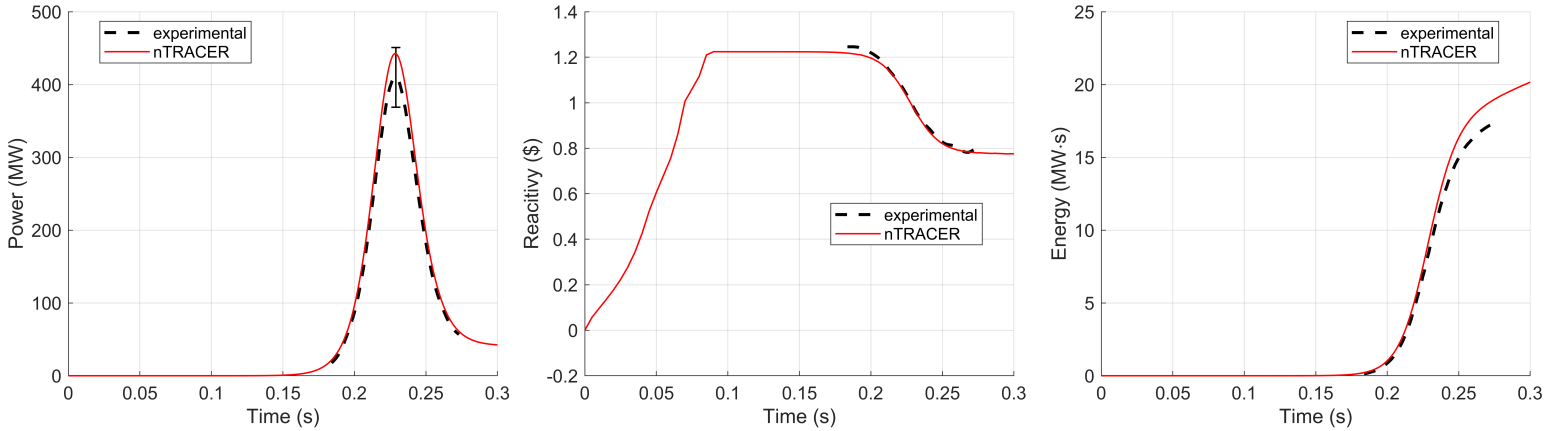


Figure 6.4 Evolution of power, reactivity, and energy for Test 60 compared with shifted experimental data

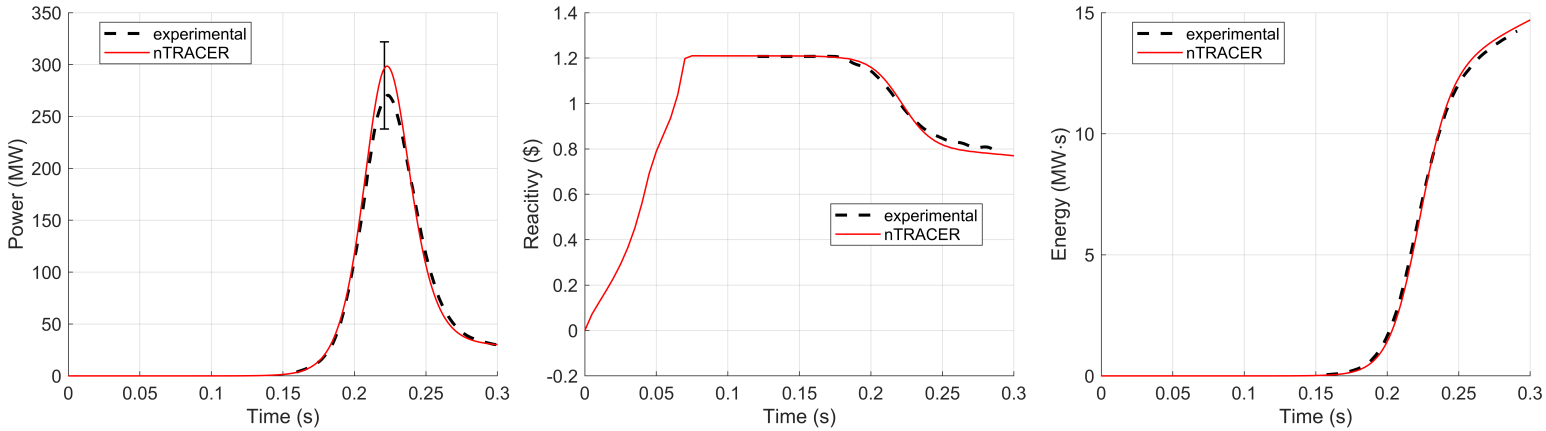


Figure 6.5 Evolution of power, reactivity, and energy for Test 70 compared with shifted experimental data

The transient behavior of core power, reactivity, and net energy release are shown in Fig. 6.6 through Fig. 6.11. As shown in Fig. 6.10, the consequence of the reactivity overestimation is observed for Test 86 configuration obtained with the 3-step procedure. Except for this case, the agreement between the experimental data and the simulation results appears to be good. Though there are uncertainties due to the assumption for the transient rod displacement, the overall shape of the calculated power history and reactivity matches well with the experimental data. Consequently, the core power and reactivity behaviors appear similar to the experimental data even after the power burst except for Test 86 for which the post burst core power and reactivity are overestimated. The reason for the relatively large difference for the high initial power case might be due to the simplified T/H module used in the nTRACER which cannot consider properly the cross flow between sub-channels. Note that the moderator feedback would have nontrivial effects on the CRA ejection at high power conditions.

The change in pin-wise power distribution and fuel rod average temperature distribution with time in Test 60 are shown in Fig. 6.12 and Fig. 6.13. Although there is no reference data for these detailed state parameters of the core, the nTRACER results show correct qualitative behaviors. As shown in Fig. 6.12, the relative power of 16-rod assemblies at the center of the core is increased as a result of ejection of the transient rod. This increase in power shows quire heterogeneous distribution. Also, as the heat accumulates, the fuel rod temperature distribution changes from an initial flat distribution to a heterogeneous distribution as shown in Fig. 6.13. These distributions reveal the advantage of the direct whole core transient calculation resulting in high fidelity transient solutions.

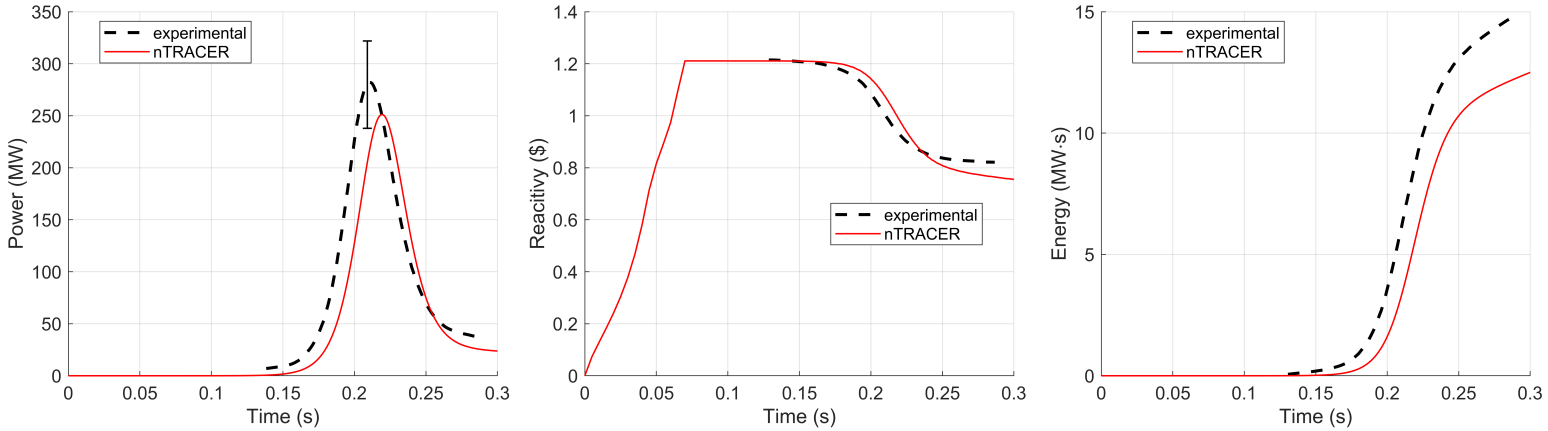


Figure 6.6 Evolution of power, reactivity, and energy for Test 43

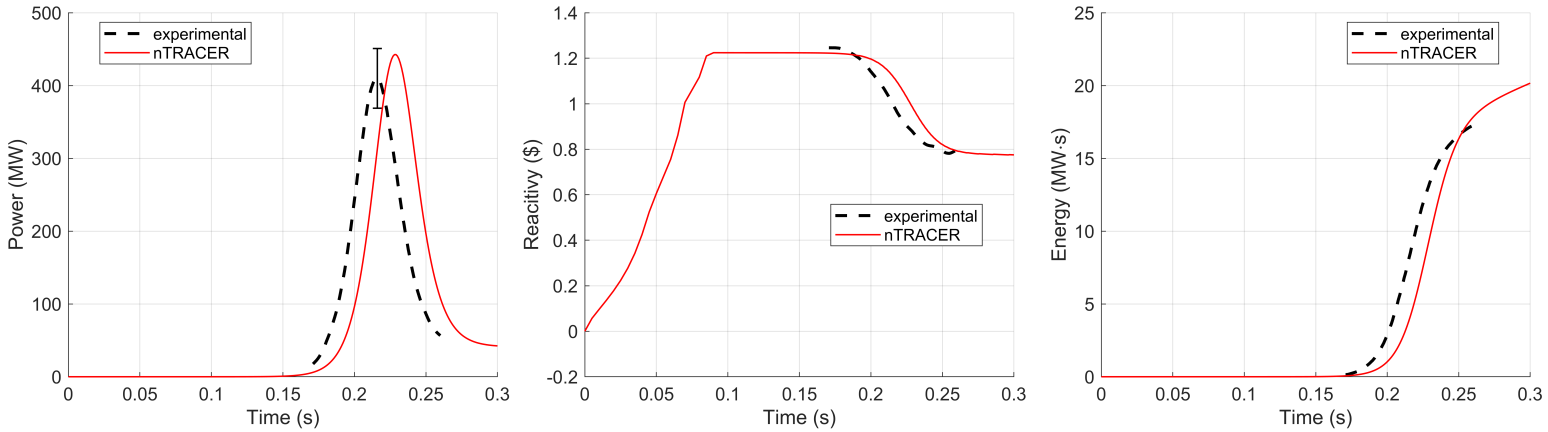


Figure 6.7 Evolution of power, reactivity, and energy for Test 60

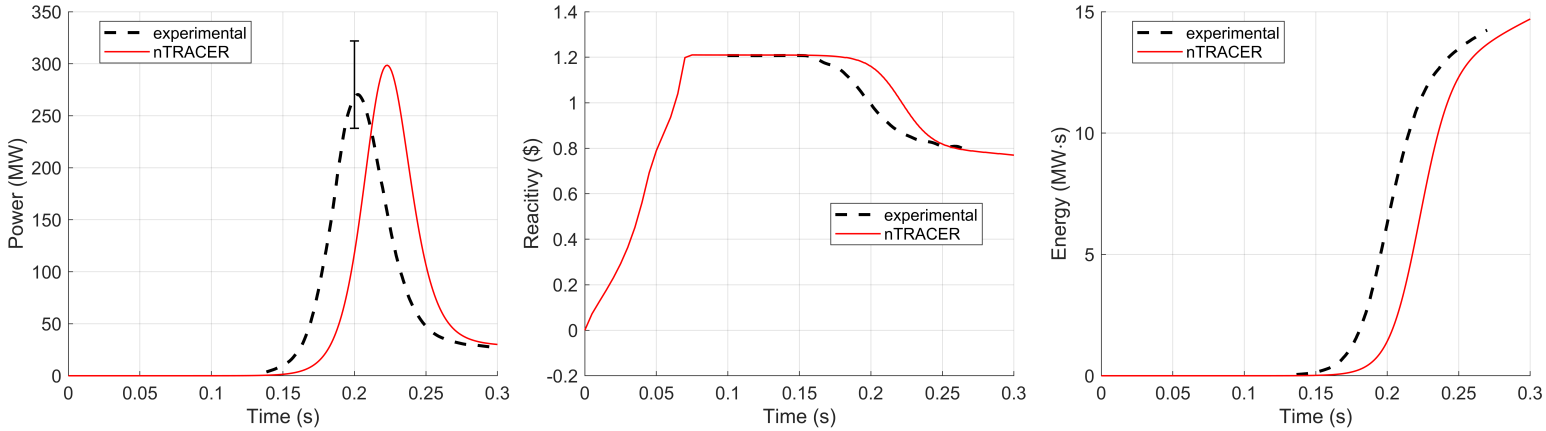


Figure 6.8 Evolution of power, reactivity, and energy for Test 70

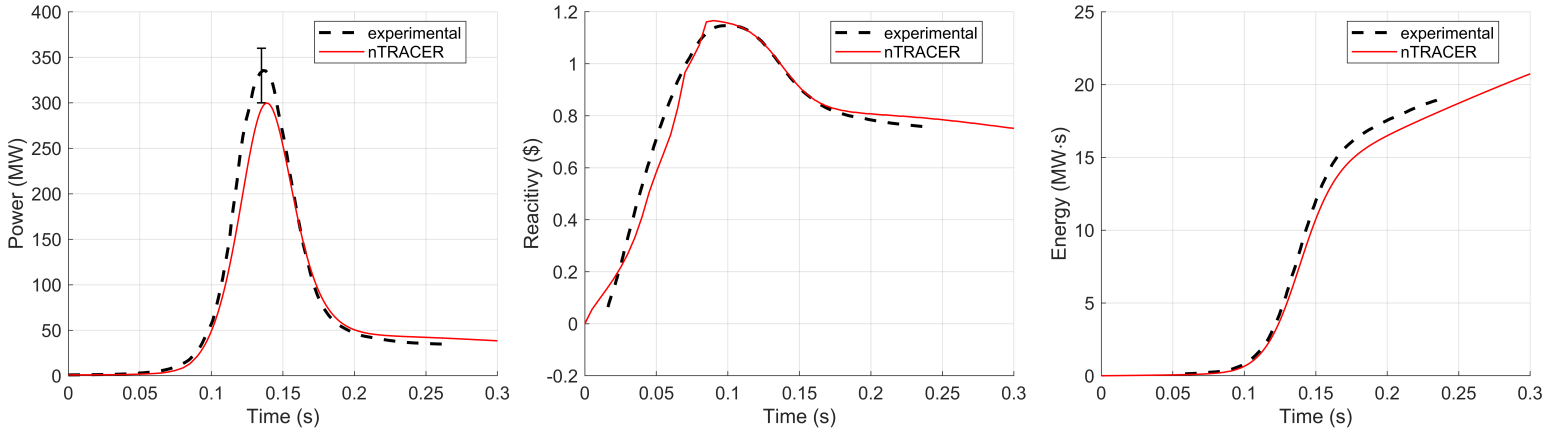


Figure 6.9 Evolution of power, reactivity, and energy for Test 81

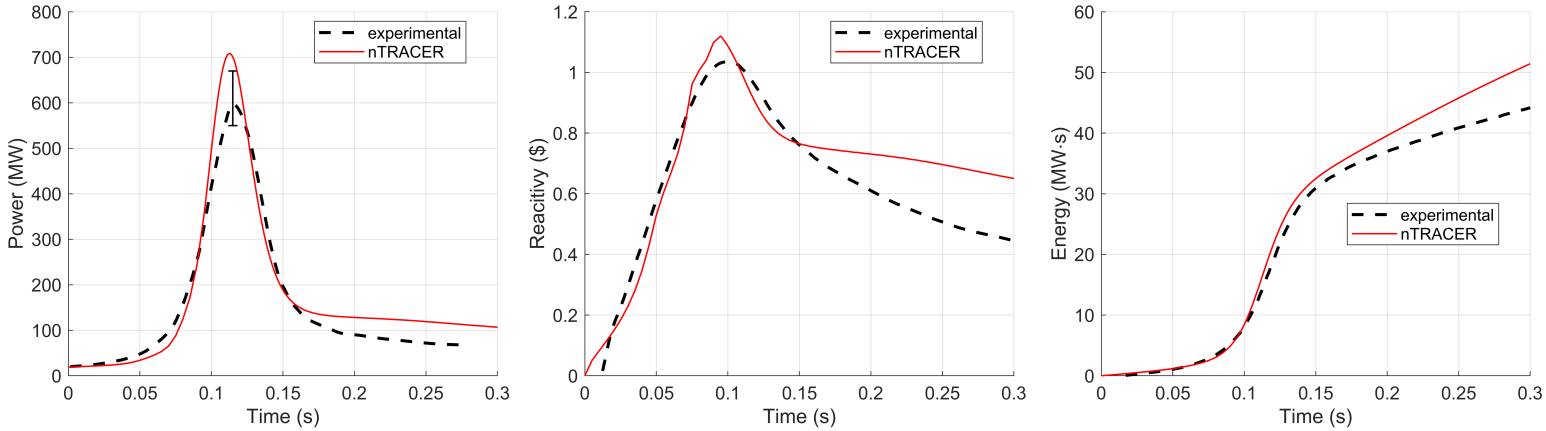


Figure 6.10 Evolution of power, reactivity, and energy for Test 86 (3-step)

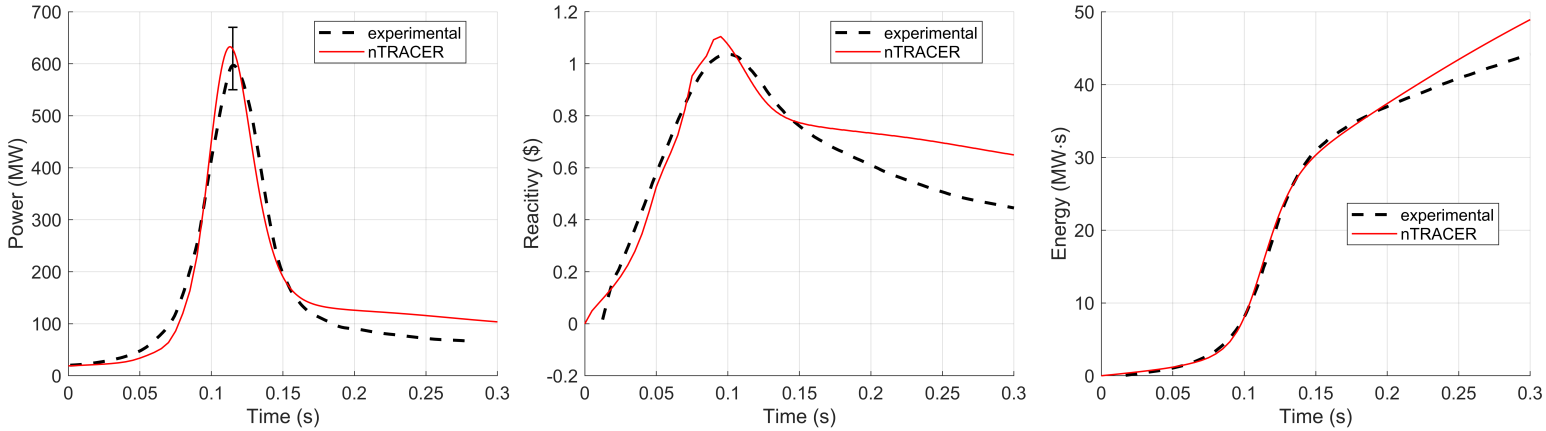


Figure 6.11 Evolution of power, reactivity, and energy for Test 86 (4-step)

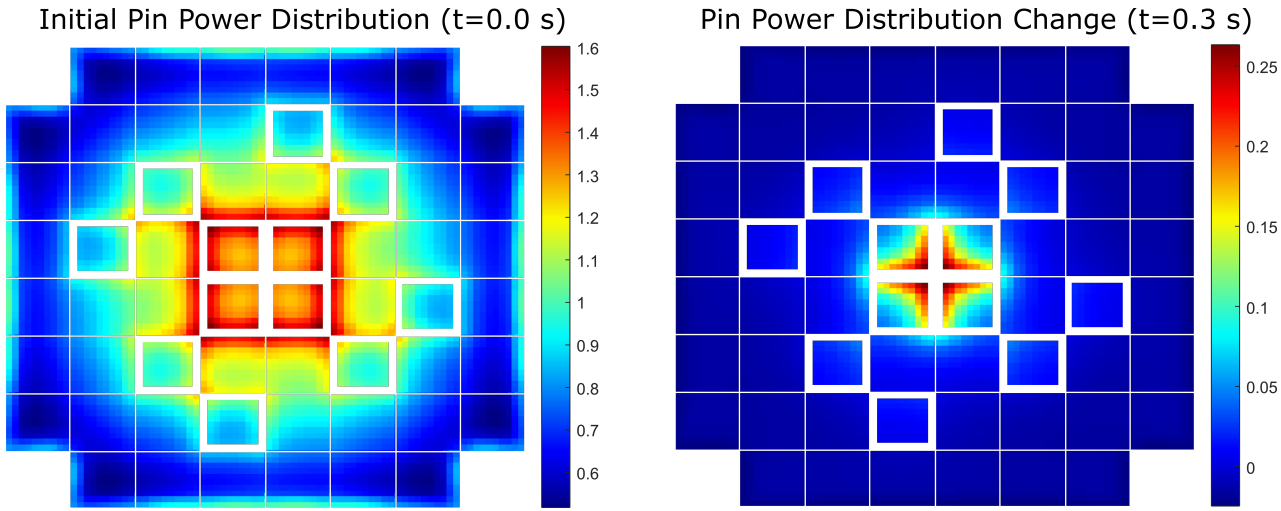


Figure 6.12 Initial pin-wise power distribution (left) and change with time (right) in Test 60

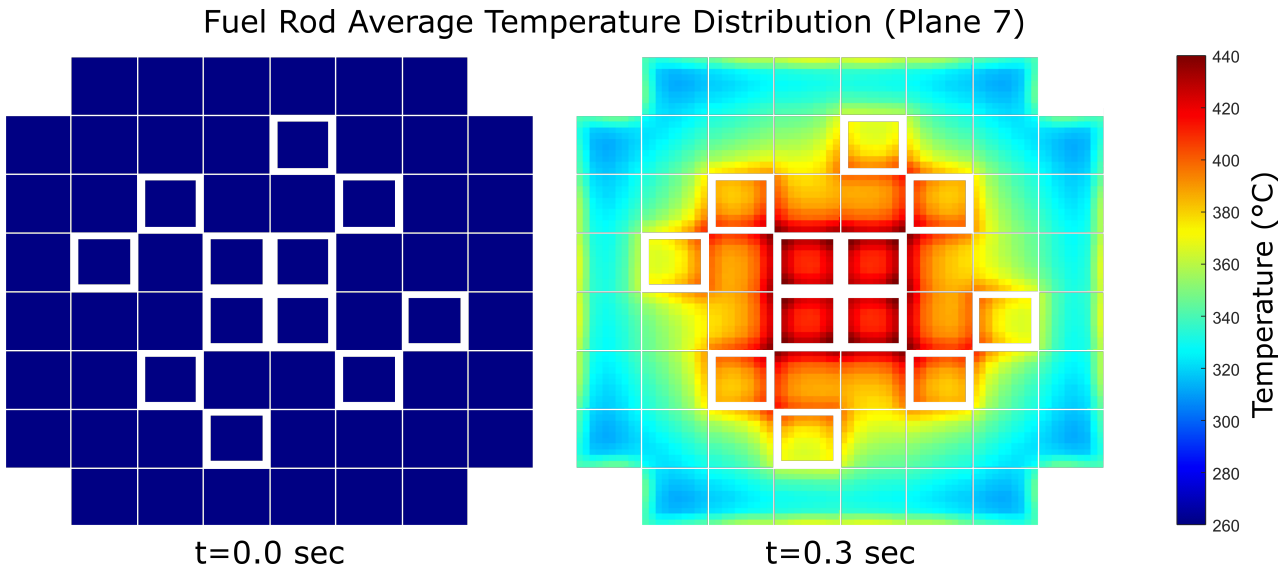


Figure 6.13 Variation of the fuel rod average temperature distribution of plane 7 (peak power plane) with time in Test 60

Table 6.6 Temperature conditions used for assembly-wise cross section generation

	Nominal condition	Branch condition
Fuel temperature (°C)	260.0	560.0
Moderator temperature (°C)	260.0	265.0

To demonstrate the advantages of direct whole core transient calculation, a two-step calculation for Test 60 was performed with a two-group nodal code that is based on a semi-analytic nodal method (SANM) within the CMFD framework (Yoon and Joo 2008). The assembly-wise homogenized cross sections were generated by nTRACER. The fuel temperature and moderator temperature, which were used in nTRACER calculations, are listed in Table 6.6. The effective delayed neutron parameters and two-group condensed neutron velocity generated by nTRACER were used in the nodal calculation and were considered to be independent in time and space. The time step size for the nodal calculation was set to 1 msec. The radial heat conduction solver and the simple T/H solver used for nTRACER calculation were also used in the nodal calculation. In order to equalize the amount of reactivity insertion, the positions of the CRA and the transient rod for the nodal calculation were also determined in the same manner as nTRACER.

The nodal calculation results compared with the experimental data and the nTRACER results are given in Table 6.7. The calculated peak power and the compensated reactivity at the peak time of the nodal calculation match well with the experimental data within the uncertainty range and the difference between calculated results for these parameters are small compared to the experimental uncertainty. However, other calculated parameters of the nodal

Table 6.7 Comparison of nodal calculation

	Nodal	Exp. data		nTRACER	
			Diff.		Diff.
Peak power (MW)	448	410 ± 41	38	443	5
Peak time (sec)	0.242	0.227 ± 0.005	0.015	0.229	0.013
Released energy to peak time (MW s)	10.0	8.5 ± 1.1	1.5	9.3	0.7
Compensated reactivity at peak time ($)	0.23	0.24 ± 0.03	-0.01	0.24	-0.01
Reactor period (ms)	10.6	9.7 ± 0.19	0.9	9.7	0.9

calculation are out of the uncertainty range. Especially large difference was observed for the reactor period. As a result, unlike nTRACER, the slope of power evolution of the nodal calculation was significantly different from the experimental data as shown in Fig. 6.14. The slope is determined by the ejected rod worth and prompt neutron generation time. Since the ejected rod worth of the nodal calculation was set equal to that of nTRACER, it becomes obvious that spatial homogenization and energy condensation caused such differences in the flux shape and this led to inaccurate prompt neutron generation time.

6.1.4 Computing Time Results

The computing time for each case is listed in Table 6.8. The computing time difference between each case was not significant. Only Test 86 required slightly more computational cost for T/H coupling due to its high initial power. The computing time for the steady-state calculations was approximately 7

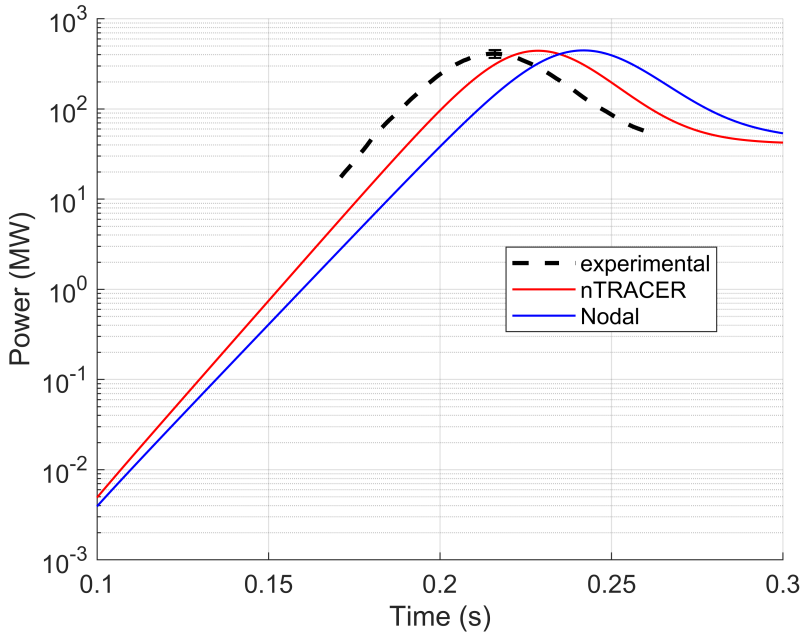


Figure 6.14 Comparison of nodal code power evolution for Test 60

minutes. For the transient calculations, the computing time for the time steps involving transport (MOC) updates was approximately 3 minutes, while it was approximately 1 minutes for the time steps without transport updates. The typical total computing time for each case was around 7 hours. Although nTRACER employs GPU computing techniques, it is not yet general to utilize GPUs in the direct whole core transport calculations of the other code. In this regard, an additional calculation for Test 60 was performed with the CPU version of nTRACER using 320 CPU cores (Intel Xeon E5-2640 v3, 2.6 GHz) as the reference for other CPU based codes. The computing time for both steady-state and time steps involving transport updates was approximately 30 minutes, while it was approximately 6 minutes for the time steps without transport update. The total computing time was around 47 hours.

Table 6.8 Computing time for simulations of RIAs in E-core

	CPU version computing time (s)	GPU version computing time (s)	Ratio (CPU / GPU)
SGFSP	9,753	532	18.3
MOC	48,057	4,472	10.7
CMFD	102,419	11,969	8.6
TH	2,140	2,378	0.9
ETC	6,913	5,110	1.4
Total	169,270	24,462	6.9

Table 6.9 shows a detailed computing times for each calculation component of test 60 simulation. Computationally intensive components such as SGFSP, MOC, and CMFD shows significant speed-up when using GPU version. On the other hand, other components that are not accelerated by GPU computing because of its complicated memory access shows similar computing time in both version of nTRACER. When comparing the overall computational time, the GPU version was 7 times faster than the CPU version. Since the computational environment used for each case is different, so the efficiency of the two versions cannot be directly compared. But it is possible to compare the price-to-performance. The price per node of the computing facility used for calculations of the GPU version is 15 million ₩. Total 5 nodes are used for calculations of GPU version the total price of computing facilities is 75 million ₩. On the other hand, total 20 nodes were used for calculation of CPU version, and the price per node is 7 million ₩, and a total of 140 million ₩ is used. As a result, the GPU verson shows about 13 times higher price-to-performance than the CPU version.

Table 6.9 Computing time comparison for simulation of test 60

	CPU version computing time (s)	GPU version computing time (s)	Ratio (CPU / GPU)
SGFSP	9,753	532	18.3
MOC	48,057	4,472	10.7
CMFD	102,419	11,969	8.6
TH	2,140	2,378	0.9
ETC	6,913	5,110	1.4
Total	169,270	24,462	6.9

6.2 APR1400 Full Core Analysis

The main target of nTRACER is operating PWRs. Therefore, applicability of direct whole core transient capability of nTRACER to operating PWR should be validated. Especially due to the asymmetry in most of the transient situation, full core analysis is required which requires far heavier computational burden than analyses with symmetry. A hypothetical super-prompt critical RIA in HZP core problem based on the 3D core problem in the benchmark suite (Yuk and Cho 2019), which is based on APR1400, is used for validation of applicability to commercial reactor. APR1400 contains 9 types of fuel assemblies which are distinguished by the number of burnable absorber, the enrichment of fuel rod, and its arrangement. The specifications of fuel assembly types are listed in Table. 6.10. The fuel assembly configuration is shown in Fig. 6.15 and the control rod assembly layout is shown in Fig. 6.16 in quarter symmetry. More detailed information can be found in the benchmark specification report(Yuk and Cho 2019).

Table 6.10 Fuel assembly types in APR1400

Assembly type	Fuel rod enrichment (wt%)	Number of fuel rods per FA	Number of burnable absorbers per FA
A0	1.71	236	-
B0	3.14	236	-
B1	3.14 / 2.64	172/52	12
B2	3.14 / 2.64	124 / 100	12
B3	3.14 / 2.64	168 / 52	16
C0	3.64 / 3.14	184 / 52	-
C1	3.64 / 3.14	172 / 52	12
C2	3.64 / 3.14	168 / 52	16
C3	3.64 / 3.14	120 / 100	16

At the initial state, the control rod bank 1∼5 are fully inserted and the shutdown bank A, B are fully withdrawn from the core. The core set to critical by adjusting the boron concentration in the moderator. The control rod assembly which is marked by circle in Fig. 6.16 is assumed to be ejected within 0.1 s at a constant speed. Note that the control rod assembly ejection is assumed to be occurred in only one of the quadrant without symmetry. The rod worth is set to 1.11 $ by adjusting B_4C concentration of control rods involved in control rod bank 2.

The fuel pin cell is discretized into sufficient number of FSRs to incorporate the intra-cell flux distribution. The fuel region is divided into 5 annular FXRs and each FXR is divided into 8 azimuthal sectors, FSRs. The moderator region in the fuel pin cell is divided into 4 annular FXRs and 32 FSRs. The ray spacing was set to 0.05 cm and the numbers of azimuthal angles and polar angles per octant of the solid angle sphere were set to 16 and 4, respectively. For axial modeling, the core is divided into 24 planes and the spacer grids are

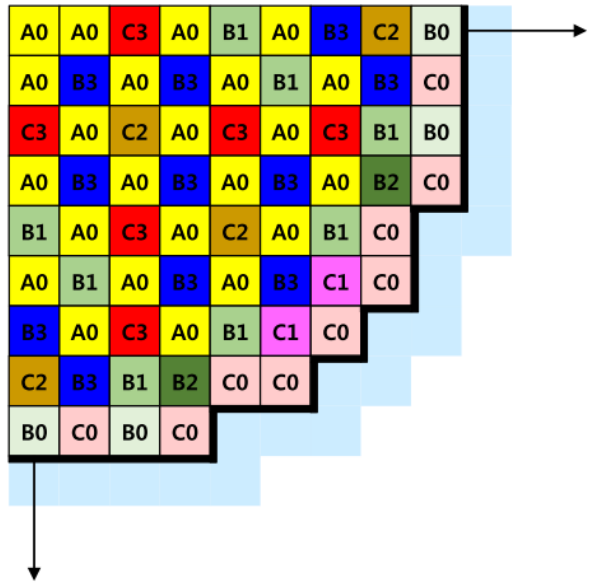


Figure 6.15 Radial configuration of APR1400 core

neglected. The transient event was simulated up to 1 s with 125 macro time steps. The macro time step sizes used in each interval of simulation are listed in Table 6.11. The micro time step size of 0.1 ms is used for PKE calculation of IQS.

Table 6.11 Time step sizes for APR1400 full core analysis

Interval	time step size (ms)	Description
~0.1 s	5	Rod ejection
0.1~0.4 s	20	Exponential power growth
0.4~0.8 s	5	Power burst
0.8~1.0 s	20	After burst

The full core simulation was run on the small cluster with 24 GPU cards NVIDIA(Tesla V100 32GB). The Tesla line GPU card is specially designed for scientific computing which has relatively expensive price. But its large memory (32 GB) is required for full core simulation because, so far, the memory size

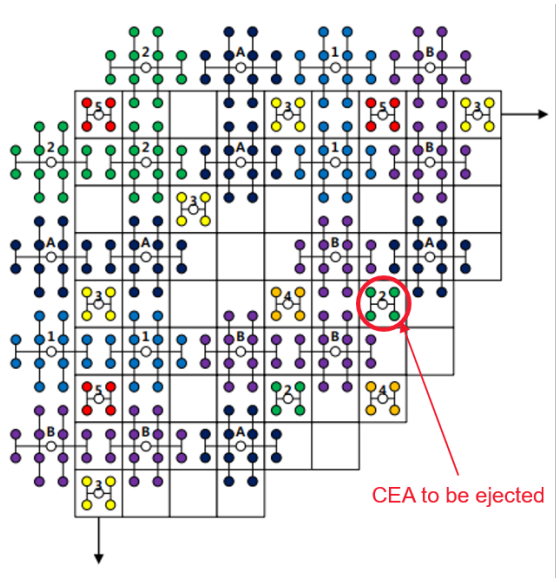


Figure 6.16 Radial configuration of control rod assembly banks

of consumer grade GPU is about 10 GB which is not enough to store data of single plane of full core geometry. The detailed specifications of computing cluster are listed in Table 6.12

Table 6.12 Specifications of GPU computing cluster used for APR1400 full core calculation

The number of nodes	6
CPU / node	2 × Intel Xeon Gold 6258 (28 cores, 2.7 GHz)
GPU / node	4 × NVIDIA Tesla V100 32GB
Memory / node	751 GB DDR4 RAM
Interconnect	Mellanox infiniband (100Gbps)

The fractional core power level history, reactivity history, and history of total released energy from the core are shown in Fig. 6.17. Fig. 6.18 shows the normalized pin power distribution change occurring during the transient. Before the transient, the power is symmetrically distributed. After the sudden

ejection of control rod assembly, the power near the ejected region rose sharply. The total run time was 19 hours. Out of 125 time steps, MOC calculation was invoked in 86 time steps, and SGFSP calculation was invoked in 85 time steps.

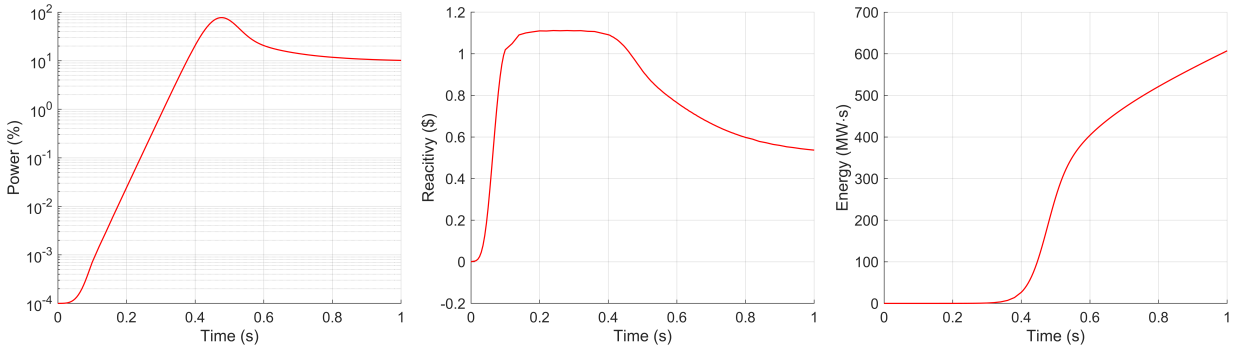
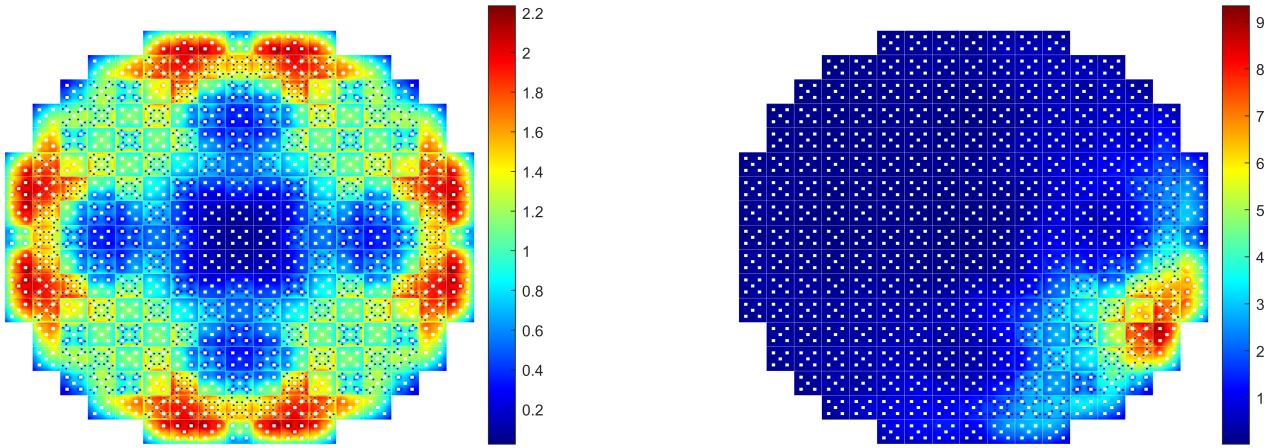


Figure 6.17 Evolution of power, reactivity, and energy for APR1400 full core calculation



(a) at 0.0 s

(b) at 0.2 s

Figure 6.18 Pin-wise power distribution at different times

7 Conclusion

In order to overcome the existing limitations due to the enormous computational burden, the direct whole core transient calculation capability of nTRACER was enhanced by improvement of methodology and utilization of GPGPU programming. The major improvements of methodology are multi-level method composed of MOC, CMFD, and PKE and adaptive solution method. The GPGPU programming is applied to computationally intensive components such as MOC and CMFD. In this work, optimization for GPU environment of transient CMFD linear solver was intensively performed. It was shown that direct whole core calculation of fast transient event such as control rod ejection in operating PWR can be finished within a day when run on relatively small heterogeneous cluster containing few dozens of GPUs.

The multi-level method doesn't often calculate the slow change in fine spaces, but accurately calculates the overall rapid change in large space, so the computational burden is alleviated. The transport calculation, which involves both MOC and SGFSP, are done intermittently. They are invoked only when the condition of the core changes significantly. The change of 1 group condensed absorption cross section and fuel temperature change were used as invoking criterion for MOC and SGFSP, respectively. However, the cross section based MOC invoking criterion has a limitation that it cannot reflect the quantitative effect of MOC update. New MOC invoking criterion that uses fine mesh residual

norm is devised in this work. Fine mesh residual at each time step is calculated with assumption of constant normalized leakage. The effectiveness of new MOC invoking criterion was evaluated for HZP control rod ejection in 5×5 fuel assembly problem. When compared to previous criterion, new criterion can estimate the effect of MOC update more accurately. As a consequence, when similar number of time steps are used for two cases with different MOC invoking criterion, peak power relative error is reduced by about 80 %. In the case of CMFD calculation, flux factorization is applied, resulting in a coupling with PKE calculation. Comparative study was done to find an optimal flux factorization option for direct whole core calculation methodology of nTRACER. Among various flux factorization methods, IQS with rescaling was most efficient. For pure neutron kinetics problem, it showed almost equivalent accuracy with PCQS, but for the problem involving T/H feedback, IQS-rescale showed better accuracy due to its tight coupling with T/H calculation. When using IQS-rescale for control rod ejection problem in SPERT III E-core, the peak power relative error is reduced by 75 % when compared to PCQS result, while the computing time increased only 12 %. When compared to BE results without any flux facroization, IQS-rescale reduces the error of the total released energy from 1.4 MWs to 0.1 MWs.

To prevent unnecessarily frequent CMFD calculations and ensure a certain level of accuracy, the time step sizes used for CMFD calculation are determined using adaptive time step control algorithm. Adaptive time step control is Accomplished by derivation of local error model that occur over each macro time interval. The error model is composed of four different types of errors. There are errors in neutronics calculations as well as error that could occur in the coupling with T/H calculation. The errors are estimated by calculating directly

by definition or by taking the difference between two different calculations. The additional calculations used for error estimation are PKE with higher order interpolation and T/H calculation with smaller time step which doesn't introduce significant overhead. A practical time step control algorithm was established with various stabilization techniques such as restart. The efficiency of adaptive time step control was examined for the HZP control rod ejection in 5×5 fuel assembly problem. The error was controlled under the tolerance in intervals where rapid change occurs as intended. When compared to the fixed time step size case using similar number of time step, adaptive time step control algorithm reduces the peak power relative error about 80 % without computing time increase. For long time simulation, the truncation error of T/H calculation becomes a main contribution of local error. In this case, multi T/H step scheme which separates the time step for neutronics and T/H calculation was proven to be efficient.

In order to obtain the maximum performance of GPU computing, CMFD linear system solution optimization was performed. First, comparative study of two different CMFD formulation was performed (group-major ordering, and multi-group direct solution). Unlike steady-state problem, fission source terms are included in CMFD matrix, so the convergence of group-major ordered CMFD solution was not fast enough. Furthermore, since the multi-group matrix is separated into 1 group matrices, the performance of GPU which is specialized in processing large amounts of data at once. So multi-group direct solution was employed as a CMFD formulation. In this formulation a large linear system for whole group is solved directly using Krylov linear system solver such as BiCGSTAB. The convergence of BiCGSTAB is determined by the preconditioner. The ILU preconditioner which is usually used as a

preconditioner for CMFD linear system solution proven to be inefficient when using GPU computing because its application is naturally sequential process. Instead of ILU, SPAI preconditioner is employed because it is suitable for massive parallel application. It approximates the inverse of CMFD matrix as a sparse matrix which has a prescribed sparsity structure. SPAI prediction algorithm for sparsity structure determination was devised. It predicts a distribution of elements of inverse matrix that has a magnitude larger than given drop criteria. For a hypothetical control rod ejection in quarter core of APR1400, when using drop tolerance of 2 %, using predicted sparsity pattern requires 13 % shorter CMFD linear system computing time than simple fixed sparsity pattern case.

The accuracy and performance of the new transient calculation module was verified through SPERT III E-core calculations. Most of the nTRACER transient results showed good agreement with the experimental data. The calculated results of the important parameters such as peak power, net energy release up to the peak time, and reactor periods were all within the experimental uncertainty. Only the calculated peak time exceeded slightly the experiment uncertainty. This was because the peak time is sensitive to the initial power and the displacement of the transient rod, which were not documented precisely. The comparison with a conventional two-step procedure results revealed the advantages of direct whole core transient calculation in that the two-step solution involved large differences exceeding the uncertainty range in major experimental parameters such as reactor period, although the nodal calculation showed good agreement with the measured peak power value. In terms of computing time, about seven-times speed-up is obtained when compared to the previous CPU based transient capability using the same number of time

steps and run on the cluster containing 320 CPU cores. When compare the cost effectiveness of two transient capabilities, new capability has 13 times effective than previous one.

The practicality of new direct whole core transient capability when applied to the realistic operating core analysis was confirmed through APR1400 calculation. The RIA in APR1400 is simulated for full-core geometry up to 1 s using 125 time steps, and finished in 19 hours when run on the cluster containing 24 GPUs. The high solution fidelity of new transient capability that is verified against the experimental data and the reasonable computing time of a few dozens of hours for a simulation of RIA in operating PWR on an affordable GPU cluster demonstrate the possibility of utilization of direct whole core transient in reactor design field. Considering the fact that the performance of GPU continues to grow even now, the direct whole core transient calculation approach would be more popular due to its necessity for high fidelity reactor analysis.

Bibliography

Boyarinov, V, P Fomichenko, J Hou, K Ivanov, A Aures, W Zwermann, and K Velkov, 2016: Deterministic time-dependent neutron transport benchmark without spatial homogenization (c5g7-td). *Nuclear Energy Agency Organisation for Economic Co-operation and Development (NEA-OECD), Paris, France*.

Cahalan, JE, and KO Ott, 1973: Delayed neutron data for fast reactor analysis. *Nuclear Science and Engineering*, **50**, 208–215.

Caron, D, S Dulla, and P Ravetto, 2017: Adaptive time step selection in the quasi-static methods of nuclear reactor dynamics. *Annals of Nuclear Energy*, **105**, 266–281.

Cho, Jin-Young, Kang-Seog Kim, Chung-Chan Lee, Han-Gyu Joo, Won-Sik Yang, TA Taiwo, and J Thomas, 2005: Transient capability for a moc-based whole core transport code decart. *Transactions of the American Nuclear Society*, **92**, 721–722.

Choi, Namjae, Junsu Kang, and Han Gyu Joo, 2018: Preliminary performance assessment of gpu acceleration module in ntracer. *Transactions of the Korean Nuclear Society Autumn Meeting, Yeosu, Korea*.

Choi, Sooyoung, Azamat Khassenov, and Deokjung Lee, 2016: Resonance self-shielding method using resonance interference factor library for practical lattice physics computations of lwrs. *Journal of Nuclear Science and Technology*, **53**, 1142–1154.

Dugone, J, 1965: *SPERT III reactor facility: E-core revision*. Technical report. Phillips Petroleum Co., Idaho Falls, Idaho. Atomic Energy Div.

Fujita, Tatsuya, and Tomohiro Sakai, 2019: Analysis of the spert-iii e-core experiment using casmo5/trace/parcs based on jendl-4.0 and endf/b-vii. 1. *Journal of Nuclear Science and Technology*, **56**, 553–571.

Gehin, Jess C, 1992: *A quasi-static polynomial nodal method for nuclear reactor analysis*. Technical report. Oak Ridge Inst. for Science and Education, TN (United States); Massachusetts ...

Grote, Marcus J, and Thomas Huckle, 1997: Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing*, **18**, 838–853.

Hursin, Mathieu, Thomas J Downar, and Brendan Kochunas, 2012: Analysis of the core power response during a pwr rod ejection transient using the parcs nodal code and the decart moc code. *Nuclear science and engineering*, **170**, 151–167.

Hursin, Mathieu, Thomas J Downar, Joo Il Yoon, and Han Gyu Joo, 2016: Assessment of assembly homogenized two-steps core dynamic calculations using direct whole core transport solutions. *Annals of Nuclear Energy*, **87**, 356–365.

Joo, Han Gyu, Jin Young Cho, Kang Seog Kim, Chung Chan Lee, and Sung Quun Zee, 2004: Methods and performance of a three-dimensional whole-core transport code decart.

Joo, Han Gyu, and Thomas J Downar, 1996: An incomplete domain decomposition preconditioning method for nonlinear nodal kinetics calculations. *Nuclear science and engineering*, **123**, 403–414.

Jung, Yeon Sang, and Han Gyu Joo, 2015: Investigation of conditional transport update in method of characteristics based coarse mesh finite difference transient calculation. *PHYSOR 2014*.

Jung, Yeon Sang, Cheon Bo Shim, Chang Hyun Lim, and Han Gyu Joo, 2013: Practical numerical reactor employing direct whole core neutron transport and subchannel thermal/hydraulic solvers. *Annals of Nuclear Energy*, **62**, 357–374.

Kochunas, Brendan, Benjamin Collins, Dan Jabaay, Thomas J Downar, and William R Martin, 2013: Overview of development and design of mpact: michigan parallel characteristics transport code. *Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering-M and C 2013*.

Kochunas, Brendan, Ang Zhu, Daniel Jabaay, Yunlin Xu, and TJ Downar, 2017: Whole-core pin-resolved pwr transient calculations in mpact. *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science & Engineering (M&C 2017)*. Volume 1.

Lewis, EE, MA Smith, N Tsoulfanidis, G Palmiotti, TA Taiwo, and RN Blomquist, 2001: Benchmark specification for deterministic 2-d/3-d mox fuel assembly transport calculations without spatial homogenization (c5g7 mox). *NEA/NSC*, **280**.

McCardell, RK, DI Herborn, and JE Houghtaling, 1969: *Reactivity Accident Test Results and Analyses for the SPERT III E-core–a Small, Oxide-fueled, Pressurized-water Reactor*. National Reactor Testing Station, US Atomic Energy Commission.

Prince, Zachary M, and Jean C Ragusa, 2019: Multiphysics reactor-core simulations using the improved quasi-static method. *Annals of Nuclear Energy*, **125**, 186–200.

Shaner, Samuel Christopher, 2014: "Transient method of characteristics via the Adiabatic, Theta, and Multigrid Amplitude Function methods". PhD thesis. Massachusetts Institute of Technology.

Taxelius, TG, 1967: *SPERT PROJECT. Quarterly Technical Report, October–December 1966*. Technical report. Phillips Petroleum Co., Idaho Falls, Idaho. Atomic Energy Div.

Tsujita, Kosuke, Tomohiro Endo, Akio Yamamoto, Yohei Kamiyama, and Kazuki Kirimura, 2013: Higher order treatment on temporal derivative of angular flux for time-dependent moc. *Proceedings of the 2013 International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering-M and C 2013*.

Yoon, Joo Il, and Han Gyu Joo, 2008: Two-level coarse mesh finite difference formulation with multigroup source expansion nodal kernels. *Journal of Nuclear Science and Technology*, **45**, 668–682.

Yuk, Seungsu, and Jin-Young Cho, 2019: *APR1400 REACTOR CORE BENCHMARK PROBLEM, BOOK*. Technical report. RPL-INERICA-004, Korea Atomic Energy Research Institute.

Zhu, Ang, Yunlin Xu, Aaron Graham, Mitchell Young, Thomas Downar, and Liangzhi Cao, 2015: Transient methods for pin-resolved whole core transport using the 2d-1d methodology in mpact. *Proc. M&C 2015*, 19–23.

Zoia, Andrea, and Emeric Brun, 2016: Reactor physics analysis of the spert iii e-core with tripoli-4®. *Annals of Nuclear Energy*, **90**, 71–82.

A SPERT III E-core Modelling

The Special Power Excursion Reactor Test (SPERT), which was conducted by Philips Petroleum Company in the early 1960s, was a series of experiments to obtain the data needed for analyzing the reactivity RIAs initiated by control rod ejection (McCardell et al. 1969). Various core configurations were used in this series of which the SPERT III E-core resembled the commercial pressurized water reactors (PWRs) in the aspect of fuel assembly structure and T/H properties. It used oxide uranium fuel and light water coolant. The facility and experimental data of the E-core were sufficiently well documented (McCardell et al. 1969; Dugone 1965), the experiments could be simulated by computer modeling later days to verify the design codes.

The SPERT E-core is loaded with 4.8 wt% UO_2 fuel rods and the cruciform transient control rod, which is used to initiate power excursion, is located at the center of the core as shown in Fig. A.1. It is surrounded by 4 fuel assemblies having 16 fuel rods. There are 48 fuel assemblies having 25 fuel rods and 8 regular control rod assemblies (CRAs). The control rods are placed at the upper section of each CRA whose lower section is fuel followers consisting of 16 fuel rods. The CRAs are marked with shading in Fig. A.1. The major geometrical and material information are provided in Table A.1.

Though there are 3 different types of fuel assemblies in the E-core each of which has its own dimension and structure(25-rod fuel assembly, 16-rod

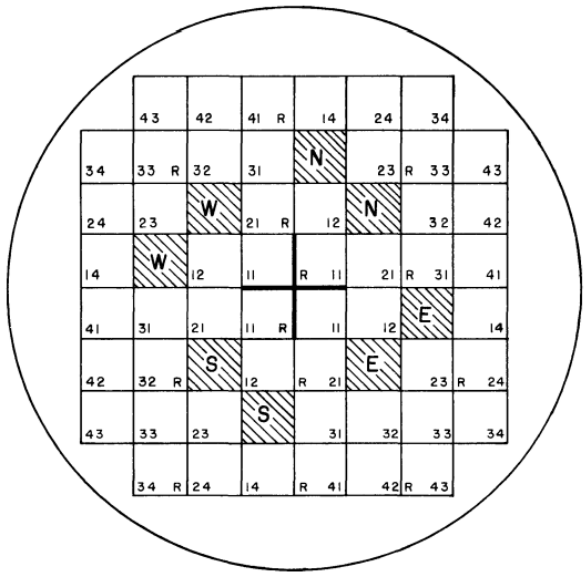


Figure A.1 Radial configuration of the SPERT III E-core (McCardell et al. 1969)

fuel assembly, and fuel follower connected with a CRA), the same fuel rod is used for all 3 types of fuel assemblies. The fuel rod is modeled with the fuel cell model shown in Figure A.2. Each fuel pellet is divided into five annular rings with eight azimuthal sectors to incorporate intra-pin flux variation. Each annular ring is assigned a different set of XSs and temperature. The moderator region is divided to 32 sub-regions to deal with the high thermal flux gradient. There is a thin boundary layer of cells added to each assembly lattice to model the complex structures at the peripheral region of a fuel assembly such as fuel cans.

The structure of an 25-rod assembly is given in Fig. A.3. 25 fuel rods are arranged in a 5×5 array with a pitch of 1.4859 cm. The fuel rods are surrounded by the fuel can which is made of Type 348 of stainless steel(SS348). Its thickness is not specified exactly. Thus, it is calculated as 0.0635 cm from the overall dimension and the flow area. Because an explicit modeling of the

Table A.1 Core characteristic data of the SPERT III E-core

		value
Fuel		4.8 wt% UO_2
Absorber		18-8 stainless steel containing 1.35 wt% B^{10}
Fuel density		10.5 g/cm^3
Fuel pellet radius		0.5334 cm
Clad inner radius		0.5410 cm
Clad outer radius		0.5918 cm
Pin pitch		1.4859 cm
Core active height		97.282 cm
25-rod fuel assembly	Dimensions	7.5565 × 7.5565 × 133.985 cm
	Flow area	27.68 cm^2
16-rod fuel assemlby	Dimensions	6.2890 × 6.2890 × 133.985 cm
	Flow area	20.39 cm^2
Fuel follower	Dimensions	6.3398 × 6.3398 × 112.673 cm
	Flow area	18.06 cm^2

slots in the fuel can which has opening slots with a total area of 774.2 cm^2 is not possible with nTRACER, the fuel can is modelled as a homogenized mixture of 75 % volume fraction of stainless steel and 25 % of water which preserves the volume ratio of slots. The radial cut of the nTRACER model for the 25-rod assembly is shown in Fig. A.4.

The transient rod is located at the center of the core and is surrounded by four 16-rod assemblies. 16-rod assemblies have the stainless can of the same material and thickness as those of the 25-rod assemblies. The transient rod is a cruciform-shaped rod which is 0.4763 cm thick and 6.5088 cm wide. The detailed shape of the transient rod assembly is shown in Fig. A.5. The upper section of the transient rod is 18-8 stainless steel. The lower section is the absorber section consisting of 1.35 wt% B^{10} stainless steel. There are guide

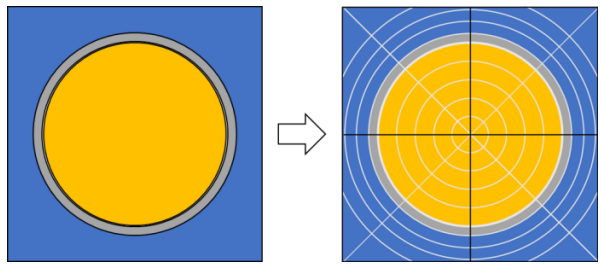


Figure A.2 Fuel cell modeling

tubes made of Zircaloy-2 which prevent the possible damages to the 16-rod assemblies by the motion of the transient rod. The existence of the guide tube is confirmed by Reference (Dugone 1965), but its exact geometry and thickness are not documented. It is assumed in the nTRACER model that the guide tube is in close contact with the fuel can, and the bushing pad attached to the transient rod. The estimated thickness of the guide tube is 0.1988 cm. Fig. A.6 shows the radial cut of the nTRACER model for the transient rod and four surrounding 16-rod assemblies. The fuel rods are modeled with a 8×8 array of cells. The fuel can, guide tube and transient rod are modelled with the cells at the periphery. Note that the transient rod is dropped to emulate control rod ejection.

There are eight CRAs in the core. The structure of a CRA is shown in Fig. A.7. It has a fuel follower at the lower section and the absorber material at the upper section. The CRAs are protected by the guide tubes made of Zircaloy-2. The fuel follower consists of 16 fuel rods which are the same as those of the other type fuel assemblies and thus the fuel can made of 304L stainless steel is at the fuel section. The thickness of this fuel can was estimated as 0.1727cm by using the flow area. Type 18-8 stainless steel plate containing 1.35 wt% B^{10}, which is a square box shape, is used as the absorber material. Its overall

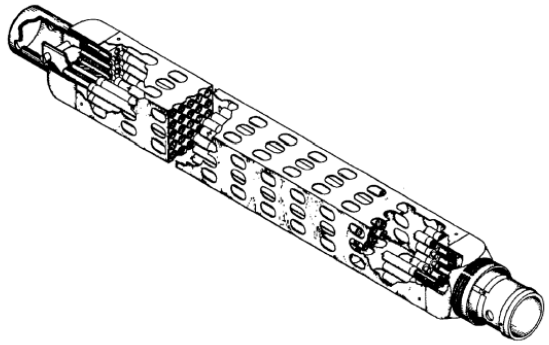


Figure A.3 25-rod fuel assembly (Dugone 1965)

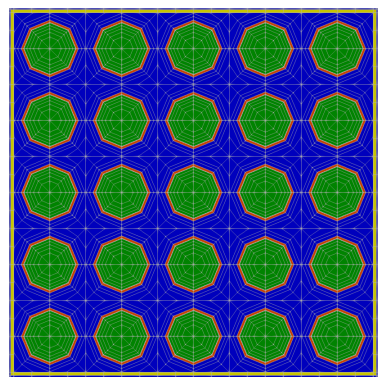


Figure A.4 25-rod fuel assembly modeling

dimension is 6.3398 cm and the thickness is 0.47244 cm. Fig. A.8 and Fig. A.9 show the nTRACER model of the fuel and absorber sections, respectively.

There is an intermediate region with a height of 11.938 cm between the bottom of the absorber section and the top of the fuel section which contains flux suppressors made of B^{10} stainless steel alloy, the same material as the absorber section. Each CRA contains 12 suppressors, six of which are 5.5474 $\times$ 2.5400 $\times$ 0.0762 cm plates and the other six are 5.9436 $\times$ 6.9056 $\times$ 0.0762 cm plates. The shape and location of the flux suppressors are shown in Figure 12. The nTRACER model of the intermediate region is shown in Figure 13. The flux suppressors are modeled as the outermost annular ring in the unit

cell that preserves the total mass of the absorber material, and it surrounds the homogenized compression spring at the center.

The number of fuel assemblies can vary to establish a desired operating condition in the E-core. Type 4F filler pieces which are the dummy spacers with the same dimensions as the 25-rod fuel assembly are placed to fill the remaining lattice positions in the core. Type 1F, 2F, and 3F filler pieces are used to fill the cylindrical geometry of the core skirt. Fig. A.12 shows the four filler piece types. The filler pieces are made of 18-8 stainless steel plates whose thickness is 0.3175 cm. The nTRACER modeling is shown in Fig. A.13. The cylindrical geometry of 1F, 2F, and 3F filler pieces are approximated with square cells. The core skirts and thermal shields are modeled as a homogeneous material. The core is discretized into 20 axial planes. The active core of 97.283 cm is modeled with 16 axial planes. The top and bottom reflectors are each modeled with two axial planes. The end plugs and compression spring in the fuel rod are modeled in the axial reflector planes. The material of the end plug is type 347 stainless steel. It is modeled in both planes of the bottom reflector and the upper plane of the top reflector. The compression spring is modeled in the lower plane of the top reflector as a homogenized material of stainless steel and void.

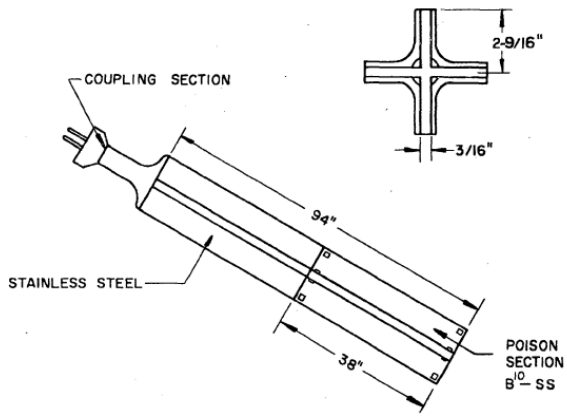


Figure A.5 Transient rod assembly (Dugone 1965)

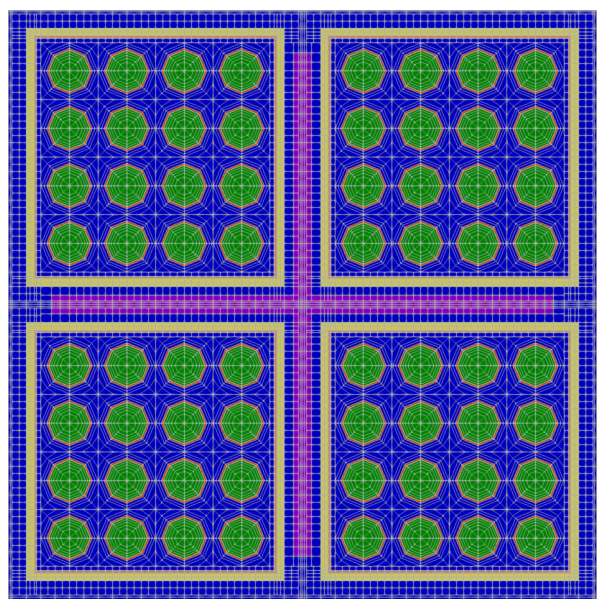


Figure A.6 Core center region modeling

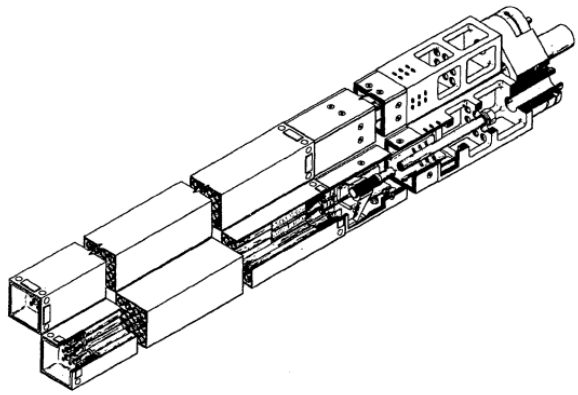


Figure A.7 Control rod assembly (Dugone 1965)

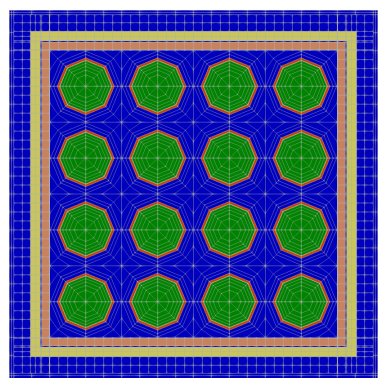


Figure A.8 CRA fuel follower modeling (green-fuel, orange-stainless, yellow-guide tube)

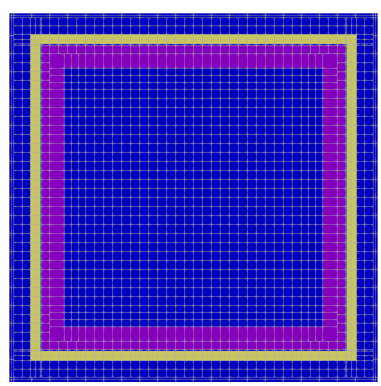


Figure A.9 CRA absorber section modeling (purple-absorber, yellow-guide tube)

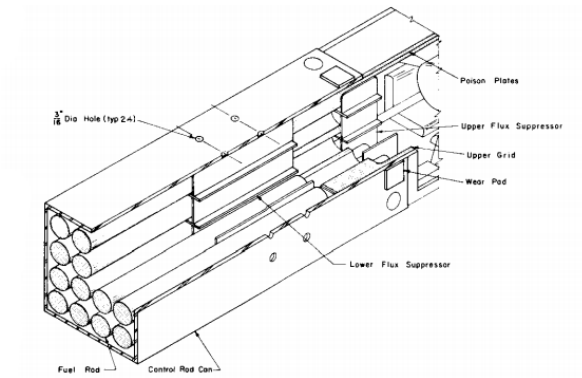


Figure A.10 Flux suppressor (Taxelius 1967)

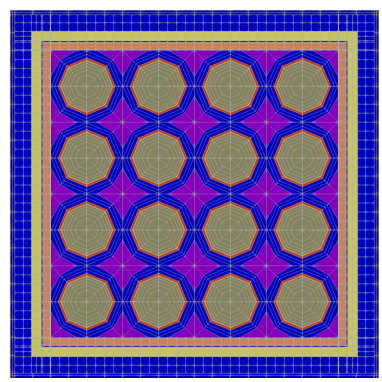


Figure A.11 CRA flux suppressor modeling (brown-spring, orange-stainless, yellow-guide tube)

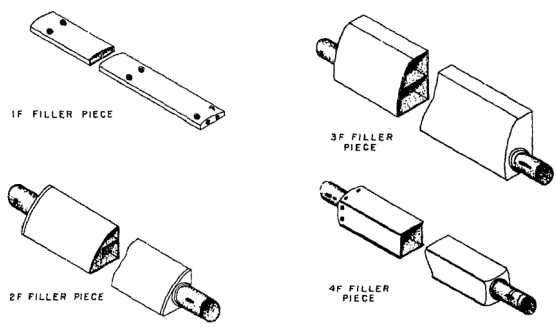


Figure A.12 Core filler pieces (McCardell et al. 1969)

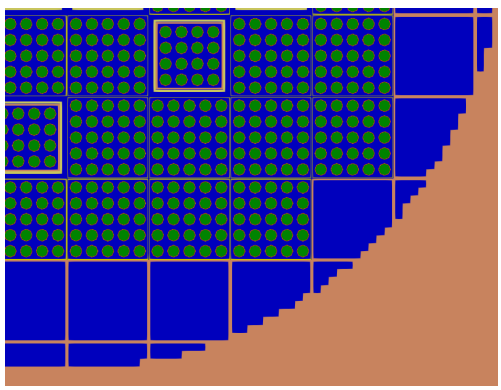


Figure A.13 Core filler modeling

초 록

원자로 규제 기준이 강화되고, 고정밀도 다물리 연계계산에 대한 수요가 증가하면서 고신뢰도 전노심 직접 과도해석이 요구되는 상황이다. 그러나 전노심 직접 계산의 막대한 계산요구량 때문에 실제적인 노심 문제 해석에 사용하는 경우 많은 계산 시간을 필요로 하거나, 수천 코어 수준의 대규모 컴퓨팅 시설에 의존해야 한다는 한계를 보였다. 본 연구는 GPU 컴퓨팅 기술 적용 및 과도해석 방법론 개선을 통해서 효율적인 과도해석능을 전노심 직접해석 코드 nTRACER에 구현하는 것을 목표로 한다.

nTRACER의 삼차원 직접 전노심 수송해석은 이차원 층별 특성곡선법 (MOC) 삼차원 소격격자 유한 차분법 (CMFD), 일차원 축방향 특성곡선법 등의 계산요소들의 연계를 통해서 이루어진다. 본 연구에서는 기존 해법에 점근사 동특성 방정식 (PKE)을 도입하여 노심 전체 중성자속의 거동을 해석하는 준정적 해법 (Quasi-static method)을 도입하였다. 이를 통해 MOC/CMFD/PKE로 이루어진 3단계 해석체계를 구현하고 각 단계마다 다른 시구간 크기를 적용하였다. 요구되는 계산량이 비교적 크고 계산을 통해서 결정하는 변수의 크기의 변화율이 작은 MOC와 CMFD 계산에는 비교적 큰 시구간을 사용하고, 계산량이 작은 PKE 계산에는 작은 시구간을 사용함으로써 계산량 대비 높은 정확도를 얻을 수 있다.

과도 상황에서 시간에 따라 각 변수의 변화율 또한 변화하기 때문에 적응형 해법을 구현하여 불필요한 MOC 및 CMFD 계산을 줄이고 정확도 대비 요구되는 계산량을 최소화하였다. MOC 계산의 경우 노심 조건이 크게 변화하는 시구간에서만 계산을 수행하는 조건적 수송계산 해법을 통해서 적응형 해법을 구현하였다.

특히 본 연구에서는 조건적 MOC 발동 기준을 기존의 단순한 일군 반응단면적 변화 기준 대신 세부 격자 잔차항 기준을 도입하여 MOC 계산의 영향을 더 정교하게 평가하였다. CMFD 계산의 시구간 크기는 적응형 시구간 조정 알고리즘을 도입하여 각 시구간에서 발생하는 에러 값이 주어진 허용치 이하로 유지되도록 조정하였다. 해당 알고리즘을 위해서 각 시구간에서 발생할 수 있는 오차 모델을 유도했고, 유도된 오차 모델을 통해 과도 계산 중 발생하는 오차를 실시간으로 추정하여 허용치 기준을 만족하는 시구간 크기를 산출한다. 5×5 핵연료 집합체 문제 해석을 통해서 적응형 해법들을 검증하였다. 새로운 MOC 발동 기준은 이전의 MOC 발동 기준보다 최대 출력 상대 오차 값을 약 80 % 감소시켰다. 적응형 시구간 조정 알고리즘 도입 결과, 검증문제 계산 시 출력이 빠르게 변화하는 구간에서 발생하는 오차가 주어진 허용치 이하로 유지되었고, 같은 수의 시구간을 사용한 고정 시구간 결과 대비 최대 출력 상대 오차가 약 80 % 감소하였다.

nTRACER 과도해석 요소 중 MOC 및 CMFD의 선형계 해법 등 연산 집약적인 요소들에 GPU 컴퓨팅이 적용되었다. GPU의 특성은 기존의 CPU와 다르기 때문에 이에 맞게 최적화가 이루어졌다. 특히 본 연구에서는 과도 CMFD 선형계 해법의 최적화가 중점적으로 수행되었다. 우선, 대규모 데이터 처리에 적합한 GPU 특성에 맞지 않는 그룹 우선 배치 방식의 선형계 해법 대신 다중 그룹 직접 해법을 적용하여 수렴 안정성과 계산 속도를 향상시켰다. 또한, CMFD 선형계 해법에서 사용되는 선조건자를 기존의 불완전 LU 분해 기반 선조건자 대신 대규모 병렬 실행이 가능한 희소 근사 역행렬 (SPAI) 선조건자로 대체하였다. SPAI 선조건자는 CMFD 행렬의 역행렬을 미리 정해진 희소행렬 구조에 따라서 근사한다. 희소행렬 구조에 따라서 SPAI 선조건자의 생성 비용과 선형계 해법 반복계산수가 결정되기 때문에 최적의 희소행렬 구조가 요구된다. 본 연구에서는 SPAI 추정 알고리즘을 도입하여 CMFD 행렬에 따른 희소행렬 구조 최적화를 수행하였다. 추정 알고리즘 도입 결과 2 %의 추정 탈락 기준치를 사용했을 때, 기존의 고정 희소행렬 구조 대비 요구되는 선형계 해법 반복계산수가 13 % 감소하였다.

새로운 직접 전노심 과도해석능의 유효성을 다양한 실제적인 노심 문제 해석을 통해서 입증하였다. SPERT III E-core 반응도 사고 실험을 해석하고 이를 실험치와 비교하여 새로운 과도해석능의 중성자 거동 해석, 동적 데이터 처리, 열궤환 모델의 정확성을 확인하였다. 서로 다른 조건의 5가지 대표 문제에 대해서 최대 출력, 노심 주기, 에너지 방출량 등 주요 실험치와 실험 불확실도 범위 이내에서 일치하였다. 같은 조건에서 수행한 이단계 해석방법을 적용 시 노심 주기에서 실험 불확실도보다 큰 오차를 보여, 직접 전노심 과도해석능을 정확도를 확인하였다. 각 계산은 20개의 상용 GPU를 장착한 클러스터에서 7시간 이내에 수행되었고, 320개의 CPU를 장착한 클러스터에서 CPU 기반 nTRACER 과도해석능을 사용한 계산보다 약 7배 빠른 속도를 보였다. 각 계산에 사용된 컴퓨팅 시설의 가격과 시간을 고려했을 때, 새로운 과도해석능은 기존 과도해석능 대비 약 13 배 높은 가격대비성능을 가진다고 볼 수 있다. APR1400에서 발생하는 가상의 제어봉 이탈 사고를 24개의 GPU를 장착한 클러스터를 사용하여 계산하였고, 125 개의 시구간을 통한 1초 동안의 과도해석을 19시간 이내에 수행하여 직접 전노심 과도해석의 실용적 활용에 대한 가능성을 확인하였다.

주요어: 전노심 직접 과도해석, 다중단계 해석법, GPU 가속, 적응형 시구간, 고신뢰도 노심 해석

학 번: 2016-21282