Nuclear Engineering and Technology 52 (2020) 2422–2430

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

journal homepage: www.elsevier.com/locate/net

Original Article

A response matrix method for the refined Analytic Function Expansion Nodal (AFEN) method in the two-dimensional hexagonal geometry and its numerical performance

Jae Man Noh

Korea Atomic Energy Research Institute, 111 Daedeok-daero 989 Beon-gil, Yuseong-gu, Daejeon, 34057, Republic of Korea

ARTICLE INFO

Article history:
Received 6 August 2019
Received in revised form
20 February 2020
Accepted 16 April 2020
Available online 22 April 2020

Keywords:
Nodal Method
Hexagonal Geometry
AFEN
Response Matrix
Dominance Ratio

ABSTRACT

In order to improve calculational efficiency of the CAPP code in the analysis of the hexagonal reactor core, we have tried to implement a refined AFEN method with transverse gradient basis functions and interface flux moments in the hexagonal geometry. The numerical scheme for the refined AFEN method adopted here is the response matrix method that uses the interface partial currents as nodal unknowns instead of the interface fluxes used in the original AFEN method. Since the response matrix method is single-node based, it has good properties such as good calculational efficiency and parallel computing affinity. Because a refined AFEN method equivalent nonlinear FDM response matrix method tried first could not provide a numerically stable solution, a direct formulation of the refined AFEN response matrix were developed. To show the numerical performance of this response matrix method against the original AFEN method, the numerical error analyses were performed for several benchmark problems including the VVER-440 LWR benchmark problem and the MHTGR-350 HTGR benchmark problem. The results showed a more than three times speedup in computing time for the LWR and HTGR benchmark problems due to good convergence and excellent calculational efficiency of the refined AFEN response matrix method.

1. INTRODUCTION

CAPP [1] is a computer code that the Korea Atomic Energy Research Institute (KAERI) has developed for the neutronic analysis of the High Temperature Gas-cooled Reactor (HTGR) core, which solves the neutron diffusion equation by the Finite Element Method (FEM) with general triangular prismatic finite elements. With high finite element order options such as quadratic or incomplete quadratic ones, CAPP has proven to be able to analyze the HTGR core accurately through its application to numerous practical design works of HTGR. However, one of its few flaws is slow calculation time, so that it takes several hours to analyze a depletion cycle of a three-dimensional HTGR core even on a multi-nodes parallel computing cluster.

A nodal method based on the Analytic Function Expansion Nodal (AFEN) method [2–6] in the hexagonal geometry has been implemented into CAPP in order to improve the computational efficiency of its high order FEM. The focus of the KAERI research and development efforts on only hexagonal prismatic HTGRs can make CAPP free from FEM adopted due to its flexibility in geometry processing. The AFEN version applied here is the refined AFEN method introduced in Refs. [5]. In this method, refinement is achieved by adding the analytical basis functions combined with the transverse-direction linear functions into the intranodal flux expansion. The flux moments that are defined by the weighted average fluxes at the interface are used as nodal unknowns corresponding to the added basis functions. Among the weighting functions proposed in the reference, the step function in the direction parallel to the interface is used as a weighting function in this paper. On the other hand, the corner point fluxes are no longer used as nodal unknowns, since it is expected that sufficient accuracy can be achieved even without them. According to Ref. [5], the refined AFEN method even without corner point fluxes provides much better accuracy than the original AFEN method with corner point fluxes [2].

Recalling that our goal is to improve the computational efficiency of the CAPP, it is of interest in this paper to find an efficient numerical scheme for the refined AFEN method. As is generally known, numerical schemes based on the response matrix method

E-mail address: jmnoh@kaeri.re.kr.

https://doi.org/10.1016/j.net.2020.04.018

can be considered more numerically efficient than the original refined AFEN method in Ref. [5]. (This general perception was also supported by the results of this paper.) The response matrix method uses the interface partial currents as nodal unknowns instead of the interface fluxes used in the original refined AFEN method. The response matrix method updates the outgoing partial currents of each node at each inner iteration step by imposing the incoming partial current boundary conditions on the interfaces of the node. Therefore, there is an advantage that the domain where the nodal unknowns and their coefficients matrixes are calculated is confined within the node independently of its neighbor nodes. In keeping with our goal of shortening the calculation time, the property that the response matrix calculation is limited to a single node is very favorable for parallel computation combined with the Red-Green-Blue (RGB) segmentation of calculational nodes described in Section 2.4. A triangular nodal response matrix formulation for hexagonal core applications can be found in Ref. [7].

Noting that the Finite Difference Method (FDM) nonlinear iteration scheme [8] is widely being used as an acceleration scheme for high-order neutron diffusion and transport methods, first of all, the nonlinear FDM response matrix method equivalent to the refined AFEN method was tried to reduce computational time. This scheme adopts two non-linear correction factors for each interface, as in Refs. [9]. However, unlike the reference, which solves a two-node problem, here we solve a single-node problem and determine the interface non-linear correction factors. This can be justified by the fact that the single-node based scheme fits more to the concept of the response matrix which can be defined as the response (i.e., outgoing partial currents) of a single node to the input (i.e., incoming partial currents). Once again, the emphasis would be to bring all of the calculations to a single-node basis, leading to many advantages in derivation of formulas, computer implementation including parallel computation, and maintenance.

Unlike the hopeful expectation caused by a big success of the nonlinear FDM as an acceleration technique, the refined AFEN method equivalent FDM response matrix method tried here could not provide a numerically stable solution [10]. Several attempts, including recommended in Refs. [10], to improve the numerical stability have been made but they were still in vain. The FDM response matrix methods would not be successful at least if they are fundamentally based on the single node AFEN solution. The numerical stability degradation is reported in Ref. [11] when applying a nonlinear FDM response matrix in a rectangular geometry in a manner quite similar to this paper.

There is a general reasoning that the stability is increased by utilizing two-node problems instead of single-node problems, because a two-node problem more quickly reflects the coupling effects between adjacent nodes than a single-node problem. However, we do not want to give up the many advantages of single-node based computing. Therefore, to assure numerical stability, we directly formulate the response matrix of the refined AFEN with interface partial currents and their moments in this paper, leaving a work developing a two-node based nonlinear FDM response matrix as a future work [10]. The numerical error analysis is provided in this paper to show the numerical performance of this method.

This paper presents the results of a two-dimensional reactor core analysis using the proposed method. The main numerical characteristics will remain almost the same even if it is expanded to three dimensions.

2. Methodology

Deriving the refined AFEN response matrix which expresses the outgoing interface partial currents into the incoming interface partial currents has two steps for convenience. In the first step, the AFEN single node is solved to obtain the relationship between the interface fluxes and the interface currents. In the second step, the response matrix is derived by replacing the interface fluxes and interface currents in this relationship with the incoming and outgoing interface partial currents.

The first step is identical to that described in Ref. [5], but it is repeated almost as it is in the reference for the sake of readability of this paper.

2.1. Refined AFEN solution in single-node [5]

2.1.1. Intranodal flux expansion

Solving the single node problem with interface current and current moment boundary conditions by the refined AFEN method starts from expending the intranodal flux distribution into the analytic basis functions with and without transverse-direction linear functions:

$$\boldsymbol{\phi}(x,y) = \boldsymbol{\varphi}(x,y) + \boldsymbol{\varphi}(u,v) + \boldsymbol{\varphi}(p,q) \tag{1}$$

where

$$\boldsymbol{\varphi}(x,y) = \sinh\left(\sqrt{\boldsymbol{\Lambda}}\,x\right)(\mathbf{A}_{xe} + \mathbf{A}_{xo}y) + \cosh\left(\sqrt{\boldsymbol{\Lambda}}\,x\right)(\mathbf{B}_{xe} + \mathbf{B}_{xo}y) \tag{2}$$

$$\boldsymbol{\Lambda} = \mathbf{D}^{-1}\boldsymbol{\Sigma} \tag{3}$$

and $\mathbf{D}$ and $\boldsymbol{\Sigma}$ are the diffusion coefficient and cross-section matrix, respectively and $\mathbf{A}$'s and $\mathbf{B}$'s are expansion coefficients. (x,y), (u,v) and (p,q) are the three coordinates in Fig. 1 introduced for convenience. And subscript e or o means an even or odd function term in the y direction, respectively.

Note that this flux expansion has twelve terms with one coefficient each and all of them completely satisfy the diffusion equation for the node. Of course, both the coefficients and the basis functions of this expansion are vectors and square matrices with the number of energy groups as its order. However, thanks to the matrix function theory, they can be treated like scalar as long as other matrix functions are not involved except the functions of $\boldsymbol{\Lambda}$ [12,13].

The average flux of the node is defined from this flux expansion as follows:

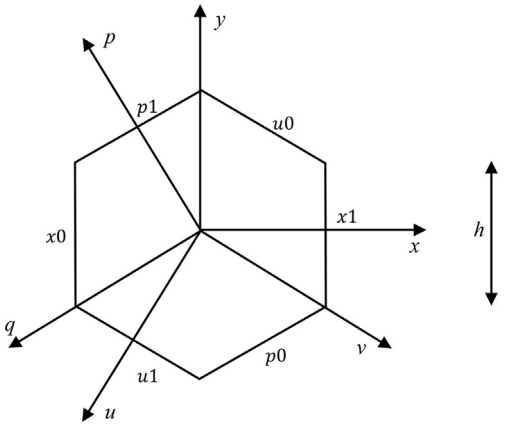

Fig. 1. Three coordinate systems and six interfaces.

$$\phi = \frac{8\sqrt{3}}{9h^2}\int_0^{\frac{\sqrt{3}}{2}h} \times \int_0^{-\frac{\sqrt{3}}{3}x+h} \frac{\phi(x,y)+\phi(-x,y)+\phi(x,-y)+\phi(-x,-y)}{4}dydx \tag{4}$$

The interface fluxes and the flux moments e.g., at the $x1$ interface are respectively defined by

$$\phi_{x1} = \frac{1}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} \phi\left(\frac{\sqrt{3}}{2}h, y\right)dy \tag{5}$$

$$\Psi_{x1} = \frac{1}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} w(y)\phi\left(\frac{\sqrt{3}}{2}h, y\right)dy \tag{6}$$

Here, $w(y)$ is the weighting function. Between the two types of weighting functions proposed in Ref. [5], i.e., the step function and the linear function, the step function is used in this paper.

$$w(y) = \begin{cases} +1 & when\ \ y \geq 0 \\ -1 & when\ \ y < 0 \end{cases} \tag{7}$$

If the step function is used, it is equivalent to the case where an interface is cut in half and the continuous condition of flux and current is applied for each half of the interface.

Strictly speaking, when applying the equivalence theory [14,15], the interface fluxes and moments in Eqs. (5) and (6) are homogeneous ones. They are multiplied by the discontinuity factors to yield the heterogeneous ones. However, for simplicity of derivation, we ignore the discontinuity factors at this moment. In implementing, of course, the discontinuity factors are involved.

Further, the interface current and the current moment at the example interface are consistently defined by

$$\mathbf{J}_{x1}^{in} = \frac{\mathbf{D}}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} \frac{\partial}{\partial x}\phi(x,y)dy\Big|_{x=\frac{\sqrt{3}}{2}h} \tag{8}$$

$$\mathbf{j}_{x1}^{in} = \frac{\mathbf{D}}{h}\int_{-\frac{h}{2}}^{\frac{h}{2}} w(y)\frac{\partial}{\partial x}\phi(x,y)dy\Big|_{x=\frac{\sqrt{3}}{2}h} \tag{9}$$

Here, superscript "in" means that the sign of the interface current is positive when it is in the direction of entry into the node.

2.1.2. AFEN solution of single-node problem

Solving the single node problem as shown in Fig. 1 to obtain the intranodal flux distribution means expressing twelve coefficients of the flux expansion Eq. (1) in terms of six interface currents and six interface current moments. This problem seems to involve inversing a 12 × 12 matrix system. However, the decoupling transformation of Reference [5] simplifies it to a problem of inversing several smaller matrixes.

This transformation transforms both the flux expansion coefficients and the nodal unknowns twice. First, the parity transformation transforms the nodal unknowns into their even and odd forms e.g., in the x-direction:

$$\boldsymbol{\omega}_{xe} = \frac{\boldsymbol{\phi}_{x1}+\boldsymbol{\phi}_{x0}}{2} - \bar{\boldsymbol{\phi}},\quad \boldsymbol{\omega}_{xo} = \frac{\boldsymbol{\phi}_{x1}-\boldsymbol{\phi}_{x0}}{2} \tag{10}$$

$$\boldsymbol{\sigma}_{xe} = \frac{\boldsymbol{\psi}_{x1}+\boldsymbol{\psi}_{x0}}{2},\quad \boldsymbol{\sigma}_{xo} = \frac{\boldsymbol{\psi}_{x1}-\boldsymbol{\psi}_{x0}}{2} \tag{11}$$

Applying symmetric thinking, the even and the odd forms of the interface currents and current moments are defined as follows:

$$\boldsymbol{\eta}_{xe}^{in} = \frac{\mathbf{J}_{x1}^{in}+\mathbf{J}_{x0}^{in}}{2},\quad \boldsymbol{\eta}_{xo}^{in} = \frac{\mathbf{J}_{x1}^{in}-\mathbf{J}_{x0}^{in}}{2} \tag{12}$$

$$\boldsymbol{\varsigma}_{xe}^{in} = \frac{\mathbf{j}_{x1}^{in}+\mathbf{j}_{x0}^{in}}{2},\quad \boldsymbol{\varsigma}_{xo}^{in} = \frac{\mathbf{j}_{x1}^{in}-\mathbf{j}_{x0}^{in}}{2} \tag{13}$$

Now, the direction transformation transforms the coefficients of the expansion flux and the nodal unknowns further as shown below.

$$\mathbf{C}_{\theta s} = \frac{\mathbf{C}_{xs}+\mathbf{C}_{us}+\mathbf{C}_{ps}}{3},\quad \mathbf{C}_{\varepsilon s} = \frac{2\mathbf{C}_{xs}-\mathbf{C}_{us}-\mathbf{C}_{ps}}{3},\quad \mathbf{C}_{\chi s} = \frac{\mathbf{C}_{us}-\mathbf{C}_{ps}}{3} \tag{14}$$

$$\boldsymbol{\omega}_{\theta s} = \frac{\boldsymbol{\omega}_{xs}+\boldsymbol{\omega}_{us}+\boldsymbol{\omega}_{ps}}{3},\quad \boldsymbol{\omega}_{\varepsilon s} = \frac{2\boldsymbol{\omega}_{xs}-\boldsymbol{\omega}_{us}-\boldsymbol{\omega}_{ps}}{3},\quad \boldsymbol{\omega}_{\chi s} = \frac{\boldsymbol{\omega}_{us}-\boldsymbol{\omega}_{ps}}{3} \tag{15}$$

$$\boldsymbol{\sigma}_{\theta s} = \frac{\boldsymbol{\sigma}_{xs}+\boldsymbol{\sigma}_{us}+\boldsymbol{\sigma}_{ps}}{3},\quad \boldsymbol{\sigma}_{\varepsilon s} = \frac{2\boldsymbol{\sigma}_{xs}-\boldsymbol{\sigma}_{us}-\boldsymbol{\sigma}_{ps}}{3},\quad \boldsymbol{\sigma}_{\chi s} = \frac{\boldsymbol{\sigma}_{us}-\boldsymbol{\sigma}_{ps}}{3} \tag{16}$$

where coefficient letter $\mathbf{C}$ is $\mathbf{A}$ or $\mathbf{B}$ and parity index s is e or o. Of course, the same transformation is applied to the currents and current moments.

On the way to expressing the transformed unknowns in terms of the transformed expansion coefficients, we can realize that the original 12 × 12 matrix equation is decoupled with four 2 × 2 matrix equations and four scalar equations. In particular, two of the four 2 × 2 matrixes are in the relationship of similarity transformation to the other two. Therefore, solving a single-node problem simply involves finding the inverses of two 2 × 2 matrixes and four scalars. For example, one of two 2 × 2 matrix systems is given by

$$\begin{bmatrix} \boldsymbol{\omega}_{\chi o} \\ \boldsymbol{\sigma}_{\varepsilon e} \end{bmatrix} = \mathbf{T}\begin{bmatrix} \mathbf{A}_{\chi e} \\ \mathbf{B}_{\varepsilon o} \end{bmatrix} \text{ and } \begin{bmatrix} \boldsymbol{\eta}_{\chi o}^{in} \\ \boldsymbol{\varsigma}_{\varepsilon e}^{in} \end{bmatrix} = \mathbf{DM}\begin{bmatrix} \mathbf{A}_{\chi e} \\ \mathbf{B}_{\varepsilon o} \end{bmatrix} \tag{17}$$

where $\mathbf{M}$ and $\mathbf{T}$ are 2 × 2 matrixes with elements of matrix functions.

By eliminating the coefficient vector, the interface fluxes and flux moments can be expressed in terms of the interface currents and current moments e.g., as follows:

$$\begin{bmatrix} \boldsymbol{\omega}_{\chi o} \\ \boldsymbol{\sigma}_{\varepsilon e} \end{bmatrix} = \mathbf{TM}^{-1}\mathbf{D}^{-1}\begin{bmatrix} \boldsymbol{\eta}_{\chi o}^{in} \\ \boldsymbol{\varsigma}_{\varepsilon e}^{in} \end{bmatrix} \tag{18}$$

where $\mathbf{TM}^{-1}$ is also a matrix function system of $\boldsymbol{\Lambda}$. It is noted that any functions of a given matrix or their operations can be evaluated relatively easily, e.g., by diagonalizing the matrix using its eigensystem. This good property has been well utilized so far. However, it will be soon realized that such good luck is no longer with the

final response matrix equation.

As mentioned in Refs. [5], the inverse of **M** in Eq. (18) is singular when one of the eigenvalues of the cross-section matrix is very small. This singularity is removed in the manner described in the reference.

2.2. Refined AFEN response matrix

The response matrix that computes the output, which are the outgoing interface partial currents going out of a node, from the input, which are the incoming interface partial currents coming into the node, is derived by noting that the interface partial currents at the interface *s* in the direction *d* are expressed in terms of the interface fluxes of Eqs. (5) and (6) and the interface currents of Eqs. (8) and (9).

$$\mathbf{P}_{ds}^{f} = \frac{\mathbf{J}_{ds}^{f}}{2} + \frac{\Phi_{ds}}{4} \tag{19}$$

$$\mathbf{p}_{ds}^{f} = \frac{\mathbf{j}_{ds}^{f}}{2} + \frac{\Psi_{ds}}{4} \tag{20}$$

where flow direction index *f* is *in* or *out*, direction index *d* is *x*, *u*, or *p*, and interface index *s* is 0 or 1. Then, the interface fluxes and currents are equivalently given by

$$\mathbf{J}_{ds}^{in} = \mathbf{P}_{ds}^{in} - \mathbf{P}_{ds}^{out}, \quad \Phi_{ds} = 2\left(\mathbf{P}_{ds}^{in} + \mathbf{P}_{ds}^{out}\right) \tag{21}$$

$$\mathbf{j}_{ds}^{in} = \mathbf{p}_{ds}^{in} - \mathbf{p}_{ds}^{out}, \quad \Psi_{ds} = 2\left(\mathbf{p}_{ds}^{in} + \mathbf{p}_{ds}^{out}\right) \tag{22}$$

Since the relationships (19) and (20) are linear and the parity and direction transformations explained in the previous section are also linear, the partial currents and moments shall have their transformed forms with respect to the both transformations and these forms shall have the relationships corresponding to those of Eqs. (21) and (22), as shown e.g., for the transformed unknowns in Eq. (17).

$$\boldsymbol{\eta}_{\chi o}^{in} = \mathbf{Z}_{\chi o}^{in} - \mathbf{Z}_{\chi o}^{out}, \quad \boldsymbol{\omega}_{\chi o} = 2\left(\mathbf{Z}_{\chi o}^{in} + \mathbf{Z}_{\chi o}^{out}\right) \tag{23}$$

$$\boldsymbol{\varsigma}_{\varepsilon e}^{in} = \mathbf{z}_{\varepsilon e}^{in} - \mathbf{z}_{\varepsilon e}^{out}, \quad \boldsymbol{\sigma}_{\varepsilon e} = 2\left(\mathbf{z}_{\varepsilon e}^{in} + \mathbf{z}_{\varepsilon e}^{out}\right) \tag{24}$$

Substituting these relationships into Eq. (18) and solving for the transformed outgoing partial currents, we finally obtain the response matrix in the transformed system as follows,

$$\begin{bmatrix} \mathbf{Z}_{\chi o}^{out} \\ \mathbf{z}_{\varepsilon e}^{out} \end{bmatrix} = \mathbf{R} \begin{bmatrix} \mathbf{Z}_{\chi o}^{in} \\ \mathbf{z}_{\varepsilon e}^{in} \end{bmatrix} \tag{25}$$

where $\mathbf{R} = -(2\mathbf{I} + \mathbf{T}\mathbf{M}^{-1}\mathbf{D}^{-1})^{-1}(2\mathbf{I} - \mathbf{T}\mathbf{M}^{-1}\mathbf{D}^{-1})$, which is a response matrix we finally want to get. Unfortunately, the matrix, $2\mathbf{I} + \mathbf{T}\mathbf{M}^{-1}\mathbf{D}^{-1}$ is not a matrix function of $\boldsymbol{\Lambda}$ anymore because it contains the diffusion coefficient matrix **D**. As previously concerned, the matrix function attribute is no longer valid. Therefore, the inversion of this matrix becomes a full inversion of a 2G × 2G matrix.

Note that the interface partial currents and moments can be easily transformed into their linearly transformed partners and vice versa. Once the interface incoming partial currents are given for a node, the interface outgoing partial currents and moments can be calculated by the response matrix, e.g., Eq. (25). Then, these outgoing partial currents become the partial currents incoming into its neighboring nodes. This provides an iterative process to solve the global core eigenvalue problem through the well-known inner-outer iteration. Generally, the number of inner iterations per outer iteration is an issue in this type of iteration. As in many other nodal methods, we used one for this value throughout this paper. This method with one inner iteration per outer iteration, called the Equipoise method, "is more economical than the standard method without any sophisticated acceleration technique for certain problems" (p.133 in Ref. [16]), for example, benchmark problems in this paper.

The response matrix equation, e.g., Eq. (25) is an efficient equation. It calculates the six outgoing interface partial currents of each node at once at each inner iteration step by imposing the boundary conditions of six incoming partial currents. Considering that one interface is shared by two nodes, it can be noted that three interface unknowns are equivalently calculated per each single node calculation. On the other hand, the original refined AFEN method [5] determines only one interface flux by solving a two-node problem with current continuity condition across the interface between the two nodes. Therefore, assuming the same number of total inner iterations is required to achieve the same accuracy, the response matrix method becomes three times more efficient than the original refined AFEN method theoretically.

2.3. Numerical performance analysis

The numerical performance of the refined AFEN response matrix in the hexagonal geometry is shown by a numerical error analysis in a general textbook on numerical methods [16].

The iterative method applied to solve the AFEN response matrix in the hexagonal geometry by the power method is given by

$$\mathbf{s}^{(t+1)} = \frac{1}{k^{(t)}} \boldsymbol{A}\, \mathbf{s}^{(t)} \tag{26}$$

$$k^{(t+1)} = \frac{\| \mathbf{A}\, \mathbf{s}^{(t)} \|}{\| \mathbf{s}^{(t)} \|} = k^{(t)} \frac{\| \mathbf{s}^{(t+1)} \|}{\| \mathbf{s}^{(t)} \|} \tag{27}$$

where $\mathbf{s}^{(0)} = \mathbf{s}_0$, $k^{(0)} = k_0$, and $t = 0, 1, \ldots$ **s** is an 1 × n iteration vector and **A** is the n × n corresponding iteration matrix. The iterative matrix depends not only on the problem itself, including boundary conditions, but also on the discretization method (e.g., original AFEN vs. AFEN response matrix) or the computational sweeping order. The nodal neutron source vector can be the iteration vector in this discussion and the sum of absolute values of the elements of the source vector can serve as the norm for the iteration vector in Eq. (27). Assume that **A** is a complete matrix and that it has a single dominant eigenvalue and a second dominant eigenvalue. Let $\lambda_1, \ldots, \lambda_n$ ($|\lambda_1| > |\lambda_2| > |\lambda_j|$ for all j) denote the eigenvalues of **A** and $\mathbf{u}_1, \ldots, \mathbf{u}_n$ the corresponding eigenvectors, which form a complete basis set. Writing the initial vector $\mathbf{s}_0$ in terms of these basis vectors

$$\mathbf{s}_0 = c_1\mathbf{u}_1 + \cdots + c_n\mathbf{u}_n \tag{28}$$

the solution of the iteration system Eq. (26) takes the explicit form

$$\mathbf{s}^{(t)} = T^{(t)}\left(c_1\mathbf{u}_1 + c_2\left(\frac{\lambda_2}{\lambda_1}\right)^t \mathbf{u}_2 \cdots + c_n\left(\frac{\lambda_n}{\lambda_1}\right)^t \mathbf{u}_n\right) \tag{29}$$

where

$$T^{(t)} = \frac{\lambda_1^t}{\prod_{i=0}^{t-1} k^{(i)}} = \prod_{i=0}^{t-1} \frac{\lambda_1}{k^{(i)}} \quad with \quad k^{(0)} = k_0 \tag{30}$$

It is certain that the iteration system Eq. (26) will converge to the multiple of the dominant eigenvector of the iteration matrix **A** as the iteration progresses because all the terms after the second term of Eq. (29) will gradually vanish.

$$\mathbf{s}^{(t)} \approx T^{(t)} c_1 \mathbf{u}_1 \equiv \mathbf{s}_\infty \quad for \ \ all \ \ \ t \gg 0 \tag{31}$$

$k^{(t)}$ will also converge to the dominant eigenvalue λ_1.

$$k^{(t)} = k^{(t-1)} \frac{\| \mathbf{s}^{(t)} \|}{\| \mathbf{s}^{(t-1)} \|} \approx k^{(t-1)} \frac{\left| T^{(t)} \right|}{\left| T^{(t-1)} \right|} = \lambda_1 \equiv k_\infty \quad for \ \ all \ \ t \gg 0 \tag{32}$$

In fact, $T^{(t)}$ must converge in order for the solution Eq. (26) to converge. This is proven by Eqs. (30) and (32) because there exists a finite number that confines $T^{(t)}$ for all $t \gg 0$.

Equation (29) implies that the rate of convergence of the iteration system is governed by the ratio $|\lambda_2 / \lambda_1|$. This ratio is called the convergence rate rather than the dominance ratio here to avoid confusion with its other possible definitions elsewhere. We define the error vector in order to derive an expression for the convergence rate.

$$\mathbf{e}^{(t)} = \mathbf{s}^{(t)} - \mathbf{s}_\infty = T^{(t)} \left(c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^t \mathbf{u}_2 \cdots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^t \mathbf{u}_n \right) \tag{33}$$

The first term, as is the argument above, will eventually dominate the whole error vector.

$$\mathbf{e}^{(t)} \approx T^{(t)} c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^t \mathbf{u}_2 \quad for \ \ all \ \ t \gg 0 \tag{34}$$

Finally, an iterative estimator of the convergence rate that serves as an indicator of the numerical performance of the iteration system can be obtained:

$$\frac{\| \mathbf{e}^{(t)} \|}{\| \mathbf{e}^{(t-1)} \|} \approx \frac{\lambda_2}{\lambda_1} \quad for \ \ all \ \ t \gg 0 \tag{35}$$

In order to estimate the convergence rate numerically, $\mathbf{s}_\infty$ is approximated by iteration until $\mathbf{s}^{(t)}$ converges within almost the truncation error level. Then, the iteration is repeated from the beginning to compute the convergence rate using Eq. (35).

Note that the inner iteration determines the neutron flux distribution for a fixed fission source distribution and that the two refined AFEN methods are mathematically equivalent. Therefore, an outer iteration for the two methods will result in the equivalent flux distribution if the infinite number of inner iterations is performed. Of course, we assumed here that the boundary condition and the initial guess for nodal unknowns are equivalent. As a result, the convergence rate or the number of outer iterations to achieve a certain level of accuracy becomes the same. Therefore, the performance of both methods depends on the performance of the inner iteration, that is, the spectral radius of their inner iteration matrix.

If a certain insufficient number of inner iterations per outer iteration is performed (e.g., once in this paper), the outer iteration matrix for the two methods is different and so is their convergence rate or the number of outer iterations. In this case, the lower the spectral radius of the inner iteration matrix, the lower the convergence rate and the less the number of outer iterations. Therefore, the convergence rate calculated by Eq. (35) becomes not only an indicator of the spectral radius of the inner iteration matrix but also an indicator of the overall numerical performance.

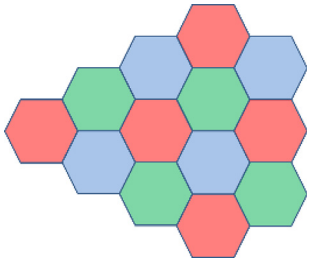

Fig. 2. RGB sweeping scheme.

2.4. RGB Sweeping Scheme

As described in INTRODUCTION, the response matrix calculations are performed only within a single node regardless of neighboring nodes. Therefore, these calculations are carried out by sequentially moving from one node to another. In this case, it is advantageous to sweep the nodes by dividing them into Red (R), Green (G), and Blue (B) nodes as shown in Fig. 2, like checkerboard sweeping for a rectangular node core.

This kind of iteration schemes is good in convergence and stability due to geometrical balance. It further enhances the advantage in parallel-computing that the response matrix method has already.

In addition, the memory required is saved by storing inputs of the response matrix calculation, i.e., incoming partial currents and outputs, i.e., outgoing partial currents in the same storage. This is because the outgoing partial currents resulting from previous two other color types of node calculations automatically become incoming partial currents for the third kind of node calculations.

3. Numerical results and discussion

The numerical performance of the refined AFEN response matrix method in the hexagonal geometry was verified against several benchmark problems including small and large light water reactor (LWR) and high temperature gas-cooled reactor (HTGR) cores. Note that both LWRs and HTGRs are thermal reactors in which nodal methods can be widely used due to their advantages in dealing the steep local gradient of the thermal flux across the material discontinuity.

Since the initial objective of implementing the refined AFEN method to CAPP is to reduce the computation time, the verification here focuses primarily on showing computational efficiency rather than accuracy. The excellent accuracy of the refined AFEN method in the reactor core analysis has already shown in Refs. [5]. The numerical error analysis is performed to quantify the computational efficiency.

3.1. Mini core problem

A mini core problem in Fig. 3 against which the refined AFEN response matrix method developed in this study was first verified was derived from VVER-440 benchmark problem [5,17]. This tiny core has seven fuel assemblies in the first and second rings of the hexagonal core, which are surrounded by twelve non-power generating control assemblies in the third ring.

The numerical error analysis was performed with two energy groups in a sixth core to show the numerical performance of the refined AFEN response matrix method. The two-group assembly homogenized cross-sections used in this analysis are directly quoted from Reference [17].

The results of the refined AFEN response matrix method were

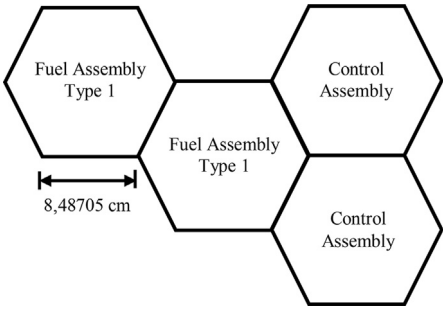

Fig. 3. Mini core problem (1/6 core).

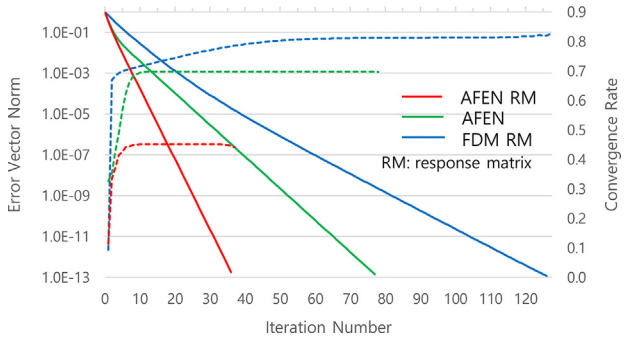

Fig. 4. Convergence pattern (mini core).

compared with those of the original refined AFEN method and the FDM response matrix method without nonlinear correction factors. Fig. 4 shows a pattern in which the node-wise neutron fission source vector converges as the iteration progresses. Each of the three solid lines in this figure is the logarithmic scale normalized norm of the error vector of the source vector given in Eq. (33) (See the left-y axis.) and each of dotted lines is the convergence rate estimated by equation (35) (See the right-y axis). Red, green and blue colors on both solid and dotted lines indicate the quantity for the refined AFEN response matrix method, the original refined AFEN method and the FDM response matrix method, respectively. This figure clearly illustrates the excellence of the refined AFEN response matrix method in numerical performance beyond comparison with the other two methods. This method with a much lower convergence rate converges more than twice as fast as the other two methods. This method reaches an asymptotic convergence state after only a few initial iterations, which is characterized by a flat convergence rate and a linearly decreasing error in logarithmic scale. In this state, the error is reduced by the power of the convergence rate as each iteration progresses. Therefore, an acceleration by extrapolation of the iterative vector becomes possible.

Some numerical performance related parameters calculated by the three methods are compared in Table 1. The k-effective difference between the two refined AFEN methods is purely due to the different boundary conditions because the two methods are mathematically equivalent. Therefore, this difference does not mean that one is better in accuracy than the other but it is slightly larger because a small node size makes this problem sensitive to the boundary condition. The boundary condition applied is the zero incoming partial current for the two response matrix methods and the zero flux for the conventional form of the refined AFEN method. This simply because the boundary condition for each nodal method corresponds to the type of its nodal unknowns.

The effect of boundary conditions on numerical performance of each nodal method can be presumed as a change in the iterative matrix caused by the boundary condition that does not correspond to the nodal method. An example taken here is the iterative matrix difference between the conventional AFEN method with its corresponding zero flux boundary condition and that with its non-corresponding zero incoming partial current boundary condition. For the former, it is certain that the row of the iterative matrix corresponding to an interface flux on the boundary, which is one of the unknowns of this nodal method, is diagonal. However, the row of iterative matrix for the latter will represent an expression that expresses the incoming partial current at the boundary interface, which is not an unknown, into terms of the interface fluxes around the boundary, which are unknowns. This row is certainly non-diagonal. This will hurt the diagonal dominance of the iterative matrix ($|a_{ii}| \geq \sum_{j \neq i} |a_{ij}|, a_{ij} \in \mathbf{A}$ in Eq. (26)) and degrade the numerical performance of the AFEN method with the zero-incoming partial current boundary condition. However, due to the small number of boundary interfaces and the small physical differences between zero flux and zero incoming partial current, the effect is significantly limited. This discussion symmetrically applies to the response matrix method case, where the zero-flux condition shows slightly poorer performance than the zero incoming partial current condition. For the three benchmark problems in this paper, the refined AFEN method with the zero-incoming partial current boundary condition increases the number of iterations needed to achieve the 10^{-7} accuracy from 41, 1454, and 545 to 44, 1483, and 549, respectively. Of course, its converged fundamental mode solutions were found to be equivalent to those of the refined AFEN response matrix method with the same boundary condition.

The second column of the table is the convergence rate estimated numerically, where the value for the refined AFEN response method is the smallest. The third column is the number of iterations expected to achieve a less than 10^{-7} accuracy in node-wise sources if the source error decreases in an asymptotic manner as described above. This is the value calculated by the following equation:

$$-7/\log_{10}(\lambda_2/\lambda_1) \tag{36}$$

The next column is the number of iterations performed to achieve the same accuracy in the actual calculation. Not only does the refined AFEN response matrix method have the smallest number of iterations, but it also has the smallest deviation between the prediction and the actual value. This means it reaches the asymptotic state very early, which is advantageous for acceleration by asymptotic extrapolation. The last column is time consumed for the calculation. This value is obtained by averaging three measurements from a PC with Intel® Core™ i7-4930 K CPU using the functions of the MS Visual Studio™ Chrono library. The refined AFEN response matrix method is 2.5 times faster than the original refined AFEN method. It is a little faster considering the number of

Table 1
Numerical performance parameters (mini core).

	k-eff	Convergence Rate	Expected Iterations	Actual Iterations	Computing Time (μsec)
AFEN RM	0.778735	0.45164	20	20	389
AFEN	0.776642	0.69846	45	41	944
FDM RM	0.865688	0.81307	78	51	293

iterations, but slower considering the efficiency of the response matrix aforementioned in the last paragraph of Section 2.2. This is probably due to the fact that the size of the core and the number of energy groups are so small that the proportion of auxiliary operations other than the iteration matrix related operations becomes not small even within a single iteration.

3.2. VVER-440 problem

The refined AFEN response matrix method were further verified against the VVER440 benchmark problem [5,17], which is a commercial size LWR core simulating an old Soviets PWR. It consists of 342 fuel assemblies, 7 non-power generating control rod assemblies, and 72 surrounding reflector assemblies. Again, the assembly homogenized cross-sections directly come from Reference [17].

The results of the numerical error analysis are shown in Fig. 5 and Table 2. All the components of the figure and the table have a completely same meaning as described in the previous section. Note that there exists the same boundary condition difference among the three methods as described in the section.

The error vector of the refined AFEN response matrix method is smaller than that of the FDM response matrix method at the early stage of iteration. But eventually it gets caught up with the FDM response matrix method near the 1,000th iteration (near 10^{-11} error). This is because, the refined AFEN response matrix method has a very comparable but slightly larger convergence rate compared with the FDM response matrix method. Nevertheless, the FDM response matrix method is not judged to have a small enough convergence rate to be able to accelerate the refined AFEN response matrix method.

The deviation between the predicted and the actual number of iterations widens as the problem size increases. Therefore, attention is required to accelerate calculation with asymptotic extrapolation. An adequate under relaxation factor in extrapolation may be desirable.

Comparing the refined AFEN response matrix method and the original refined AFEN method is more exciting. Although the refined AFEN response matrix method has a much smaller convergence rate (The scale is how far from one.), it seems to be inferior to the original refined AFEN method in terms of the error size at the early stage of iteration. However, eventually, it requires three times shorter computing time to achieve 10^{-7} accuracy. Considering that it does not seem to be overwhelmingly faster (3.4 times faster) than the number of iterations has reduced (3.2 times), the proportion of auxiliary operations is still too significant to show the advantage of the calculational efficiency of the response matrix.

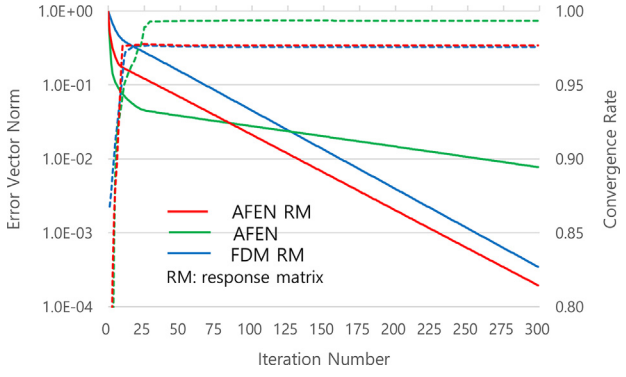

Fig. 5. Convergence pattern (VVER-440).

3.3. MHTGR-350 problem

The MHTGR-350 problem [18] is a 350MWth hexagonal prismatic block type HTGR core with graphite moderator and helium coolant. As shown in Fig. 6, it has an active core of 66 fuel blocks in the fourth, fifth and sixth rings of the core, surrounded by graphite reflectors with about three rings thick inward and outward. Due to spectrum shift in the graphite-moderated reactor, the ten-energy group system rather than the two-group system is used for the analysis of the MHTGR-350 core. The ten-group cross-sections directly cited from reference [18] are listed in Tables 3 and 4 for the fuel blocks and the reflector block, respectively.

Fig. 7 and Table 5 illustrate the results of the numerical error analysis. See also Section 3.1 for the meaning of the figure and table.

The k-effective value calculated for this problem by CAPP with the cubic finite element option and the zero incoming current boundary condition is 1.09346, which was used as a reference value to verify the other finite element options [18]. This boundary condition is consistent with the refined AFEN response matrix method. Note that even the small difference of about 20 pcm between these two consistent values does not mean which is more accurate.

The thickness of the reflector is increased compared with the previous two problems even though the relative dimension reduction in the graphite-moderated reactor is taken into account. Therefore, the influence of the boundary condition is reduced, and the k-effective value deviation between the two AFEN methods is greatly reduced.

The results of the error analysis show that the convergence patterns for the VVER-440 problem, such as the order of the three methods in convergence speed driven by the convergence rate size, remain the same for this problem. The error of the original refined AFEN method is the smallest in the early iteration stage but eventually caught by the refined AFEN response matrix method. However, the big difference from the results of the error analysis of VVER-440 is also shown: The convergence rate deviation between the two AFEN methods has greatly narrowed, which in turn leads to a big reduction in the difference in the number of iterations to achieve the 10^{-7} accuracy (from 3.2 times for VVER-440 to 1.2 times for MHTGR-350).

This should have meant less reduction in computing time but, there is another reversal, so the reduction is almost the same as for the VVER-440 problem (3.4 times faster). This is because the proportion of the iterative matrix related operations increases significantly compared to that of the auxiliary operations as the number of neutron energy groups increases from two to ten. The calculational efficiency of the refined AFEN response matrix method, which is at least three times higher per iteration, becomes easier to be realized for this problem.

Table 2
Numerical performance parameters (VVER-440).

	k-eff	Convergence Rate	Expected Iterations	Actual Iterations	Computing Time (msec)
AFEN RM	1.009645	0.97672	684	460	64
AFEN	1.008632	0.99358	2501	1454	216
FDM RM	1.018224	0.97583	659	463	31

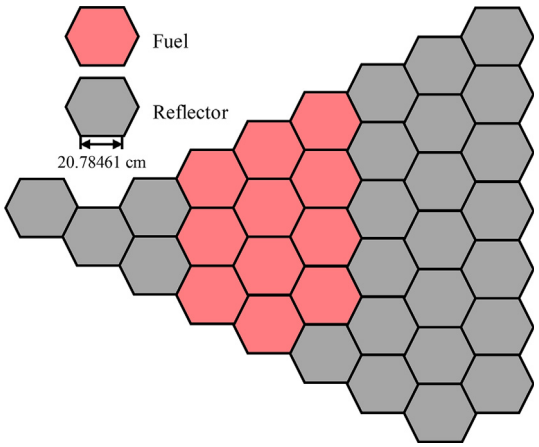

Fig. 6. MHTGR-350 problem (1/6 core).

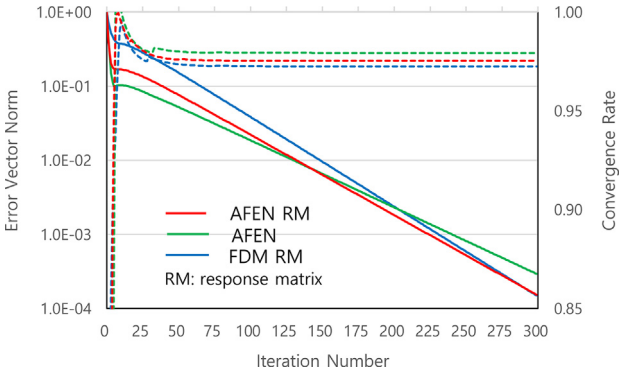

Fig. 7. Convergence pattern (MHTGR-350).

The results showed again that the convergence rate of FDM is not large enough to accelerate the AFEN response matrix method and that a careful asymptotic acceleration is required with under-relaxation factors.

4. Conclusion

In order to improve efficiency of the CAPP code in the analysis of the hexagonal reactor cores, we have tried to implement a refined AFEN method in the hexagonal geometry whose accuracy has been well proven. In the nodal method, the corner point fluxes are no longer nodal unknowns, but the flux moments that are defined by the weighted average fluxes at the interface take the role of nodal unknowns.

In addition, the numerical method for the refined AFEN method adopted here is the response matrix method that uses the interface partial currents as nodal unknowns instead of the interface fluxes used in the original refined AFEN method. This method has an advantage that all the iterative matrix calculations are single-node based (independent from neighboring nodes), which is not only very efficient but also very favorable for parallel computation.

Among response matrix methods, first tried is a two-factor nonlinear FDM response matrix method that determines the nonlinear correction factors by solving the single node problem by the refined AFEN method. Unfortunately, the nonlinear FDM response matrix method equivalent to the refined AFEN method tried here could not provide a numerically stable solution. Therefore, to assure numerical stability while maintaining the good attributes of the response matrix method, we developed the direct formulation of the refined AFEN response matrix with interface partial currents and their moments in this paper.

To show the numerical performance of the refined AFEN response matrix method against the original AFEN method, the numerical error analyses were performed for several benchmark problems including the two-group VVER-440 benchmark problem representing the LWR core and the ten-group MHTGR-350

Table 3
Cross-sections of fuel block (MHTGR-350).

g	D	Σ_a	$\nu\Sigma_f$	$\Sigma_{1\to g}$	$\Sigma_{2\to g}$	$\Sigma_{3\to g}$	$\Sigma_{4\to g}$	$\Sigma_{5\to g}$	$\Sigma_{6\to g}$	$\Sigma_{7\to g}$	$\Sigma_{8\to g}$	$\Sigma_{9\to g}$	$\Sigma_{10\to g}$
1	1.391	4.217E-4	3.969E-4										
2	9.999E-1	5.398E-3	3.369E-3	4.594E-3									
3	1.016	6.679E-3	2.411E-3		1.273E-2		6.264E-3	5.949E-6	1.159E-5	7.806E-6	5.964E-7	2.271E-7	1.693E-7
4	1.003	5.149E-3	6.505E-3			2.704E-2		4.092E-2	7.372E-3	1.274E-3	1.564E-4	5.056E-5	3.535E-5
5	9.852E-1	9.273E-3	1.382E-2			1.752E-4	4.908E-2		5.866E-2	1.213E-2	1.809E-3	5.994E-4	4.573E-4
6	9.748E-1	1.122E-2	1.683E-2			1.906E-5	9.655E-3	5.716E-2		2.907E-2	5.261E-3	2.024E-3	1.309E-3
7	9.612E-1	1.140E-2	1.723E-2			7.494E-6	5.393E-3	3.943E-2	9.280E-2		5.587E-2	2.005E-2	1.837E-2
8	8.987E-1	1.583E-2	2.603E-2			4.066E-7	4.503E-4	4.094E-3	1.199E-2	3.683E-2		5.400E-2	3.058E-2
9	8.123E-1	2.378E-2	4.159E-2			2.970E-8	4.948E-5	4.899E-4	1.629E-3	4.815E-3	1.848E-2		3.440E-2
10	7.841E-1	3.955E-2	7.191E-2				8.949E-6	9.867E-5	2.812E-4	1.174E-3	2.828E-3	8.921E-3	

The fission yield is 1.0 for g = 1 and 0.0 for the other groups.

Table 4
Cross-sections of reflector block (MHTGR-350).

g	D	Σ_a	$\Sigma_{1\to g}$	$\Sigma_{2\to g}$	$\Sigma_{3\to g}$	$\Sigma_{4\to g}$	$\Sigma_{5\to g}$	$\Sigma_{6\to g}$	$\Sigma_{7\to g}$	$\Sigma_{8\to g}$	$\Sigma_{9\to g}$	$\Sigma_{10\to g}$
1	1.162	5.758E-6										
2	9.097E-1	8.073E-6	6.922E-3									
3	9.148E-1	2.931E-5		1.759E-2		4.319E-3	6.713E-6	1.355E-5	8.718E-6	6.723E-7	2.584E-7	1.944E-7
4	9.007E-1	5.942E-5			3.252E-2		4.366E-2	8.290E-3	1.387E-3	1.759E-4	5.748E-5	4.058E-5
5	8.963E-1	7.607E-5			2.142E-4	7.140E-2		6.528E-2	1.325E-2	2.030E-3	6.819E-4	5.254E-4
6	8.920E-1	8.791E-5			2.335E-5	1.456E-2	6.708E-2		3.138E-2	5.928E-3	2.296E-3	1.502E-3
7	8.789E-1	1.117E-4			9.185E-6	8.250E-3	4.645E-2	1.061E-1		6.183E-2	2.275E-2	2.111E-2
8	8.290E-1	1.587E-4			4.979E-7	7.024E-4	4.908E-3	1.382E-2	4.316E-2		5.965E-2	3.468E-2
9	7.599E-1	2.361E-4			3.708E-8	7.802E-5	5.851E-4	1.896E-3	5.607E-3	2.124E-2		3.740E-2
10	7.269E-1	4.171E-4				1.420E-5	1.180E-4	3.263E-4	1.369E-3	3.258E-3	9.883E-3	

Table 5
Numerical performance parameters (MHTGR-350).

	k-eff	Convergence Rate	Expected Iterations	Actual Iterations	Computing Time (msec)
AFEN RM	1.093230	0.97531	645	450	290
AFEN	1.092759	0.97936	773	545	971
FDM RM	1.050210	0.97248	578	413	55

benchmark problem representing the HTGR core. Although the difference varies depending on the problem, the refined AFEN response matrix method consistently shows a smaller convergence rate for the two benchmark problems. It also shows a more than three times speedup in computing time for both problems. The reason of the speedup is explained differently for each of them: that for the VVER-440 problem is mainly due to a big cut in the number of iterations caused by a far smaller convergence rate, on the other hand, that for MHTGR-350 problem is mainly due to the advantage of the computational efficiency of the response matrix method.

In addition, it was found from the results of the error analyses that the convergence rate of this method is smaller than or at least comparable to that of the FDM response matrix. This was presented as a cause of poor performance of the nonlinear FDM schemes in accelerating the refined AFEN response matrix method.

Finally, it can be concluded in short that the refined AFEN response method significantly outperforms the conventional AFEN method in analyzing the LWR cores and the HTGR cores.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by a grant from the Nuclear R&D Program of the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (Grant code: 2017M2A8A1014757).

References

[1] H.C. LEE, T.Y. HAN, C.K. JO, J.M. NOH, Development of the HELIOS/CAPP code system for the analysis of pebble type VHTR cores, Ann. Nucl. Eng. 71 (2014) 120.
[2] J.M. NOH, N.Z. CHO, A new approach of analytic basis function expansion to neutron diffusion nodal calculation, Nucl. Sci. Eng. 116 (1994) 165.
[3] N.Z. CHO, J.M. NOH, Analytic function expansion nodal method for hexagonal geometry, Nucl. Sci. Eng. 121 (1995) 245.
[4] N.Z. CHO, Y.H. KIM, K.W. PARK, Extension of analytic function expansion nodal method to multigroup problems in hexagonal-Z geometry, Nucl. Sci. Eng. 126 (1997) 35.
[5] S.W. WOO, N.Z. CHO, J.M. NOH, The analytic function expansion nodal method refined with transverse gradient basis functions and interface flux moments, Nucl. Sci. Eng. 139 (2001) 156.
[6] N.Z. CHO, J.J. LEE, Analytic function expansion nodal (AFEN) method in hexagonal-z three dimensional geometry for neutron diffusion calculation, J. Nucl. Sci. Technol. 43 (2006) 1320.
[7] J.Y. Cho, H.G. Joo, B.O. Cho, S.Q. Zee, Hexagonal CMFD Formulation Employing Triangle-Based Polynomial Expansion Nodal Kernel, ANS M&C Topical Meeting, Salt Lake City, Utah, USA, 2001, 2001.
[8] K.S. Smith, Nodal method storage reduction by non-linear iteration, Trans. Am. Nucl. Soc. 44 (1983) 265.
[9] K.S. MOON, N.Z. CHO, J.M. NOH, Acceleration of the analytic function expansion nodal method by two-factor two-node nonlinear iteration, Nucl. Sci. Eng. 132 (1999) 193.
[10] J.M. NOH, An AFEN equivalent hexagonal nodal method based on the single-node nonlinear FDM response matrix, *Trans. KNS Autumn Mtg*, Yeosu, Korea (2018). October 25-26.
[11] H.G. Joo, et al., One-node solution based nonlinear analytic nodal method, in: Trans. KNS Autumn Mtg, Pohang, Korea, 2000. May 26-27 (in Korean).
[12] J.M. NOH, et al., A general approach to multigroup extension of the analytic function expansion nodal method based on matrix function theory, in: Proc. 1996 Joint Intl. Conf. Mathematical Methods and Super Computing for Nuclear Applications, Saratoga Springs, New York, Octorber 6-10 vol. 1, American Nuclear Society, 1997, p. 144, 1997.
[13] N. DUNFORD, J.T. SCHWARTZ, Linear Operators Part I: General Theory, Interscience Publishers, New York, 1971.
[14] K. KOEBKE, A new approach to homogenization and group condensation, in: Proc. IAEA Technical Committee Mtg, International Atomic Energy Agency, Lugano, Switzerland, 1978, p. 303. November 1978, IAEA-TECDOC 231.
[15] K.S. SMITH, Spatial Homogenization Methods for Light Water Reactor Analysis, Massachusetts Institute of Technology, 1980. PhD Thesis.
[16] S. Nakamura, Computational Methods in Engineering and Science, with Applications to Fluid Dynamics and Nuclear Systems, John Wiley & Sons, Inc., 1977.
[17] Y.A. CHAO, Y.A. SHATILLA, Conformal mapping and hexagonal nodal methods-II; implementation in the ANC-H code, Nucl. Sci. Eng. 121 (1995) 210.
[18] *CAPP v3.0 Verification Report*, KAERI/TR-7647/2019, Korea Atomic Energy Research Institute KAERI, 2019.