Proceedings of the Korean Nuclear Society Autumn Meeting
Taejon, Korea, October 2000

The *p*-Version of the Finite Element Method and Domain Decomposition Approach for Two-Dimensional Diffusion Equations

Keon Woo Park and Nam Zin Cho

Korea Advanced Institute of Science and Technology
Department of Nuclear Engineering
373-1 Kusong-dong, Yusong-gu
Taejon, Korea 305-701

Abstract

Based on the p-version of the finite element method and domain decomposition, a novel code is developed for solving the two-dimensional multigroup neutron diffusion equations. In this study, the reactor domain is decomposed into subdomains and each subdomain is solved independently by the *p*-version of the finite element method, and the subdomains are coupled by use of incoming and outgoing partial currents along the interfaces. The boundary partial currents are obtained analytically from fluxes, which are approximated by hierarchical basis functions. For this domain decomposition, the computation can be implemented on parallel computers. The method was tested on the IAEA-2D benchmark problem and a MOX-fuel loaded core. The results show that the method can generate accurate core multiplication factor and assembly average quantities.

I. Introduction

The finite element method, one of the widely used numerical methods for solving certain types of differential equations, is based on approximating the solution by piecewise smooth functions, specifically polynomials. In general, there are two kinds of finite element method: the *h*-version and the *p*-version. In the *h*-version, polynomial basis functions are fixed over each element and accuracy is achieved by refining the mesh size *h*. In the *p*-version, the mesh is fixed and accuracy is achieved by increasing the degree *p* of polynomial basis functions[1].

There are several finite element methodologies in reactor analysis, and the methods are all *h*-versions. Kang and Hansen[3] studied the application of the finite element method to problems in neutron diffusion in space, energy, and time. They used Hermite polynomial as basis functions. Nakata and Martin[4] presented finite elements with response matrix approach (FERM). This approach uses a dual finite element basis, one for fluxes and one for partial currents respectively. They divided the problem into local and global core levels. Galerkin weighted residual method was used to obtain partial currents in both levels. Dilber and Lewis[5] developed variational nodal method with response matrix for diffusion and transport

problems. They used a functional that guarantees the satisfaction of nodal balance. Hennart *et al*.[6] developed higher order nodal finite elements formulations, which use transverse integration procedure.

In this paper, the *p*-version of the finite element method and domain decomposition are used to solve multigroup neutron diffusion equations. The problem domain is decomposed into several subdomains, typically homogenized fuel assemblies, and each of subdomain is solved independently with coupled coefficients. The coupled coefficients for each subdomain interfaces are incoming and outgoing partial currents, obtained analytically from fluxes, which are approximated by hierarchical basis functions. Since this method do not use response matrix methodology, it does not invert the element stiffness matrix directly. To efficiently solve the element stiffness matrix iteratively, standard multi-*p* V (SMPV) cycle is adopted in inner iterations. The computation is implemented on parallel computers.

II. Theory and Methodology

1. Finite element method and interface coupling

Without loss of generality, the *G*-group neutron diffusion equations are written as

$$-\nabla D_g \cdot \nabla \Phi_g + S_{rg} \Phi_g = S_g, g = 1,2,\Lambda, G, \tag{1}$$

where the source term S_g contains the fission and scattering contributions,

$$S_g = \frac{?_g}{k_{eff}} \sum_{g'=1}^{G} ?S_{fg'} \Phi_{g'} + \sum_{g' \neq g}^{G} S_{sgg'} \Phi_{g'} \cdot$$

Eq. (1) is subject to the natural boundary condition on the surfaces,

$$J_g^{\pm} = \frac{1}{4} \Phi_g \mu \frac{1}{2} D_g \frac{\partial \Phi_g}{\partial n}. \tag{2}$$

We first multiply Eq. (1) by a suitable set of test functions Ψ, and then perform an integration with Green's formula of integration by parts (we skip group index g to simplify the following equations)[2],

$$\int_\Omega (D \nabla \Phi \cdot \nabla \Psi + \Sigma_r \Phi \Psi) d\Omega = \int_\Omega S \Psi d\Omega + \int_\Gamma D \frac{\partial \Phi}{\partial n} \Psi d\Gamma, \tag{3}$$

where Γ stands for total boundaries.

In Eq. (1), Ω is a fairly regular domain, can thus be discretized in rectangular elements Ω_e:

$$\Omega = \sum_{e=1}^{E} \Omega_e \cdot$$

In Eqs. (2) and (3), the following equation can be achieved :

$$A(\Phi, \Psi) = f(\Psi), \tag{4}$$

where

$$A(\Phi, \Psi) = \sum_{e=1}^{E} \left\{ \int_{\Omega_e} (D_e \nabla \Phi_e \cdot \nabla \Psi_e + \Sigma_{re} \Phi_e \Psi_e) d\Omega_e + \int_{\Gamma_e} \frac{1}{2} \Phi_e \Psi_e d\Gamma_e \right\}, \tag{5}$$

$$f(\Psi) = \sum_{e=1}^{E} \left\{ \int_{\Omega_e} S_e \Psi_e d\Omega_e + 2 \int_{\Gamma_e} J_e^- \Psi_e d\Gamma_e \right\}, \tag{6}$$

where the coefficients D_e and Σ_{re} are assumed to be constant in finite element Ω_e.

After discretizing the problem domain into finite elements, boundary conditions are used as follows (boundary conditions of bottom and top are similar to those of left and right)[8] :

1) Reflective boundary condition (left) :

$$J^-\Big|_{\Gamma_e^{left}} = J^+\Big|_{\Gamma_e^{left}} = \frac{1}{4}\Phi_e - \frac{1}{2}D_e\frac{\partial\Phi_e}{\partial n}\Big|_{\Gamma_e^{left}} \tag{7}$$

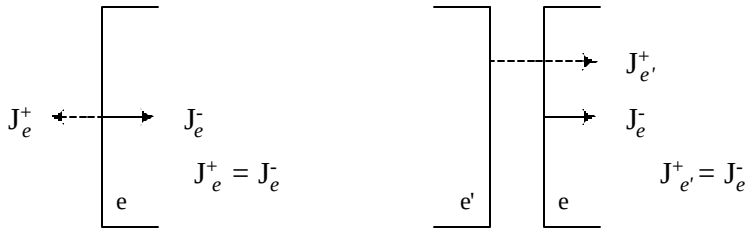
2) Vacuum boundary condition (right) :

$$J^-\Big|_{\Gamma_e} = \frac{1}{4}\Phi_e + \frac{1}{2}D_e\frac{\partial\Phi_e}{\partial n}\Big|_{\Gamma_e} = 0 \tag{8}$$

3) Interface coupling condition : This is by use of a relation between incoming (J^-) and outgoing (J^+) partial currents along the subdomain interfaces. That is, the incoming partial currents of subdomains are equal to the outgoing partial currents from its neighboring subdomains.

$$J^-\Big|_{\Gamma_e^{left}} = J^+\Big|_{\Gamma_{e'}^{right}} = \frac{1}{4}\Phi_{e'} - \frac{1}{2}D_{e'}\frac{\partial\Phi_{e'}}{\partial n}\Big|_{\Gamma_{e'}^{right}} \tag{9}$$

Since the reactor domain is divided into subdomains, it is necessary to couple each other among the subdomains on the subdomain interfaces.



a) Reflective boundary condition b) Interface coupling condition

Fig. 1. The boundary conditions of subdomains at interface.

2. Hierarchical shape functions and standard multi-*p* (SMPV) cycle

In the *p*-version of the finite element method, hierarchical shape functions are used. We discuss briefly here a set of hierarchical shape functions that are associated with quadrilateral elements[7].

Let $\Omega_e = (-1,1)\times(-1,1)$ be a standard quadrilateral element (see Fig. 2). There are four nodal shape functions associated with each vertex,

$$
\begin{aligned}
N_1(x,h) &= \frac{1}{4}(1-x)(1-h),\\
N_2(x,h) &= \frac{1}{4}(1+x)(1-h),\\
N_3(x,h) &= \frac{1}{4}(1+x)(1+h),\\
N_4(x,h) &= \frac{1}{4}(1-x)(1+h),
\end{aligned}
\tag{10}
$$

which are exactly the same as the shape functions of four-noded quadrilaterals. For $p \geq 2$, there are $p-1$ shape functions associated with every side. For example, the bottom side has the following form :

$$
N_i^{(b)} = \frac{1}{2}(1-h)j_i(x), \quad i = 2,\Lambda\ p, \tag{11}
$$

where

$$
j_i(x) = \sqrt{\frac{2i-1}{2}}\int_{-1}^{x} P_{i-1}(t)dt, \tag{12}
$$

and P_i is the ith Legendre polynomial.

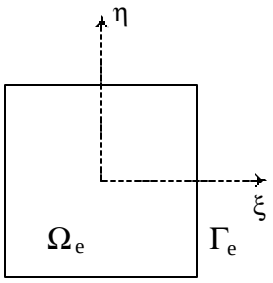


Fig. 2. The standard quadrilateral element.

For $p \geq 4$, there are $(p-2)\times(p-3)/2$ internal shape functions :

$$
N_{i,j} = j_{i+1}(x)j_{j+1}(h), \quad 0 \leq i+j \leq p-4. \tag{13}
$$

Now, in element Ω_e, the flux shape Φ_e is expressed as follows :

$$
\begin{aligned}
\Phi_e &= \sum_{i=1}^{4} f_i N_i \\
&+ \sum_{i=2}^{p}\sum_{u} f_i^{(u)} N_i^{(u)} \qquad (\text{if } p \geq 2,\ u = \text{left, right, bottom, top}) \\
&+ \sum_{i,j=0}^{i+j\leq p-4} f_{i,j} N_{i,j} \quad (\text{if } p \geq 4).
\end{aligned}
\tag{14}
$$

If p=8, there are 47 shape functions consisting of 4 nodal, 28 side and 15 internal shape functions.

This set of shape functions is hierarchical; that is, the finite element space V_{p-1} that is spanned by the above shape functions with degree up to $p-1$ is completely embedded in the space V_p that is spanned by

the above shape functions with degree up to *p*. These characteristics are used conveniently for a standard multi-*p* V cycle method, which looks exactly like the multigrid V-cycle algorithm[7] ; the only difference is that the multilevel computations here are conducted at levels of different degrees of shape functions, not at different grids.

3. Parallel implementation

In this paper, parallel computation was performed with MPI (Message Passing Interface). The message passing model posits a set of processes that have only local memory but are able to communicate with other processes by sending and receiving message[10]. The efficiency of this parallel computation depends on the number of meshes. This is because many meshes need much communication time. Figure 3 shows the configuration of sharing spaces with each processor. All computations were performed in KAIST*GALAXY, a Linux based PC clustering system.

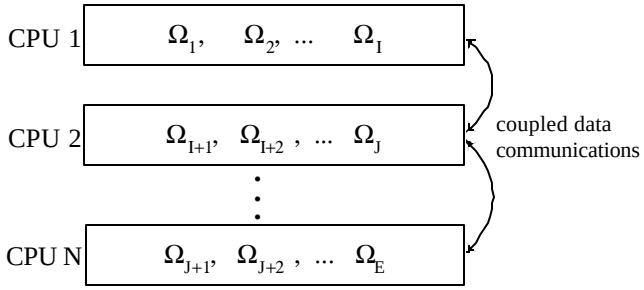


Fig. 3. Configuration of sending and receiving data in each processor.

III. Numerical Results and Discussion

A code was developed based on the *p*-version of the finite element and domain decomposition methodology. The code divides the reactor domain into several subdomains, which is solved independently, and the subdomains are coupled using incoming and outgoing partial currents, which is obtained analytically from fluxes, that are approximated with hierarchical basis functions. For subdomains are solved independently, parallel implementation was performed.

The code uses power method in outer iterations and SMPV (Standard Multi-*P* V cycle) in inner iterations. Results of the code are compared with those of VENTURE[9], which uses the fine-mesh mesh-centered finite difference method. All calculations were done using the 1/4 symmetry and convergence criteria were 10^{-8} for reactor eigenvalue and 10^{-7} for assembly average fluxes. All benchmark problems were solved with one element per assembly (so an element size is equal to the assembly length), and assembly homogenized cross sections were used. For purposes of summarizing results for comparison, all the errors were represented by taking the same form :

$$\text{Relative percent error} = \frac{\text{tested value} - \text{reference value}}{\text{reference value}} \times 100\ (\%),$$

while the maximum errors with sign are maximal in the sense of absolute values, and the average errors are defined as absolute arithmetic average values.

1. IAEA-2D Benchmark Problem

To test performance of the newly developed code, the accuracy was examined to the well known IAEA-2D benchmark problem[9]. The IAEA-2D benchmark problem is composed of a two-zone core containing 177 fuel assemblies each having a length of 20 *cm.* The core is surrounded radially by 20 *cm* of pure water and four rodded assemblies are symmetrically deployed inside.

Numerical tests for this problem are given in Table 1. The results of the assembly average flux (fast and thermal) errors for p=4,6,8 are displayed in Figs. 4-5. In the figures, assembly average fluxes are compared with those of VENTURE, which uses 0.3125 *cm* for mesh sizes. It is observed that when degree p=4, the maximum errors in fast and thermal assembly are –2.17 and –2.45% respectively, but these errors are significantly reduced with the increase of degree p. If p=5 or higher, average flux errors in both codes are less than 0.5%, and when p=6, even maximum flux errors are within 0.5%. The maximum errors occur in the assemblies adjacent to the reflector.

Table 1. Summary of the results for IAEA-2D problem

Degree	p=4	p=5	p=6	p=7	p=8
Eigenvalue error (%)*	0.00	0.00	0.00	0.00	0.00
Max. assembly fast flux error (%)	-2.17	0.61	-0.25	-0.07	0.03
Avg. assembly fast flux error (%)	0.78	0.27	0.08	0.02	0.01
Max. assembly thermal flux error (%)	-2.45	0.73	-0.18	0.12	0.11
Avg. assembly thermal flux error (%)	0.91	0.19	0.09	0.03	0.02
CPU times (sec)	6.5	7.7	19.3	27.5	36.3

* Reference eigenvalue : 1.0295858

The code uses domain decomposition, thus parallel implementations can be performed. Table 2 and Fig. 6 show the results when the code is applied on parallel computers. The speedup is defined as $T_1(N)$ / $T_p(N)$, where $T_p(N)$ is the time required to solve the given problem of size N using P processors.

In the table and figure, it is observed that higher degree p shows good results compared to relatively lower degree p.

Table 2. Parallel implementations for IAEA-2D problem

Degree p	1 CPU		2 CPUs			4 CPUs		
	#outer	Time	#outer	Time	speedup	#outer	Time	speedup
4	590	6.5	707	6.9	0.9	707	4.2	1.5
6	707	19.3	720	13.4	1.4	720	7.8	2.5

8	642	36.3	689	23.3	1.6	689	13.1	2.8

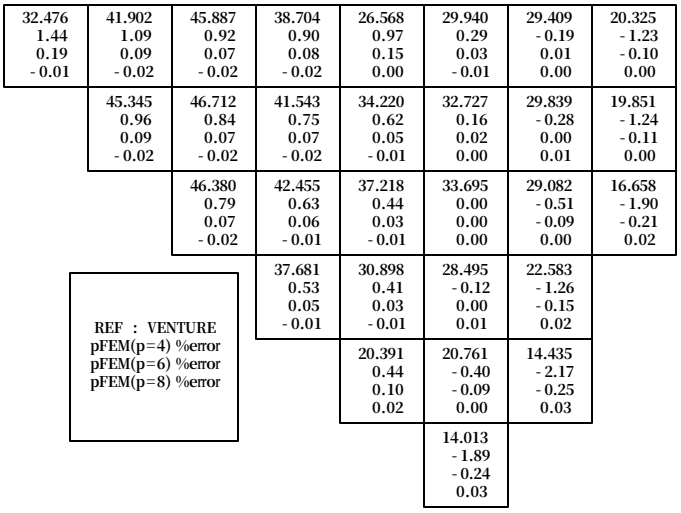


Fig. 4. Results of the IAEA-2D benchmark problem : fast flux.

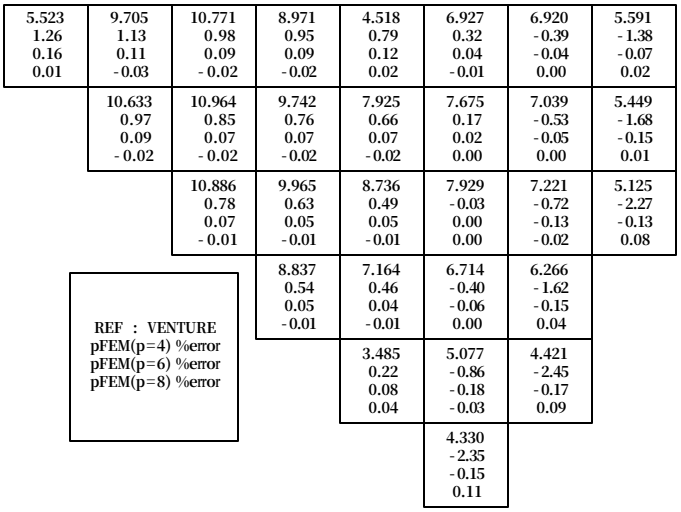


Fig. 5. Results of the IAEA-2D benchmark problem : thermal flux.

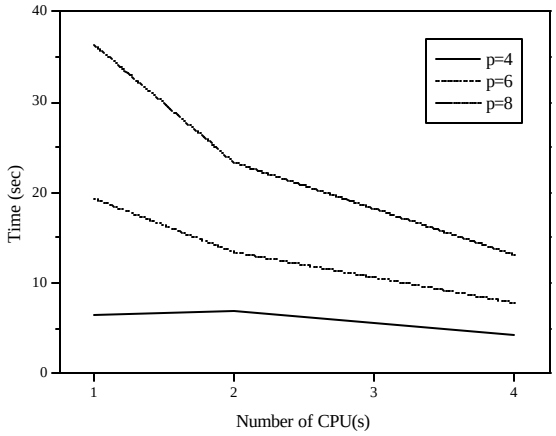


Fig. 6. Comparison of computing times for various CPU(s).

2. MOX-fuel loaded core

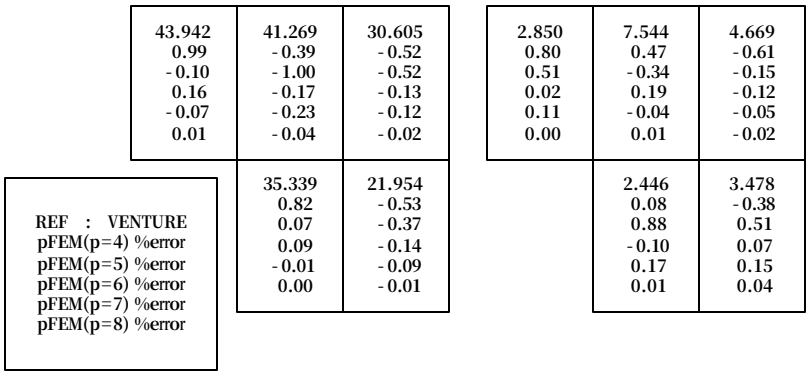
This is the benchmark test for multigroup case. We tested the code on a four-group reactor core configuration which consists of three types of homogeneous fuel assemblies including MOX fuel assemblies, which have a length of 21 *cm* [12]. The results are compared with those of VENTURE, which uses 0.21 *cm* for mesh sizes.

The maximum and average errors in assembly power and eigenvalue are summarized in Table 4 and Fig. 7. In the table and figure, it is observed that when the degree p=6 is used, the maximum assembly flux errors are decreased within 0.2%. The results show that the code is well applicable to multigroup case.

Table 4. Summary of results for MOX-fuel loaded core

Degree	p=4	P=5	p=6	p=7	p=8
Eigenvalue error*	-0.04	-0.03	-0.01	0.00	0.00
Max. assembly fast flux error (%)	0.99	-1.00	-0.17	-0.23	-0.04
Avg. assembly fast flux error (%)	0.59	0.49	0.14	0.12	0.02
Max. assembly thermal flux error (%)	0.80	0.88	0.19	0.17	0.04
Avg. assembly thermal flux error (%)	0.47	0.42	0.11	0.10	0.02
CPU times (sec)	1.1	1.7	3.7	4.2	6.1

* Reference eigenvalue : 1.0672635



a) Total fast flux (group 1,2,3) b) Thermal flux (group 4)

Fig. 7. Results of MOX-fuel loaded core analysis : flux.

IV. Conclusions

A computer code based on the *p*-version of the finite element method and the domain decomposition was developed to solve the multigroup neutron diffusion equations. The code considers the problem domain decomposed into several subdomains. These subdomains are solved independently with coupled data, which are incoming and outgoing partial currents. These partial currents are obtained from fluxes, which are solved with *p*-version of the finite element method. This code was tested on the IAEA-2D benchmark problem for large core case and a small core including MOX fuel assemblies for multigroup case. The results show accurate assembly average quantities for both cases.

Parallel implementation was performed on IAEA-2D benchmark problem. It was observed that when 4 CPUs were used for solving the problem, computing times were significantly reduced.

Application to these benchmark problems shows that this code can provide accurate solutions for two-dimensional problems, but requires relatively long computing times. Since the code do not use acceleration schemes, this limitation may be overcome when fully parallel algorithms, or some acceleration schemes in inner and outer iterations such as coarse group rebalance (CGR)[13] are considered.

References

[1] I. Babuska *et al*., "The *p*-version of the Finite Element Method," *SIAM. Numer. Anal*., **18**, 515 (1981).

[2] E. B. Becker *et al*., *Finite Element an Introduction*, **1,** (1981).

[3] C. M. Kang and K. F. Hansen, "Finite Element Methods for Reactor Analysis," *Nuclear Science and Engineering*, **51**, 456 (1973).

[4] H. Nakata and William R. Martin, "The Finite Element Response Matrix Method," *Nuclear Science and Engineering*, **85**, 289 (1983).

[5] I. Dilber and E. E. Lewis, "Variational Nodal Methods for Neutron Transport," *Nuclear Science and*

Engineering, **91**, 132 (1985).

[6] J. P. Hennart *et al*., "Efficient Higher Order Nodal Finite Element Formulations for Neutron Multigroup Diffusion Equations," *Nuclear Science and Engineering*, **124**, 97 (1996).

[7] Ning Hu and I. Norman Katz , "Multi-*p* Methods : Iterative Algorithms for the *p*-version of the Finite Element Analysis ," *SIAM J. Sci. Comput*., **16**, 1308 (1995).

[8] Hong Wu Cheng and Nam Zin Cho, "A Spectral-Galerkin Coarse-Mesh Approach to Solving the Two-Dimensional Multigroup Diffusion Equations," *Ann. Nucl. Energy*, **24**, 883 (1997).

[9] *ANL Benchmark Problem Book*, ANL-7416, Suppl. 3, Argonne National Laboratory (1985).

[10] M. Snir *et al*., *MPI: The Complete Reference*, The MIT Press (1996).

[11] Jae Man Noh and Nam Zin Cho, "A New Approach of Analytic Basis Function Expansion to Neutron Diffusion Nodal Calculation," *Nuclear Science and Engineering*, **116**, 165 (1995).

[12] Nam Zin Cho and Jae Man Noh, "A Multigroup Diffusion Nodal Scheme : Hybrid of AFEN and PEN Methods," *Proceedings of the Korean Nuclear Society Autumn Meeting*, 29, Seoul, Korea (1995).

[13] Do Sam Kim and Nam Zin Cho, "Acceleration of Three-Dimensional AFEN Nodal Codes via Coarse Group Rebalance and Direct Matrix Inverse," *Proc. M & C, Reactor Physics and Environmental Analysis in Nuclear Applications*, pp. 168-174, Madrid, Spain (1999).