An Experiment of Robust Parallel Algorithm for the Eigenvalue problem of a Multigroup Neutron Diffusion based on modified FETI-DP : Part 2

Jonghwa Chang
Korea Atomic Energy Research Institute,989-111 Daedeokdaero, Yuseong-gu, Daejeon, Korea, 305-353
jhchang@kaeri.re.kr

1. Introduction

Detailed modeling of a reactor core requires huge number of unknown variables which results large memory size as well as long computing time. Traditionally, spatial homonization, coarse mesh analysis, and pin power reconstruction are used to find detailed information such as peak power.

Today, we can use a computer cluster consist of a few hundreds CPUs with reasonable budget. Such computer system enables us to do detailed modeling of reactor core. The detailed modeling will improve the safety and the economics of a nuclear reactor by eliminating un-necessary conservatism or missing consideration. To take advantage of such a cluster computer, efficient parallel algorithms must be developed.

Mechanical structure analysis community has studied the domain decomposition method to solve the stress-strain equation using the finite element methods. One of the most successful domain decomposition method in terms of robustness is FETI-DP. [1], [2] We have modified the original FETI-DP to solve the eigenvalue problem for the multi-group diffusion problem in previous study. [3]

In this study, we report the result of recent modification to handle the three-dimensional sub-domain partitioning, and the sub-domain multi-group problem.

2. Recipe

2.1 Prescription of Algorithm

The multi-group diffusion eigenvalue problem is written in following form.

$$\left(D_g\nabla^2+\sigma_g\right)\phi_g=\lambda\chi_g s+\sum_{g'\neq g}\sigma_{g'\to g}\phi_{g'}$$

where $s=\sum_g \nu\sigma_{fg}\phi_g$.

A general eigenvalue problem can be setup to solve above diffusion equation by using appropriate method such as finite element method. [4] After discretization, the problem is written in following matrix form.

$$A\phi=\lambda B\phi.$$

This problem is converted to the inverse eigenvalue problem for faster and stable convergence.

$$A^{-1}B\phi=\frac{1}{\lambda}\phi.$$

We have adopted Arnoldi method to find eigenvalue in the Krylov subspace. [5] Householder projection method [6] is adopted to find the orthogonal bases which is required to find approximation vectors in the Krylov subspace. [7]

Arnoldi iteration requires a product of the matrix for eigenvalue problem or a solution of a linear system for the inverse eigenvalue problem;

$$A\varphi=By$$

Above linear system is solved by FETI-DP parallel algorithm. The global equation on the Lagrange multiplier will be established by collecting results at each subdomain. To solve the resulting linear system of Lagrange multipliers, we adopted PBiCGSTAB (preconditioned bi conjugate gradient stabilized), which use two residual vectors as presented at previous study. [3] The FETI-DP procedure requires solution of the multi-group equation in each subdomain. The subdomain problem is solved by the PGMRES (Preconditioned Generalized Minimum Residual method). [7]

2.2 PGMRES

The multi-group diffusion equation for each sub-domain can be written as following.

$$\begin{pmatrix} A_1 & -S_{2\to1} & -S_{3\to1} & \cdots & -S_{G\to1} \\ -S_{1\to2} & A_2 & -S_{3\to2} & & -S_{G\to2} \\ -S_{1\to3} & -S_{2\to3} & A_3 & & -S_{G\to3} \\ \vdots & & & \ddots & \vdots \\ -S_{1\to G} & -S_{2\to G} & -S_{3\to G} & \cdots & A_G \end{pmatrix}\begin{pmatrix}\phi_1\\ \phi_2\\ \phi_3\\ \vdots\\ \phi_G\end{pmatrix}=\begin{pmatrix}b_1\\ b_2\\ b_3\\ \vdots\\ b_G\end{pmatrix}$$

Above equation can be solved using the block successive over-relaxation [8] as presented in previous study. [3] The speed of bSOR method is sensitive to optimum choice of relaxation parameter, ω. However, it is difficult to select optimum relaxation parameters for each sub-domain. We adopted PGMRES (Pre-conditioned Generalized Minimum Residual Method) with a pre-conditioner based on the successive over-relaxation in this study. The bSSOR(block symmetric successive over-relaxation) pre-conditioner can be expressed in following form.

$$M=\left(I-\omega ED^{-1}\right)\left(D-\omega F\right),$$

where D, E, and F are the diagonal, lower, and upper sub-matrix of A, respectively. [7] The preconditioned linear system is expressed in following form.

$$M^{-1}A\phi = M^{-1}b$$

Since M is product of a block lower triangular matrix and a block upper matrix, inverse of pre-conditioner can be easily computed with a forward sweep followed by a backward sweep. Sub matrices of *D* are symmetric positive definite when proper FEM discretization is applied. Inversion of matrix *D* can be performed using pre-computed Cholesky decomposed matrices.

Significant speed up was achieved with modification to use PGMRES instead of bSOR which was used in previous presentation. Dependency on the relaxation parameter in the pre-conditioner is not sensivtive compared with bSOR. We have fixed ω as 1.5 in this study.

2.3 Sub domain partitioning

FETI-DP algorithm requires conformal geometry at the interface between sub-domains. The continuity of flux and current at the sub-domain interfaces (dual points) are satisfied through the Lagrange multipliers. The flux continuity at the primal (cornor) point where more than 3 subdomains meet, are satisfied by a global equation on the primal point flux.

Total computing time is sum of 1) the CPU time spent to solve the global equation for primal nodes and the FETI iteration for dual nodes, 2) the longest CPU time spent to solve the sub-domain-wise FEM for internal points and dual points, and 3) the time required to exchange the interface values with global equation. So, we can choose following criteria on the sub-domain partitioning to achieve minimum computing time with given number of subdomains.

a. Number of internal and dual points are even,
b. Total number of dual points is small,
c. Total number of primal points is small.

This is a vertices-edges graph partitioning problem with optimization criteria. Criterion a) is satisfied by minimizing the suprenum of number of internal and dual points of sub-domain. Criterions b) is satisfied by minimizing the sum of the number of dual points. We can impose appropriate weighting factor to define the cost of segmentation. The graph partitioning problem is an NP-complete problem which requires factorial times large operations that results prohibitively long computing time. Many heuristics approaches were proposed to obtain reasonable solution. [9]

We used a graph partitioning software MeTIS. [10] For conformal partitioning, the vertex of graph is chosen as the finite element volume.

Figure 1 displays the volume elements (in 2D) and the graph. Centers of triangles are used as the vertices of a graph. Dual points are corresponding to the edges of partitioned graph. In the example of Figure 1, 14 vertices are partitioned into 2 equal parts with 7 vertices each. 3 edges are connected between the partitions.

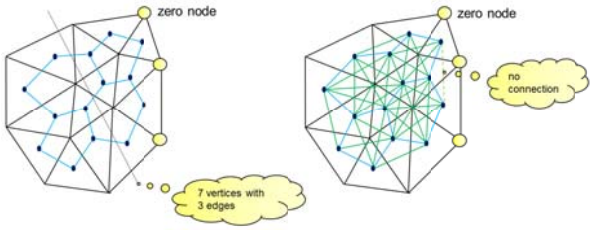

Figure 1. Node graph (left: edge graph, right: vertices connection graph)

A simple method to generate graph is based on neighboring edges as shown in left side of Figure 1. When there is flux zero boundary condition, the size of sub domain matrix is significantly reduced. So, it may be reasonable to count the number of matrix size, or the non-zero connected edges as displayed on right side of Figure 1.

Figure 2 displays a result of sub-domain partitioning in 16 segment using MeTIS.

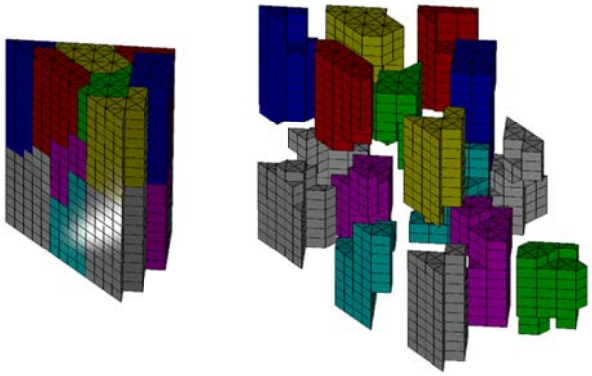

Figure 2. A sub-domain partition in 16 segments (IAEA3D problem)

3. Results

IAEA3D benchmark problem is solved for our experiment. [11] Basic mesh size of the IAEA3D problem is 20 cm in planar and axial direction. We studied 2x2x2 refined mesh with L2 bases function in both directions. Total number of nodes is 36630 excluding the zero boundary points. For 2 group problem, the global matrix has 73260 unknowns in 36 band structure.

We used a 24 node linux-cluster in this study. Each node of the cluster is an Intel Core2 Quad CPU Q9650 which has four 3 GHz cores. Cluster nodes are connected by standard Gigabit Ethernet.

3.1. Sub domain partitioning

The node graph is generated by taking centers of triangular pipes as graphic vertices. Vertex adjacency is determined by existence of common FEM base point between neighboring triangular pipe as shown at the right side of Figure 1. We adopted the vertex weight method for partitioning the graph. Weight of a vertex is taken as the number of nodes in the corresponding

FEM base element excluding zero boundary node. There are 9 nodes on L2 bases for a normal FEM element (or graph vertex).

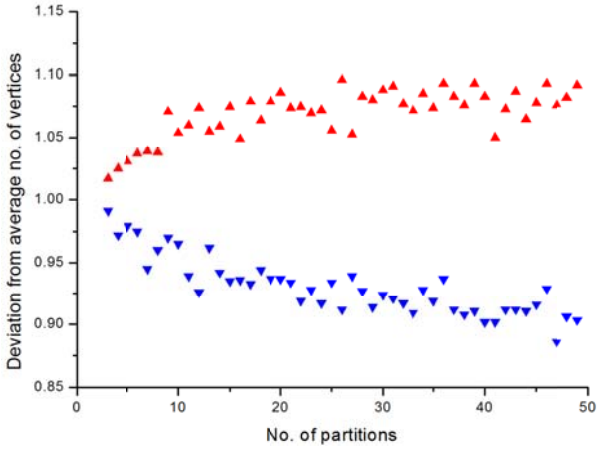

Figure 3. Graph partition

Figure 3 displays the relative number of vertices distribution resulting from the graph partitioning. Evenness decreases as the number of partition increase. About 10% deviation is observed for large number of partitions.

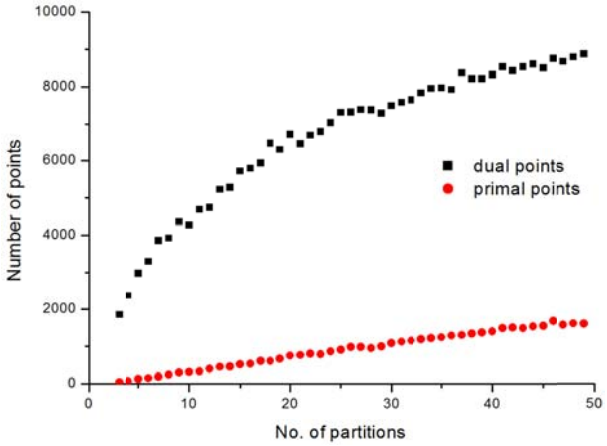

Figure 4. Primal and dual points

Figure 4 displays the number of primal and dual points. Number of primal points which affects the size of global equation, increases almost linearly as the number of partitions is increasing. The number of dual points increases nearly proportional to 2/3 power of the number of partitions. Most of the communication burden is due to exchange of the flux value at the dual points, it is important to reduce the number of dual points.

3.2. CPU time

Figure 5 displays the elapsed CPU time depending on the partitioning. Two sets of test calculations in different day. Reported CPU time is sum of the user time and the system time. The CPU time varies depending on the computing environment where the other jobs are processed simultaneously. Total CPU time is decreasing as the number of partition is increasing as expected.

Total CPU time is sum of 1) the time for primal node problem, 2) the maximum time used for sub-domain problems, and 3) the time for MPI communication between main process and sub-processes.

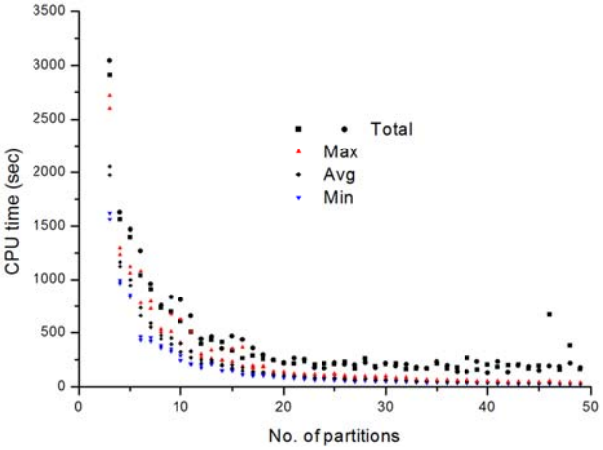

Figure 5. CPU time

Figure 5 displays the maximum, the average, and the minimum CPU time used by sub-domains. Those times are decreasing as the number of partitions are increasing. The gap between the total time and the maximum time and is due to 1) and 3).

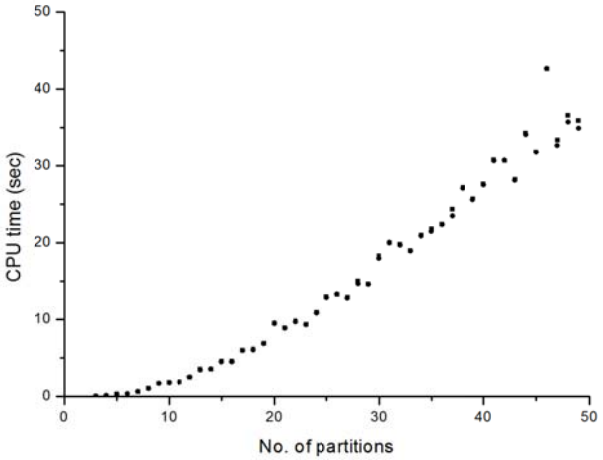

Figure 6. CPU time used by primal point problem

Figure 6 displays the time used by primal node problem to match global solution as specified by FETI. As the number of primal points increases linearly with number of partition, the computing time may increase quadratically after a certain minimum. Comparing the CPU time shown at Figure 6 with the number of primal points shown at Figure 4, the CPU time rise at 46 partitions is due to increase of the number of primal points at the partition. This situation may be improved with better partitioning strategy.

Overall computing time decreases as number of partition increases. However the overall CPU time may

increases for large number of partitions due to increase in the number of primal points.

Effect of MPI communication time is not observed in this study.

3. Conclusion and Further Study

Modified FETI-DP algorithm has been successfully applied for the eigenvalue problem of multi-group neutron diffusion equation. The overall CPU time is decreasing as number of sub-domains (partitions) is increasing. However, there may be a limit in decrement due to increment of the number of primal points will increase the CPU time spent by the solution of the global equation.

Even distribution of computational load (criterion a) is important to achieve fast computation. The sub-domain partition can be effectively performed using suitable graph theory partition package such as MeTIS. However actual CPU time is not only depending on the size of the matrix but also depending on the spectral radius matrix which is, in turn, dependent on the region-wise diffusion coefficients. Better strategy can be developed for computational load balancing of sub-domains considering the neutron cross-section such as diffusion coefficient.

REFERENCES

[1] C. Farhat, M. Lesoinne, and K. Pierson, "A scalable dual-primal domain decomposition method," *Numer. Linear Algebra Appl.*, vol. 7, pp. 687-714, 2000.

[2] C. Farhat and et al., "FETI-DP: a dual- primal unified FETI method - part I: A faster alternative to the two-level FETI method," *Int. J. Numer. Meth. Engng*, vol. 50, pp. 1523-1544, 2001.

[3] Jonghwa Chang, "An Experiment of Robust Parallel Algorithm for the Eigenvalue problem of a Multigroup Neutron Diffusion based on modified FETI-DP," in *Trans. KNS spring meeting, Jeju, Korea, May 29-30, 2014.*

[4] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The Finite Element Method: Its Basis and Fundamentals, 6th ed.*: Elsevier, 2005.

[5] Y. Saad, *Numerical Methods for Large Eigenvalue Problems, 2nd ed.*: SIAM, 2011.

[6] H. F. Walker, "Implementation of the GMRES method using Householder transformations," *SIAM K. Sci. Stat. Comput.*, vol. 9, no. 1, pp. 152-163, Jan. 1988.

[7] Y. Saad, *Iterative Methods for Sparse Linear Systems.*: PWS Pub., 1996.

[8] S. Nakamura, *Computational Methods in Engineering and Science*.: John Wiely & Sons, 1977.

[9] Wikipedia. Graph partition. [Online]. http://en.wikipedia.org/wiki_Graph_partition

[10] G. Karypis. (2014, Mar.) METIS - Serial Graph and Hypergraph Partitioning Software. [Online]. http://glaros.dtc.umn.edu/gkhome/views/metis/index.html

[11] G. Casini, "Thermal Benchmark Experiment Compilation," CEC Euratom, NEACRP-L-114, 1974.

[12] M. Galassi and et al., "GNU Scientific Library," The GSL Team, 2011.

[13] T.A. Davis, *Direct Methods for Sparse Linear Systems.*: SIAM, 2006.

[14] B. Hendrickson. (2014, Mar.) Chaco: Software for Partitioning Graphs. [Online]. http://www.sandia.gove/~bahendr/chaco.html