

Heat Conductive Analysis with Balancing Domain Decomposition Method

Shioya, Ryuji

Department of Intelligent Machinery and Systems, Faculty of Engineering, Kyushu University

Kanayama, Hiroshi

Department of Intelligent Machinery and Systems, Faculty of Engineering, Kyushu University

Ogino, Masao

Department of Intelligent Machinery and Systems, Faculty of Engineering, Kyushu University

Mukaddes, Abul Mukid Mohammad

Department of Intelligent Machinery and Systems, Faculty of Engineering, Kyushu University

<https://hdl.handle.net/2324/6079>

出版情報 : Theoretical and Applied Mechanics Japan. 52, pp.43-53, 2003-10. 創泉堂出版
バージョン :
権利関係 :

Heat Conductive Analysis with Balancing Domain Decomposition Method

Ryuji SHIOYA*, Hiroshi KANAYAMA*, A.M.M. MUKADDES**
and Masao OGINO**

**Faculty of Engineering, Kyushu University, Fukuoka*

***Graduate School of Engineering, Kyushu University, Fukuoka*

Balancing Domain Decomposition (BDD) proposed by J. Mandel is an effective preconditioning technique for reducing the number of iterations of iterative Domain Decomposition Method(DDM). Until now many researches have been done on the implementation of BDD in different areas including elasticity problems and semiconductor simulation. In this paper, we implement BDD in another area that is 3-dimensional(3-D) heat conductive analysis in solid. We construct the BDD preconditioner in parallel based on Hierarchical Domain Decomposition Method (HDDM). We report the comparative performance of BDD with diagonal scaling and original DDM without preconditioning to analyse heat conductive problems with about 2 million degrees of freedom and over 11 million degrees of freedom. With BDD the convergence rates are reduced effectively and become independent of the number of subdomains.

1 Introduction

In the area of engineering and science, we can really solve challenging linear algebraic problems by combining direct and iterative methods. Domain Decomposition Method (DDM) provides a rather natural way to combine those two approaches in finite element analysis of large scale problems. DDM proceeds by partitioning the whole domain (the problem) into subdomains (subproblems), Finite Element Analysis (FEA) is defined and solved on each subdomain in parallel and then partial solutions are glued together to get the global solution.

As the iterative DDM satisfies continuity among subdomains through iterative calculations like Conjugate Gradient (CG) method, it is absolutely necessary to reduce the number of iterations with a preconditioning technique especially for large problems. In our work, we choose Balancing Domain Decomposition(BDD) [1] as it is an effective preconditioning technique for reducing the number of iterations and speed-up of iterative DDM. BDD is close in spirit to multigrid methods [2] and is a variation of Neumann-Neumann algorithm. It involves solution of a coarse problem in each iteration of iterative DDM. Application of BDD is based on the selection of such coarse problem depending upon problems to be analysed. Many researches have been done on the implementation of BDD in different areas including elasticity problems [3] and semiconductor simulation [4].

This paper implements BDD in another area that is 3-dimensional(3-D) heat conductive analysis in solid by choosing a suitable coarse problem. In heat conductive analysis with either original DDM or

DDM with diagonal scaling, it has been found that the number of iterations increases with the number of subdomains. But with BDD the convergence rates are reduced and become independent of the number of subdomains. We construct this preconditioner in parallel based on Hierarchical Domain Decomposition Method (HDDM)[5]. To show the effect of BDD on heat conductive problems, we investigate two types of model. They are High Temperature Test Reactor (HTTR)[6] (2 million degrees of freedom) and Advanced Boiling Water Reactor (ABWR)[7] (over 11 million degrees of freedom).

The organisation of this paper is as follows. In section 2, the Schur complement method of DDM is described, while we present the balancing domain decomposition algorithm and its application in the next section. In section 4 we describe diagonal scaling. Some numerical results are reported in section 5.

2 Schur Complement of DDM

Consider a system of linear algebraic equations

$$Ku = f, \quad (1)$$

arising from a finite element discretization of a heat conductive problem in a domain Ω . The domain Ω is split into k non-overlapping subdomains $\Omega^{(1)}, \dots, \Omega^{(k)}$, each of which is the union of elements. In non-overlapping type of DDM, we consider two portions of each subdomain. One is the interior portion which has no node that is shared by other subdomains and another is the interface portion upon which all the nodes are shared by at least one other subdomain.

Let B be the set of all indices of the discretized nodes which belong to the interfaces between subdomains and I be the set of all indices of the nodes which belong to the interiors of subdomains. Grouping the points corresponding to B into the vector u_B and those corresponding to I into u_I , we get the reordered problem of (1):

$$\begin{pmatrix} K_{II} & K_{IB} \\ K_{IB}^T & K_{BB} \end{pmatrix} \begin{pmatrix} u_I \\ u_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix}. \quad (2)$$

Elimination of u_I in (2) leads to the reduced equation for u_B :

$$Su_B = g, \quad (3)$$

where

$$g = f_B - K_{IB}^T K_{II}^{-1} f_I \quad \text{and} \quad S = K_{BB} - K_{IB}^T K_{II}^{-1} K_{IB}. \quad (4)$$

S is usually referred to as *Schur complement matrix*. The matrix S inherits the symmetric positive definiteness property from K . Let $u^{(i)}$ be the vector of degrees of freedom corresponding to all elements in subdomain $\Omega^{(i)}$, and let $N^{(i)}$ denote the 0-1 matrix that maps the degrees of freedom $u^{(i)}$ into global degrees of freedom; then $u^{(i)} = N^{(i)T} u$ and, by the standard subassembly process,

$$K = \sum_{i=1}^k N^{(i)} K^{(i)} N^{(i)T}, \quad (5)$$

where $K^{(i)}$ is the local stiffness matrix corresponding to subdomain $\Omega^{(i)}$.

The subdomain stiffness matrix $K^{(i)}$, $u^{(i)}$ and the 0-1 matrix $N^{(i)}$ are split according to the equation (2).

$$K^{(i)} = \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{IB}^{(i)T} & K_{BB}^{(i)} \end{pmatrix}, \quad (6)$$

$$u^{(i)} = \begin{pmatrix} u_I^{(i)} \\ u_B^{(i)} \end{pmatrix}, \quad (7)$$

$$N^{(i)} = \left(N_I^{(i)}, N_B^{(i)} \right). \quad (8)$$

For a matrix K arising from a finite element discretization, the *Schur complement matrix* of (3) can also be written as

$$S = \sum_{i=1}^k N_B^{(i)} S^{(i)} N_B^{(i)T}, \quad (9)$$

where $S^{(i)}$ is referred to as the local Schur complement matrix associated with the subdomain $\Omega^{(i)}$. We assume that the subdomain matrix $K^{(i)}$ is symmetric and positive semidefinite with a nonsingular submatrix $K_{II}^{(i)}$. Then the local Schur complement $S^{(i)}$ is also positive semidefinite.

The reduced system (3) is posed in the space of the interface degrees of freedom, which we denote by V . Similarly, let $V^{(i)}$ be the space of the interface degrees of freedom for the subdomain $\Omega^{(i)}$. Then, interpreting matrix as mapping, we have

$$S: V \rightarrow V, \quad S^{(i)}: V^{(i)} \rightarrow V^{(i)}, \quad N_B^{(i)}: V^{(i)} \rightarrow V. \quad (10)$$

When implementation is considered on a distributed environment, the matrix S is usually not assembled [5]. We solve the Schur complement system by Preconditioned Conjugate Gradient(PCG) method which uses the BDD preconditioner.

3 Balancing Domain Decomposition

3.1 Algorithm

In this subsection, we present the BDD preconditioner for the Schur complement system. Let r be the residual of system (3) and M be the preconditioner, then in PCG method, one must compute $z = M^{-1}r$. In order to solve this problem, BDD solves a coarse problem in each iteration. An important design choice for the BDD preconditioner is the selection of a *weight matrix* $D^{(i)}$ and a *key matrix* $Z^{(i)}$ that are explained in the subsection 3.2 and 3.3, respectively. After selecting these matrixes, we are ready to define the action of $z = M^{-1}r$ in the BDD method.

Algorithm of BDD is summarized as follows :

Balance the original residual by solving the auxiliary problem for an unknown vector $(\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(k)})^T$

$$Z^{(i)T} D^{(i)T} N_B^{(i)T} \left(r - S \sum_{j=1}^k N_B^{(j)} D^{(j)} Z^{(j)} \lambda^{(j)} \right) = 0, \quad i = 1, \dots, k \quad (11)$$

and set

$$s = r - S \sum_{j=1}^k N_B^{(j)} D^{(j)} Z^{(j)} \lambda^{(j)}, \quad (12)$$

$$s^{(i)} = D^{(i)T} N_B^{(i)T} s, \quad i = 1, \dots, k. \quad (13)$$

Find any solution $u^{(i)}$ for each of the local problems

$$S^{(i)}u^{(i)} = s^{(i)}, \quad i = 1, \dots, k, \quad (14)$$

balance the residual by solving the auxiliary problem for $(\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(k)})^T$

$$Z^{(i)T} D^{(i)T} N_B^{(i)T} (r - S \sum_{j=1}^k N_B^{(j)} D^{(j)} (u^{(j)} + Z^{(j)} \mu^{(j)})) = 0, \quad i = 1, \dots, k, \quad (15)$$

and average the result on the interfaces according to

$$z = \sum_{i=1}^k N_B^{(i)} D^{(i)} (u^{(i)} + Z^{(i)} \mu^{(i)}). \quad (16)$$

The entries of the global coarse matrix in (11) are constructed from the local contributions by using the formula $Z^{(i)T} D^{(i)T} N_B^{(i)T} S N_B^{(j)} D^{(j)} Z^{(j)}$, where i and j are indexes of subdomains.

In this work, using the key matrix $Z^{(i)}$ defined in section 3.3, the solutions $(\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(k)})^T$ and $(\mu^{(1)}, \mu^{(2)}, \dots, \mu^{(k)})^T$ of coarse problems in (11) and (15) become vectors. In (14), $u^{(i)}$ ($i = 1, \dots, k$) are the solutions of the local problems that are solved completely in parallel (see (20)).

Let W be the subspace of V defined by

$$W = \{v \in V : v = \sum_{i=1}^k N_B^{(i)} D^{(i)} u^{(i)}, u^{(i)} \in \text{Range } Z^{(i)}\}, \quad (17)$$

P be the S -orthogonal projection onto W and

$$M_{NN} = \sum_{i=1}^k N_B^{(i)} D^{(i)} S^{(i)\dagger} D^{(i)T} N_B^{(i)T}, \quad (18)$$

where $S^{(i)\dagger}$ is the pseudoinverse of $S^{(i)}$. Then the BDD algorithm gives an explicit form of the preconditioner M as,

$$M^{-1} = ((I - P)M_{NN}S(I - P) + P)S^{-1}. \quad (19)$$

Finally, note that the local Schur complement matrix $S^{(i)}$ may be singular. To overcome this difficulty, we use a small regularization parameter α , replacing problems with $S^{(i)}$ by non-singular problems with

$$S^{(i)\dagger} = [S^{(i)} + \alpha \max\{\text{diag}(K_{BB}^{(i)})\}I]^{-1}, \quad (20)$$

where $\max\{\text{diag}(K_{BB}^{(i)})\}$ is the maximum value of diagonal elements of matrix $K_{BB}^{(i)}$ in (6) and I is the identity matrix. In our analysis, we use the value of α as 10^{-3} .

3.2 Definition of $D^{(i)}$

The BDD preconditioner uses a collection of matrixes $D^{(i)}$ that form a decomposition of unity on the space V ,

$$\sum_{i=1}^k N_B^{(i)} D^{(i)} N_B^{(i)T} = I. \quad (21)$$

In this study, we use $D^{(i)}$ as $n^{(i)} * n^{(i)}$ diagonal matrix where $n^{(i)}$ is the dimension of $V^{(i)}$. The diagonal elements are equal to the reciprocal of the number of subdomains with which the degrees of freedom is associated.

3.3 Definition of $Z^{(i)}$ for Heat Conductive Problems

Let $m^{(i)}$ be the number with $0 \leq m^{(i)} \leq n^{(i)}$ and $Z^{(i)}$ be the $n^{(i)} \times m^{(i)}$ matrix such that

$$\text{Null } S^{(i)} \subset \text{Range } Z^{(i)}, \quad i = 1, \dots, k. \quad (22)$$

From the property of $S^{(i)}$ and elementary arguments on the linear algebra, we can find that $x_B^{(i)} \in \text{Null } S^{(i)}$ for each $x^{(i)} \in \text{Null } K^{(i)}$. On the other hand, the heat equation with heat flux condition on the whole boundary is determined up to an additive constant.

Therefore, in heat conductive analysis, we use $Z^{(i)}$ as follows[1]:

$$Z^{(i)} = (1 \dots 1)^T \quad (23)$$

where the number of element "1" is for each interface point in subdomain i .

3.4 Parallel BDD based on HDDM sytem

We construct the BDD preconditioner in parallel based on HDDM system which classifies processors into 3 groups, Grand Parent, Parent and Child. The role of Grand Parent is to control all the Parent and Child processors. It also executes the balancing of residuals. The Parent processors send data to Child processors and store data received from Child processors. The preconditioner is made by Parent processors. The role of Child processors is to receive data from Parent processors, analyse data, send results and solve the local problems (14). We refer to [5] for details on HDDM system.

4 Diagonal Scaling

In order to compare BDD with another preconditioner, diagonal scaling is considered in this work. In diagonal scaling, the simplest choice is choosing a preconditioner as a diagonal matrix whose diagonal elements are corresponding ones of S in (3). But the global Schur complement matrix S is never assembled explicitly. So we choose a diagonal matrix as a preconditioner whose elements are the same as the corresponding ones of K_{BB} in (2). We denote this preconditioner as DIAG.

5 Numerical Results

The purpose of our computational tests is to demonstrate fast convergence of the BDD on complicated 3D problems. BDD is applied to the finite element analysis of two types of model. They are High Temperature Test Reactor (HTTR)[6] and Advanced Boiling Water Reactor (ABWR)[7]. These models are expressed by 10 node tetrahedral elements. We compare BDD with diagonal scaling (denoted by DIAG in Tables and Figures) and original DDM (denoted by DDM in Tables and Figures).

5.1 HTTR model

A High Temperature Test Reactor shown in Figure 1 is a graphite made, helium cooled reactor that can supply high temperature heat as high as 1000°C and has a potential of obtaining high thermal efficiency. The reactor core is designed to generate 30 MW.

As the boundary conditions for this model, we consider some high temperature on the six faces of lower plane and some low temperature on those of upper plane as shown in Figure 2. We also assume 1000°C in the lower cavity portion and 900°C in the upper cavity portion. In other parts, we use the natural boundary condition. In order to evaluate the performance of BDD, heat conductive analysis has been performed on this model which is divided into 2 parts, 3,000 subdomains, 1,167,268 elements and 1,893,340 nodes for HDDM. The problem of this model is solved by 10 Pentium 4 (2 GHz) processors.

The computational results (Table 1) of this model show that the number of iterations is reduced with BDD. A little time is consumed to make the preconditioner that is shown as preprocessing time (included in computational time) in Table 1. In case of DIAG and original DDM, the number of iterations increases with the number of subdomains while in case of BDD, it increases very little, that is shown in Table 2. The CG residual values are shown in Figure 3 in contrast to the number of CG iterations. BDD converges rapidly than DIAG and DDM.

5.2 ABWR Model

In this research, we investigate the temperature distribution through ABWR model to test the efficiency of the BDD method. As boundary conditions for this model (Figure 4), we consider some temperature values in the lower circular portion and some flux values on the side surface as shown in Figure 4(right). In other portions, the natural boundary condition is considered. This model is divided into 16 parts, 1,200 subdomains per part, 7,486,792 elements and 11,794,506 nodes. The problem of this model is solved by 128 processors of HITACHI SR8000 (250 MHz) machine.

The computational results of this model are given in Table 3, while the CG history is shown in Figure 5. With BDD the number of iterations is reduced to 1/62 of original DDM and computational time is reduced to 1/6. Figure 6 shows the temperature distribution of ABWR model. The deep color represents high temperature region and the light color represents low temperature region.

6 Conclusions

The domain decomposition method with BDD preconditioner for thermal analysis has been developed in the current study. We have constructed the BDD preconditioner in parallel. We have shown that BDD converges faster but it needs more memory. This system can be applied to any 3-D heat conductive analysis in solid and effective performance can be obtained. The future work relating to this study is to apply BDD to larger models like over 35 million degrees of freedom for thermal analysis.

References

- [1] Mandel, J., “Balancing domain decomposition”, *Communication on Numerical Methods in Engineering*, 9 (1993), pp.233-241.
- [2] Axelsson, O. and Gustafsson I., “Preconditioning and two-level multigrid methods of arbitrary degree of approximation”, *Math. Comp.*, 40 (1993), pp.219-242.
- [3] Shioya, R., Ogino, M., Kanayama, H. and Tagami, D., “Large scale finite element analysis with a balancing domain decomposition method”, *Key Engineering Materials*, Vols.243-244 (2003), pp.21-26.
- [4] Giraud, L., Kostar, J., Marrocco, A. and Rioual, J.-C., “Domain Decomposition Method in Semiconductor Device Modeling”, Technical Report(2001), TR/PA/01/51.
- [5] Yagawa, G. and Shioya, R., “ Parallel finite elements on a massively parallel computer with domain decomposition”, *Computing Systems in Engineering* (1993), 4:4-6, pp.495-503.
- [6] Shioya, R., Kanayama, H., Tagami, D. and Imamura, E., “A domain decomposition approach for non-steady heat conductive analysis”, *Advances in Computational Engineering and Science* (2001), 189.pdf, pp.1-6.
- [7] Yoshimura, S., Shioya, R., Noguchi, H. and Miyamura, T., “Advanced general-purpose computational mechanics system for large scale analysis and design”, *Journal of Computational and Applied Mathematics* 149 (2002), pp.279-296.

Table 1: Comparison of DDM, DIAG and BDD for HTTR

	DDM	DIAG	BDD
Iteration counts	238	113	23
Comput. time [sec]	843	428	425
Time/iteration [sec]	3.30	3.30	13.25
Average memory [MB]	160	160	343
Preprocessing time [sec]	-	-	61

Table 2: Number of Iterations for Different Number of Subdomains (HTTR)

Subdomains	DDM	DIAG	BDD
1,400	193	100	22
1,600	200	104	23
2,000	214	104	22
2,400	221	112	23
3,000	238	113	23

Table 3: Comparison of DDM , DIAG and BDD for ABWR

	DDM	DIAG	BDD
Iteration counts	5,652	1,215	90
Comput. time [min]	312	72.38	46.7
Time/iteration [sec]	3.24	3.22	14
Average memory [MB]	134	140	550
Preprocessing time [min]	-	-	18.1

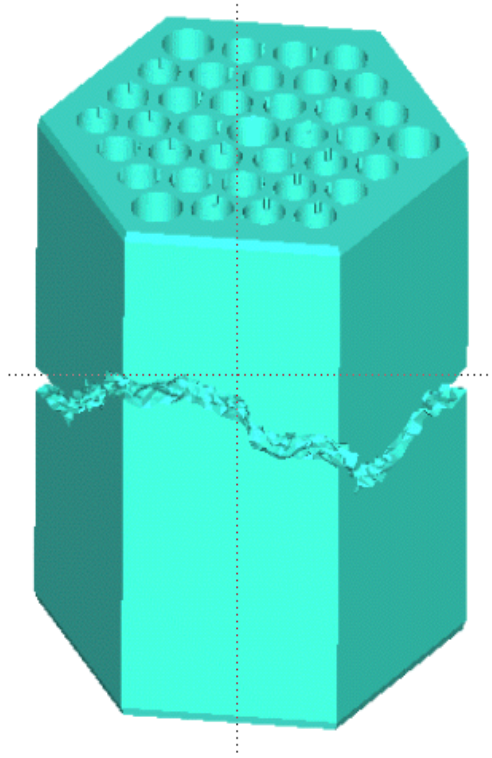


Figure 1: HTTR Model

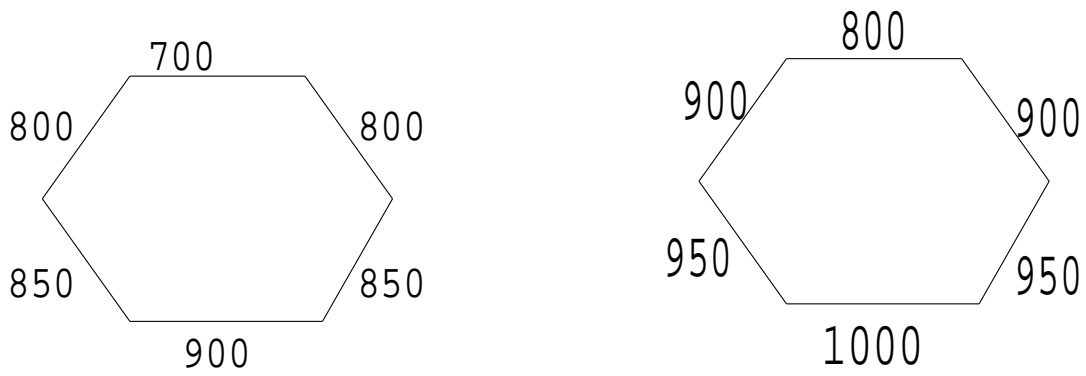


Figure 2: Temperature(centigrade) Boundary Conditions of Upper Plane(left) and Lower Plane(right) of HTTR

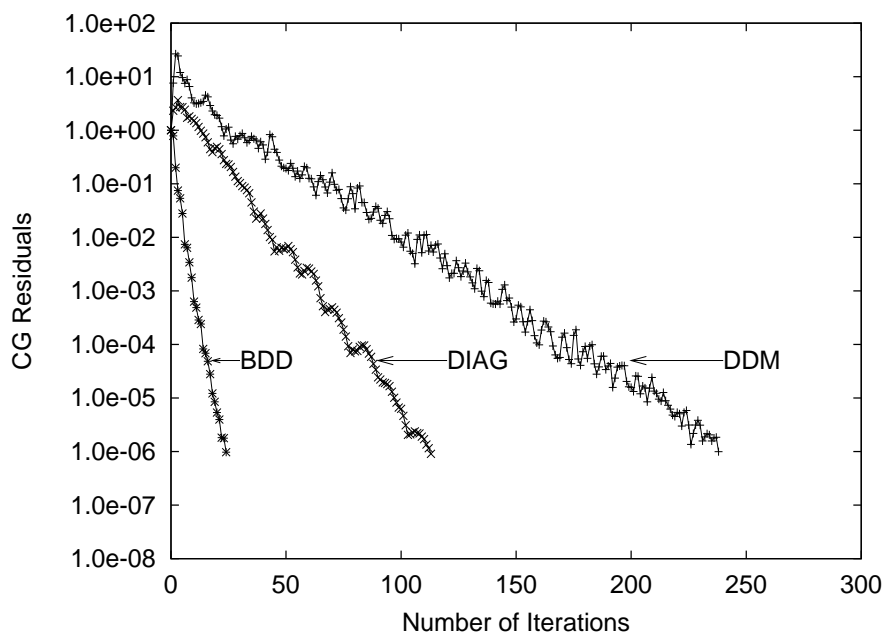


Figure 3: Residual History of CG (HTTR)

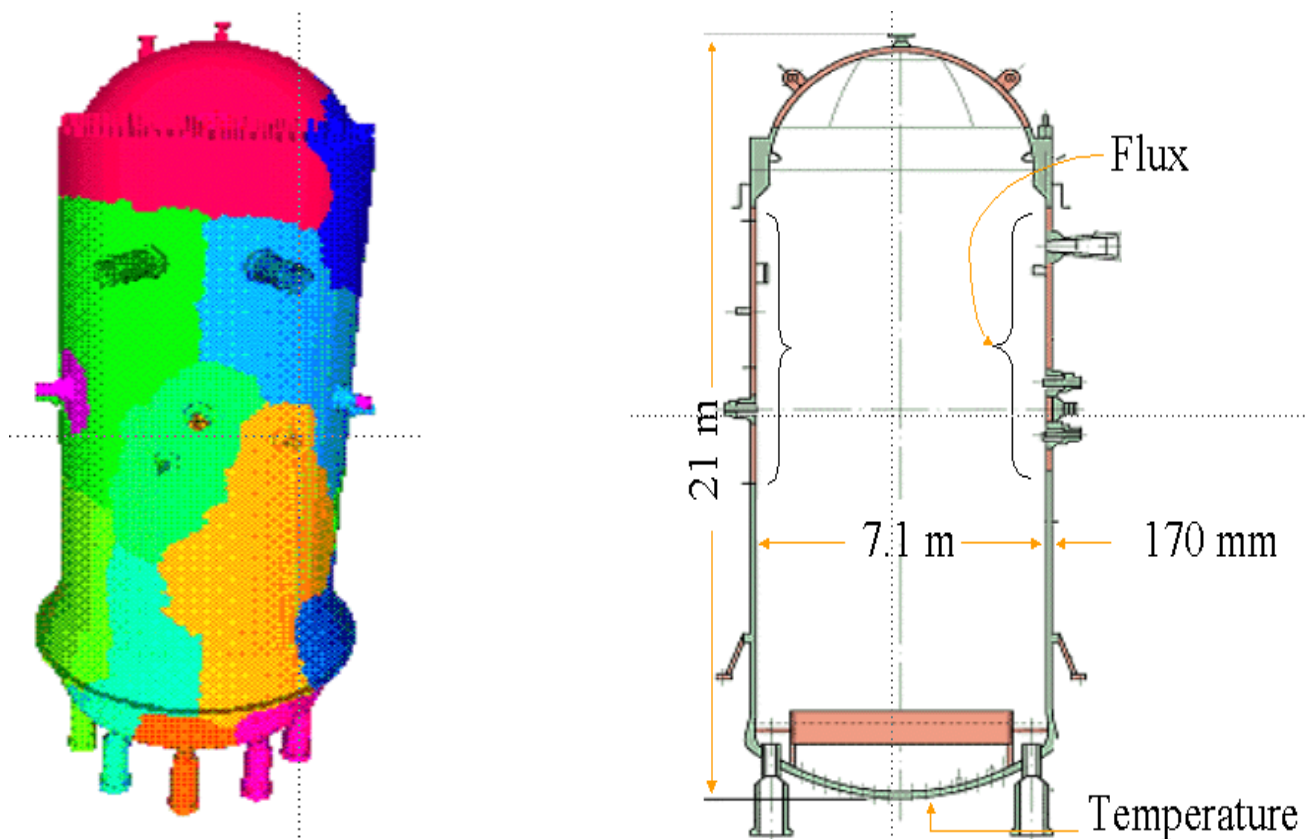


Figure 4: ABWR Model(left) and Boundary Conditions(right)

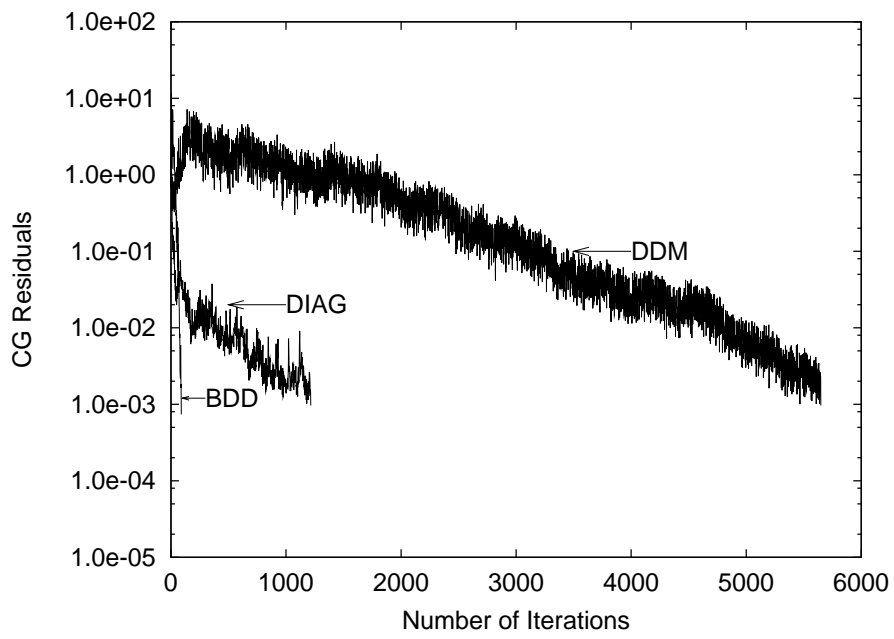


Figure 5: Residual History of CG (ABWR)

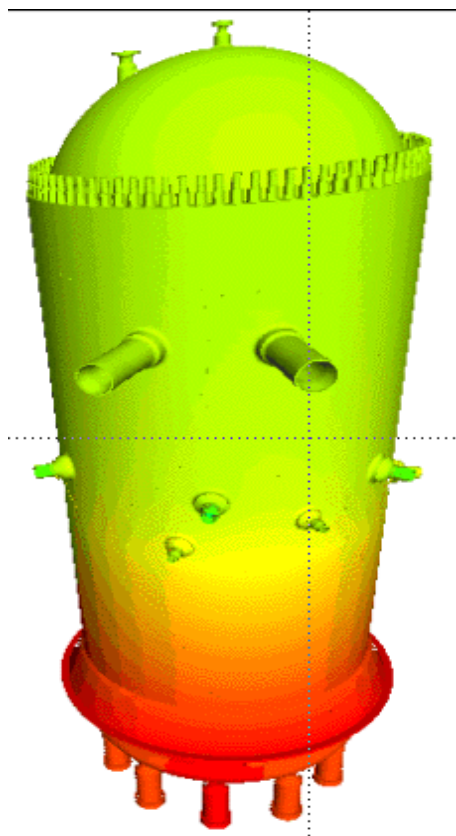


Figure 6: Temperature Distribution of ABWR