

LARGE SCALE THERMAL-SOLID COUPLING ANALYSIS USING INEXACT BALANCING DOMAIN DECOMPOSITION

Abul Mukid Mohammad Mukaddes

*Department of Industrial and Production Engineering
Shah Jalal University of Science and Technology, Bangladesh
email: mukaddes1975@gmail.com*

and

Ryuji Shioya

Faculty of Information Science and Arts, Toyo University, Japan

ABSTRACT

In this research, a system of thermal-solid coupling analysis is developed with the implementation of Inexact Balancing Domain Decomposition with a diagonal scaling (IBDD-DIAG) in both thermal and solid analysis. The IBDD-DIAG is an improved version of Balancing Domain Decomposition (BDD), where an incomplete factorization based parallel direct method is employed to solve a coarse space problem, and the diagonal-scaling is employed to precondition local fine space problems instead of the Neumann-Neumann preconditioner. The developed system performed heat conductive analysis to have temperature distributions in solid models and then performed the structural analysis to see deformation or expansion due to temperature differences. Both of the analyses employed the Hierarchical Domain Decomposition Method (HDDM) with parallel IBDD-DIAG. It is shown that the iterative procedure converges rapidly and the convergence is independent of the number of subdomains, namely, numerical scalability is satisfied. The present system is implemented on massively parallel processors and succeeds in solving a thermal-solid coupling problem of 12 millions of nodes.

Keywords: Finite element analysis, thermal-solid coupling problems, Domain Decomposition Method, Incomplete Balancing Domain Decomposition

1. INTRODUCTION

Engineering products that have complex geometries may crack during the heating process or fail during operations. To avoid such failures and reduce the cost of production, the behavior that occurs in the products during manufacturing or while operating must be predicted in advance. These behaviors include the stress from external loads as well as the thermal stresses from the temperature difference in the products. The purpose of this research is to develop a system that could be used to analyze heat transfer problems that have complex geometries for the temperature distribution. The predicted temperature combined with the applied external loads is then used to compute the deformation and thermal stresses of the products. The partial differential equation described the thermal and structural problems can be coupled when the stress in the structural problem is a function of temperature from the thermal problem [1]. The

motivation of the coupling analysis comes from thermal-fluid coupling problem [1] and the large scale fluid-structure coupling problem [2].

Large scale problems need to be solved for the improvement of accuracy. The conventional algorithm like Domain Decomposition Method (DDM) needs much time to solve the large scale problems and moreover it is not scalable [3]. A preconditioner should be used to reduce the computation time and the number of iterations. A suitable preconditioner might make the DDM scalable. The present research attempts to develop a system of thermal-solid coupling analysis to address these requirements.

By using the Hierarchical Domain Decomposition Method (HDDM) [4] with a preconditioned iterative solver in order to perform both thermal and solid analyses, it is hoped that the developed system would perform the heat conductive analysis to

achieve temperature distributions in solid models and then perform the structural analysis to see deformation or expansion due to temperature differences. The HDDM employs a preconditioned iterative solver in order to perform both thermal and solid analyses.

Moreover Balancing Domain Decomposition (BDD) [5] has received much attention in the last few years. The main reason for the popularity of this method is undoubtedly, the need to take the advantage of parallel computers. BDD is close in spirit to multigrid methods and is a variation of Neumann-Neumann algorithm. It involves solution of a coarse problem in each iteration of iterative DDM. Moreover to efficiently solve a coarse space problem derived from equilibrium conditions for singular problems associated with a number of subdomains appeared in the BDD algorithm, an Inexact Balancing Domain Decomposition with Diagonal Scaling (IBDD-DIAG) is proposed for structural problem [6].

In this paper, this inexact balancing domain decomposition method is investigated in the analysis of thermal-solid coupling problem. In the IBDD-DIAG formulation, a coarse space problem is approximated by an incomplete factorization coarse operator based parallel direct method, and the diagonal scaling is employed to precondition local fine space problems instead of the Neumann-Neumann preconditioner. The thermal-solid coupling problem of a nuclear pressure vessel model with 12 million nodes is successfully analyzed with this IBDD-DIAG. The numerical results show better performance of IBDD-DIAG.

2. THERMAL-SOLID ANALYSIS

By considering a heat conduction equation on a domain Ω , defining \bar{f} as internal heat generation, \bar{T} temperature applied on the boundary Γ_T , \bar{Q} the heat flux applied on the boundary Γ_C , the fundamental equations of this heat conduction problem is given by:

$$\begin{cases} q = -\lambda \text{grad} T & \text{in } \Omega \\ \text{div } q = \bar{f} & \text{in } \Omega \\ q \cdot n = \bar{Q} & \text{on } \Gamma_C \\ T = \bar{T} & \text{on } \Gamma_T \end{cases} \quad (1)$$

where T is temperature, q the heat flux, λ the thermal conductivity and n an outer normal unit vector, respectively. The finite element (quadratic tetrahedral) discretization of (1) yielded a linear system of the for

$$Ax = f \quad (2)$$

where A is the global stiffness matrix, x is an unknown vector of temperature and f is a known vector.

Again we considered a structural problem concerning a domain Ω . Hence, \bar{F}_i is the traction force applied on the boundary Γ_F , \bar{B}_i the body force applied in the domain Ω and \bar{u} the prescribed displacement on the boundary Γ_u . Fundamental equations of this structural problem are summarized as follows:

$$\begin{aligned} \tau_{ij,j} + \bar{B}_i &= 0 & \text{in } \Omega \\ \varepsilon_{ij} &= (u_{i,j} + u_{j,i})/2 & \text{in } \Omega \\ \tau_{ij} &= D_{ijmn} \varepsilon_{mn}^e & \text{in } \Omega \\ \tau_{ij} n_j - \bar{F}_i &= 0 & \text{on } \Gamma_F \\ u_i &= \bar{u}_i & \text{on } \Gamma_u \end{aligned} \quad (3)$$

where, i, j take the value 1 to 3, m, n take the value 1 to 3, u_i is displacement, ε_i a strain tensor, τ_i stress tensor, D_{ijmn} a coefficient tensor of the Hooke's law and n_j an outer normal vector on the boundary Γ , respectively. Here, if we consider the high temperature distribution on the whole domain Ω , elastic strain as ε^e , total strain ε and thermal strain ε^t then we have:

$$\varepsilon^e = \varepsilon - \varepsilon^t \quad (4)$$

Again, the thermal ε^t is considered as follows:

$$\varepsilon^t = \begin{cases} \alpha(T - T_0) & (m = n) \\ 0 & (m \neq n) \end{cases} \quad (5)$$

where T_C is the reference temperature, α the thermal expansion coefficient and T the temperature which is the output of the thermal

analysis. The value of thermal strain $\boldsymbol{\varepsilon}^t$ in the equation (5) was used in the equation (3) through the equation (4). The thermal strain $\boldsymbol{\varepsilon}^t$ took the value $\alpha(T - T_0)$ when $m=n$ and otherwise it took 0. The finite element (quadratic tetrahedral) discretization of (3) yielded a linear system of the form:

$$Ku = b \quad (6)$$

where K , u and b are, respectively, the stiffness matrix, the displacement vector and the force vector.

3. THERMAL-SOLID COUPLING ANALYSIS

The present system is conducted to predict temperature distributions in solid models and then to investigate the thermal expansion or deformation due to the temperature difference. Analysis steps are as follows:

- 1) Read the input data for the heat conductive analysis and decompose the model by ADVENTURE_Metis [6].
- 2) Analyze heat conductive problems with IBDD-DIAG based on the HDDM system
- 3) Gather temperature information of all nodes of the model from outputs of heat conductive problems.
- 4) Read temperature information of all nodes and other input data for structural analysis and then decompose the model by ADVENTURE_Metis.
- 5) Analyze structural problems with IBDD-DIAG based on the HDDM system.

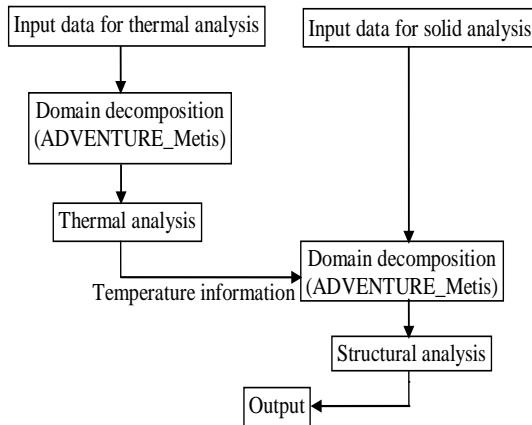


Fig-1. Flow chart of thermal-solid coupling problems

Fig. 1 shows the flow chart of thermal-solid coupling analysis with the developed system. The name of the ADVENTURE module used in each analysis is shown in parentheses.

4. DOMAIN DECOMPOSITION METHOD

In this method the domain Ω was decomposed into N non-overlapping subdomains, $\{\Omega_i\}_{i=1,\dots,N}$. As usual the stiffness matrix K (it represents A in equation 2 and K in equation 6) could be generated by subassembly:

$$K = \sum_{i=1}^N R^{(i)} K^{(i)} R^{(i)T} \quad (7)$$

where $R^{(i)T}$ is the 0-1 matrix which translates the global indices of the nodes into local numbering. Denoting $u^{(i)}$ as the vector corresponding to the elements in $\Omega^{(i)}$, it can be expressed as $u^{(i)} = R^{(i)T} u$. Each $u^{(i)}$ was split into degrees of freedom $u_B^{(i)}$, which correspond to $\partial\Omega^{(i)}$, called interface degrees of freedom and remaining interior degrees of freedom $u_I^{(i)}$. The subdomain matrix $K^{(i)}$, vector $u^{(i)}$ and 0-1 matrixes were then split accordingly

$$K^{(i)} = \begin{pmatrix} K_{II}^{(i)} & K_{IB}^{(i)} \\ K_{IB}^{(i)T} & K_{BB}^{(i)} \end{pmatrix}, \quad (8)$$

$$u^{(i)} = \begin{pmatrix} u_I^{(i)} \\ u_B^{(i)} \end{pmatrix}, \quad (9)$$

$$\text{and } R^{(i)} = \begin{pmatrix} R_I^{(i)} & R_B^{(i)} \end{pmatrix}. \quad (10)$$

After eliminating the interior degrees of freedom, problem (2) was reduced to a problem on interface

$$Su_B = g \quad (11)$$

where $S = \sum_{i=1}^N R_B^{(i)} S^{(i)} R_B^{(i)T}$ is assumed to be positive definite, u_B is the vector of the unknown variables on the interface, g is a known vector and $S^{(i)}$ are the local Schur complements of subdomain $i=1,\dots,N$, assumed to be positive semi-definite. The problem (11) is solved by a preconditioned CG method which solves the problem,

$$z = M^{-1} r \quad (12)$$

where r is the residual of (11) and M is a preconditioner. When the interface problem is solved iteratively, of course, an efficient solution of

the large scale problems depends on how one chooses an efficient and scalable preconditioner.

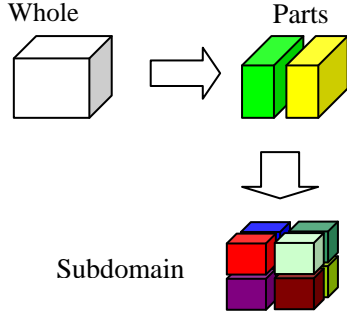


Fig. 2 Hierarchical Domain Decomposition Method

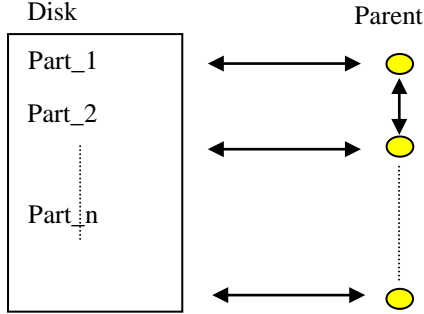


Fig.3 Parallel processor scheme

ii) Hierarchical Domain Decomposition Method (HDDM)

Constructing the DDM algorithms for parallel computers, a good principle is to divide the original domain into parts, which are further decomposed into smaller subdomains as shown in fig.-2. In this research, Hierarchical Domain Decomposition Method (HDDM) [4] which is a well known parallel DDM is adopted.

Fig.3 shows the parallel processor scheme of HDDM system. In this scheme, the Parent processors perform the FEA by themselves and also coordinate the CG iteration. Here, the number of parts should be the number of processors.

5. BALANCING DOMAIN DECOMPOSITION (BDD)

The BDD preconditioning technique proposed by Mandel [5] uses at each CG iteration solution of the local Neumann-Neumann problems on the subdomains coupled with a coarse problem in a

coarse space. The BDD preconditioner is of the form:

$$M_{BDD}^{-1} = Q_c + (I - Q_c S) Q_l (I - S Q_c) \quad (13)$$

where Q_l is the local level part and Q_c is the coarse level part of the preconditioner.

5.1 Local level

The local level part of the preconditioner basically involved the solution of local problems, where Q_l is expressed by

$$Q_l = \sum_{i=1}^N R_B^{(i)} D^{(i)} S^{(i)+} D^{(i)T} R_B^{(i)T} \quad (14)$$

The dagger (+) indicates pseudo-inverse, since the $S^{(i)}$ is singular for floating subdomain. The BDD method uses a collection matrixes $D^{(i)}$ that determines partition of unity on interface [5,9],

$$\sum_{i=1}^N R_B^{(i)} D^{(i)} R_B^{(i)T} = I \quad (15)$$

The simplest choice for $D^{(i)}$ is the diagonal matrix with diagonal elements equal to the reciprocal of the number of subdomains with which the degree of freedom is associated.

5.2 Coarse level

The application of the coarse term $Q_c = R_0 (R_0^T S R_0)^{-1} R_0^T$ amounts to the solution of a coarse problem whose coefficient matrix is $S_W = R_0^T S R_0$. The operator R_0 translates the coarse degrees of freedom to the corresponding global degrees of freedom and is defined by

$$R_0 = [R_B^{(1)} D^{(1)} Z^{(1)}, \dots, R_B^{(N)} D^{(N)} Z^{(N)}] \quad (16)$$

For the scalar heat conductive problem, $Z^{(i)}$ is a column constant vector [8,9] and can be defined by

$$Z^{(i)} = (1, \dots, 1)^T \quad (17)$$

For the structural problem $Z^{(i)}$ [8] comes from the dof of rigid body of motion.

5.3 Simplified diagonal scaling (DIAG)

A diagonal matrix is considered as a preconditioner whose diagonal elements are constructed from the corresponding ones of $K_{BB}^{(i)}$. The diagonal matrix is defined as

$$Q_{DIAG} = \sum_{i=1}^N R_B^{(i)} (diag(K_{BB}^{(i)}))^{-1} R_B^{(i)T} \quad (18)$$

6. INEXACT BALANCING DOMAIN DECOMPOSITION

To implement the coarse grid correction with high parallel efficiency, inexact balancing based on an incomplete parallel Cholesky factorization is employed in the present BDD method. In general, such an incomplete factorized operator is typically used together with some iterative computations to compensate the incompleteness. In this paper, however, the coarse problem is approximated by the incomplete factorized operator without iterations. This incomplete balancing process decreases computation costs for preprocessing and improves parallel efficiency but may reduce the convergence rate compared with exact balancing. However, total computation time is expected to be reduced. Remarkably, with the original exact BDD preconditioner, a coarse grid correction is performed after a local subdomain correction in each iteration. However, with the present inexact BDD preconditioner, a coarse grid correction is also implemented to the CG residual vector before a local subdomain correction due to incomplete deletion of components of the coarse space. The incomplete balancing technique is applied to the BDD considering a diagonal scaling as a local level preconditioner. The new preconditioning technique is marked as IBDD-DIAG and is defined as:

$$M_{IBDD-DIAG}^{-1} = \tilde{Q}_c + (I - \tilde{Q}_c S) Q_{DIAG} (I - S \tilde{Q}_c) \quad (19)$$

where \tilde{Q}_c is constructed from the incomplete factorized coarse operator.

The implementation of the IBDD-DIAG preconditioner (19) goes as follows:

Step 1: Balance the original residual by approximating the coarse problem using the incomplete coarse operator for an unknown vector $\tilde{\lambda} \in \mathfrak{R}^N$:

$$\tilde{S}_0 \tilde{\lambda} = R_0^T r \quad (20)$$

Step 2: Set

$$\tilde{s} = r - SR_0 \tilde{\lambda} \quad (21)$$

Step 3: Perform the diagonal scaling and average these results

$$\tilde{u} = \sum_{i=1}^N R_B^{(i)} \left(\text{diag} \left(K_{BB}^{(i)} \right) \right)^{-1} R_B^{(i)T} \tilde{s} \quad (22)$$

Step 4: Compute

$$\hat{s} = r - S \tilde{u} \quad (23)$$

Step 5: Approximate the coarse problem again for an unknown vector $\tilde{\mu} \in \mathfrak{R}^N$

$$\tilde{S}_0 \tilde{\mu} = R_0^T \hat{s} \quad (24)$$

Step 6: Find the preconditioned vector

$$z = \tilde{u} + R_0 \tilde{\mu} \quad (25)$$

The \tilde{S}_c means the corresponding term of the coarse matrix of $R_0^T SR_0$, which is factorized incompletely. Hence, it is said that the residual is incompletely balanced in (21). The implementation of incomplete balancing reduces the computation costs for factorization of the coarse matrix and for forward elimination and backward substitution of the problem (20) and (24) and consequently the amount of work of each iteration is reduced. For this reason although IBDD-DIAG preconditioner may increase the number of iterations, a speed up is achieved for large scale problems in the massively parallel computer.

7. NUMERICAL RESULTS AND DISCUSSIONS

Thermal and structural analysis on two models shown in fig. 4 are performed.

7.1 HTGR Model [8,9]

i) Model description and computational conditions

The HTGR model shown in Fig. 4 is graphite made, helium cooled reactor core whose height was 580 mm. The convergence criterion was that the norm of the relative residual is reduced to 10⁻⁶. The mesh sizes of this model are shown in Table 1.

ii) Computational performances

The computation performances for both thermal and structural analysis of HTGR model are shown in Table 2. Fig. 5 shows the convergence history for thermal analysis of HTGR model. Both results show that BDD and IBDD-DIAG converges rapidly. Regarding the memory requirements of the various preconditioning approaches, IBDD-DIAG which employ a diagonal-scaling as a local subdomain correction, reduce memory requirements by around 40% compared with BDD preconditioner.

7.2 Model B: ABWR [9]

i) Model description and computational conditions

In this research, as a large scale and real shape model problem having a bad convergence, the

present method is applied to a 12 million node unstructured mesh for a precise model of Advanced Boiling Water Reactor (ABWR)[9] as shown in Fig. 4 . The model is expressed with 3,000 subdomains, 7,486,792 elements, 11,794,506 nodes in the HDDM system.

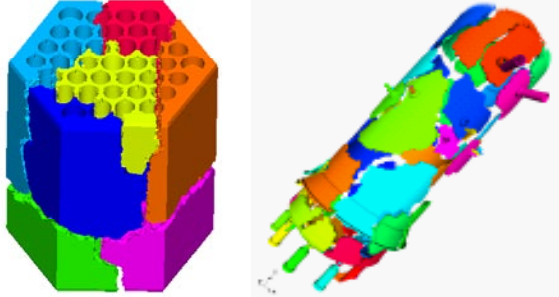


Fig. 4 Part decomposition of HTTR (left) and ABWR model (right)

Table 1: Mesh sizes of HTGR model

| | Number of nodes (dof) | Number of elements | Number of subdomains |
|------------|-----------------------|--------------------|----------------------|
| Thermal | 1,893,340 (1,893,340) | 1,167,268 | 3,200 |
| Structural | 1,893,340 (5,680,020) | 1,167,268 | 4,000 |

Table 2: Performances of BDD and IBDD-DIAG (HTGR)

| | | Num. of iteration | Compu-tation time (sec) | Memory/ PE (MB) |
|---------------------|-----------|-------------------|-------------------------|-----------------|
| Thermal Analysis | DIAG | 527 | 777.50 | 174 |
| | BDD | 33 | 224.61 | 277 |
| | IBDD-DIAG | 70 | 358.34 | 185 |
| Structural Analysis | DIAG | 1,570 | 6,895 | 321 |
| | BDD | 118 | 3,075 | 544 |
| | IBDD-DIAG | 85 | 2,599 | 980 |

Table 3 Performances of BDD and IBDD-DIAG (ABWR) for thermal analysis

| | Sub # 3,000 | | | Sub #6,000 | | |
|-----------|-------------|------------|-----------|------------|------------|-----------|
| | #Iter. | Time (sec) | Mem. (MB) | Iter. | Time (sec) | Mem. (MB) |
| DIAG | 3,201 | 1,259 | | 4,295 | 1,103 | |
| BDD | 59 | 522 | | 58 | 1,758 | |
| IBDD-DIAG | 124 | 264 | | 105 | 294 | |

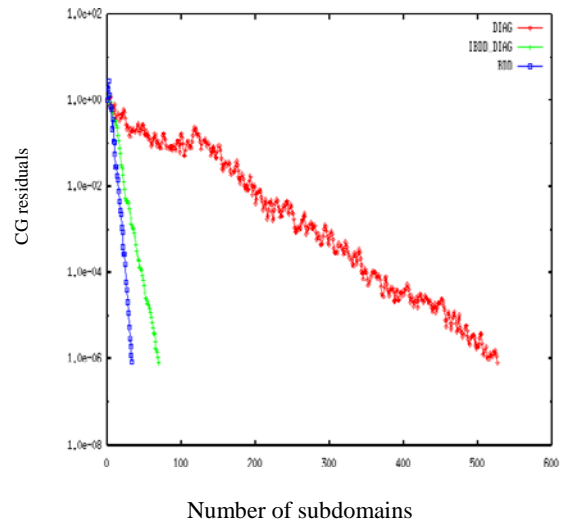


Fig. 5 Convergence history of thermal analysis (HTGR)

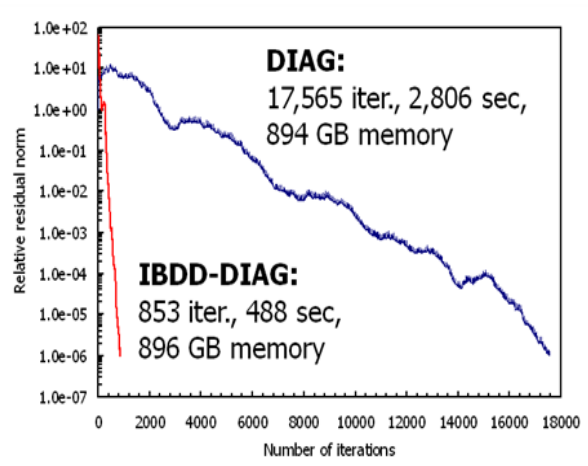


Fig. 6 Convergence history (ABWR model)

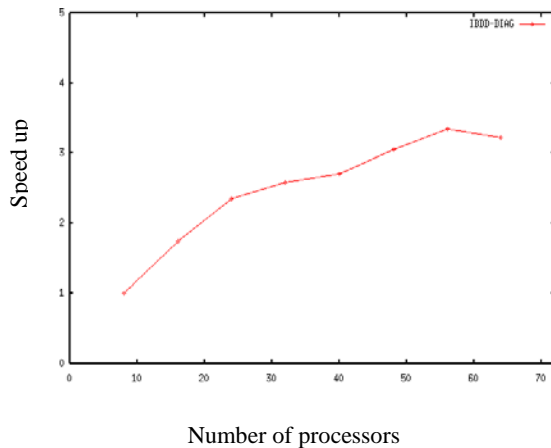


Fig. 7 Parallel scalability (HTGR Model)

ii) Computational performances

IBDD-DIAG is investigated in the thermal analysis of ABWR model using two different numbers of subdomains. The results are shown in Table 3. In 3,000 subdomains, the exact BDD type preconditioners reduce the number of iterations to about 2% and the computation time to about 40%. In 3,000 subdomains, the IBDD type preconditioners reduce the number of iteration to about 4% and the computational time to about 22%. It is found that with almost the same memory size as DIAG, IBDD-DIAG shows the best performance in computation time. Next in 6,000 subdomains, the exact BDD type preconditioners are slower than DIAG. It has the reason that the completely Cholesky factorization of a coarse grid operator gives almost all computation time. Here, IBDD-DIAG shows good performance in computational time with less memory size than in the case of 3,000 subdomains. Therefore, the IBDD-DIAG is an effective method to analyze large scale thermal-solid coupling problems. The convergence history for thermal analysis of ABWR model is shown in fig. 5 which shows the better convergence of IBDD-DIAG. Again fig. 6 predicts that IBDD-DIAG is parallelly scalable [10].

8. CONCLUSION

In this study a thermal-solid coupling system is successfully implemented on a 12 million dof ABWR model. The computational speed of the HDDM system was improved dramatically by employing the IBDD-DIAG method as an efficient

preconditioner. The IBDD-DIAG exhibits an excellent performance in terms of memory requirements, convergence rate and computation time. Furthermore, this system has been successfully implemented on parallel computer.

References

1. Reddy, J. N. and Gartling, D. K. 2000. *The finite element method in heat transfer and fluid dynamics*, CRC press, USA.
2. Gang, Li., Wenbin, C. and Yaochu, F. 2008. *A fluid-structure coupling analysis of large scale hyperbolic cooling tower subjected to wind loads*. Proc. of 8th World Congress on Computational Mechanics (WCCM'8).
3. Farhat, C., Chen, P.S. and Mandel, J., 1995, A Scalable Lagrange Multiplier Based Domain Decomposition Method for Time-Dependent Problems, *International Journal for Numerical Methods in Engineering*, Vol. 38 , p. 3831-3853.
4. Yagawa, G. and Shioya, R. 1993. Parallel finite elements on a massively parallel computer with domain decomposition. *Comput. Systems Eng.* 4:495-503.
5. Mandel, J. 1993. Balancing domain decomposition. *Comm. Numer. Methods Eng.* 9:223-241.
6. M. Ogino, R. Shioya and H. Kanayama. 2008, An inexact balancing preconditioner for large scale structural analysis, Vol.2. No. 1
7. <http://adventure.q.t.u-tokyo.ac.jp>
8. Shioya, R., Ogino, M., Kanayama, H. and Tagami, D. 2003. Large scale finite element analysis with a balancing domain decomposition method. *Key Eng. Materials* 243-244:21-26.
9. Shioya, R., Kanayama, H., Mukaddes, A.M.M. and Ogino, M. 2003, Heat conductive analysis with balancing domain decomposition. *Theor. Appl. Mech.* 52:43-53.
10. Mukaddes, A.M.M., Ogino, M., Kanayama, H. and Shioya, R. 2006. A scalable Balancing Decomposition based preconditioner for large scale heat transfer problems. *JSME Int. J. Ser. B.* 49:533-540.