PARALLEL FINITE ELEMENT ANALYSIS OF A LARGE SCALE NON-STEADY HEAT CONDUCTIVE PROBLEM

*A.M.M. Mukaddes, **Ryuji Shioya and **Masao Ogino

*Department of IPE, Shah Jalal University of Science and Technology, Bangladesh, **Kyushu University, Japan
mukaddes-ipe@sust.edu

ABSTRACT

This paper describes the parallel finite element analysis of large scale problem for a non-steady heat conductive problem using domain decomposition method. In order to solve the issue of scalability of a domain decomposition method, the developed system employs the balancing domain decomposition method and hierarchical data management technique among the parallel processors. The present system is successfully applied to the FEM analysis of over two million degrees of freedom on a massively parallel computer, SR8000.

Keywords: Balancing Domain Decomposition, Parallel Computing, Finite Element Method, Heat Conductive Problems.

1. INTRODUCTION

In order to solve large scale problems, using finite element method, more processing power and memory of computer are required. Using single processor computer, we encounter their physical limits. To save the computation time and memory, it is well known that, the parallel computers particularly Multiple Instruction Multiple Data (MIMD) including clustered workstation computer seems promising. A MIMD type computer has many processors with local memory, and can reduce computation time by distributing tasks among processors. However, we need special algorithm for parallel computing to solve problems with high performance using this kind of computer.

As a parallel numerical algorithm for the finite element analysis, domain decomposition method has become increasingly popular for the solution of large scale linear systems arising from 3-dimensional problems of different engineering fields. This paper develops the domain decomposition method (DDM) combined with a suitable preconditioning technique for the analysis of 3-dimensional heat conductive problems in parallel processors. In this method the whole domain to be analyzed is fictitiously divided into a number of subdomains without overlapping. Finite element analysis is performed in each subdomain while an iterative scheme is applied to the solution of the interface problem.

Again, when the interface problem is solved iteratively, usually via a preconditioned Conjugate Gradient (CG) algorithm, the overall domain decomposition method becomes a genuine iterative solver whose success hinges on two important properties: numerical scalability and parallel scalability. A domain decomposition method based on an iterative scheme is said to be numerically scalable if after preconditioning the number of iterations does not grow with the number of subdomains[1].
As an efficient and scalable domain decomposition preconditioner for the solution of algebraic systems from the approximation of a partial differential equation problem, BDD has received much attention in the last few years. This method is introduced by Jan Mandel [2]. Domain decomposition method combined with Mandel’s BDD algorithm is now exclusively employed for the solution of huge structural problems[3] and steady state heat conductive problems[1,4]. In this paper, present authors develop a domain decomposition approach based on the balancing domain decomposition method for the non-steady heat conductive problems.

In the present study, the above method have been implemented in parallel codes that have been successfully tested in massively parallel computer, SR8000 to solve a large scale non-steady heat conductive problem which model is a reactor core component. Numerical results show that the convergence of the developed method is almost independent of the number of subdomains.

2. PROBLEMS AND DISCRETIZATIONS

We consider a heat conductive problem in a domain $\Omega$. Defining $\bar{f}$ as internal heat generation, $\bar{u}$ temperature applied on the boundary $\Gamma_u$, $\bar{Q}$ the heat flux applied on the boundary $\Gamma_Q$ and $\alpha_c (u - u_e)$ convection heat transfer on the boundary $\Gamma_c$, the fundamental equations of this heat conductive problem are given by:

\[
\begin{align*}
q &= -k \text{grad} u \quad \text{in } \Omega \\
\rho \alpha_c \frac{\partial u}{\partial t} &= \text{div} q + \bar{f} \quad \text{in } \Omega \\
q \cdot n &= \bar{Q} \quad \text{on } \Gamma_Q \\
q \cdot n &= \alpha_c (u - u_e) \quad \text{on } \Gamma_c \\
u &= u \quad \text{on } \Gamma_u
\end{align*}
\]

(1)

where $u$ is temperature, $\frac{\partial u}{\partial t}$ a time derivative, $q$ the heat flux, $k$ the thermal conductivity, $\alpha_c$ the convective heat transfer coefficient, $u_e$ the external temperature, $\rho$ the mass density, $c$ the specific heat and $n$ an outer normal unit vector, respectively. The finite element (quadratic tetrahedral) discretization of (1) yields a linear system of the form

\[
\begin{align*}
C \frac{du}{dt} + Ku &= f(t) \\
K &= K_c + K_h \\
f(t) &= R_q + R_f + R_h(t)
\end{align*}
\]

(2)

where the coefficient matrix $C$ of the time derivative of the nodal temperature is the capacitance matrix, the coefficient matrix $K_c$ and $K_h$ are the matrixes related to conduction and convection, respectively. The vectors $R_q$, $R_f$ and $R_h$ are vectors arising from heat flux, internal heat generation and surface
convection, respectively. Defining the time increment as $\Delta t$ and with a parameter $\varphi (0 < \varphi \leq 1)$, the following equation is obtained when Equation (2) is discretized on time.

$$\left[ [C]^n + \varphi \Delta [K]^n \right] u^{n+1} = \Delta [f]_n + \left[ [C]^n - (1 - \varphi) \Delta [K]^n \right] u^n$$

(3)

where superscript $n$ is the time step. In Equation (3), the dependency of variables are omitted.

3. SUBSTRUCTURING (REDUCTION TO THE INTERFACE PROBLEM)

The domain $\Omega$ is decomposed into $N$ non-overlapping subdomains, $\{ \Omega_i \}_{i=1,...,N}$. As usual the global stiffness matrix $K$ can be generated by subassembly:

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

(4)

where $R^{(i)T}$ is the 0-1 matrix which translates the global indices of the nodes into local numbering. With superscript $T$ we denote the transpose of a matrix (or vector). Let $u^{(i)}$ be the vector corresponding to the elements in $\Omega^{(i)}$ and it can be expressed as $u^{(i)} = R^{(i)T} u$. Each $u^{(i)}$ is split into degrees of freedom $u_B^{(i)}$, which correspond to $\partial \Omega^{(i)}$, called interface degrees of freedom and the remaining interior degrees of freedom $u_I^{(i)}$. The subdomain matrix $K^{(i)}$, the vector $u^{(i)}$ and the 0-1 matrices are then split accordingly:

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

(5)

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

(6)

and $R^{(i)T} = \begin{pmatrix} R_I^{(i)T} & 0 \\ 0 & R_B^{(i)T} \end{pmatrix}$. (7)

After eliminating the interior degrees of freedom, the system (2) reduces to a system on the interface:

$$Su_B = g$$

(8)

where $S = \sum_{i=1}^{N} R_B^{(i)} S^{(i)} R^{(i)T}$ is symmetric positive definite since $K$ is symmetric 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 $i = 1,...,N$, assumed to be positive semi-definite. The problem (8) can be solved by a preconditioned CG method which requires solving the following auxiliary problem:

$$z = M^{-1} r$$

(9)

where $r$ is the residual of (8) and $M$ is a preconditioner. An efficient solution of the large scale problem depends on how we choose an efficient and scalable preconditioner. This issue is dealt with in the next section.
4. BALANCING DOMAIN DECOMPOSITION

The BDD preconditioner proposed by Mandel is constructed by solutions 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:

\[ M_{BDD}^{-1} = Q_c + (I - Q_c S)Q_l (I - SQ_c) \]  \hspace{1cm} (10)

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

4.1 Local Level

The local level part of the preconditioner basically involves the solution of the local problems. The BDD algorithm employs the Neumann-Neumann preconditioner, \( Q_l \) is then expressed by

\[ Q_l = \sum_{i=1}^{N} R_B^{(i)} D^{(i)} S^{(i)\dagger} D^{(i)T} R_B^{(i)T} . \]  \hspace{1cm} (11)

The dagger (+) indicates the generalized inverse, since \( S^{(i)} \) are singular for floating subdomains[4]. The BDD method uses a collection of matrices \( D^{(i)} \) that determine the partition of unity on the interface,

\[ \sum_{i=1}^{N} R_B^{(i)} D^{(i)} R_B^{(i)T} = I . \]  \hspace{1cm} (12)

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.

4.2 Coarse Level

The application of the coarse term \( Q_c = R_0 (R_0^T SR_0)^{-1} R_0^T \) amounts to the solution of a coarse problem whose coefficient matrix is \( S_0 = R_0^T SR_0 \). The operator \( R_0 \) translates the coarse degrees of freedom to the corresponding global degrees of freedom and is defined by

\[ R_0 = [R_B^{(1)} D^{(1)} Z^{(1)}, \ldots, R_B^{(N)} D^{(N)} Z^{(N)}] . \]  \hspace{1cm} (13)

For the scalar heat conductive problem, \( Z^{(i)} \) is a column constant vector[2,4] and can be defined by

\[ Z^{(i)} = (1\ldots1)^T \]  \hspace{1cm} (14)

where the number of element “1” is for each interface point in subdomain \( i \). The operator \( R_0 \) is a \( N \) by \( N \) matrix, where \( N \) is the dimension of \( S \).

5. PARALLEL IMPLEMENTATION

In parallel implementation of DDM, a physical object, i.e. a whole domain to be analyzed, is hierarchically decomposed into two levels. As the first hierarchical level, the whole domain is Decomposed into some “parts”. Next as the second hierarchical level, each part is then decomposed into so-called subdomains.

This hierarchically structured DDM classifies processors into 3 groups, ‘Grand Parent’, ‘Parent’ and ‘Child’. One of the processors is assigned as Grand Parent, a few as Parent, and others as Child. The number of processors assigned as Parent is the same as that of parts.
The role of Grand Parent is to organize all processor communications (i.e. message passing) which occur between all processors. Parents prepare mesh data, manage FEA (Finite Element Analysis) results, and coordinate the CG iteration, including convergence decision for the CG iteration. Parents send data to Child processors, where FEA is performed in parallel. After the FEA, Child processors send the results to Parents. This computation will be repeated until the CG iteration is convergent. Fig. 1 shows the flow chart of the iterative domain decomposition method of non-steady heat conductive analysis.

6. NUMERICAL RESULTS

We begin with the non-steady heat conduction on a test model. The size of the model is 4000 x 4000 x 8000 mm. The time dependent heat transfer boundary conditions are set on the inner surface of the model. The heat transfer coefficient varies along the X-coordinate (Fig. 2) but constant in time while the external temperature (sink temperature) varies with both on the X-coordinate and time (Fig. 3). The temperature increases from 0 hrs (start up) to 21st hrs, it remains constant from 21st to 23rd hrs and again it decreases from 23rd hrs to 24th hrs (shut down) as shown in Fig. 3. Natural boundary conditions (heat flux is zero) are considered on the other portions.

Material properties of this model are shown in Table 1. In the non-steady heat conductive analysis, the following parameters are used: \( \Delta t = 200 \text{ sec}, \varphi = 0.5 \), and the maximum time step is 432. The initial temperature is 15°C. The model is expressed by 43,230 elements, divided into 6 parts and again divided into 120 subdomains. This analysis is performed by Pentium 4 (2.0GHz) consisting of 6 processing units. As a result of analysis, the computational performances are listed in Table 2 and a temperature distribution is shown in Fig. 4.

The present system is again applied to the finite element analysis of a nuclear reactor core component, the High Temperature Gas cooled Reactor (HTGR) model [4] (Fig. 5). The model is expressed with 32 parts, 3,200 subdomains, 1,519,644 elements, 2,438,883 nodes in the HDDM system and is analyzed with 32 SR8000 processors. For non-steady heat conductive analysis, following parameters are used: \( \lambda = 20.934 \left[ \text{W/m} \cdot \text{K} \right], \ c = 41.87 \left[ \text{J} / \text{Kg} \cdot \text{K} \right], \ \rho = 5000 \left[ \text{Kg} / \text{m}^3 \right], \ \Delta t = 1,000 [s], \ \varphi = 0.5 \) and time step is 10. Comparative results of a simplified diagonal scaling and the BDD method are shown in Table 2. The Fig. 6 predicts the numerical scalability of the method.
Table 1 Material properties of the test model

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal conductivity $\lambda$ [W/m.K]</td>
<td>8.24e-6</td>
</tr>
<tr>
<td>Specific heat $c$ [J/Kg.K]</td>
<td>0.139</td>
</tr>
<tr>
<td>Density $\rho$ [Kg/m$^3$]</td>
<td>7.85e-6</td>
</tr>
</tbody>
</table>

Table 2 Number of iterations of first non-steady loop

<table>
<thead>
<tr>
<th>Method</th>
<th>DDM</th>
<th>DIAG</th>
<th>BDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test model</td>
<td>71</td>
<td>41</td>
<td>21</td>
</tr>
<tr>
<td>HTGR</td>
<td>218</td>
<td>112</td>
<td>23</td>
</tr>
</tbody>
</table>

7. CONCLUSION

The parallel finite element system based on BDD is developed in this study and is applied to non-steady heat conductive analysis of over 2 million degrees of freedom problems. The future work is to speed up of the finite element analysis of a non-steady heat conductive problem.

REFERENCES