Parallel Adaptive Analysis of Incompressible Flow Using a Meshless Method

Minoru Shirazaki and Genki Yagawa

University of Tokyo, Japan

Abstract
In this paper, adaptive flow analysis using a meshless method named the “free-mesh method” on distributed-memory parallel computers is discussed. The free-mesh method is well compatible with the parallel computing because the global matrices of the finite element method, the mass, advection, diffusion and gradient matrices, are independently computed node-by-node, and, in addition because the node-element connectivity information is unnecessary. Therefore, the free-mesh method has compatibility with the parallel adaptive analysis. Adaptive techniques based on the global h-method are selected here, and the matrices for the updated node-distribution are computed by the free-mesh method. Some techniques to obtain high parallel efficiency such as techniques for the load balancing and the reduction of the interprocessor communication are presented.

1. Introduction
In computational fluid dynamics (CFD), particularly in the engineering field, the finite element method (FEM) has been successful due to its ability to analyze domains of arbitrary shape. Recently, the degrees of freedom in analysis models tend to be extremely large, and the geometries of the domain have become complex. To deal with such models with limited computer resources, two main issues must be solved. First, the computational time must be reduced. For this issue, efficient parallel methods, such as the parallelization of the processing loops [1-2] and the domain decomposition method (DDM) [3-4], have been studied. Second, the difficulties of global mesh generation over the whole domain of analysis must be avoided. To solve this issue, meshless analysis techniques, such as the reproducing kernel particle method (RKPM)[5], the diffuse element method (DEM)[6], the element-free Galerkin method (EFGM)[7] and so on, have been proposed.

On the other hand, a meshless method, “free-mesh method” (FMM) which proposed by Yagawa and Yamada [8], can effectively address both the issues simultaneously. FMM,
which is based on the usual FEM, can seamlessly process node-by-node local meshing and construction of system equations; therefore, PMM has the following features [9-10]: (1) The difficulty of global mesh generation over the domain of analysis can be avoided due to the local meshing process. (2) The entire FMM process can be easily parallelized since the process is performed node-by-node and is independent. (3) One can easily reconstruct system equations for the updated node-distribution in problems including adaptation and moving boundaries because of the seamless process through the local meshing and the construction of equations. By using FMM, an efficient adaptive flow analysis system on distributed-memory parallel computer can be developed.

In this paper, we present the coupling of the meshless method with an adaptive regenerating nodes strategy for the solution of unsteady viscous flows governed by the incompressible Navier-Stokes equation on parallel computers. A global h-type method is selected as the adaptive method, and almost all the processes are performed node-by-node and are independently. The strategy for the parallel adaptive analysis using FMM is as follows: first, a posteriori error estimation over the whole domain of analysis is performed, second, according to the error estimation, adaptive nodal density distribution for the next computational step is computed, and third, nodes based on the nodal density are regenerated. The first to the third processes are done on each processor with interprocessor communication. To attain workload balancing among processors, the new regenerated nodes are redistributed to processors, so that the number of nodes on each processor is as equal as possible.

The following section discusses FMM and the third section discusses a posteriori error estimation using FMM. The fourth section presents a node generation algorithm and some techniques for parallelization are shown in the following section. Conclusions are summarized in the final section.

2. Free Mesh Method
The basic flow chart of FMM is shown in Fig.1. Node coordinates and nodal density are given as input information without element data, where the nodal density of a node \( P_i \) called the central node is defined as the inverse value of the distance between the central node \( P_i \) and the nearest node pertinent to it.

A conceptual figure of FMM is shown in Fig.2. First, all the nodes called candidate nodes within the circular region around the central node \( P_i \), where the radius of the circle depends on the nodal density of the central node \( P_i \), are nominated. From these candidate nodes, several satellite nodes are selected by a rule proposed by Yagawa and Yamada [8]. In the next step, local triangular elements around the current central node are constructed temporarily by connecting the satellite nodes with the central node. Simultaneously, each row vector of the global matrices (the mass, advection, diffusion and gradient matrices) associated with the current central node is assembled from the local element matrices independently, and also is stored node-by-node. Global system equations can be obtained by repeating these procedures for all the nodes. These procedures can be performed on parallel environment.
3. A Posteriori Error Estimation using Free Mesh Method

In this paper, we use an error estimator proposed by Zienkiewicz et al. [11-12] for the posteriori error estimation. The error on a local element $\Omega$ is measured by the energy norm:

$$||e||^2 = \int_{\Omega} \left( (\tau_{ij} - \hat{\tau}_{ij})^t (\varepsilon_{ij} - \hat{\varepsilon}_{ij}) \right) d\Omega \quad (1)$$

where $\tau_{ij}$ and $\varepsilon_{ij}$ are the exact values of the deviatoric stress and the strain rates, while $\hat{\tau}_{ij}$ and $\hat{\varepsilon}_{ij}$ are solutions computed by FMM. The error is approximated as described below because the exact solution is not practically available. When the linear elements are used as the local elements, the solutions $\hat{\tau}_{ij}$ and $\hat{\varepsilon}_{ij}$ are constant and are not continuous between two neighboring elements. Therefore, acceptable distributions of the stress and the strain rates can be obtained by a nodal averaging with respect to local elements around the central node, i.e.,

$$\tau_{ij} \approx \hat{\tau}_{ij} = \sum_{k=1}^{n_{P_i}} (\hat{\tau}_{ij})_k / n_{P_i} \quad (2)$$

$$\varepsilon_{ij} \approx \hat{\varepsilon}_{ij} = \sum_{k=1}^{n_{P_i}} (\hat{\varepsilon}_{ij})_k / n_{P_i} \quad (3)$$

where $(\hat{\tau}_{ij})_k$, $(\hat{\varepsilon}_{ij})_k$ and $n_{P_i}$ denote the stress on the $k$-th local element, the strain and the number of the local elements, respectively. $\hat{\tau}_{ij}$ and $\hat{\varepsilon}_{ij}$ are better approximations than $(\hat{\tau}_{ij})_k$ and $(\hat{\varepsilon}_{ij})_k$.

In this analysis based on FMM, the error on a current central node $||e||_n$ can be estimated by averaging the contribution of local elements.

$$||e||^2_n = \sum_{k=1}^{n_{P_i}} ||e||^2_{\Omega_{P_i,k}} / n_{P_i} \quad (4)$$

$$||e||^2_{\Omega_{P_i,k}} = \int_{\Omega_{P_i,k}} \left( (\hat{\tau}_{ij}' - \hat{\tau}_{ij})^t (\hat{\varepsilon}_{ij}' - \hat{\varepsilon}_{ij}) \right) d\Omega \quad (5)$$

Here, $\Omega_{P_i,k}$ denotes the $k$-th local element, and $\hat{\tau}_{ij}'$ and $\hat{\varepsilon}_{ij}'$ are the approximate stress and strain rates in the local elements.

The percentage error $\eta$ is defined as:

$$\eta = \frac{||e||}{||E||} \quad (6)$$

$$||e||^2 = \sum_{n=1}^{N} ||e||^2_n, \quad ||E||^2 = \sum_{n=1}^{N} ||E||^2_n, \quad (7)$$

$$||E||^2_n = \sum_{k=1}^{n_{P_i}} ||E||^2_{\Omega_{P_i,k}} / n_{P_i}, \quad ||E||^2_{\Omega_{P_i,k}} = \int_{\Omega_{P_i,k}} \left( \hat{\tau}_{ij}^t \hat{\varepsilon}_{ij} - p^t \hat{\varepsilon}_{kk} \right) d\Omega \quad (8)$$

where $N$ denotes the number of nodes and $p$ is the pressure. We define a maximum permissible error for energy norm $\bar{\eta}$, and require that the condition $\eta > \bar{\eta}$ is satisfied in
the whole domain. Assuming that the error is equally distributed over the nodes, the maximum permissible error of each node $\bar{e}_{\text{max}}$ is:

$$\bar{e}_{\text{max}} = \bar{\eta} \frac{\|E\|}{\sqrt{N}}$$ (9)

Figure 3 shows a nodal density distribution based on the error estimation at non-dimensional time $t = 10$ for a lid-driven cavity problem, where the Reynolds number is 1000.

4. Algorithm of node generation based on nodal density control

New nodal density $D^{\text{new}}$ can be obtained from the maximum permissible error of each node $\bar{e}_{\text{max}}$ and current nodal density $D$:

$$D^{\text{new}} = D^{\text{old}} \xi^{1/q}, \quad \xi = \frac{\|e\|_n}{\bar{e}_{\text{max}}}$$ (10)

where $q$ denotes the order of the interpolation function. By using the new nodal density $D^{\text{new}}$, new nodes can be generated.

The node generation procedures are as follows [13]: first, a number of initial points are prepared, as shown in Fig.4. Second, the initial points are checked one by one to be adopted as a node to be generated when the following two criteria are satisfied:

1. The point is inside the analysis domain (IN/OUT check).
2. The distance between the point and the nearest node already adopted is shorter than the value representing the nodal density field.

The value is defined as the inverse of the nodal density. Finally, the nodes with nodal density can be obtained as the nodal information.

Figures 5–13 show the node distribution and the velocity vector for the lid-driven cavity problem ($Re = 1000$).

5. Attaining workload balancing and reducing interprocessor communication

In FMM, the workload of a processor depends on the number of nodes processed by the processor. In the initial phase, in order to attain workload balancing among the processors, the analysis domain is decomposed into a number of subdomains by the recursive coordinate bisection (RCB) [14], so that the number of nodes in each subdomain is as equal as possible. As the node distribution is updated, the workload becomes unbalanced. Therefore, the domain decomposition with respect to the nodes must also be updated in order to regain load balance for the next computational step. A simple technique for the domain decomposition, which is the interprocessor communication not for information of each node but for information of each set of nodes, is as follows. (Fig.14(a)-(c)):

1. On each slave processor, nodes in the subdomains (e.g. every one thousand nodes) are grouped so that the number of nodes in each group is as equal as possible for all
the slave processors. The groups of the nodes are given a representative coordinate (Fig.14(a)).

2. Each slave processor sends information of the groups of nodes, which includes only the number of nodes and the representative coordinate, to the master processor (Fig.14(b)).

3. The master processor distributes these groups of the nodes instead of ordinary nodes to slave processors by RCB. Information of the partitions are sent to each slave processor (Fig.14(c)).

4. According to the information, dynamic workload balancing is attained by migrating the groups of the nodes from loaded processors to less loaded neighbors.

Using the above procedures, the nodes in the analysis domain can be assigned to each processor as equally as possible with reasonable communication costs.

6. Conclusions
Adaptive flow analysis using a meshless method on distributed-memory parallel computers was discussed. Adaptive techniques based on the global h-method were used here, and the system equations for the updated node-distribution were computed by the free-mesh method. To achieve high parallel efficiency, we introduced a domain decomposition and interprocessor communication techniques for updated nodes. In the conference, the parallel efficiency and some numerical examples will be presented.

References

Fig. 1: Flow chart of the free mesh method

Fig. 2: Local elements around current central node
Fig. 3: Nodal density based on the error estimation

Fig. 4: Node generation

Fig. 5: Initial node distribution (2522 nodes)

Fig. 6: Node distribution (t = 5, 2493 nodes)

Fig. 7: Velocity vector (t = 5, 2493 nodes)

Fig. 8: Node distribution (t = 10, 2469 nodes)

Fig. 9: Velocity vector (t = 10, 2469 nodes)
Fig. 10: Node distribution ($t = 15$, 2464 nodes)

Fig. 11: Velocity vector ($t = 15$, 2464 nodes)

Fig. 12: Node distribution ($t = 20$, 2468 nodes)

Fig. 13: Velocity vector ($t = 20$, 2468 nodes)

Fig. 14: Node redistribution procedures on parallel environment: (a) Some nodes in each slave are grouped. (b) Each slave sends the information of the groups to the master. (c) The master distributes these groups of the nodes instead of ordinary nodes to slave processors by Recursive Bisection.