INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 24, 1341–1352 (1997)

PARALLEL OVERLAPPING SCHEME FOR VISCOUS INCOMPRESSIBLE FLOWS

M. KAIHO*, M. IKEGAWA AND C. KATO

1st Department, Mechanical Engineering Research Laboratory, Hitachi, Ltd., 502 Kandatsu, Tsuchiura, Ibaraki 300, Japan

SUMMARY

A 3D parallel overlapping scheme for viscous incompressible flow problems is presented that combines the finite element method, which is best suited for analysing flow in any arbitrarily shaped flow geometry, with the finite difference method, which is advantageous in terms of both computing time and computer storage. A modified ABMAC method is used as the solution algorithm, to which a sophisticated time integration scheme proposed by the present authors has been applied. Parallelization is based on the domain decomposition method. The RGB (recursive graph bisection) algorithm is used for the decomposition of the FEM mesh and simple slice decomposition is used for the FDM mesh. Some estimates of the parallel performance of FEM, FDM and overlapping computations are presented. ©1997 by John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **24**: 1341–1342, 1997

No. of Figures: 10. No. of Tables: 2. No. of References. 8.

KEY WORDS: parallel; overlapping; FEM/FDM; Navier–Stokes

1. INTRODUCTION

The overlapping grid technique has gradually become one of the most popular algorithms in the field of computational fluid dynamics, allowing flows with arbitrarily shaped flow geometries to be analysed, such as flow around aeroplanes or in solving moving body problems. This technique divides the global analysis region into subregions and solves each subregion separately, independently of the other subregions. Computed results are exchanged between subregions as boundary conditions. From a mesh-generating point of view, this scheme is thought to be better than conventional schemes, because it is easier to generate an FEM mesh, particularly if it is only required near an arbitrarily shaped wall, rather than an FEM mesh, which must be generated throughout the entire region of interest.

Nakahashi and Obayashi[1] presented an FDM/FEM zonal approach to analyse compressible flows in turbine cascades and compressor blade rows. In this approach the regions near turbine cascades and compressor blades are covered by a boundary-fitted grid and the remaining regions are covered by a finite element mesh. The authors, however, have proposed an FEM/FDM overlapping scheme for viscous incompressible flows, which they have applied to two-dimensional moving body problems around high-speed trains.[2] The strategy behind this scheme is to use the FEM in regions near the wall boundaries and the FDM in other regions. This combination enables accurate solutions to be obtained for boundary layers and reduces computational storage and time requirements. The modified ABMAC method proposed by Viecelli[3] has been used as the basic algorithm for solving the

*Correspondence to: M. Kaiho, 1st Department, Mechanical Engineering Research Laboratory, Hitachi, Ltd., 502 Kandatsu, Tsuchiura, Ibaraki 300, Japan

CCC 0271–2091/97/121341–12 $17.50

Received 1 April 1996
Revised 5 September 1996

Navier–Stokes equations. In addition, the time integration scheme proposed by the authors[4] with second-order accuracy has been used to ensure stable computations. This scheme was extended to the three-dimensional heat and mass transfer and moving body problems in Reference 5 by the authors. Ogawa and Fujii[6] applied the overlapping technique to three-dimensional high-speed train problems using boundary-fitted grids both for subregions near the train and for other subregions.

The parallelization of this scheme is important and necessary to solve large-scale problems such as the lake fluid motion problems computed in Reference 5 or to obtain more accurate solutions for moving body problems such as high-speed train problems. We have applied the domain decomposition technique to parallelize this scheme. It is well known that the RSB (recursive spectral bisection) method[7] can produce the best surface/volume ratio among various domain decomposition algorithms. However, the standard RSB method requires a lot of CPU time and memory to achieve good decomposition, because an eigenvalue problem has to be solved in the algorithm. To improve the CPU and memory requirements, multilevel formulations of RSB were developed by various research groups. On the other hand, it is well known that the RGB (recursive graph bisection) method[8] requires less CPU time and memory than standard RSB, although RGB produces a relatively worse surface/volume ration than the RSB method. We selected the RGB method for the first try of performance estimates. For FDM parallelization, simple slice decomposition was used.

In this paper we first describe the FEM/FDM overlapping scheme and its parallelization, then show the parallel performance of FEM, FDM and overlapping computations using simple benchmark problems such as three-dimensional cavity problems and flow around a sphere.

2. FEM/FDM Overlapping Scheme

2.1. Solution algorithm

The governing equations for viscous incompressible flow problems are given by the Navier–Stokes equations and are expressed in tensor form using the summation conventions

$$\dot{U}_i + (U_j U_i)_{,j} = -P_{,i}/\rho + \nu U_{i,jj}, \tag{1}$$

$$U_{i,i} = 0, \tag{2}$$

where U_i is the velocity component in the x_i-direction, ρ is the density, ν is the kinematic viscosity and the superscripted dot indicates a time derivative. The boundary conditions are given on the boundary $\Gamma(\Gamma = \Gamma_1 + \Gamma_2)$ as

$$U_i = \bar{U}_i \quad \text{on } \Gamma_1, \tag{3}$$

$$n_i P/\rho - \nu n_j U_{i,j} = 0 \quad \text{on } \Gamma_2, \tag{4}$$

where the overbar indicates a prescribed value and n_i is the direction cosine with respect to the x_i-axis of the normal n drawn outwards on the boundary Γ. Equation (3) represents fixed velocity conditions and equation (4) represents stress-free conditions.

The finite element method based on the ABMAC method[3] is a time-marching method; to ensure stable and accurate computations, the time integration scheme proposed by the authors[4] has been used. To explain the time integration method briefly, the Navier–Stokes equation (1) is rewritten as

$$\dot{U}_i = F_i - \Pi_{,i}, \tag{5}$$

$$F_i = -(U_i U_j)_{,j} + \nu U_{i,jj}, \tag{6}$$

where $\Pi = P/\rho$. If the values of U_i and Π at time $t = n\Delta t$ are assumed known and are represented as U_i^n and Π^n, the velocity components at time $t = (n+1)\Delta t$, U_i^{n+1}, can be obtained using Taylor series expansion for U_i:

$$U_i^{n+1} \approx U_i^n + \dot{U}_i^n \Delta t + \ddot{U}_i^n \Delta t^2/2 = U_i^n + (F_i^n - \Pi_{,i}^n)\Delta t + (\dot{F}_i^n - \dot{\Pi}_{,i}^n)\Delta t^2/2. \tag{7}$$

When the Reynolds number is large, the advection terms dominate and equation (6) can be approximated as $F_i \approx -U_j U_{i,j}$. Using this relationship, equation (7) can be transformed as

$$U_i^{n+1} = U_i^n + (F_i^n + \nu_{jk}^n U_{i,jk}^n - \Pi_{,i}^n)\Delta t - \dot{\Pi}_{,i}^n \Delta t^2/2, \tag{8}$$

where

$$\nu_{jk}^n = U_j^n U_k^n \Delta t/2. \tag{9}$$

By performing time integration of (1) through the use of (8) we are able to establish a stable and accurate analysis,[4] eliminating latent numerical instabilities unavoidably introduced through the conventional Euler integration scheme which neglects the higher-order terms in (7).

The ABMAC method,[3] which is a fully explicit, two-step prediction–correction-type scheme, was applied to solve (1) by both FEM and the FDM. Flow analysis based on the ABMAC method employs the following two-step procedure in which time steps are separated by the increment Δt:

step 1

$$\tilde{U}_i = U_i^n + (F_i^n + \nu_{jk}^n U_{i,jk}^n - \Pi_{,i}^n)\Delta t, \tag{10}$$

step 2

$$U_i^{n+1} = U_i^n + \delta U_i, \qquad \Pi^{n+1} = \Pi^n + \delta\Pi, \tag{11}$$

where

$$\delta U_i = \sum \mathrm{d}\delta U_i^k, \qquad \delta\Pi = \sum \mathrm{d}\delta\Pi^k. \tag{12}$$

where

$${}^0U_i^{n+1} = \tilde{U}_i, \qquad {}^kU_i^{n+1} = {}^{k-1}U_i^{n+1} + \mathrm{d}\delta U_i^k, \quad k = 1, 2, \ldots, L, \tag{13}$$

$${}^0\Pi = \Pi^n, \qquad {}^k\Pi^{n+1} = {}^{k-1}\Pi^{n+1} + \mathrm{d}\delta\Pi^k, \quad k = 1, 2, \ldots, L, \tag{14}$$

where

$$\mathrm{d}\delta\Pi^k = -\Delta\tau\, {}^{k-1}U_{i,i}^{n+1}, \tag{15}$$

$$\mathrm{d}\delta U_i^k = -\Delta t\, d\delta\Pi_{,i}^k/2, \tag{16}$$

$\Delta\tau$ is a relaxation parameter for the pressure calculation and L is the number of iterations for step 2. These iterations continue up to L cycles until the average value of the divergence of ${}^kU_i^{n+1}$ falls below some defined error limit. In this modified ABMAC method the critical Courant number is about 0·5 for three-dimensional application problems from our experience, because Euler forward time integration is used for the momentum equation in step 1. The Galerkin method was employed for the finite element formulations and a conventional central difference scheme for a collocation grid was used for discretizations of (10)–(16) in the finite difference formulations. For details of the FEM formulations for the ABMAC method, see Reference 2.

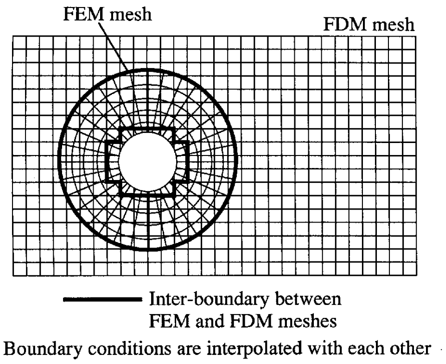


Figure 1. Strategy of FEM/FDM overlapping scheme

2.2. *Data exchange between FEM and FDM domains*

We will now consider the 2D analysis of flow using an FDM mesh and an FEM mesh as shown in Figure 1. Originally, the finite difference method utilized a staggered mesh for flow analysis. In the present analysis we have defined velocity components at grid points and pressure at the centre of a grid cell (collocation grid) as mentioned above. This is done to maintain consistency within the finite element method and to simplify computer programming. The value of velocity at nodal points and the value of pressure at the centre of an element on the FEM interboundary are projected from the FDM mesh as essential boundary conditions for the FEM analysis and *vice versa*. Before the projection the element number in which an FEM interboundary node is located and the local co-ordinate values in the element have to be computed. It is relatively easy to obtain the element number and the local co-ordinate values for the FEM interboundaries, while the reverse case requires some manipulation. The same shape function as used in the finite element formulation was applied to the projection of FDM interboundary nodes. To obtain the element numbers and the local co-ordinate values for the FDM interboundaries, the Newton–Raphson method was employed. See Reference 2 for details of the above procedures. The speed of these procedures has a great influence on the overall performance of the overlapping scheme when solving moving body problems, because these procedures have to be called at every time step. The speed of the parallel versions of these procedures has not been investigated so far. Therefore moving body problems were excluded from this paper.

3. PARALLEL IMPLEMENTATION

3.1. *FEM computations*

In this research the Hitachi SR4300, whose specifications are given in Table I was used for parallel computations. The SR4300 is an MIMD machine which has a native message-passing library for communications between nodes. Peak performance is 266 Mflops per node. Our system has 48 nodes and global peak performance is 12·6 Gflops. The system has 40 nodes of 64 MB memory and eight nodes of 128 MB memory, so a total of 3·6 GB of memory is available. The system constitutes a kind of large workstation cluster.

For parallel implementation of unstructured mesh computations the domain decomposition method is generally used and can be implemented more easily for explicit schemes than implicit schemes. As mentioned in Section 1, the RSB (recursive spectral bisection) method[7] is well known to be able to produce the minimum surface/volume ratio for a domain, which means that the best possible communication performance between domains can be achieved. However, the standard RSB method

Table I. Machine Specification

Machine	Hitachi SR4300
Number of processors	48
Peak performance	266 Mflops ×48 = 12·8 Gflops
Memory	64 MB ×40 ± 128 MB ×8 = 3·6 GB
Disk	9 GB ×4 + 1 GB ×44 = 80 GB

needs a lot of computing time and memory to decompose the mesh, because an eigenvalue problem has to be solved in the algorithm. To ameliorate these problems, multilevel formulations of RSB were developed by various research groups. In the present implementation the RGB (recursive graph bisection) method,[8] which is well known to require less computing time and memory than the standard RSB method, was selected for the first try of performance estimates. The strategy for FEM parallelization is shown in Figure 2. We will now consider the correction of the velocity component $U_{i\mathrm{A}}$ in domain A. In normal computations, elemental residues $\mathrm{d}U_{i\mathrm{e}}$ calculated in elements adjacent to a node are scattered to the node and the scattered nodal residual $\mathrm{d}U_i$ is added to U_i. At a node on the interboundary, scattering is different from normal computations. First, elemental residuals $\mathrm{d}U_{i\mathrm{eB}}$ in domain B are calculated in elements adjacent to the node on the interboundary. These $\mathrm{d}U_{i\mathrm{eB}}$ are then scattered to $\mathrm{d}U_{i\mathrm{B}}$ as nodal residuals. The value of $\mathrm{d}U_{i\mathrm{B}}$ is sent from domain B to domain A and added to $\mathrm{d}U_{i\mathrm{eb}}$ calculated in domain A. Finally, $\mathrm{d}U_{i\mathrm{A}}$ is added to $U^n_{i\mathrm{A}}$ and the corrected $U^{n+1}_{i\mathrm{A}}$ is obtained. The reserve case requires exactly the same procedures.

In the actual implementation, good communication performance cannot be achieved if node-to-node communication is used. Therefore, in the present implementation, domain-to-domain communication has been employed by packing all the residual data which should be sent from one domain to another. This packing technique is essential to obtain high communication performance.

3.2. FDM computations

Figure 3 shows the strategy for FDM parallelization. As mentioned in Section 1, slice domain decomposition was used for FDM parallelization as shown in Figure 3. The analysis domain is sliced along the axis whose dividing number is the largest. The broken line shows imaginary analysis

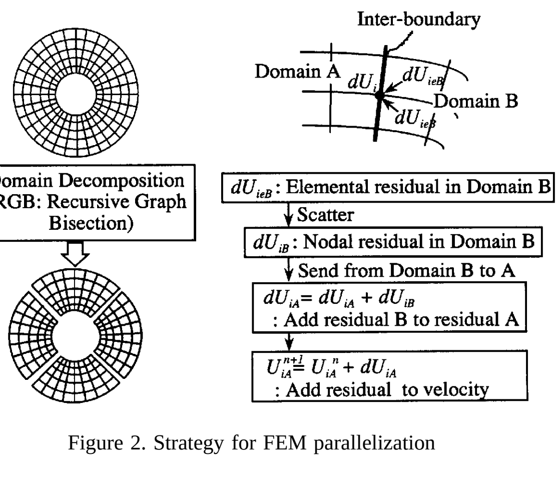


Figure 2. Strategy for FEM parallelization

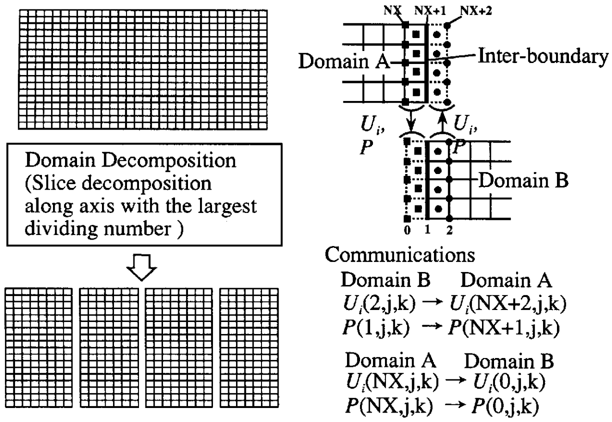


Figure 3. Strategy for FDM parallelization

regions for each domain, and velocity and pressure in imaginary analysis regions are brought from adjacent domains. After this procedure, nodal velocities on the interboundary are calculated as usual. For the example in Figure 3, $U_i(2, j, k)$ and $P(1, j, k)$ in domain B are sent to $U_i(NX + 2, j, k)$ and $P(NX + 1, j, k)$ in domain A respectively and $U_i(NX, j, k)$ and $P(NX, j, k)$ in domain A are set to $U_i(0, j, k)$ and $P(0, j, k)$ in domain B respectively. Packing communications of U_i and P have been employed for the purpose of achieving good communication performance. Communication occurs only between adjacent domains in the case of slice domain decomposition, and the size of data to be communicated is generally larger than for FEM communications. This is because a larger mesh size is generally used for the FDM than for the FEM in the present overlapping scheme.

3.3. Overlapping computations

Two implementation types can easily be considered for the overlapping scheme as shown in Figure 4. Type A is SPMD (single-programme multile-data)-type programming and can basically utilize the structure of a serial programme. In type A both FEM and FDM meshes are decomposed into the same number of domains. For the example in Figure 4 the meshes are decomposed into four domains: FEM1, FEM2, FEM3 and FEM4 for the FEM mesh and FDM1, FDM2, FDM3 and FDM4 for the FDM mesh. Then processor 1 computed FEM1 and FDM1, processor 2 computes FEM2 and FDM2, etc. For the purpose of easy explanation, the domain decomposition of the FEM mesh in Figure 4 is simplified and the number of domains adjacent to one domain is only two, as in the FDM decomposition. However, the number of adjacent domains is more than two in general for an arbitrarily shaped mesh and communications for FEM computations are randomly executed over all the domains, while communications for FDM computations are executed only between adjacent domains. Communications for overlapping computations are basically executed over all the domains, although these communications could create load imbalance, to be discussed later. FEM and FDM computations can be separately achieved because of mutual independence. Therefore type B can be considered for parallel implementation of the overlapping scheme. In type B the FEM mesh is generally decomposed into a different number of domains from the FDM mesh. In the example in Figure 4, both FEM and FDM meshes are decomposed into two domains only for simplicity of explanation. The FEM mesh is decomposed into FEM1 and FEM2 and the FDM mesh is decomposed into FDM1 and FDM2. Processor 1 computed only FEM1, processor 2 computes FEM2, processor 3 computes FDM1 and processor 4 computed FDM2. Only communications for overlapping

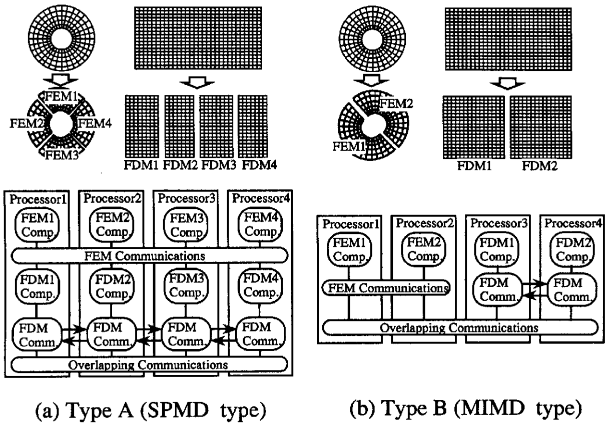


Figure 4. Parallelization types for overlapping scheme

computations are executed over all the domains. Table II gives a comparison between types A and B. Both types A and B have to consider load balancing between domains for each FEM or FDM mesh. In addition, type B has to consider load balancing between FEM and FDM computations. It is difficult to achieve good load balancing between FEM and FDM computations, because the computing speed of each FEM or FDM computing module depends largely on the number of elements in one domain and processors have to be appropriately distributed to each FEM or FDM computation. Therefore, from the load-balancing point of view, type A is superior to type B. On the other hand, the surface/volume ratio of type A for the decomposed domain is larger than that of type B. This generally means that more communication time is required for type A than for type B, because the number of nodes or elements in one domain for type A is smaller than that for type B. From the viewpoint of communication time, type B is superior to type A. We suppose that the difficulty of load balancing in type B dominates the overall performance of overlapping computations, although we have not yet investigated that point. Therefore we selected type A.

Figure 5 shows communications between FEM and FDM meshes. In this example of the two-dimensional circular cylinder problem, FEM1 and FEM3 are located over FDM1 and FDM2, FEM2 is located in FDM1 and FEM4 is located in FDM2. FDM3 and FDM4 require no communication. That is, communications between FEM and FDM meshes are concentrated on processors 1 and 2. For this reason, communication in this implementation could create imbalance.

Finally, the flow chart of a single programme for overlapping computations is shown in Figure 6. As mentioned above, this implementation excludes the solutions to moving body problems, which is

Table II. Comparison of types A and B

	Type A	Type B
Load balancing	Between domains for each mesh	Between domains for each mesh and between FEM and FDM
Surface/volume ratio	Large (number of nodes in one domain is smaller than for type B)	Small

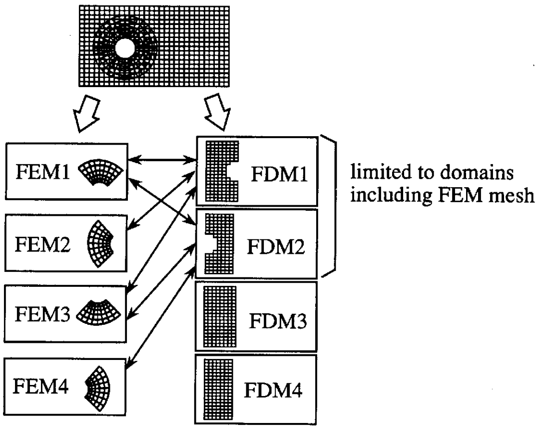


Figure 5. Communications between FEM and FDM meshes

why the communication data-generating part for overlapping is located outside the time loop. The performance of this part has a large influence on the overall performance of the programme for moving body problems.

4. PERFORMANCE ESTIMATES

4.1. FEM performance

The programme in this implementation can deal with a single mesh, i.e. a single FEM mesh or a single FDM mesh. To estimate FEM computations, the three-dimensional rectangular mesh shown in Figure 7 was used and the simple duct flow problem was solved. The reason for using the simple rectangular mesh is to compare fairly the FEM performance with the FDM performance discussed in the next subsection. In the case of a more complex-shaped FEM mesh the performance could be

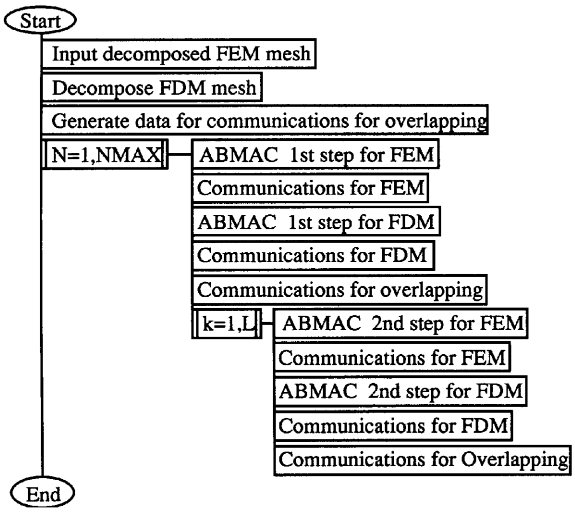
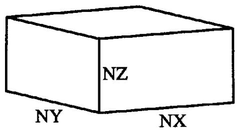


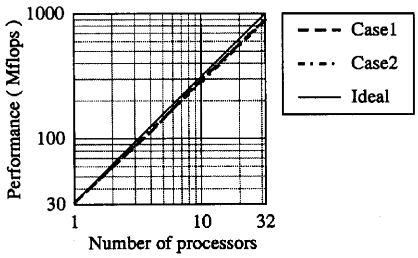
Figure 6. Flow chart for parallel overlapping scheme



(a) Mesh configuration

No. of CPU	Case 1		Case 2	
	NX×NY×NZ	No. of nodes	NX×NY×NZ	No. of nodes
1	26×26×26	19,683	26×26×26	19,683
2	33×33×33	39,304	52×26×26	38,637
4	42×42×42	79,507	104×26×26	76,545
8	53×53×53	157,464	208×26×26	152,361
16	67×67×67	314,432	416×26×26	303,993
32	85×85×85	636,056	832×26×26	607,257

(b) Problem size



(c) Performance

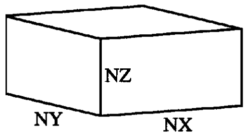
Figure 7. Performance of FEM computations

worse than the following results. The performance does not depend on the flow problem to be solved but depends on the number of iterations for step 2 of the ABMAC method.

In these estimates the number of iterations was set to 50 for the measurements of the CPU time, from the authors' experience, to get accurate solutions. Therefore communications were executed 51 times per time loop. This iteration is the same throughout the measurements in this paper. *NX*, *NY* and *NZ* are the dividing numbers along the *x*-, *y*- and *z*-axis respectively. The relationship between the number of processors and computing performance including the communication part for the FEM was measured for two cases in the figure. For both cases 1 and 2 the size of the problem increases almost linearly with the number of processors. In case 1, *NX*, *NY* and *NZ* increase simultaneously, while only *NX* increases in case 2. The maximum number of nodes for 32 processors is about 640,000. The performance curves shows good parallel performance. Up to 32-processor calculations, about 90% parallel efficiency was achieved. It was proved that the domain decomposition method by the RGB algorithm for FEM computations shows good parallel performance, owing to the effect of packing communication data.

4.2. FDM performance

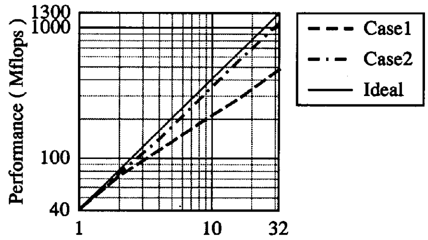
To estimate FDM computations, the same three-dimensional rectangular problem as shown in Figure 7 was used. The relationship between the number of processors and computing performance including the communication part for the FDM was measured for the two cases in Figure 8 similarly to the FEM estimate. For both cases 1 and 2 the size of the problem increases almost linearly with the number of processors. In case 1, *NX*, *NY* and *NZ* increase simultaneously, while only *NX* increases in case 2. In case 2 the number of interboundary nodes does not change with the problem size because of the slice domain decomposition. The maximum number of nodes for 32 processors is about 6,300,000. The performance curve for case 1 does not show good performance, whilst that for case 2 shows good parallel performance and about 84% parallel efficiency was achieved for calculations using up to 32 processors. This is because the surface/volume ratio of case 1 is larger than that of case 2 and the data size for communications is 10 times larger than that for the FEM cases. Because case 1 is more general than case 2, it can be proved that the slice decomposition method for FDM



(a) Mesh configuration

No. of CPU	Case 1		Case 2	
	NX×NY×NZ	No. of nodes	NX×NY×NZ	No. of nodes
1	57×57×57	195,112	57×57×57	195,112
2	72×72×72	389,017	114×57×57	386,860
4	91×91×91	778,688	228×57×57	770,356
8	115×115×115	1,560,896	456×57×57	1,537,348
16	146×146×146	3,176,523	912×57×57	3,071,332
32	184×184×184	6,331,625	1824×57×57	6,139,300

(b) Problem size



(c) Performance

Figure 8. Performance of FDM computations

computations shows bad parallel performance for very-large-scale problems even if the effect of packing communication data is included. Improvements in this communication speed have to be achieved in future.

4.3. *FEM/FDM performance*

Figure 9 shows a numerical example of three-dimensional flow around a sphere at a Reynolds number of 100. The figure includes the decomposed sphere mesh divided into eight domains and the

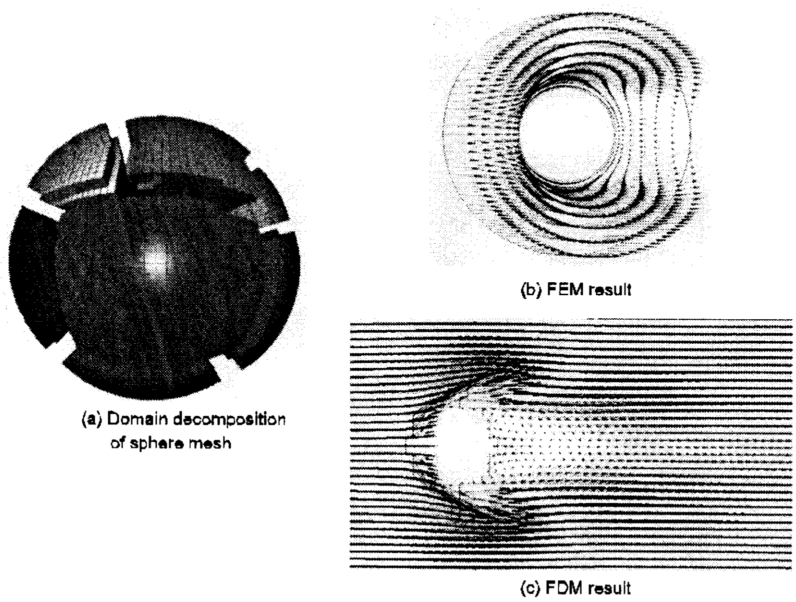
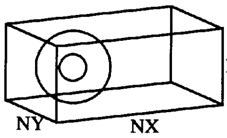


Figure 9. Numerical example of overlapping scheme

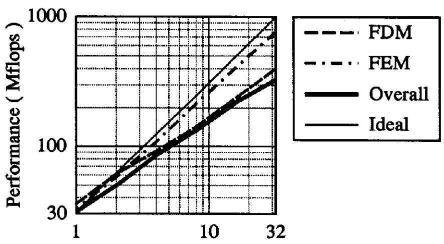


(a) Mesh configuration

No. of CPU	FEM No. of nodes	FEM No. of I.B*	FDM NX×NY×NZ	FDM No. of nodes	FDM No. of I.B*
1	5,824	728	50×25×25	34,476	152
2	10,816	1,352	64×32×32	70,785	314
4	21,184	2,648	80×40×40	136,161	460
8	42,354	4,706	100×50×50	262,701	696
16	83,256	6,938	128×64×64	545,025	1,146
32	163,234	9,602	160×80×80	1,056,321	1,588

I.B* : Inter-Boundary

(b) Problem size



(c) Performance

Figure 10. Performance of FEM/FDM overlapping computations

fluid velocity profiles for FEM and FDM meshes at the centre section of the sphere. The computed results show qualitatively the flow pattern around the sphere. From this computation it was able to be confirmed that this implementation was correct and does not include abnormal communications. To estimate the performance of FEM/FDM overlapping computations, the three-dimensional sphere problem is shown in Figure 10. The problem size based on the number of processors is shown in the table. The performance curves for each FEM or FDM computation seem to have the same tendency as those of single-mesh computations. Efficiencies are slightly worse than those for single-mesh measurements because the problem size is smaller. The overall performance curve is a little worse than the FDM curve but not too far from the FDM curve. This means that communications for overlapping show almost the same parallel performance as FDM computations and have little influence on overall performance in this implementation. However, it is possible for the overlapping communication part to dominate the overall performance if the FDM computation part is improved. Improvement of this part is necessary to achieve higher overall performance.

5. CONCLUSIONS

A three-dimensional parallel FEM/FDM overlapping scheme was developed for the purpose of solving large-scale problems. An SPMD-type programming model was used for this implementation. Performance was estimated using the Hitachi SR4300, which is basically an MIMD machine with a kind of large workstation cluster. Estimates for FEM computations using a domain-decomposed mesh by the RGB method show good parallel performance and can maintain about 90% parallel efficiency for computations involving up to 32 processors. From performance estimates for FDM computation using slice decomposition, communications for FDM calculation were proved to be improved for large-scale problems. From overall performance estimates it was proved that communications for overlapping have little influence on overall performance in this implementation. Speed-up of the data generation part for overlapping communications has not yet been considered. Consequently, moving body problems have been excluded in this paper.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Neil Hurley and Dr. Darach Golden of Hitachi Dublin Laboratory for providing the domain decomposition programmes used for the FEM mesh.

REFERENCES

1. K. Nakahashi and S. Obayashi 'FDM–FEM zonal approach for viscous flow computations over multiple bodies', *AIAA Paper 87-0604*, 1987.
2. M. Ikegawa, *et al.*, 'FEM/FDM composite scheme for viscous incompressible flow analysis'. *Comput. Methods Appl. Mech. Eng.*, **112**, 149–163 (1994).
3. J. A. Viecelli, 'A computing method for incompressible flows bounded by walls', *J. Comput. Phys.*, **8**, 119–143 (1971).
4. M. Ikegawa, *et al.*, 'Three-dimensional turbulent flow analysis in a cleanroom by the finite element method', in FED Vol. 66, ASME, New York, 1988, pp. 161–167.
5. M. Kaiho, M. Ikegawa and C. Kato, 'Application of a 3-dimensional FEM/FDM overlapping scheme to heat and mass transfer and moving-boundary problems', *Int. J. Comput. Fluid Dyn.* in press.
6. T. Ogawa and K. Fujii, 'Numerical simulation of compressible flow induced by a train moving in a tunnel'. *AIAA Paper 93-2951*, 1993.
7. H. Simon, 'Partitioning of unstructured problems for parallel processing', *Comput. Syst. Eng.*, **2**, 135–148 (1991).
8. C. Farhat and M. Lesoinne, 'Automatic partitioning of unstructured meshes for the parallel solution of problems in computer mechanics', *Int. j. numer. methods eng.*, **36**, 745–764 (1993).