Computer Assisted Mechanics and Engineering Sciences, 1: 159–176, 1994.

Local mesh refinement for incompressible fluid flow with/without free surfaces

Haruo Terasaka, Hikaru Kajiwara and Nobuatsu Tanaka
Nuclear Engineering Laboratory, Toshiba Corporation
4-1, Ukishima-cho, Kawasaki-ku, Kawasaki, 210, Japan

(Received August 17, 1994)

This paper presents a local mesh refinement (LMR) technique and its application to incompressible fluid flows with or without free surface boundaries. In the LMR method patches of fine grid are embedded in arbitrary regions of interest. Hence, more accurate solutions can be obtained with a lower number of computational cells. LMR is very suitable for the simulation of free surface movements because free surface flow problems generally require a finer computational grid to obtain adequate results. By means of this technique one can place finer grids only near surfaces and thus greatly reduce the total number of cells and computational costs. This paper introduces two LMR codes. Numerical examples calculated with the codes demonstrate well the advantages of the LMR method.

1. INTRODUCTION

Local Mesh Refinement (LMR) is a kind of the zone decomposition method. The basic idea of the technique is based on local adaptive mesh refinement (LAMR) developed by Berger et al. [1] to track shock wave propagation in compressible flows. LAMR uses a sequence of overlapping grids of increasing fineness. The grid is adaptively refined until the solution error estimate is below a certain desired level. The main difficulty of this method is the need for data structures not usually found in numerical software, which makes programs highly complex.

To adapt LAMR to incompressible flow analysis, it is converted into LMR by neglecting the adaptiveness, but retaining the simple data structure used in conventional flow analysis programs. In LMR, grid patches with an arbitrary level of fineness are embedded in arbitrary regions of interest. Although the grid patches are "frozen", so that the computational grid system does not change, the fine mesh regions are confined within the patches, and the increase in the number of cells is restrained unlike the conventional program. As a result, more accurate solutions can be obtained with lower computation costs.

The LMR technique is extended, which makes it applicable to free surface flow problems. Several methods have previously been used to approximate free surfaces in finite difference numerical simulations. From among the methods, the volume of fluid (VOF) method [2] is chosen for this study because of its flexibility and wide applicability. A special interpolation scheme is employed to re-distribute the fluid volume fraction in coarser cells into finer cells contained within the coarser ones. The interpolation scheme is very important to maintain the mass conservation of the fluid.

This paper introduces two LMR programs, FVM2D and LMR3D, developed by the authors. FVM2D solves two-dimensional incompressible flow problems with fixed boundaries by using the artificial compressible (AC) scheme. Since calculations of fluid dynamics in each patch can be processed independently and the AC scheme calculates the pressure field directly from the time evolution equation instead of solving the Poisson equation, this combination of LMR and the AC scheme is very suitable for parallel processing computers. LMR3D is a three-dimensional code for incompressible flow with free surface boundaries. It can solve a wide range of free surface problems in Cartesian or cylindrical coordinate systems with the LMR grid. The free surfaces are approximated

by the VOF method. Several numerical examples are solved to verify the codes. The results clearly demonstrate the advantages of the LMR methods.

2. OUTLINE OF FVM2D

FVM2D is a two-dimensional LMR program which solves incompressible viscous flow problems without free boundaries by means of the AC method. An outline of the code is given below.

2.1. Basic conservation equation

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \tag{1}$$

where

$$Q = \begin{bmatrix} p \\ u \\ v \end{bmatrix}, \quad F = \begin{bmatrix} \frac{u}{\beta\rho_0} \\ (u^2+p) - \nu_0 \frac{\partial u}{\partial x} \\ uv - \nu_0 \frac{\partial v}{\partial x} \end{bmatrix}, \quad G = \begin{bmatrix} \frac{v}{\beta\rho_0} \\ uv - \nu_0 \frac{\partial u}{\partial y} \\ (v^2+p) - \nu_0 \frac{\partial v}{\partial y} \end{bmatrix},$$

and β, ρ_0 and ν_0 are artificial compressibility, fluid density and kinematic viscosity, respectively.

2.2. Difference scheme

Equation (1) is discretized by the following numerical schemes:

- in time — the 1st-order Euler scheme, or the 2nd-order RRK (Rational Runge-Kutta) scheme
- in space — the 3rd-order TVD (Total Variation Diminishing) scheme

2.3. Grid system

Figure 1 is an example of an LMR grid system in FVM2D. The grid system is constructed by a finite union of rectangles. The rectangular patches are classified by their fineness level. FVM2D has five levels of fineness and the refinement ratio, $r = \frac{\Delta x_{l-1}}{\Delta x_l}$, equals 2. A cell in a grid patch of level $(l-1)$ is therefore divided into four sub-cells of level l. The lowest level equals zero, and the level zero grid is the same as that of the usual computational grid. It covers all the computational regions and so it is called the fundamental grid system. A non-uniform grid is available as the fundamental grid system in FVM2D. The highest grid level is 4, in which the spacing Δx_4 equals $\frac{1}{16}\Delta x_0$, and a cell in the fundamental grid is divided into 16×16 sub-cells at level 4. The number of patches is restricted only by computer storage. Users of FVM2D can place an arbitrary number of grid patches with the desired resolutions at arbitrary locations on the fundamental grid, within the limitations mentioned above.

2.4. Mesh arrangement

All of the basic variables, i.e. u, v and p, are defined at each cell center. A linear interpolation is employed to obtain boundary values between grids of two different levels, as shown in Fig. 2. The figure shows that the grid level of the left half is $(l-1)$ and that of the right half is l. To calculate the flux vector between the grids of level l and $(l-1)$, the interpolated variables, as indicated by squares, are necessary.

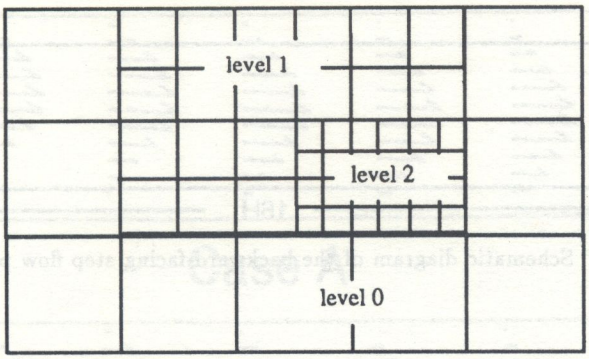


Fig. 1. Sketch of the LMR grid system used in FVM2D

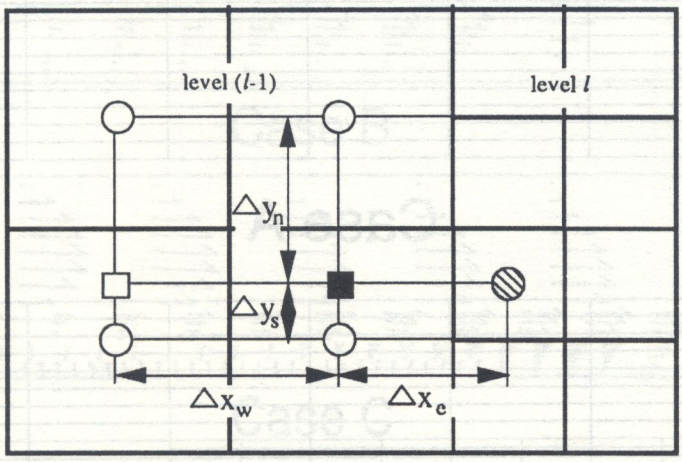


Fig. 2. Linear interpolation

2.5. Data structure

To achieve efficient vector operation, the variables in the code have a one-dimensional array data structure, namely $u(n)$, $v(n)$, $p(n)$ and so on. The index n is the number of the cell and is calculated from the following expression:

$$n(i,j,i_s,j_s) = n_0(i,j) + i_s - 1 + 2^{l(i,j)}(j_s - 1), \tag{2}$$

where i and j are the cell indices of the coarsest grid, $n_0(i,j)$ is the first number of the sub-cell included in cell (i,j) of the zero level grid, $l(i,j)$ denotes the grid level of the sub-cell, and i_s and j_s denote the sub-cell indices.

3. Numerical examples with FVM2D

3.1. Backward-facing step flow

A laminar backward-facing step flow with $Re = 229$ is calculated to verify the code. The Reynolds number, Re, is defined by UH/ν_0, where U denotes the averaged inlet velocity and H is the step height. Figure 3 is a schematic diagram of the back step flow and Figure 4 shows the four grid systems used in this study. The first grid system, Case A, comprises a 9×12 uniform fundamental grid. The fourth one, Case D, is also a fundamental grid, but twice as fine as the first one. The second system, Case B, is composed of a fundamental grid and a level 1 grid patch. The third one has the same fundamental grid as Case B, plus grid patches of level 1, level 2 and level 3.

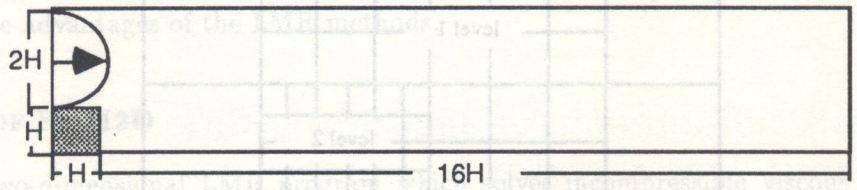
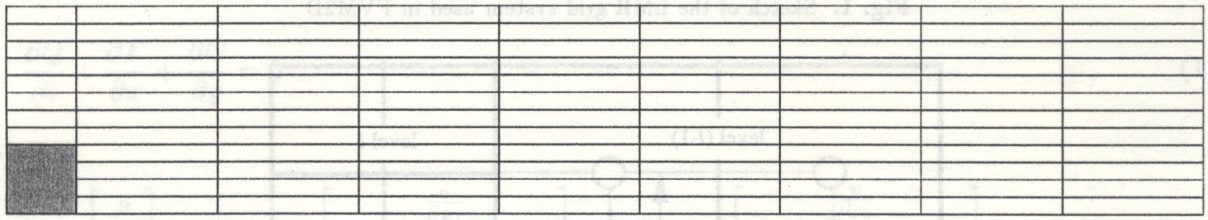
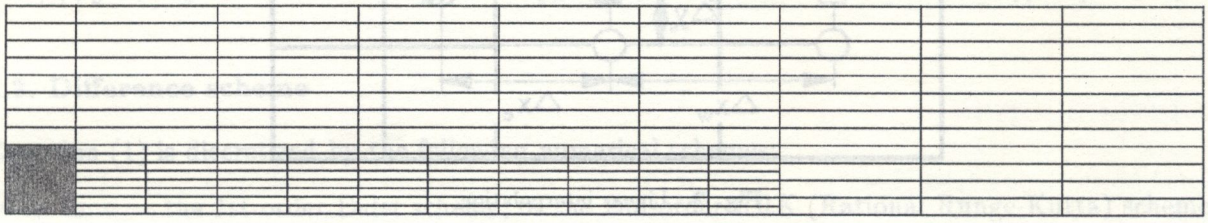


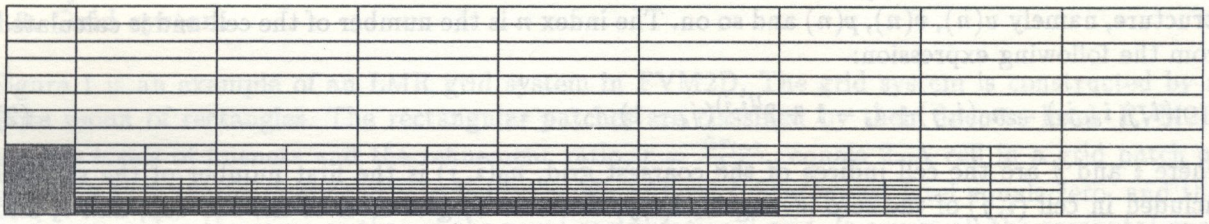
Fig. 3. Schematic diagram of the backward-facing step flow problem



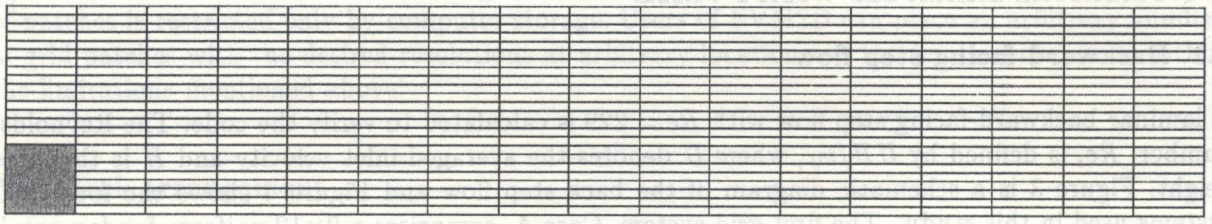
Case A



Case B



Case C



Case D

Fig. 4. Grid systems for the backward step flow

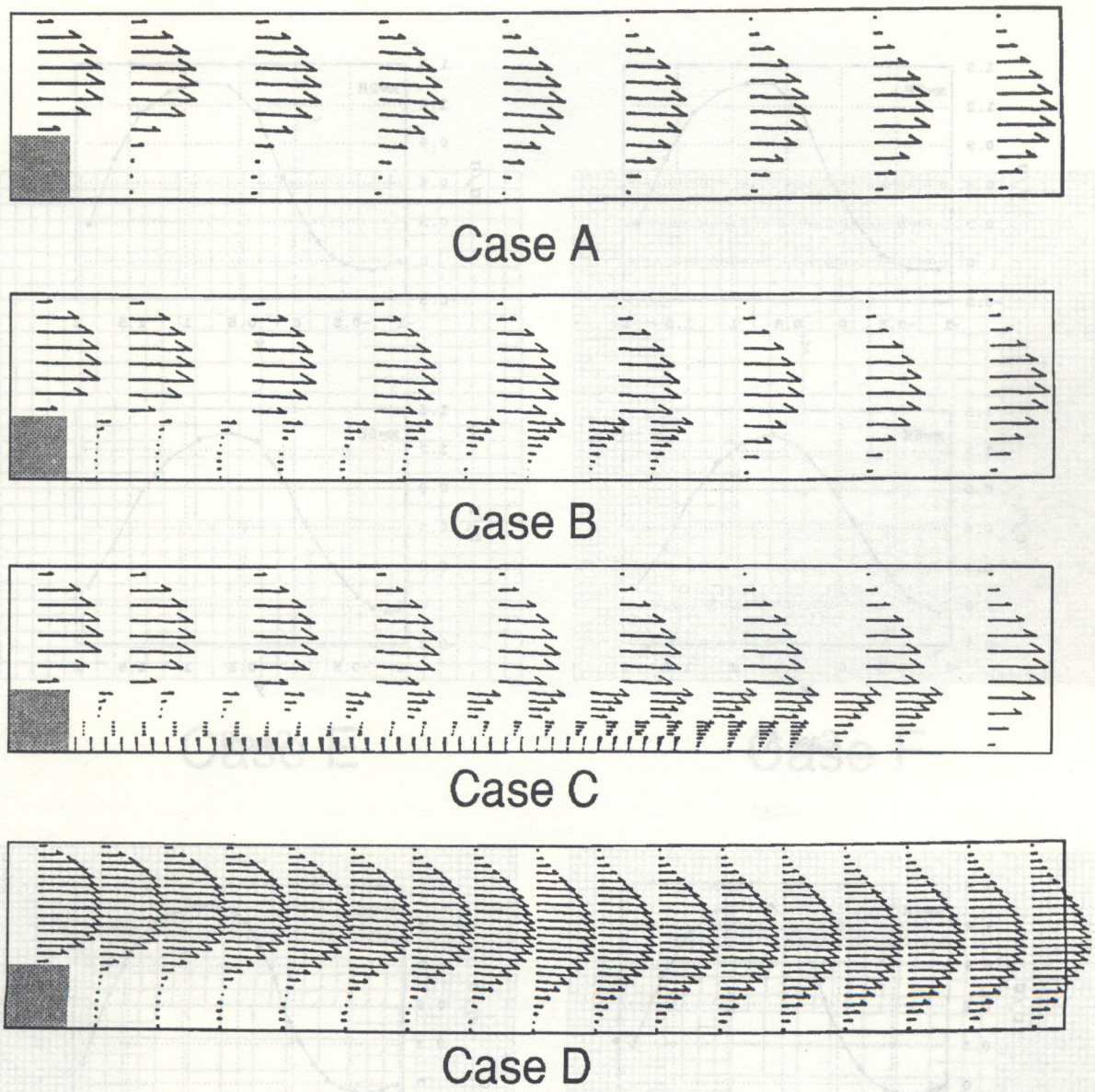
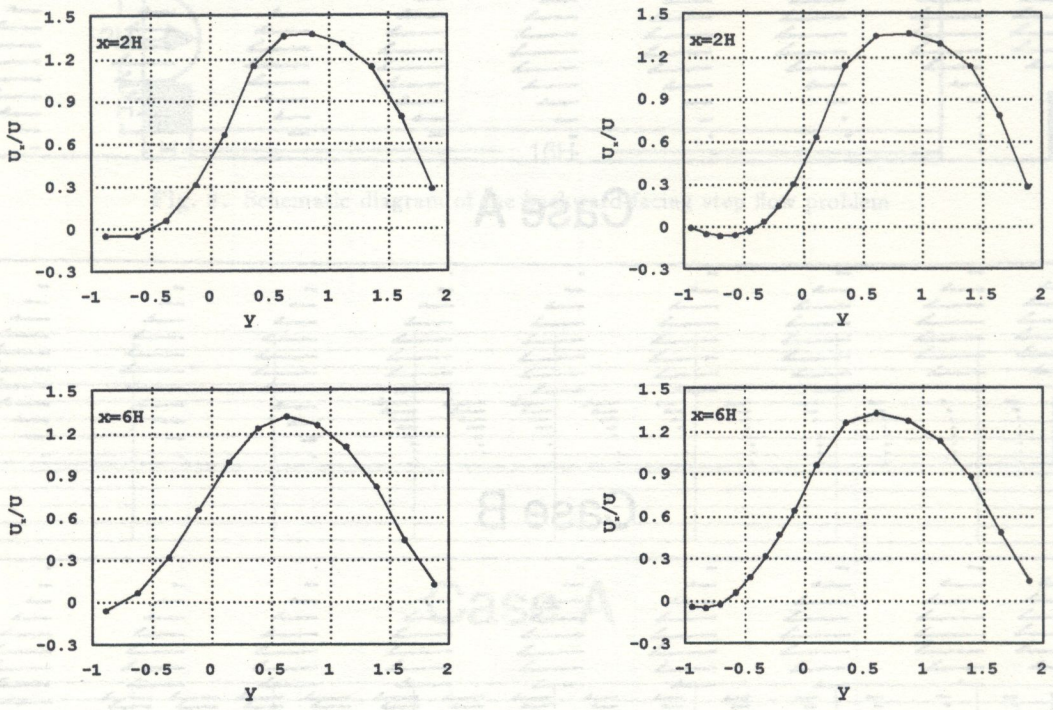


Fig. 5. Velocity vectors for the backward step flow

Figures 5 and 6 show the velocity vector distributions and the profiles of non-dimensional velocity, $u(y)/U$, obtained from each grid system. From these figures, it is clear that the velocities in the patches have almost the same accuracy as the results obtained from Case D. The computation time in Case B is about 40% that of Case D. This means that almost the same results are obtained from the LMR calculation at less than half the computational cost.

3.2. Shear-driven flow in a square cavity

Cavity flow problems are frequently solved to qualify a code since they are easy to solve and there is a good reference by Ghia et al. [3]. The authors [4] once solved a cavity flow using a 256×256 uniform mesh on a massively parallel processor, CM-2, and compared the results with the reference. A square cavity flow at $Re = 1000$ is again solved with FVM2D to demonstrate LMR performance. The results of Cases A to D are obtained with the use of uniform fundamental grid systems with different numbers of grid points: 16×16, 32×32, 64×64, and 128×128, respectively. These cases are not illustrated here. The results of Cases E to H are obtained from LMR grid systems, as shown in Fig. 7.



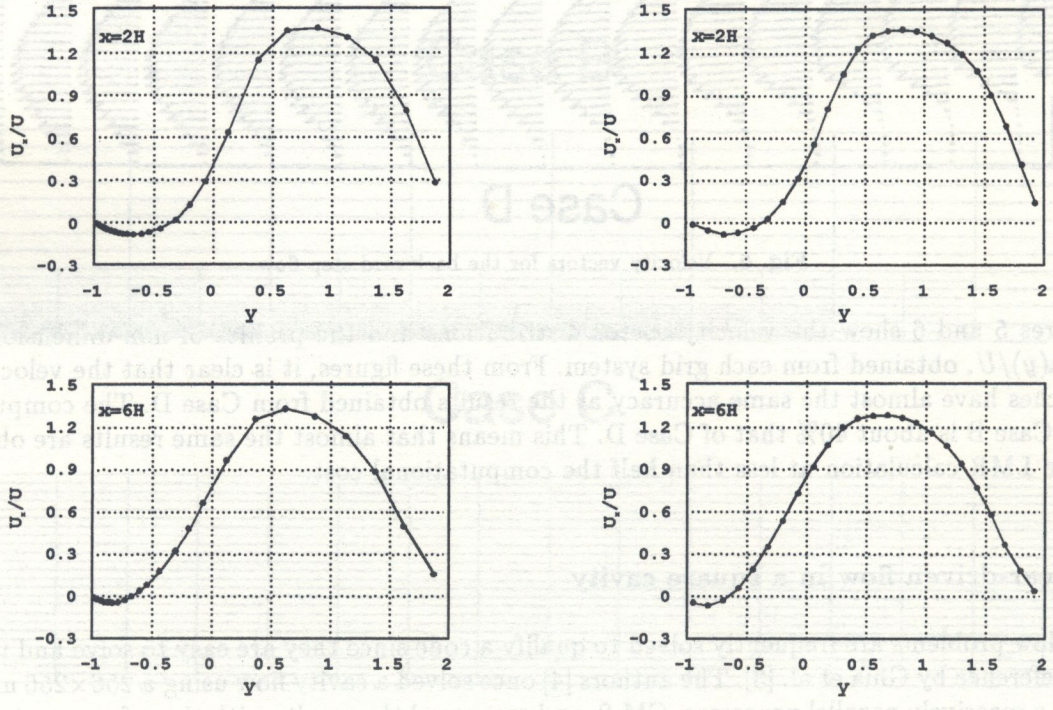


Fig. 6. Velocity profiles at $x = 2H$ and $6H$

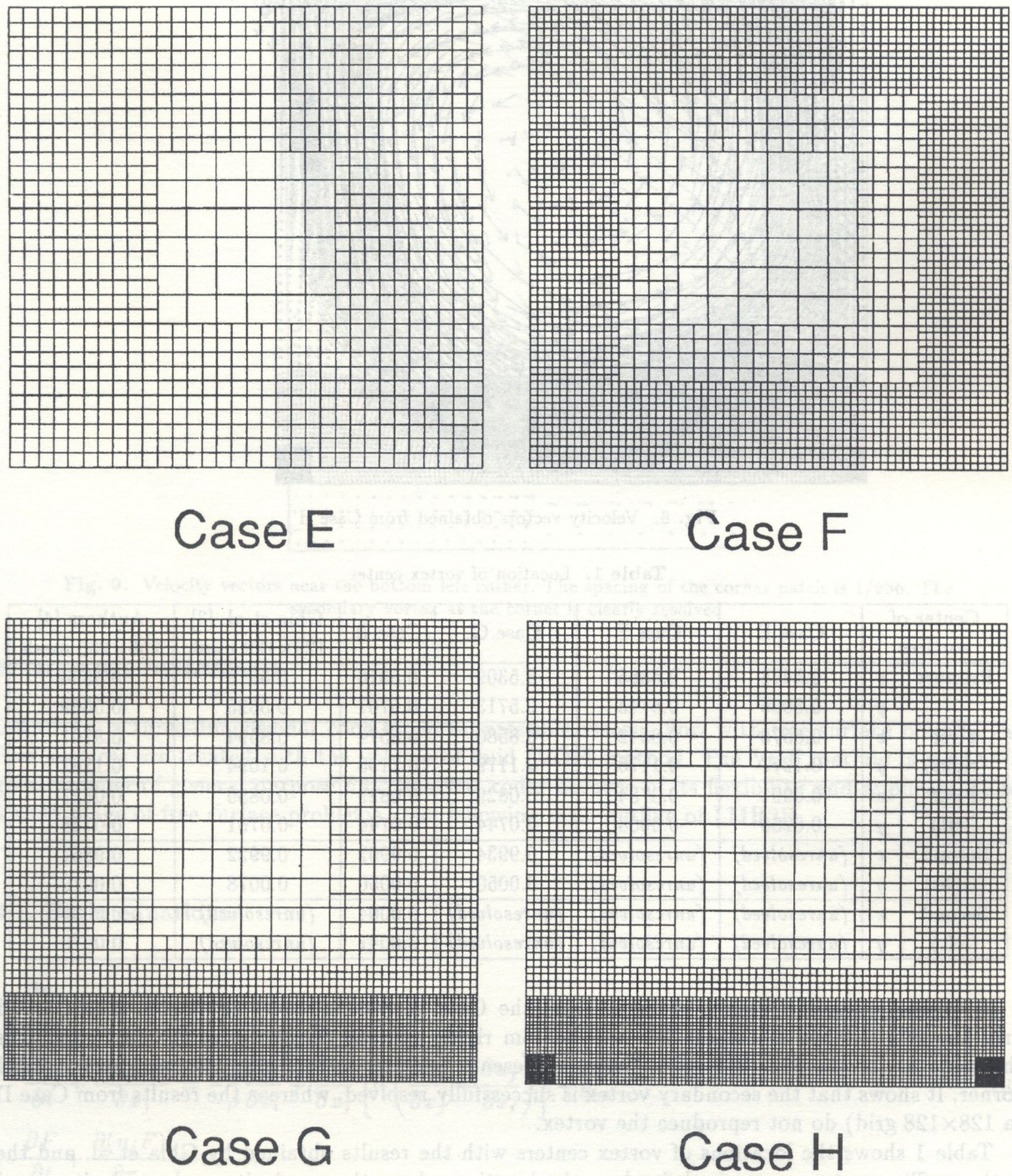


Fig. 7. LMR grids for the driven square cavity flow

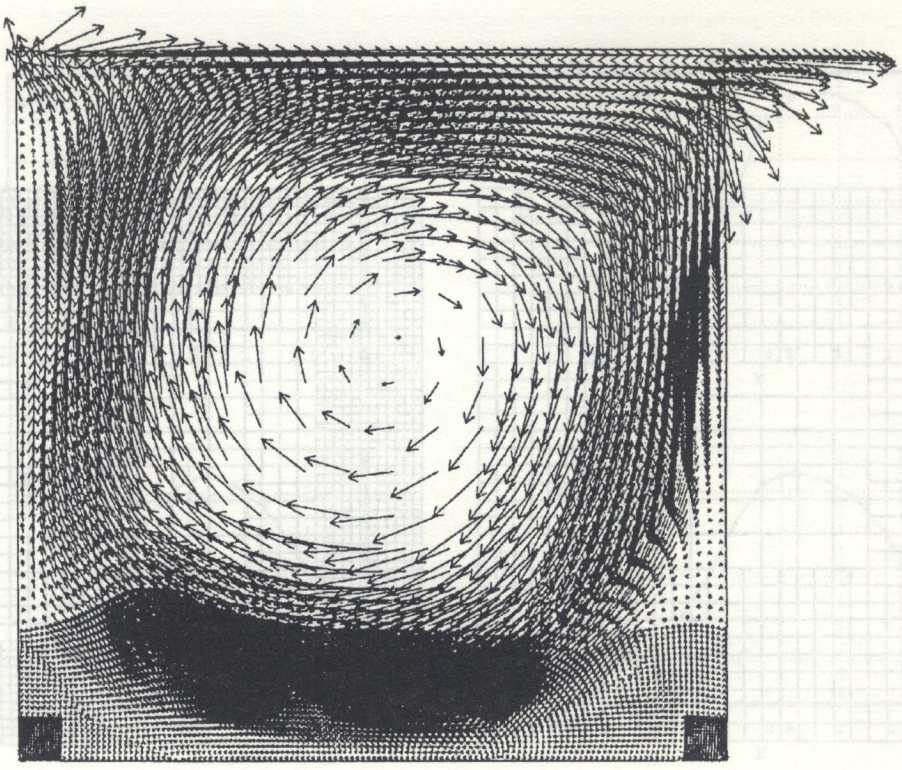


Fig. 8. Velocity vectors obtained from Case H

Table 1. Location of vortex center

Center of vortices		Case E	Case F	Case G	Case H	Ghia et al. [3] 129×129 mesh	Authors [4] 256×256 mesh
Primary	x	0.5478	0.5365	0.5309	0.5316	0.5313	0.5354
	y	0.5906	0.5743	0.5713	0.5761	0.5625	0.5669
First	x	0.8578	0.8542	0.8580	0.8574	0.8594	0.8701
(right)	y	0.1277	0.1176	0.1179	0.1195	0.1094	0.1142
First	x	0.0991	0.0784	0.0829	0.0829	0.0859	0.0827
(left)	y	0.0260	0.0608	0.0744	0.0744	0.0761	0.0748
Second	x	*(unresolved)*	*(unresolved)*	0.9934	0.9902	0.9922	0.9861
(right)	y	*(unresolved)*	*(unresolved)*	0.0050	0.0050	0.0078	0.0079
Second	x	*(unresolved)*	*(unresolved)*	*(unresolved)*	0.0004	*(unresolved)*	0.0039
(left)	y	*(unresolved)*	*(unresolved)*	*(unresolved)*	0.0047	*(unresolved)*	0.0079

A typical numerical result, obtained from the Case H grid, is shown in Fig. 8. The Case H grid has two patches at bottom left and bottom right, and the resolution of the two patches is the same as for the 256×256 grid. Figure 9 presents velocity vectors near the bottom left hand corner. It shows that the secondary vortex is successfully resolved, whereas the results from Case D (a 128×128 grid) do not reproduce the vortex.

Table 1 shows the locations of vortex centers with the results obtained by Ghia et al. and the authors. The vortex center is defined as the location where the u-velocity and v-velocity equal zero, and is calculated by linear interpolation. This may introduce some error because the reference results reported by Ghia et al. and the authors have been obtained from values of the stream function. The table, however, clearly demonstrates that accuracy is improved with the adoption of higher-level grids, even if they cover only a small region.

Figure 10 shows the u-velocity and v-velocity profiles along the vertical and horizontal center lines of the cavity. This figure also shows that the LMR technique produces good results.

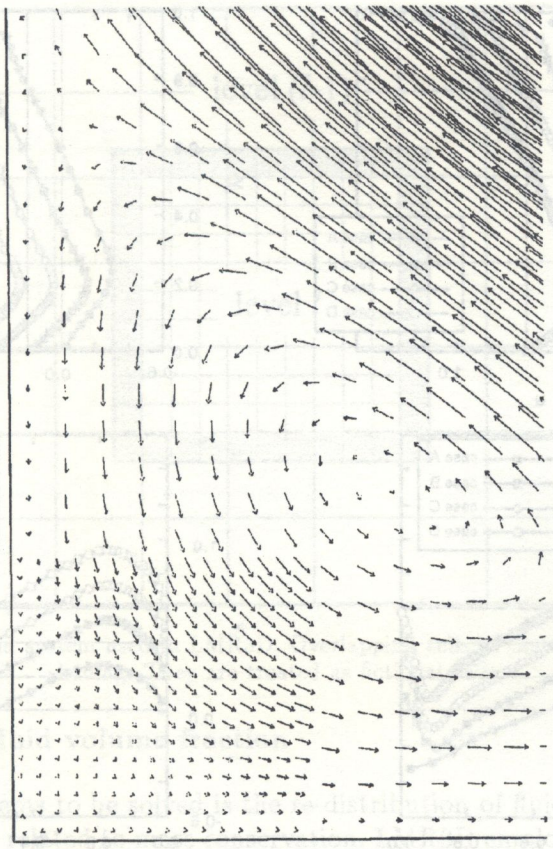


Fig. 9. Velocity vectors near the bottom left corner. The spacing of the corner patch is 1/256. The secondary vortex at the corner is clearly resolved

4. OUTLINE OF LMR3D

LMR3D is a three-dimensional code for incompressible viscous flow with free surface boundaries. Free surfaces are treated with the volume of fluid (VOF) method. The VOF method is used in a great number of general purpose fluid analysis codes because of its flexibility and applicability to a wide range of free surface problems. The following is an outline of LMR3D.

4.1. Basic equations

$$\frac{\partial u_j}{\partial x_j} = 0\,,$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\nu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] + g_i\,, \tag{3}$$

$$\frac{\partial F}{\partial t} + \frac{\partial (u_j F)}{\partial x_j} = 0\,.$$

The third equation is a kinematic equation governing a volume-fraction function, F, which specifies the fraction of volume containing fluid per unit volume. This cell-centered function assumes a value of unity in cells full of fluid, a value of zero in empty cells, and intermediate values in surface cells. The surface is located at $F = 0.5$. The kinematic equation is usually solved by means of the donor-acceptor method.

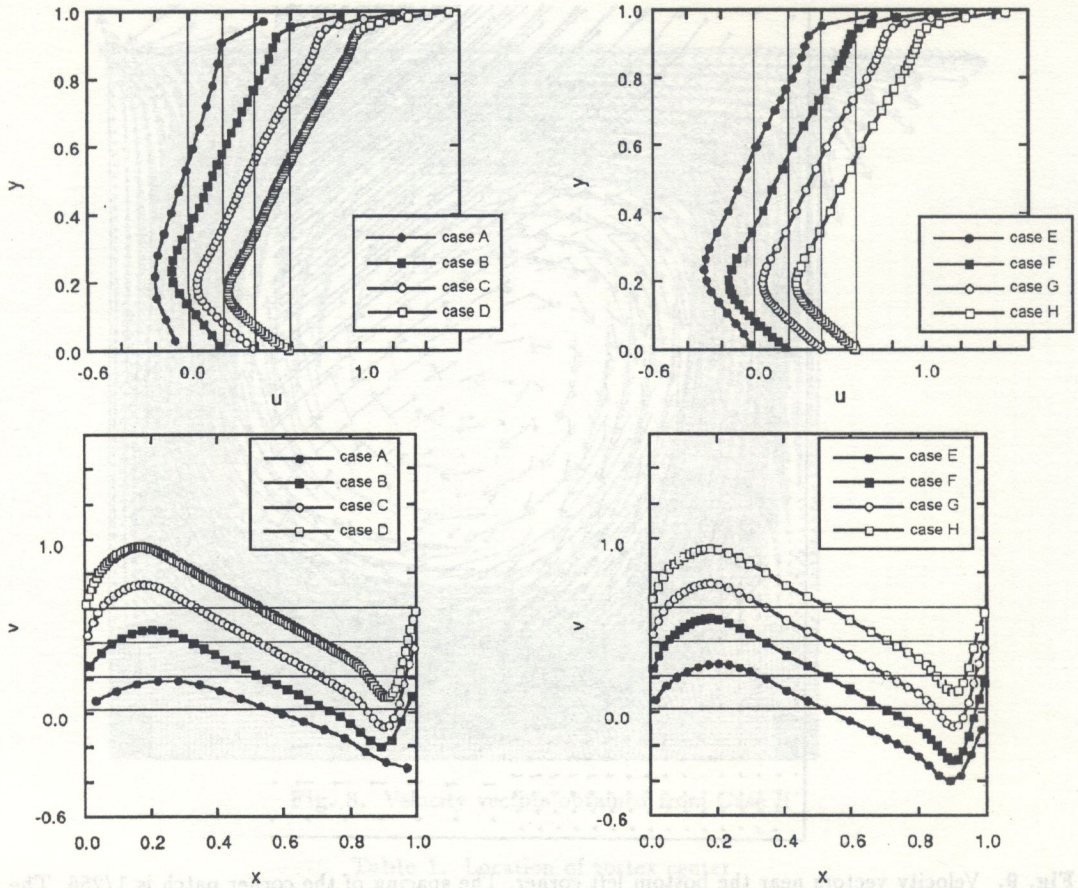


Fig. 10. The u-velocity and v-velocity along the vertical and horizontal center lines of the square cavity; left: result with fundamental grid, right: result with LMR grid

4.2. Difference scheme

LMR3D calculates fluid dynamics using a fully implicit scheme, SIMPLEST. SIMPLEST secures unconditional numerical stability, so the time-step size is not limited by the Courant number criterion when fixed boundary problems are solved. The terms in the set of basic equations are discretized by the following numerical schemes:

- in time — the 1st-order Euler scheme
- in space — the 1st-order upwind scheme or QUICK scheme

4.3. Grid system

LMR3D uses a staggered mesh arrangement in which all the vector components, u, v and w, are defined at the center of cell faces and both scalar variables, p and F, are located at the cell center. An overlapping grid system is employed to connect patches of different fineness level smoothly. Patches of level l are also presupposed to exist in the next-coarser grid patches, level $(l-1)$, for simplification. As shown in Fig. 11, the outer boundary cells of a level l grid patch overlap with a level $(l-1)$ patch. Boundary values for the grid of level l are set at the overlapping cells by interpolation and fixed during each time cycle. The highest grid level available in LMR3D is 2, although this is not essential and extension to a higher level is straightforward. The refinement ratio, r, is set to 3 because this is suitable for a staggered mesh arrangement. Then, a cell in the grid of level $(l-1)$ is divided into $3\times3\times3$ sub-cells of level l. The variable mesh approximation is also used on the zero level grid.

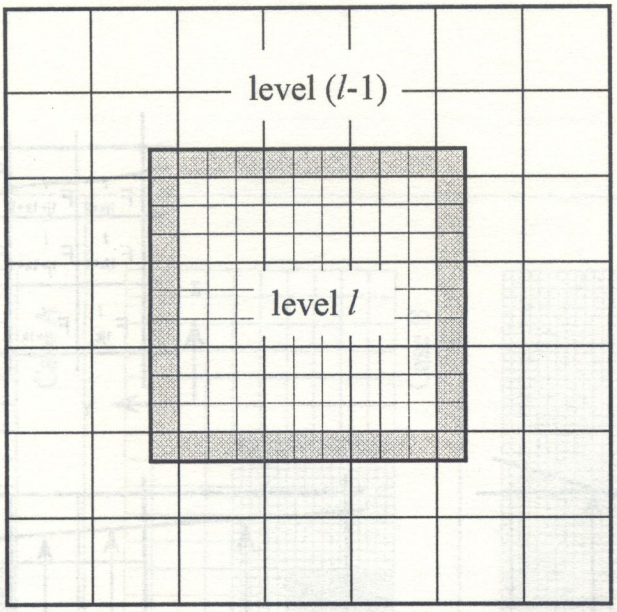


Fig. 11. An example of grid system used in LMR3D. Overlapping cells of the l level grid are indicated by shading. They are treated as fictional cells

4.4. Re-distribution of fluid volume fraction

One of the important problems to be solved is the re-distribution of fluid contained in coarser cells to sub-cells. This is directly related to mass conservation. LMR3D employs an interpolation scheme for this fluid volume fraction re-distribution. Figure 12 illustrates the interpolation method. Let the fluid in a cell (i, j, k) of level $(l-1)$ be re-distributed into sub-cells of level l, and let us assume that the surface boundary near the cell can be approximated by means of a plane cutting through the cell. Then, the height of the surface above the bottom line of the surface cell is defined at each location. After interpolating the surface heights corresponding to the sub-cells, the fluid volume fractions of the sub-cells are obtained as follows,

$$F^l_{1,j+\beta,k+\gamma} = \max\left\{0,\ \min\left[3\left(h^l_{j+\beta} - (1+\frac{\gamma}{3})\right),\ 3\right]\right\}, \tag{4}$$

where, $h^l_{j+\beta}$ is the surface height and $F^l_{1,j+\beta,k+\gamma}$ is the fluid volume fraction to be re-distributed to the sub-cell $(1, j+\beta, k+\gamma)$, as shown in the figure. For fluid mass conservation,

$$F^{l-1}_{i,j,k} = \frac{1}{27}\sum_{\alpha=0}^{2}\sum_{\beta=0}^{2}\sum_{\gamma=0}^{2} F^l_{i+\alpha,j+\beta,k+\gamma} \tag{5}$$

should hold.

5. Numerical examples

5.1. Flow around a rectangular column

A three-dimensional flow around a rectangular column without free surfaces is solved to verify LMR3D. A schematic diagram of the flow is shown in Fig. 13. Three calculations, listed in Table 2, are made to compare accuracy and computing time. Figure 14 shows the computational grid systems used in this study. The total number in the coarse grid (Case A) is 1 800, in the LMR grid (Case B) 10 548, and in the finer grid (Case C) 48 600. Figure 15 shows the flow vector distribution on the horizontal plane, Figure 16 that on the vertical plane of each case and Figure 17 compares the

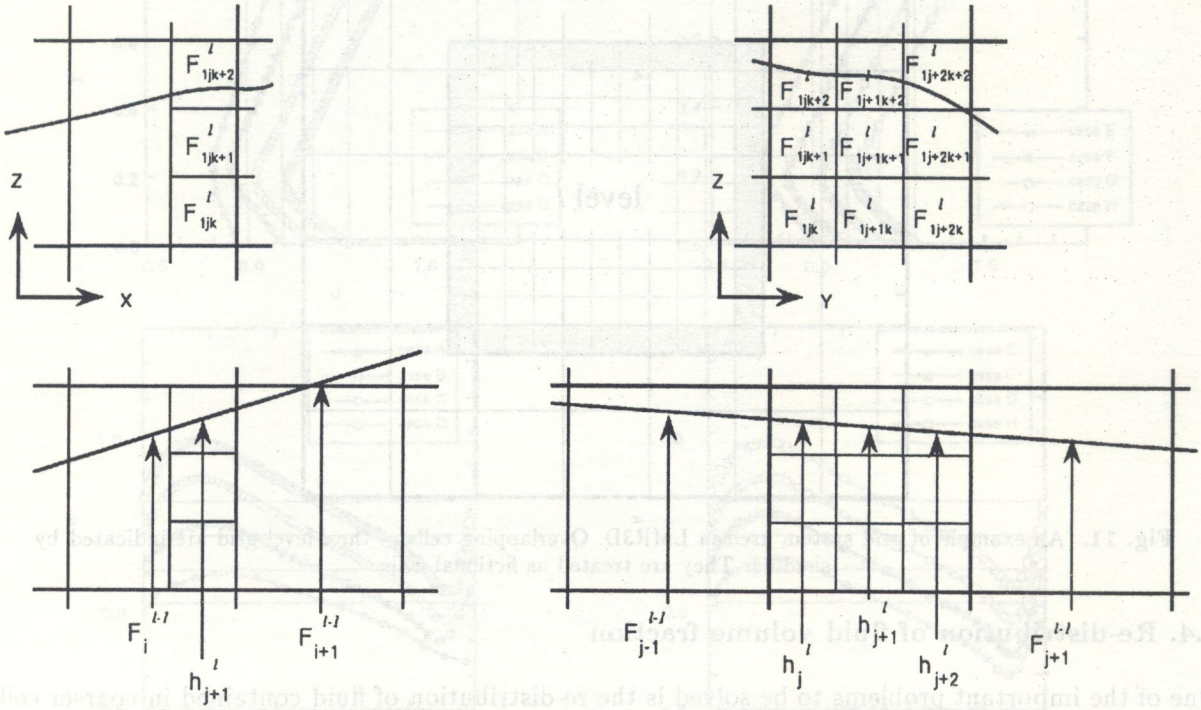


Fig. 12. Fluid re-distribution into sub-cells

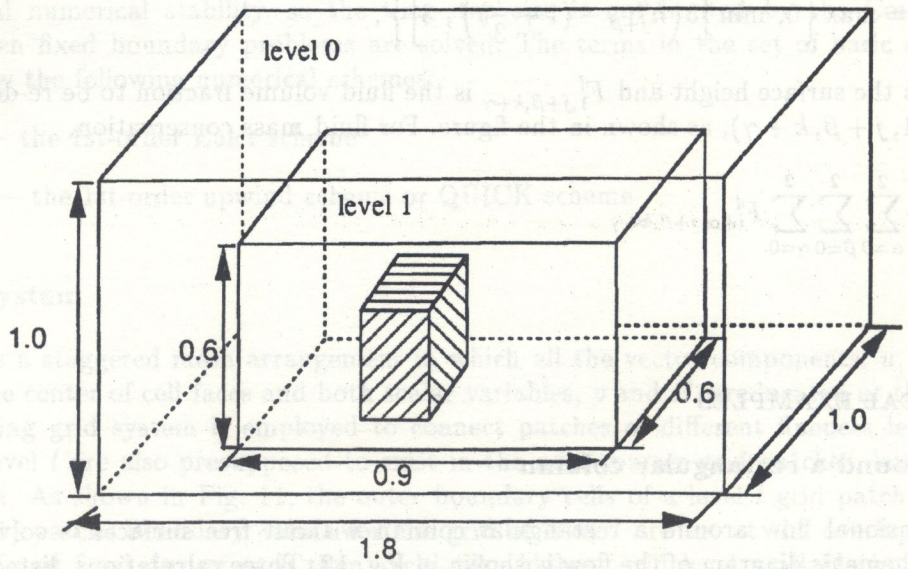


Fig. 13. Flow configuration around a rectangular column

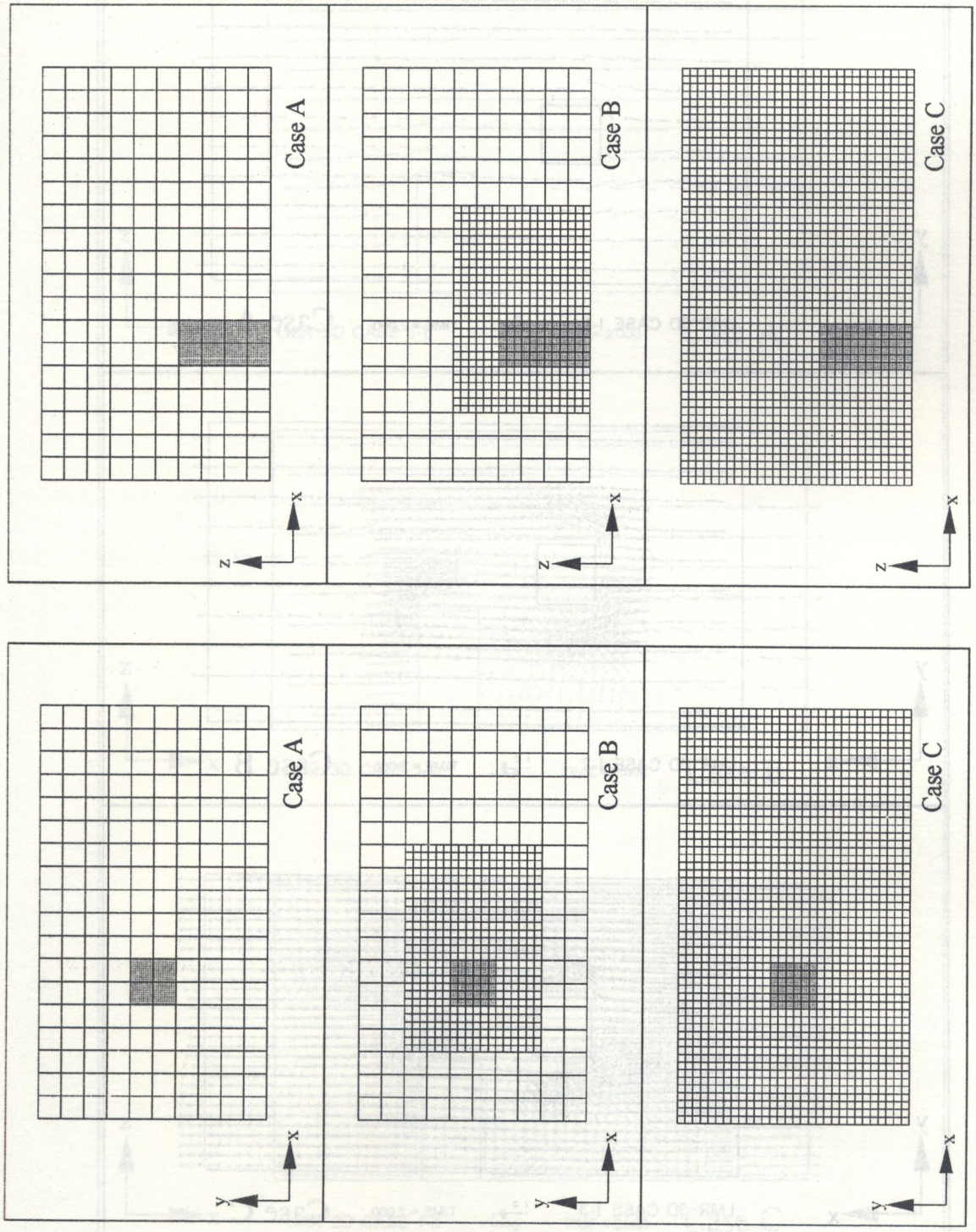


Fig. 14. Grid systems for the flow around a rectangular column; left: horizontal view, right: vertical view

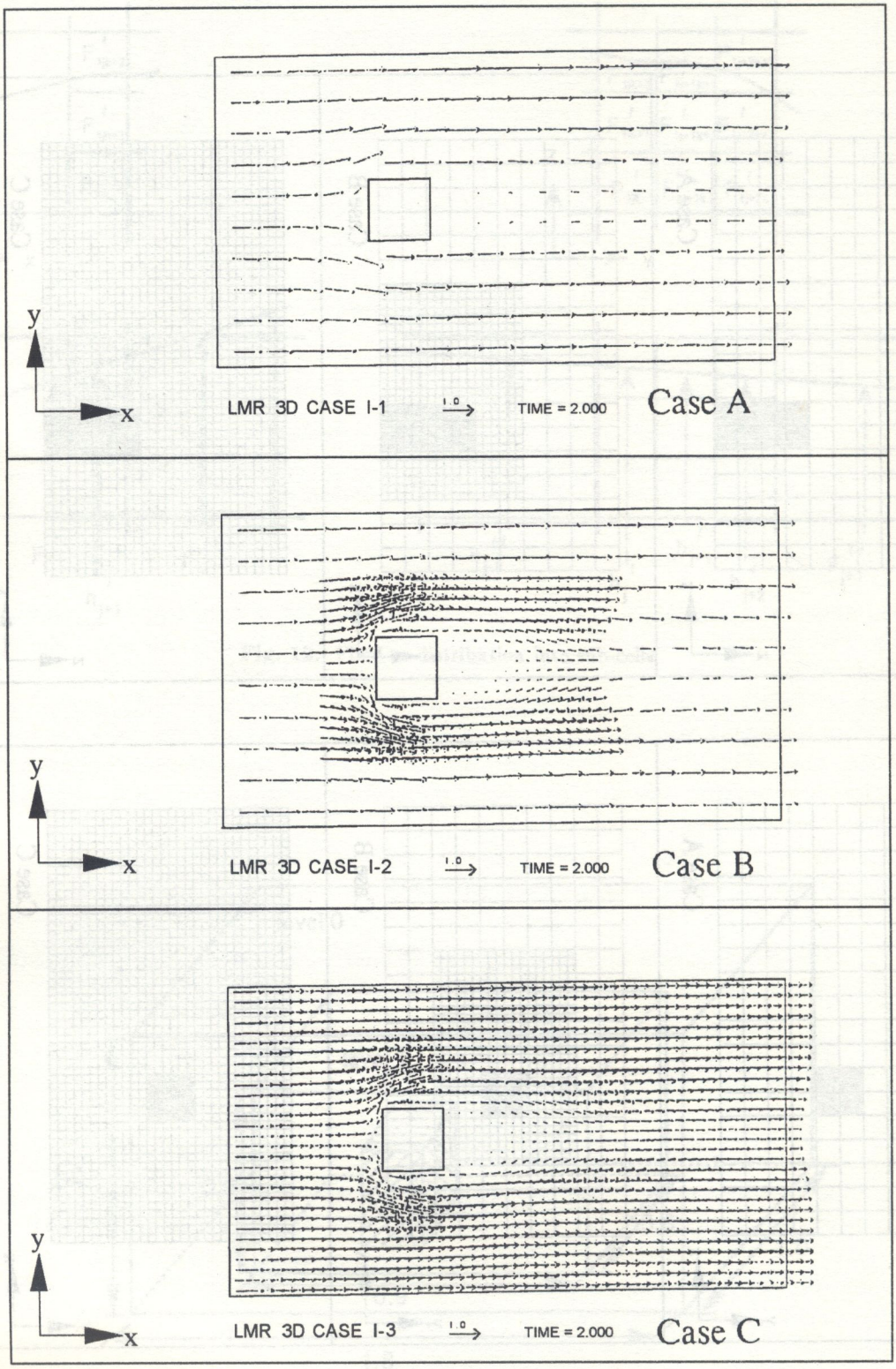


Fig. 15. Horizontal view of velocity vectors at $t = 2.0$, $z = 0.25$

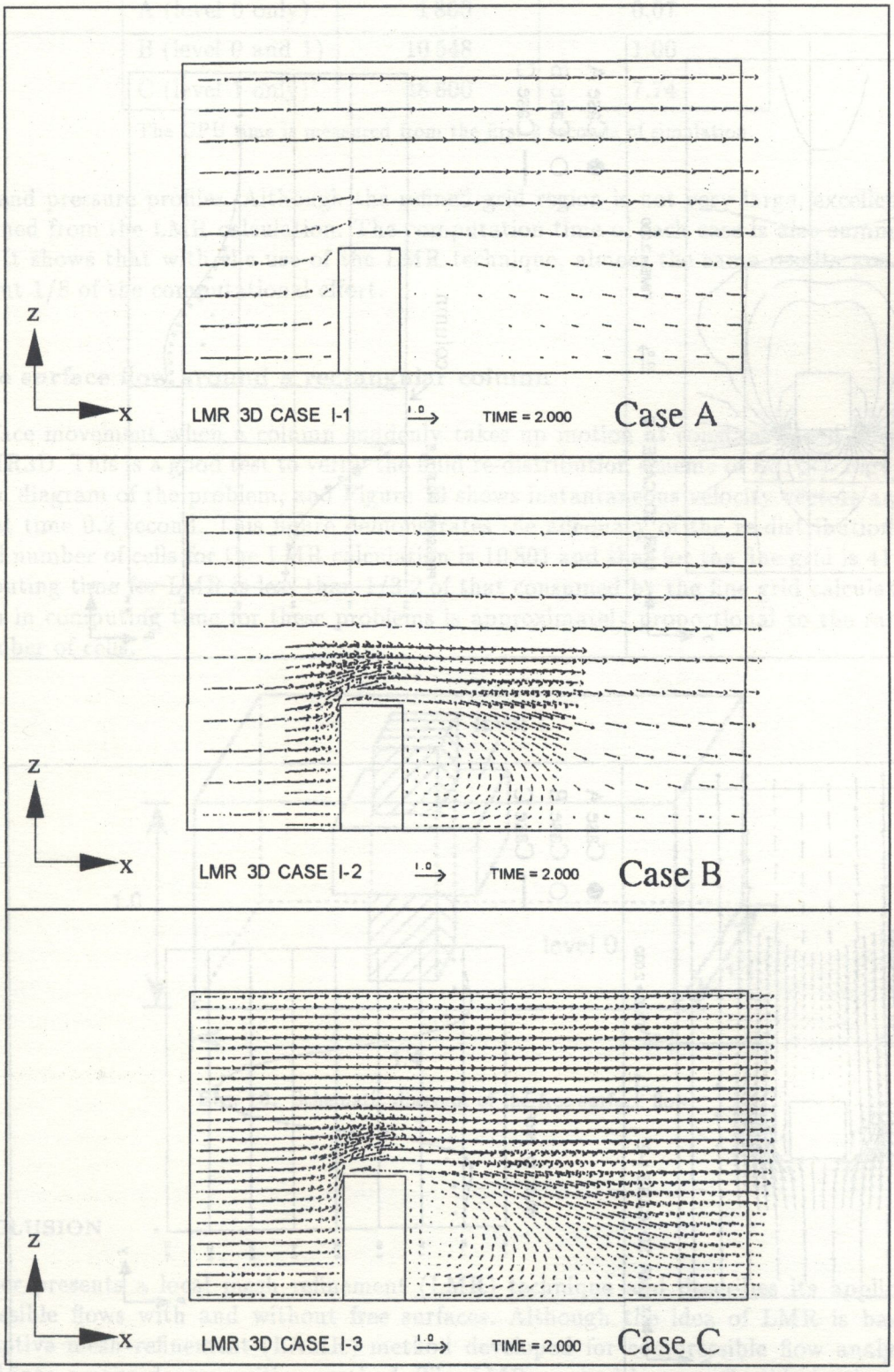


Fig. 16. Vertical view of velocity vectors at $t = 2.0$, $y = 0.45$

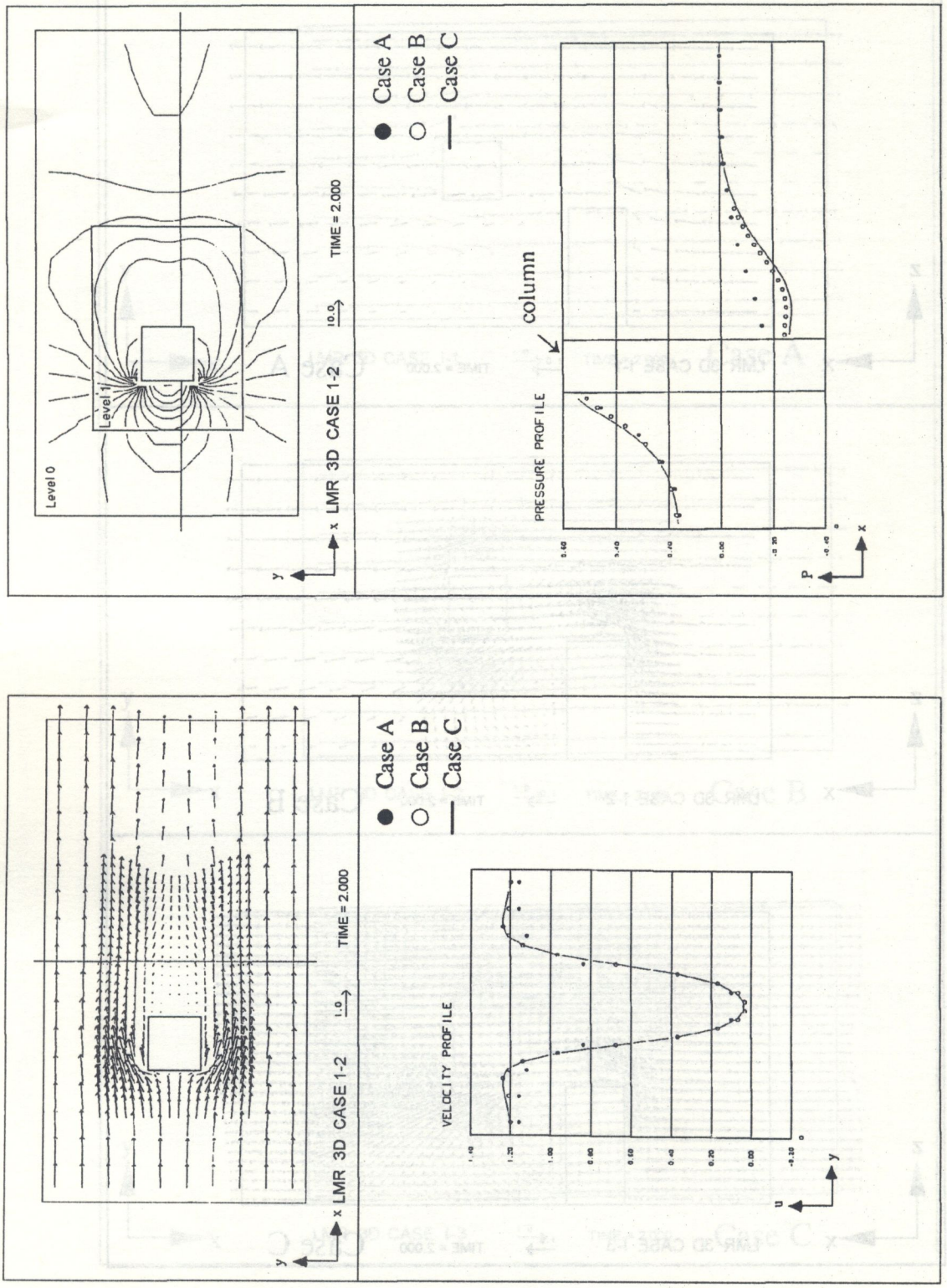


Fig. 17. Comparison of velocity and pressure profiles at time $t = 2.0$; left: U-velocity profiles at $x = 0.9$, $z = 0.25$, right: pressure profiles at $y = 0.45$, $z = 0.25$

Table 2. Numerical conditions and CPU times

Case	Number of cells	CPU time ratio
A (level 0 only)	1 800	0.07
B (level 0 and 1)	10 548	1.00
C (level 1 only)	48 600	7.74

The CPU time is measured from the first 2 seconds of simulation.

velocity and pressure profiles. Although the refined grid region is not very large, excellent results are obtained from the LMR calculation. The computation time of each case is also summarized in Table 2. It shows that with the use of the LMR technique, almost the same results are obtained with about 1/8 of the computational effort.

5.2. Free surface flow around a rectangular column

Free surface movement when a column suddenly takes up motion at constant speed is calculated using LMR3D. This is a good test to verify the fluid re-distribution scheme of Eq. (4). Figure 18 is a schematic diagram of the problem, and Figure 19 shows instantaneous velocity vectors around the column at time 0.2 second. This figure demonstrates the adequacy of the re-distribution scheme. The total number of cells for the LMR calculation is 10 801 and that for the fine grid is 41 580, and the computing time for LMR is less than 1/3.2 of that consumed by the fine grid calculation. The reduction in computing time for these problems is approximately proportional to the ratio of the total number of cells.

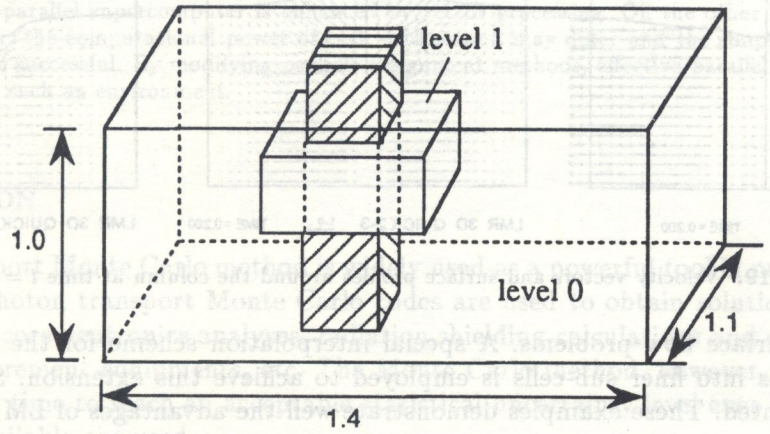


Fig. 18. Schematic diagram of the free surface flow

6. Conclusion

This paper presents a local mesh refinement (LMR) technique and describes its application to incompressible flows with and without free surfaces. Although the idea of LMR is based on a local adaptive mesh refinement (LAMR) method developed for compressible flow analysis, it is converted into a zone decomposition method. The LMR method is more flexible in the composition of computational grid systems than usual zonal methods. It can place an arbitrary number of patches with arbitrary levels of fineness in any regions of interest. This flexibility enables us to use a high-resolution grid system with a reduced number of cells in total. Therefore, more accurate solutions are obtained at a lesser computational cost. Furthermore, the LMR method can be

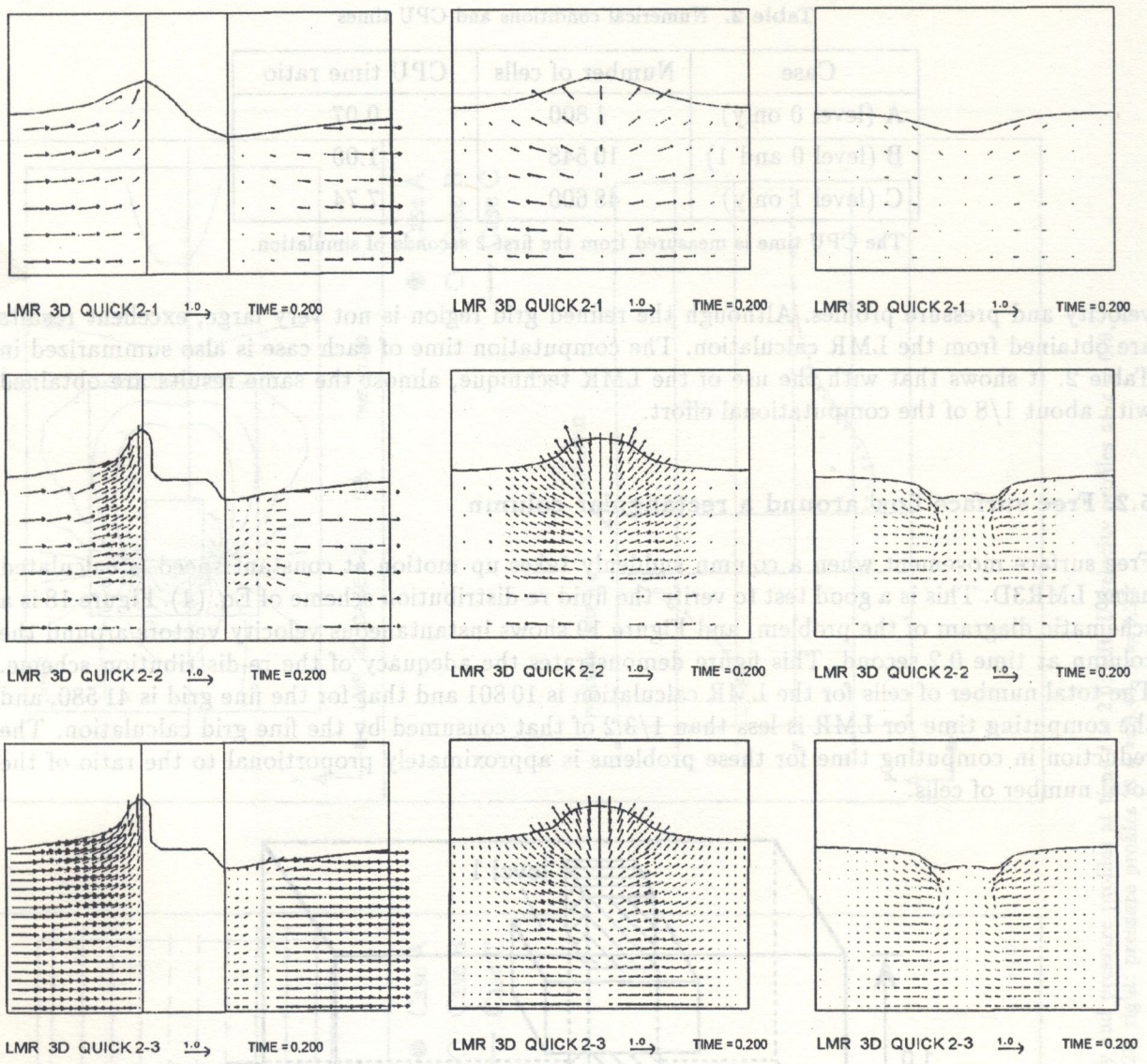


Fig. 19. Velocity vectors and surface profiles around the column at time $t = 0.2$

extended to free surface flow problems. A special interpolation scheme for the re-distribution of fluid in coarser cells into finer sub-cells is employed to achieve this extension. Several numerical examples are presented. These examples demonstrate well the advantages of LMR.

REFERENCES

[1] M.J. Berger, P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, **82**: 64–84, 1989.

[2] C.W. Hirt, B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, **39**: 201–225, 1981.

[3] U. Ghia, K.N. Ghia, C.T. Shin. High-resolutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics*, **48**: 387–411, 1982.

[4] N. Tanaka, H. Terasaka. A numerical study of incompressible viscous flow with massively parallel processors. *Computational Fluid Dynamics Journal*, **2**: 145–160, 1993.