Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6410--06-8947

Application of FCT to Incompressible Flows

Junhui Liu
Carolyn R. Kaplan

Center for Reactive Flow and Dynamical Systems
Laboratory for Computational Physics and Fluid Dynamics

David R. Mott
Elaine S. Oran

Senior Scientist for Reactive Flow Physics
Laboratory for Computational Physics and Fluid Dynamics

April 7, 2006

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 07-04-2006	**2. REPORT TYPE** Memorandum Report	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE

Application of FCT to Incompressible Flows

5a. CONTRACT NUMBER

5b. GRANT NUMBER
64-3615-0-6

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Junhui Liu, David R. Mott, Carolyn R. Kaplan, and Elaine S. Oran

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6410--06-8947

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The flux-corrected transport algorithm (FCT) has been used to solve the Navier-Stokes equation for incompressible flows on colocated grids. In the approach presented here, FCT is used to calculate either the convection terms or to integrate the entire momentum equation. This has the effect of minimizing numerical diffusion globally and adding diffusion selectively where it is required to stabilize the algorithm. Since an odd-even decoupling instability arises in standard algorithms that update a pressure correction in the Poisson equation, we have avoided this instability by using an intermediate velocity to update the full pressure, rather than the pressure correction. We have tested the new algorithm by computing a series of 2-D flows, and the numerical results agree well with either analytical solutions or other published simulation results. We have also applied this algorithm to a 3-D microchannel flow containing staggered herringbone structures to demonstrate the general capability of the method.

15. SUBJECT TERMS

Incompressible flow	Poisson equation	Channel flow	Cavity flow
Flux-corrected transport	Microflow	Boundary-layer flow	Herringbone structures

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Junhui Liu
a. REPORT Unclassified	**b. ABSTRACT** Unclassified	**c. THIS PAGE** Unclassified	UL	36	**19b. TELEPHONE NUMBER** *(include area code)* (202) 767-6590

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

Contents

APPLICATION OF FCT TO INCOMPRESSIBLE FLOWS

1. Introduction

Flux-corrected transport (FCT) algorithms [1-3] are high-order, monotone, conservative, and positivity preserving methods for solving single or coupled continuity equations with source terms. One of the major applications of FCT has been to solve the Navier-Stokes equations for unsteady, compressible flows. In this context, FCT has been applied to a wide range of problems including flows associated with shocks and shock interactions, flames and detonations, turbulent flows, acoustics, plasma interactions, and astrophysics, many of which are described in [4]. This paper presents the first step in applying FCT to incompressible flows. In the past, other nonlinear monotone algorithms developed for compressible flows were used to solve incompressible flow problems. These include the higher-order Godunov [5] and MUSCL [6] methods, which were used to calculate the convection terms in the momentum equation.

There are two major numerical instabilities that arise in standard approaches for solving incompressible flows. The first is similar to what is encountered in compressible algorithms in that it originates from the discretization of the convection terms. Upwind or upwind-biased methods are often used for convection terms to control this type of numerical instability. In this paper, we use LCPFCT [7], a standard version of FCT, to integrate either the convection terms or the whole momentum equation. The flux limiting procedure embedded in FCT helps to stabilize oscillations introduced by the discretization of convection terms.

The second numerical instability results from the pressure-correction procedure [8, 9], and introduce oscillations in the pressure field. To eliminate these pressure oscillations, incompressible algorithms often use staggered grids in which the velocities and the other variables are evaluated at different grid locations. The drawback of the staggered grid is that it can be cumbersome for nonorthogonal grids with complex geometries. Algorithms for compressible flows, including FCT, generally use colocated grids, where all of the variables are evaluated at the same locations. Over the years, several approaches have been developed

---

Manuscript approved November 14, 2005.

to control this numerical instability in the incompressible algorithms that use colocated grids. These include introducing a third-order pressure gradient [8, 9] to interface velocities or an extra source term in the Poisson equation [6, 10, 11]. In this paper, we have found that this instability results from an odd-even decoupling problem in the discretization of the pressure gradients in the source term of the Poisson equation. Pressure gradients appear in the source term because the pressure correction, rather than the pressure itself, is updated by the Poisson equation, as shown in Section 2. Therefore, we can avoid this numerical instability by updating the pressure rather than the pressure correction in the Poisson equation, so that the discretization of the Poisson equation can be free from the odd-even decoupling problem.

We have used this FCT-based method to simulate a two-dimensional (2D) microchannel flow, a 2D boundary-layer flow, and a 2D cavity driven flow. These results are compared with either analytical solutions or other published numerical results. Furthermore, a three-dimensional (3D) microchannel flow containing patterned, herringbone-shaped structures is simulated to demonstrate the general capability of this method.

2. Governing Equations and Algorithm

Incompressible flows are governed by the momentum equation and the divergence-free condition:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \nabla p - \mu \nabla^2 \mathbf{u} \; = \; 0, \tag{1}$$

$$\nabla \cdot \mathbf{u} \; = \; 0, \tag{2}$$

where ρ is the density, $\mathbf{u}$ is the velocity vector (bold font indicates a vector variable in this paper), p the pressure, and μ the viscosity.

2.1. Time Advancement

Using a fraction-step method to discretize equation (1) in time, one obtains

$$\rho \mathbf{u}^{n+1} \; = \; \rho \mathbf{u}^n + \Delta t \mathbf{H}^{n+\frac{1}{2}} - \Delta t \nabla p^{n+1}, \tag{3}$$

where

$$\mathbf{H} \; = \; -\nabla \cdot (\rho \mathbf{u}\mathbf{u}) + \mu \nabla^2 \mathbf{u} \; .$$

Here n, $n+\frac{1}{2}$, and $n+1$ denote times levels. The pressure p^{n+1} is calculated by satisfying the divergence condition $\nabla \cdot \mathbf{u}^{n+1} = 0$, and the final velocity $\mathbf{u}^{n+1}$ is obtained from equation (3).

2.2. Standard Approach for Updating Pressure

Following a usual approach, an intermediate velocity $\mathbf{u}^*$ can be written [8, 9]

$$\rho \mathbf{u}^* \; = \; \rho \mathbf{u}^n + \Delta t \mathbf{H}^{n+\frac{1}{2}} - \Delta t \nabla p^n, \tag{4}$$

and equation (3) becomes

$$\rho \mathbf{u}^{n+1} \; = \; \rho \mathbf{u}^* - \Delta t \nabla p', \tag{5}$$

where p' is the pressure-correction term and $p' = p^{n+1} - p^n$. Taking the divergence of equation (5) and applying the divergence-free condition (2) gives the Poisson equation for p',

$$\nabla^2 p' = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \tag{6}$$

The quantity $\rho \mathbf{u}^{n+1}$ can be calculated from equation (5) after the pressure field is updated by equation (6). As observed previously [12], this standard approach is subject to pressure oscillations.

2.2. Source of the Oscillatory Behavior in the Pressure Field and Its Remedies

If we substitute the intermediate velocity $\mathbf{u}^*$, evaluated from equation (4), into the Poisson equation (6) and assume the divergence-free condition is satisfied at the previous time level n, equation (6) becomes

$$\nabla^2 p' = \nabla \cdot \mathbf{H}^{n+\frac{1}{2}} - \nabla \cdot (\nabla p^n). \tag{7}$$

In this equation, a compact three-point central-difference algorithm is usually used to discretize $\nabla^2 p'$, for example, the discretization of its x-component can be,

$$\frac{\partial^2 p'_{i,j,k}}{\partial {x_i}^2} = (p'_{i+1,j,k} - 2p'_{i,j,k} + p'_{i-1,j,k})/\Delta x^2,$$

where the odd and the even points are coupled. Because of the nonlinearity of the convection terms, the discretization of $\nabla \cdot \mathbf{H}^{n+\frac{1}{2}}$ is also free from the odd-even decoupling problem. On the other hand, the odd and even points are decoupled in the discretization of $\nabla \cdot (\nabla p^n)$, because a central-difference algorithm is usually used for ∇p^n in momentum equation (4). For example, the x-component of $\nabla \cdot (\nabla p^n)$ has a dirscretization form

$$\frac{\partial^2 p^n_{i,j,k}}{\partial {x_i}^2} = (p^n_{i+2,j,k} - 2p^n_{i,j,k} + p^n_{i-2,j,k})/4\Delta x^2,$$

where the odd and the even points are decoupled. The decoupling will cause oscillations in the source term of the Poisson equation and thus in the pressure correction p'. Various approaches have been proposed in the past to remedy this problem, including filtering out the oscillations [13] or adding extra terms to either the cell face velocities or to the Poisson equation [6, 8].

Since it is the discretization of the pressure gradients in the source term of the Poisson equation (6) that introduces pressure oscillations, we can eliminate these pressure oscillations by constructing an intermediate velocity without the contribution of the pressure gradient [9], so that the pressure gradients will not be included in the source term of equation (6). For example, we can use

$$\rho \mathbf{u}^* = \rho \mathbf{u}^n + \Delta t \mathbf{H}^{n+\frac{1}{2}}, \tag{8}$$

and the Poisson equation becomes

$$\nabla^2 p^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \tag{9}$$

where the pressure rather than the pressure correction is calculated. If the intermediate velocity $\mathbf{u}^*$, evaluated from equation (8), is substituted into equation (9), the Poisson equation can be also written as

$$\nabla^2 p^{n+1} = \nabla \cdot \mathbf{H}^{n+\frac{1}{2}}. \tag{10}$$

Thus, the source term of the Poisson equation equation (9) only has the contribution from the convection terms. As we have discussed above, the discretization of convection terms does not have the odd-evening coupling problem. In addition, if the odd-even coupled algorithm

$$\frac{\partial^2 p_{i,j,k}^{n+1}}{\partial {x_i}^2} = (p_{i+1,j,k}^{n+1} - 2p_{i,j,k}^{n+1} + p_{i-1,j,k}^{n+1})/\Delta x^2$$

is applied to the operator $\nabla^2 p^{n+1}$ on the left side of equation (9), equation (9) does not have the odd-even decoupling problem and no extra steps are needed to control the oscillations in the pressure field. In this paper, LCPFCT [7] is used to obtain the intermediate velocity, and its flux limiting procedure helps to control the oscillations caused by the nonlinear convection terms. Finally, the velocity is updated by

$$\rho \mathbf{u}^{n+1} = \rho \mathbf{u}^* - \Delta t \nabla p^{n+1}. \tag{11}$$

There are cases in which the intermediate velocity may need to include pressure gradients due to the requirement of a specific algorithm, intermediate velocity including the pressure gradients, such as that defined in equation (4), can be first calculated for such occasions. The contribution of the pressure gradient is then removed from the intermediate velocity after the integration of the momentum equation, prior to the pressure update from equation (9).

Rhie and Chow [8] added third-order derivatives of pressure to the cell face values of velocity $\mathbf{u}^*$ in the Poisson equation (6) to control these pressure oscillations. These third-order derivatives result in three extra fourth-order derivatives on the right side of the Poisson equation. The x-component of these extra terms has the form:

$$(p^n_{i+2,j,k} - 4p^n_{i+1,j,k} + 6p^n_{i,j,k} - 4p^n_{i-1,j,k} + p^n_{i-2,j,k})/4\Delta x^2,$$

which is the difference between the odd-even uncoupled algorithm,

$$(p^n_{i+2,j,k} - 2p^n_{i,j,k} + p^n_{i-2,j,k})/4\Delta x^2 \tag{12}$$

and the odd-even coupled algorithm,

$$(p^n_{i+1,j,k} - 2p^n_{i,j,k} + p^n_{i-1,j,k})/\Delta x^2. \tag{13}$$

The inclusion of these fourth-order derivatives changes the discretization of $\nabla \cdot (\nabla p^n)$ (in the source term of equation (7)) from the odd-even uncoupled algorithm (12) to the odd-even coupled form (13), and thus eliminates the odd-even decoupling problem. If this modified $\nabla \cdot (\nabla p^n)$ is moved to the left side of the Poisson equation (7), the resulting equation would be the same as that shown in equation (9). Therefore, the approach proposed by Rhie and Chow is equivalent to updating the pressure rather than the pressure correction in the Poisson equation, such as that done in equation (9). Using equation (9) is more straightforward and does not need to deal with an extra variable p' and the specification of its boundary conditions.

For some cases in which pressure gradient has a large temporal variation, the contribution of pressure gradients to the velocity field $\mathbf{u}^{n+1}$ in equation (11) may alter the positivity and monotonicity properties and degrade the stability condition. One way to remedy this problem is to apply the FCT flux limiter to equation (11). Since there is no convection term in equation (11), the convection velocity can be assumed to be zero. We have noticed that applying the flux limiter to equation (11) improves the stability.

3. Numerical Results

A series of 2D flows were used to test this FCT-based algorithm. These include a channel flow, a boundary-layer flow and a lid-driven cavity flow. We select these 2D problems due to the availability of analytical solutions and published data. The conditions describing these simulations are given in Table 1. A 3D herringbone-mixer flow is also simulated to demonstrate the capability of this approach.

The Gauss-Seidel method is used to solve the Poisson equation (9). Because of the pressure-velocity relationship shown in equation (11), a zero-pressure gradient is used for the pressure boundary condition, and the intermediate velocity at boundary intefaces is replaced by the real velocity. Although the test problems used here are steady flows, the FCT-based algorithm is time accurate with an overall accuracy close to second order. The mask coefficient that controls the global numerical diffusion is set to 1.0 in all simulations to minimize numerical diffusion [3].

3.1. 2D Channel Flow

The first row of Table I gives the simulation parameters, and Figure 1 shows the simulation set-up. The channel height is $200\,\mu m$ and the length is $1600\,\mu m$. We use a small Reynolds number to avoid a long entrance region. The velocity and pressure distributions in Figure 2 show that the flow field is fully developed at the end of the channel. This is further confirmed by Figure 3a, where the velocity profile agrees well with the analytical solution of a fully developed channel flow. The pressure profile shown in Figure 3b has a large drop at the entrance region and gradually reaches the linear profile further down the channel. To demonstrate the effect of the odd-even decoupling problem in the source term of the Poisson equation, we have also used the Poisson equation (6) to compute the pressure. Pressure oscillations are observed in Figure 4, which is consistent with the analysis in Section 2.

3.2. 2D Boundary-Layer Flow

Figure 5 shows the computational set-up, and the second row of Table 1 describes simulation parameters. A flat plate with a length of $45\,cm$ is placed $5\,cm$ away from the inflow plane. A slip boundary condition is used for the section between the inflow plane and the plate. The Reynolds number at the end of the plate is 3×10^5, which assures that the flow is laminar. A velocity profile near the end of the plate and the skin friction along the wall are shown in Figure 6. The agreement between the numerical results and analytical solutions is excellent in regions where boundary-layer theory applies. This FCT-based algorithm is capable of simulating flows with much higher Reynolds numbers, but such simulations would create turbulence conditions that require either much finer resolution or turbulence models to give correct results.

3.3. 2D Lid-Driven Cavity Flow

There is only one dominant velocity gradient in both channel and boundary-layer flows. A cavity flow, on the other hand, has significant velocity gradients in all directions. Figure 7 shows the set-up for a lid-driven cavity flow, and the descriptive parameters are given in the third row of Table 1. The Reynolds number is 1000, and the velocity distribution is shown in Figure 8. The x-component of the velocity along the vertical line at $x = L/2$ and the y-component along the horizontal line at $y = L/2$ are compared with those predicted by Ghia *et al.* [14] As in the first two test problems, the agreement is very good.

3.4. 3D Herringbone-Mixer Flow

The three simulations shown above have demonstrated that the FCT-based algorithm is capable of simulating relative simple, idealized 2D incompressible flows. Here, it is applied to a 3D microchannel flow containing staggered, herringbone structures similar to those used in the experiment by Stroock *et al.* [15] and simulated by Kaplan *et al.* [16]. Table

2 summarizes the computing parameters, and Figure 10 presents the geometry and the boundary conditions. There are 24 herringbone structures placed in a rectangular channel with a dimension of $4000\,\mu m \times 90\,\mu m \times 200\,\mu m$ and a grid dimension of $800 \times 18 \times 40$. The individual herringbones are $20\,\mu m$ high in the y direction, and spaced $150\,\mu m$ apart in the x direction. The channel contains two sequences of herringbone structures, and each structure consists of six herringbones with the short segment on the left and the long segment on the right, followed by six more herringbones with short segment on the right and the long segment on the left. The ratio of the length of the long segment to short segment is 2:1. No-slip boundary conditions are used for the herringbone surfaces. The incoming velocity is $U_\infty = 1.0\,cm/s$. Because we are using a Cartesian grid and the herringbones are set at angles to the grid, a stair-cased grid is used for the surface of the herringbone structures, as shown in Figure 11. This type of grid arrangement has effect of using walls with rough surfaces. The height of the roughness is 5% of the height of the herringbone sub-channel for this grid size. The herringbone sub-channel is the space between neighboring herringbone segments. Since the velocity is small and the hydraulic Reynolds number of the channel is only 0.89, the effect of this surface roughness can be ignored.

Figure 12 shows the transverse velocity field around the herringbone segments. The herringbones deflect the flow and introduce rotational motions. The magnitude of the transverse velocity can be as high as 40% of the incoming velocity. Figures 13a and 13b show the velocity vector field (u, w) at $y = 17.5\,\mu m$, a plane intersecting the structure, and $y = 22.5\,\mu m$, a plane barely above the structures. The vector fields of these two planes show how the herringbones distort the flow. Figure 14a and 14b present the velocity field (u, v) and the x-component of the velocity at $z = 97.5\,\mu m$. These figures show that flow passes those structures in a way similar to a cavity flow.

Figure 15 shows the three components of the velocity at $y = 45\,\mu m$, which is $25\,\mu m$ above the structures. The herringbone structures introduce large flow variations even in regions that are far above these structures. Similar to what Kaplan *et al.* have observed, the flow pattern in each herringbone sequence are essentially the same. This is confirmed by Figure 16, where the transverse velocity fields at two identical locations in the first and the second sequences are shown. This trend does not change with the magnitude of the incoming velocity at least up to $100\,cm/s$, as shown in Figures 17 and 18, which show results from another two simulations with $U_\infty = 10\,cm/s$ and $100\,cm/s$, respectively. This similarity may allow periodic boundary conditions to be used in the x direction when multiple herringbone sequences are used, thus, reducing the computational effort in future studies [16]. Furthermore, Figures 16-18 show that the ratio of the transverse velocity to the incoming velocity decreases as the incoming velocity increases.

4. Summary

A FCT-based algorithm has been developed to simulate incompressible flows. The flux limiting procedure embedded in FCT helps to control the numerical instability introduced by the convection terms. Another type of oscillation is associated with the pressure calculation. This type of oscillation is introduced by the odd-even decoupling in the source term of the Poisson equation when the pressure correction, rather than pressure itself, is updated by Poisson equation. An algorithm that constructs an intermediate velocity without the contribution of the pressure gradient is used to remove this odd-even decoupling problem. This approach is more straightforward than the standard approach of updating pressure correction, since there is no extra effort needed to control the pressure oscillation.

This FCT-based algorithm has been tested by computing a series of 2D flows (a channel flow, a boundary-layer flow, and a lid-driven flow) and a 3D microchannel flow containing staggered herringbone structures. The numerical results agree well with either analytical solutions or other published simulation results. The herringbone structures in the 3D flow introduce a transverse velocity with a magnitude of up to 40% of the incoming velocity. Periodicity of the flow field is found among the herringbone sequences, and it could be used to reduce the computational effort in future studies.

5. Acknowledgments

This work was sponsored by the Office of Naval Research, and computational resources were provided by the Laboratory for Computational Physics and Fluid Dynamics at NRL.

References

[1] J.P. Boris and D.L. Book, Flux-Corrected Transport I: SHASTA A Fluid Transport Algorithm That Works, J. Comput. Phys. 11 (1973) 38.

[2] J.P. Boris, D.L. Book, Solution of the continuity equation by the method of flux-corrected transport, Methods in Computational Physics 16 (1976) 85.

[3] J. Liu, E.S. Oran, C.R. Kaplan, Numerical diffusion in the FCT algorithm, revisited, J. Comput. Phys. 208 (2005) 416.

[4] E.S. Oran and J.P. Boris, Numerical Simulation of Reactive Flow, Cambridge University Press, 2001.

[5] J. B. Bell, P. Colella, A Second-Order Projection Method for the Incompressible Navier-Stokes Equations, J. Comput. Phys. 85 (1989) 257.

[6] J. Waltz, An Implicit Finite Element Fractional Step Scheme for Incompressible Flows, *AIAA* Paper 2002-0431, 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada (2003).

[7] J.P. Boris, A.M. Landsberg, E.S. Oran, J.H. Gardner, LCPFCT – A Flux Corrected Transport Algorithm for Solving Generalized Continuity Equations, NRL Memorandum Report 6410-93-7192, Naval Research Laboratory, Washington, DC, 1993.

[8] C. M. Rhie, W. L. Chow, Numerical Study of the Turbulent Flow Past an Airfoil with Trailing Edge Separation, *AIAA* Journal, 21(11), 1525-1532 (1983).

[9] J.H. Ferziger, M. Peric, Computational Methods for Fluid Dynamcis, 3rd. Edition, Springer, Berlin, 2002.

[10] R. Lohner, C. Yang, E. Onate, S. Idelssohn, An unstructured grid-based, parallel free surface solver, Applied Numerical Mathematics 31 (1999) 271-293.

[11] O. Soto, R. Lohner, J. Cebral, An Implicit Monolithic Time Accurate Finite Element Scheme for Incompressible Flow Problems, *AIAA* paper 2001-2616, 15th AIAA Computational Fluid Dynamics Conference, Anaheim, California, 2001.

[12] S. J. Rubin, Numerical Studies of Incompressible Viscous Flow in a Driven Cavity, NASA SP-378, 1975.

[13] Van der Wijingaart, R.J.F. (1990), Composite Grid Techniques and Adaptive Mesh Refinement in Computational Fluid Dynamics, Report CLaSSiC-90-07, Dept. Computer Science, Stanford Univ.

[14] U. Ghia, K.N. Ghia, and C.T. Shin, High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method. J. Comput. Phys. 48 (1982) 387.

[15] Stroock, A.D., Dertinger, S.K.W., Ajdari , A. , Mezic, I., Stone, H.A. & Whitesides, G.M. 2002 Chaotic mixer for microchannels , Science 295 , 647651.

[16] C.R. Kaplan, D.R. Mott, E.S. Oran, Towards the Design of Efficient Micromixers, *AIAA* Paper No. 2004-0931, American Institute of Aeronautics and Astronautics, Reston, VA, 2004.

Table 1

Parameters for 2D simulations

Cases	Δx	Δy	Re_L	CFL	U_∞
Channel Flow	40μm	5μm	713	0.3	500 cm/s
Boundary-Layer Flow	0.5μm	0.02μm	3×10^5	0.3	4000 cm/s
Cavity Flow	0.01μm	0.01μm	1000	0.3	10 cm/s

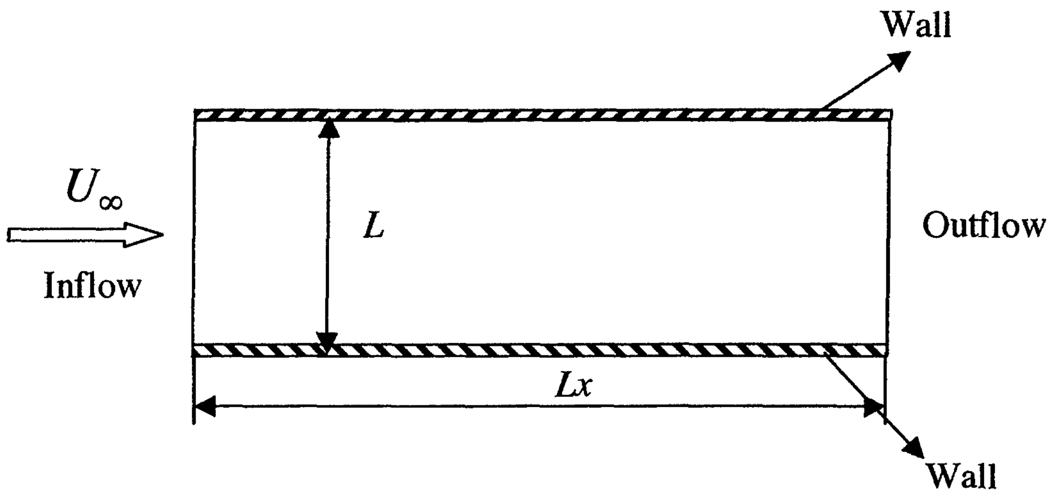

Figure 1. Schematic diagram of a 2D channel flow. U_∞ is the inflow velocity. Inflow boundary: subsonic inflow conditions. Outflow boundary: subsonic outflow conditions. Wall boundary: adiabatic and no-slip conditions.

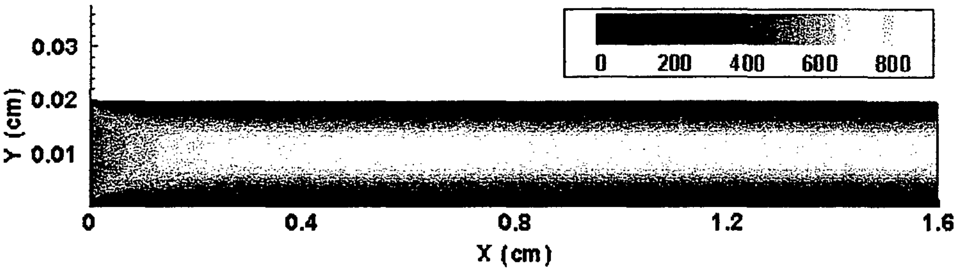

Figure 2a. Distribution of the streamwise velocity in a 2D channel.

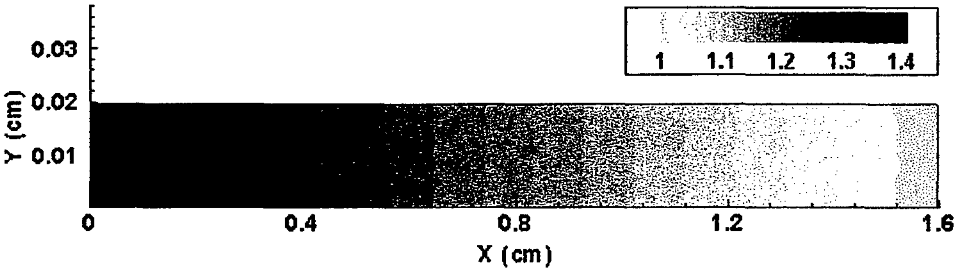

Figure 2b. Pressure distribution (scaled by p_{exit}) in a 2D channel.

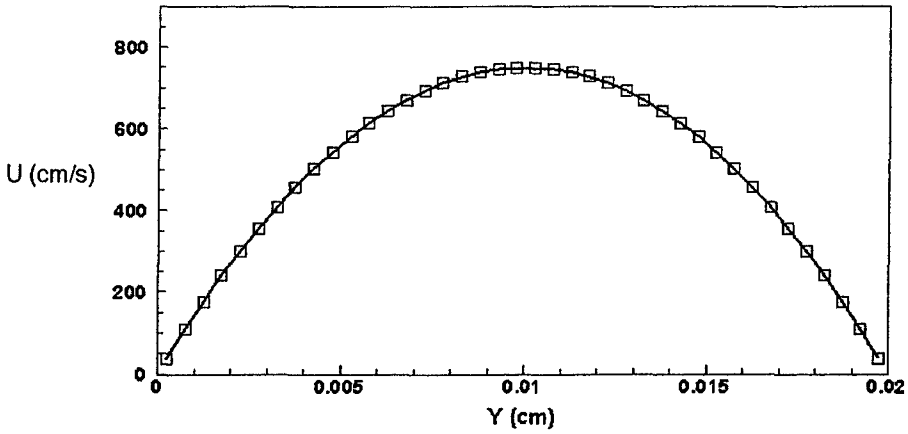

Figure 3a. Velocity profiles at the end of a 2D channel. Solid line: simulation result. Symbols: parabolic profile.

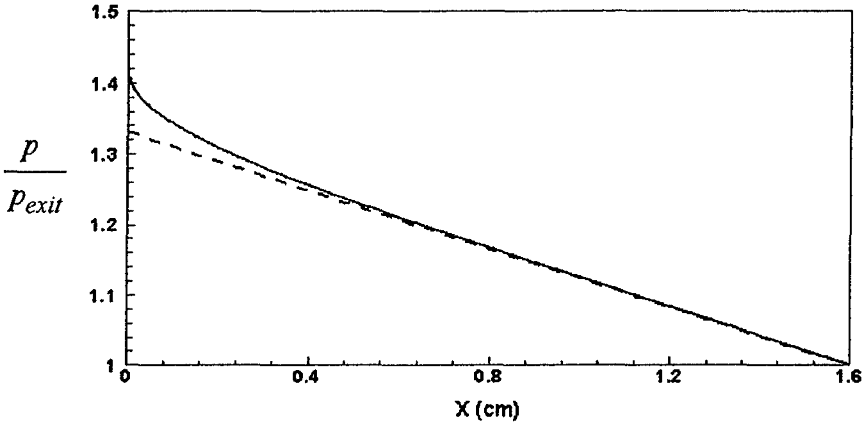

Figure 3b. Pressure distribution at the center of a 2D channel. Solid line: simulation result. Dashed line: analytical solution.

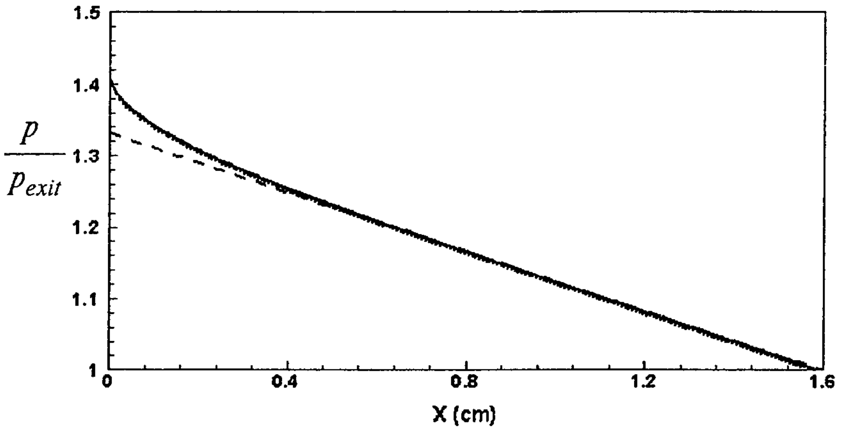

Figure 4a. Pressure distribution at the center of a 2D channel when pressure gradient is included in the intermediate velocity.

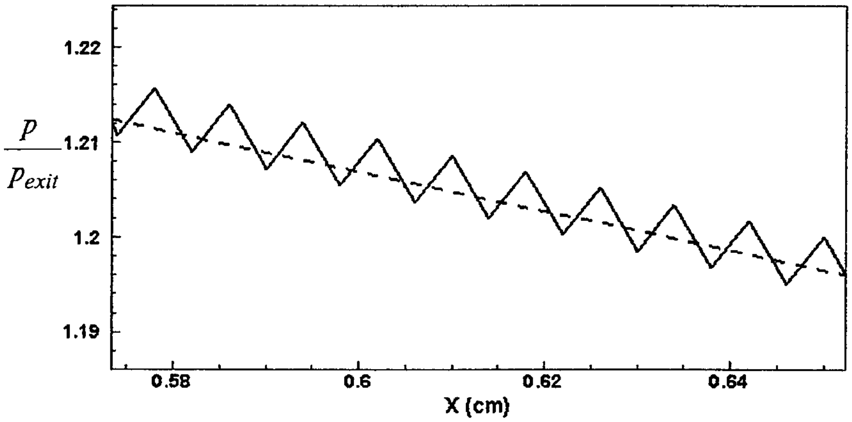

Figure 4b. Enlargement of Figure 4a.

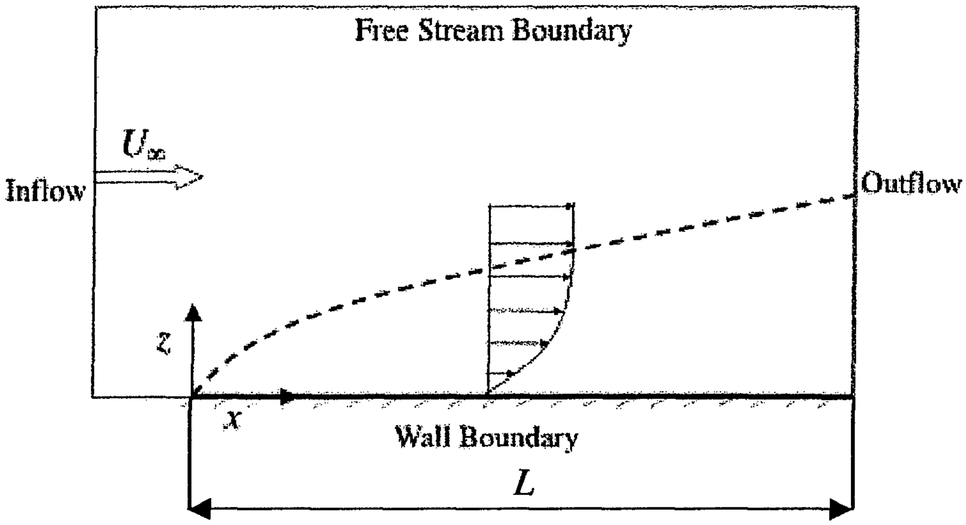

Figure 5. The schematic diagram of a boundary-layer flow on a flat plate. U_∞ is the inflow velocity, and L is the length of the plate. Inflow boundary: subsonic inflow conditions. Outflow boundary: subsonic outflow conditions. Wall boundary: adiabatic and no-slip conditions. Free stream boundary: far-field conditions.

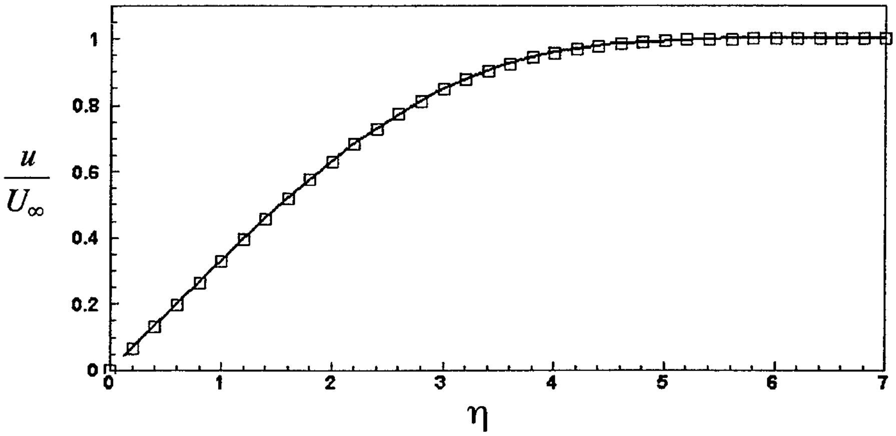

Figure 6a. Normalized velocity profiles as a function of the dimensionless coordinate η, $\eta = y\sqrt{\frac{U_\infty}{\nu x}}$, at $x = 40.5\mathrm{cm}$. Solid line: simulation result. Symbols: solution of boundary-layer theory.

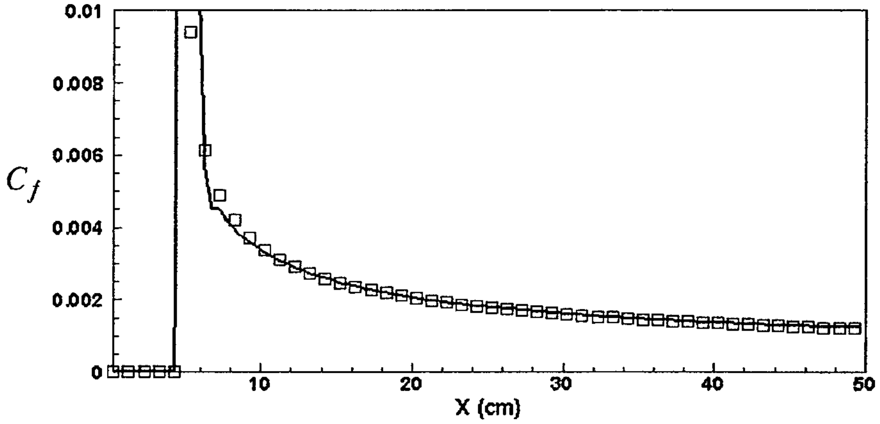

Figure 6b. Skin friction as a function of the streamwise coordinate x in the boundary-layer simulation. Solid line: simulation result. Symbols: solution of boundary-layer theory.

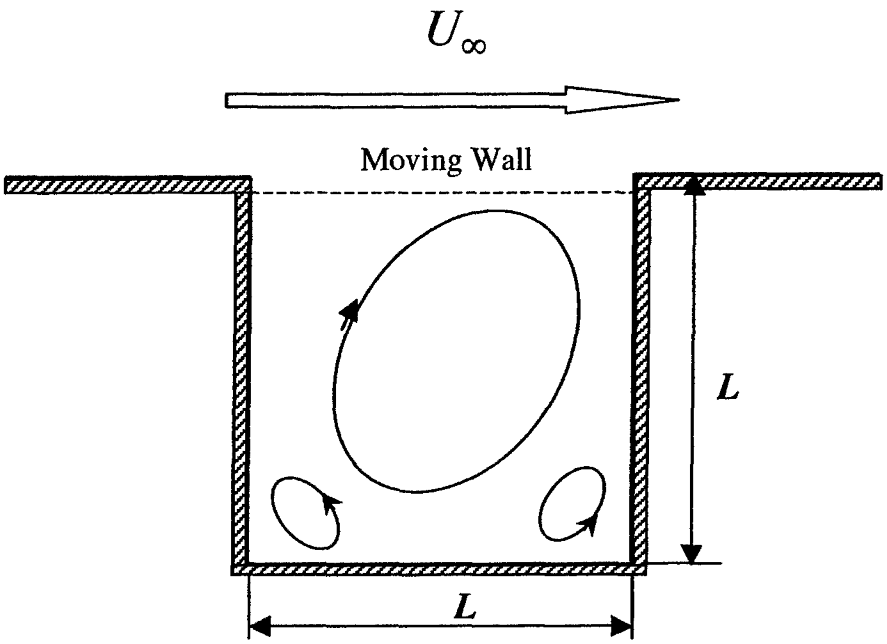

Figure 7. Schematic diagram of a lid-driven cavity flow. U_∞ is the velocity of the top moving wall, and L is the height and the width of the cavity. No-slip boundary conditions are used for the three stationary walls, and the slip condition is applied to the top moving wall.

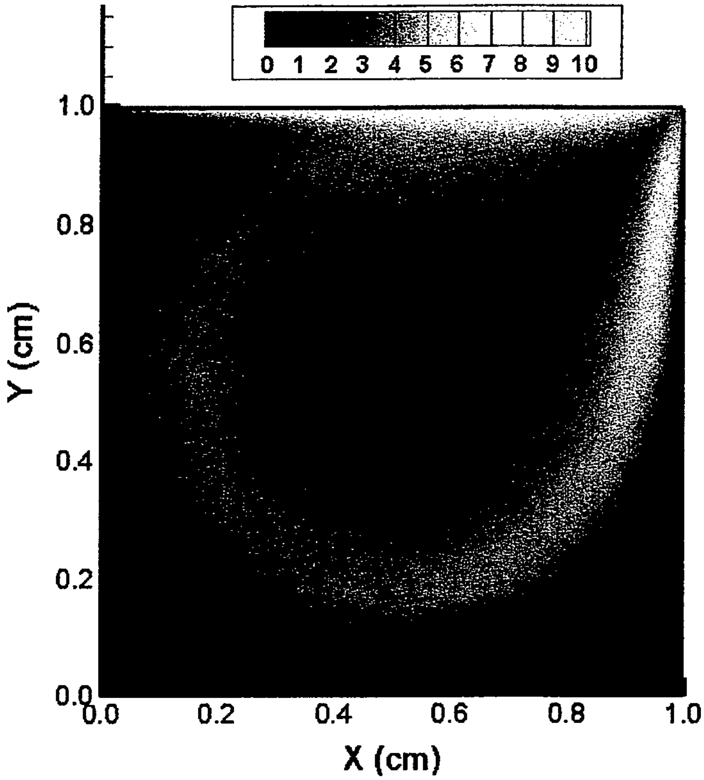

Figure 8. Distribution of velocity magnitude, $\sqrt{u^2 + v^2}$, in a lid-driven cavity flow with the Reynolds number of 1000.

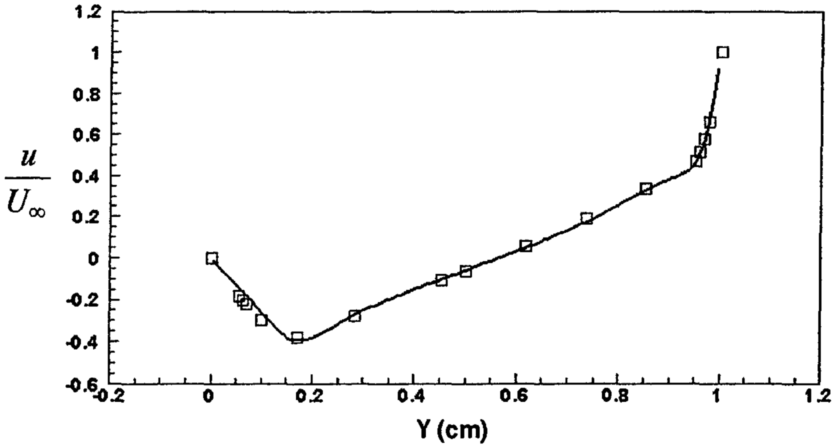

Figure 9a. Vertical profile of the x-component of the velocity at the centerline of the cavity box. Solid line: the present simulation result. Symbols: simulation by Ghia *et al.* [14].

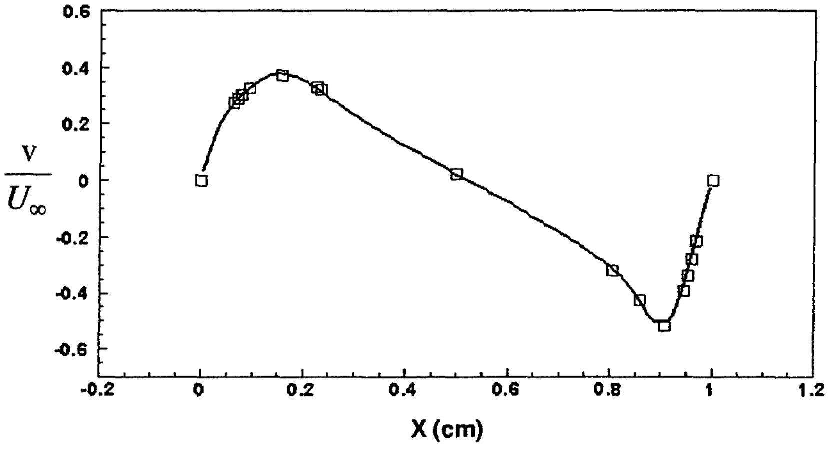

Figure 9b. Horizontal profile of the y-component of the velocity at the centerline of the cavity box. Solid line: the present simulation results. Symbols: simulation by Ghia *et al.* [14].

Table 2
Simulation parameters for a microchannel flow containing herringbone structures

$\Delta x = \Delta y = \Delta z$	$Lx \times Ly \times Lz$	ν	$\nu \frac{\Delta t}{\Delta x^2}$	U_∞
$5 \mu m$	$4000 \mu m \times 90 \mu m \times 200 \mu m$	$0.014\ (cm^2/s)$	0.05	$1.0\ cm/s$

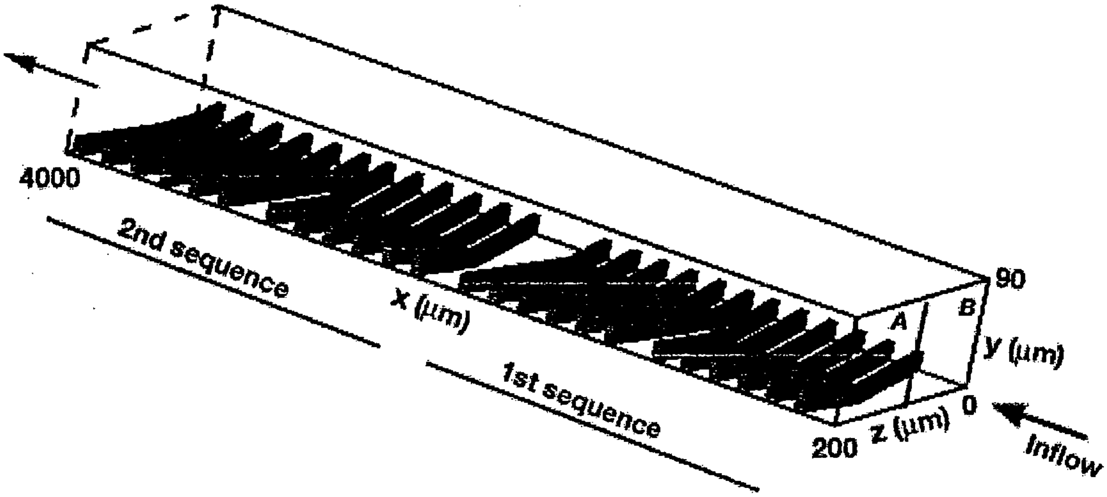

Figure 10. Geometry of the staggered herringbone structures. The four sides are wall boundaries. The incoming velocity is specified at the inflow boundary, and zero-gradient conditions are used at the outflow boundary.

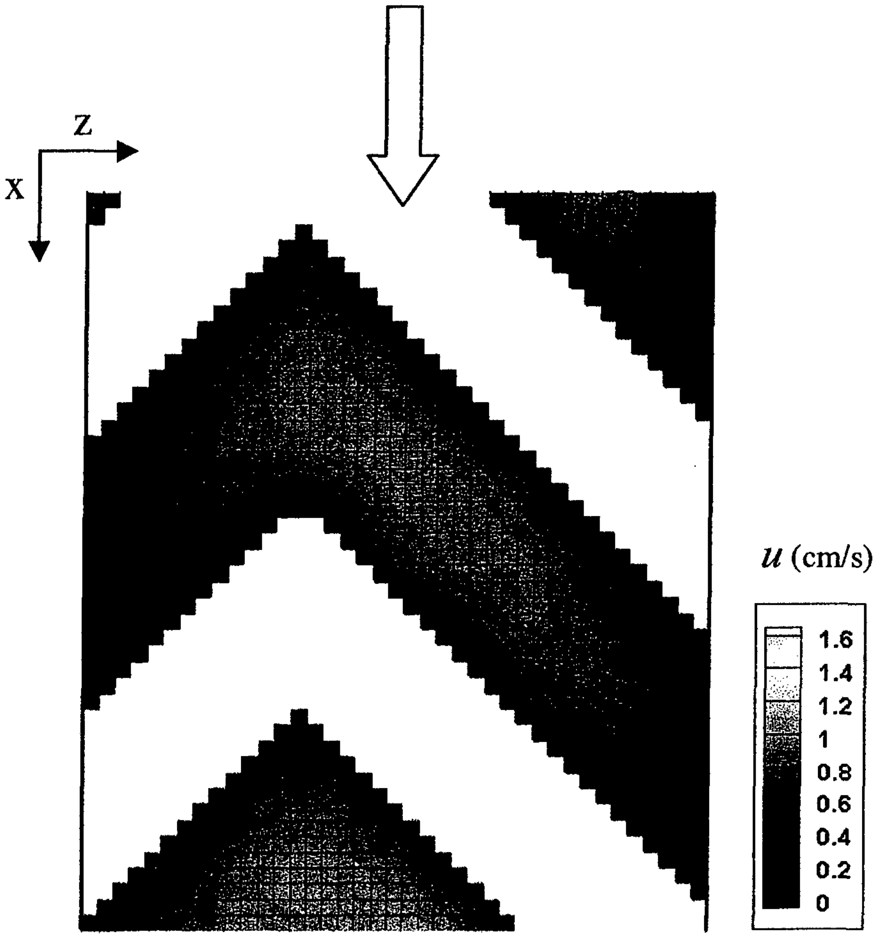

Figure 11. Staircased grid on a y plane intersecting with the herringbone structures. The grid points shown above are the cell centers, which are overlaid with the x-component of the velocity at $y = 17.5$ μm.

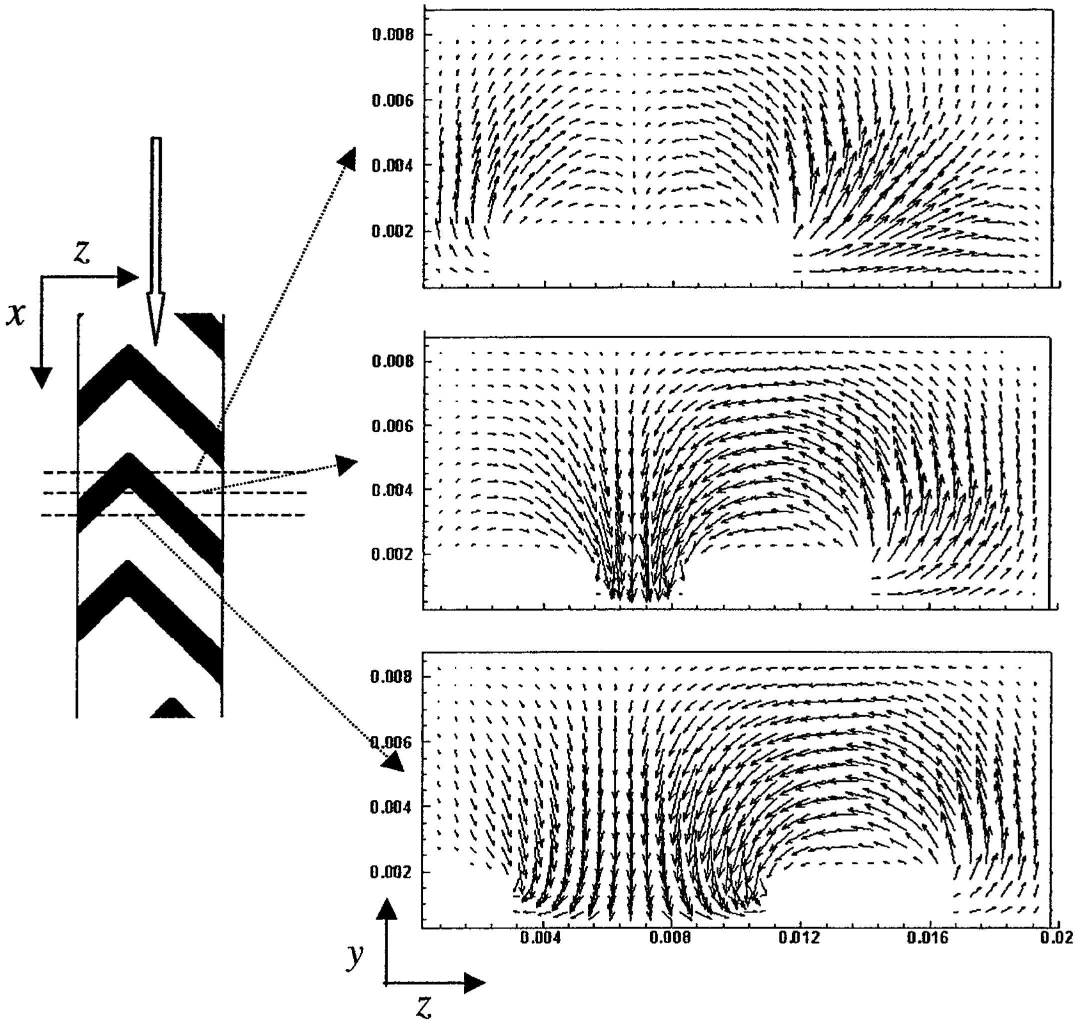

Figure 12. The secondary velocity field (v, w) at $x = 597.5\mu m$, $622.5\mu m$, and $647.5\mu m$, respectively, in the herringbone simulation.

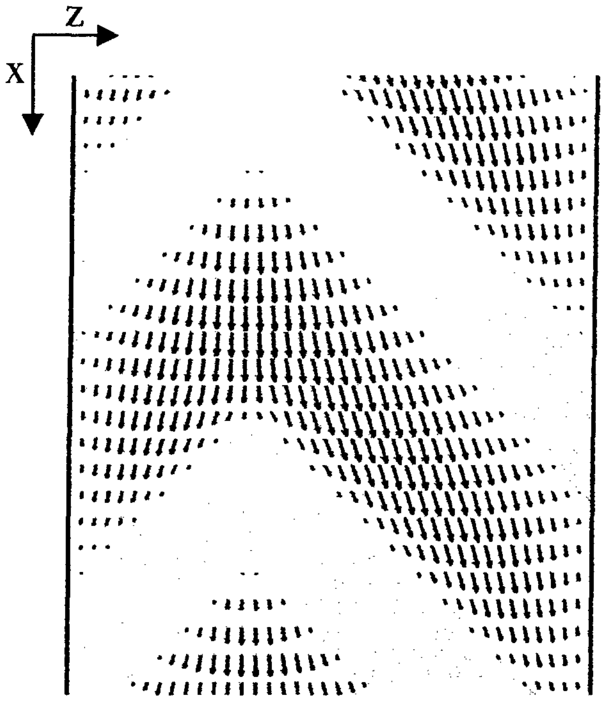

Figure 13a). Velocity vector field (*u*, *w*) at $y = 17.5\mu m$ in the herringbone simulation.

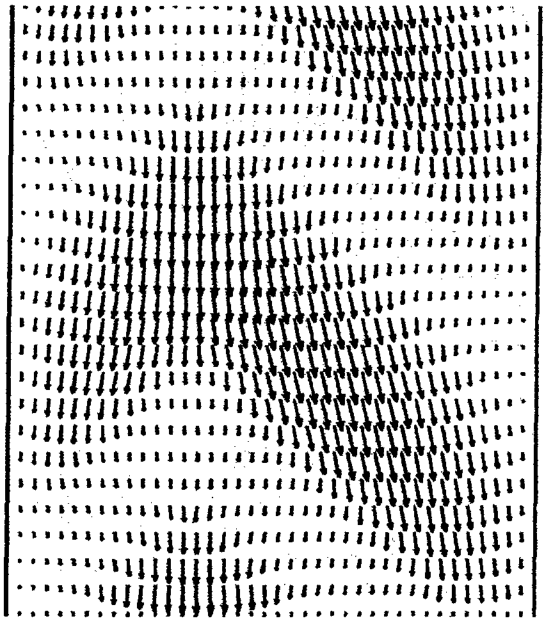

Figure 13b). Velocity vector field (*u*, *w*) at $y = 22.5\mu m$, barely above the herringbone structures in the herringbone simulation.

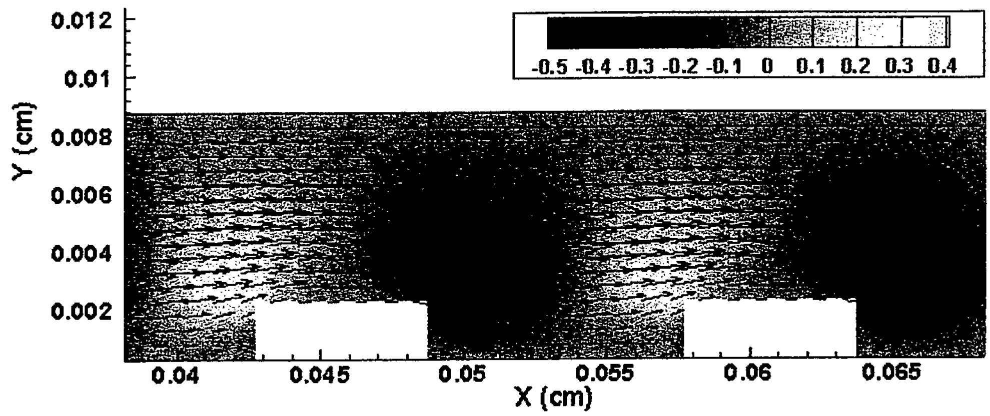

Figure 14a. Velocity vector field (*u, v*) at $z = 97.5\ \mu$m in the herringbone simulation, overlaid with the contours of the y-component of the velocity.

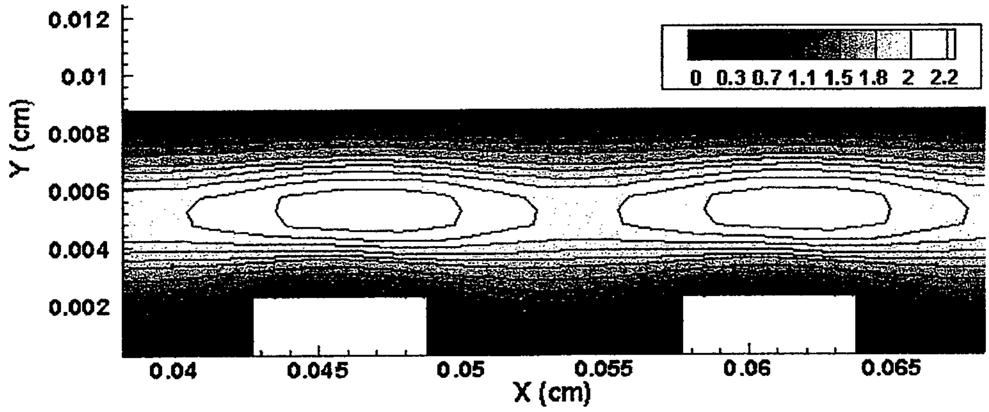

Figure 14b. The x-component of the velocity at $z = 97.5\ \mu$m in the herringbone simulation.

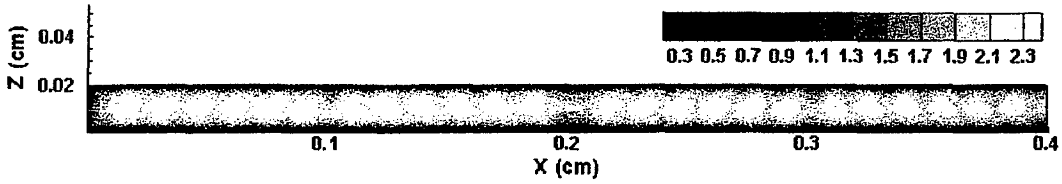

Figure 15a). The *x*-component of the velocity at $y = 47.5\ \mu\text{m}$ in the herringbone simulation.

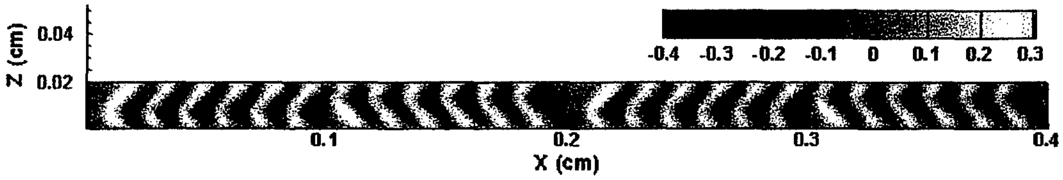

Figure 15b). The *y*-component of the velocity at $y = 47.5\ \mu\text{m}$ in the herringbone simulation.

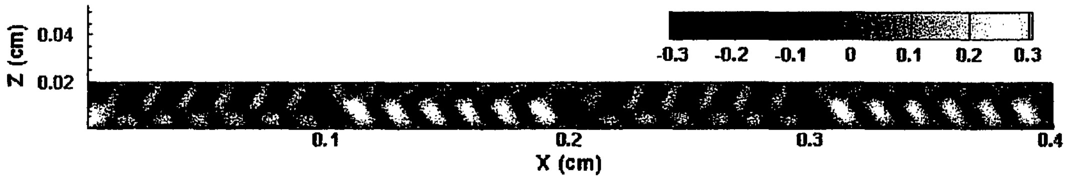

Figure 15c). The *z*-component of the velocity at $y = 47.5\ \mu\text{m}$ in the herringbone simulation.

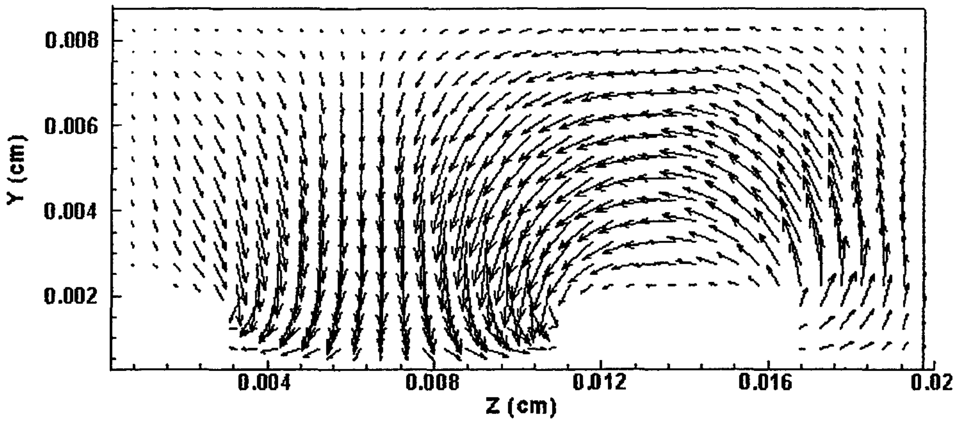

Figure 16a). Vector field of the secondary velocity (v, w) (normalized by the incoming velocity U_∞) at $x = 647.5\mu\text{m}$ in the herringbone simulation.

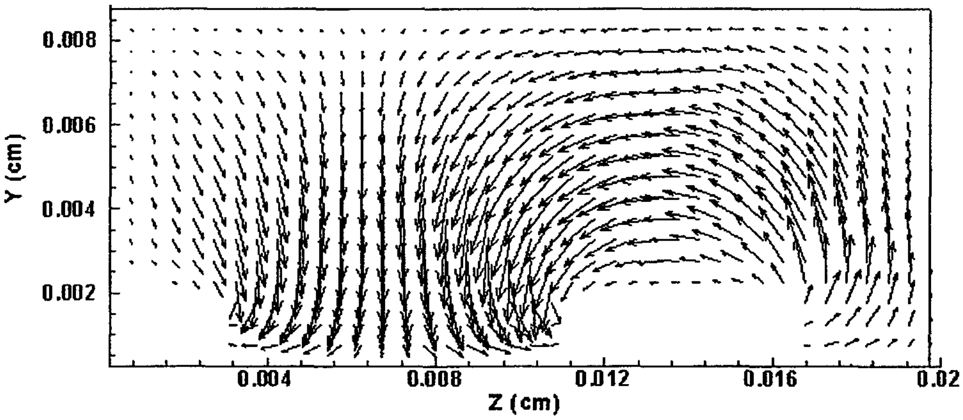

Figure 16b). Vector field of the secondary velocity (v, w) (normalized by the incoming velocity U_∞) at $x = 2647.5\mu\text{m}$ in the herringbone simulation.

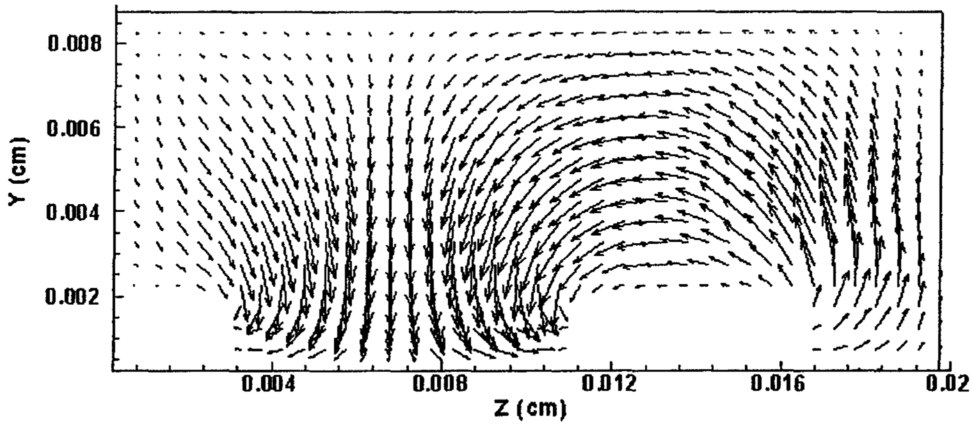

Figure 17a). Vector field of the secondary velocity (v, w) (normalized by the incoming velocity U_∞) at $x = 647.5\mu m$ for the incoming velocity $U_\infty = 10$ cm/s in the herringbone simulation.

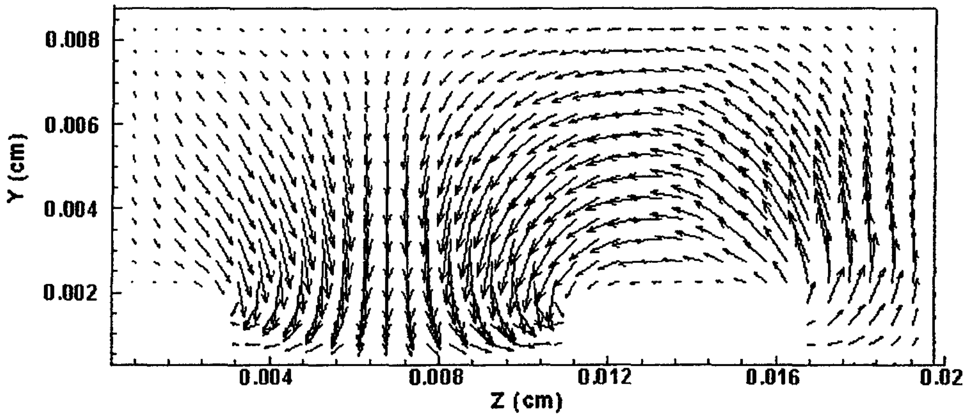

Figure 17b). Vector field of the secondary velocity (v, w) (normalized by the incoming velocity U_∞) at $x = 2647.5\mu m$ for the incoming velocity $U_\infty = 10$ cm/s in the herringbone simulation.

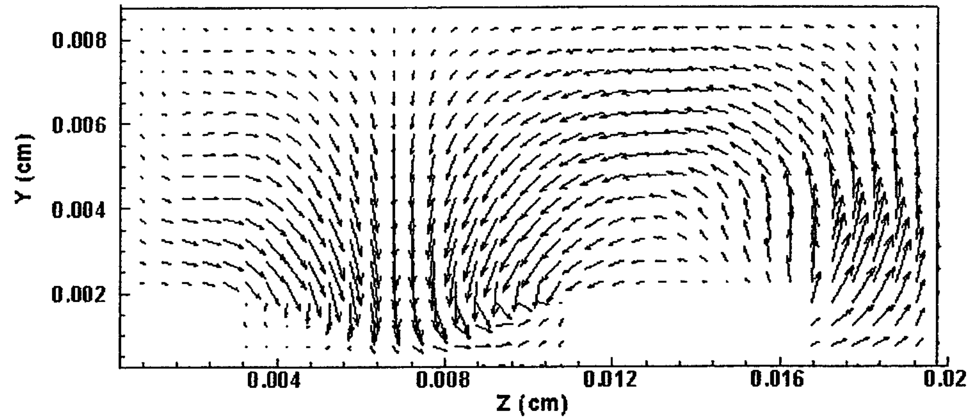

Figure 18a). Vector field of the secondary velocity (v, w) (normalized by the incoming velocity U_∞) at $x = 647.5\mu\text{m}$ for the incoming velocity $U_\infty = 100$ cm/s in the herringbone simulation.

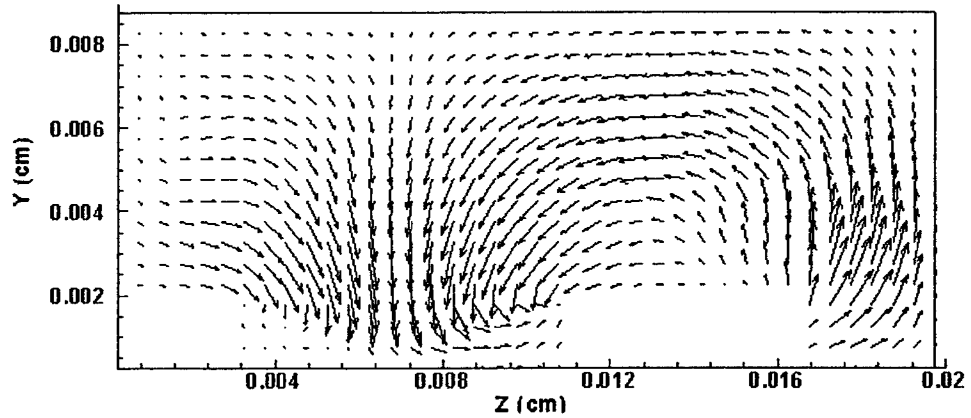

Figure 18b). Vector field of the secondary velocity (v, w) (normalized by the incoming velocity U_∞) at $x = 2647.5\mu\text{m}$ for the incoming velocity $U_\infty = 100$ cm/s in the herringbone simulation.