USE OF THE SPLITTING SCHEME AND MULTIGRID METHOD TO COMPUTE FLOW SEPARATION

CHANG QIANSHUN

Institute of Applied Mathematics, Academia Sinica, Beijing, China

SUMMARY

The splitting difference scheme is used to study flow separation. Flows behind a circular cylinder are computed as a model problem. In view of the nature of the flow, the variables are transformed. The boundary condition for the pressure is given from an intermediate velocity. The free-slip velocity boundary conditions on the rigid wall are given by interpolation. The multigrid algorithm is applied to the pressure iteration. We also choose better initial values for the model problem by means of the multigrid algorithm idea.

KEY WORDS Multigrid Method Flow Separation

INTRODUCTION

In recent years there have been many techniques developed for the numerical solution of the incompressible Navier–Stokes problem. Most of these methods were developed for equations of the stream function and the vorticity function. The main drawback of these methods is the use of a boundary condition for the vorticity at the solid wall, which is absent in hydrodynamics. The obvious limitation of methods of solving for the stream function and vorticity function is the difficulty of extending them to the case of three-dimensional flows. Therefore, we consider the numerical solution of the Navier–Stokes initial–boundary value problem, expressed in the natural variables.

The MAC-type (marker and cell) methods are the most important of the difference methods of solving the Navier–Stokes problem. The MAC method was first published by Harlow and co-workers.[1,2] Chorin[3] used an iterative technique to determine pressure and velocity simultaneously which is, in fact, a numerical construction of the appropriate projection operator. The idea of orthogonal decomposition is used by many splitting methods of solving the Navier–Stokes problem, which were proposed afterwards.

A modified formulation of boundary conditions for the MAC method was presented in Reference 4. The modification enables homogeneous boundary conditions to be obtained for the pressure.

Mathematicians of the Soviet Union gave a new computational formula for velocity boundary conditions on the solid surface.[5,6]

Numerical fluid dynamics is the most challenging field for the multigrid method. China *et al.*[7] have used the vorticity–stream-function formulation of the two-dimensional Navier–Stokes

0271–2091/87/070719–13$06.50

Received 14 September 1986
Revised 5 November 1986

equations and a multigrid method to obtain solutions for the driven flow in a square cavity. Brandt[8,9] discussed the application of the multigrid algorithm to hydrodynamics.

In this paper, we use the splitting scheme and a multigrid algorithm to compute flows behind a circular cylinder. Main attention is given to the computation of flow separation.

DIFFERENTIAL EQUATIONS AND BASIC ALGORITHM

The Navier–Stokes initial-boundary value problem written in vector form is

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} + \nabla P = \mu\Delta\mathbf{u}, \quad t > 0, \mathbf{x} \in \Omega, \tag{1}$$

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

$$\mathbf{u}|_{\partial\Omega} = \mathbf{u}_{\mathrm{B}}, \tag{3}$$

$$\mathbf{u}|_{t=0} = \mathbf{u}_0, \tag{4}$$

where $\mathbf{u} = (u, v)$ is the velocity vector, P is the ratio of the pressure to the constant density (for brevity, we refer to P simply as pressure) and μ is the kinematic viscosity coefficient.

In order to deduce the basic scheme and the pressure boundary condition, we consider differences in the time direction first:

$$\frac{1}{\tau}[\mathbf{u}^{n+1} - \mathbf{u}^n] = -\mathbf{u}^n\cdot\nabla\mathbf{u}^n - \nabla P^{n+1} + \mu\Delta\mathbf{u}^n.$$

Let the intermediate velocity be

$$\tilde{\mathbf{u}} = \mathbf{u}^n + \tau[-\mathbf{u}^n\cdot\nabla\mathbf{u}^n + \mu\nabla\mathbf{u}^n], \tag{5}$$

then

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \tau\nabla P^{n+1}. \tag{6}$$

It follows from (2) that

$$\Delta P^{n+1} = \frac{1}{\tau}\nabla\cdot\tilde{\mathbf{u}}. \tag{7}$$

Assume that ξ is the unit vector in outward normal direction on the boundary. From (3) and (6), we have the pressure boundary condition

$$\left.\frac{\partial P^{n+1}}{\partial \xi}\right|_{\partial\Omega} = \frac{1}{\tau}(\mathbf{u} - \mathbf{u}_{\mathrm{B}}^{n+1})\cdot\xi|_{\partial\Omega}.$$

This condition takes a simpler form on the rigid wall:

$$\left.\frac{\partial P^{n+1}}{\partial \xi}\right|_{\partial\Omega} = \frac{1}{\tau}\mathbf{u}\cdot\xi|_{\partial\Omega}. \tag{8}$$

Combining the formulae (7) and (8), the intermediate velocity on the rigid wall can be cancelled. Therefore, the condition (8) is convenient.

The above-mentioned formulae (5)–(8), (3) and (4) are a basic splitting algorithm for solving the Navier–Stokes problem.

Because we compute flows behind a circular cylinder, the Navier–Stokes problem is expressed in polar co-ordinate form:

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial r}+\frac{v}{r}\frac{\partial u}{\partial \varphi}-\frac{v^2}{r}=-\frac{\partial P}{\partial r}+\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 u}{\partial \varphi^2}-\frac{u}{r^2}-\frac{2}{r^2}\frac{\partial v}{\partial \varphi}\right],$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial r}+\frac{v}{r}\frac{\partial v}{\partial \varphi}+\frac{uv}{r}=-\frac{1}{r}\frac{\partial P}{\partial \varphi}+\mu\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 v}{\partial \varphi^2}-\frac{v}{r^2}+\frac{2}{r^2}\frac{\partial u}{\partial \varphi}\right],$$

$$\frac{\partial}{\partial r}(ru)+\frac{\partial v}{\partial \varphi}=0, \qquad (r,\varphi)\in\Omega, \quad t>0$$

$$\mathbf{u}|_{\partial\Omega}=0, \quad \mathbf{u}|_{\infty}=\mathbf{u}_{\infty},$$

$$\mathbf{u}|_{t=0}=\mathbf{u}_0.$$

Thus, the computational region is a rectangle for flows behind a circular cylinder. In order to compute the flow separation, fine grids are necessary near the rigid wall. The variable r is transformed again. Let $z=\ln r$ and the Navier–Stokes problem is

$$\frac{\partial u}{\partial t}+\mathrm{e}^{-z}\left(u\frac{\partial u}{\partial z}+v\frac{\partial u}{\partial \varphi}-v^2\right)= = \mathrm{e}^{-z}\frac{\partial P}{\partial z}+\mu\mathrm{e}^{-2z}\left(\frac{\partial^2 u}{\partial z^2}+\frac{\partial^2 u}{\partial \varphi^2}-u-2\frac{\partial v}{\partial \varphi}\right), \tag{9}$$

$$\frac{\partial v}{\partial t}+\mathrm{e}^{-z}\left(u\frac{\partial v}{\partial z}+v\frac{\partial v}{\partial \varphi}+uv\right)=-\mathrm{e}^{-z}\frac{\partial P}{\partial \varphi}+\mu\mathrm{e}^{-2z}\left(\frac{\partial^2 v}{\partial z^2}+\frac{\partial^2 v}{\partial \varphi^2}-v+2\frac{\partial u}{\partial \varphi}\right), \tag{10}$$

$$\mathrm{e}^{-z}\frac{\partial}{\partial z}(\mathrm{e}^{z}u)+\frac{\partial v}{\partial \varphi}=0, \qquad (z,\varphi)\in\Omega, \quad t>0 \tag{11}$$

$$\mathbf{u}|_{\partial\Omega}=0, \quad \mathbf{u}|_{\infty}=\mathbf{u}_{\infty}, \tag{12}$$

$$\mathbf{u}|_{t=0}=\mathbf{u}_0, \tag{13}$$

where $u=\partial r/\partial t$, $v=r(\partial\varphi/\partial t)$.

DIFFERENCE SCHEME

We employ the formulae (5)–(7) to deduce the difference scheme. In the scheme, the pressures are stored at cell centres and the velocities at cell sides (see Figure 1). The difference scheme for (9)–(11) has the following form:

$$\frac{\tilde{u}_{i+(1/2),j}-u^n_{i+(1/2),j}}{\tau}=-\mathrm{e}^{-z_{i+(1/2)}}\left[u^n_{i+(1/2),j}\frac{u^n_{i+1,j}-u^n_{i,j}}{h_1}+v^n_{i+(1/2),j}\frac{u^n_{i+(1/2),j+(1/2)}-u^n_{i+(1/2),j-(1/2)}}{h_2}-(v^n_{i+(1/2),j})^2.\right] \tag{14}$$

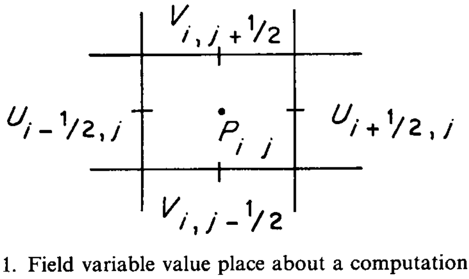

Figure 1. Field variable value place about a computational cell

$$
\begin{aligned}
&+\mu \mathrm{e}^{-2z_{i+(1/2)}}\left[\frac{u^n_{i+(3/2),j}-2u^n_{i+(1/2),j}+u^n_{i-(1/2),j}}{h_1^2}\right.\\
&+\frac{u^n_{i+(1/2),j+1}-2u^n_{i+(1/2),j}+u^n_{i+(1/2),j-1}}{h_2^2}\\
&\left.-u^n_{i+(1/2),j}-\frac{2}{h_2}(v^n_{i+(1/2),j+(1/2)}-v^n_{i+(1/2),j-(1/2)})\right],
\end{aligned}
$$

$$
\begin{aligned}
\frac{\tilde{v}_{i,j+(1/2)}-v^n_{i,j+(1/2)}}{\tau}=&-\mathrm{e}^{-z_i}\left[u^n_{i,j+(1/2)}\frac{v^n_{i+(1/2),j+(1/2)}-v^n_{i-(1/2),j+(1/2)}}{h_1}\right.\\
&\left.+v^n_{i,j+(1/2)}\frac{v^n_{i,j+1}-v^n_{i,j}}{h_2}+u^n_{i,j+(1/2)}v^n_{i,j+(1/2)}\right]\\
&+\mu \mathrm{e}^{-2z_i}\left[\frac{v^n_{i+1,j+(1/2)}-2v^n_{i,j+(1/2)}+v^n_{i-1,j+(1/2)}}{h_1^2}\right.\\
&\left.+\frac{v^n_{i,j+(3/2)}-2v^n_{i,j+(1/2)}+v^n_{i,j-(1/2)}}{h_2^2}-v^n_{i,j+(1/2)}+\frac{2}{h_2}(u^n_{i,j+1}-u^n_{i,j})\right],
\end{aligned}
\tag{15}
$$

$$
\frac{1}{h_1^2}\mathrm{e}^{-z_i}(P^{n+1}_{i+1,j}-2P^{n+1}_{i,j}+P^{n+1}_{i-1,j})+\frac{1}{h_2^2}\mathrm{e}^{-z_i}(P^{n+1}_{i,j+1}-2P^{n+1}_{i,j}+P^{n+1}_{i,j-1})=\frac{1}{\tau}\tilde{D}_{i,j},\tag{16}
$$

$$
\tilde{D}_{i,j}\equiv\frac{1}{h_1}\mathrm{e}^{-z_i}(\mathrm{e}^{z_{i+(1/2)}}\tilde{u}_{i+(1/2),j}-\mathrm{e}^{z_{i-(1/2)}}\tilde{u}_{i-(1/2),j})+\frac{1}{h_2}(\tilde{v}_{i,j+(1/2)}-\tilde{v}_{i,j-(1/2)}),\tag{17}
$$

$$
u^{n+1}_{i+(1/2),j}=\tilde{u}_{i+(1/2),j}-\frac{\tau}{h_1}\mathrm{e}^{-z_{i+(1/2)}}(P^{n+1}_{i+1,j}-P^{n+1}_{i,j}),\tag{18}
$$

$$
v^{n+1}_{i,j+(1/2)}=\tilde{v}_{i,j+(1/2)}-\frac{\tau}{h_2}\mathrm{e}^{-z_i}(P^{n+1}_{i.j+1}-P^{n+1}_{i,j}),\tag{19}
$$

where $1\leqslant i\leqslant \mathrm{IT}$, $1\leqslant j\leqslant \mathrm{JT}$, IT and JT give the total numbers of points in the z and φ directions, respectively, τ is the step size of time, h_1 and h_2 are the step sizes in the z and φ directions, respectively.

It is easy to see that (14)–(19) approximate the differential equations (9)–(11) with an error of order $O(\tau, h^2)$, where $h=\mathrm{Max}(h_1, h_2)$.

First, we compute the intermediate velocities $\tilde{u}$, $\tilde{v}$, ignoring the pressure. Then Poisson's equations for the pressure are solved. Finally, using the new pressure values, we modify the intermediate velocities $\tilde{u}$, $\tilde{v}$ and obtain new values of the velocities u^{n+1}, v^{n+1} at a new time.

It may be noted that some of the velocities and pressures are required at points where these quantities are not stored. We use an average of adjacent values to give the value at such a point: for example

$$
\begin{aligned}
u_{i+(1/2),j+(1/2)}&=\tfrac{1}{2}(u_{i+(1/2),j+1}+u_{i+(1/2),j}),\\
v_{i+(1/2),j}&=\tfrac{1}{4}(v_{i,j+(1/2)}+v_{i,j-(1/2)}+v_{i+1,j+(1/2)}+v_{i+1,j-(1/2)}).
\end{aligned}
$$

In the difference scheme, momentum equations are explicit, since we consider flows with higher Reynolds number. The pressure equations are implicit and are solved by a multigrid method.

APPLICATION OF THE MULTIGRID METHOD

In the difference scheme (14)–(19), only formulation (16) to solve the pressure is implicit. Therefore, we employ a multigrid method to solve it. Because pressure values at the previous time step are good approximations for pressure values at the current time step, we use such iterative initial values: $P_{i,j}^{n+1(0)} = P_{i,j}^{n}$.

In order that this paper is reasonably self-contained, we consider a brief description of the multigrid method applied to the equation

$$\Delta u = F, \quad \text{in} \quad \Omega.$$

Its approximation solution u_h on the fine grid Ω_h satisfies

$$\Delta_h u_h = F_h, \quad \text{in} \quad \Omega_h.$$

The operator Δ_H is an appropriate approximation of the operator Δ_h on a coarser grid Ω_H. The operator I_h^H is a restriction operator from the fine grid Ω_h to the coarse grid Ω_H. The operator I_H^h is an interpolation operator from the coarse grid to the fine grid. We have an approximate solution $u_h^{(s)}$ of the equation $\Delta_h u_h = F_h$. In order to find a new approximation $u_h^{(s+1)}$, the process of the multigrid method is illustrated in Figure 2.

Of course, this basic principle can be applied recursively employing coarser and coarser grids. The structure of one iteration step (cycle) in a multigrid method is illustrated with a few pictures which are given in Figure 3.

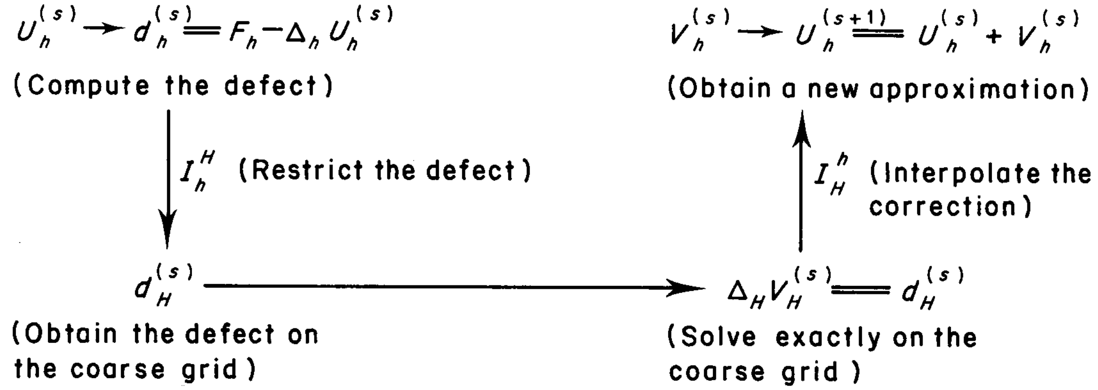
$U_h^{(s)} \to d_h^{(s)} = F_h - \Delta_h U_h^{(s)}$
(Compute the defect)
I_h^H (Restrict the defect)
$d_H^{(s)}$
(Obtain the defect on the coarse grid)
$\Delta_H V_H^{(s)} = d_H^{(s)}$
(Solve exactly on the coarse grid)
I_H^h (Interpolate the correction)
$V_h^{(s)} \to U_h^{(s+1)} = U_h^{(s)} + V_h^{(s)}$
(Obtain a new approximation)

Figure 2. Process of the multigrid method

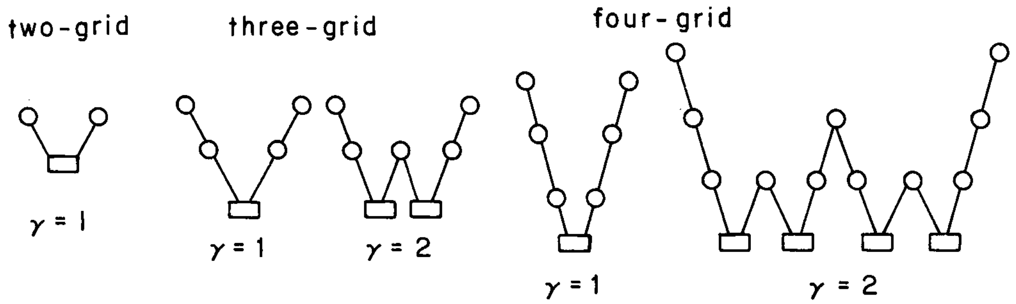

Figure 3. Structure of one multigrid cycle: ○, □, \ and / mean smoothing, solving exactly, fine-to-coarse and coarse-to-fine transfer, respectively

In our computation, the ratio of step size on the coarse grid Ω_H to that on the fine grid Ω_h is 2.

Note that the coarse grid pressures are located at points different from the location of the fine grid pressures in the difference scheme. We choose the restriction operator from fine grid to coarse grid as an average of four neighbouring points with the same weight 0·25:

$$P^H_{i,j} = \tfrac{1}{4}(P^h_{2i-1,2j-1} + P^h_{2i-1,2j} + P^h_{2i,2j-1} + P^h_{2i,2j}).$$

Bilinear interpolation is employed as the interpolation operator from coarse grid to fine grid and this is written as

$$P^h_{2i,2j} = \tfrac{1}{16}(9P^H_{i,j} + 3P^H_{i+1,j} + 3P^H_{i,j+1} + P^H_{i+1,j+1}).$$

MODEL PROBLEM AND BOUNDARY CONDITIONS

As a model problem, we consider incompressible, viscous flows past a circular cylinder of radius a, with the axis of the circular cylinder perpendicular to the velocity of the incident flow $\mathbf{u}_\infty$. For simplicity, we take $a = 1$ and $|\mathbf{u}_\infty| = 1$. Thus the Reynolds number $Re = 2/\mu$.

Considering symmetry, the computational region is the rectangle

$$\{z, \varphi \,|\, z \in [0, z_\infty], \varphi \in [0, \pi]\}.$$

It is enough to compute flow separation that we take $z_\infty = 3$, i.e. $r_\infty = 20{\cdot}086$.

The boundary conditions for the difference scheme (14)–(19) have the following form:

(i) Conditions on the unperturbed flow are specified at $z = z_\infty$:

$$u^{n+1}_{\mathrm{IT}+(1/2),j} = \cos\varphi, \quad V^{n+1}_{\mathrm{IT}+1,j+(1/2)} = -\sin\varphi, \quad P^{n+1}_{\mathrm{IT}+1,j} = 0.$$

(ii) A rigid wall of no-slip is located at $z = 0$:

$$u^{n+1}_{(1/2),j} = 0, \quad V^{n+1}_{0,j+(1/2)} = -V^{n+1}_{1,j+(1/2)}.$$

Using the formula (8), we obtain the boundary condition for the pressure as

$$\frac{1}{h_1} \mathrm{e}^{-z_{1/2}}(P^{n+1}_{1,j} - P^{n+1}_{0,j}) = \frac{1}{\tau}\tilde{u}_{(1/2),j}.$$

(iii) Rigid walls of free-slip are located at $\varphi = 0$ and $\varphi = \pi$. A rigid wall of free-slip is considered to represent a plane of symmetry, rather than a true wall:

$$V^{n+1}_{i,(1/2)} = 0, \quad V^{n+1}_{i,\mathrm{JT}+(1/2)} = 0.$$

For tangential velocities, simple formulae are given by symmetry[1] as

$$u^{n+1}_{i+(1/2),0} = u^{n+1}_{i+(1/2),1}, \quad u^{n+1}_{i+(1/2),\mathrm{JT}+1} = u^{n+1}_{i+(1/2),\mathrm{JT}}. \tag{20}$$

From the formulae (14)–(19), we can see that the boundary conditions (20) are employed only in computing the intermediate velocities $\tilde{u}_{i+(1/2),\mathrm{I}}$ and $\tilde{u}_{i+(1/2),\mathrm{JT}}$. Therefore, boundary conditions for the intermediate velocities $\tilde{u}_{i+(1/2),\mathrm{I}}$ and $\tilde{u}_{i+(1/2),\mathrm{JT}}$ are considered instead of (20). These boundary conditions are given by our deduction as

$$\tilde{u}_{i+(1/2),\mathrm{I}} = \tfrac{5}{6}u^n_{i+(1/2),\mathrm{I}} + \tfrac{1}{6}u^n_{i+(1/2),2}, \quad \tilde{u}_{i+(1/2),\mathrm{JT}} = \tfrac{5}{6}u^n_{i+(1/2),\mathrm{JT}} + \tfrac{1}{6}u^n_{i+(1/2),\mathrm{JT}-\mathrm{I}} \tag{21}$$

which are obtained by quadratic interpolation. The boundary conditions for the pressure are similar to those of the rigid wall of no-slip:

$$\frac{1}{h_2}\mathrm{e}^{-z_i}(P^{n+1}_{i,1} - P^{nH}_{i,0}) = \frac{1}{\tau}\tilde{V}_{i,(1/2)}, \quad \frac{1}{h_2}\mathrm{e}^{-z_i}(P^{n+1}_{i,\mathrm{JT}+1} - P^{n+1}_{i,\mathrm{JT}}) = \frac{1}{\tau}\tilde{V}_{i,\mathrm{JT}+(1/2)}.$$

STABILITY

It is difficult to derive the full necessary and sufficient conditions of numerical stability for the Navier–Stokes problem. But, some valuable conditions are obtained by theoretical analysis.

The basic condition of stability is that fluid must not be permitted to flow across more than one computational cell in one time step, i.e.

$$\tau < \operatorname*{Min}_{i,j}\left[\frac{e^{z_i}\cdot h_1}{|u_{ij}|},\frac{e^{z_i}\cdot h_2}{|V_{i,j}|}\right]. \tag{22}$$

This condition is also necessary for accuracy, because the convective flux approximations in the difference scheme (14)–(19) assume exchanges between adjacent cells only.

When the viscous diffusion terms are important, we have to consider a stability condition for the explicit approximation of the parabolic equations. This restricts the time increment to

$$\tau < \operatorname*{Min}_{i,j}\frac{1}{2\mu\left[\left(\frac{1}{e^{z_i}\cdot h_1}\right)^2+\left(\frac{1}{e^{z_i}\cdot h_2}\right)^2\right]}. \tag{23}$$

Using a heuristic stability analysis proposed Hirt,[10] a necessary stability condition is given by

$$\mu > \operatorname*{Max}_{i,j}\left[\frac{\tau}{2}u_{i,j}^2,\frac{\tau}{2}V_{i,j}^2\right]. \tag{24}$$

This condition can be explained as avoiding the occurrence of truncation errors that have the form of negative diffusion.

We use the above-mentioned three conditions (22)–(24) as necessary ones. They are compared with stability conditions given by our computational experience. The results are given in Table I.

Table I. Comparison on step size of stability

Reynolds number	1		10		100
Grid	32 x 32	16 x 16	64 x 64	16 x 16	64 x 64
Stability condition given by the conditions (22)–(24)	$\tau < 0{\cdot}0012$	$\tau < 0{\cdot}0053$	$\tau < 0{\cdot}0028$	$\tau < 0{\cdot}053$	$\tau < 0{\cdot}028$
The longest step size of stability given by computational experience	0·001	0·005	0·002	0·05	0·002

From Table I, we see that the conditions (22)–(24) in the cases $Re = 1$ and $Re = 10$ give step sizes of stability, which are almost sufficient for stability. But, the sufficient stability condition is much more stringent than the conditions (22)–(24) in the case $Re = 100$. The reason is that there is a large stagnation region in the flow field of $Re = 100$.

The truncation error is unavoidable in finite difference approximations, and they do influence the accuracy of a calculation. In order that the effects of μ are not to be obscured by truncation error, the following condition is necessary: $Re < 4(\mathrm{IT})^2$. In fact, a grid which is much finer than the grid given by the $Re < 4(\mathrm{IT})^2$ is taken in the computation of flow separation. In order to obtain correct separation phenomena, we have to take the grid IT = JT = 64 at the cases $Re = 10$ and $Re = 100$.

DISCUSSION OF COMPUTATIONAL RESULTS

Analysis of the multigrid algorithm

At the present time, a fairly large number of applications of the multigrid algorithm are known. Most of these applications deal with the steady problem. Experience with multigrid applications to an evolution problem is sparse. In the Navier–Stokes problem considered here, velocity and pressure vary with time and they tend to a steady solution if the Reynolds number is not very high. We have to analyse and decide how to employ the multigrid algorithm. In the initial period of computation, there are large increments of the pressures and velocities in one step of time. We compared the multigrid algorithm with the Gauss–Seidel iterative method for solving the pressure equations (16). In the grid IT = 64, JT = 64, time $t^n = 0$ an initial values are

$$u^0_{i+(1/2),j} = \frac{i}{64}\cos\left((j-\tfrac{1}{2})\frac{\pi}{64}\right),\quad V^0_{i,j+(1/2)} = \frac{-(i-\frac{1}{2})}{64}\sin\left(j\frac{\pi}{64}\right),\quad P^0_{i,j} = 0.$$

The results are given in Table II.

In Table II, $M = 1$ means that the Gauss–Seidel iteration is used instead of the multigrid algorithm and $M = 4$ means that we use the multigrid algorithm with four grids 64 × 64, 32 × 32, 16 × 16 and 8 × 8. It follows from Table II that the multigrid algorithm is very efficient in the initial period of computation. But the solution of our problem tends to a stable one and the increments of the pressures and velocities in one step of time are decreased with time. When the increments are small enough, the simple Gauss–Seidel iteration is better than the multigrid algorithm. For example, in the same case as that decribed by Table II, the computational results are given in Table III for the later time $t = 1$.

Therefore, we employ the multigrid algorithm in the initial computation. When the number of iterations in the multigrid algorithm is 1, the Gauss–Seidel iteration is used. If the number of Gauss–Seidel iterations is more than 5, we employ the multigrid algorithm again.

Choice of the initial values

In our problem, the solution of the Navier–Stokes equations tends to a steady solution. Therefore, computational results are independent of the initial values. In view of idea of the multigrid method, we can choose better initial values to save computational time. First, the Navier–Stokes problem is solved on a coarse grid with IT = 16 and JT = 16. Then, interpolating

Table II. The results at $t = 0$

Reynolds number Re =	10					1	
The number of levels in the multigrid M =	1	4	4	5	5	1	4
Type of cycle γ =		1	2	1	2		1
The number of iterations	9638	17	8	33	8	11,824	19
CPU time in one step of time (in seconds)	562·7	8.27	6.50	12·35	4·98	623·0	9·21

Table III. The results at $t = 1$

Reynolds number $Re =$			10		
The number of levels in the multigrid $M =$	1	4	4	5	5
Type of cycle $\gamma =$	/	1	2	1	2
The number of iterations	3	1	1	1	1
CPU time in one step of time (in seconds)	0·74	1·01	1·11	1·01	1·14

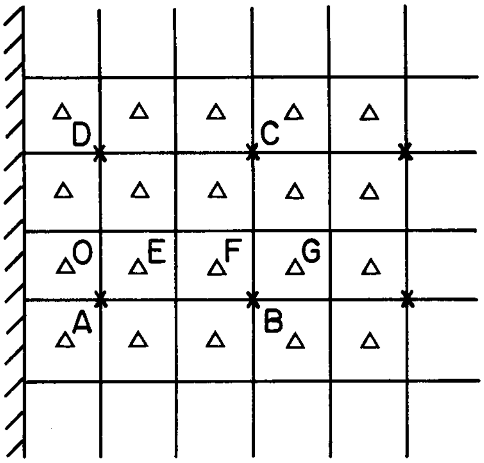

Figure 4. Cells near the rigid wall

the computational values of the pressures and velocities, their initial values on the next finer grid are obtained. The problem is solved on the grid with IT = 32 and JT = 32. Finally, we interpolate again and compute the flow separation of the Navier–Stokes equations on the fine grid with IT = 64 and JT = 64. Using this method, the flow separation for $Re = 10$ is computed and its CPU time is only 29 min on the IBM-3083 computer. If we use general initial values on the grid with IT = 64 and JT = 64, for example

$$u^0_{i+(1/2),j} = \frac{i}{64}\cos\left((j-\tfrac{1}{2})\frac{\pi}{64}\right), \quad V^0_{i,j+(1/2)} = \frac{-(i-\frac{1}{2})}{64}\sin\left(j\frac{\pi}{64}\right), \quad P^0_{i,j} = 0,$$

then CPU time for the flow separation at $Re = 10$ is 134 min.

This method of choosing initial values is efficient. It is also suitable for some problems of parabolic type.

Choice of extrapolation operator near the rigid wall

The pressure values outside and on the rigid wall are unknown. Therefore in the multigrid method, we have to use extrapolation operator near the rigid wall to determine the approximate pressure values on the fine grid. Three formulae are considered (for notation see Figure 4):

(a) bilinear extrapolation from the coarse grid:

$$P_{\mathrm{O}}^{h}=\tfrac{15}{16}P_{\mathrm{A}}^{H}-\tfrac{3}{16}P_{\mathrm{B}}^{H}+\tfrac{5}{16}P_{\mathrm{D}}^{H}-\tfrac{1}{16}P_{\mathrm{C}}^{H} \tag{25}$$

(b) quadratic extrapolation in one direction of the fine grid:

$$P_{\mathrm{O}}^{h}=3P_{\mathrm{E}}^{h}-3P_{\mathrm{F}}^{h}+P_{\mathrm{G}}^{h} \tag{26}$$

(c) linear extrapolation in one direction of the fine grid

$$P_{\mathrm{O}}^{h}=2P_{\mathrm{E}}^{h}-P_{\mathrm{F}}^{h}. \tag{27}$$

In the case $t=0$, $Re=10$, $\mathrm{IT}=64$, $\mathrm{JT}=64$ and $M=4$ (the number of levels in the multigrid), we compared the three formulae. The results are given in Table IV.

Using the formula (27), iterative error tends to a constant, which is much more than 10^{-4}. In view of our experience, the formula (25) is better than the formula (26), especially on a coarser grid.

Computational results

We compute flows behind a circular cylinder in the cases $Re=1$, 10, 100. The boundary conditions (20) and (21) for the rigid wall of free-slip are compared. The computational results are given in Table V.

Comparing the results with those of experiment, we know that the boundary condition (21) for the rigid wall of free-slip is better than the condition (20). The pictures of streamlines for the problem (9)–(13) are given in Figures 5–7. We can see from Figures 6 and 7 that there are a line, along which velocity equals to zero, and a stagnation region in each Figure.

Table IV. Comparison of the extrapolation formulae

Formula	(25)		(26)		(27)	
Type of cycle $\gamma=$	1	2	1	2	1	2
The number of iterations	17	8	17	8	∞	∞

Table V. The results at $t=t_{\infty}$ (the solution is steady)

Re	Boundary condition	P_0	P_π	C_f	C_p	C_D	θ_s
100	(20)	−0·4897	1·1284	0·2884	0·8866	1·1750	66°
	(21)	−0·5696	1·0612	0·2856	0·9149	1·2005	66°
10	(20)	−0·8125	1·6237	1·3180	1·7797	3·0977	29·5°
	(21)	−0·8468	1·3870	1·3156	1·7442	3·0598	29·5°
1	(21)	−3·2666	3·6557	6·3220	6·5144	12·8364	0°

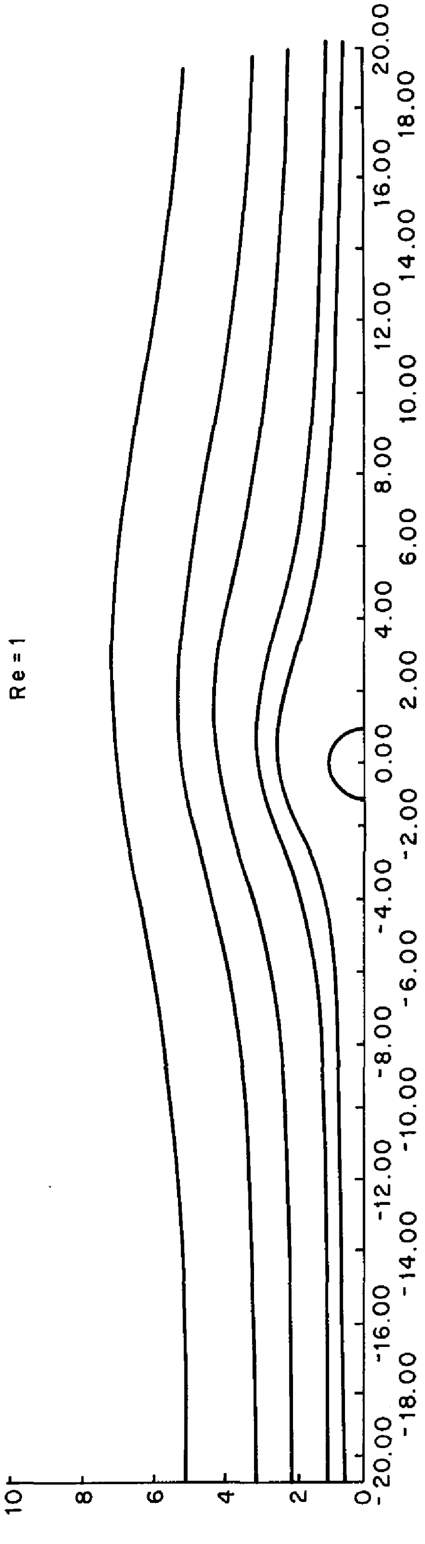

Figure 5.

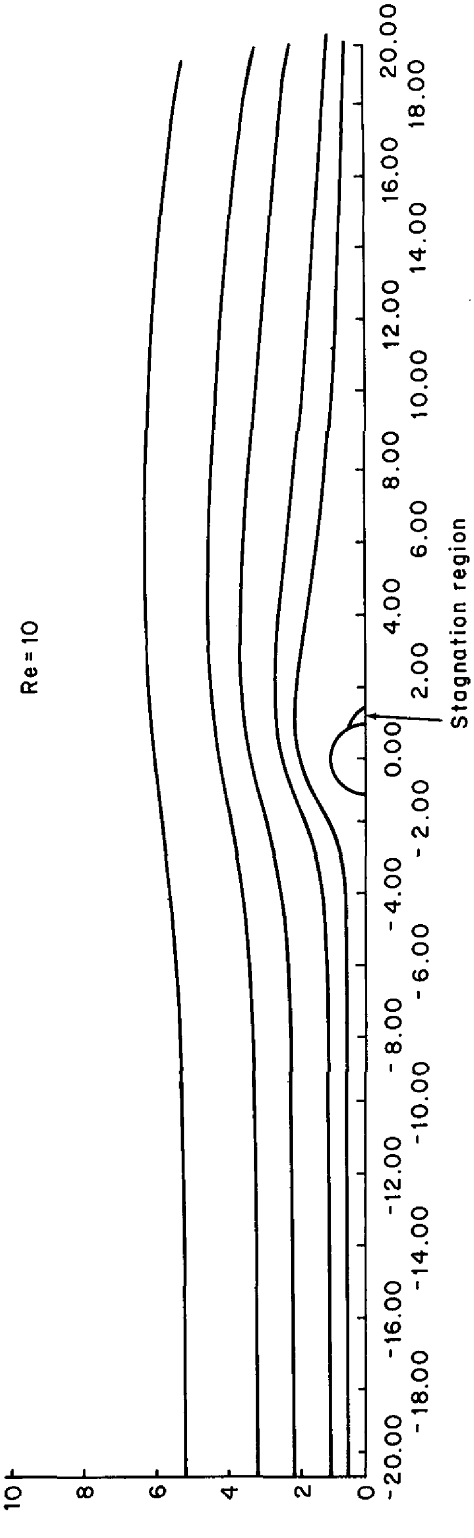

Figure 6.

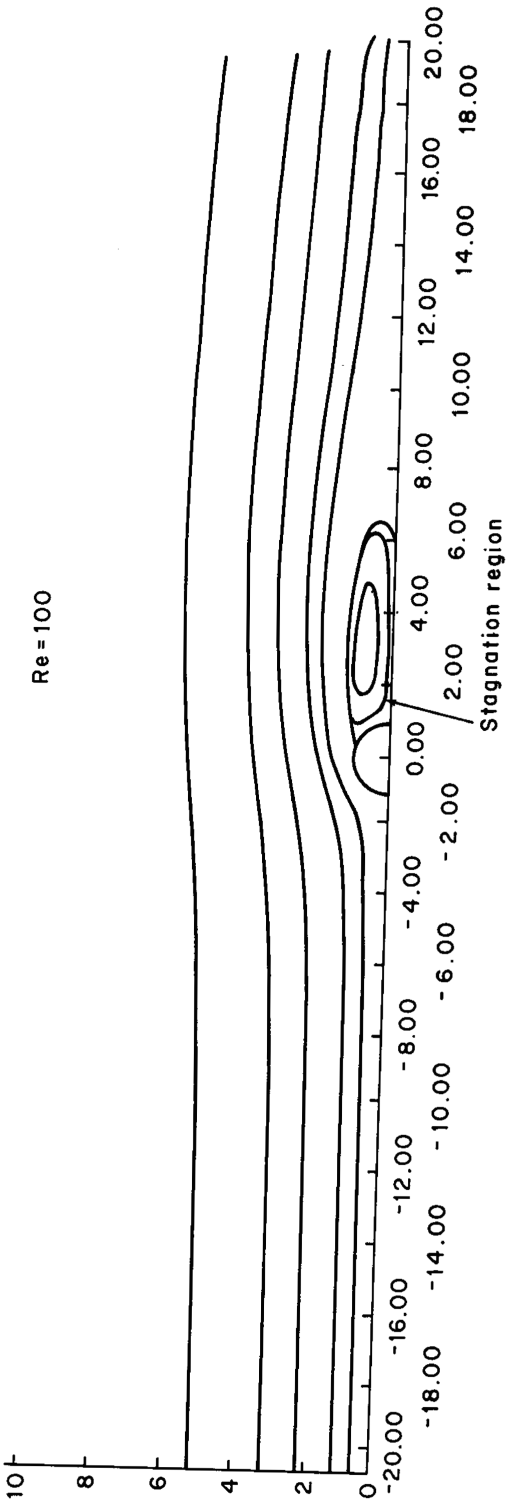
Re = 100
Stagnation region
-20.00
-18.00
-16.00
-14.00
-12.00
-10.00
-8.00
-6.00
-4.00
-2.00
0.00
2.00
4.00
6.00
8.00
10.00
12.00
14.00
16.00
18.00
20.00
0
2
4
6
8
10

Figure 7.

ACKNOWLEDGEMENTS

This work was performed in the period during which the author visited GMD (Gesellschaft fur Mathematik und Datenverarbeitung) in West Germany. The author would like to express appreciation to U. Trottenberg.

REFERENCES

1. F. H. Harlow and J. E. Welch, 'Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface', *Phys. Fluids*, **8**, 2182–2189 (1965).
2. J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, 'The MAC method', *LA-3425*, 1966.
3. A. J. Chorin, 'Numerical solution of Navier–Stokes equations', *Math. Comput.*, **22**, 745–762 (1968).
4. A. A. Amsden and F. H. Harlow, 'The SMAC method', *LA-4370*, Los Alamos National Laboratory, California University, 1970.
5. O. M. benoyepkoBcknñ, B. A. TyiynH and B. B. UjeHHnkoB 'Use of the splitting method to solve problems of the dynamics of a viscous incompressible fluid', *Hcy. BiluncA. MaT. n MaT. opnz.*, **15**, 197–207 (1975).
6. B. A. TyiynH, 'The flow of a viscous fluid past three-dimensional bodies', *Hcy. BiluncA. MaTn Mat. opnz.*, **16**, 529–534 (1976).
7. 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**, 387–411 (1982).
8. A. Brandt, J. E. Dendy and H. Ruppel, 'The multigrid method for semi-implicit hydrodynamic codes', *J. Comput. Phys.*, **34**, 348–370 (1980).
9. A. Brandt, 'Multigrid techniques: 1984 guide with applications to fluid dynamics', *GMD-Studien No. 85*, 1984.
10. C. W. Hirt, 'Heuristic stability theory for finite-difference equations', *J. Comput. Phys.*, **2**, 339–355 (1968).