INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 12, 965–983 (1991)

# AN ALGORITHM THAT ACCELERATES CONVERGENCE RATES FOR INCOMPRESSIBLE NAVIER–STOKES PROBLEMS

T. M. SHIH

*Department of Mechanical Engineering, University of Maryland, College Park, MD 20742, U.S.A.*

AND

B. C. HWANG

*David Taylor Research Center, Annapolis, MD 21402, U.S.A.*

## SUMMARY

An algorithm, called the Algebraic Continuity Equations Solver (ACES), is developed based on the concept that two algebraic equations (three for 3D problems) can be generated from rearranging the discretized continuity equations. These rearranged equations are used to re-compute the two velocity components (three for 3D problems), whose values are already obtained from solving the momentum equations. When written in a Navier–Stokes computer code, this algorithm is equivalent to a fairly concise set of statements and can be implemented immediately after the computation of the continuity equation. In our analysis, ACES is used in conjunction with a grid having nodal velocity components at the vertices and the nodal pressure at the centre of each computational cell.

With the aid of ACES, correction of velocity components during the iteration can be inexpensively made, leading to faster convergence rates or rendering otherwise divergent computations convergent.

Test problems include benchmark problems such as lid-driven cavity flows and buoyancy-driven cavity flows of various parametric values and grid sizes. A 3D time-dependent flow in an irregular geometry is also investigated. Discussions are presented to clarify some relevant issues. A possible reason why we think ACES is capable of improving the convergence rates is also given.

KEY WORDS Navier–Stokes Incompressible Convergence

## INTRODUCTION

The primitive-variable formulation has been widely adopted for computing incompressible Navier–Stokes flows.[1–3] Conventionally, in the formulation the momentum equations are used to solve for the velocity components and the continuity equation is reserved for computing the pressure.[4]

In this analysis we will focus our attention on one of the most important properties of numerical schemes: the convergence rate, viewed as the measure of how fast the computed solution reaches values that remain unchanged even if iterations continue. It is known to us that the same scheme may lead to different convergence rates under different conditions of flow characteristics or geometries. For example, the convergence rates generally become slow when the convection is large and/or the computational cells are slender or skewed. Thus a seemingly viable

0271-2091/91/100965–19$09.50

*Received 23 June 1989*
*Revised 31 July 1990*

scheme for certain problems may become totally inadequate under other unfavourable conditions.

We will demonstrate that it is beneficial to rearrange the discretized continuity equations, to generate a set of redundant algebraic equations and to re-compute the velocity field using this set of equations. This set of algebraic equations is redundant because it consists of linear combinations of the continuity equations. Such an algorithm, when used in conjunction with an existing numerical method, is found to actually increase the convergence rate of the solution or to make the otherwise divergent solution convergent.

Although the text description is based on the use of finite difference methods, it is likely that the proposed algorithm can be used in conjunction with other numerical techniques such as the finite element method. Also, for clarity, the figures and most of the equations are two-dimensional. The algorithm, however, is readily extendable to three-dimensional flow problems, as one of the test cases illustrates.

In the following text we first briefly describe an existing primitive-variable formulation. We then present the modified continuity equations and the grid on which these equations will be discretized and computed. Next, the proposed algorithm is described and several test cases and their results are presented. Convergence rates of the solution obtained by using the proposed scheme are compared with those rates obtained without using the proposed algorithm. This comparison is followed by discussion of the results and other relevant issues.

## GOVERNING CONTINUUM EQUATIONS

The time-dependent incompressible Navier–Stokes flows are governed by the continuity equation,

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

the momentum equation,

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{Re} \nabla^2 \mathbf{u} - \nabla p - \mathbf{g}, \tag{2a}$$

and the energy equation,

$$\frac{\partial \theta}{\partial t} + \mathbf{u} \cdot \nabla \theta = \frac{1}{Pe} \nabla^2 \theta, \tag{2b}$$

where $\mathbf{u} = (u, v, w)^{\mathrm{T}}$, $\mathbf{g} = (0, g, 0)^{\mathrm{T}}$ and $Pe$ is the Peclect number defined as $PrRe$. Also, the Boussinesq approximation has been implied in equation (2a).

## A PRIMITIVE-VARIABLE APPROACH

We will first briefly describe the pressure gradient (PG) method,[5–7] an existing primitive-variable formulation applicable to solving equations (1) and (2).

The PG method is based on two strategies: the first establishes how pressure and mass flow can be related; the second leads to the construction of a modified non-staggered grid from a grid originally staggered. With these strategies the pressure gradients can be expressed by the velocity components located within the nine-point stencil. For simplicity we will present our description on a 2D uniform grid. The PG method, however, is applicable to 3D problems with body-fitted co-ordinates as well.

*Relating the pressure with the mass flow*

The net mass flow over the shaded control volume shown in Figure 1 can be approximated by

$$\text{net flow} = \text{out flow} - \text{inflow} = 0{\cdot}5h(u_{\mathrm{NE}} + u_{\mathrm{E}} + v_{\mathrm{NE}} + v_{\mathrm{N}}) - 0{\cdot}5h(u_{\mathrm{N}} + u_{\mathrm{C}} + v_{\mathrm{C}} + v_{\mathrm{E}}).$$

A basic requirement for a working primitive-variable scheme is its capability to find a pressure field where the net flow over every computational cell is zero. Net non-zero flow leaving a computational cell ($\nabla \cdot \mathbf{u}$ is positive) during iterations implies that the iterated pressure within that computational cell is too large, forcing too much flow out from the cell. The magnitude of the true pressure may be less. Subsequently the pressure for the next iteration should be reduced. Therefore we can relate the pressure with the net flow by

$$p^{(k+1)}(x, y) = p^{(k)}(x, y) - \gamma(\text{net flow})/h^2, \quad \gamma > 0, \tag{3}$$

which indeed interprets the aforementioned concept mathematically. The superscript $k$ denotes the $k$th iteration. If only steady state is considered, the iteration step can be viewed the same as the time step. If transient (either true or false) problems are computed, $k$ denotes the iteration number at each time step. Clearly, for negative net flows, similar statements can be made and equation (3) remains valid.

It is reported[8] that at least four additional ways of modifying the continuity equations exist which also lead to equation (3). These modifications, along with their starting equations, are summarized in Table I.

One way to estimate the magnitude of the proportionality $\gamma$ in equation (3) is described in the Discussion section. In this analysis, $\gamma$ is treated as a constant. A similar formulation for variable $\gamma$ can be obtained following the same derivation for constant $\gamma$.

For the square designated by ne in Fig. 1, equation (3) can be written as

$$p_{\mathrm{ne}}^{(k+1)} = p_{\mathrm{ne}}^{(k)} - 0{\cdot}5\gamma(u_{\mathrm{NE}} + v_{\mathrm{NE}} + u_{\mathrm{E}} - v_{\mathrm{E}} - u_{\mathrm{C}} - v_{\mathrm{C}} - u_{\mathrm{N}} + v_{\mathrm{N}})/h. \tag{4}$$

As an illustration, we present an iterated computer output of the nodal pressure and the net mass flow emerging from the computational cell located at $i = 6$ and $j = 6$ in the classical lid-driven cavity flow problem ($Re = 1$, $h = 0{\cdot}1$). At the fourth iteration, $\nabla \cdot \mathbf{u}$ is 0·542, indicating that a net amount of fluid leaves the control volume and that the pressure within the control volume may be too large. According to equation (3), the pressure is decreased by an amount proportional to the net outflow. This correction procedure is repeated to $k = 6$, when the net flow is reversed ($-0.737$),

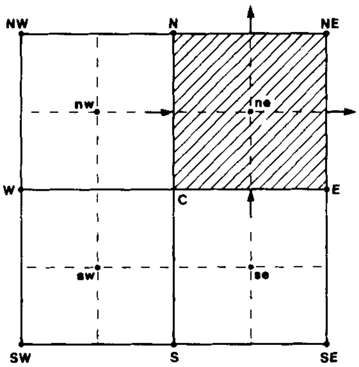


Figure 1. A nine-point computational stencil for the pressure gradient (PG) method

Table I. Four ways of modifying the continuity equation to lead to equation (3), a short description of each modification, and the starting equation(s)

| |
|---|
| *Method of artificial compressibility*[9, 12]<br>The continuity is modified on the basis of the compressible flow assumption.<br>$\partial p/\partial t+\beta\nabla\cdot\mathbf{u}=0.$ |
| *Penalty function method*[11]<br>The pressure is assumed to be a product of a large number and $\nabla\cdot\mathbf{u}$.<br>$p=-\lambda\nabla\cdot u,$<br>where $\lambda$ is a large number. |
| *Source-diminishing method*<br>The computation starts at $a=0$ when mass sources are present in the flow field and when the pressure is uniform (zero) throughout the flow field. The source is forced to diminish when $a$ is gradually increased to unity.<br>$(1-a)p+a\nabla\cdot\mathbf{u}=0.$ |
| *Pressure substitution method*<br>The discretized momentum equations are substituted into the discretized continuity equation to yield equation (3).<br>Equation (6a) and the discretized version of equation (1). |

suggesting that the pressure value has been underestimated and should be increased. After nine iterations, $\nabla\cdot\mathbf{u}$ becomes positive again (0·073), forcing the pressure to reduce. Finally, at 106 iterations, $\nabla\cdot\mathbf{u}$ diminishes to $10^{-6}$ and the pressure solution has converged.

### *Constructing a modified non-staggered grid*

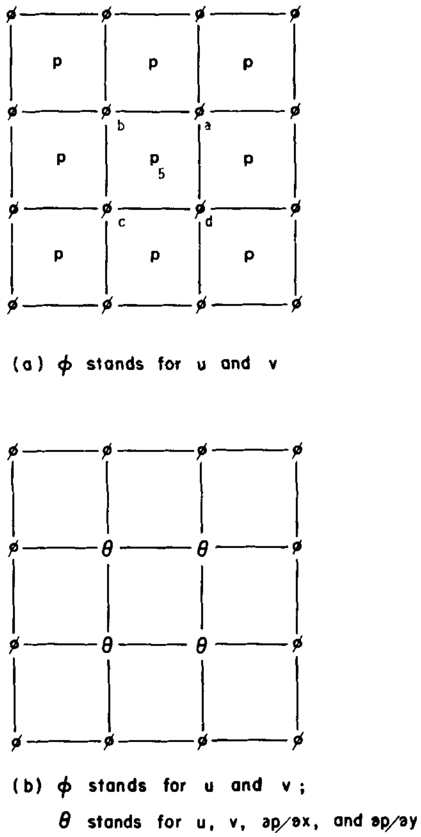
It is the pressure gradient, not the pressure itself, that appears in the Navier–Stokes equations and affects the momentum change. Therefore, for incompressible flows in which an absolute pressure value is immaterial and the equation of state is not needed, it is possible to determine the entire flow velocity field without the knowledge of pressure distribution.

If the pressure gradients can be compactly expressed in terms of the velocity components within the nine-point stencil (for 2D problems), the nodal unknowns $u$, $v$, $\partial p/\partial x$ and $\partial p/\partial y$ will be located at the intersections of grid lines. The grid staggeredness is conveniently eliminated, as depicted in Figures 2(a) and 2(b). Therefore, when equations similar to equation (4) have been written for $p_{\mathrm{nw}}$, $p_{\mathrm{sw}}$ and $p_{\mathrm{se}}$, we can derive, after straightforward algebra,

$$\begin{aligned}(\partial p/\partial x)_{\mathrm{C}}^{(k+1)}&=(p_{\mathrm{ne}}+p_{\mathrm{se}}-p_{\mathrm{nw}}-p_{\mathrm{sw}})^{(k)}/2h\\&=(\partial p/\partial x)_{\mathrm{C}}^{(k)}-\gamma(2u_{\mathrm{E}}+2u_{\mathrm{W}}-4u_{\mathrm{C}}+u_{\mathrm{NE}}+u_{\mathrm{NW}}-2u_{\mathrm{N}}+u_{\mathrm{SE}}\\&\quad+u_{\mathrm{SW}}-2u_{\mathrm{S}}+v_{\mathrm{NE}}+v_{\mathrm{SW}}-v_{\mathrm{NW}}-v_{\mathrm{SE}})/4h^2.\end{aligned}\tag{5a}$$

Similarly, the $y$-direction pressure gradient can be written as

$$\begin{aligned}(\partial p/\partial y)_{\mathrm{C}}^{k+1}&=(p_{\mathrm{ne}}+p_{\mathrm{nw}}-p_{\mathrm{se}}-p_{\mathrm{sw}})^{(k)}/2h\\&=(\partial p/\partial x)_{\mathrm{C}}^{(k)}-\gamma(2v_{\mathrm{N}}+2v_{\mathrm{S}}-4v_{\mathrm{C}}+v_{\mathrm{NE}}+v_{\mathrm{SE}}-2v_{\mathrm{E}}+v_{\mathrm{NW}}\\&\quad+v_{\mathrm{SW}}-2v_{\mathrm{W}}+u_{\mathrm{NE}}+u_{\mathrm{SW}}-u_{\mathrm{NW}}-u_{\mathrm{SE}})/4h^2.\end{aligned}\tag{5b}$$



(a) $\phi$ stands for u and v

(b) $\phi$ stands for u and v ;
$\theta$ stands for u, v, $\partial p/\partial x$, and $\partial p/\partial y$

Figure 2. Change from (a) a staggered grid to (b) a modified non-staggered grid

In writing the computer code, it is unnecessary to distinguish the pressure gradients on the left-hand side from those on the right-side of equations (5a) and (5b). The former is the current value being computed; the latter is the previous iterated value. It can also be proved[6] that equations (5a) and (5b) imply equation (3) as long as the global mass is conserved.

The transport equation of the $x$-direction momentum can be discretized using the implicit scheme as

$$\text{acceleration} + \text{convection} = \text{diffusion} - \text{pressure gradient}, \tag{6}$$

where

$$\text{acceleration} = (u_C - \bar{u}_C)/\mathrm{d}t,$$

with the overbar denoting 'at previous time step',

$$\text{convection} = [-0{\cdot}5(u_C + |u_C|)u_W + |u_C|u_C + 0{\cdot}5(u_C - |u_C|)u_E$$
$$-0{\cdot}5(v_C + |v_C|)u_S + |v_C|u_C + 0{\cdot}5(v_C - |v_C|)u_N]/h,$$

$$\text{diffusion} = (u_W + u_S + u_E + u_N - 4u_C)/h^2 Re$$

and the pressure gradient is given by equation (5a). We then rearrange equation (6) to obtain an underrelaxed expression for $u_C$ as

$$u_C = a[\text{terms not involving } u_C] + (1-a)u_C.$$

Intentionally adding $u_C - u_C$ to the right-hand side for easy algebraic manipulation and adding superscripts for clarity, we obtain

$$u_C^{(k+1)} = u_C^{(k)} + \gamma_u \, \text{res}u^{[(k),(k+1)]}, \tag{6a}$$

where res$u$ is the residual of the discretized $u$-momentum equation, defined as

$$\text{res}u = -\text{acceleration} - \text{convection} + \text{diffusion} - \text{pressure gradient}. \tag{7}$$

The superscript $[(k),(k+1)]$ denotes that unswept unknowns are at $(k)$ and swept unknowns are at $(k+1)$. The underrelaxation parameter is found to be

$$\gamma_u = a \Bigg/ \left( \frac{1}{dt} + \frac{4}{h^2 Re} + \frac{|u_C|^{(k)}}{h} + \frac{|v_C|^{(k)}}{h} \right). \tag{7a}$$

In the present analysis the underrelaxation parameter $a$ is chosen to be between 0·4 and 1·0. The algebraic equation for $v_C$ can be derived similarly. If the second-order-accurate upwind formulation is used, it can be readily shown that

$$\gamma_u = a \Bigg/ \left( \frac{1}{dt} + \frac{4}{h^2 Re} + \frac{1{\cdot}5|u_C|^{(k)}}{h} + \frac{1{\cdot}5|v_C|^{(k)}}{h} \right). \tag{7b}$$

It can also be shown by straightforward algebra that equation (6a) is identical to the result obtained from rearranging equation (6) and then collecting terms containing $u_C$. It is our preference to use equation (6a) rather than the rearranged form because (i) no algebraic manipulation is required prior to coding the computer programme and (ii) the value of res$u$ is eventually needed for testing the convergence status.

Finally, if only the steady state solution is of interest, we can simply set d$t$ to be a large number, say $10^7$, and set the number of time steps to be one, without having to revise the computer code significantly.

## ALGEBRAIC CONTINUITY EQUATION SOLVER (ACES)

It is well known that for ensuring convergence of the solution in computations for incompressible flows governed by the continuity and momentum equations, an iterative scheme such as the Gauss–Seidel method requires that the resulting coefficient matrix be diagonally dominant. The momentum equations alone, when discretized in conjunction with certain upwinding treatments for the convection terms, generally can be cast into a set of diagonally dominant algebraic equations. Therefore, with regard to convergence, no major difficulty asscoiated with computing the momentum equations arises.

Satisfying the continuity constraint, which is responsible for solving the pressure for incompressible flows, is a more challenging task. The constraint not only lacks diagonal dominance in the discretized sense but also contains just the velocity components and not the pressure. In other words, the flow's 'attempt' to conserve its mass for every computational cell generally hinders the convergence process (see Discussion section).

The key to accelerating the convergence rate for computations of incompressible flows lies in manipulating and guiding the velocity components to satisfy the continuity constraint faster.

Taking the gradient of equation (1), we generate three equations as

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 v}{\partial x \partial y}+\frac{\partial^2 w}{\partial x \partial z}=0, \tag{8a}$$

$$\frac{\partial^2 u}{\partial x \partial y}+\frac{\partial^2 v}{\partial y^2}+\frac{\partial^2 w}{\partial y \partial z}=0, \tag{8b}$$

$$\frac{\partial^2 u}{\partial x \partial z}+\frac{\partial^2 v}{\partial y \partial z}+\frac{\partial^2 w}{\partial z^2}=0. \tag{8c}$$

Equations (8a)–(8c) can be used to re-compute $u$, $v$ and $w$ respectively. Although the coefficient matrix of the discretized equations of (8a)–(8c) is not truly diagonally dominant, use of an iterative scheme such as the Gauss–Seidel method generally leads to convergent solutions because of the second-derivative term $\partial^2 u/\partial x^2$ in equation (8a), $\partial^2 v/\partial y^2$ in equation (8b) and $\partial^2 w/\partial z^2$ in equation (8c).

Several legitimate ways to discretize equations (8a)–(8c) exist. For example, referring to Figure 1, which represents a 2D computational cell, we can transform the 2D version of equation (8a) into

$$u_{\mathrm{W}}-2u_{\mathrm{C}}+u_{\mathrm{E}}+\tfrac{1}{4}(v_{\mathrm{NE}}+v_{\mathrm{SW}}-v_{\mathrm{SE}}-v_{\mathrm{NW}})=0, \tag{9a}$$

using the straightforward central finite difference, or into

$$\tfrac{1}{6}(u_{\mathrm{NW}}-2u_{\mathrm{N}}+u_{\mathrm{NE}})+\tfrac{2}{3}(u_{\mathrm{W}}-2u_{\mathrm{C}}+u_{\mathrm{E}})+\tfrac{1}{6}(u_{\mathrm{SW}}-2u_{\mathrm{S}}+u_{\mathrm{SE}})+\tfrac{1}{4}(v_{\mathrm{NE}}+v_{\mathrm{SW}}-v_{\mathrm{SE}}-v_{\mathrm{NW}})=0, \tag{9b}$$

using the Galerkin finite element method with bilinear square elements. Neither equation (9a) nor equation (9b), however, is helpful in accelerating the convergence rate with regard to the 4/1 grid (see Discussion section). The discretized result that contributes most beneficially to the acceleration of convergence is the one derived from algebraically manipulating the discretized continuity equations. For example, the 2D version of equation (8a) should be transformed into

$$\tfrac{1}{4}(u_{\mathrm{NW}}-2u_{\mathrm{N}}+u_{\mathrm{NE}})+\tfrac{1}{2}(u_{\mathrm{W}}-2u_{\mathrm{C}}+u_{\mathrm{E}})+\tfrac{1}{4}(u_{\mathrm{SW}}-2u_{\mathrm{S}}+u_{\mathrm{SE}})+\tfrac{1}{4}(v_{\mathrm{NE}}+v_{\mathrm{SW}}-v_{\mathrm{SE}}-v_{\mathrm{NW}})=0, \tag{10a}$$

which can be readily shown to be the same as the algebraically manipulated result:

$$(\text{c. eq. of cell ne})+(\text{c. eq. of cell se})-(\text{c. eq. of cell nw})-(\text{c. eq. of cell sw})=0, \tag{10b}$$

where c. denotes continuity. Equation (10a) or (10b) can be rearranged to obtain an expression for $u_{\mathrm{C}}$. Similar discretizing procedures apply to equations (8b) and (8c). Since equation (10b) is merely a rearranged form of the original discretized continuity equations, neither additional equations nor additional boundary conditions are introduced.

It is possible to replace equation (10b) with other arrangements, such as adding c. eq. of cell nw and c. eq. of cell sw in a one-sided fashion. However, this choice cannot provide us with a diagonally dominant set of albegraic equations for $u_{\mathrm{C}}$. Neither is it adequate for the situation where the flow is westbound. Equation (10b) is the arrangement that is impartial to the flow direction and at the same time gives the maximum value of the coefficient for $u_{\mathrm{C}}$.

When the geometries of the flow systems are irregular, it is common practice to introduce the body-fitted co-ordinates $\xi$ and $\eta$. In these circumstances the continuity equation in a 2D computational domain becomes

$$y_\eta \frac{\partial u}{\partial \xi}-y_\xi \frac{\partial u}{\partial \eta}+x_\xi \frac{\partial v}{\partial \eta}-x_\eta \frac{\partial v}{\partial \xi}=0, \tag{11a}$$

which can be rearranged into

$$\frac{\partial \tilde{u}}{\partial \xi}+\frac{\partial \tilde{v}}{\partial \eta}=0, \tag{11b}$$

where $\tilde{u}$ and $\tilde{v}$ are the transformed velocity components,[12] defined as

$$\tilde{u}=y_\eta u-x_\eta v, \tag{12a}$$

$$\tilde{v}=x_\xi v-y_\xi u. \tag{12b}$$

With the aid of equations (12a) and (12b), the ACES is also applicable to the problems involved with the body-fitted co-ordinates. For example, in equation (10a) all the velocity components are replaced by their counterparts defined in equations (12a) and (12b).

## NUMERICAL PROCEDURE

The numerical procedure adopted in this study is the point-by-point Gauss–Seidel iteration method with the first- or second-order-accurate upwind difference for the convection terms. The continuity equation is replaced by equation (3) and equations (5a) and (5b) to account for the pressure and the pressure gradients respectively. The computer programme is written in the following logical manner.

1. Specify the input data, such as the values of governing parameters, the mesh size, the convergence criteria, initial guesses, etc.

The cycle of time marching (for true unsteady problems) starts here. For steady state problems, set the number of iterations to one and $dt$ to a large number.

2a. Specify the velocity boundary conditions of Dirichlet type. If the velocity boundary conditions are independent of time, this step can be taken out from the time-marching cycle.

The cycle of outer Gauss–Seidel iteration starts here.

2b. Specify the velocity boundary conditions of Neumann type.
3. Compute nodal $u$s and $v$s using equation (6a). We will call this step the inner Gauss–Seidel iteration. For all runs here, the number of inner iterations is taken to be one.
4. Compute nodal pressures using equation (4).
5. Compute pressure gradients using equations (5a, b).
6. Test the convergence status. If the solution has converged, go to step 8, preparing for the next time step.
7. Use the ACES (equation (10b)) to re-compute the velocity components. If the body-fitted co-ordinates are used, $\tilde{u}$ and $\tilde{v}$ should replace $u$ and $v$. Then $u$ and $v$ should be recovered by rearranging equations (12a) and (12b) as follows:

$$u=(x_\xi \tilde{u}+x_\eta \tilde{v})/J, \tag{13a}$$

$$v=(y_\xi \tilde{u}+y_\eta \tilde{v})/J, \tag{13b}$$

where $J$ is the Jacobian, defined as

$$J=x_\xi y_\eta-x_\eta y_\xi.$$

The cycle for outer Gauss–Seidel iteration ends here. The number of outer iterations is to be used as the indicator of convergence rate.

8. Print the output.
9. For time-dependent computations, return to step 2 for the next time step. If the steady state or a predetermined time duration has been reached, go to step 10.

The cycle for time marching ends here.

10. Stop.

Numerical experiments indicate that in step 7 it is optimal to choose the number of inner iterations (to be distinguished from the outer Gauss–Seidel iteration) to be approximately four. If the number is lower than four, the velocity components will not be sufficiently corrected to satisfy equations (5a) and (5b). If the number is higher than four, computation time will be spent needlessly.

In the present analysis the solution is considered to be convergent and the computation is terminated when the following three criteria are simultaneously satisfied.

(a) Residuals of all the continuity equations reduce to not greater than 0·1% of the maximum residual (in a certain cell) in the first iteration.
(b) Residuals of all the momentum equations reduce to not greater than 0·1% of the maximum residual (at a certain grid point) in the first iteration.
(c) Ratios $(u(i,j,k)-u_p(i,j,k))/u(i,j,k)$ (where $u_p(i,j,k)$ is the previously iterated values at grid point $(i,j,k)$) reduce to not greater than 0·001. Those with absolute values of $u(i,j,k)$ less than 0·005 are not counted. The implementation of this cut-off value eliminates the possibility of needlessly continuing the computation when most of the nodal velocities except a few near-zero ones have converged.

For all the test problems considered here, these criteria appear to be adequate because the present solutions differ insignificantly from those obtained under much more stringent conditions.

The underrelaxation parameter $\gamma$ is designated as

$$\gamma = 2{\cdot}2/Re.$$

The derivation of this estimate is given in the Discussion section. For the buoyancy-driven cavity flow, $Re$ is taken to be unity. For Euler flows where $Re$ is infinity, we have used a pressure gradient correction procedure to enforce the continuity constraint. This topic is currently under investigation and may be published elsewhere.

## TEST CASES

### *Lid-driven cavity flow*

The computer programme, without the ACES, has been tested against various benchmark problems.[13,14] For example, for the classical lid-driven cavity flow[15] at $Re = 400$, the solution computed here on a grid of 50 × 50 compares favourably with the literature result,[16] as shown in Table II(a). This solution is obtained using the second-order-accurate upwind difference and assuming the flow velocity at the two top corners to be zero (contained flow). The purpose of conducting this exercise and presenting this table is to suggest that our computer code can be believed to be free of errors and hence that the following observations are deemed somewhat reliable (at least for the chosen test problems).

Table II(a). Velocity distributions at $x = 0·5$ for the lid-driven cavity flow ($Re = 400$). The subscript H denotes Harwell Report

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $y$ | 0·02 | 0·04 | 0·06 | 0·08 | 0·1 | 0·12 | 0·14 | 0·16 |
| $u_H$ | −0·0311 | −0·0597 | −0·0871 | −0·1140 | −0·1408 | −0·1681 | −0·1957 | −0·2230 |
| $u$ | −0·0312 | −0·0604 | −0·0884 | −0·1162 | −0·1444 | −0·1734 | −0·2029 | −0·2325 |
| $y$ | 0·18 | 0·2 | 0·22 | 0·24 | 0·26 | 0·28 | 0·3 | 0·32 |
| $u_H$ | −0·2494 | −0·2736 | −0·2946 | −0·3104 | −0·3204 | −0·3244 | −0·3220 | −0·3130 |
| $u$ | −0·2610 | −0·2871 | −0·3092 | −0·3259 | −0·3362 | −0·3394 | −0·3355 | −0·3251 |
| $y$ | 0·34 | 0·36 | 0·38 | 0·4 | 0·42 | 0·44 | 0·46 | 0·48 |
| $u_H$ | −0·2986 | −0·2805 | −0·2593 | −0·2358 | −0·2115 | −0·1869 | −0·1624 | −0·1383 |
| $u$ | −0·3093 | −0·2892 | −0·2663 | −0·2417 | −0·2163 | −0·1910 | −0·1659 | −0·1414 |
| $y$ | 0·5 | 0·52 | 0·54 | 0·56 | 0·58 | 0·6 | 0·62 | 0·64 |
| $u_H$ | −0·1146 | −0·0912 | −0·0682 | −0·0452 | −0·0223 | 0·0009 | 0·0243 | 0·0478 |
| $u$ | −0·1174 | −0·0938 | −0·0704 | −0·0471 | −0·0238 | −0·0003 | 0·0235 | 0·0476 |
| $y$ | 0·66 | 0·68 | 0·7 | 0·72 | 0·74 | 0·76 | 0·78 | 0·8 |
| $u_H$ | 0·0716 | 0·0957 | 0·120 | 0·1440 | 0·1679 | 0·1917 | 0·2146 | 0·2364 |
| $u$ | 0·0720 | 0·0967 | 0·1217 | 0·1467 | 0·1716 | 0·1962 | 0·2203 | 0·2436 |
| $y$ | 0·82 | 0·84 | 0·86 | 0·88 | 0·9 | 0·92 | 0·94 | 0·96 |
| $u_H$ | 0·2575 | 0·2777 | 0·2974 | 0·3199 | 0·3486 | 0·3983 | 0·4798 | 0·6104 |
| $u$ | 0·2661 | 0·2880 | 0·310 | 0·3347 | 0·3672 | 0·4166 | 0·4969 | 0·6227 |
| $y$ | 0·98 | 1 | | | | | | |
| $u_H$ | 0·7950 | 1 | | | | | | |
| $u$ | 0·7978 | 1 | | | | | | |

Table II(b). Errors of velocity distributions at $x = 0·5$ for the lid-driven cavity ($Re = 400$) in comparison with the result of Harwell[16]

| $n$ | 20 | 40 | 60 | 80 | Harwell |
|---|---|---|---|---|---|
| Error | 0·133 | 0·068 | 0·029 | 0·016 | 0·0 |

It is reported[17] that use of the 4/1 grid, i.e. with four nodal velocities at the vertices and one nodal pressure at the centre of a computational cell, leads to two constraints on the velocity boundary conditions: one on the velocities normal to the boundary and the other on the velocities tangential to the boundary. The problem of contained lid-driven cavity flows violates the second constraint and therefore should not be solved on the 4/1 grid.

For the reason mentioned above, the present primitive-variable approach is slightly modified to eliminate the second constraint and to yield a set of consistent algebraic continuity equations. Briefly, near the top of the boundary we introduce a set of nodal *u*s (positioned at one-half grid size below the lid) as additional nodal unknowns. They are solved by using the momentum equations (instead of being the average of neighbouring *u*s) and are used in the continuity equation. With this treatment the resulting coefficient matrix is associated with only one zero eigenvalue.

The numerical error introduced by the scheme is estimated by computing

$$\text{error} = \sum [u(0·5, y) - u_{\text{Harwell}}(0·5, y)]^2$$

Table III. Number of iterations for lid-driven cavity flows ($n = 20$)

| *Re* | 0 | 1 | 10 | 100 | 400 | 1000 |
|---|---|---|---|---|---|---|
| PG + ACES | 167 | 168 | 179 | 298 | 591 | 1116 |
| PG | 399 | 409 | 386 | 348 | 630 | * |

* denotes 'non-convergent'.

Table IV. Number of iterations for lid-driven cavity flows ($Re = 400$)

| *n* | 2 | 5 | 10 | 20 | 50 |
|---|---|---|---|---|---|
| PG + ACES | 14 | 270 | 455 | 591 | 738 |
| PG | 20 | * | 539 | 630 | * |

* denotes 'non-convergent'.

for the lid-driven cavity flow at $Re = 400$ at various grid sizes ($n = 1/h$) and is listed in Table II(b).

When the computer code is implemented with the ACES, the solution is found to be identical to that obtained without using the ACES, as expected. Having established this identity, we then are in the position to place our emphasis on the comparison of convergence rates.

*Varying Reynolds numbers for lid-driven cavity flow*

On a grid of $20 \times 20$ we compute the lid-driven cavity flow for $Re = 0$ (representing Stokes flow), 1, 10, 100, 400, 1000 and 10000. The number of iterations (NI) required for the solution to achieve convergence is obtained either with the ACES or without the ACES and the comparison is shown in Table III. It is noted that for steady state problems the present procedure without the ACES is equivalent to Chorin's Artificial Compressibility Method (ACM). Also, the CPU time of a typical run on a SUN IV workstation is found to be approximately 8 s.

The ACES consistently results in faster convergence than the ACM for *Re* ranging from zero to 10000. At $Re = 1000$ the solution computed by using the ACM did not converge at the 2000th iteration and most likely will not converge regardless of the allowed iteration number. Although it may be possible for the ACM to attain convergence by reducing the relaxation factors in the continuity equation and in the momentum equations, the NI will be considerably higher than 2000. For example, with $a = 0{\cdot}1$ and $\gamma = 2{\cdot}2/Re$, and at the 2000th iteration, there are 36 cell continuity residuals (out of 400) and 45 nodal momentum residuals (out of 361) whose values remain greater than the predetermined criterion value.

*Varying mesh sizes for lid-driven cavity flow*

At $Re = 400$ we also vary the mesh size for $n = 2, 5, 10, 20$ and 50 for the lid-driven cavity flow and tabulate the NI for both methods in Table IV. The NI for the ACES is again uniformly smaller than that for the ACM. On finer grids ($n > 40$) the solution computed by the ACM did not converge at the 2000th iteration.

*A recirculating flow with mass crossing the system boundary*

To assess the ACES against problems with inflows and outflows crossing the system boundary, we apply it to the problem of a recirculating flow whose computed velocity vector plot ($Re = 100$) is shown in Figure 3. A single jet enters the enclosure at the lower portion of the left boundary

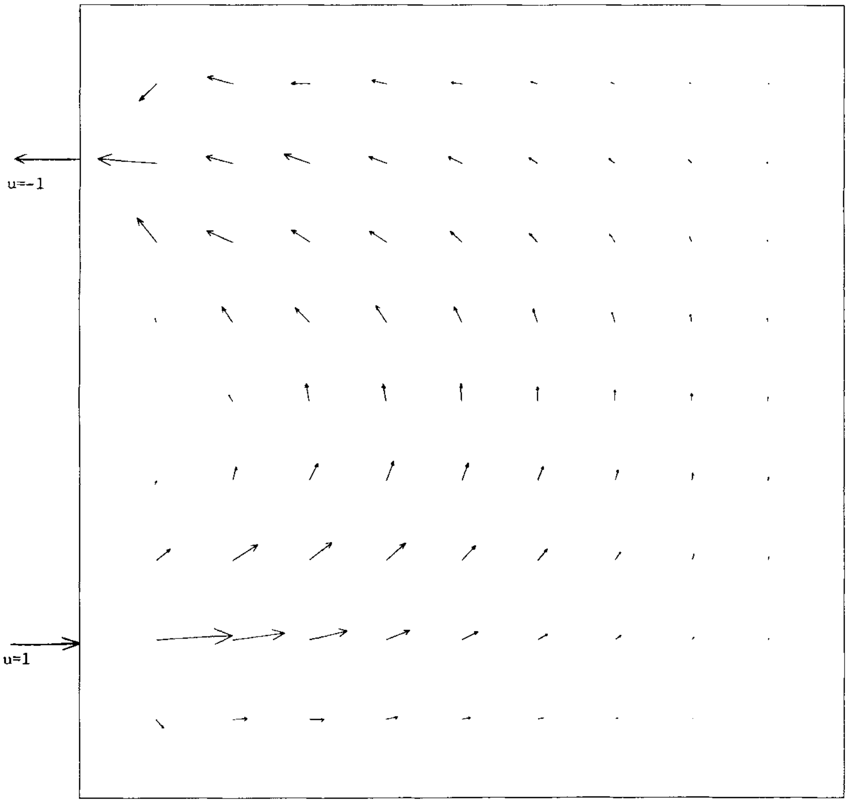


Figure 3. Velocity vector plot of a jet-driven recirculating flow

Table V. Number of iterations for a jet-driven recirculating flow

| *Re* | 0 | 1 | 10 | 100 | 400 | 1000 |
|---|---|---|---|---|---|---|
| PG+ACES | 173 | 173 | 175 | 202 | 277 | 2227 |
| PG | 161 | 161 | 184 | 221 | 387 | * |

* denotes 'non-convergent'.

($j=3$), turns around and leaves the system at $j=9$. Such a flow is different from the first three test cases, in which no fluid flows into and out from the system. As a test problem it is also more stringent than the one-direction, developing-to-fully-developed channel flow.

The NIs are obtained by using both the ACES and the ACM and are listed in Table V for $nx = 30, ny = 10$ and various Reynolds number based on the height of the enclosure. In the very-low-*Re* regime, where the viscous effect dominates, it is fairly easy for the solution to converge and the ACES appears redundant. For larger *Re*s the implementation of the ACES reduces NI and at $Re = 1000$ makes the otherwise non-convergent solution convergent.

*Buoyancy-driven cavity flows*

Another benchmark flow problem is the buoyancy-driven cavity flow[18] schematically shown in Figure 4. The differences between this problem and the lid-driven cavity flow are the existence of the body force in the flow field and the coupling of the energy equation with the momentum equation. It is interesting to further examine if the ACES remains effective in accelerating the convergence rate. On a grid of 10 × 10 we obtain the numerical solution for $Gr = gL^3(T_h - T_c)/v^2 T_c = 10, 100, 1000, 10000$ and 100000. The NIs obtained using either the ACES or the ACM are presented in Table VI.

For low *Gr*s, when the solution can easily converge, the ACES is hardly helpful. For high *Gr*s, however, the ACES makes the otherwise divergent solution converge. It is noted that for $Gr = 100000$ the relaxation parameter $a$ is taken to be 0·7. Although it is possible to obtain a convergent solution using the ACM method with very low values of $a$, the NIs are generally more than twice the number attained by the ACES if the former solution does converge.

*A 3D unsteady flow within an irregular configuration*

The flow problems in the previous cases are steady and more or less academic. To further test the ACES for solving more complex flows, we apply it to a 3D unsteady flow problem, simulating gases leaving an annular diffuser, entering the dump-type collector box, making a sharp turn upwards and exiting from the top plane. The system schematic is shown in Figure 5. The upward velocity component at the exit plane is plotted in Figure 6. Relevant data input are $Re = 200000$, $nx = 20$, $ny = 20$, $nz = 5$ and $t = 1, 5$ and 10. A simplified zero-equation turbulence model is used; the inlet velocity profile at the entrance of the dump-type collector is assumed to be of quadratic form.

When the body-fitted co-ordinates are used, the global mass balance in the computational domain, not the physical domain, must be guaranteed. In other words, we must ensure that the

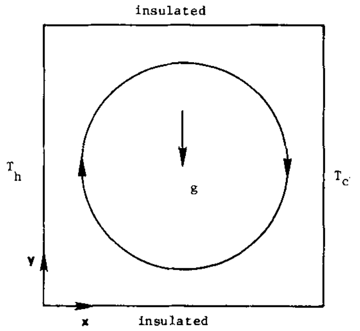


Figure 4. The buoyancy-driven benchmark problem

Table VI. Number of iterations for buoyancy-driven cavity flows ($n = 10$)

| *Gr* | 10 | 100 | 1000 | 10000 | 100000 |
|---|---|---|---|---|---|
| PG+ACES | 171 | 171 | 173 | 259 | 945 |
| PG | 151 | 151 | 173 | * | * |

* denotes 'non-convergent'.

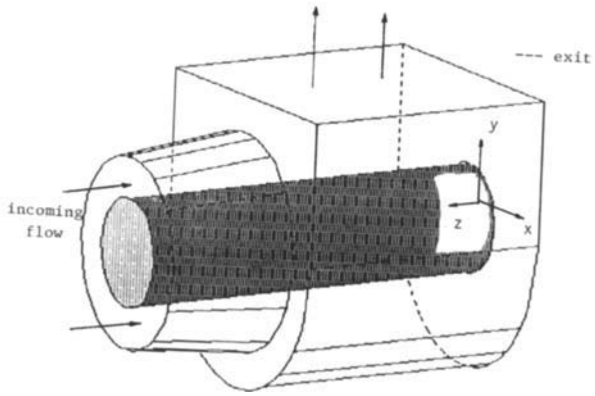
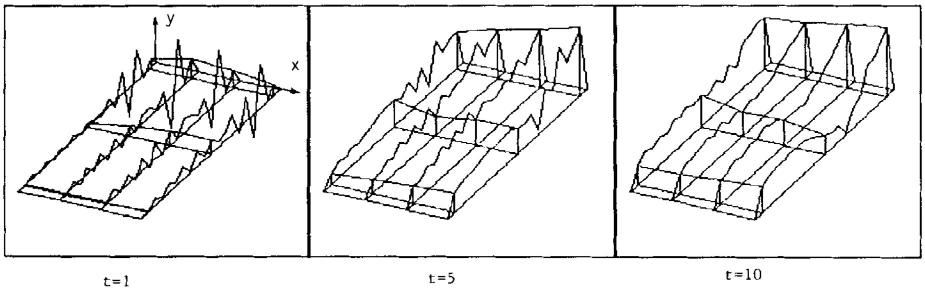


Figure 5. System schematic of a 3D flow in a dump-type collector box



t=1

t=5

t=10

Figure 6. Unsteady exit flow velocity at three different time steps

sum of the transformed normal velocity, defined in equations (12a) and (12b), be zero on the boundary. For example, if the Neumann boundary condition is used for the outflow at the exit, we should use

$$\partial\tilde{v}/\partial\eta = 0$$

but not

$$\partial v/\partial y = 0.$$

Although the flow is clearly 3D, the configuration of the system (representing only the dump-type collector) is independent of the $z$-co-ordinate, permitting the use of the 2D body-fitted co-ordinates. Without the ACES inserted in the computer code, the solution has failed to converge.

## DISCUSSION

The reason why the ACES leads to fast convergence or robustness for large-convection flows can be offered as follows. Consider a fictitious lid-driven cavity flow satisfying the momentum equations and a modified continuity equation

$$\nabla\cdot\mathbf{u} + S = 0,$$

where $S$ denotes a mass source (or sink), so distributed that the pressure gradients vanish everywhere in the flow field. Applying the ACM to this problem in conjunction with the first-order upwind difference at $Re = 400$ and $nx = ny = 20$, we obtain a convergent solution at the

134th iteration. The solution converges rapidly primarily because the continuity constraint is not needed and therefore the coefficient matrix of the discretized momentum equations alone is diagonally dominant.

When $S$ is absent, the pressure gradients are no longer zero. Now the continuity constraint should be imposed and the iteration number drastically increases to 630. This increase supports our conjecture that the flow is 'trying very hard' to meet the mass conservation criterion. In a non-rigorous sense we may state that approximately 80% of the effort of a numerical method is devoted to satisfying the continuity equation. Therefore, if a scheme is capable of satisfying the continuity equation rapidly, it is likely that this scheme can make the solution converge rapidly as well.

Table VII displays the maximum values of residuals of the discretized continuity equations, resc, and those of the discretized $x$-direction momentum equation, res$u$, at various iterations for the lid-driven cavity flow at $Re = 10$ and $n = 20$. It can be seen that at the same iteration cycle the values obtained using the ACES are considerably lower than those obtained without using the ACES.

In the remainder of this section we will discuss several issues pertaining to the use of ACES.

### *Other grid systems*

The concept of the ACES can be extended to other types of mesh layouts such as the staggered grid shown in Figure 7(a) and the non-staggered grid shown in Figure 7(b). Referring to Figure 7(a), we write the continuity equations responsible for $p_1$ and $p_2$ as

$$u_C - u_W + v_{NW} - v_{SW} = 0, \tag{14a}$$

$$u_E - u_C + v_{NE} - v_{SE} = 0, \tag{14b}$$

Subtracting equation (14b) from equation (14a) and rearranging the result, we obtain

$$u_C = \tfrac{1}{2}(u_E + u_W + v_{NE} + v_{SW} - v_{SE} - v_{NW}) = 0. \tag{15a}$$

Similarly, on the non-staggered grid shown in Figure 7(b) we can also derive an expression for $u_C$ as

$$u_C = \tfrac{1}{2}(u_{EE} + u_{WW} + v_{NE} + v_{SW} - v_{SE} - v_{NW}). \tag{15b}$$

Unlike equation (10a) of the ACES and equation 15(a) of the staggered grid, equation (15b) contains nodal velocity components located beyond the nine-point computational stencil of $u_C$. Therefore, when the velocity components adjacent to the boundary (such as $u_W$) are computed, fictitious variables (such as $u_f$) outside the computational domain must also be found by extrapolation.

Table VII. Comparison of residuals of continuity equation (resc) and residuals of $u$-momentum equation (res$u$)

| Iteration | resc(PG) | resc(PG+ACES) | res$u$(PG) | res$u$(PG+ACES) |
|---|---|---|---|---|
| 40 | 1·492 | $2{\cdot}028\times10^{-2}$ | 12·225 | 0·221 |
| 80 | 1·141 | $2{\cdot}193\times10^{-3}$ | 11·601 | $2{\cdot}341\times10^{-2}$ |
| 120 | 0·126 | $7{\cdot}667\times10^{-4}$ | 0·767 | $7{\cdot}674\times10^{-3}$ |
| 160 | $2{\cdot}698\times10^{-2}$ | $3{\cdot}925\times10^{-4}$ | $4{\cdot}368\times10^{-2}$ | $3{\cdot}983\times10^{-3}$ |
| 200 | $1{\cdot}009\times10^{-2}$ | $0(10^{-5})$ | $3{\cdot}952\times10^{-2}$ | $0(10^{-4})$ |

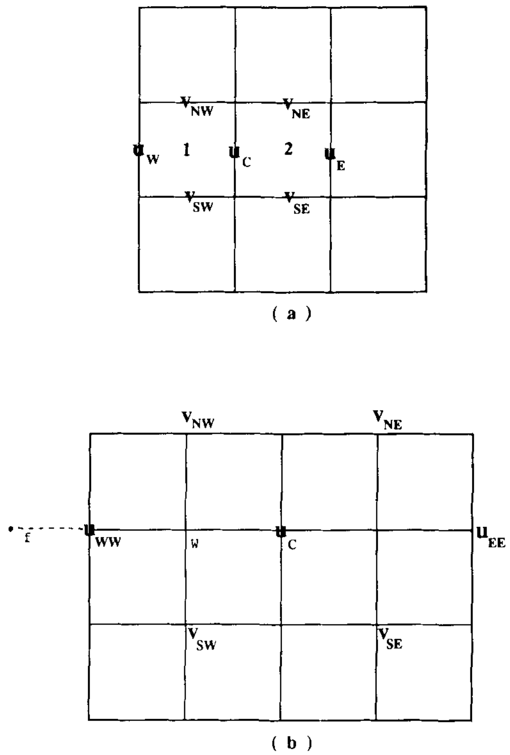


Figure 7. (a) A staggered grid. (b) A non-staggered grid

*Inadequacy of equation (9a)*

Equation (9a) is of simpler form than equation (10a). Naturally, it would be desirable to use the simpler form in the ACES. Unfortunately, this attempt leads to global imbalance of mass and violates the compatibility condition for the pressure, as will be explained below.

The mass balance over the shaded control volume shown in Figure 8 yields

$$u_E - u_C + v_{ne} - v_{se} = 0. \tag{16}$$

If we make a reasonable assumption that

$$v_{ne} = (v_{NE} + v_N + v_C + v_E)/4$$

and

$$v_{se} = (v_E + v_C + v_S + v_{SE})/4,$$

equation (16) becomes

$$u_E - u_C + (v_{NE} + v_N - v_S - v_{SE})/4 = 0. \tag{17a}$$

Similarly, we can also obtain

$$u_C - u_W + (v_N + v_{NW} - v_{SW} - v_S)/4 = 0. \tag{17b}$$

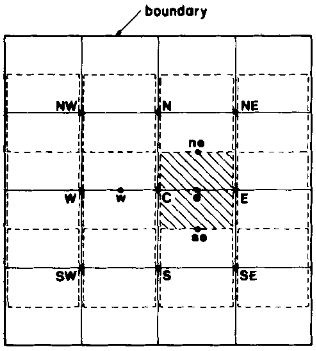


Figure 8. A deficient grid in which the control volumes for the mass conservation do not cover the computational domain entirely

Subtracting equation (17b) from equation (17a) leads to equation (9a), which we thought might be used to compute $(\partial p/\partial x)_C$. However, in Figure 8 the computational domain will be incompletely covered by all the control volumes (drawn as the dashed line). Therefore the boundary conditions on the top and bottom cannot enter the problem.

More importantly, it is clear that $P_e$ and $P_w$ will not be used to compute $(\partial p/\partial y)_C$, suggesting that $(\partial p/\partial x)_C$ and $(\partial p/\partial y)_C$ are almost unrelated. In other words, it is likely that the algebraic version of the compatibility condition for the pressure $(\nabla \times \nabla p = 0)$,

$$\left(\frac{\partial p}{\partial x}\right)_N - \left(\frac{\partial p}{\partial x}\right)_S = \left(\frac{\partial p}{\partial y}\right)_E - \left(\frac{\partial p}{\partial y}\right)_W,$$

is violated. Such violation indicates that we have unknowingly introduced some mysterious body forces into the Navier–Stokes equations.

### *Time saving for computation of complicated flow problems*

Generally, more complicated flow problems are governed by more transport equations. For example, for computations of turbulent fires (as opposed to laminar plumes) we need both larger numbers of transport equations and more terms in all transport equations to account for the turbulence, combustion and radiation. Between step 3 and step 4 of the numerical procedure we need to insert DO loops for computing turbulence quantities, species mass fractions, radiation intensities, etc. In this case, implementation of the ACES constitutes a relatively small portion of the entire cycle of the Gauss–Seidel outer iteration.

### *Usefulness in debugging computer codes and in studying the convergence rate*

A basic Navier–Stokes computer code generally consists of two major parts: one part solving the momentum equations and the other solving the continuity equation. One way to debug a computer code or to analyse the convergence rate of this code is to intentionally separate these two parts and to solve either part alone.

When solving the continuity equation alone, we assume that a non-uniform external body force exists and that the magnitude of this force is balanced by the residual of the momentum equation. When solving the momentum equation alone, we assume that a non-uniform mass source exists and that the pressure gradients vanish in the flow field. In the former case, unfortunately, it is difficult to solve the original continuity equation using an iterative method since the discretized version is not diagonally dominant.

With the aid of the ACES algorithm we produce equation (10a) or (10b) by rearranging the continuity equations, and these two equations behave convergently when solved alone by using the Gauss–Seidel method. Therefore it is now possible to check, for example, if the continuity equation is coded correctly and if the boundary condition of the flow velocity is prescribed properly. If they are, it is our experience that generally the 'solution' (incorrect, of course) converges very rapidly.

### *Specification of $\gamma$*

The choice of a proper value for $\gamma$ is crucial, since the solution diverges if $\gamma$ is too large and converges too slowly if $\gamma$ is too small. It is proposed in this analysis that the value of $\gamma$ be reasonably estimated by substituting the momentum equations in the continuity equation. For clarity, we will present the derivation on a uniform grid for Stokes flows. The derivation on a non-uniform and/or irregular grid for Navier–Stokes flows is algebraically more complicated but conceptually the same

The discretized $x$- and $y$-direction momentum equations at grid point a shown in Figure 2(a) can be rewritten respectively as

$$\left(\frac{4}{h^2Re}\right)u_a = \frac{p_5}{2h} + H_a, \tag{18a}$$

$$\left(\frac{4}{h^2Re}\right)v_a = \frac{p_5}{2h} + G_a, \tag{18b}$$

where $H_a$ is the collection of all terms not involved with $p_5$ or $u_a$, and $G_a$ is that of all terms not involved with $p_5$ or $v_a$. Similar equations (18c)–(18h) can be derived at grid points $b$, $c$ and $d$. Substituting equations (18a)–(18h) in

$$u_a + v_a - u_b + v_b - u_c - v_c + u_d - v_d = 0, \tag{19}$$

we obtain

$$p_5 = -\frac{h}{4}(H_a + G_a - H_b + \ldots - G_d). \tag{20}$$

Utilizing equations (18a)–(18h) to eliminate $H_a$, $G_a$, etc. and intentionally adding a term $p_5 - p_5$ to the right-hand side of equation (20), we obtain

$$p_5 = p_5 - \frac{1}{hRe}(u_a + v_a - u_b + v_b - u_c - v_c + u_d - v_d). \tag{21}$$

Comparing equation (21) with the continuity equation for $p_5$, i.e.

$$p_5 = p_5 - \frac{\gamma}{2h}(u_a + v_a - u_b + v_b - u_c - v_c + u_d - v_d),$$

we find that

$$\gamma = 2/Re.$$

In the present analysis we have used 2·2 instead of 2 because the former leads to slightly faster convergence rates. Derivation of non-uniform $\gamma$ can be conducted straightforwardly for Navier–Stokes flows in irregular geometries but is considered beyond the scope of the present work.

## CONCLUSIONS

On the basis of our present analysis, we find that use of the Algebraic Continuity Equation Solver (ACES), which is constructed from rearranging the discretized continuity equations, increases the convergence rate or makes otherwise divergent computations convergent.

We have applied this concept to various flow problems representing a wide range of flow conditions and geometries. The results, except those of low-Reynolds-number flows, consistently support the above finding. Future work should include conducting more tests and improving the algorithm such that convergence rates for low-Reynolds-number flows increase.

ACKNOWLEDGEMENT

This work is sponsored by David Taylor Research Center, Annapolis. Part of the computer time is provided by the San Diego Supercomputer Center. Communication with Dr. C. H. Tan is enlightening.

## REFERENCES

1. T. M. Shih, 'A survey on numerical heat transfer', *Numer. Heat Transfer*, **5**, 369–420 (1982).
2. T. M. Shih, 'A survey on numerical heat transfer (1982–83)', *Numer. Heat transfer*, **8**, 1–24 (1985).
3. T. M. Shih, 'A survey on numerical heat transfer (1984–85)', *Numer. Heat transfer*, **10**, 1–29 (1987).
4. R. Teman, *Navier–Stokes Equations*, North-Holland, Amsterdam, 1979.
5. T. M. Shih, C. H. Tan and B. C. Hwang, 'Effects of grid staggering on numerical schemes', *Int. j. numer. methods fluids*, **9**, 193–212 (1989).
6. T. M. Shih and B. C. Hwang, 'Pressure gradient method for incompressible unsteady flows ', *Technical Report DTRC-PAS-88-12*, 1988.
7. C. H. Tan, 'A numerical scheme for solving incompressible Navier–Stokes equations', *Ph.D. Dissertation.*, Mechanical Engineering Department, University of Maryland, 1988.
8. T. M. Shih, B. C. Hwang and C. H. Tan, 'The pressure equation revisited and its applications to electronic packaging', Presented at *ASME Annual Winter Meeting*, Dallas, TX, 1990.
9. A. J. Chorin, 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.*, **2**, 12–26 (1967).
10. E. Turkel, 'Preconditioned method for solving the incompressible and low speed compressible equations, *NASA ICASE Report 86-14*, 1986.
11. T. J. R. Hughes, W. K. Liu and A. Brooks Jr., 'Finite element analysis of incompressible viscous flows by the penalty function formulation', *J. Comput. Phys.*, **30**, 1–30 (1979).
12. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation*, North-Holland, Amsterdam, 1985.
13. G. J. Yoo and T. M. Shih, 'Numerical Predictions of turbulent flows using ACES' *Technical Report 88-T2*, SET, Inc., 1988.
14. G. J. Yoo, 'Turbulent flows in diffusers with shaft', *Technical Report 89-T3*, SET, Inc., 1989.
15. O. R. Burggraf, 'Analytical and numerical studies of the structure of steady separated flows', *J. Fluid Mech.*, **24**, 113–151 (1966).
16. K. H. Winters and K. A. Cliffe, 'A finite element study of driven laminar flow in a square cavity', *Technical Report, AERE-R9444*, Harwell, 1979.
17. P. M. Gresho and R. L. Sani, 'On pressure boundary conditions for the incompressible Navier–Stokes equation', *Int. j. numer. methods fluids*, **7**, 1111–1145 (1987).
18. G. De Vahl Davis and I. P. Jones, 'Natural convection in a square cavity', *Int. j. numer. methods fluids*, **3**, 227–240 (1983).