INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 8, 897–912 (1988)

A SIMPLE NUMERICAL TECHNIQUE FOR TRANSIENT CREEP FLOWS WITH FREE SURFACES

MEHDI GOLAFSHANI

Morton Thiokol, Inc., Wasatch Operations, P.O. Box 524, Mail Stop 280, Brigham City, Utah 84302, U.S.A.

SUMMARY

A simple, but powerful iterative technique is presented for the numerical solution of the time-dependent flow of an incompressible viscous fluid with or without a free surface. The usual numerical stability restrictions related to the viscous acceleration terms are avoided using standard implicit differencing techniques. The properties and accuracy of the method are illustrated by several calculational examples.

KEY WORDS Free surface Iterative Algorithm Numerical Transient Fluid flow

INTRODUCTION

There exist many fluid flow problems of practical importance in which the viscous forces are dominant. Several numerical techniques have been developed for the solution of incompressible viscous flow problems.[1,2,9] These methods, however, apply mainly to problems for which the inertial forces are larger than the viscous forces, i.e. the Reynolds number is much greater than one. Owing to limitations related to numerical stability restrictions, these methods cannot be efficiently applied to flow problems for which $Re \ll 1$. The method devised by Pracht[3] was especially suited for the low-Reynolds-number range. However, it is somewhat difficult to implement for general-purpose flows including free surfaces and there is no evidence of its extensive use in the literature.

This report describes a technique that has been developed by the author to investigate incompressible flow in the low-Reynolds-number range. It has been developed as an extension to the SOLA–VOF[2] program and, owing to its use of full Navier–Stokes equations, it can be equally applied to flow problems in the intermediate-Reynolds-number range. This new technique is called the 'cell iterative adjustment technique', from hereon termed CIAT.

In order to see the advantages of the CIAT algorithm, a brief review of the deficiencies of previous techniques[1,2,9] is in order. In all of these techniques viscous stresses are modelled in an explicit way. In any explicit prediction process of this kind there is a limit to the size of δt for which computed solutions will remain accurate and numerically stable. In fact, for a constant kinematic viscosity v, the time step must be limited to satisfy the inequality

$$\delta t < \delta x^2\, \delta y^2 / 2v(\delta x^2 + \delta y^2), \tag{1}$$

where δx and δy are the mesh intervals in the x- and y-directions respectively.

To demonstrate a consequence of this limit, we can rewrite equation (1) as

$$N_t > 2(N_x^2 + N_y^2)/Re, \tag{2}$$

0271–2091/88/080897–16$08.00

Received 9 July 1987
Revised 23 October 1987

where N_t is the number of time steps, each of duration δt, which is required to carry the solution to a characteristic time $T = N_t \delta t$ and N_x and N_y are the number of mesh cells used to resolve a characteristic flow dimension L in the x- and y-directions respectively ($L = N_x \delta x$ and $L = N_y \delta y$). Generally N_x and N_y must be of the order of 10 or greater to have adequate resolution. Thus, for low Reynolds numbers, the number of time steps, N_t, needed to calculate a characteristic flow time can be quite large. In fact, if one is interested in studying a certain fluid flow at $Re = 0{\cdot}01$ with a minimum numbers of mesh cells in the x- and y-directions (i.e. $N_x = N_y = 10$), equation (2) indicates that $N_t > 40000$ and therefore to study the flow up to a characteristic time of $T = 1$ s the numerical solution has to be performed with a time step of the order of 10^{-5} s. This would be a long calculation by most standards, requiring two or three hours of computer run time, and would not be suitable for extensive parametric studies.

It is now obvious that removal of the stability requirement given by equation (1) is of utmost importance for the numerical modelling of low-Reynolds-number flows. In order to do this, the viscous stresses must be modelled in an implicit way. To solve the implicit coupled Navier–Stokes equations requires a rather lengthy numerical procedure, but this is mitigated by the larger allowable δt. It is this type of implicit treatment of the Navier–Stokes equations that makes practical the solution of low-Reynolds-number flows. Further, the CIAT algorithm provides an accurate and efficient solution to such problems.

EQUATIONS OF MOTION

The differential equations to be solved are written in terms of Cartesian co-ordinates (x, y). For cylindrical co-ordinates (r, z) the x-co-ordinate is interpreted as the radial direction and the y-co-ordinate is transformed to the axial co-ordinate. In the following $\xi = 0$ corresponds to Cartesian geometry while $\xi = 1$ corresponds to cylindrical geometry.

The mass continuity equation for a constant density fluid is

$$u_x + v_y + \xi u/x = 0, \tag{3}$$

where the velocity components (u, v) are in the co-ordinate directions (x, y) or (r, z). The equations of motion for the fluid velocity components in the two co-ordinate directions (u, v) are the Navier–Stokes equations

$$\rho[u_t + (uu)_x + (uv)_y + \xi uu/x] = -p_x + \rho g_x + f_x, \tag{4}$$

$$\rho[v_t + (vu)_x + (vv)_y + \xi uv/x] = -p_y + \rho g_y + f_y. \tag{5}$$

In these equations (g_x, g_y) are body accelerations, ρ is the fluid density and (f_x, f_y) are the viscous accelerations. For a variable dynamic viscosity μ the viscous accelerations are

$$f_x = -[(\tau_{xx})_x + (\tau_{xy})_y + \xi(\tau_{xx} - \tau_e)/x], \tag{6}$$

$$f_y = -[(\tau_{xy})_x + (\tau_{yy})_y + \xi \tau_{xy}/x], \tag{7}$$

where

$$\tau_{xx} = -2\mu u_x, \qquad \tau_{yy} = -2\mu v_y, \qquad \tau_{xy} = -\mu(u_y + v_x), \qquad \tau_e = -2\mu u/x.$$

In this report we are only interested in the solution of highly viscous, Newtonian fluid flows, i.e. constant viscosity. Thus equation (4) and (5) can be written as

$$\rho[u_t + (uu)_x + (uv)_y + \xi uu/x] = -p_x + \rho g_x + \mathrm{VISX}, \tag{8}$$

$$\rho[v_t + (vu)_x + (vv)_y + \xi uv/x] = -p_y + \rho g_y + \mathrm{VISY}, \tag{9}$$

where

$$\text{VISX} = \mu[2u_{xx} + (u_y + v_x)_y + 2\xi(u/x)_x],$$

$$\text{VISY} = \mu[2v_{yy} + (u_y + v_x)_x + \xi(u_y + v_x)/x].$$

Fluid configurations for free surface calculations are defined in terms of a volume-of-fluid (VOF) function $F(x, y, t)$. This function is defined in such a way that its value is unity at any point occupied by the fluid and zero elsewhere. When averaged over a computational cell, the value of F is equal to the fractional volume of the cell occupied by fluid. In particular, a unit value of F corresponds to a cell full of fluid, whereas a zero value indicates that the cell contains no fluid. Cells with F-values between zero and one contain a free surface.

The time dependence of F is governed by the kinematic equation

$$F_t + (Fu)_x + (Fv)_y + \xi Fu/x = 0. \tag{10}$$

This equation states that F moves with the fluid.

FINITE DIFFERENCE FORMULATION

A first step toward the solution of the governing equations is to choose a finite difference grid system. A staggered grid system of variable rectangular cells of width δx_i and height δy_j is used (Figure 1).

As shown for a typical cell in Figure 2, u-velocities are stored at the centres of cell faces normal to the x-direction and v-velocities are stored at the centres of cell faces normal to the y-direction. Pressures p and fluid volumes F are stored at the cell centres.

A desired finite difference approximation to the governing equations is one which is first-order-accurate in time and second-order-accurate in space. A generic form for the finite difference approximation to equation (3) is

$$(u^{n+1}_{i,j} - u^{n+1}_{i-1,j})/\delta x_i + (v^{n+1}_{i,j} - v^{n+1}_{i,j-1})/\delta y_j + \xi(u^{n+1}_{i,j} + u^{n+1}_{i-1,j})/(2\text{XI}_i) = 0, \tag{11}$$

where XI_i is the x-location of the centre of the ith cell and the superscript $n+1$ indicates the advanced time level $t + \delta t$.

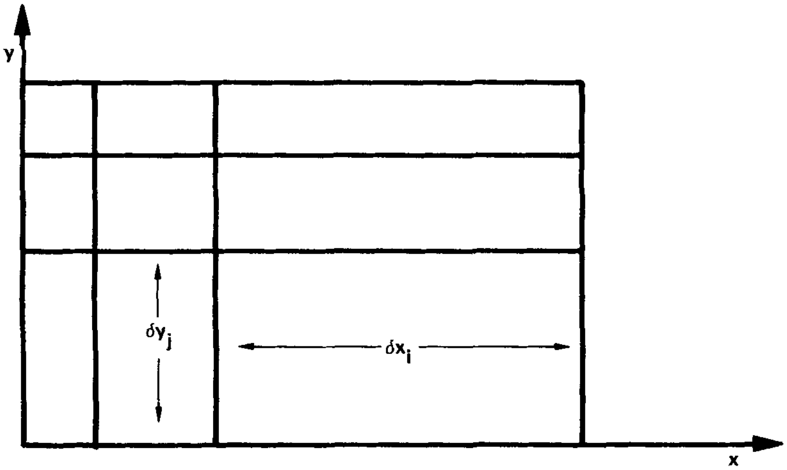


Figure 1. Schematic of finite difference cells

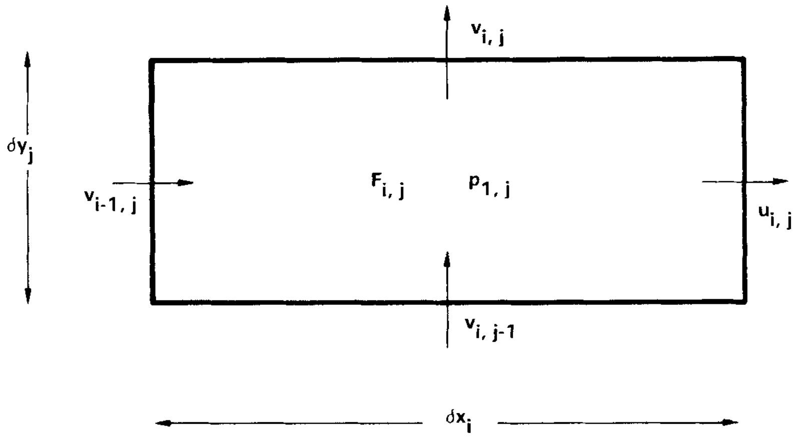


Figure 2. Location of variables in a typical cell

Similarly a generic form for the finite difference approximation of equations (8) and (9) is

$$u_{i,j}^{n+1}=u_{i,j}^{n}+\delta t[(p_{i,j}^{n+1}-p_{i+1,j}^{n+1})/(\rho\delta x)_i-\mathrm{FUX}_{i,j}^{n}-\mathrm{FUY}_{i,j}^{n}+\mathrm{VISX}_{i,j}^{n+1}+g_x], \tag{12}$$

$$v_{i,j}^{n+1}=v_{i,j}^{n}+\delta t[(p_{i,j}^{n+1}-p_{i;j+1}^{n+1})/(\rho\delta y)_j-\mathrm{FVX}_{i,j}^{n}-\mathrm{FVY}_{i,j}^{n}+\mathrm{VISY}_{i,j}^{n+1}+g_y], \tag{13}$$

where

$$(\rho\delta x)_i=\rho(\delta x_i+\delta x_{i+1})/2,$$

$$(\rho\delta y)_j=\rho(\delta y_j+\delta y_{j+1})/2.$$

Here the superscript n indicates that the terms are evaluated at time t, while the superscript $n+1$ means that the terms are evaluated at time $t+\delta t$. The advective and viscous acceleration terms have an obvious meaning, e.g. FUX means the advective flux of u in the x-direction and VISX is the x-component of viscous acceleration.

The difference approximation used for advective fluxes is a weighted upstream differencing approximation. The general form of this approximation for one of the fluxes such as $\mathrm{FUX}=u\partial u/\partial x$ is[2]

$$\mathrm{FUX}=(u_{i,j}/\delta x_\alpha)\,[\delta x_{i+1}\mathrm{DUDXL}+\delta x_i\mathrm{DUDXR}+\alpha\,\mathrm{sgn}(u)(\delta x_{i+1}\mathrm{DUDXL}-\delta x_i\mathrm{DUDXR})], \tag{14}$$

where

$$\mathrm{DUDXR}=\mathrm{RUXR}\,(u_{i+1,j}-u_{i,j}),\qquad \mathrm{DUDXL}=\mathrm{RUXC}\,(u_{i,j}-u_{i-1,j}),$$

$$\mathrm{RUXR}=1/\delta x_{i+1},\qquad \mathrm{RUXC}=1/\delta x_i,$$

$$\delta x_\alpha=\delta x_{i+1}+\delta x_i+\alpha\,\mathrm{sgn}(u)\,(\delta x_{i+1}-\delta x_i)$$

and $\mathrm{sgn}(u)$ means the sign of $u_{i,j}$. When $\alpha=0$, this approximation reduces to a second-order-accurate, central differencing scheme. When $\alpha=1$, first-order upstream differencing is recovered. The basic idea behind equation (14) is to weight the upstream quantity being fluxed more than the downstream value. The weighting factors are $1+\alpha$ and $1-\alpha$ for the upstream and downstream directions respectively. The derivatives are also weighted by cell size in such a way that the correct order of approximation is maintained in a variable mesh.

The finite difference approximations employed for viscous acceleration terms are the standard central differences which are second-order-accurate. Thus

$$\mathrm{VISX}_{i,j}=\nu(\mathrm{DUDXSQ}+\mathrm{DUDYSQ}+\mathrm{DVDXDY}+\xi\mathrm{DUXDX}), \tag{15}$$

$$\mathrm{VISY}_{i,j}=\nu[\mathrm{DVDXSQ}+\mathrm{DVDYSQ}+\mathrm{DUDXDY}+\xi(\mathrm{DUDYRX}+\mathrm{DVDXRX})], \tag{16}$$

where

$$\begin{aligned}
\mathrm{DUDXSQ}&=4\mathrm{RDXR}[\mathrm{RUXR}(u_{i+1,j}-u_{i,j})-\mathrm{RUXC}(u_{i,j}-u_{i-1,j})],\\
\mathrm{DUDYSQ}&=[\mathrm{RUYT}(u_{i,j+1}-u_{i,j})-\mathrm{RUYB}(u_{i,j}-u_{i,j-1})]/\delta y_j,\\
\mathrm{DVDXDY}&=[\mathrm{RVXT}(v_{i+1',j}-v_{i,j})-\mathrm{RVXB}(v_{i+1,j-1}-v_{i,j-1})]/\delta y_j,\\
\mathrm{DUXDX}&=2\mathrm{RDXR}[\delta x_i\,\mathrm{RUXR}(u_{i+1,j}/X_{i+1}-u_{i,j}/X_i)\\
&\qquad+\delta x_{i+1}\,\mathrm{RUXC}(u_{i,j}/X_i-u_{i-1,j}/X_{i-1})],\\
\mathrm{DVDXSQ}&=[\mathrm{RVXR}(v_{i+1,j}-v_{i,j})-\mathrm{RVXL}(v_{i,j}-v_{i-1,j})]/\delta x_i,\\
\mathrm{DVDYSQ}&=4\,\mathrm{RDYT}[\mathrm{RVYT}\,(v_{i,j+1}-v_{i,j})-\mathrm{RVYC}(v_{i,j}-v_{i,j-1})],\\
\mathrm{DUDXDY}&=[\mathrm{RUYR}(u_{i,j+1}-u_{i,j})-\mathrm{RUYL}(u_{i-1,j+1}-u_{i-1,j})]/\delta x_i,\\
\mathrm{DUDYRX}&=[\mathrm{RUYR}(u_{i,j+1}-u_{i,j})+\mathrm{RUYL}(u_{i-1,j+1}-u_{i-1,j})]/2\mathrm{XI}_i,\\
\mathrm{DVDXRX}&=[\mathrm{RVXR}(v_{i+1,j}-v_{i,j})+\mathrm{RVXL}(v_{i,j}-v_{i-1,j})]/2\mathrm{XI}_i,
\end{aligned}$$

and

$$\begin{array}{lll}
\mathrm{RDXR}=1/(\delta x_i+\delta x_{i+1}), & \mathrm{RUYT}=2/(\delta y_j+\delta y_{j+1}), & \mathrm{RUYB}=2/(\delta y_{j-1}+\delta y_j),\\
\mathrm{RVXT}=2/(\delta x_i+\delta x_{i+1}), & \mathrm{RVXB}=\mathrm{RVXT}, & \mathrm{RDYT}=1/(\delta y_j+\delta y_{j+1}),\\
\mathrm{RVXR}=\mathrm{RVXT}, & \mathrm{RVXL}=2/(\delta x_{i-1}+\delta x_i), & \mathrm{RVYT}=1/\delta y_{j+1},\\
\mathrm{RVYC}=1/\delta y_j, & \mathrm{RUYR}=\mathrm{RUYT}, & \mathrm{RUYL}=\mathrm{RUYT}.
\end{array}$$

Here X_i is the location of the right-hand boundary of the ith cell in the x-direction. It might seem peculiar to write the velocity derivatives such as $\partial u/\partial x$ and $\partial v/\partial y$ in terms of multipliers like RUXR and RVYT, but this allows for easy application of zero-gradient, velocity boundary conditions (see next section on boundary conditions) at the free surface by simply setting RUXR, RUXC, etc. to zero.

As noted earlier, the kinematic equation (10) governs the evolution of the VOF function F through time and space. When this equation is integrated over a computational cell, the changes in F for a cell reduce to fluxes of F across the cell faces. However, standard finite difference approximations lead to a smearing of the F-function and interfaces lose their definition. Therefore special care must be taken in computing the fluxes of F to preserve the sharp definition of interfaces. Fortunately the fact that F is a step function with the values of zero or one permits the use of a flux approximation that preserves its discontinuous nature. The method employed is to use a type of donor–acceptor flux approximation. The essential idea is to use information about F downstream as well as upstream of a flux boundary to establish a crude interface shape and then to use this shape in computing the flux. The reader is referred to Reference 2 for greater discussion of this technique.

Having described the governing partial differential equations and their respective finite-difference approximations, attention is now given to the boundary conditions. Boundary condition specifications must be supplied in order to have a complete mathematical description for the fluid flow.

BOUNDARY CONDITIONS

Basically there are two kinds of boundary conditions. The first type is at the calculational domain boundaries, while the second type is at the free surface. At the mesh boundaries a variety of conditions may be imposed by using a layer of fictitious cells surrounding the mesh. Consider, for example, the left boundary (see Figure 3). If this is a rigid free-slip wall, the normal velocity there must be zero and the tangential velocity should have no normal gradient. Thus

$$u_{1,j}=0, \quad v_{1,j}=v_{2,j}, \quad p_{1,j}=p_{2,j}, \quad F_{1,j}=F_{2,j}$$

for all j. If the boundary is a no-slip rigid wall, then the tangential velocity component at the wall should be zero; that is,

$$u_{1,j}=0, \quad v_{1,j}=-v_{2,j}, \quad p_{1,j}=p_{2,j}, \quad F_{1,j}=F_{2,j}$$

for all j.

For continuative or outflow boundaries an optimum prescription is one that permits the fluid to flow out of the mesh with a minimum of upstream influence. The continuative boundary conditions used at the left wall are the same as the ones used in Reference 2, which are

$$u_{1,j}=u_{2,j} \qquad v_{1,j}=v_{2,j} \quad p_{1,j}=p_{2,j}, \quad F_{1,j}=F_{2,j}$$

for all j.

A constant pressure boundary condition at the left wall is set by keeping the pressure in column $i=2$ constant and otherwise treating the boundary as continuative.

A fixed velocity boundary condition at the left wall is set by keeping the velocity $u_{1,j}$ at a specified inflow value and otherwise treating the boundary as continuative.

All these boundary conditions are imposed on the velocities computed from the momentum equations at all times. Boundary conditions similar to the ones above are used at the right, top and bottom boundaries of the calculational domain. Of course the normal and tangential velocities at the top and bottom boundaries are v and u respectively.

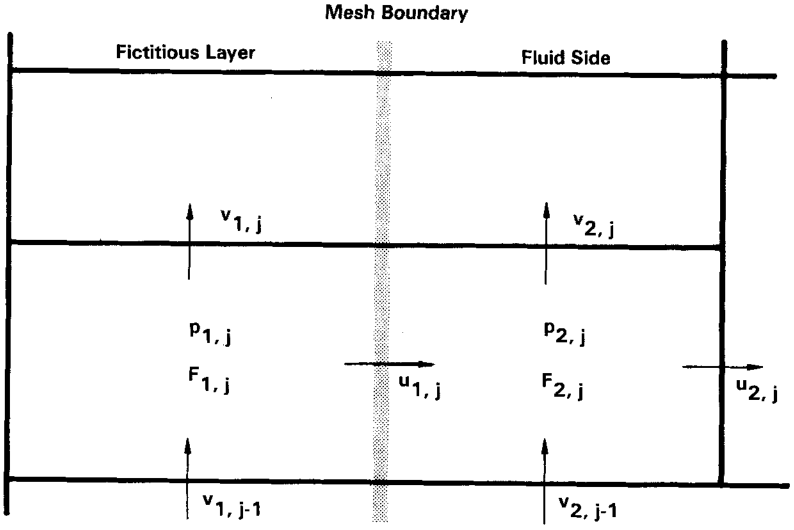


Figure 3. Field variables near a wall

The correct free surface boundary conditions in cells containing a free surface (a cell containing fluid, e.g., $F \neq 0$, but with one or more empty neighbours) are the vanishing of the tangential stress and the continuity of a sum of pressure and a viscous stress term.[10] This means that the sum of in-cell pressure and a viscous stress term should be equal to an applied surface pressure. However, use of this iteration-dependent boundary condition requires some consideration,[2] such as a special relaxation parameter. Straight application of this normal stress boundary condition could cause considerable complications, as in Reference 3.

In order to avoid these problems and simplify the procedure for setting free surface pressure boundary conditions, the viscous stress terms at a free surface are separately set to zero. As a result an iteration-independent technique[4] can be used. In this technique the pressure is assigned a specified value, say p_s, at the surface. This is done by choosing the pressure $p_{i,j}$ at the centre of a surface cell assuming a hydrostatic distribution exists within the surface cell. The hydrostatic distribution depends on the net acceleration in the direction normal to the surface (this direction is defined by a flagging scheme). This pressure is not changed during the iterative solution but is treated as a constant boundary value. Finally the surface pressure p_s is set equal to the sum of the pressures of the neighbouring empty regions.

The tangential stress conditions are satisfied by setting all velocity derivatives that involve velocity components outside the surface to zero. However, in order to correctly account for fluid advection, the velocities must be set on every cell boundary between a surface cell and an empty cell. If the surface cell has only one neighbouring empty cell (for a cell indexed as (i, j), the neighbouring cells are the ones indexed by $(i+1, j)$, $(i-1, j)$, $(i, j+1)$ and $(i, j-1)$), the boundary velocity is set to ensure that the continuity equation is satisfied within the surface cell. When there are two or more empty neighbours, the velocities on faces open to the empty cells are set to have a zero normal velocity derivative (i.e. $\partial u/\partial x$ and $\partial v/\partial y$). The velocity on the side opposite the neighbouring cell is then reset to ensure that the continuity equation is satisfied. For a wider discussion of boundary conditions at a free surface the reader is advised to consult Reference 2.

THE CIAT ALGORITHM

The calculational procedure will first be developed for flows without a free surface. This will allow for a simple, yet concise presentation of the theoretical concept behind CIAT. The method will then be generalised to allow for calculation of the flows with free surfaces.

Equations (11)–(13) form the basis of the proposed method of solution. To solve these equations, one assumes that the field variables are available at the beginning of a time cycle, either as a result of a previous time cycle calculation or specified initial conditions. The objective of the solution for each time cycle is to obtain a set of pressures and velocities that satisfy these equations.

Owing to the implicit form of the viscous and pressure terms in equations (12) and (13), we have three sets of coupled simultaneous equations to solve for each time step. Most investigators employ an iterative solution algorithm known as SIMPLE.[5] Solution by this method is determined by first calculating an intermediate velocity field based on an estimated pressure field and then obtaining appropriate correction so as to satisfy the continuity equation. This method has been used by many workers in a variety of complex single- as well as multi-phase fluid flow and heat transfer computations.

A major deficiency of the SIMPLE algorithm, as noted by Patankar[6] and experienced by the present author in application to free surface problems, is that the method is prone to divergence unless some relaxation is used. Optimum relaxation factor values are usually problem-dependent and are best obtained by numerical experimentation for the problem at hand. However, this experimentation can be quite frustrating and costly and there is no guarantee of success.

As noted earlier, the method of Pracht[3] is somewhat difficult to implement for the general treatment of boundary conditions including ones at the free surface. Thus it was decided to develop an inherently stable methodology which also lends itself to easy application of boundary conditions. This method, called CIAT, focuses attention on locally satisfying the conservation equations for a given finite difference cell such as the one depicted in Figure 2.

It is apparent that for each such cell there are five variables that have to be determined. These variables are $p_{i,j}$, $u_{i,j}$, $v_{i,j}$, $u_{i-1,j}$ and $v_{i,j-1}$. However, they must be obtained in such a way that the cell continuity equation (equation (11)) as well as the appropriate momentum equations (equation (12) for $u_{i,j}$, equation (13) for $v_{i,j}$ and similar ones for $u_{i-1,j}$ and $v_{i,j-1}$) are satisfied.

Suppose that at a given iteration level in the new time cycle we have a set of intermediate velocities and pressures denoted by u^*, v^* and p^* which will in general not satisfy the conservation equations. These starred quantities will locally produce a net mass and momentum source for the cell (i, j). These sources are defined by

$$\mathrm{SU}_{i,j}=(u_{i,j})^n-u^*_{i,j})/\delta t+(p^*_{i,j}-p^*_{i+1,j})/(\rho\,\delta x)_i-\mathrm{FUX}^n_{i,j}-\mathrm{FUY}^n_{i,j}+\mathrm{VISX}^*_{i,j}+g_x, \tag{17}$$

$$\mathrm{SU}_{i-1,j}=(u^n_{i-1,j}-u^*_{i-1,j})/\delta t+(p^*_{i-1,j}-p^*_{i,j})/(\rho\,\delta x)_{i-1}-\mathrm{FUX}^n_{i-1,j}-\mathrm{FUY}^n_{i-1,j}+\mathrm{VISX}^*_{i-1,j}+g_x, \tag{18}$$

$$\mathrm{SV}_{i,j}=(v^n_{i,j}-v^*_{i,j})/\delta t+(p^*_{i,j}-p^*_{i,j+1})/(\rho\,\delta y)_j-\mathrm{FVX}^n_{i,j}-\mathrm{FVY}^n_{i,j}+\mathrm{VISY}^*_{i,j}+g_y, \tag{19}$$

$$\mathrm{SV}_{i,j-1}=(v^n_{i,j-1}-v^*_{i,j-1})/\delta t+(p^*_{i,j-1}-p^*_{i,j})/(\rho\,\delta y)_{j-1}-\mathrm{FVX}^n_{i,j-1}-\mathrm{FVY}^n_{i,j-1}+\mathrm{VISY}^*_{i,j-1}+g_y, \tag{20}$$

$$\mathrm{SM}_{i,j}=(1/\delta x_i+\xi/2\mathrm{XI}_i)u^*_{i,j}-(1/\delta x_i-\xi/2\mathrm{XI}_i)u^*_{i-1,j}+(v^*_{i,j}-v^*_{i,j-1})/\delta y_j. \tag{21}$$

In the above equations SU, SV and SM are the momentum source terms for u-velocity and for v-velocity and the mass source term respectively. Also the superscript $n+1$ on starred quantities has been left out for conciseness.

Now the goal is to correct the cell pressure and cell edge velocities so as to annihilate these source terms. Let us suppose that the corrected pressure and velocities are

$$p_{i,j}=p^*_{i,j}+p'_{i,j}, \tag{22}$$

$$u_{i,j}=u^*_{i,j}+u'_{i,j}, \tag{23}$$

$$v_{i,j}=v^*_{i,j}+v'_{i,j}, \tag{24}$$

$$u_{i-1,j}=u^*_{i-1,j}+u'_{i-1,j}, \tag{25}$$

$$v_{i,j-1}=v^*_{i,j-1}+v'_{i,j-1}, \tag{26}$$

where the primed terms indicate corrections.

It should be noted that the adjacent cell pressures and velocities have their latest available values and are left unchanged at this stage. Substitution of equations (22)–(26) into equation (17)–(21) gives, after rearrangement, the following system of linear equations for pressure and velocities corrections:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix} \begin{Bmatrix} u'_{i,j} \\ u'_{i-1,j} \\ v'_{i,j} \\ v'_{i,j-1} \\ p'_{i,j} \end{Bmatrix} = \begin{Bmatrix} \mathrm{SU}_{i,j} \\ \mathrm{SU}_{i-1,j} \\ \mathrm{SV}_{i,j} \\ \mathrm{SV}_{i,j-1} \\ -\mathrm{SM}_{i,j} \end{Bmatrix}, \tag{27}$$

where

$$a_{11}=1/\delta t+\nu\{2\,\mathrm{RDXR}[\mathrm{RUXR}(2+\xi\delta x_i/X_i)$$
$$+\mathrm{RUXC}(2-\xi\delta x_{i+1}/X_i)]+(\mathrm{RUYT}+\mathrm{RUYB})/\delta y_j\},$$

$$a_{12}=-\nu[2\,\mathrm{RDXR}\,\mathrm{RUXC}(2-\xi\delta x_{i+1}/X_{i-1})],$$

$$a_{13}=\nu\,\mathrm{RVXT}/\delta y_j,\qquad a_{14}=-\nu\,\mathrm{RVXB}/\delta y_j,\qquad a_{15}=-1/(\rho\delta x)_i,$$

$$a_{21}=-\nu[\mathrm{RDXL}\,\mathrm{RUXC}(2+\xi\delta x_{i-1}/X_i)],$$

$$a_{22}=1/\delta t+\nu\{2\,\mathrm{RDXL}[\mathrm{RUXC}(2+\xi\delta x_{i-1}/X_{i-1})$$
$$+\mathrm{RUXL}(2-\xi\delta x_i/X_{i-1})]+(\mathrm{RUYT}+\mathrm{RUYB})/\delta y_j\},$$

$$a_{23}=-\nu\,\mathrm{RVXT}/\delta y_j,\qquad a_{24}=\nu\,\mathrm{RVXB}/\delta y_j,\qquad a_{25}=1/(\rho\,\delta x)_{i-1},$$

$$a_{31}=\nu\,\mathrm{RUYR}(1/\delta x_i+\xi/2\,\mathrm{XI}_i),\qquad a_{32}=-\nu\,\mathrm{RUYL}(1/\delta x_i-\xi/2\,\mathrm{XI}_i),$$

$$a_{33}=1/\delta t+\nu[4\,\mathrm{RDYT}(\mathrm{RVYT}+\mathrm{RVYC})+\mathrm{RVXR}(1/\delta x_i+\xi/2\,\mathrm{XI}_i)+\mathrm{RVXL}(1/\delta x_i-\xi/2\,\mathrm{XI}_i)],$$

$$a_{34}=-4\nu\,\mathrm{RDYT}\,\mathrm{RVYC},\qquad a_{35}=-1/(\rho\delta y)_j,\qquad a_{41}=-\nu\,\mathrm{RUYB}(1/\delta x_i+\xi/2\,\mathrm{XI}_i),$$

$$a_{42}=\nu\,\mathrm{RUYB}(1/\delta x_i-\xi/2\,\mathrm{XI}_i),\qquad a_{43}=-4\nu\,\mathrm{RDYB}\,\mathrm{RVYC},$$

$$a_{44}=1/\delta t+\nu[4\,\mathrm{RDYB}(\mathrm{RVYC}+\mathrm{RVYB})+\mathrm{RVXR}(1/\delta x_i+\xi/2\,\mathrm{XI}_i)+\mathrm{RVXL}(1/\delta x_i-\xi/2\,\mathrm{XI}_i)],$$

$$a_{45}=1/(\rho\delta y)_{j-1},\qquad a_{51}=1/\delta x_i+\xi/2\,\mathrm{XI}_i,\qquad a_{52}=-1/\delta x_i+\xi/2\,\mathrm{XI}_i,$$

$$a_{53}=1/\delta y_j,\qquad a_{54}=-1/\delta y_j,\qquad a_{55}=0$$

and

$$\mathrm{RDXL}=1/(\delta x_{i-1}+\delta x_i),\qquad \mathrm{RDYB}=1/(\delta y_{j-1}+\delta y_j),$$
$$\mathrm{RUXL}=1/\delta x_{i-1},\qquad \mathrm{RVYB}=1/\delta y_{j-1}.$$

Examination of the determinant of the coefficient matrix in equation (27) reveals that this determinant, irrespective of the values of δt and other variables, is always non-zero. Therefore equation (27) always will possess a unique solution. In general, the solution has to be obtained via an efficient matrix solver. However, in the case of Cartesian geometry with uniform grid spacing the coefficient matrix in equation (27) will simplify and direct solution is possible.

At each time cycle the sequence of important operations that have to be performed are as follows:

1. The computational mesh is swept cell by cell starting with $i=j=2$, the first non-boundary cell in the mesh. Sweeping is first carried out on i and then on j. Each complete sweep constitutes a single iteration. However, prior to the start of the iteration, all the explicit terms that do not change during the iteration process are computed.
2. In each cell (i, j) find SU, SV, SM and the coefficient matrix which is needed in the solution of equation (27) using the most current values available.
3. Solve equation (27) for pressure and velocities corrections.
4. Find new estimates for cell pressure and the velocities located on the sides of the cell according to equations (22)–(26) where the starred quantities are the most current values available.
5. After visiting all cells, impose the necessary boundary conditions.
6. Return to Step 1 and repeat the whole procedure until a converged solution is obtained.

Convergence of a time cycle is achieved when the SU, SV and SM absolute values of all cells fall below some small number, say ε. Typically ε is of the order of 10^{-3}, although it can vary with the specific problem being solved. Finally it should be noted that to start the iteration scheme for a new time cycle one uses the pressure and velocity field obtained from the previous time cycle as the values for p^*, u^* and v^*.

The method of solution for flows with free surfaces is basically the same as that for confined flows. The only difference is that the CIAT algorithm is only performed for cells that contain fluid and have no empty neighbours. After convergence of the iteration process, the kinematic equation for $F(x, y, t)$ is solved by donor–acceptor flux approximation methods. At this point, cells are reflagged as being full, empty or containing a free surface. The final step is to calculate the surface pressure as described in the boundary condition section.

Finally it should be noted that at locations where boundary conditions are specified the sources of momentum would be set to zero. An example would be a solid boundary at the left wall of all cells in column $i=2$. Since $u_{1,j}=0$ at this location, $SU_{1,j}$ is also set to zero.

SAMPLE CALCULATIONS

Several calculational examples in Cartesian co-ordinates are performed to illustrate the capabilities of the CIAT technique. All calculations were performed on a MicroVax II computer using single-precision arithmetic.

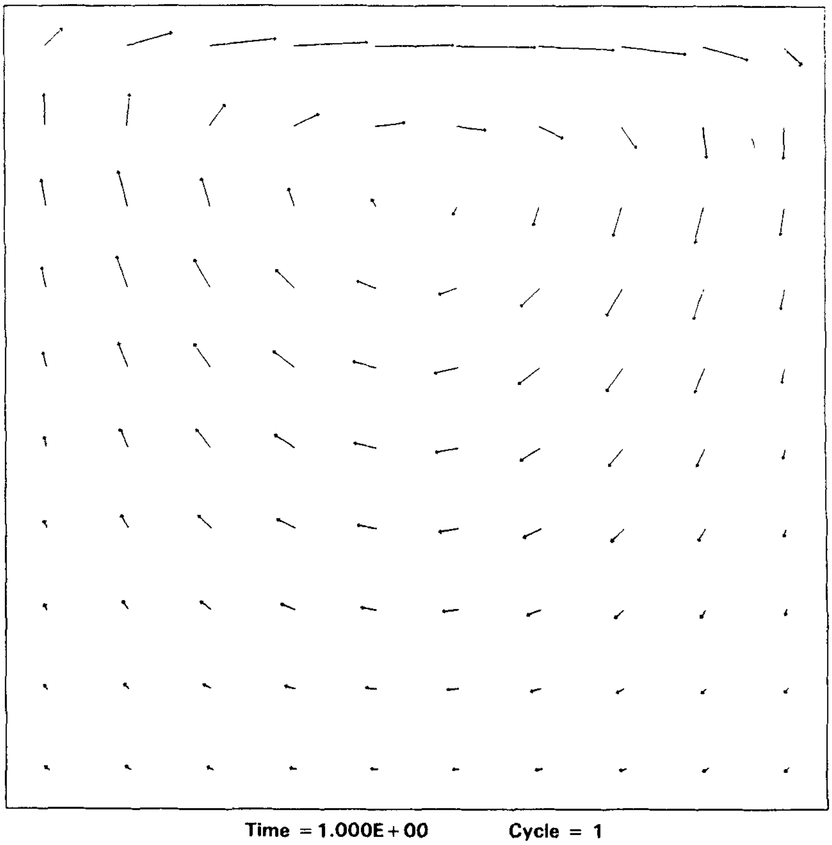


Figure 4. Velocity vector for driven cavity at $Re=0{\cdot}001$

In the first example fluid initially at rest in a square cavity of length $L=1$ is set into circulatory motion by the top boundary moving in its own plane. This problem was chosen to illustrate the advantage of implicit viscous stress modelling and the accuracy of solution using CIAT for confined flows. In addition cavity flow has been investigated experimentally and numerically so that a useful comparison can be made.

Calculations for the square cavity problem were performed for varying values of Reynolds number so that the unique features of the method could be tested. To illustrate the stability of CIAT under harsh circumstances, calculations were run with $Re=0{\cdot}001$ using a time step of $\delta t=1$ and uniform mesh spacing of $\delta x=\delta y=0{\cdot}1$. The time step chosen gave a 400000-fold violation of the explicit stability condition given by equation (1). Excluding any plotting time, this calculation required about 1·45 CPU minutes.

Figure 4 shows the steady-state velocity vector configuration for $Re=0{\cdot}001$. Horizontal velocity profiles along a vertical line through the vortex centre are shown in Figure 5, which compares the CIAT solution to the unsteady equations for $Re=0{\cdot}001$ with the result from the steady equations for $Re=0$ by Burggraf.[7]

The second example is the slow flow of a fluid having free surfaces. The example illustrated in Figures 6 and 7 represents the highly viscous slumping motion that occurs if a rectangular block

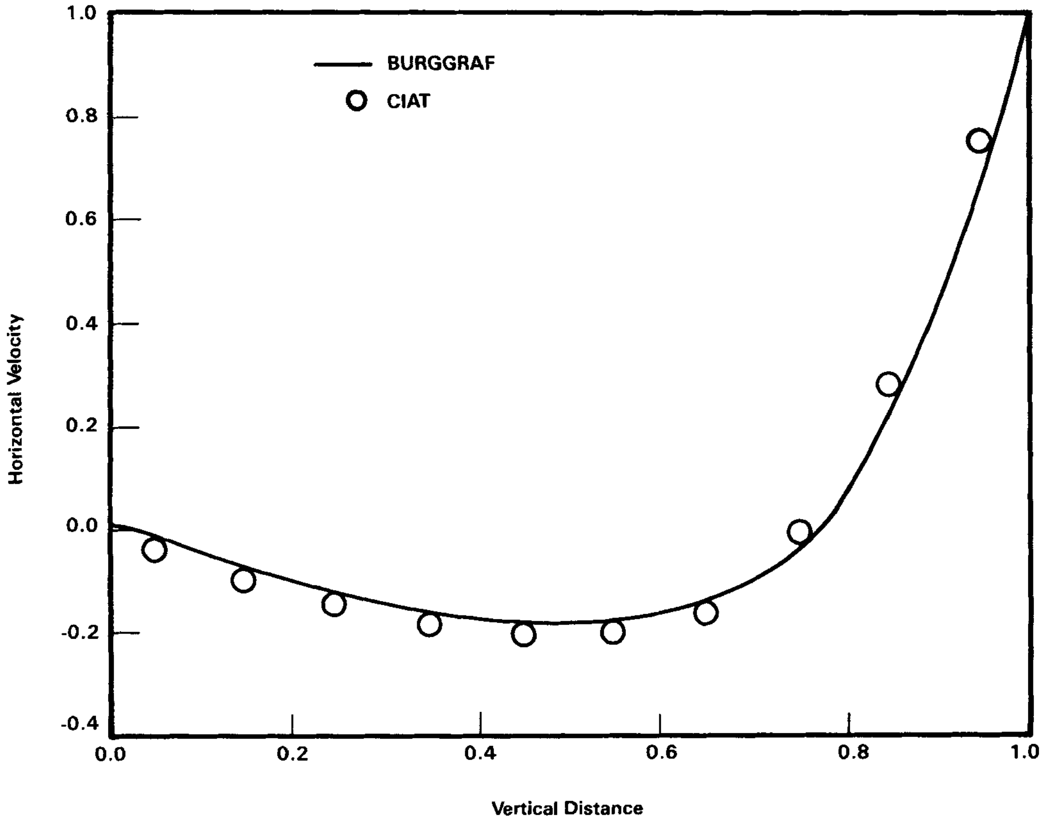
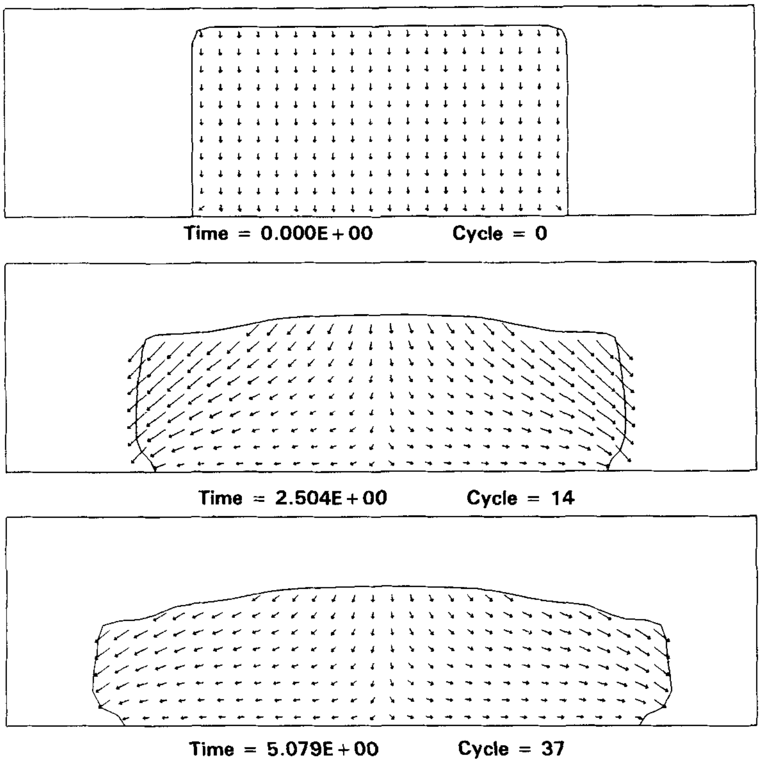


Figure 5. Comparison of horizontal velocity profiles along a vertical line through the centre of a vortex between CIAT at $Re=0{\cdot}001$ and Burggraf[7] at $Re=0$

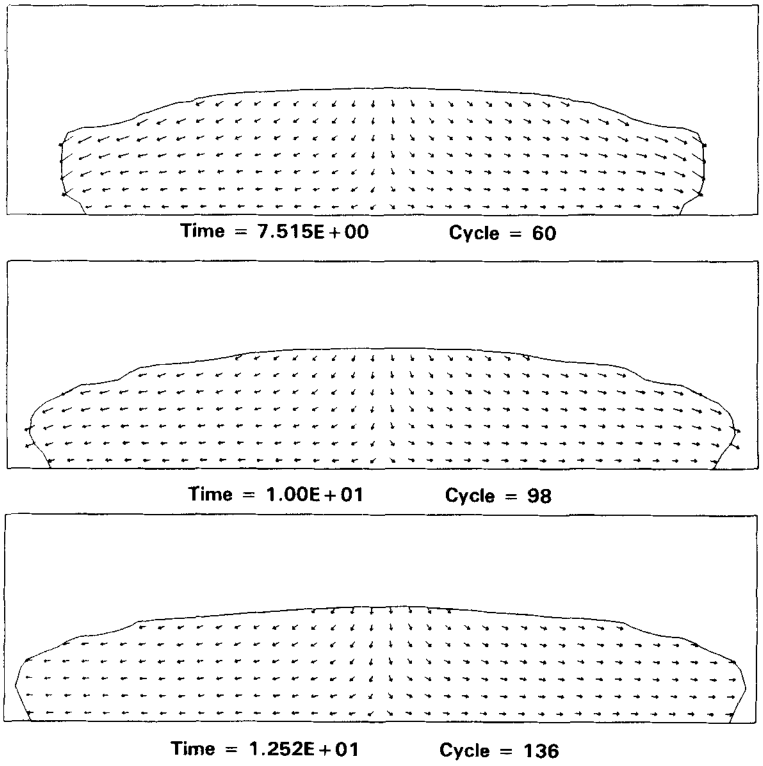


Time = 0.000E+00 Cycle = 0

Time = 2.504E+00 Cycle = 14

Time = 5.079E+00 Cycle = 37

Figure 6. Velocity vectors and fluid shapes for slow motion of a block of tar, at times 0, 2·5, and 5 s



Time = 7.515E+00 Cycle = 60

Time = 1.00E+01 Cycle = 98

Time = 1.252E+01 Cycle = 136

Figure 7. Velocity vectors and fluid shapes for slow motion of a block of tar, at times 7·5, 10, 12·5 s

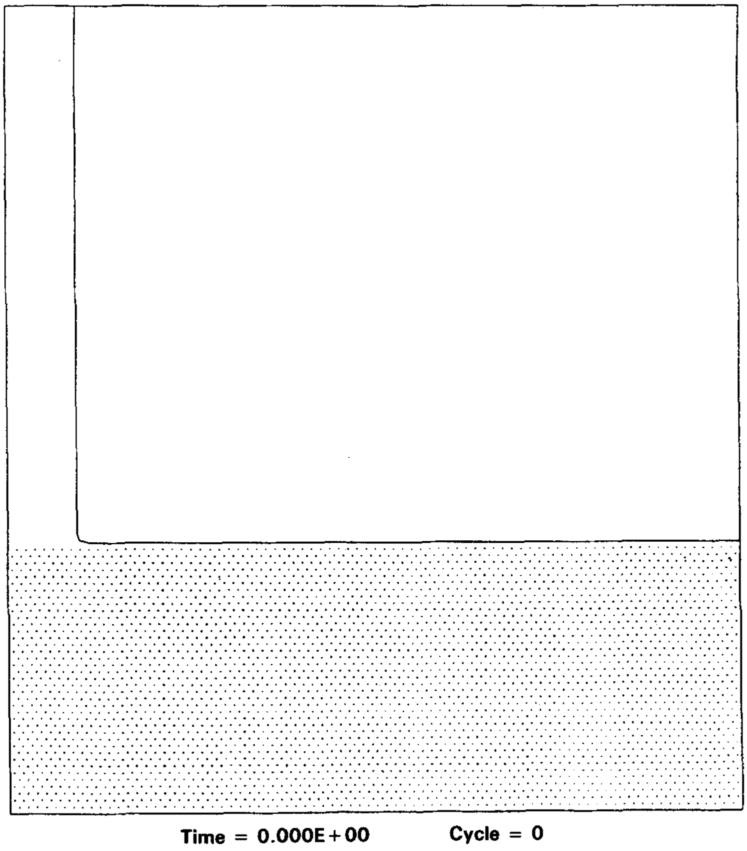


Figure 8. Initial fluid configuration for flow of resin into a partially filled rectangular box

of tar is placed on a non-slip surface and left undisturbed. The tar has a viscosity of 300 kpoise and a density of 2000 kg m^{-3}. There were 200 cells with uniform mesh spacing of $\delta x = 0{\cdot}2$ and $\delta y = 0{\cdot}183$. The calculation was started with a time step of $\delta t = 0{\cdot}1$ s and allowed to vary depending on local cell velocities. This insured compliance with the time step limit imposed by explicit differencing of convective fluxes. During the course of this calculation δt varied between 0·05 and 0·25. The results compare well with those of Pracht.[3] These calculations required about 86·4 CPU minutes to complete.

The final example is a simulation of resin flow into a rectangular box.[8] The box cross-sectional area is 4×2 inches and is initially filled with clear resin having a viscosity of 1·0 kpoise (see Figure 8). Dyed resin of the same viscosity is poured at a rate of 0·065 lb m s^{-1} through a 0·38 inch slit down the left wall of the box. The Reynolds number based on the flow rate and the width of the slit is about $8{\cdot}93 \times 10^{-6}$.

As shown in Figure 8, 3200 massless particles are used to mark the initial fluid distribution. Subsequent motion of these particles at each time cycle reflects the evolution of the initial fluid configuration. Figure 9 shows the initial fluid configuration as well as the location of free surfaces after 20 s of continuous pouring of dyed resin into the box.

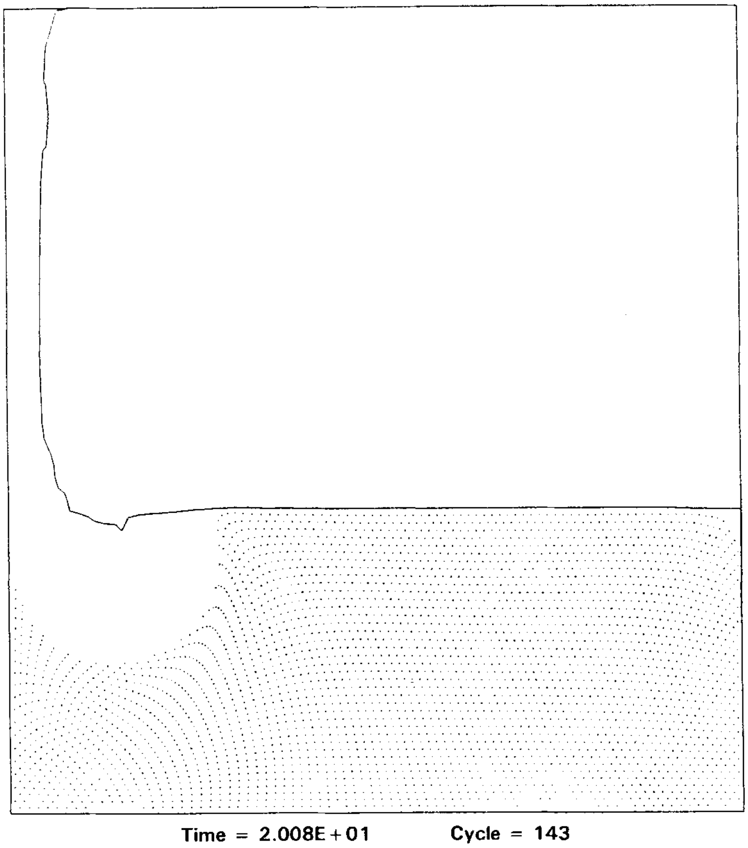


Figure 9. Fluid configuration at time 20 s for flow of resin into a partially filled rectangular box

Figure 10 illustrates the experimental fluid configuration at a time of 20 s. Comparison of Figures 9 and 10 shows excellent qualitative agreement between the CIAT prediction and experiment. For these runs a non-uniform grid was used. Smaller cells were packed near the left boundary, while larger cells were placed near the right-hand boundary. A total of 1300 cells was used and the calculation required about 522 CPU minutes to complete. Included in this time was the CPU time necessary to move and perform the book-keeping of the massless particles. A thin empty layer is seen in Figure 9, near the right wall, which has to be interpreted carefully. This is primarily a result of coarse grid resolution in this region. The procedure that is used to interpolate velocity components necessary to move the particles gives zero tangential components near a wall which is incorrect in regions of coarse resolution. However, this problem does not affect the prediction of the free surface, which is done independently using the VOF method.

In this paper we have explained a new technique which is very suitable for calculation of flow at low Reynolds numbers as well as at intermediate Reynolds numbers. The method has been demonstrated to be accurate and stable. It is easily adaptable to solution in three-dimensional space. Currently an effort is underway to incorporate CIAT into a multigrid solution algorithm where it will be used as an interior scheme for the solution of large-grid-size portions of calculation.

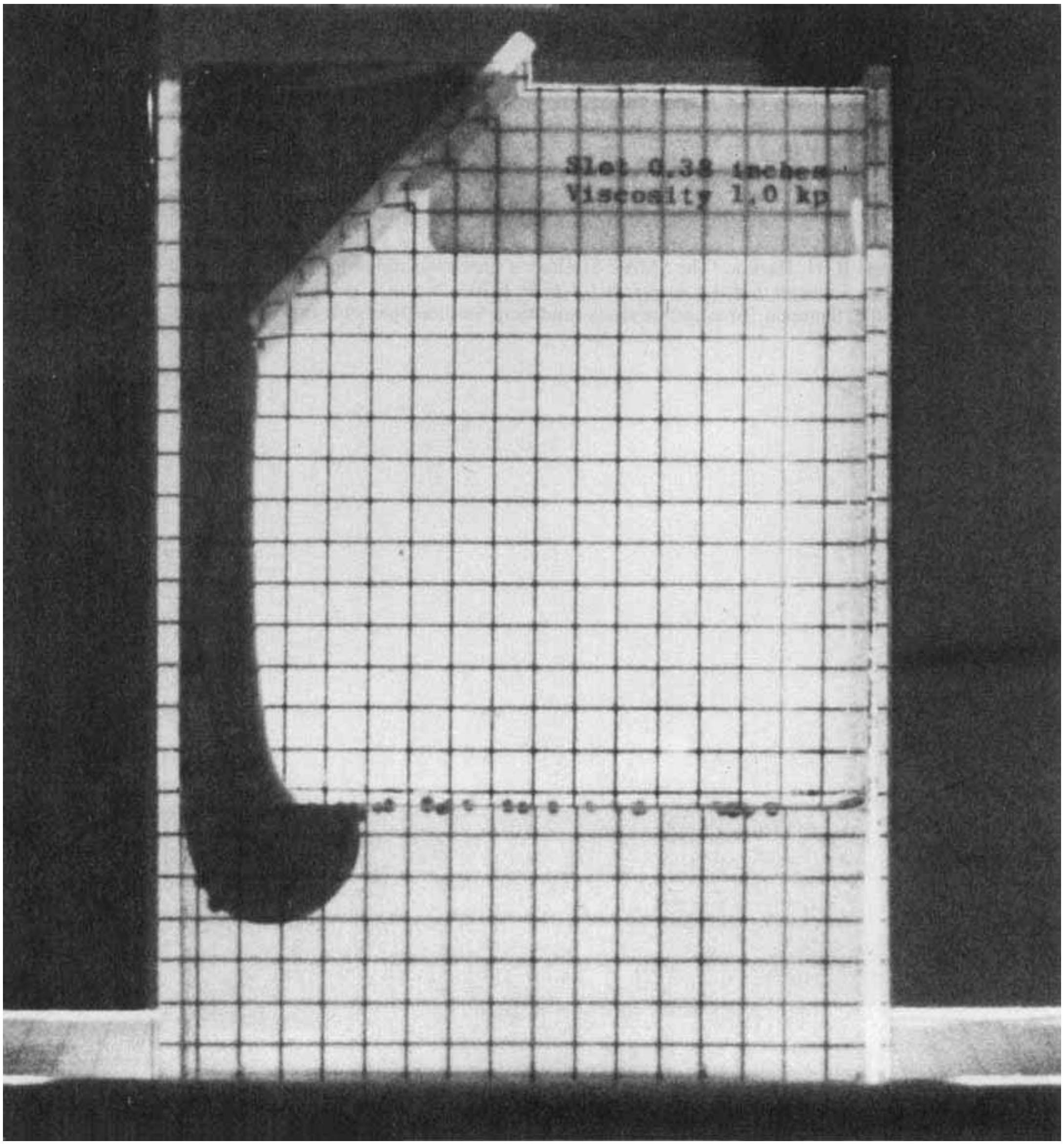


Figure 10. Experimental fluid configuration at time 20 s for flow of resin into a partially filled rectangular box

ACKNOWLEDGEMENT

The author wishes to thank Dr. Tony Hirt of Flow Science Inc. for numerous helpful discussions.

REFERENCES

1. J. E. Welch, F. H. Harlow, J. P. Shannon and B. J. Daly, 'The MAC Method: a computing technique for solving viscous, incompressible, transient fluid-flow problems involving free surfaces', *Los Alamos Scientific Laboratory report LA-3425*, 1966.
2. B. D. Nichols, C. W. Hirt and R. S. Hotchkiss, 'SOLA–VOF a solution algorithm for transient fluid flow with multiple free boundaries', *Los Alamos Scientific Laboratory report LA-8355*, 1980.

3. W. Pracht, 'A numerical method for calculating transient creep flows', *J. Comp. Phys.*, **7**, 46–60 (1971).
4. J. M. Sicilian and C. W. Hirt, 'HYDR-3D: a solution algorithm for transient 3D flows', Flow Science, Inc. report FSI-84-00-1 (1984).
5. S. V. Patankar and D. B. Spalding, 'A calculational procedure for heat, mass and momentum transfer in three-dimensional parabolic flows', *Int. J. Heat Mass Transfer*, **15**, 1787–1805 (1972).
6. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, McGraw-Hill Book Company, New York, 1980.
7. O. R. Burggraf, 'Analytical and numerical studies of the structure of steady separated flows', *J. Fluid. Mech.*, **24**, 113–152 (1966).
8. M. Salita, 'Analysis of the flowfield during casting of uncured solid propellants: final report', Morton Thiokol, Inc. memo 2814-82-M047, 1982.
9. A. A. Amsden and R. H. Harlow, 'The SMAC Method: a numerical technique for calculating incompressible fluid flows', *Los Alamos Scientific Laboratory report LA-4370*, 1970.
10. C. W. Hirt and J. P. Shannon, 'Free-surface stress conditions for incompressible-flow calculations', *J. Comp. Phys.*, **2**, 403–411 (1968).