JOURNAL OF COMPUTATIONAL PHYSICS 62, 66–82 (1986)

The Computation of Compressible and Incompressible Recirculating Flows by a Non-iterative Implicit Scheme

R. I. ISSA

Department of Mineral Resources Engineering, Imperial College of Science and Technology, London SW7 2BP, England

AND

A. D. GOSMAN AND A. P. WATKINS

Department of Mechanical Engineering, Imperial College of Science and Technology, London SW7 2AZ, England

Received October 28, 1983; revised November 7, 1984

The PISO algorithm, which is presented in a companion paper, is a non-iterative method for solving the implicity discretised, time-dependent, fluid flow equations. The algorithm is here applied in conjunction with a finite-volume technique employing a backward temporal difference scheme to the computation of compressible and incompressible flow cases. The results of calculations are compared with similar ones obtained with an existing iterative method. It is shown that for time-evolving flows the splitting error of PISO is negligibly small at the level of time-step required to eliminate the temporal truncation error, while the avoidance of iteration results in a substantial reduction in computing effort over that required by iterative methods. It is also demonstrated that PISO is stable for fairly large time steps, which renders it useful for steady-state calculations as well.

INTRODUCTION

In a companion paper [1], a method (PISO) is presented for the solution of the implicitly discretised fluid flow equations by splitting of operations. The method dispenses with outer[1] iteration and is equally applicable to both compressible and incompressible flows. In the aforementioned paper, the splitting technique is presented and is then assessed for accuracy and stability in relation to a linearised form of

[1] The term "outer iteration" is used to distinguish the iteration process on the coupled sets of equations for different variables from the iteration process which may be used to solve the simultaneous nodal finite-difference equations for a single variable.

0021-9991/86 $3.00

the equations. It was found that for such a system, the temporal error introduced by the splitting scheme (herein called splitting error) is of higher order in δt than the errors embodied in most of the temporal difference schemes currently used in time-discretising the original differential equations (herein called temporal discretisation error). This not only would enable the control of the splitting error by the time-step size δt (rather than by recourse to iteration), but should also lead, at least in the case of low-order temporal difference schemes, to the attainment of time-accurate solutions at δt values comparable to those dictated by the accuracy of the difference scheme. Also, depending on the spatial difference scheme used, the method can be shown to retain some of the stability endowed by implicit differencing.

In principle, therefore, PISO should be efficient for time-dependent calculations as iteration is disposed of without paying the penalty of having to reduce δt in order to reduce splitting errors. At the same time, because of the ability to cope with large δt, the method should also be useful for applications to steady-state problems. Whether these merits are preserved when it is applied to the full non-linear equations which govern fluid flow is the subject of the present paper.

First, it is useful to state the objectives which need to be fulfilled in order to validate the procedure; these are as follows.

(i) To demonstrate that the method is capable of handling both compressible and incompressible flows.

(ii) To show, at least for one particular combination of spatial and temporal discretisation schemes and a given spatial mesh size, that the temporal errors introduced by the splitting procedure vanish with δt and do so at least as fast as the discretisation errors due to the temporal differencing scheme (first-order-accurate in the present case). The implication here is that PISO should not hinder the achievement of a time-accurate solution at the δt values dictated by the accuracy of the difference scheme.

(iii) To verify the saving in computing effort achieved by dispensing with iteration when computing time-accurate solutions.

(iv) To demonstrate the ability to handle large time-steps which renders the method also useful for steady-state calculations.

To accomplish these objectives, comparisons are made against computations with an existing iterative method employing the same spatial and temporal difference practices. This method yields the exact solution (to within the iteration tolerance imposed) to the discretised equations over one time-step, i.e., the errors in the calculated fields are solely due to the spatial and temporal discretisation practices employed. By refinement of δt, the value at which the particular temporal difference scheme achieves a time-accurate solution can thus be determined. Since PISO is intended as an alternative to iteration, a comparison between the computing efforts required by the two will serve as a measure of relative performance.

Other non-iterative, time-marching schemes exist which are either semi-implicit,

such as those in [2, 3], or fully implicit and based on time-splitting (e.g., ADI) of the spatial fluxes, such as those of [7–9]. The first group of methods is subject to well-known restrictions on δt imposed by stability considerations and is hence at a disadvantage. The second group has to be based on the compressible form of the continuity equation, which restricts their application to that particular class of flows. Furthermore, the latter methods in general require the solution of block simultaneous sets of equations, which can be a far more complex affair than the solution of scalar ones as is the case with PISO.

The best established methods using pressure and velocity as working variables are those found in [5, 6], namely, the SIMPLE and SIMPLER schemes, both of which are iterative. Both were developed originally for steady-state incompressible flow although extensions to time-dependent, compressible flow have been made, for example, in [4]. Of the two, SIMPLER is reported to be the most efficient and robust. Other variants on these schemes exist, in particular the PUP method in [12], which shares some features with the incompressible version of PISO (see [1] for a discussion of the similarities and differences). A more recent method which also uses the same working variables, but employs block iteration, is reported in [10] to be efficient and robust for steady-state incompressible flow.

In what follows, the incorporation of the PISO methodology into a finite-volume procedure is presented. The scheme employs a staggered grid for the storage of velocity and pressure, and uses backwards differencing for the representation of the temporal variations (i.e., the Euler implicit scheme). The method is applied to both compressible and incompressible flows to demonstrate its versatility. A comparison is made against computations performed with fully iterative methods based on the SIMPLE algorithm [4–6] and employing an identical difference scheme in order to assess the relative efficiency and stability of PISO in transient flows. A study is also made of the performance of PISO for a steady-state incompressible flow calculation.

The incompressible case considered is that of a laminar flow ($Re = 100$) through a suddenly expanding circular pipe with an open outlet. The compressible case is that of laminar flow in a similar pipe but with a closed end, where a peak Re of 1000 and a peak Mach number of 0.2 are attained.

The Governing Equations and Their Discretisation

The Transport Equations

The governing equations are essentially the same as those given in [1], and are restated here in Cartesian tensor notation. The equations for continuity, momentum, and total energy in laminar compressible flow are

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_j)}{\partial x_j} = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i)+\frac{\partial}{\partial x_j}(\rho u_j u_i)=-\frac{\partial P}{\partial x_i}+\frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_j}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right] \tag{2}$$

and

$$\frac{\partial}{\partial t}(\rho e)+\frac{\partial}{\partial x_i}(\rho u_j e)=\frac{\partial}{\partial x_j}\left(\frac{\mu}{Pr}\frac{\partial T}{\partial x_j}\right)+\frac{\partial}{\partial x_j}(P u_j)$$
$$+\frac{\partial}{\partial x_j}\left[u_i\mu\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right] \tag{3}$$

where μ is the laminar viscosity and Pr is the Prandtl number. The total energy e is related to the temperature by

$$e=C_v T+\tfrac{1}{2}u_i u_i \tag{4}$$

where C_v is the constant volume specific heat. The equation of state taken here is that of a perfect gas

$$P=R\rho T \tag{5}$$

where R is the gas constant.

Evidently for constant density flow, the time derivative term in Eq. (1) vanishes and Eqs. (3) and (5) become redundant.

For the present applications, the above equations are transformed into cylindrical polar coordinates (x, r) and corresponding velocity components (u, v) with axial symmetry assumed. The transport equations thus take the form

$$\frac{\partial}{\partial t}(\rho\phi)+\frac{\partial}{\partial x}(\rho u\phi)+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v\phi)=\frac{\partial}{\partial x}\left(\Gamma\frac{\partial\phi}{\partial x}\right)+\frac{1}{r}\frac{\partial}{\partial r}\left(r\Gamma\frac{\partial\phi}{\partial r}\right)+S_\phi \tag{6}$$

where ϕ now stands for any of the dependent variables u, v, and e, Γ is the "diffusion" coefficient of property ϕ, and S_ϕ contains all the remaining terms present in the parent equations, as well as terms arising from the coordinate transformation. The continuity equation becomes

$$\frac{\partial\rho}{\partial t}+\frac{\partial}{\partial x}(\rho u)+\frac{1}{r}\frac{\partial}{\partial r}(r\rho v)=0. \tag{7}$$

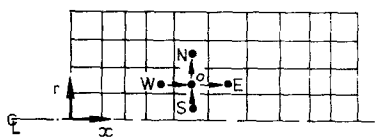


FIG. 1. The computational grid.

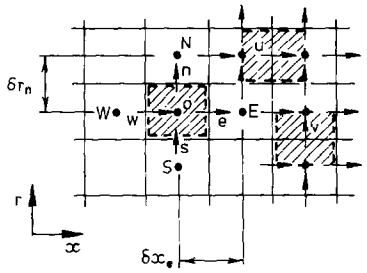


FIG. 2. Control volumes for scalars and velocities.

Discretisation

A staggered grid arrangement in which velocity nodes are located in between pressure ones, as depicted in Figs. 1 and 2, is used. The discretisation of the transport equations is effected by a finite-volume technique for which purpose control volumes (indicated in Figs. 1 and 2) surrounding each variable location are defined. The formulation of the difference equations follows standard practices employed in earlier work [4–6], hence only a brief description of it is provided below. The spatial variations are approximated by hybrid upwind/central difference formulae giving first- to second-order spatial accuracy. It should be stressed, however, that the applicability of PISO is in no way restricted to this choice, as its application in [14] to a nine-point skew upwind scheme verifies. The temporal difference scheme used is the Euler implicit formula which, although of only first-order accuracy, is chosen mainly because of the simplicity of its implementation. Here again, the validity of PISO is in principle not restricted to this choice, as inspection of the methodology in [1] will reveal.

The derivation of the difference equations is now illustrated by considering Eq. (6) as an example. Integration of this equation over a control volume such as those shown in Fig. 2 gives

$$\int_w^e \int_s^n \frac{(\rho\phi)^{n+1} - (\rho\phi)^n}{\delta t} r dr dx + \int_s^n \left[\rho u\phi - \Gamma \frac{\partial \phi}{\partial x} \right]_e r dr - \int_s^n \left[\rho u\phi - \Gamma \frac{\partial \phi}{\partial x} \right]_w r dr + \int_w^e \left[r\rho v\phi - \Gamma r \frac{\partial \phi}{\partial r} \right]_n dx - \int_w^e \left[r\rho v\phi - \Gamma r \frac{\partial \phi}{\partial r} \right]_s dx = \int_w^e \int_s^n S_\phi r dr dx \tag{8}$$

where the subscripts refer to the locations indicated in Fig. 2, and superscripts n and $n+1$ denote successive time levels. As implied by the Euler implicit temporal difference scheme used, all the spatial fluxes (which appear non-superscripted) in Eq. (8) are evaluated at time $n+1$. These spatial fluxes are now approximated by

the hybrid upwind/centred difference scheme mentioned earlier. Taking the flux F_w at the w-face of the control volume as an example, it can be represented as

$$F_w = \int_s^n \left[\rho u \phi - \Gamma \frac{\partial \phi}{\partial x} \right]_w r dr = M_w \left[\alpha_w \phi_W + (1 - \alpha_w) \phi_0 \right] \tag{9}$$

where M_w is the mass flux through the cell face w and is defined as

$$M_w = (\rho u a)_w \tag{10}$$

and a is the cell-face area. The quantity α is a weighting factor which depends on the cell Peclet number Pe as

$$\begin{aligned} \alpha_w &= \frac{1}{2}\left(1 + \frac{2}{Pe}\right) && \text{for} \quad |Pe| \leqslant 2 \\ &= 1 && \text{for} \quad Pe > 2 \\ &= 0 && \text{for} \quad Pe < -2 \end{aligned} \tag{11}$$

and Pe is given by

$$Pe = \left(\frac{\rho u \delta x}{\Gamma} \right)_w . \tag{12}$$

Expression (9) is now rewritten as

$$F_w = A_w(\phi_W - \phi_0) + M_w \phi_0 \tag{13}$$

where the coefficient A_w defined by

$$A_w = (M\alpha)_w \tag{14}$$

has been introduced.

Similar relations to (13) above can be formulated for the fluxes through the n and s faces (where v should replace u in expressions (10) and (12)) as well as the e face of the cell. Substitution of these expressions into the integral equation (8) yields the difference equation

$$(B - A_0)\, \phi_0^{n+1} = H'(\phi^{n+1}) + \bar{S}_\phi + B^n \phi_0^n . \tag{15}$$

In Eq. (15), the quantity B is defined by

$$B = \frac{\rho r \delta r \delta x}{\delta t} \tag{16}$$

and $\bar{S}_\phi$ is the volume integral of the source term S_ϕ. The operator H' relates to the ϕ values prevailing at the surrounding nodes and is given by

$$H'(\phi) = A_e\phi_E + A_w\phi_W + A_n\phi_N + A_s\phi_S \tag{17}$$

while the central coefficient A_0 is defined by

$$A_0 = -(A_e + A_w + A_n + A_s + \text{div } M) \tag{18}$$

where div $M = M_e - M_w + M_n - M_s$.

Equation (15), it should be observed, corresponds to the momentum and energy equations (7) and (9) in Ref. [1] when ϕ stands for u, v, or e. The continuity equation is similarly discretised by integrating Eq. (7) over a control volume surrounding a pressure node, to get

$$\beta(\rho^{n+1} - \rho^n) + (\rho ua)_e - (\rho ua)_w + (\rho va)_n - (\rho va)_s = 0 \tag{19}$$

where β is given by

$$\beta = \frac{r\delta r\delta x}{\delta t} \tag{20}$$

and the mass fluxes are evaluated at the $n+1$ time level. The pressure equation can now be derived from a combination of the discretised continuity relation (19) and the momentum equations which are given by Eq. (15) when ϕ stands for u and v; this procedure is outlined below.

The velocities u_e, v_n, etc., in Eq. (19) are to be replaced by expressions obtained from the momentum equations. Taking the velocity u_e as an example, Eq. (15) for this quantity can be cast into the form

$$u_e = [H'_e - (P_0 - P_E)\, a_e + C_e]/(B - A_0) \tag{21}$$

where the pressure gradient term is now written out explicitly, and the quantity C contains all other terms in the parent equation. The mass flux at the e-face of the cell can now be evaluated from Eq. (21) as

$$(\rho ua)_e = \tilde{H}_e - D_e(P_0 - P_E) + \tilde{C}_e \tag{22}$$

where the following have been introduced:

$$\tilde{H}_e = \left(\frac{\rho aH'}{B - A_0}\right)_e \tag{23}$$

$$D_e = \left(\frac{\rho a^2}{B - A_0}\right)_e \tag{24}$$

$$\tilde{C}_e = \left(\frac{C\rho a}{B - A_0}\right)_e. \tag{25}$$

Similar expressions for the mass fluxes at faces w, n, and s of the cell can be derived, all of which may be substituted into Eq. (19) to yield

$$D_0 P_0 = J(P) - \text{div}\, \tilde{H} - \text{div}\, \tilde{C} - \beta(\rho^{n+1} - \rho^n) \tag{26}$$

where the operator J is defined by

$$J = D_e P_E + D_w P_W + D_n P_N + D_s P_S \tag{27}$$

and

$$D_0 = D_e + D_w + D_n + D_s. \tag{28}$$

Equation (26) corresponds to the pressure equation (12) in [1], and its form is similar to the general ϕ equation (15).

Method of Solution

The system of Eqs. (15) for u, v, and e (the latter of which becomes redundant in the case of incompressible flow) together with Eq. (26) for the pressure are solved by the PISO algorithm, a full description of which appears in [1]. The non-linearity in Eq. (15) arising from the dependency of the coefficients A and B on the field variables themselves is handled by evaluating these coefficients from the old time level values. Although this practice is only first-order accurate in time it is of the same order of accuracy as the temporal difference scheme, and is therefore consistent with it. Furthermore, for the same reason, all the computations for the compressible flow case presented herein were made with the two-stage version of the PISO algorithm presented in [1]. Indeed, computations with the three-stage variant that were performed did not bring any improvement in accuracy while requiring substantially higher computing effort.

The set of nodal simultaneous algebraic equations for each variable are solved by a line successive over-relaxation procedure to a specified convergence level. Recent work [15] shows that considerable saving can be made if Stone's strongly implicit procedure [13] is used instead, especially for the pressure equation. Further saving can be made in the case of steady-state calculations, when temporal accuracy is of no consequence, by relaxing the convergence tolerance on these sets of equations. However, the pressure equations must still be converged to at least 10% of the initial residuals, otherwise the method is forced to take more time-steps to arrive to the final solution, to the detriment of efficiency.

For the purpose of evaluating the performance of PISO, a comparison is made against calculations carried out with an iterative method. The method chosen is that based on the well-established SIMPLE algorithm in [5, 6], which is comprehensively documented in the literature. The method was initially developed for steady-state problems, but has been extended to compressible transient flows (as

in [4]). For the steady-state case, the temporal derivative terms in the transport equations are suppressed and iteration replaces time-marching with under-relaxation used to procure stability and convergence. This under-relaxation is related to the time step δt of an equivalent transient calculation as shown in [12] and in the Appendix. For unsteady-state problems, iteration is employed at each time step to satisfy all the equations simultaneously. Under-relaxation must be used here also in order to ensure the convergence of the iteration process. As is the case with the PISO computations, a line successive over-relaxation scheme was employed for each set of algebraic simultaneous equations.

RESULTS

Incompressible Flow

The incompressible flow case chosen is that of an axisymmetric laminar flow in a duct with a sudden enlargement at entry (Fig. 3). The ratio of the diameter at inlet to that of the duct is 1:2 and the length of the duct is 4 times its diameter. The time-varying velocity profile at inlet is assumed to be spatially uniform across inlet and the flow is started from rest with an inlet velocity rising linearly from zero at $t=0$ to its final steady value of V at $t=L/V$, where L is the length of the duct. The Reynolds number (based on the peak velocity V and the duct diameter) is 100. At the outlet, zero velocity gradients are assumed, while at the walls, the usual no-slip conditions are imposed. The mesh size is 20×20 (uniform in both directions) throughout and is kept unaltered. The tests carried out are designed to serve three purposes. The first is to demonstrate that the accuracy of the PISO method is at least as good as the accuracy of the temporal difference scheme used. The second is to illustrate the stability of the scheme for large δt, which makes it equally suited to the calculation of steady-state problems. And third, to show that the first objective is achieved at substantially lower computing effort than with existing methods utilising sequential iteration, as the one compared here.

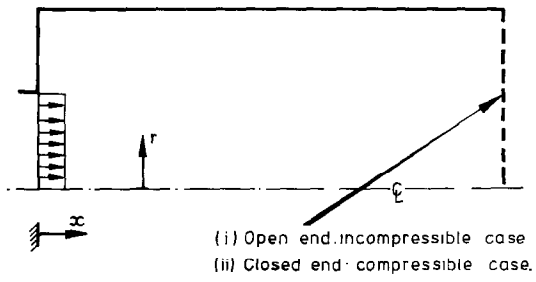


FIG. 3. Geometry of duct with sudden enlargement.

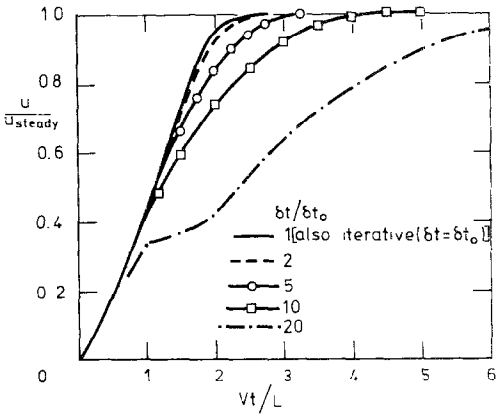


FIG. 4. Predicted velocity transience on centreline (incompressible case).

Calculations were first performed with the time-marching version of SIMPLE using various values of the time-step size δt to find the maximum allowable value (say, δt_0) for which the temporal truncation error is acceptably small. Under-relaxation was necessary to stabilise the computations, values of 0.5 being used for the velocities in the momentum equations (but not for the pressure). A similar exercise was then performed with PISO using step sizes that are multiples of δt_0. Figures 4 and 5 show the predicted transient behaviour of the axial velocity (normalised by the absolute steady-state value) at two locations: the first (Fig. 4) is on the centre line halfway between inlet and outlet and the second (Fig. 5) is located at the point where the maximum reverse axial velocity occurs (i.e., in the recirculation zone). The different curves in each case are those obtained with PISO for different

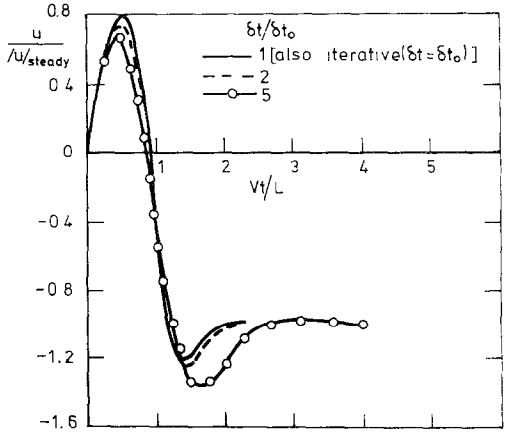


FIG. 5. Predicted velocity in recirculation zone (incompressible case).

values of $\delta t/\delta t_0$, as well as that for the time-accurate solution obtained with the iterative method. The latter curve is indistinguishable from that pertaining to $\delta t/\delta t_0 = 1$, verifying that the PISO solution is at least as accurate as the temporal difference scheme used, and that this algorithm allows the use of the maximum valued of δt permitted by the accuracy of that difference scheme.

The benefits of avoidance of iteration were verified by the finding that the ratio of the computing time required by the two methods (with $\delta t/\delta t_0 = 1$) was 9.3 in favour of PISO.

Figure 4 also shows that PISO remains stable for $\delta t/\delta t_0$ as large as 20. This is very useful when only the steady-state solution is sought (in which case temporal accuracy is of no consequence). Taking large time-steps permits rapid arrival at the steady state with minimum computing effort, as exemplified in Fig. 6. This figure shows the computing times (on a CDC 6400 machine) required respectively by PISO and the steady-state version of SIMPLE (in which time derivatives are omitted) to reach that state. In the iterative calculations the under-relaxation factor λ (imposed on the velocities only) was varied to determine the optimum value, i.e., that yielding the minimum computing time. A similar exercise was then performed with PISO, using now δt as the controlling parameter; thus the abscissa in Fig. 6 is either λ or δt (made dimensionless by dividing by the convection time scale $\delta x/V$) according to the method.

Two facts emerge from an examination of this plot. The first is that PISO arrives at the solution at a much lower level of computing effort than the other method, whatever the value of λ or δt. The second is that the PISO curve is remarkably flat over a wide range of values, thus indicating stability and robustness. In contrast,

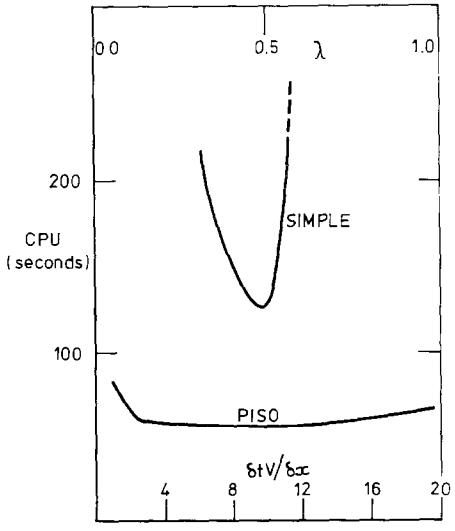


FIG. 6. Computing time for the calculation of steady-state incompressible flow.

the performance of SIMPLE is fairly sensitive to the relaxation parameter and the scheme diverges beyond a value of 0.55 (which is equivalent to $\delta t V/\delta x$ of 1.2 for a time-marching method).

It should, however, be pointed out that the SIMPLE scheme used for these comparisons is not the most efficient of the existing iterative methods. For example, several versions of the same scheme have been proposed and tested in [12], some of which show marked improvement in efficiency over the original one. Other techniques, notably SIMPLER and PUP in [6] and [12], respectively, are also reported to be much faster as well as more stable.

A qualitative assessment of the results gleaned from [12] shows that the improvement in speed over SIMPLE by the best of these methods for steady-state calculations is about the same as that achieved by PISO for steady-state problems (including ones not presented here). This is not surprising as the latter shares some common features with the others as explained in [1]. However, the improvement in speed attained by these other methods over SIMPLE still does not match the margin of saving achieved by PISO in time-evolving flows, thanks to its non-iterative strategy.

Compressible Flow

In this flow case, the geometry of the duct is similar to that of the previous case, with the exception that the downstream end is now closed, so that the fluid mass inside the duct varies with time. The velocity profile at inlet, which is again assumed to be uniform in the radial direction, is taken to vary sinusoidally with time with a peak velocity V corresponding to a Reynolds number of 1000 and a Mach number of 0.2. The period of the cycle is taken as twice the time taken for a fluid particle traveling at the peak velocity to traverse the length of the duct; the computations are carried out over a full cycle of velocity variations starting from rest. The density and temperature at inlet are assumed to be uniform and constant with time and the duct wall temperatures are taken to be constant at the same value as that at inlet. A uniform mesh of 20×20 is used throughout.

As in the incompressible case, calculations were first performed with the time-marching, compressible version of SIMPLE using various values of δt to find the value δt_0 for which temporal truncation errors are negligible. Here again under-relaxation was necessary to ensure convergence and values of $\lambda = 0.5$ were used for the velocities and temperature (but not pressure). The exercise was then repeated with PISO using step sizes that are multiples of δt_0. The results of those computations are displayed in Figs. 7 to 10, which show the predicted transience of the axial velocity (normalised by the peak inlet value) and of the pressure (normalised by the pressure at inlet) at two locations in the duct. The first point (Figs. 7 and 8) is located at one-third of the duct length downstream of inlet and at 80% of the duct radius, this being the location where some of the highest reverse velocities occur. The second point (Figs. 9 and 10) is located on the centreline halfway down the length of the duct. In all these figures, the time coordinate appears normalised

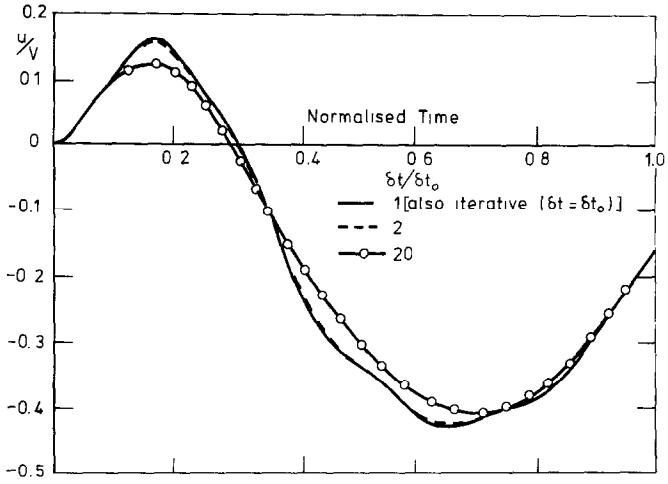


FIG. 7. Predicted velocity transience in recirculation zone (compressible case).

by the period of the cyclic inlet-velocity variations. These figures indicate that the solutions given by PISO for different values of the time-step size ($\delta t/\delta t_0$) converge to an acceptable level of temporal accuracy at the same value of δt as that required by the iterative method, i.e., δt_0. This verifies that the accuracy of PISO for compressible flow is as good as the accuracy of the temporal difference scheme employed. The most significant finding emerging from these calculations was. however, that the PISO computations required only 0.19 of the computing effort demanded by the iterative scheme.

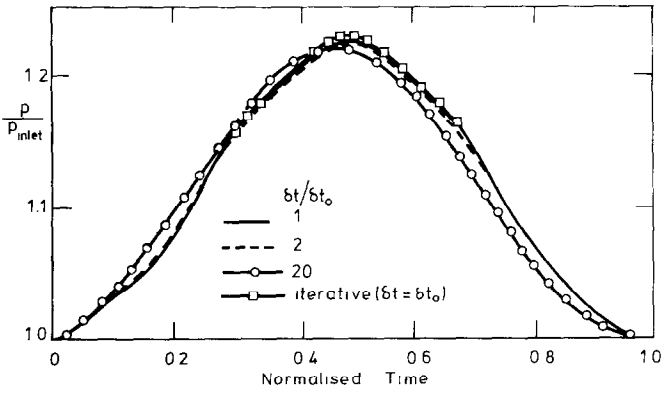


FIG. 8. Predicted pressure transience in recirculation zone (compressible case).

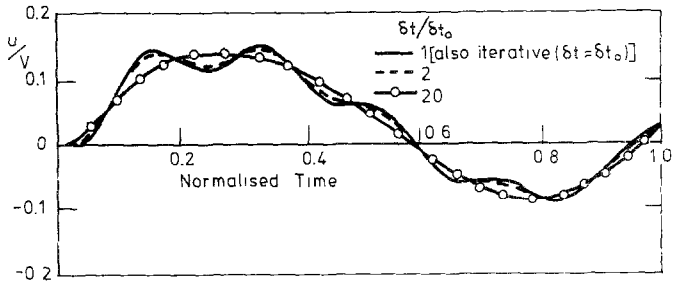


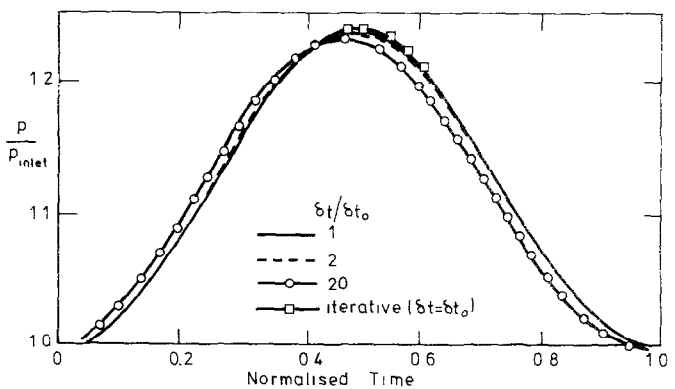
FIG. 9. Predicted velocity transience on centreline (compressible case).

The figures also demonstrate that time-step sizes of up to $20\,\delta t_0$ can be taken with PISO without any signs of instability, whereas solutions could not be obtained with the iterative method for values greater than $3\,\delta t_0$, as the iteration process then failed to converge with the chosen set of values of the under-relaxation parameter λ. Here again, the stability of PISO for large δt renders it useful for the computation of steady-state compressible flows.

CONCLUSIONS

The PISO algorithm described in [1] has been implemented in a finite-volume method which employs Euler's implicit temporal difference scheme and a hybrid upwind/centred spatial difference scheme. The resulting procedure was applied to the computation of two cases of axisymmetric laminar flow in circular ducts with abrupt enlargement. The first case was for incompressible fluid with an open duct-end and the second was for a compressible flow with a closed duct-end.

The results of the computations verify the findings of the analysis in [1]



581/62 1-6 FIG. 10. Predicted pressure transience on centreline (compressible case).

regarding the accuracy and stability of the algorithm. This is achieved through a comparison against calculations performed with an existing iterative method. The most important fact emerging from these comparisons is that PISO is many times faster than its iterative counterpart for transient flows, whether compressible or incompressible. Furthermore, it exhibits stable behaviour for large time-step sizes which makes it a reliable technique for steady-state calculations.

Recent work [15] where the PISO method was applied to turbulent flows (for which the $k-\varepsilon$ model of turbulence was used) shows that the same significant saving over iterative schemes can be achieved for such flows also. This is the case provided that the splitting procedure for the source terms in the k and ε equations outlined in [1] is implemented in conjunction with PISO.

APPENDIX: The Relation between Time-Marching and Under-Relaxed Steady-State Procedures

Consider a model implicitly discretised scalar equation (such as Eq. (15) in the text) for time-dependent flow in the absence of sources and for a constant density fluid. It can be written as

$$(B - A_0)\,\phi_0^{n+1} = H'(\phi^{n+1}) + B\phi_0^n. \tag{A.1}$$

The corresponding equation in steady-state form in which the time derivative terms are suppressed would be obtained if the term B in Eq. (A.1) is set to zero and the superscripts now stand as iteration counters. An iterative procedure based on this form of the equation invariably requires under-relaxation in order to stabilise the computations. A standard practice adopted in introducing under-relaxation is to base the new iteration level value (i.e., at $n+1$) on

$$\phi_0^{n+1} = \lambda\phi_0 + (1-\lambda)\,\phi_0^n \tag{A.2}$$

where λ is the relaxation factor and the superscript denotes the value that would be obtained if no relaxation was used. Thus the equation actually solved in an iterative, under-relaxed procedure becomes

$$-\frac{A_0}{\lambda}\,\phi_0^{n+1} = H'(\phi^{n+1}) - \frac{1-\lambda}{\lambda}\,A_0\phi_0^n. \tag{A.3}$$

Equation (A.3) is obtained from the substitution of relation (A.2) into the form of Eq. (A.1) in which the term B is set to zero. It is not difficult to discern the similarities between the time derivative terms in Eq. (A.1) and the damping terms due to under-relaxation in Eq. (A.3). Equating these terms gives

$$-\frac{1-\lambda}{\lambda}\,A_0 = B. \tag{A.4}$$

Now, B is defined in Eq. (16) in the text as

$$B = \frac{\rho r \delta r \delta x}{\delta t}. \tag{A.5}$$

Consider, for the moment, the case of convection–dominated flow (which is often the case); the coefficient A_0 is given approximately by

$$-A_0 = \rho u r \delta r + \rho v r \delta x. \tag{A.6}$$

Substitution of Eqs. (A.5) and (A.6) in Eq. (A.4) gives

$$\left(\frac{u\delta t}{\delta x} + \frac{v\delta t}{\delta r}\right) = \frac{\lambda}{1-\lambda}. \tag{A.7}$$

It is evident from (A.7) that the relaxation parameter λ is related to the local Courant numbers, $u\delta t/\delta x$, $v\delta t/\delta r$, of the equivalent time-dependent procedure (a similar result was also obtained in [12]). For example, if v is zero, the case of $\lambda = 0$ corresponds to $\delta t = 0$, while the case of $\lambda = 1$ corresponds to the case of $\delta t = \infty$. The case of $\lambda = 0.5$ corresponds to a local Courant number $u\delta t/\delta x$ of 1.

Similar findings can be reached for diffusion-dominated flows, where now the relaxation factor λ becomes related to the parameters $\Gamma\delta t/(\rho\delta x^2)$ and $\Gamma\delta t/(\rho\delta r^2)$. In all cases, therefore, there is a definite relationship between the value of the time-step size in a time-marching procedure and the under-relaxation factor employed in steady-state algorithms, a relationship which relates local flow properties with the mesh size and δt.

ACKNOWLEDGMENT

This work was carried out under the financial support of the Science and Engineering Research Council.

REFERENCES

1. R. I. ISSA. *J. Comput. Phys.* **61** (1985).
2. F. H. HARLOW AND J. E. WELCH, *Phys. Fluids* **8** (1965), 2182.
3. F. H. HARLOW AND A. A. AMSDEN, *J. Comput. Phys.* **8** (1971), 197.
4. A. D. GOSMAN AND A. P. WATKINS, *in* "Proceedings 1st Symposium on Turbulent Shear Flows, 1977.
5. L. S. CARETTO, A. D. GOSMAN, S. V. PATANKAR, AND D. B. SPALDING, *in* "Proceedings 3rd International Conference on Numerical Methods in Fluid Dynamics, 1972," p. 60.
6. S. V. PATANKAR, "Numerical Heat Transfer and Fluid Flow," McGraw–Hill, New York, 1980
7. W. R. BRILEY AND H. MCDONALD, *J. Comput. Phys.* **24** (1977), 372.
8. R. M. BEAM AND F. R. WARMING, *AIAA J.* **16** (1978), 393.
9. R. W. MACCORMACK, *AIAA* 19th Aerospace Sci Meeting, *AIAA*-81-0110, 1981.

10. J. P. VAN DOORMAAL AND G. D. RAITHBY, *in* "Proceedings 10th *IMACS* World Congress on System Simulation and Sci. Comp., 1981," p. 128.
11. B. E. LAUNDER AND D. B. SPALDING "Mathematical Models of Turbulence," Academic Press, London/New York, 1972.
12. G. D. RAITHBY AND G. E. SCHNEIDER, *Numer. Heat Transfer* **2** (1979), 417.
13. H. L. STONE, *SIAM J. Numer. Anal.* **5** (1968), 530.
14. R. W. BENODEKAR, A. J. H. GODDARD, A. D. GOSMAN, AND R. I. ISSA, *in* "Proceedings, *ASME* Fluids Engineering Congress, 1983."
15. A. D. GOSMAN, Y. Y. TSUI, AND A. P. WATKINS, *SAE* paper 840229, 1984.