INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2001; **36**: 497–518 (DOI: 10.1002/fld.140)

A parallel pressure implicit splitting of operators algorithm applied to flows at all speeds

N. W. Bressloff*,[1]

School of Engineering Sciences, Computational Engineering and Design Centre, University of Southampton, Southampton, U.K.

SUMMARY

A parallel implementation of the pressure-based implicit splitting of operators (PISO) method is described and applied to both compressible and incompressible flows. The treatment of variables at the interfaces between adjacent blocks is highlighted, and, for compressible flow, a straightforward method for the implicit handling of density is described. Steady state and oscillatory flow through a sudden expansion are considered at low speeds for both two- and three-dimensional geometries. Extension of the incompressible method to compressible flow is assessed for subsonic, transonic and supersonic flow through a two-dimensional bump. Although good accuracy is achieved in these high-speed flows, including the automatic capturing of shock waves, the method is deemed unsuitable for simulating steady state high-speed flows on fine grids due to the requirement of very small time steps. Copyright © 2001 John Wiley & Sons, Ltd.

KEY WORDS: compressible; finite volume; incompressible; oscillatory; PISO

1. INTRODUCTION

The pressure implicit splitting of operators (PISO) algorithm was devised as a non-iterative technique for the solution of the implicitly discretized time-dependent flow equations [1]. Its main strengths relative to other pressure based procedures concern its avoidance of iteration, underrelaxation and the need for any modification or user intervention when applying it to steady state simulations. It has been shown to be accurate and robust [2]. However, relative to the SIMPLE-type algorithms it is more complicated, and it becomes increasingly expensive for steady state calculations on fine grids (due to the need for small time steps) [3].

The suitability of PISO as a general purpose parallel solver for compressible and incompressible flows forms the subject of the present article. Parallelization of the PISO algorithm

* Correspondence to: School of Engineering Sciences, Computational Engineering and Design Centre, University of Southampton, Highfield, Southampton SO17 1BJ, U.K. Tel.: +44 2380 595473; fax: +44 2380 593230.
[1] E-mail: nwb@soton.ac.uk

Received February 2000
Revised May 2000

is first described for incompressible flow. It is based on the multi-block method for structured grids [4]. Momentum interpolation [5] is used (for the evaluation of control volume face fluxes) on a collocated variable arrangement to avoid pressure–velocity decoupling. This procedure demands additional information exchange at the interface between adjoining blocks, as is explained below. Convection fluxes are discretized using deferred correction [6] and van Leer's high-resolution CLAM scheme [7], and diffusion fluxes are approximated by central differences. Second-order accuracy in time is provided by a three-time level method [8]. The parallel performance of the method is assessed for two- and three-dimensional steady state flow through a sudden expansion. Calculations are performed on an SGI Origin 2000 and on a cluster of dual processor Pentium III 500 MHz PCs, both using the message passing interface (MPI) for information exchange between separate processors. The value of the parallel algorithm, both in terms of problem size (and memory usage) and calculation time is considered for oscillatory flow in a three-dimensional channel.

Although the non-iterative nature of PISO represents one of the principal advantages of the method, particularly for transient flows, it produces a sensitivity of grid dependence to the granularity of parallelization. This is due to the effect of fixing the values of variables in ghost cells at the interface between adjacent blocks when solving the flow matrices for a particular block. The transient development to steady state of the backward-facing step flow provides some evidence for this effect, and further information is provided by simulation of unsteady oscillatory flow through a similar geometry.

The compressible formulation of PISO is described in Issa [1] and validated by Issa *et al.* [9] and again by Issa and Javareshkian [10]. However, there is little additional evidence of its application to high-speed flows. Furthermore, the validation in Reference [10] did not include supersonic flow. In contrast, recently there have been a number of articles describing the treatment of compressible flows by extending the SIMPLE-type pressure correction algorithms developed for incompressible flows [11–14]. They commonly employ finite volume steady state solutions on a collocated variable arrangement, and implicitly incorporate the influence of pressure on density in the pressure correction equation. Density is thus derived from pressure using an equation of state. An overview of the development of these techniques can be found in Demirdzic *et al.* [12].

Whereas iterative pressure-based SIMPLE-type algorithms model steady compressible flow using the steady state form of the Navier–Stokes equations, traditional compressible flow solution techniques employ an unsteady form of the Navier–Stokes or Euler equations [15,16], treating density as a primary variable. Although they can be extended to low Mach number and incompressible flows using artificial compressibility [17], such approaches tend to be inefficient and are questionable when applied to unsteady incompressible flows [18] due to the weak coupling between density and pressure.

While the non-iterative nature of PISO makes it particularly suitable for unsteady flows at all speeds, and for low-speed steady state flows, its application to steady state flow on fine grids at higher speeds is less attractive due to the exhaustive computational time demanded by the requirement of very small time steps. Nonetheless, it is instructive to consider how the algorithm is parallelized, and to assess its accuracy for a range of compressible flow regimes. A simpler method to that described by Issa [1] to account for the effect of density variations on mass fluxes is described. The complete parallel, compressible PISO approach is verified by

applying it to subsonic, transonic and supersonic inviscid flows over a bump in a two-dimensional channel. Whereas the incompressible test cases are treated using Cartesian co-ordinates, curvilinear co-ordinates are needed for the compressible flow bump geometries. In these cases, contravariant vectors are employed in the momentum interpolation procedure and Cartesian velocity components are used as dependent variables in the momentum equations [19].

2. NUMERICAL ANALYSIS

The PISO method is a time stepping procedure that solves the momentum equations using pressure from the previous time step [1]. Although momentum is conserved in this velocity predictor step, mass conservation has to be satisfied as well. This is achieved by substituting expressions for fluxes, derived from the predicted velocity field, into the mass continuity equation. The resulting equation is a pressure correction equation, and the solution to it yields the predicted pressure field and corrections for the velocities and fluxes. A second pressure correction is derived and solved to yield a corrected pressure field and a second velocity (and flux) correction. Error analysis performed in Reference [1] shows that further corrector stages are not necessary as they increase the accuracy beyond that produced by the methods of spatial and temporal discretization.

The conservation form of the Navier–Stokes equations (written here using tensor notation) are

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

and

$$\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_j u_i)}{\partial x_j} = \frac{\partial \tau_j}{\partial x_j} - \frac{\partial p}{\partial x_i} + S_i \tag{2}$$

where u_j signify velocity components in the x_j-directions, and ρ, p, τ and t denote density, pressure, shear stress and time respectively. When the finite volume method is applied to Equations (1) and (2), algebraic equations of the form

$$a_c \phi_c = \sum a_v \phi_v + S \tag{3}$$

are derived, where ϕ denotes one of the velocity variables or, in the case of the continuity equation, the pressure correction; S represents the source term. The coefficient a depends on the methods of discretizing the original partial differential equation (PDE) [20]. The subscript 'c' signifies the centre cell, while the summation includes all other cells, subscript v, in its computational molecule. Standard compass notation is used here such that lower case subscripts (w, e, s, n, b, t) denote face values on the corresponding west, east, south, north, bottom or top faces, while upper case subscripts (W, E, S, N, B, T) define cell-centred values

of the adjacent control volumes. The first step of the PISO method is to solve the momentum equations to obtain a predicted velocity field denoted below by single asterisk superscripts.

In order to derive the pressure correction equation, the algebraic momentum equations are re-written such that the pressure difference is removed from the source term, leading to

$$a_c\phi_c = \sum a_v\phi_v + S' - \Delta pA \tag{4}$$

where S' denotes all source terms excluding the pressure term. Since Equation (4) is written for cell-centred values, face values can be defined by

$$\phi_{\text{face}} = \left.\frac{\sum a_v\phi_v + S'}{a_c}\right|_{\text{face}} - \left.\frac{A\Delta p}{a_c}\right|_{\text{face}} \tag{5}$$

where the 'face' subscript denotes interpolation between adjacent cell-centred values. The face flux is then given by the product of this value and the face area, A. Before solving the pressure correction equation, the predicted velocities ϕ^* and fluxes F^* are communicated between adjacent processors.

Having defined face fluxes as above, it is now possible to discuss the application of the momentum interpolation approach in the PISO algorithm. When employing Equation (4) in a time stepping procedure, some method must be sought which compensates for the use of a pressure field and fluxes from the previous time step. The PISO algorithm accomplishes this by introducing two corrections to the velocity field and the cell face fluxes. The first is derived from the following two forms of Equation (4):

$$a_c\phi_c^* = \sum a_v\phi_v^* + S' - \Delta p^m A \tag{6}$$

$$a_c\phi_c^{**} = \sum a_v\phi_v^* + S' - \Delta p^* A \tag{7}$$

where m signifies values at the previous time step, single-asterisk expressions denote values associated with the predicted velocity and pressure fields for the new time step, while double-asterisk expressions signify the corrected set of values. Subtracting Equation (6) from Equation (7) yields

$$\phi_c^{**} = \phi_c^* - \frac{A}{a_c}\Delta p' \tag{8}$$

and a corrected face flux

$$F_{\text{face}}^{**} = F_{\text{face}}^* - \left.\frac{A^2\Delta p'}{a_c}\right|_{\text{face}} \tag{9}$$

where the pressure correction $p' = p^* - p^m$ and the predicted flux, F^*_{face}, is given by the interpolation procedure described above. Substitution of fluxes for all cell faces into the continuity equation according to the finite volume method then yields a pressure correction equation. Solution of this provides an updated pressure field from which the velocity field and all face fluxes are corrected. At the interface between adjoining blocks, values for the momentum, pressure and pressure correction fields for each block are stored in the ghost cells of the other. Further storage is required to handle the second line of momentum values needed for the second-order deferred correction scheme. Exchange of fluxes and mass conservation are handled according to the methods described in Reference [4]. In addition to these standard interface procedures, the form of Equation (5) requires exchange of the centre coefficient, a_c, and the first expression on the right-hand side of Equation (5).

The second velocity correction is obtained by writing a third version of Equation (4) as

$$a_c \phi_c^{***} = \sum a_v \phi_v^{**} + S' - \Delta p^{**} A \tag{10}$$

in which triple-asterisk values represent those associated with this second correction, and p^{**} denotes the corrected pressure field. Subtraction of Equation (7) from Equation (10) yields

$$\phi_c^{**} = \phi_c^{**} + \frac{1}{a_c}\left(\sum a_v \phi_v^{**} - \sum a_v \phi_v^{*}\right) - \frac{A}{a_c}\Delta p'' \tag{11}$$

and face fluxes

$$F^{***}_{\text{face}} = F^{**}_{\text{face}} + A\left(\frac{\sum a_v \phi_v^{**} - \sum a_v \phi_v^{*}}{a_c}\right)\Bigg|_{\text{face}} - \frac{A^2 \Delta p''}{a_c}\Bigg|_{\text{face}} \tag{12}$$

where the second pressure correction $p'' = p^{**} - p^*$. Again, substitution of face fluxes into the finite volume continuity equation yield pressure corrections and the treble-asterisk set of velocity corrections, all of which become the current time step values to be used in the subsequent time step. The extra terms in Equations (11) and (12) (relative to Equations (8) and (9)) necessitate exchange of these expressions at the interface between blocks, in addition to the other information exchange described above.

2.1. *Compressible flow*

When adapting the pressure correction method to solve compressible flows, there are effectively three additional factors to include: (i) the density variation due to compressibility; (ii) the solution of an energy equation to yield temperature; and (iii) compressible flow boundary conditions.

2.1.1. Density variation. In pressure-based procedures for incompressible flow, the density is absorbed into the coefficients of the finite volume algebraic equations. However, density variation cannot be ignored for compressible flows, and so, in the continuity equation,

allowance must be made for the effect of pressure on density. Also, the unsteady term in both the momentum and continuity equations is treated implicitly. For a three-time step method, the predicted and corrected velocities are

$$\left(\frac{3\delta V}{2\delta t}\rho^m + a_c\right)\phi_c^* = \sum a_v\phi_v^* + S' - \Delta p^m A + \frac{\delta V}{2\delta t}(4\rho^{m-1}\phi_c^{m-1} - \rho^{m-2}\phi_c^{m-2}) \tag{13}$$

and

$$\left(\frac{3\delta V}{2\delta t}\rho^m + a_c\right)\phi_c^{**} = \sum a_v\phi_v^* + S' - \Delta p^* A + \frac{\delta V}{2\delta t}(4\rho^{m-1}\phi_c^{m-1} - \rho^{m-2}\phi_c^{m-2}) \tag{14}$$

In Equations (13) and (14) δV and δt denote the volume of the cell and the time step, and superscripts $m-1$ and $m-2$ represent values at the previous two time steps. While it would be desirable to include the predicted density, ρ^*, in the coefficient on the left-hand side of Equation (14), this introduces an additional unknown at this stage and so the density from the previous time step, ρ^m, is used. In this form, subtraction of Equation (13) from Equation (14) again leads to the correction equations (8) and (9). However, when substituting for velocity components and density into the continuity equation, ϕ^{**} is used from Equation (8) and $\rho^m + \rho'$ is used in place of ρ^*. The resulting four terms are

$$\rho^m A\phi_c^* - \frac{\rho^m A^2}{a_c^+}\Delta p' + \rho' A\phi_c^* - \frac{\rho' A^2}{a_c^+}\Delta p' \tag{15}$$

where $a_c^+ = a_c + (3\delta V/2\delta t)\rho^m$. The first two terms are the same as those for the incompressible equations. The last term is the product of two corrections and is ignored. The third term contributes to the mass imbalance due to the density correction, and it is included implicitly in the pressure correction equation by replacing ρ' by the pressure correction using an equation of state. For a perfect gas, at temperature T

$$p = \rho RT \tag{16}$$

and, density as well as density corrections are evaluated from Equation (16)

$$\rho = \frac{p}{RT} \quad \text{and} \quad \rho' = \frac{p'}{RT} \tag{17}$$

R denotes the universal gas constant. When this density correction is substituted in to the third term of Equation (15) and combined with other pressure correction terms, it is important to ensure that coefficients remain positive. An UPWIND scheme is used such that the coefficient of p' obtained from

$$\rho'\phi A = \frac{A\phi}{RT}p' \tag{18}$$

represents a positive addition to either the centre coefficient or the adjacent cell coefficient. Consider the p' terms in ρAu on an east face

$$-\rho^m \frac{A^2}{a_c^+}(p'_E - p'_c) + \frac{F_e}{RT}p' = bp'_c - bp'_E + \frac{F_e}{RT}p'_c, \quad \text{if } F_e > 0$$
$$= bp'_c - bp'_E + \frac{F_e}{RT}p'_E, \quad \text{if } F_e < 0 \tag{19}$$

where

$$b = \frac{\rho^m A^2}{a_c^+} \tag{20}$$

Thus the east coefficient and centre coefficients are respectively

$$b + \frac{\max(-F_e, 0)}{RT} \tag{21}$$

$$b + \frac{\max(F_e, 0)}{RT} \tag{22}$$

In the first velocity correction, the temperature field from the previous time step is used. Before applying the second velocity correction, the energy equation is solved to yield a predicted temperature. The details of this part of the algorithm are considered in the next section. The second velocity correction equation is given by

$$\left(\frac{3\delta V}{2\delta t}\rho^m + a_c\right)\phi_c^{***} = \sum a_v \phi_v^{**} + S' - \Delta p^{**} A + \frac{\delta V}{2\delta t}(4\rho^{m-1}\phi_c^{m-1} - \rho^{m-2}\phi_c^{m-2}) \tag{23}$$

Subtracting Equation (14) yields the second velocity corrections and these are now substituted into the continuity equation with the density represented by the sum of the predicted density ρ^* and the second correction ρ''. This now produces the following terms for each of the flux-density products in the finite volume continuity equation:

$$\rho^* A\phi_c^{**} - \frac{\rho^* A^2}{a_c^+}\Delta p'' + \rho'' A\phi_c^{**} - \frac{\rho'' A^2}{a_c^+}\Delta p'' + (\rho^* + \rho'')A\left(\frac{\sum a_v \phi_v^{**} - \sum a_v \phi_v^*}{a_c^+}\right) \tag{24}$$

The terms

$$\frac{\rho'' A^2}{a_c^+}\Delta p'' \quad \text{and} \quad \rho'' A\left(\frac{\sum a_v \phi_v^{**} - \sum a_v \phi_v^*}{a_c^+}\right) \tag{25}$$

are negligible and can be ignored. The first three terms are treated in the same way as those in Equation (15), and the remaining term is manipulated according to the procedure described by Equations (17)–(22). The energy (and temperature) is corrected at this stage and then a further velocity correction is applied

$$\left(\frac{3\delta V}{2\delta t}\rho^m + a_c\right)\phi_c^{****} = \sum a_v \phi_v^{***} + S' - \Delta p^{***} A + \frac{\delta V}{2\delta t}(4\rho^{m-1}\phi_c^{m-1} - \rho^{m-2}\phi_c^{m-2}) \tag{26}$$

When Equation (23) is subtracted from Equation (26) and the resulting expression for ϕ_c^{****} substituted with $(\rho^{**} + \rho''')$ into the continuity equation, the following terms are produced by each velocity component:

$$\rho^{**} A \phi_c^{***} - \frac{\rho^{**} A^2}{a_c^+}\Delta p''' + \rho''' A \phi_c^{***} - \frac{\rho''' A^2}{a_c^+}\Delta p''' + (\rho^{**} + \rho''')A\left(\frac{\sum a_v \phi_v^{***} - \sum a_v \phi_v^{**}}{a_c^+}\right) \tag{27}$$

which are treated in the same way as those in Equation (24). The procedure described above treats density variation in a way that is analogous to the SIMPLE-type methods, and thus avoids the tricky manipulation of a number of density ratios required by the original PISO algorithm [1]. In addition, this avoids the need to exchange density ratios at the interfaces between blocks, and only density needs to be exchanged (along with energy) in addition to the variables exchanged for incompressible flow.

2.1.2. Energy equation. For isentropic flow at a total temperature T_0 and Mach number M, the temperature can be derived from the isentropic relation

$$\frac{T_0}{T} = 1 + \frac{\gamma - 1}{2} M^2 \tag{28}$$

where γ denotes the ratio of specific heat capacities, C_p/C_v. Rearranging Equation (28) and substituting for M in terms of the speed $|V|$

$$T = T_0 - \frac{|V|^2}{2C_p} \tag{29}$$

Thus, it is not necessary to solve the full energy equation. If Equation (29) is substituted in Equation (17), the density is given by

$$\rho = \frac{p}{RT_0 - \dfrac{|V|^2}{2C_p}} \tag{30}$$

and so, for isentropic flow, Equation (30) is used to evaluate the density before the second and third velocity corrections.

If the full energy equation is employed, an equation of the form of Equation (3) is solved after the first pressure correction equation, and then an explicit correction is applied to the energy after solution of the second pressure correction equation. The details of this part of the algorithm can be found in Issa [1].

2.1.3. Boundary conditions. For an inviscid subsonic inflow boundary, total pressure and total enthalpy are defined and static pressure is extrapolated to the ghost cell. Thus, the Mach number is calculated from the isentropic expression for total pressure, p_0

$$M = \frac{2}{\gamma - 1}\left(\left(\frac{p_0}{p}\right)^{(\gamma-1)/\gamma} - 1\right)^{0.5} \tag{31}$$

For the test cases considered below, the velocity, U_{in}, is perpendicular to the boundary, so, from the definition of Mach number

$$U_{\text{in}} = M(\gamma RT)^{0.5} \tag{32}$$

The density at the boundary is evaluated from the equation of state. It is not possible to fix the Mach number at the subsonic inflow boundary, but it is effectively fixed when used in conjunction with a constant static pressure (outflow) boundary for which the static pressure is set at a value based on the total pressure and the desired Mach number using the isentropic relation for pressure

$$\frac{p_0}{p} = \left(1 + \frac{\gamma - 1}{2} M^2\right)^{\gamma/(\gamma-1)} \tag{33}$$

All other variables at a constant static pressure boundary are extrapolated from within the domain to the ghost cell.

At a supersonic outflow boundary all variables, including pressure and pressure correction, are extrapolated from the interior of the domain to the ghost cell. However, since it is not feasible to utilize the ghost cell pressure correction when solving the pressure correction equation, the term involving the ghost cell must be decomposed into terms based on the method of extrapolation. Consider an east face supersonic outflow boundary. The pressure correction difference term is

$$-\frac{\rho_e A_e^2}{a_c}(p''_E - p''_c) \tag{34}$$

But, if extrapolating p''_E as $2p''_c - 2p''_W$, Equation (34) becomes

$$-\frac{\rho_e A_e^2}{a_c}(p''_c - p''_W) \tag{35}$$

Thus, $\rho_e A_e^2/a_c$ is subtracted from the centre and west coefficients.

Supersonic inflow is fully specified by setting the stagnation pressure and temperature and the Mach number.

3. RESULTS

3.1. Steady state incompressible flow

The classical backward-facing step problem is used here to validate the present method and to assess its parallel efficiency. This flow is characterized by the Reynolds number, defined as $Re = 2Uh/v$, where U, h and v denote the mean inlet velocity at the step, the step height and the kinematic viscosity respectively. This definition is identical to that given in Armaly *et al.* [21]. A parabolic inlet flow at a Reynolds number of 800 through a 2:1 expansion is commonly used for validation [22]. At this Reynolds number, the steady state (laminar) solution produces a primary region of recirculation immediately downstream of the step, and a secondary vortex on the opposite wall, as shown in Figure 1. Also shown in this figure are the domain decomposition boundaries for the parallel calculation using two, four or eight processors. The domain is similarly subdivided when using six processors.

The streamlines in Figure 1 were obtained on a grid comprising 400×80 control volumes—Grid C. Two coarser grids were used in determining the grid dependence of the solution—Grid B (200×80) and Grid A (104×80). Grid refinement beyond 80 control volumes in the cross-stream direction did not effect the results presented here. Figure 2(a) shows the transient movement of the primary vortex reattachment point for grids A, B and C using four processors. Time is non-dimensionalized on the mean inlet velocity and the downstream channel height. A non-dimensional time step, $\Delta T = 0.002$ is used in all simulations, and the growth of the recirculation regions appears to have finished after a time $T = 600$. At $T = 0$, all velocities, fluxes and pressure are set to zero.

Although there is a noticeable difference in Figure 2(a) between the curve for Grid A and those for Grids B and C, the differences between the finest two grids is negligible. Equally, very small differences occurred for the transient response of the reattachment point on the finest grid for different numbers of processors, whereas on Grid A significant differences were observed. These results demonstrate an important sensitivity of the solution to the coarseness of parallel domain decomposition. Further evidence for this phenomenon is provided in the next subsection.

The non-dimensional reattachment length for the primary vortex, R_p/h, and the separation and reattachment lengths for the secondary vortex, S_s/h and R_s/h, calculated here for Grid C show excellent agreement with other recent results (cf. Table I).

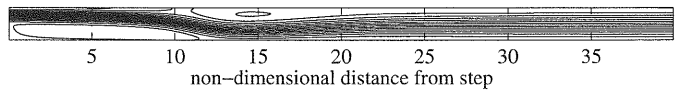

Figure 1. Streamlines of two-dimensional backward-facing step flow at $Re = 800$.

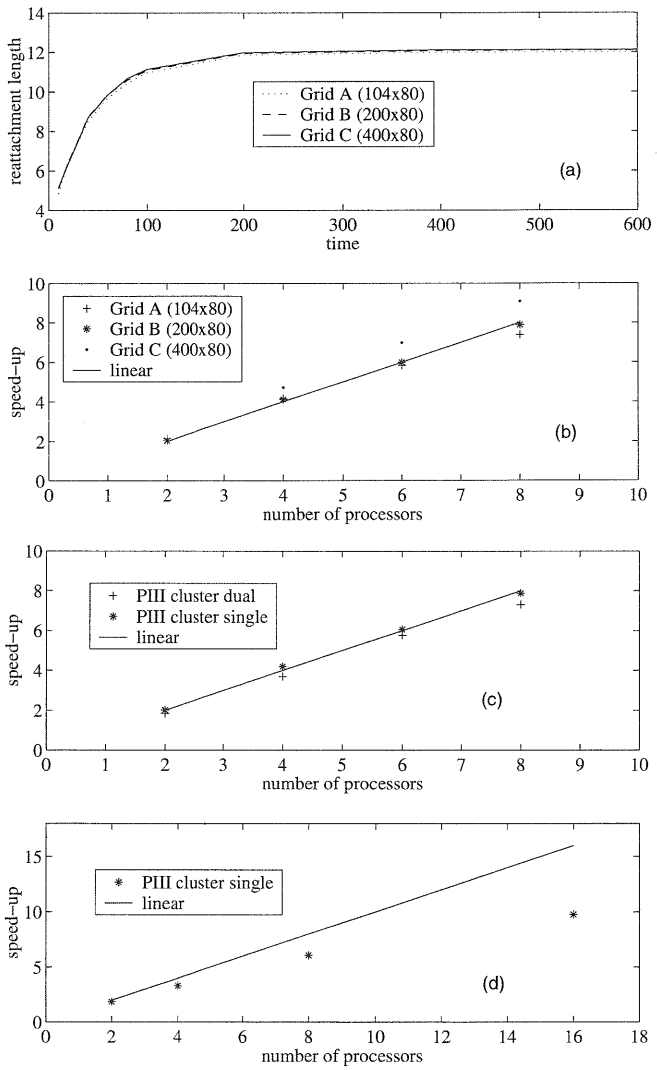

Figure 2. Flow through a sudden expansion at $Re = 800$. (a) Transient development of primary vortex in two-dimensional flow; (b) parallel efficiency on different grids in two-dimensional flow; (c) speed-up on cluster in two-dimensional flow; (d) speed-up on cluster in three-dimensional flow.

The performance of the parallel PISO algorithm on an Origin 2000 is shown in Figure 2(b) for two, four, six and eight processors and for the three grids described above. Speed-up is measured relative to a single processor calculation without using MPI. The super linear speed-up obtained in nearly all cases and the greater efficiency of the larger number of

Table I. Numerical predictions of the separation and reattachment lengths for two-dimensional flow at $Re = 800$.

	R_p/h	S_s/h	R_s/h
Present results	12.12	9.64	20.94
Barton [22]	12.19	9.75	20.89
Gartling [27]	12.20	9.70	20.95
Sani and Gresho [28]	12.44	10.18	20.50

processors on the finer grid most likely result from the improved performance of the cache in handling smaller datasets produced by parallelization.

A similar analysis is depicted in Figure 2(c) for a cluster of eight dual processor Pentium III PCs. However, a further comparison is presented between running both or only one processor per node. It is not surprising that the parallel performance of the Origin 2000 (a dedicated parallel machine) is superior to that achievable with the cluster, particularly since the nodes in the latter are only connected via a 100 Mbit/s switching hub. Likewise, when operating both processors on each node rather than using a single processor on twice the number of nodes, inferior speed-up is to be expected since the processors have to share the same memory bus and ethernet connection.

While the foregoing analysis provides some insight into the parallel performance of the PISO algorithm on different architectures, a more rigorous test of efficiency is provided by large problems that use close to the full memory capacity of the machine. Using up to 16 processors on a PIII cluster, flow through a three-dimensional sudden expansion is considered here. The geometry comprises a spanwise extrusion (of 18 step heights) of the two-dimensional back-step described above with a mirror symmetry at one side of the span and a solid wall at the other. This configuration has been used in recent simulations [23] of three-dimensional effects in the flow studied experimentally by Armaly [21]. Details of the flow are described in Reference [23]. For the present parallel performance analysis, Grid B from above (200×80 cells) is used with 50 spanwise cells and required approximately 80 per cent of the 512 Mb memory on a single node.

Figure 2(d) depicts a relatively poor speed-up performance, particularly on 16 processors, and raises the question concerning the potential for improving parallel performance by using a high-speed interconnect in place of a 100 Mbit/s ethernet switch.

3.2. Unsteady incompressible flow

Although the development of recirculation regions in the previous example provides some transient data for comparison, oscillatory flow through a sudden expansion facilitates the validation of the algorithm in a genuinely unsteady flow. This problem has been experimentally studied by Sobey [24] and extensively analysed by Tutty and Pedley [25]. Tutty and Pedley [25] demonstrated good qualitative agreement with Sobey's [24] experimental observations, particularly with respect to the generation of a vortex wave, the general shape and motion of the eddies and the effect of varying the Strouhal number. However, significant discrepancies are apparent between the flow patterns shown in figure 10 of Reference [24] and those shown

in figure 7 of [25]. Also, more recently, Barton [22] made a direct comparison between numerical prediction and Sobey's experimental results at a Reynolds number $Re = 320$ and a Strouhal number $St = 0.012$, and found the predicted length of the vortex structure to be almost twice that observed experimentally. Notwithstanding the need to establish the reason(s) for these discrepancies, the vortex structure of these flow types provides useful validation data. A Reynolds number $Re = 750$ and a Strouhal number $St = 0.006$ are used in the present article with Re and St defined by

$$Re = \frac{U_m h}{v}, \qquad St = \frac{h}{U_m T_0} \tag{36}$$

h, U_m and T_0 denote the step height, the peak velocity and the period of oscillation of the flow respectively. The upstream inlet profile is given by

$$U = U_m \sin \omega t \tag{37}$$

where $\omega = 2\pi/T_0$. The inlet and zero gradient outflow boundaries are fixed at $18h$ and $36h$, upstream and downstream of the step respectively.

Figure 3 shows the streamlines at various points in the first half cycle. The development of a vortex wave is clearly visible and it is just on the point of breaking up at $T = 0.4922$. The eddies are approximately at their peak strength and nearly span the height of the channel. A fuller discussion concerning the physics of this flow can be found in Tutty [25].

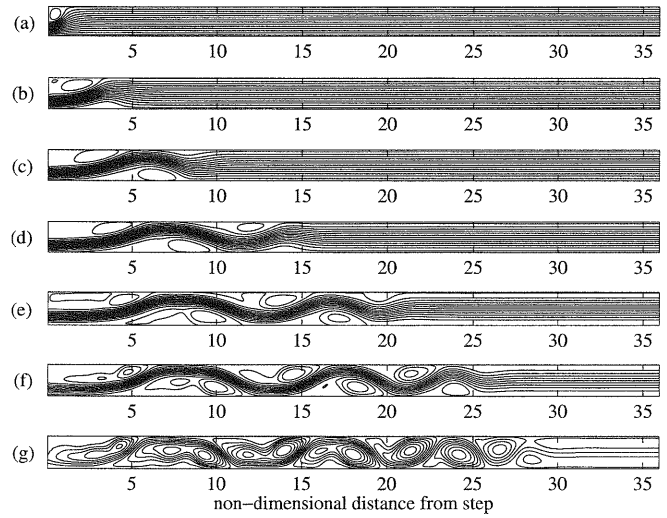

Figure 3. Streamlines of two-dimensional oscillatory flow at various times for $Re = 750$ and $St = 0.006$: (a) $T = 0.0703$; (b) $T = 0.1406$; (c) $T = 0.2109$; (d) $T = 0.2812$; (e) $T = 0.3516$; (f) $T = 0.4219$; (g) $T = 0.4922$.

The lower wall vorticity is shown at the mid-point in the cycle for two grids and for various numbers of processors in Figure 4. The vorticity is non-dimensionalized on the peak velocity and the step height. In all simulations, the domain downstream of the step is subdivided equally, and the upstream section is solved on a separate processor. The time step is $\delta t = 3.125 \times 10^{-6}$, which was found to be the largest time step that could be used on the finest grid without impairing accuracy. The slight discrepancies between the vorticity curves observed for different numbers of processors on Grid D (320×80 control volumes) was found to diminish when doubling the grid density in the x-direction and is almost non-existent, as shown here for Grid E (1280×80 control volumes) which has four times the grid density in the x-direction relative to Grid D. These observations reinforce the findings in the previous subsection concerning the granularity of parallelization. An increase in the grid resolution in the cross-stream direction did not influence the results shown in Figure 4. Indeed, the grid resolution in Grid E is consistent with the grid refinement deemed acceptable in Tutty [25]. Comparing the present results with figure 4(c) in Tutty [25], the vorticity structure is replicated extremely closely, particularly with respect to the location of peak values of vorticity. However, there would appear to be a slightly larger discrepancy between the magnitude of the vorticity at these locations; the values in Tutty [25] are marginally higher than those presented here. The peak values of vorticity are tabulated in Table II for Grid E for the points a–f in Figure 3.

In an attempt to explain the discrepancies between experimental evidence and numerical simulations described above, work is being conducted on three-dimensional simulations of the oscillatory flow reported in Reference [24]. Some of the early results are presented here in

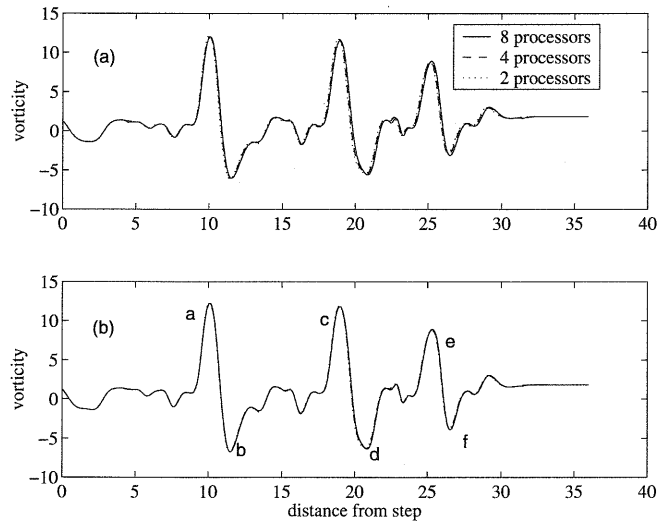

Figure 4. The effect of mesh size and number of processors on lower wall vorticity in two-dimensional oscillatory flow at $Re = 750$ and $St = 0.006$. (a) Grid D (320×80) control volumes, (b) Grid E (1280×80) control volumes.

Table II. Peak values of non-dimensional vorticity (see Figure 4) on the lower wall for two-dimensional oscillatory flow at $Re = 750$, $St = 0.006$, $T = 0.5$ (Grid E).

N	a	b	c	d	e	f
8	12.25 (10.11)	6.69 (11.49)	11.85 (18.97)	6.34 (20.85)	8.91 (25.33)	3.90 (26.54)
4	12.25 (10.08)	6.67 (11.49)	11.89 (18.97)	6.27 (20.80)	8.87 (25.30)	3.81 (26.51)
2	12.34 (10.08)	6.72 (11.46)	11.96 (18.94)	6.23 (20.77)	8.82 (25.30)	3.68 (26.51)

order to underline the need for parallel simulations of such flows. The geometry represents a semi-span of the laboratory experiment with a side-wall and a symmetry plane at either end of the domain. Grid B (320×80) from the two-dimensional simulation above is extruded to 100 cells that are clustered adjacent to the side wall. Nine blocks are used with one upstream of the sudden expansion and eight (equally sized) downstream. Thus, over 3 million computational cells are employed requiring over 3 Gbytes of memory. Using a time step of $\delta t = 3.125 \times 10^{-6}$, a full cycle of the sinusoidal pulse takes 18 days to run. Figures 5–7 depict variations in the streamfunction at the same Reynolds and Strouhal numbers as considered for the two-dimensional simulation and at the same points in the cycle shown in Figure 3. While the development of the vortex wave on the centre line (cf. Figure 5) closely resembles the two-dimensional flow in the first third of the cycle, it appears to break down shortly after this and effectively lose its vortex structure well before $T = 0.5$. Figures 6 and 7 show the streamfunction variation at non-dimensional distances of 3.47 and 5.93 from the centreline (46 and 79 per cent of the semi-span respectively). There is evidence here for the earlier breakdown of the vortex wave

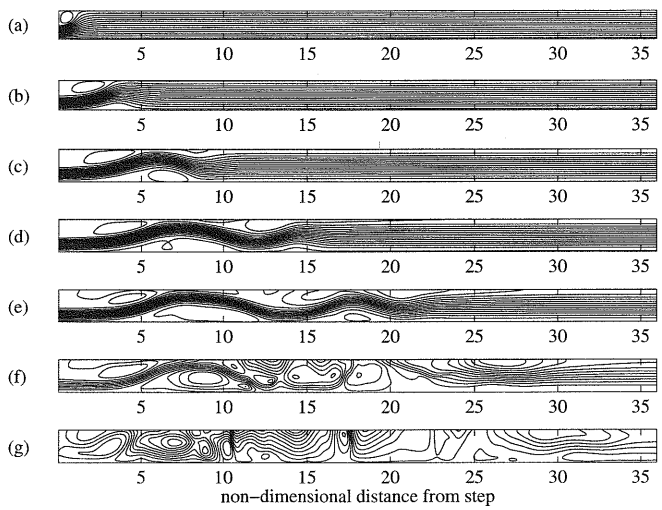

Figure 5. Streamfunction of three-dimensional oscillatory flow at various times for $Re = 750$ and $St = 0.006$ on the centreline (see Figure 3 for key).

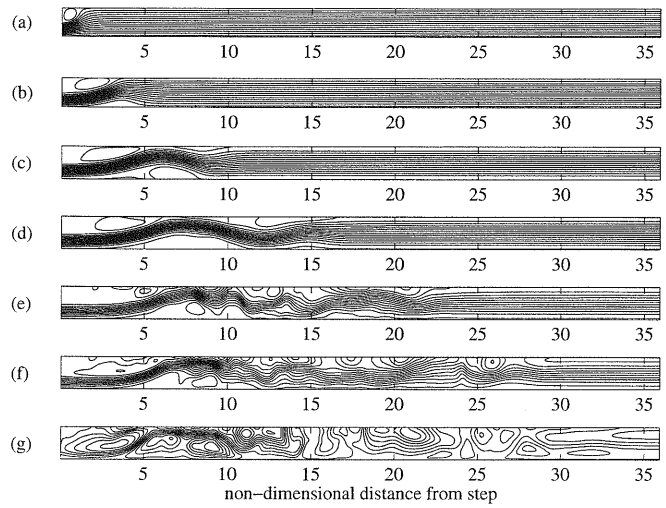

Figure 6. Streamfunction of three-dimensional oscillatory flow at various times for $Re = 750$ and $St = 0.006$ at non-dimensional distance 3.47 from the centreline (see Figure 3 for key).

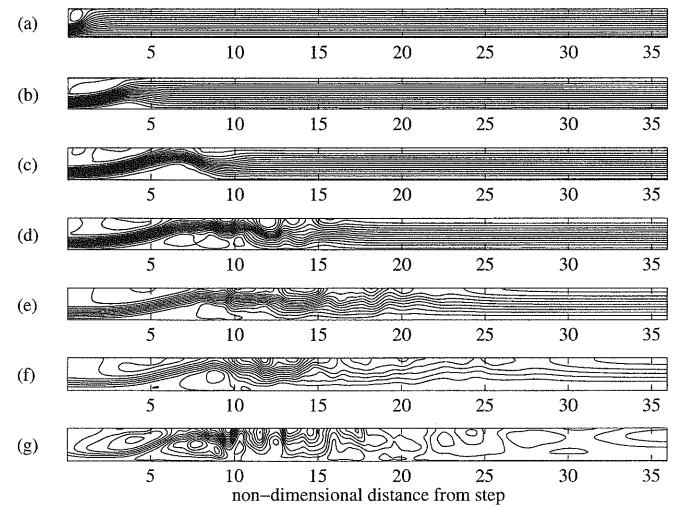

Figure 7. Streamfunction of three-dimensional oscillatory flow at various times for $Re = 750$ and $St = 0.006$ at non-dimensional distance 5.93 from the centreline (see Figure 3 for key).

closer to the side-wall. Interestingly, however, the first two vortex cells immediately downstream of the expansion appear to persist for longer than those on the centreline, and more closely resemble those in the two-dimensional flow. Further analysis is needed to explain these features and the influence of secondary effects in the three-dimensional flow simulations. Particle-tracking techniques are likely to be necessary and this will require vast amounts of data storage. Simulations are also underway at the Reynolds and Strouhal numbers used by Sobey [24] to enable a direct comparison with his experimental flow visualizations. The breakdown of the vortex wave described above is not apparent at the lower Reynolds numbers studied in Reference [24].

Combined with the need to perform a number of simulations across a range of Reynolds numbers and Strouhal numbers, the length of the run-times and the amount of memory required mean that such a study of these three-dimensional flows would not be possible without a parallel code. The parallel PISO algorithm is particularly attractive since it does not require iteration between successive time steps.

3.3. Steady state compressible flow

Three inviscid flow types (subsonic, transonic and supersonic) through a channel with a circular arc bump are used to validate the parallel PISO algorithm applied to compressible flows. For the subsonic and transonic calculations the thickness-to-chord ratio of the bump is 10 per cent, and for the supersonic case it is 4 per cent. In all cases, the width of the channel is equal to the length of the bump, and the length of the channel is three times the length of the bump. Thus, these configurations represent the standard test cases proposed at the Gesellschaft Angewandte Mathematik und Mechanik (GAMM) conference [26].

For the parallel calculations, the domain is decomposed into three streamwise sections: upstream of the bump, the bump itself and downstream of the bump. Non-uniform grids are employed. For the subsonic and transonic cases, the same four grids (from 40×20 to 320×120) are used to assess the grid dependence of the solutions. Also, the same boundary conditions are imposed, the Mach number being fixed by the (constant) total pressure and the fixed static pressure at the downstream outflow boundary. All other variables are extrapolated at the outlet. At the inlet, the flow is defined by fixed total properties and by the pressure that is extrapolated from within the solution domain. Consequently, variation of the Mach number is achieved by simply adjusting the outlet static pressure. The results presented below are for time steps of 2.0×10^{-5} s for the finest grid and 1.0×10^{-4} s for the others; smaller time steps were tested and did not effect the results. The solutions were obtained at the end of a two-step process in which, first, run-time visualization was used to roughly detect when the steady state had been reached followed by a further number of time steps equal to the number reached at that point. In all cases, the solution did not change during this second stage.

At a Mach number $M = 0.5$ the flow remains subsonic throughout the channel; there are no shock waves and the solution is symmetrical. This is shown in terms of isomach lines for the finest grid in Figure 8(a). Comparison between grids for the Mach number distribution on upper and lower walls is depicted in Figure 8(b). Although the two coarsest grids underpredict the peak values, they are within approximately 4 per cent of those for the finest grid.

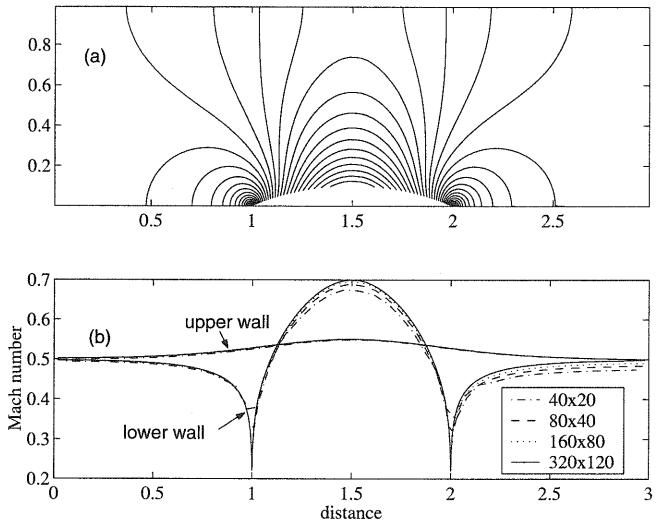

Figure 8. Subsonic inviscid flow through a channel with a circular arc bump in one wall: (a) Mach contours using a grid with 320 × 120 cells; (b) Mach number distribution on upper and lower walls for various grids.

By reducing the outlet static pressure, the Mach number increases, and the flow eventually becomes transonic over a portion of the bump. A shock wave is produced and the flow returns to subsonic downstream of the shock. The isomach lines (for the finest grid) and the Mach number distribution for $M = 0.675$ are shown in Figure 9. The location of the shock wave, the shape of the Mach number variation on the walls and the grid dependence of the solution agree very closely with other recent results [12,13]. However, the maximum Mach number (1.25) is lower in the present work due to the method for imposing boundary conditions which produces an inlet Mach number, $M = 0.645$.

For supersonic flow throughout the channel all variables are specified at the inlet and all are extrapolated at the outlet. At a relatively low (uniform) inlet Mach number, $M = 1.4$, separate oblique shock waves are produced at the leading and trailing edges of the bump. The angle of the leading edge shock is such that it experiences two reflections before merging with the downstream shock at the outflow (cf. Figure 10(a)). At higher Mach numbers, the shocks subtend larger angles with the upstream direction and the leading edge shock undergoes only one reflection within the channel. The isomach contours for such a flow at $M = 1.65$ are shown in Figure 11(a).

A number of difficulties were encountered applying the PISO algorithm (in the form used for the other flow regimes) to the supersonic flows. Two adaptations were applied to generate the results presented here. First, an energy equation was solved (with $\mu = 0$) instead of deriving temperature from the total enthalpy. Second, a blending scheme was implemented for density when interpolating on a cell face using 60 per cent of the UPWIND value and 40 per cent of the central difference value. On the coarse grid, this was found to be the optimum blend in generating a steady state solution. A similar technique was used in Demirdzic *et al.* [12].

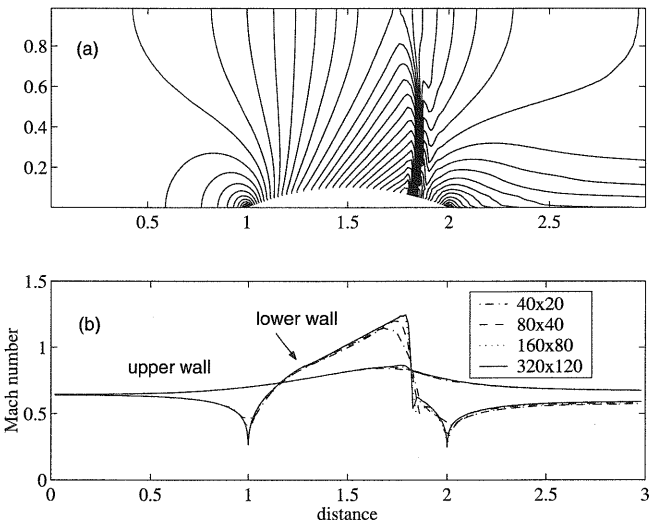

Figure 9. Transonic inviscid flow through a channel with a circular arc bump in one wall: (a) Mach contours using a grid with 320×120 cells; (b) Mach number distribution on upper and lower walls for various grids.

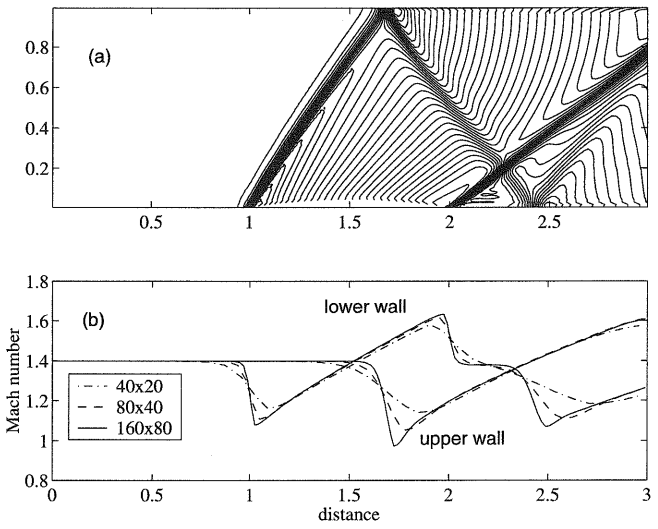

Figure 10. Supersonic inviscid flow through a channel with a circular arc bump in one wall $M = 1.4$: (a) Mach contours using a grid with 160×80 cells; (b) Mach number distribution on upper and lower walls for various grids.

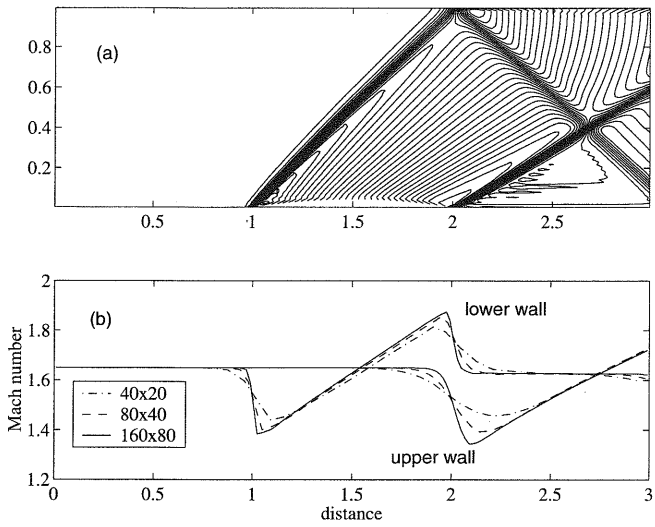

Figure 11. Supersonic inviscid flow through a channel with a circular arc bump in one wall $M = 1.65$: (a) Mach contours using a grid with 160×80 cells; (b) Mach number distribution on upper and lower walls for various grids.

The Mach number variation on the upper and lower walls for inlet Mach numbers $M = 1.4$ and 1.65 are shown in Figures 10(b) and 11(b) respectively. In both cases, refining the grid does not change the positions of the shocks, but it does improve their steepness and resolution. The 160×80 grid is equivalent to the finest grid in Reference [12], and the peak values for $M = 1.65$ differ by less than 4 per cent. A time step of 5.0×10^{-5} s was used for this and the two coarser grids.

An additional problem emerged when attempting to further refine the grid. A number of calculations were attempted on a grid with 320×120 cells and with a number of progressively smaller time steps. Although the solution would initially proceed as for the other grids with shock waves growing from the bump corners, the expected steady state solution could not be obtained. Generally, a large back pressure would develop preventing the sensible development of the flow. While it is hoped to resolve the reason(s) for this failure of the method—in particular, so that it can be usefully applied to transient supersonic flow—this initial investigation highlights the unsuitability of the time stepping approach to steady state, fine grid calculations due to the need for prohibitively expensive small time steps.

4. CONCLUSIONS

This article has described a parallel PISO algorithm employing momentum interpolation on a collocated variable grid arrangement. The requirements for the parallel implementation of the technique have been highlighted. The algorithm applies the same procedure to flows at all

speeds and the implicit treatment of density variation in compressible flows is automatically included in a more straightforward way than in the original PISO method.

Successful application of the method has been demonstrated for the transient development to steady state of flow over a backward-facing step, for the unsteady oscillatory flow through a sudden expansion, and for compressible flow at subsonic, transonic and supersonic speeds through standard bump geometries. Stable, accurate solutions have been obtained in all cases, and shock waves have been captured automatically at the correct locations. However, difficulties have been encountered when employing a relatively fine grid in the supersonic flow. Thus, although the method provides a general purpose tool for modelling steady state and transient flows at all speeds, it is not recommended for steady state solutions on fine grids, particularly at high speed. Techniques are now being investigated to improve the stability of the algorithm under these circumstances.

The large requirements of time and memory for three-dimensional simulations of developing and oscillatory incompressible flow through a sudden expansion have demonstrated the suitability of the parallel algorithm for modelling such flows. A dedicated high performance parallel computer (an SGI Origin 2000) and a cluster of PCs have been used to assess the parallel efficiency of the PISO algorithm. Although the parallel cluster of machines does not match the efficiency of the Origin 2000, the former is significantly less expensive and easily enables the analysis of large problems not studied previously. Further improvements in performance are likely if a high-speed interconnect were to be used to link the separate compute nodes in the cluster.

REFERENCES

1. Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics* 1986; **62**: 40–65.
2. Barton IE. Comparison of simple- and piso-type algorithms for transient flows. *International Journal for Numerical Methods in Fluids* 1998; **26**: 459–483.
3. Jang DS, Jetli R, Acharya S. Comparison of the piso, simpler and simplec algorithms for the treatment of the pressure–velocity coupling in steady flow problems. *Numerical Heat Transfer* 1986; **10**: 209–228.
4. Thakur S, Wright J, Shyy W. A pressure-based composite method with conservative interface treatment. AIAA paper 96-0298, 1996.
5. Majumdar S. Role of underrelaxation in momentum interpolation for calculation of flow with nonstaggered grids. *Numerical Heat Transfer* 1988; **13**: 125–132.
6. Darwish MS. A new high-resolution scheme based on the normalized variable formulation. *Numerical Heat Transfer Part B* 1993; **24**: 353–371.
7. van Leer B. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics* 1974; **14**: 361–370.
8. Kim S-W, Benson TJ. Comparison of the smac, piso and iterative time advancing schemes for unsteady flows. *Computers and Fluids* 1992; **21**(3): 435–454.
9. Issa RI, Gosman AD, Watkins AP. *Journal of Computational Physics* 1986; **62**: 66–82.
10. Issa RI, Javareshkian MH. Application of tvd schemes in pressure-based finite volume methods. *Proceedings of the Fluids Engineering Division Summer Meeting* 1996; **3**: 159–164.
11. Karki KC, Patankar SV. Pressure-based calculation procedure for viscous flows at all speeds in arbitrary configurations. *AIAA Journal* 1989; **27**(9): 1167–1174.
12. Demirdzic I, Lilek Z, Peric M. A collocated flnite.volume method for predicting flows at all speeds. *International Journal for Numerical Methods in Fluids* 1993; **16**: 1029–1050.
13. Date AW. Solution of Navier–Stokes equations on nonstaggered grid at all speeds. *Numerical Heat Transfer Part B* 1997; **33**: 451–467.

14. Farhanieh M, Golafshani M, Shayeste SA. Numerical solution of compressible flows using a pressure-based method. *Iranian Journal of Science and Technology* 1998; **22**(3): 329–346.
15. Beam RM, Warming RF. An implicit factor scheme for the compressible navier-stokes equations. *AIAA Journal* 1978; **16**: 393–402.
16. MacCormack RW, Paullay AJ. Computational efficiency achieved by time splitting of the finite difference operators. AIAA Paper 72-154, 1972.
17. Soh W, Goodrich J. Unsteady solution of incompressible navier-stokes equations. *Journal of Computational Physics* 1988; **79**: 113–134.
18. Ferziger HJ, Peric M. *Computational Methods for Fluid Dynamics*. Springer: Berlin, 1996.
19. Melaaen MC. Calculation of fluid flows with staggered and nonstaggered curvilinear nonorthogonal grids—the theory. *Numerical Heat Transfer Part B* 1992; **21**: 1–19.
20. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, 1980.
21. Armaly BF, Durst F, Pereira CF, Schonung B. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics* 1983; **127**: 473–496.
22. Barton IE. Improved laminar predictions using a stabilized time-dependent simple scheme. *International Journal for Numerical Methods in Fluids* 1998; **28**: 841–857.
23. Chiang TP, Shen TWH. A numerical revisit of backward-facing step flow problem. *Physics of Fluids* 1999; **11**(4): 862–874.
24. Sobey IJ. Observation of waves during oscillatory channel flow. *Journal of Fluid Mechanics* 1985; **151**: 395–426.
25. Tutty OR, Pedley TJ. Oscillatory flow in a stepped channel. *Journal of Fluid Mechanics* 1993; **247**: 179–204.
26. Rizzi A, Viviand H (eds). Numerical methods for the computation of inviscid transonic flow with shock waves. In *Notes on Numerical Fluid Mechanics*. Vieweg: Braunschweig, 1981.
27. Gartling DK. A test problem for outflow boundary conditions—flow over a backwardfacing step. *International Journal for Numerical Methods in Fluids* 1990; **11**: 953–967.
28. Sani RL, Gresho PM. Resume and remarks on the open boundary condition minisymposium. *International Journal for Numerical Methods in Fluids* 1994; **18**: 983–1008.