Received August 1995
Revised April 1996

LIMPO: an improved version of the PISO algorithm for turbulent swirling flows

V. Semião and M.G. Carvalho
Mechanical Engineering Department, Instituto Superior Técnico, University of Lisbon, Portugal

Introduction

The accuracy of computed complex flows is hindered not only by the validity of the particular physical assumptions employed, but also by the accuracy of the numerical schemes used to discretize the set of governing partial differential equations. In order to reduce the inherent discretization errors present in all first-order solution procedures commonly employed these days, a numerical grid comprising a high number of grid points is required, leading to very large central processing unit (CPU) times and highly expensive computations. This situation is particularly severe for complex flows where the detailed geometry has to be described by a large number of grid nodes.

One way of reducing the CPU time is to improve the method of solving the pressure-velocity coupled system. The best established methods are the SIMPLE method[1], the SIMPLER[2] and the SIMPLEST[3,4] in which the equations for each variable are solved repeatedly in succession. Others utilize block iteration, such as the SIVA scheme[5], in which the "variables" block is solved simultaneously for a single point (or line). Another iterative method of handling the pressure-velocity coupling arising in the implicitly discretized fluid flow equations – PISO – was presented by Issa[6]. This method was applied by Issa *et al.*[7] to the computation of two cases of transient axisymmetric laminar flow in circular ducts with abrupt enlargement in both compressible and incompressible situations. The results of the computations were compared with another existing iterative method, and the PISO appeared to be faster than its iterative counterpart for transient flows, either compressible or incompressible, and it exhibited a stable behaviour for large time-step sizes which makes it a reliable technique for steady-state calculations.

The choice of the main dependent variables for fluid flows must be such that the primitive variables (i.e., velocity, pressure and density) should be restrained in the equations as the working variables. Either the density is chosen to stand as a main dependent variable, wherein the pressure is evaluated from it via an equation of state, or the opposite is done. The many existing methods developed specifically for incompressible flows, for example Patankar and Spalding[1], choose to treat the pressure, rather than the density, as a main dependent variable. In order to determine the pressure which, while appearing in each of the momentum equations, vanishes from the continuity relation in the incompressible limit, a pressure equation is usually derived by joint manipulation of those equations (e.g. [2,8]). The resulting pressure equation replaces the continuity relation while the momentum equations retain their role for determining the velocity field, the equations set being coupled via the pressure and the velocities.

In the present paper an iterative method for handling the pressure-velocity coupling of the fully-implicitly discretized equations, for two-dimensional turbulent axisymmetric steady flows, with and without swirl, is presented. The new method (LIMPO for Linking Implicit Method for Pressure with Operators-splitting) is based on the PISO algorithm and utilizes the splitting of operations in the solution of the momentum and pressure equations. In this algorithm, the calculation of the turbulent quantities (k and ε) as well as the tangential velocity (w) are embedded into the standard PISO algorithm through the introduction of new predictor and corrector steps for those quantities. This leads to a further reduction of the computational effort required to achieve the converged solution, as far as the number of iterations and the CPU time are concerned. In the work of Carvalho *et al.*[9], this kind of procedure was already followed by embedding the temperature calculations in the PISO algorithm, for the solution of the energy and aerodynamic equations of the laminar batch flow in an industrial glass furnace tank.

Comparisons of the predicted results and of the computational effort, using the LIMPO algorithm, are made herein against computations with existing iterative methods employing the same spatial difference practices. Two different geometries were used to compare and draw conclusions about the computational effort required by the three methods: a circular step and a circular step preceded by a conical quarl.

The SIMPLE algorithm is a very well-established iterative procedure, although it has to some extent been superseded by SIMPLER and SIMPLEST algorithms[4]. The latter exhibits an efficiency frequently comparable to that of PISO. From the above mentioned algorithms, using pressure and velocity as working variables, SIMPLE and PISO algorithms were chosen to assess the potential of LIMPO, by comparison of their performances. The results have shown a much better performance of the LIMPO algorithm as it allows a considerable reduction in the number of iterations and in the CPU time required to obtain exactly the same converged solution.

Mathematical model

The transport equations

The governing equations of a two-dimensional axisymmetric turbulent flow, with swirl, can be written in a general form, which runs:

$$\frac{\partial}{\partial x}(\rho u\Phi)+\frac{1}{r}\frac{\partial}{\partial r}(\rho rv\Phi)=\frac{\partial}{\partial x}(\Gamma_{\Phi}\frac{\partial\Phi}{\partial x})+\frac{1}{r}\frac{\partial}{\partial r}(r\Gamma_{\Phi}\frac{\partial\Phi}{\partial r})+S_{\Phi} \qquad (1)$$

with the corresponding values of Φ, Γ_Φ and S_Φ indicated in Table I. The quantities C_μ, C_1, C_2, C_D, σ_k and σ_ε appearing in Table I are standard constants of the k-ε turbulence model (see, e.g. [10]), and are used in these predictions.

Variable Φ	Γ_Φ	S_Φ
1	0	0 (continuity equation)
u	μ_{eff}	$-\frac{\partial p}{\partial x}+\frac{1}{r}\frac{\partial}{\partial r}(r\mu_{eff}\frac{\partial v}{\partial x})+\frac{\partial}{\partial x}(\mu_{eff}\frac{\partial u}{\partial x})$
v	μ_{eff}	$-\frac{\partial p}{\partial r}+\frac{1}{r}\frac{\partial}{\partial r}(r\mu_{eff}\frac{\partial v}{\partial r})+\frac{\partial}{\partial x}(\mu_{eff}\frac{\partial u}{\partial r})-\frac{2\mu_{eff}v}{r^2}+\frac{\rho w^2}{r}$
w	μ_{eff}	$-\frac{\rho vw}{r}-\frac{1}{r}\frac{\partial}{\partial r}(\mu_{eff}w)+\mu_{eff}\frac{\partial(w/r)}{\partial r}$
k	$\frac{\mu_{eff}}{\sigma_k}$	$G_k-C_D\rho\varepsilon$
ε	$\frac{\mu_{eff}}{\sigma_\varepsilon}$	$\frac{C_1\varepsilon G_k-C_2\rho\varepsilon^2}{k}$

where:

$$\mu_{eff}=\mu+\mu_t$$

$$\mu_t=\rho C_\mu k^2/\varepsilon$$

$$G_k=2\mu_t[(\frac{\partial u}{\partial x})^2+(\frac{\partial v}{\partial r})^2+(\frac{v}{r})^2]+$$

$$+\mu_t[(\frac{\partial u}{\partial r}+\frac{\partial v}{\partial x})^2+(\frac{\partial w}{\partial x})^2+(\frac{\partial w}{\partial r}-\frac{w}{r})^2]$$

Table I. The source terms S_Φ and the diffusion coefficients Γ_Φ for different Φ variables of equation (1)

Finite difference formulation

The staggered grid arrangement in which the nodes for velocities are located in between the grid nodes for pressure is used. The discretization of the transport equations is effected by finite volume techniques for which control volumes (or cells) surrounding each variable location are defined. Standard practices employed in earlier works[1,2] were followed to formulate the difference equations, hence only a brief description of it is provided below. Spatial variations – convective and diffusive terms in equation (1) – were approximated by a hybrid upwind/central difference formula giving first to second-order spatial accuracy.

Integration of equation (1) over a cell yields the final difference equation:

$$A_P\Phi_P = A_N\Phi_N + A_S\Phi_S + A_E\Phi_E + A_W\Phi_W + \overline{S_\Phi} \tag{2}$$

where:

$$\overline{S}_\Phi = \int_s^n \int_w^e S_\Phi r dr dx \tag{3a}$$

$$A_{N,S,E,W} = M_{n,s,e,w}\alpha_{n,s,e,w} \tag{3b}$$

$$A_P = A_N + A_S + A_E + A_W + \mathrm{div}M \tag{3c}$$

$$\mathrm{div}M = M_n - M_s + M_e - M_w \tag{3d}$$

In the previous equations, M_i stands for the mass flux through the cell face i, and α_i is a weighing factor depending on the local (cell) Peclet number – Pe_i – as follows:

$$\begin{aligned} \alpha_i &= \frac{1}{2}(1 + \frac{2}{Pe_i}) && |Pe_i| \le 2 \\ &= 1 && Pe_i > 2 \\ &= 0 && Pe_i < 2 \end{aligned} \tag{3e}$$

The finite difference equation (2), it should be noted, corresponds to the momentum equations (when Φ stands for u and v), to the continuity equation (Φ = 1), to the turbulence model equations (Φ = k, ε) and to the tangential velocity transport equation (Φ = w), as described in Table I.

The pressure equation is derived from the combination of the continuity and momentum equations as presented below. For the sake of convenience, a symbolic operator form, which caters for most of the widely-used spatial formulae, is employed from now on to present the discretized equations. Rewriting equation (2), for the u and v velocity components, using the symbolic operator form, it runs:

$$A_P u_P = H(u) - \Delta_x p + S_u \tag{4}$$

$$A_P v_P = H(v) - \Delta_r p + S_v \tag{5}$$

where Δ_i is the finite difference expression of $(\partial/\partial x_i)$ and $H(\Phi_j) = \sum_{j=N,S,E,W} A_J \Phi_J$.

The mass balance for a cell becomes:

$$M_n - M_s + M_e - M_w = 0 \tag{6}$$

Equations (4) and (5) give the values of u and v velocities at the interfaces of the Φ cells, which are used to update the flux values M_e, M_w, M_n and M_s of the continuity equation (6). Substitution of equations (4) and (5) into equation (7) yields the following discretized pressure equation:

$$D_p p_P = J(p) - \mathrm{div}\bar{H} - \mathrm{div}\bar{S}_P \tag{7}$$

that can also be obtained by taking the divergence of the momentum equations (4) and (5), and where:

$$J(p) = D_n p_N + D_s p_S + D_e p_E + D_w p_W \tag{8a}$$

$$D_p = \sum_{j=n,s,e,w} D_j \tag{8b}$$

$$D_{i=n,s,e,w} = \left(\frac{\rho a^2}{A_P^{v,u}}\right)_{i=n,s,e,w} \tag{8c}$$

$$\bar{H}_{i=n,s,e,w} = \left(\frac{\rho a H(u,v)}{A_P^{v,u}}\right)_{i=n,s,e,w} \tag{8d}$$

$$\bar{S}^P_{\;i=n,s,e,w} = \left(\frac{\rho a S^P}{A_P^{v,u}}\right)_{i=n,s,e,w} \tag{8e}$$

In the previous and following equations **a** stands for the cell-face area.

Methodology of LIMPO

The splitting of operations (or factorization) is not a new concept and is often invoked either in temporal differencing or in the solution of the discretized equations as in the Alternating Direction Implicit (ADI) technique. However, an extension of that principle to apply to the coupling between variables, namely the pressure and velocities in turbulent swirling flows, whereby operations involving different variables are split into a series of predictor-corrector steps, has not been previously reported.

Letting the superscripts *, ** and *** denote intermediate field values obtained during the splitting process, and **n** the iteration level, and using a two-stage scheme, similar to the one used by Issa[6], the factorization of the momentum and pressure equations runs as follows:

Momentum predictor step. The equations for momentum (4) and (5) are solved implicitly in this step, using guessed values (or values from previous iteration – *n* –) for pressure and source terms:

$$A_P u_P^* = \sum_{j=N,S,E,W} A_j u_j^* + D_u(p_w^n - p_e^n) + S_u^n \tag{9a}$$

$$A_P v_P^* = \sum_{j=N,S,E,W} A_j v_j^* + D_v(p_s^n - p_n^n) + S_v^n \tag{9b}$$

The solution of these equations yields the u^* and v^* fields, that satisfy the momentum equations but do not satisfy the continuity equation.

Momentum first corrector step. The first correction equations for momenta are:

$$A_P u_P^{**} = \sum_{j=N,S,E,W} A_j u_j^{**} + D_u(p_w^* - p_e^*) + S_u^n \tag{10a}$$

$$A_P v_P^{**} = \sum_{j=N,S,E,W} A_j v_j^{**} + D_v(p_s^* - p_n^*) + S_v^n \tag{10b}$$

Subtracting equations (9) from equations (10) one obtains:

$$A_P(u_P^{**} - u_P^*) = \sum_{j=N,S,E,W} A_j(u_j^{**} - u_j^*) + D_u[\,(p^* - p^n)_w - (p^* - p^n)_e\,] \tag{11a}$$

$$A_P(v_P^{**} - v_P^*) = \sum_{j=N,S,E,W} A_j(v_j^{**} - v_j^*) + D_v[\,(p^* - p^n)_s - (p^* - p^n)_n\,] \tag{11b}$$

Combining the divergence of the previous equations with the continuity equation (6), and ignoring, as in Patankar[2], the terms $\sum_j A_j(u_j^{**} - u_j^*)$ and $\sum_j A_j(v_j^{**} - v_j^*)$, the pressure correction equation becomes:

$$A_P(p_P^* - p_P^n) = \sum_{j=N,S,E,W} A_j(p_j^* - p_j^n) + S_P^* \tag{12}$$

where:

$$A_P = \sum_{j=N,S,E,W} A_j \tag{13a}$$

$$A_{j=N,S,E,W} = \frac{\rho a_{j=N,S,E,W} D_{v,u}}{A_P^{v,u}} \tag{13b}$$

$$S_P^* = M_s^* - M_n^* + M_w^* - M_e^* \tag{13c}$$

Equation (12) is solved implicitly for $(p^* - p^n)$, which is then inserted into equations (11a) and (11b) to get explicitly the corrected velocities u^{**} and v^{**}. These velocities completely satisfy the continuity equation but satisfy only approximately the momentum equations, as the terms $\sum_j A_j(u_j^{**} - u_j^*)$ and $\sum_j A_j(v_j^{**} - v_j^*)$ were ignored. In order to overcome this approximation, the following steps are added to the process of operations splitting.

*Turbulent parameters (*k *and* ε*) and tangential velocity (*w*) predictor step.* This is a new step embedded in the standard PISO algorithm procedure. It consists of the evaluation of the source terms of u and v momentum equations using the most recently calculated values of the participating variables (k, ε and w).

The w^* velocity and the turbulent quantities k^* and ε^* are then calculated by solving their finite-difference equations (2), using the u^{**} and v^{**} values to evaluate the convective terms of their transport equations. The values of w^* will affect the equations of motion directly through the source term of the v velocity, as shown in Table I, while the k^* and ε^* quantities will affect those equations only indirectly through the value of $\mu_t^* = \rho C_\mu (k^*)^2/\varepsilon^*$.

Momentum second corrector step. With these predicted values of w^*, k^* and ε^*, the source terms of u and v velocities, respectively S_u^* and S_v^*, must be updated and included in the momentum equations. The second momentum correction equations are:

$$A_P u_P^{***} = \sum_{j=N,S,E,W} A_j u_j^{**} + D_u(p_w^{**} - p_e^{**}) + S_u^* \tag{14a}$$

$$A_P v_P^{***} = \sum_{j=N,S,E,W} A_j v_j^{**} + D_v(p_s^{**} - p_n^{**}) + S_v^* \tag{14b}$$

Subtracting equations (10) from equations (14) the following equations are obtained:

$$A_P(u_P^{***} - u_P^{**}) = \sum_{j=N,S,E,W} A_j(u_j^{**} - u_j^*) + \\ + D_u[(p^{**} - p^*)_w - (p^{**} - p^*)_e] + (S_u^* - S_u^n) \tag{15a}$$

$$A_P(v_P^{***} - v_P^{**}) = \sum_{j=N,S,E,W} A_j(v_j^{**} - v_j^*) + \\ + D_v[(p^{**} - p^*)_s - (p^{**} - p^*)_n] + (S_v^* - S_v^n) \tag{15b}$$

Again, combining the divergence of the previous equations with the continuity equation, the second pressure correction equation is obtained and is equal to:

$$A_P(p_P^{**} - p_P^*) = \sum_{j=N,S,E,W} A_j(p_j^{**} - p_j^*) + S_P^{**} \tag{16}$$

where:

$$A_P = \sum_{j=N,S,E,W} A_j \tag{17a}$$

$$A_{j=N,S,E,W} = \frac{\rho a_{j=N,S,E,W} D_{v,u}}{A_P^{v,u}} \tag{17b}$$

$$S_P^{**} = D_s^{**} - D_n^{**} + D_w^{**} - D_e^{**} \tag{17c}$$

$$D^{**}_{j=n,s,e,w} = [\frac{1}{A_P^{v,u}} (\sum_{i=N,S,E,W} A_i^{v,u} (u_i^{**} - u_i^{*}) + (S^{*}_{v,u} - S^{n}_{v,u})) \rho a]_{j=n,s,e,w} \quad (17d)$$

By solving implicitly equation (16), the second correction values for the pressure field ($p^{**} - p^{*}$) are evaluated. Inserting these values into equations (15a) and (15b) the velocity fields u^{***} and v^{***} are calculated explicitly. It should be noted that the u^{***} and v^{***} values satisfy simultaneously the continuity and the momentum equations.

Turbulent parameters (k *and* ε) *and tangential velocity* (w) *corrector step.* This is a new step embedded in the standard PISO algorithm procedure.

The w^{**} velocity and the turbulent quantities k^{**} and ε^{**} are then calculated, again by solving their finite-difference equations (2), and using the u^{***} and v^{***} fields to calculate their convective terms.

The values of u^{***}, v^{***}, w^{**}, k^{**}, ε^{**} and p^{**} constitute the set of starting values for the following iteration or, in the iteration of convergence, the set of the final solution values.

The test cases – presentation and discussion of results

The aim of the present work is the demonstration of the capability of the numerical algorithm presented herein, when compared with other algorithms, rather than the validation of the predicted results by comparison with experimental values. In order to verify the savings in computing effort achieved with the LIMPO algorithm, when compared with the SIMPLE and the PISO algorithms, two axisymmetric geometries were chosen to predict the turbulent, swirling and non-swirling flows within them. The predicted values using any of the above mentioned algorithms coincide, the number of iterations and the required CPU time being the parameters of comparison.

Tube with sudden enlargement

Figure 1 shows the first geometry used to test the LIMPO algorithm, consisting of a duct with a sudden enlargement of entry. The mass flow of water is 9.61 ×

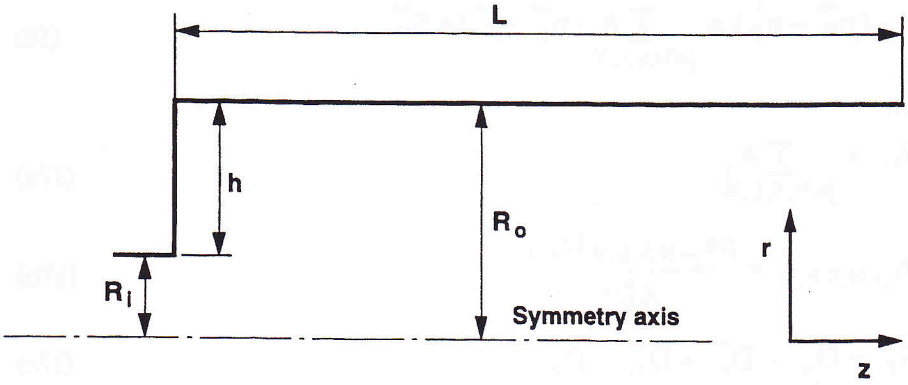

Figure 1.
Sketch of the tube with single sudden enlargement (test case 1)

10^{-3} kgs^{-1}, at a temperature of 300K, and an inlet velocity of 0.5 ms-1, corresponding to a Reynolds number of 7.75×10^3. The dimensions of the tube are, according to the nomenclature used in Figure 1, $R_0 = 1.24 \times 10^{-2}$ m, R_i/R_0 = 0.5 and L/R_0 = 16. In order to access the sensitivity of the method to the number of nodes used in the simulations, predictions were performed for grids comprising 16 × 16, 32 × 32 and 64 × 64 nodes in the z and r directions, constituting RUN 1, RUN 2 and RUN 3 respectively. The numerical results were considered converged when the normalized residuals of the governing equations were less than 5×10^{-4}. The inlet swirl number S (see [11]),

$$S = \frac{G_\theta}{G_z r} \tag{18}$$

where r is the nozzle radius and G_θ and G_z are, respectively, the axial flux of swirl momentum and the axial flux of axial momentum, was varied between 0 and 1.5, with intermediate values of 0.3, 0.5, 0.8 and 1.

Figures 2, 3, 4 and 5 show the axial velocity profiles – U velocity – at different axial stations, computed from the three grids used, for swirl numbers equal to 0, 0.5, 1 and 1.5. The predicted values are independent of the algorithm used. It can be seen that the results produced by the coarsest grid are not satisfactory for swirling flows. Indeed, as depicted in Figures 3-5, the radial dimension of the central recirculation zone induced by the inlet swirling flow is underpredicted by the coarsest grid.

In order to compare the performance of the new algorithm (LIMPO) with those of SIMPLE and PISO, an optimization of the under-relaxation factors of the calculated variables was performed. Figure 6 contains the values for the optimized factors used to obtain the results presented herein. It is clear from the table that the LIMPO algorithm allows for higher values of the under-relaxation factors. This effect becomes more evident with the refinement of the grid. The allowance for looser under-relaxation is due to the embedding of the k, ε and w calculations into the standard PISO algorithm, that permits the calculation of all variables in two steps (predictor and corrector).

Figure 7 contains a comparison of the performance of SIMPLE, PISO and LIMPO algorithms for different swirl numbers, used in the predictions with the three above mentioned grids. As it can be observed, LIMPO algorithm always performs better than the two other schemes for both the required CPU time and number of iterations to achieve convergence (the CPU time is referred to a VAX DEC 7620). When compared with SIMPLE and PISO algorithms, LIMPO allows the computation of the same numerical results with a reduction of, at least, 35 per cent of the number of iterations. In most cases this reduction exceeds the value of 50 per cent. Although always with a decrease in its value when compared with the other algorithms, the CPU time required by the LIMPO scheme exhibits in some cases a more discrete reduction – see Figure 6, RUN 3, swirl number of 0.8. This is due to the extra time consumed during steps c and

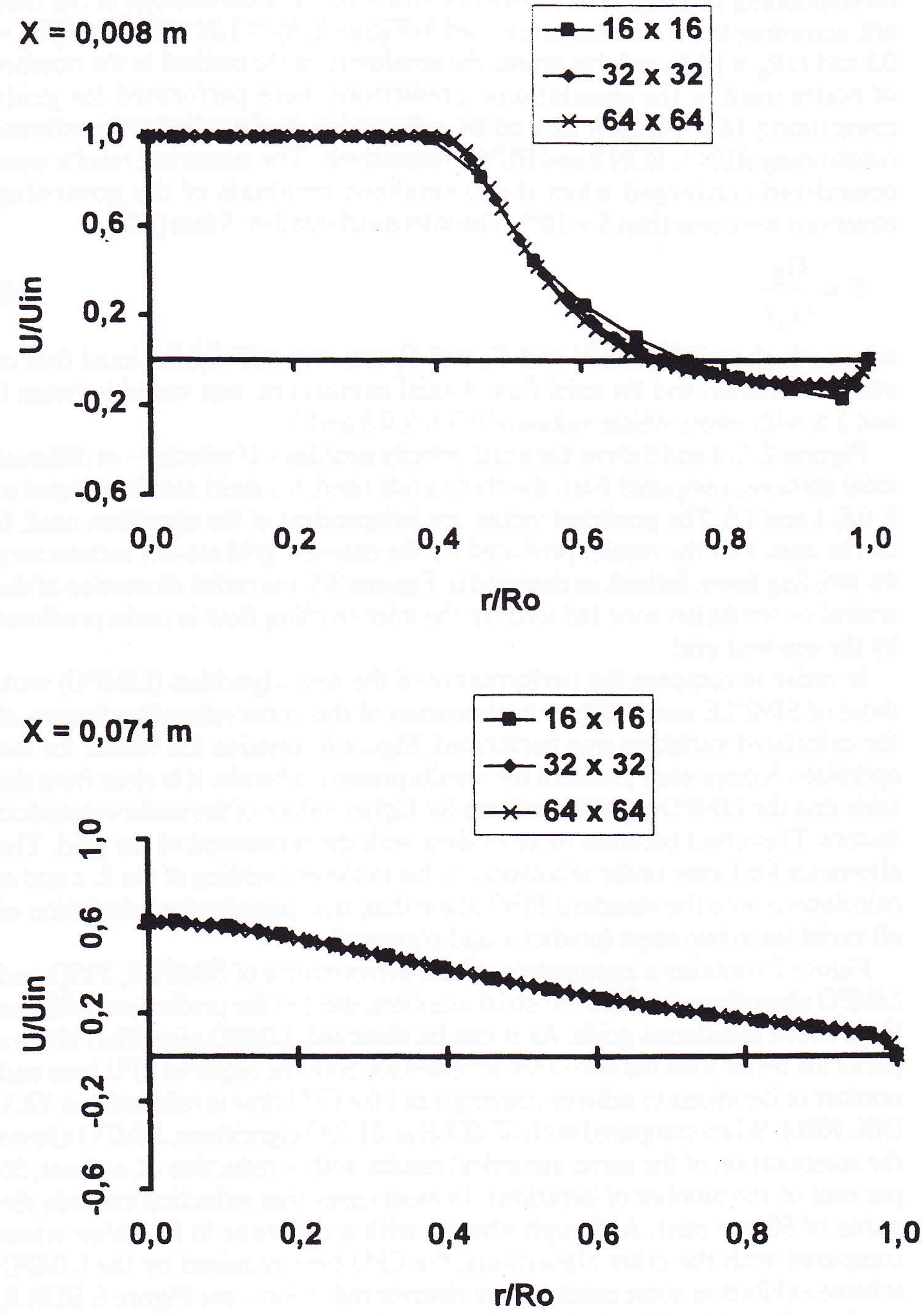

Figure 2.
Non-dimensional axial velocity profiles (test case 1 – swirl number = 0)

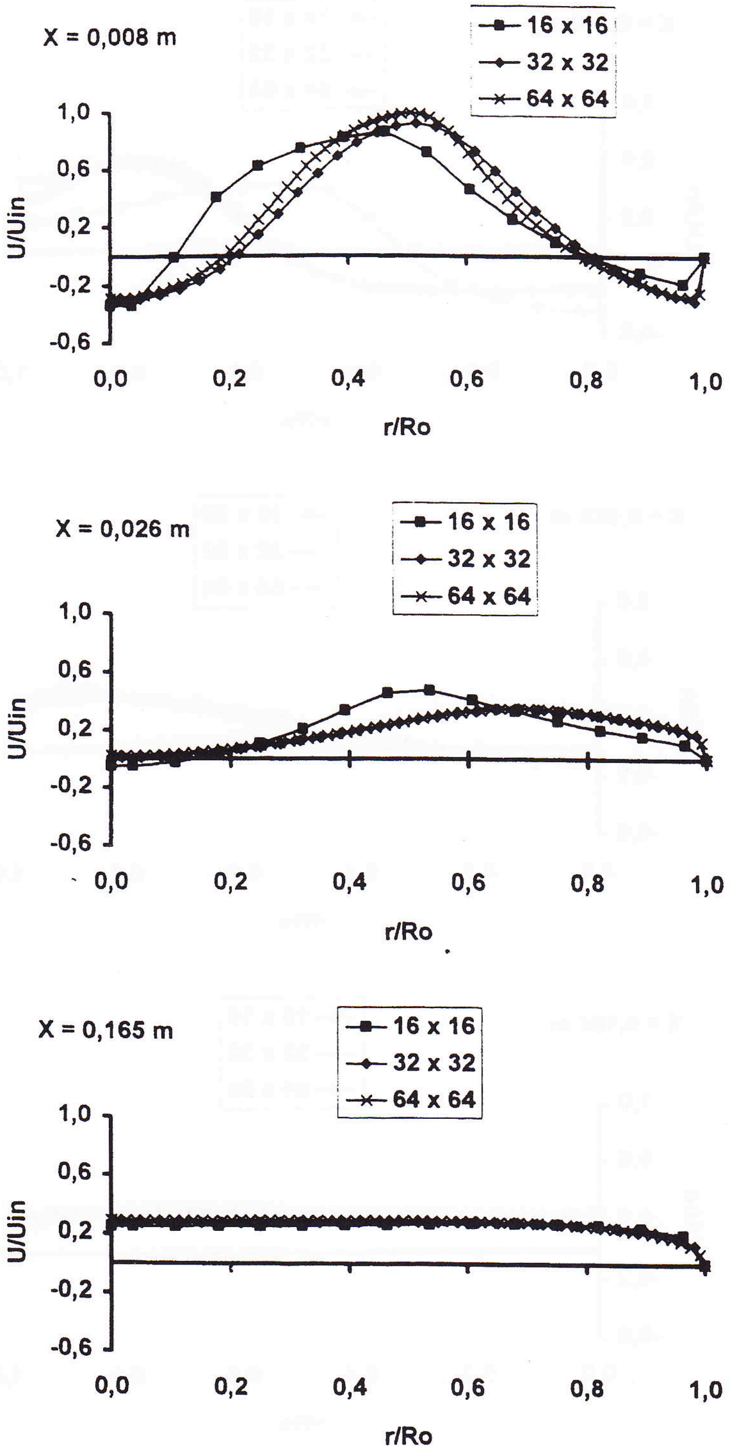

Figure 3. Non-dimensional axial velocity profiles (test case 1 – swirl number = 0.5)

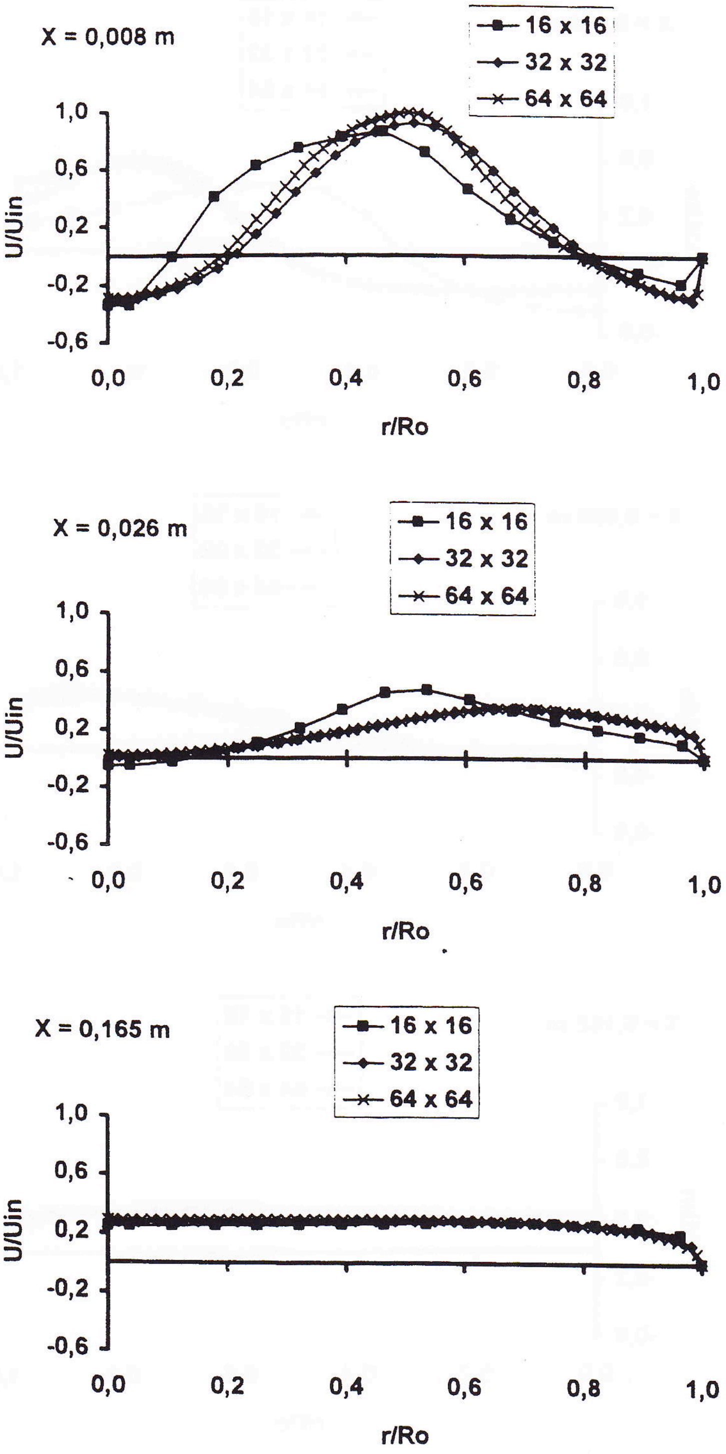

Figure 3.
Non-dimensional axial velocity profiles (test case 1 – swirl number = 0.5)

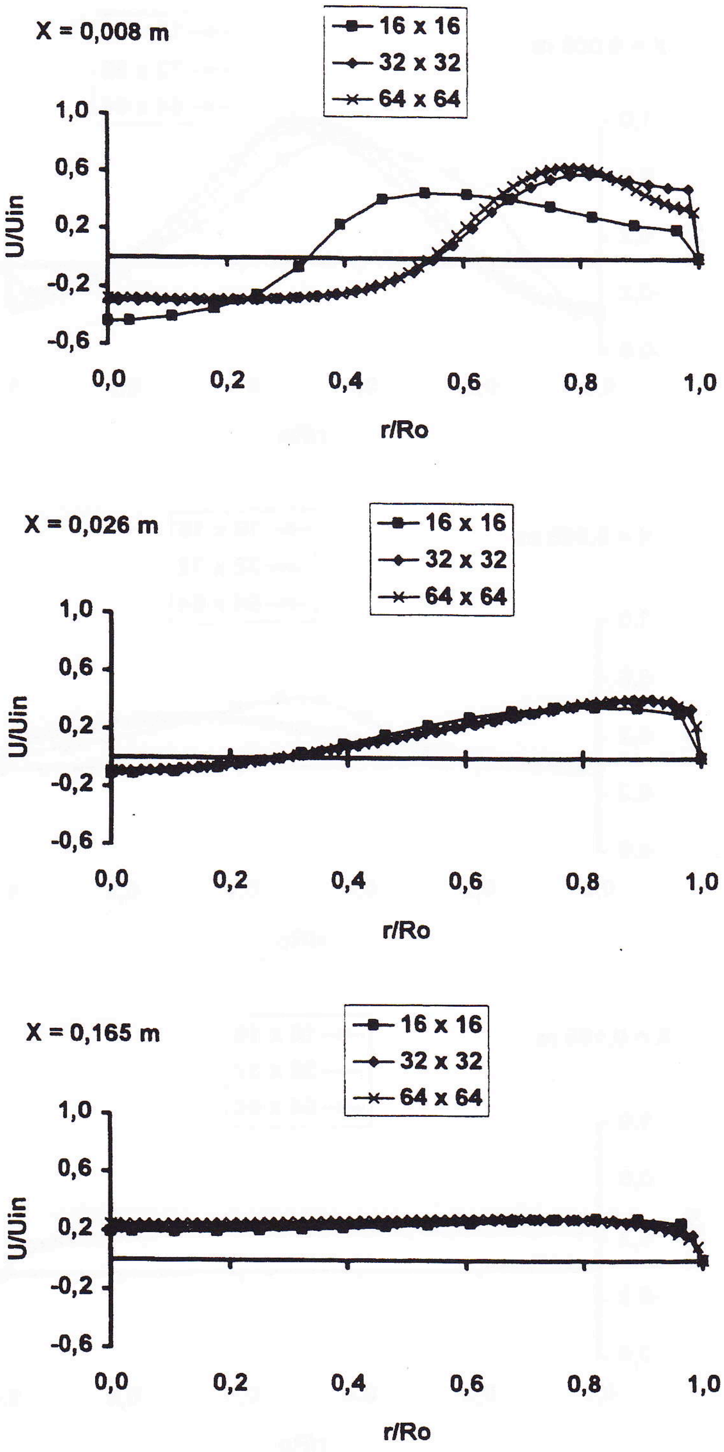

Figure 4.
Non-dimensional axial velocity profiles (test case 1 – swirl number = 1)

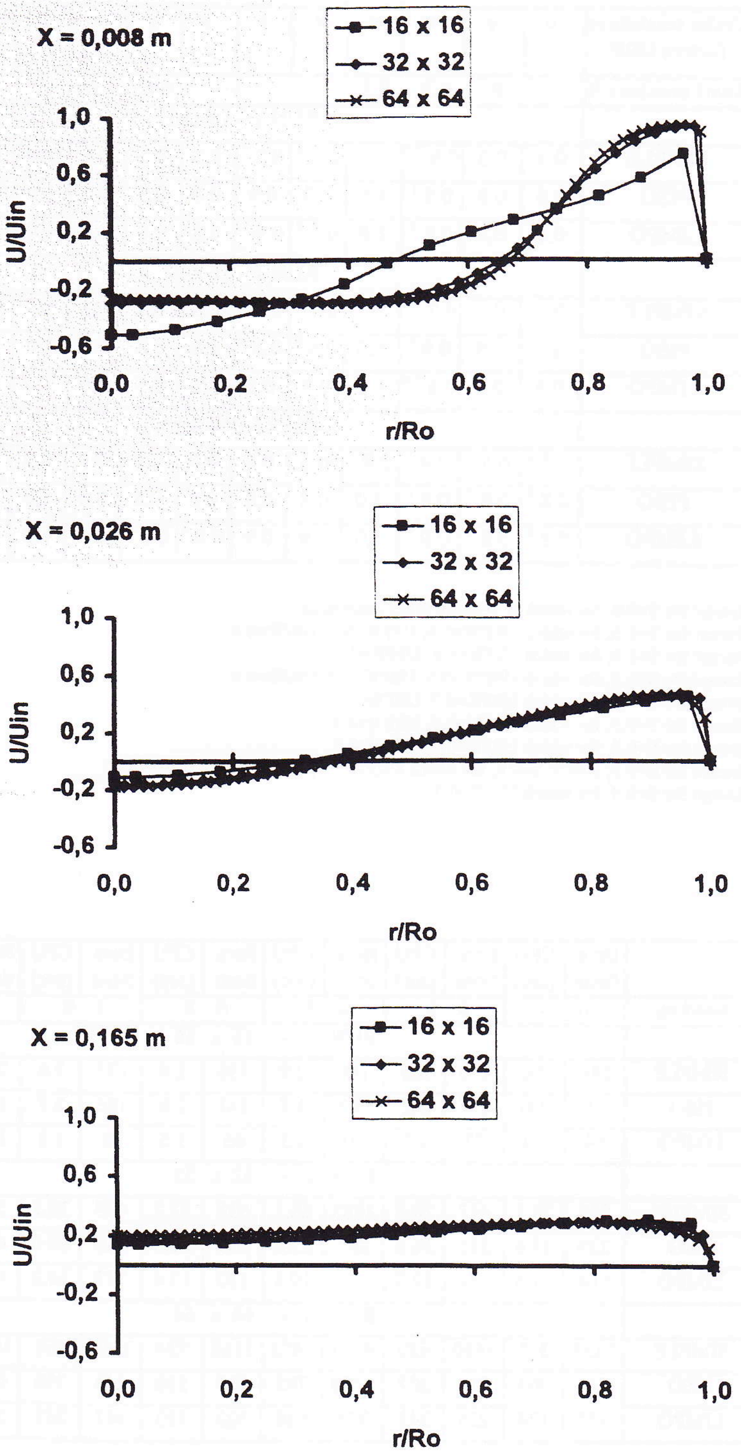

Figure 5.
Non-dimensional axial velocity profiles (test case 1 – swirl number = 1.5)

	Under relaxation factors URF	u	v	w	p	k	ε	u	v	w	p	k	ε
	Swirl numbers S	0; 0.3; 0.5						0.8; 1.0; 1.5					
		RUN 1 - 16 x16											
	SIMPLE	0.5	0.5	0.8 [1,2]	1.0	0.9 [2]	0.9 [2]	0.5	0.5	0.8	1.0	0.9	0.9
A	PISO	0.8	0.8	0.8 [1]	1.0	0.9	0.9	0.8	0.8	0.8	1.0	0.9 [3]	0.9 [3]
L	LIMPO	0.8	0.8	0.8 [1]	1.0	0.9 [2]	0.9 [2]	0.8	0.8	0.8	1.0	0.9	0.9
G		RUN 2 - 32 x 32											
O	SIMPLE	0.5	0.5	0.7 [1,4]	1.0	0.7 [4]	0.7 [4]	0.5	0.5	0.8	1.0	0.8	0.8
R	PISO	0.8	0.8	0.8 [1]	1.0	0.8 [5]	0.8 [5]	0.8	0.8	0.8	1.0	0.8	0.8
I	LIMPO	0.8	0.8	0.8 [1,4]	1.0	0.9	0.9	0.8	0.8	0.9	1.0	0.9	0.9
T		RUN 3 - 64 x 64											
H	SIMPLE	0.5 [6]	0.5 [6]	0.9 [1]	0.8 [9]	0.8	0.8	0.5	0.5	0.9	0.8	0.9	0.9
M	PISO	0.8 [7]	0.8 [7]	0.8 [1]	1.0	0.8	0.8	0.8	0.8	0.8	1.0	0.8	0.8
	LIMPO	0.8 [8]	0.8 [8]	0.9 [1]	1.0	0.9	0.9	0.8 [8]	0.8 [8]	0.9	1.0	0.9	0.9

1 - Except for S=0.0, for which w velocity is not calculated
2 - Except for S=0.5, for which URFW=0.6, URFK=0.8, URFE=0.8
3 - Except for S=1.5, for which URFK=0.8, URFE=0.8
4 - Except for S=0.5, for which URFW=0.9, URFK=0.9, URFE=0.9
5 - Except for S=0.5, for which URFK=0.7, URFE=0.7
6 - Except for S=0.5, for which URFU=0.6, URFV=0.6
7 - Except for S=0.3, for which URFU=0.7, URFV=0.7
8 - Except for S=0.3; S=0.5; S=0.8, for which URFU=0.9, URFV=0.9
9 - Except for S=0.5, for which URFP=0.7

Figure 6.
The under-relaxation factors used with the three studied algorithms for test case 1

		Itera tions	CPU (sec)	Itera tions	CPU (sec)	Itera tions	CPU (sec)	Itera tions	CPU (sec)	Itera tions	CPU (sec)	Itera tions	CPU (sec)
	Swirl no	0. 0		0. 3		0. 5		0. 8		1. 0		1. 5	
		RUN 1 - 16 x 16											
	SIMPLE	146	2.0	146	2.3	186	2.9	156	2.4	173	2.6	219	3.3
A	PISO	71	1.6	137	2.2	101	1.7	141	2.6	166	2.7	169	2.8
L	LIMPO	44	1.1	73	1.7	60	1.5	65	1.5	76	1.8	111	2.5
G		RUN 2 - 32 x 32											
O	SIMPLE	583	33.1	417	28.8	1285	88.1	537	37.2	498	39.2	587	40.5
R	PISO	229	17.6	315	24.8	295	22.6	339	26.0	366	28.0	462	34.2
I	LIMPO	114	11.3	161	17.2	171	19.8	130	13.4	137	14.2	157	18.3
T		RUN 3 - 64 x 64											
H	SIMPLE	1438	375	1650	620	4293	1472	2168	724	1862	664	1655	605
M	PISO	845	303	710	302	1728	782	839	330	943	399	996	401
	LIMPO	377	174	255	241	744	626	322	323	447	243	504	250

Figure 7.
Comparison of the performance of the three numerical algorithms, test case 1

d of the LIMPO algorithm (turbulent parameters and tangential velocity predictor and corrector steps).

Tube with sudden enlargement preceded by a quarl

Figure 8 shows the second geometry used to test the LIMPO algorithm, consisting of two axial inlets followed by a conical quarl and a sudden enlargement. The total mass flow of air is 8.21×10^{-3} kgs^{-1}, at 298K, the primary air flow being 6.8×10^{-4} kgs^{-1}. These flows correspond to inlet air velocities of 117.1 ms^{-1} and 26.7 ms^{-1} respectively for primary and secondary flows. The dimensions are $R_0 = 1.5 \times 10^{-1}$ m, $R_i/R_0 = 0.1$, $h/R_0 = 0.2$, $H/R_0 = 0.82$ and $L/R_0 = 10$. The predictions were performed for a grid comprising 60×50 nodes in the z and r directions respectively. The quarl is an important device in the flow structure. Therefore, although it represents only a small portion of the physical geometry, 20 per cent of the grid nodes were used to overlay the quarl domain. As above, the results were considered converged when the normalized residuals of the equations were less than 5×10^{-4}. The inlet swirl number S for the secondary air, defined by equation (18), was made equal to zero and unity. As in the previous test case for this geometry, the predicted values coincide for the three algorithms used.

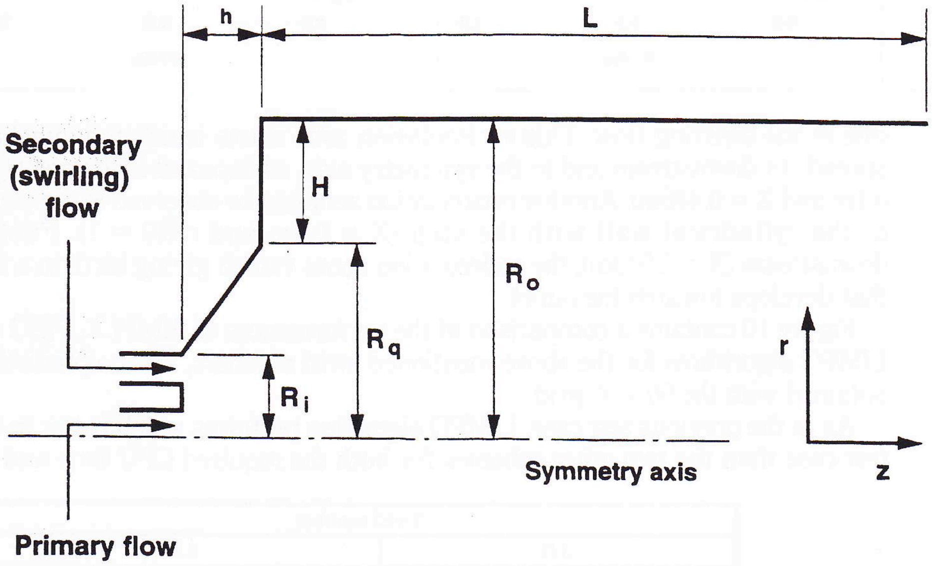

Figure 8.
Sketch of the tube with a step preceded by a conical quarl (test case 2)

Figure 9 shows the non-dimensional axial velocity profiles, for an inlet swirl number of the secondary air equal to unity, at several axial stations. Within the quarl region (X = 0.054m) the flow is mainly constituted by two emerging jets, the inner jet having a greater velocity. The negative values of the U velocity denote the existence of a recirculation zone, promoted by the pressure gradients

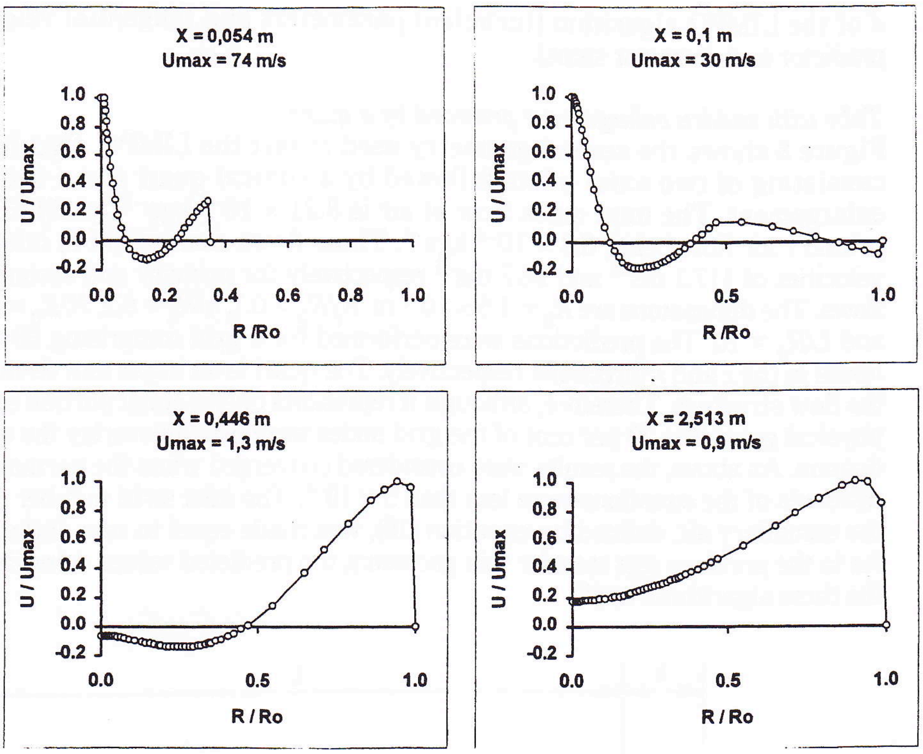

Figure 9.
Non-dimensional axial velocity profiles (test case 2 – swirl number = 1)

due to the swirling flow. This recirculation zone starts inside the quarl and spreads to downstream and to the symmetry axis, as depicted in Figure 9 (X = 0.1m and X = 0.446m). Another recirculation zone can be observed in the corner of the cylindrical wall with the step (X = 0.1m and r/R0 = 1). Further downstream (X = 2.513m), the recirculation zones vanish giving birth to a flow that develops towards the outlet.

Figure 10 contains a comparison of the performances of SIMPLE, PISO and LIMPO algorithms for the above mentioned swirl numbers, for the predictions obtained with the 60 × 50 grid.

As in the previous test case, LIMPO algorithm performs much better in this test case than the two other schemes for both the required CPU time and the

	Swirl number														
	0.0							1.0							
	under-relaxation					No	CPU	under-relaxation						No	CPU
Algorithm	u	v	p	k	ε	iter.	(sec)	u	v	w	p	k	ε	iter.	(sec)
SIMPLE	0.5	0.5	1.0	0.6	0.6	5261	625	0.5	0.5	0.5	1.0	0.7	0.7	4513	678
PISO	0.7	0.7	1.0	0.8	0.8	3959	602	0.7	0.7	0.8	1.0	0.8	0.8	2919	500
LIMPO	0.7	0.7	1.0	0.8	0.8	2550	525	0.7	0.7	0.8	1.0	0.8	0.8	1333	412

Figure 10.
Comparison of the behaviour of the three numerical algorithms, test case 2

number of iterations to achieve convergence. The same numerical results were obtained with LIMPO with, at least, a reduction of 36 per cent in the number of iterations and of 13 per cent in the required CPU time.

Conclusions

The new algorithm presented herein, LIMPO, developed for the coupling between pressure and velocities, allows the computation of two-dimensional turbulent swirling flows with considerable savings in computing effort, when compared with well-established algorithms – SIMPLE and PISO. Indeed, the number of iterations and the required CPU time to achieve the converged solution can be reduced by up to 50 per cent of the figures obtained with SIMPLE and PISO algorithms.

The geometries chosen to evaluate the potential of the LIMPO algorithm were two tubes, the first having a single sudden enlargement and the latter a step preceded by a conical quarl. With the first geometry, LIMPO proved to perform better than the other two algorithms, independently of the number of grid nodes used and of the inlet swirl number. With the latter geometry, the new algorithm exhibited a much better performance, as far as computing effort is concerned, demonstrating its potential for turbulent swirling flows, even within complex geometries.

References

1. Patankar, S.V. and Spalding, D.B., "A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows", *International Journal of Heat and Mass Transfer*, Vol. 15, 1972.
2. Patankar, S.V., *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, NY, 1980.
3. Spalding, D.B., "Internal Report HTS/80/1", Mechanical Engineering Department, Imperial College of Science Technology and Medicine, London, 1980.
4. Markatos, N.C. and Pericleous, K.A., *International Journal of Heat and Mass Transfer*, Vol. 27, 1984, p. 755.
5. Caretto, L.S., Gosman, A.D., Patankar, S.V. and Spalding, D.B., "Two calculation procedures for steady, three-dimensional flows with recirculation", *Proceedings of the 3rd International Conference on Numerical Methods in Fluid Dynamics*, Vol. II, Paris, 1972, p. 60.
6. Issa, R.I., "Solution of the implicitly discretized fluid flow equations by operator splitting", *Journal of Computational Physics*, Vol. 62, 1986, pp. 40-65.
7. Issa, R.I., Gosman, A.D. and Watkins, A.P., "The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme", *Journal of Computational Physics*, Vol. 62, 1986, pp. 66-82.
8. Issa, R.I. and Lockwood F.C., "On the predictions of two-dimensional supersonic viscous interactions near walls", *AIAA Journal*, Vol. 8, 1977, p. 182.
9. Carvalho, M.G., Oliveira, P. and Semião, V., "A three dimensional modelling of an industrial glass furnace", *Journal of the Institute of Energy*, Vol. LXI – 448, 1988, pp. 143-56.
10. Launder, B.E. and Spalding, D.B., *Mathematical Models of Turbulence*, Academic Press, New York, NY, 1972.
11. Gupta, A.K., Lilley, D.G. and Syred, N., *Swirl Flows, Energy and Engineering Science Series*, Abacus Press, 1984.

In future issues

Botz, Hagedorn, "Dynamic simulation of multibody systems including planar elastic beams using AUTOLEV".

Fafard, Mallikarjuna, Savard, "A general multi-axle vehicle model to study the bridge-vehicle interaction".

Nikolic, Mihanovic, "Non-linear finite element analysis of posttensioned concrete structures".

Lee, "Dynamic stability of a rotating cantilever beam with in-plane base acceleration".

Level, Cherki, Tison, "On evaluating structures sensitivities to prescribed displacement uncertanties using fuzzy numbers".

Hernandez, Albizuri, Ajuria, Hormaza, "An adaptive meshing automatic scheme based on the strain energy density function".

Deb, Paul, Thakkar, "Simplified non-linear dynamic analysis of base isolated buildings subject to general plane motion".

Salajegheh, "Optimum design of plate structures with frequency constraints".

Rojc, "On a mixed approach to the finite element solution of large strain elasto-plastic problems".

Stokarski, Ling, "On elasto-plastic finite element analysis of some frictional contact problems with large sliding".

Siebrits, Peirce, "Implementation and application of elastodynamic boundary element discretizations with improved stability properties".

Lin, Li, Zhang, Williams, "Structural seismic response to inhomoeneous random field".

Bazezew, Bruch, Sloss, "Optimal (distributed or boundary) control of the vibrations of continuous systems solved numerically in a space-time domain".

Bonet, "Recent developments in the incremental flow formulation for the numerical simulation of metal forming processes".

Stein, Yng, Kreienmeyer, "Coupling of BEM and FEM by a multiplicative Schwarz method and its parallel implementation".

Shaw, Graysmith, "Automated procedures for Boolean operations on finite element meshes".

Mueller, Burns, "An alternative linearization technique".

Kratzig, Koenke, Harte, Rosenstein, "On adaptive remeshing techniques for crack simulation problems".

Glaser, Armero, "On the formulation of enhanced strain finite elements in finite deformations".