11th World Congress on Computational Mechanics (WCCM XI)
5th European Conference on Computational Mechanics (ECCM V)
6th European Conference on Computational Fluid Dynamics (ECFD VI)
E. Oñate, J. Oliver and A. Huerta (Eds)

A GENERALISED CONVERGENCE ACCELERATED PRESSURE–BASED SEGREGATED ALGORITHM

V. PRŽULJ

Ricardo Software, Ricardo UK Ltd, Shoreham–by–Sea, West Sussex, BN43 5FG, UK
e-mail: vlado.przulj@ricardo.com, web page: www.ricardo.com/en-GB/What-we-do/Software

Key words: CFD, Pressure–Based Method, SIMPLE Algorithm, PISO, Finite–Volume, Co–located, Unstructured Grid.

Abstract. This paper reports on the generalised pressure–based sequential algorithm applicable to polyhedral grids and incompressible/compressible flows. The algorithm is formulated in the context of a cell–centred finite–volume method. The central theme of the paper is the convergence acceleration of the algorithm. This is achieved by a straightforward treatment of cell neighbours velocity corrections as well as the pressure corrections due to grid non–orthogonality. These corrections appear in the pressure correction equation of the well–known SIMPLE method but they are commonly neglected. Here, and akin to the PISO method, the above corrections are treated explicitly by designing two or more correction steps. The full inclusion of the cell neighbour corrections requires an under–relaxation typically equal to the pressure under–relaxation. The algorithm is applied to a number of benchmark problems, covering laminar and turbulent as well as incompressible and compressible flows. The significantly improved convergence rate and satisfactory accuracy of the algorithm are demonstrated. An appreciable reduction of computing time is possible when using optimal under–relaxation factors and an optimal number of pressure corrections.

1 INTRODUCTION

The SIMPLE-like (Semi–Implicit Method for the Pressure Linked Equations) algorithms have been a core of the pressure–based finite-volume methods. Since its formulation on orthogonal staggered grid arrangement in 1972, the algorithm has evolved to satisfy demands of unstructured grids with co–located storage of flow variables as well as to cope with all speed flows. In addition, the applicability of the SIMPLE–base methods have been extended to complex flow phenomena which characterise multicomponent and multiphase non-reacting or reacting fluid flows. A detailed description of these methods is available in a monograph by [1] as well as in [2].

The SIMPLE algorithm is classified as the segregated one as all flow equations are solved sequentially during an iterative process. An *outer* or SIMPLE iteration involves a *predictor* and a *corrector* step. In the predictor step, the discretized momentum equations for the velocity vector are solved implicitly using a linear equation solver. During the corrector step, the pressure correction equation, derived from the discretized continuity equation, is solved using linear equation solver. Then both velocity and pressure are corrected using the pressure correction field. It is the treatment of the pressure correction equation

which determines the SIMPLE performance. Namely, the velocity corrections from neighbouring cells (neighbour corrections) and the contribution of the cell-face pressure correction gradient associated with the grid non-orthogonality (non-orthogonal corrections), are neglected in the SIMPLE. The neglect of these two corrections leads to the great simplicity of the algorithm. However, their neglect is also the reason why the algorithm does not have a rapid convergence, see discussion in [1, p. 173]. The SIMPLEC algorithm [3] can partially compensate for the neglect of neighbour corrections by approximating them. The PISO method [4] provides two or more pressure correction steps so that neighbour corrections are accounted for in the second and further steps.

Using deferred–correction approach, additional correction steps are introduced by [1, p. 247] and [5] in order to take into account grid non-orthogonality corrections. In this work, the pressure correction steps are devised in order to account for the velocity corrections from neighbouring cells. Accounting for both neighbour and non–orthogonal corrections leads towards the fully implicit algorithm characterised by significantly improved velocity–pressure coupling, and consequently by accelerated convergence.

A finite–volume cell–centred method, with co-located storage is briefly described in the next section. This method is a kernel of Ricardo's new generation CFD solver VECTIS-MAX , and some aspects of the method have been presented in [5, 6]. After that, a generalised and accelerated formulation of the original SIMPLE algorithm is presented, followed by the discussion of results as obtained for a number of benchmark test cases.

2 OVERVIEW OF FINITE–VOLUME DISCRETISATION

Considering a non–moving control volume V, shown in Figure 1 (left), the RANS equations can be cast in a semi–discrete form, [5, 7], as:

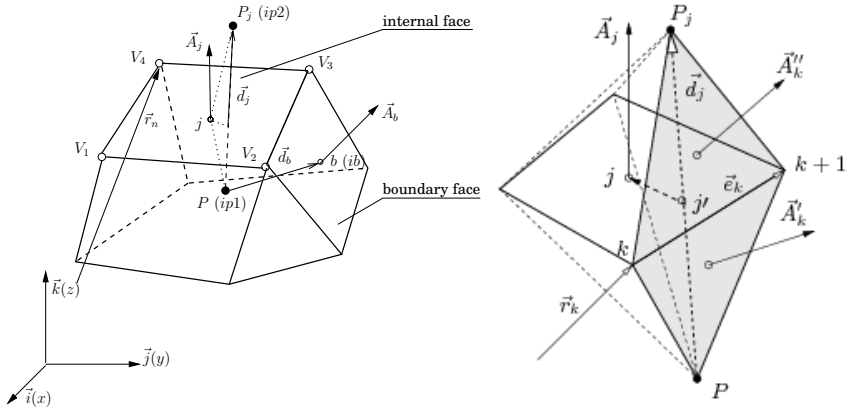

Figure 1: Polyhedral control volumes and nomenclature: main cell–based (left) and around cell–face j

$$\frac{d}{dt}\left(\rho V\phi\right)_P + \sum_{j=1}^{n_f}\left(C_j - D_j - \left(s^A_{\phi k}A_k\right)_j\right) = \left(s^V_\phi\right)_P V_P \,, C_j = \dot{m}_j\phi_j \,, D_j = \left(\Gamma_\phi \nabla\phi\cdot\vec{A}\right)_j \qquad (1)$$

In the above equation, ϕ is a flow variable and Γ_ϕ represents its diffusion coefficient; time and density are denoted as t and ρ, respectively. For a cell–face j, outward face area vector is denoted as $\vec{A}_j$, while convective and diffusion fluxes are C_j and D_j, respectively. The source terms s^V_ϕ and s^A_ϕ are related to

the cell volume and cell face, respectively; n_f is the number of cell-faces enclosing the cell. The mass fluxes $\dot{m}_j$ must satisfy the integral mass balance over each cell:

$$\frac{d}{dt}\left(\rho V\right)_P + \sum_{j=1}^{n_f} \dot{m}_j = 0 \ , \ \dot{m}_j = \rho_j \vec{U}_j \cdot \vec{A}_j \tag{2}$$

where $\vec{U}_j$ is the mass conserving face velocity obtained by the Rhie–Chow interpolation practice [1, 8] which avoids spurious checker–boarded pressure fields.

Unsteady terms are integrated over each time interval Δt by either the first order accurate Euler or second order accurate three time level *implicit* scheme, see [1]. Both schemes are implicit and unconditionally stable. Details regarding the evaluation of variable face values and cell gradients as well as the discretisation of diffusion fluxes can be found in [5, 6]. The convective fluxes are linearised by using the mass fluxes from the previous iteration. Apart from upwind and central differencing schemes, bounded MINMOD and AVL SMART schemes designed for unstructured grids can be used, see [9]. With the help of Figure 1 (right), derivation outlined in [5] leads to the following expression for the cell–face gradient:

$$\nabla\phi_j \approx \overline{\nabla\phi}_j + \frac{\vec{A}_j}{\vec{A}_j \cdot \vec{d}_j}\left[\left(\phi_{P_j} - \phi_P\right) - \overline{\nabla\phi}_j \cdot \vec{d}_j\right] \ , \ \overline{\nabla\phi}_j = f_j \nabla\phi_P + (1 - f_j)\nabla\phi_{P_j} \tag{3}$$

where $f_j = |\vec{r}_{P_j} - \vec{r}_j|/(|\vec{r}_{P_j} - \vec{r}_j| + |\vec{r}_j - \vec{r}_P|)$ is the face interpolation factor, and $\vec{r}$ denotes the position vector and $\vec{d}_j$ is the distance vector between cells P and P_j.

The outcome of the discretisation of considered transport equation for ϕ is a linearised algebraic equation for each cell. In the case of momentum, the velocity vector equation at the new *outer* iteration (l) and within the new time step (n) reads:

$$a_P \vec{U}_P = \sum_{j=1}^{n_f} a_j \vec{U}_{P_j} + \vec{S}_U - V_P \nabla p_P + (\vec{S}_{U,t})^{(n-1)} + (\vec{S}_{U,t})^{(n-2)} \ , \ a_P = \sum_{j=1}^{n_f} a_j + a_{P,t} \ , \tag{4}$$

The central coefficient a_P comprises contributions from neighbouring cell coefficients a_j and from the unsteady coefficient $a_{P,t}$ (for example, $a_{P,t} = (\rho V)_P^{(n-1)}/\Delta t$ for the Euler time–discretisation scheme). In a segregated solution method, the neighbouring coefficients, the central coefficient and the pressure term $(-V_P \nabla p_P)$ are evaluated using variable values from the previous iteration $(l-1)$. Momentum sources $(\vec{S}_U)$ do not include the pressure term – they are also calculated using variable values from the previous iteration. Similarly, unsteady source terms $(\vec{S}_{U,t})^{(n-1)}$ and $(\vec{S}_{U,t})^{(n-2)}$ involve values from the previous time steps. The initial guesses for the velocity and pressure at the new time step/iteration are taken to be those from the previous time step/iteration.

The system of linearised algebraic equations for each flow variable is under–relaxed implicitly. The system is then arranged in a sparse matrix form $[\mathbf{A}]\phi = \mathbf{b}_\phi$, and solved by Bi–CGSTAB linear equation solver using incomplete lower–upper (ILU) preconditioners. The non–linearity and coupling of the solved transport equations is handled by the sequential SIMPLE–based algorithm. Normalised residuals, defined as $R_\phi = \sum_{cells} |a_P\phi_P - b_\phi| / \sum_{cells} |a_P\phi_P|$, provides a convergence criterion for the solution of coupled equations. For the mass conservation equation, the normalised residual is defined as $R_m = \sum_{cells} |\text{mass imbalance}| / \sum_{cells} |\text{mass imbalance}|_{ref}$, where the reference mass imbalance is the maximum imbalance in the first five iterations, [10]. The solution is considered converged when all normalised residuals drop below a certain value, typically below $1. \times 10^{-5}$.

3 GENERALISED FORMULATION OF THE SIMPLE ALGORITHM

The SIMPLE algorithm ensures that the calculated flow field, once converged, simultaneously satisfies both the continuity and momentum equations. The first *predictor* step of the algorithm is the implicit solution of the discretized momentum Equation (4) based on the pressure field from the previous outer iteration. In what follows, a general formulation of the SIMPLE *corrector* step is described in terms of the key ingredients, namely the definition of mass conserving cell–face velocity, and derivation of the pressure correction equation from the discretized continuity equation. The formulation draws on previous work of [1, 7, 10], and it has been presented for the steady-state flows in [5].

Cell–face velocity for mass conservation. On co-located grids, the use of linear interpolation to obtain the pressure and velocities at the cell faces usually leads to de–coupling of two fields. In order to overcome this problem a special interpolation practice [8], [1] is required for the mass conserving face velocity in Equation (2). Re–calling the discretized momentum Equation (4), the velocity vector corresponding to the current outer iteration l can be explicitly expressed as

$$\vec{U}_P = \vec{h}_P - \frac{V_P}{a_P}\nabla p_P + \frac{1}{a_P}\left((\vec{S}_{U,t})^{(n-1)} + (\vec{S}_{U,t})^{(n-2)}\right),\ \vec{h}_P = \frac{\sum_j a_j \vec{U}_{P_j} + \vec{S}_U}{a_P} \tag{5}$$

Following derivation in [5], where the use of the cell-face gradient Equation (3) for the pressure is essential, a similar equation for the CV around the cell–face centre 'j' can be obtained as:

$$\vec{U}_j = \overline{\vec{U}}_j - \overline{\left(\frac{V}{a_P}\right)}_j \frac{\vec{A}_j}{\vec{A}_j \cdot \vec{d}_j}\left[p_{P_j} - p_P - \overline{\nabla p}_j \cdot \vec{d}_j\right] + \frac{\rho_j}{\Delta t}\overline{\left(\frac{V}{a_P}\right)}_j \left(\vec{U}_j^{(n-1)} - \overline{\vec{U}}_j^{(n-1)}\right) \tag{6}$$

$$\overline{\vec{U}}_j = f_j \vec{U}_P + (1-f_j)\vec{U}_{P_j}\ ,\ \overline{\nabla p}_j = \frac{1}{2}(\nabla p_P + \nabla_{P_j})\ ,\ \overline{\left(\frac{V}{a_P}\right)}_j = \frac{1}{2}\left(\frac{V_P}{a_P} + \frac{V_{P_j}}{a_{P_j}}\right) \tag{7}$$

Notably, the above cell–face velocity formulation avoids the under-relaxation factor dependence by using non under–relaxed momentum discretisation, Equation (4), as the starting point so that $(a_P)_j$ coefficients are not affected by the under-relaxation velocity factor α_U. Even without this approach, the effect of under–relaxation on the given numerical grid is much smaller than the discretisation error, see [1, p.194]. Both central and therefore the cell-face coefficients are formulated to be the same for all three velocity components. To eliminate even though small time step size dependence, the last, transient term in Equation (6) is formulated in terms of the unsteady coefficient $\rho V/\Delta t$ which is equivalent to the Euler scheme coefficient on the non-moving grid. The above transient term is similar to the one used by [2, p. 360].

Pressure correction equation. Following the predictor step, the calculated velocities, denoted now as $\vec{U}_P^*$ and $\vec{U}_j^*$, are based on the pressure and density fields from the previous iteration $p_P^* = p_P^{(l-1)}$, $\rho_P^* = \rho_P^{(l-1)}$, respectively. The mass fluxes $\dot{m}_j^*$, computed by using the cell–face velocity from Equation (6), do not generally satisfy the continuity equation (2). A mass imbalance would thus result. The basis of the SIMPLE algorithm is to drive this mass imbalance to a negligible small value. In the first corrector step, this is achieved by introducing the corrections: $\vec{U}_P = \vec{U}_P^* + \vec{U}_P'$, $p_P = p_P^* + p_P'$ and $\rho_P = \rho_P^* + \rho_P'$. Substituting the corrected variables into discrete momentum Equations (5) and (6), and recognising that

terms with the 'star' $(...)^*$ variables are in balance, one can link the velocity and pressure corrections at cells and cell–faces as:

$$\vec{U}'_P = \vec{h}'_P - \frac{V_P}{a_P}\nabla p'_P \ , \ \vec{h}'_P = \frac{1}{a_P}\left(\sum_j a_j \vec{U}'_{P_j} + \vec{S}'_U\right) \ , \ \vec{S}'_U \approx 0 \ , \ \vec{U}'_j = \overline{\vec{h}'_j} - \overline{\left(\frac{V}{a_P}\right)}_j \nabla p'_j \tag{8}$$

$$\vec{U}'_{P_j} \approx -\frac{V_{P_j}}{a_{P_j}}\nabla p'_{P_j} \Rightarrow \overline{\vec{h}'_j} = -\frac{1}{2}\left[\frac{1}{a_P}\left(\sum_j a_j \frac{V_{P_j}}{a_{P_j}}\nabla p'_{P_j}\right)_P + \frac{1}{a_{P_j}}\left(\sum_j a_j \frac{V_{P_j}}{a_{P_j}}\nabla p'_{P_j}\right)_{P_j}\right] \tag{9}$$

$$\vec{U}'_j = -\overline{\left(\frac{V_P}{a_P}\right)}_j \frac{\vec{A}_j}{\vec{A}_j \cdot \vec{d}_j}\left(p'_{P_j} - p'_P\right) + \vec{H}'_j \ , \ \vec{H}'_j = \overline{\vec{h}'_j} - \overline{\left(\frac{V_P}{a_P}\right)}_j \left(\overline{\nabla p'}_j - \frac{\vec{A}_j}{\vec{A}_j \cdot \vec{d}_j}\overline{\nabla p'}_j \cdot \vec{d}_j\right) \tag{10}$$

In the above equations, the velocity corrections from neighbouring cells, represented by $\vec{h}'_P$ and $\overline{\vec{h}'_j}$, are not known during the first correction step. Also, the last term in Equation (10) which describes the contribution of the cell-face pressure correction gradient on non-orthogonal grids is not known.

Introduction of mass fluxes $\dot{m}_j = \dot{m}^*_j + \dot{m}'_j$ into the mass conservation equation (2) effectively converts this equation into an equation for the pressure correction:

$$\frac{d}{dt}\left(\rho'_P V_P\right) + \sum_{j=1}^{n_f} \dot{m}'_j = -S^*_m \Rightarrow a'_P p'_P = \sum_{j=1}^{n_f} a'_j p'_{P_j} - S^*_m - S'_m \tag{11}$$

where a'_P and a'_j are the central and neighbour coefficients of the pressure correction equation. Mass source S^*_m, and mass flux correction $\dot{m}'_j$ read

$$S^*_m = \frac{d}{dt}\left(\rho^*_P V_P\right) + \sum_j \dot{m}^* \ , \dot{m}^*_j = \rho^*_j \vec{U}^*_j \cdot \vec{A}_j \ , \ \dot{m}'_j \approx \rho^*_j \vec{U}'_j \cdot \vec{A}_j + \rho'_j \vec{U}^*_j \cdot \vec{A}_j \tag{12}$$

Next, the density corrections are approximated by:

$$\rho' \approx \left(\frac{\partial \rho}{\partial p}\right)_T p' = C_\rho p' \ , \ \rho'_j = \left[C_{\rho P} \max\left(\dot{m}^*_j, 0\right) p'_P - C_{\rho P_j} \max\left(-\dot{m}^*_j, 0\right) p'_{P_j}\right] \frac{\beta_p}{\dot{m}^*_j} \tag{13}$$

where the coefficient C_ρ can be found from the equation of state (for an ideal gas $C_\rho = 1/(R_g T)$). The cell–face density correction ρ'_j is approximated by a fraction of the upwind scheme [5] where $\beta_p = \alpha_p$ (α_p being the under–relaxation factor for pressure) is recommended. Finally, the last term in Equation (11) S'_m accounts for the neighbour velocity corrections and for the grid non-orthogonal pressure corrections:

$$S'_m \approx \sum_j \rho^*_j \vec{H}'_j \cdot \vec{A}_j \tag{14}$$

where $\vec{H}'_j$ is given by Equation (10). As mentioned earlier, the velocity corrections from neighbouring cells as well as pressure corrections due to grid non-orthogonality are not known during the first corrector step. Thus the first correction proceeds with $\vec{h}'_P = 0$ and $\vec{H}'_j = S'_m = 0$, i.e. in accordance with the SIMPLE corrector step.

The pressure correction equation (11) has a form similar to the general discretized equation (4). Its distinctive features are discussed in [7], [1]. After implicit solution of the algebraic equations for p' (before that p' is initialised zero everywhere), the cell velocities, pressure and mass fluxes are corrected as discussed above. The pressure is corrected only by a fraction of p': $p_P = p_P^* + \alpha_p p'_P$, where α_p is typically 0.1 to 0.3.

Accounting for velocity corrections from neighbouring cells. A general approach which accounts for neighbours velocity corrections on orthogonal (either staggered or co–located) grid was described in [1, p. 176], and it has been related to the PISO algorithm [4]. Accordingly, two or more corrector steps should be carried out in order to solve iteratively Equation (8) for $\vec{U}'_P$ by treating neighbours velocity corrections $\vec{h}'_P$ in an explicit manner.

Thus another corrector step can be designed by assuming that the second corrected velocities $\vec{U}_P = \vec{U}_P^* + \vec{U}'_P + \vec{U}''_P$, pressure $p_P = p_p^* + p'_P + p''_P$, and mass fluxes $\dot{m}_j = \dot{m}_j^* + \dot{m}'_j + \dot{m}''_j$ satisfy the discrete momentum Equations (5) and (6), and the mass conservation Equation (2), respectively. The new corrections for the cell and face velocities are then derived as:

$$\vec{U}''_P = \vec{h}'_P - \frac{V_P}{a_P}\nabla p''_P + \vec{h}''_P \;,\; \vec{U}''_j = -\overline{\left(\frac{V_P}{a_P}\right)}_j \frac{\vec{A}_j}{\vec{A}_j \cdot \vec{d}_j}\left(p''_{P_j} - p''_P\right) + \vec{H}'_j + \vec{H}''_j \tag{15}$$

The new mass flux correction becomes

$$\dot{m}''_j = \rho_j^* \vec{U}''_j \cdot \vec{A}_j + \rho''_j \vec{U}_j^* \cdot \vec{A}_j \tag{16}$$

where the second density correction ρ''_j is given by Equation (13) with p' replaced by p''. The mass balance, Equation (2), is already enforced by the corrected mass fluxes $(\dot{m}_j^* + \dot{m}'_j)$. Therefore, the second pressure correction equation can be derived from the following balance equation:

$$\frac{d}{dt}\left(C_{\rho P}\, p''_P\, V_P\right) + \sum_{j=1}^{n_f} \dot{m}''_j = 0 \Rightarrow a'_P p''_P = \sum_{j=1}^{n_f} a'_j p''_{P_j} - S'_m - S''_m \;,\; S''_m \approx \sum_j \rho_j^* \vec{H}''_j \cdot \vec{A}_j \tag{17}$$

where $\vec{H}''_j$ has the same form as $\vec{H}'_j$, i.e. super-scripts $(...)'$ in Equation (10) are replaced with $(...)''$.

In the second corrector step, the cell neighbours velocity corrections $\vec{h}'_P$, cell–face corrections $\vec{H}'_j$ as well as the mass source S'_m can be calculated using pressure corrections from the previous step. The corresponding unknown $\vec{h}''_P$, $\vec{H}''_j$ and S''_m corrections are ignored in this step. After solving the second pressure correction Equation (17) additional corrector steps can be constructed in the same way as the second step. Note that the second and further pressure correction equations have the same coefficient matrix as Equation (11) for p'.

In the present formulation, both cell neighbours and non-orthogonal grid corrections (represented by $\vec{H}'_j$, $\vec{H}''_j$ terms) are fully accounted by solving additional pressure correction equations. As demonstrated in [1, 5], the inclusion of non-orthogonal grid corrections improves convergence rate for highly non–orthogonal grids, i.e. for the grids with angles between $\vec{A}_j$ and $\vec{d}_j$ greater than $45°$.

Initial numerical tests showed that full inclusion of neighbour corrections can cause divergence. Thus both neighbour and non-orthogonal corrections are under-relaxed as $\alpha_{corr}\vec{H}'_j$ where α_{corr} is the new under–relaxation factor. An optimal choice, ensuring numerical robustness, is to define this factor equal to the pressure under–relaxation, $\alpha_{corr} = \alpha_p$.

4 RESULTS AND DISCUSSION

The pressure–correction method described above is applied to several benchmark test cases, covering two–dimensional (2D) and three–dimensional (3D) incompressible and compressible laminar/turbulent flows. The aim is to demonstrate the accuracy of the method and establish its convergence characteristics versus computational time. All results have been obtained using parallel runs on the cluster of Linux machines with Intel Nehalem-based Xeon processors (3.2Ghz, each 4 cores). Numerical parameters used in the present simulation are given in Table 1. For turbulent cases, the time scale bounded $k-\varepsilon$ (TSB) model [11] is employed. The convergence criterion is set to either $1.\times 10^{-7}$ or $1.\times 10^{-6}$ for all variables.

Table 1: Numerical parameters used for present simulations.

Test case	Dim.	Mach	Turb.	Cells	Cores	α_U	α_p	α_ϕ	Δt, s	Convection
Cavity 45°, $Re=100$	2D	0	Lam.	160×160	4	0.90	0.05	–	–	MINMOD
Cubic cavity Re=1000	3D	0	Lam.	510623	6	0.95	0.05	–	–	AVL-SMART
Cubic cavity Re=1000	3D	0	Lam.	2299968	16	0.95	0.05	–	–	AVL-SMART
Cubic cavity Re=3200	3D	0	Lam.	510623	6	0.95	0.5	–	0.05	AVL-SMART
Rotating tube, 103 rad/s	3D	0	Lam.	158111	4	0.9	0.3	–	–	MINMOD
Delery's-C bump	2D	0.39	TSB	19000	4	0.8	0.2	0.90	–	MINMOD
Cylinder-flare	3D	3.0	TSB	576496	6	0.8	0.2	0.90	10^{-4}	MINMOD

2D and 3D lid-driven cavity flows. The two–dimensional lid–driven laminar, incompressible flow in skewed cavities, with side walls inclined at 45° and 30°, was benchmarked by [12]. Here, the case with 45°, at the Reynolds number $Re=100$ is considered. The effect of pressure correction steps on the convergence rate is shown in Figure 2 (left) in terms of normalised residuals plotted against SIMPLE outer iterations. Evidently, a number of required iterations to achieve the converged solution is always

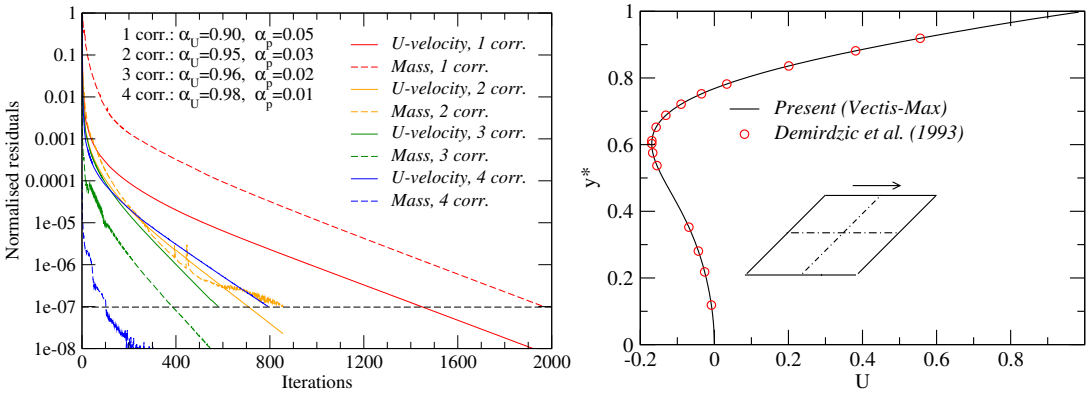

Figure 2: 2D 45° cavity at $Re=100$. Convergence rate (left) and x–velocity (U) profile along the centreline connecting the top and bottom walls (right).

significantly reduced when performing two or more correction steps. Here, slightly different values of under-relaxation factors for the velocity and pressure (quite low for the pressure) have been employed for each correction step. The x–velocity profile along the cavity centreline is plotted in Figure 2 (right). The visual comparison with the benchmark results shows no differences – the actual differences with

respect to the tabulated benchmark results are less than 0.1%.

In the next case, laminar lid–driven flows in a cubic cavity, Figure 3 (left), are simulated at two Reynolds numbers $Re = 1000$ and $Re = 3200$. For the latter Re–number the flow is unsteady.

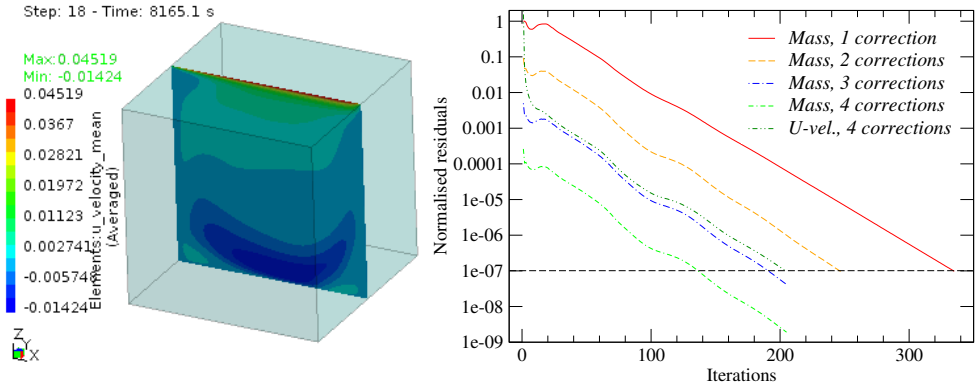

Figure 3: Cubic cavity. Geometry with the centre plane location where velocity profiles are considered (left) and residuals histories for the steady–state flow at $Re = 1000$ as computed on the grid with 510623 cells (right).

In the benchmark study of [13] for $Re = 1000$, the use of Cartesian grid with $64^3 = 262144$ cells introduced the maximum error of 2.8% for extrema on velocity profiles. In the present work, a medium size and a fine grid with 510623 and 2299968 hexahedral cells, respectively, are employed. The mass residuals shown in Figure 3 (right) indicate an improved convergence rate with introduction of additional pressure correction steps. For each run, under–relaxation factors are kept constant. As the result, the velocity residuals are not affected by a number of correction steps.

Present predictions at $Re = 1000$ are compared with the benchmark results of [13]. Figure 4 (left) demonstrates a very good agreement for the U–velocity profile along the centre line (z–axis) connecting the bottom and top moving wall. The fine grid results are practically indistinguishable from the medium grid results.

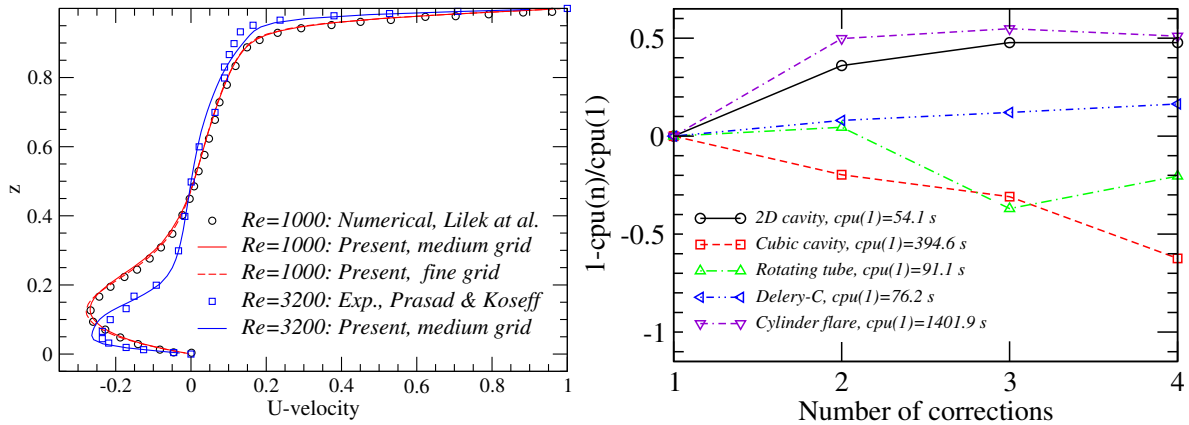

Figure 4: Comparison of cubic cavity U–velocity profiles with benchmark/experimental results (left) and computational time versus correction steps as calculated for all test cases (right).

In the same figure, a time-averaged U–velocity as obtained from the unsteady simulation at $Re = 3200$ can be compared with the measured data of [14]. Averaging is done over the last $4000\ s$ of the total simulation time of $8165\ s$ (in total 163300 time steps with $\Delta t = 0.05\ s$). Considering the size of the employed medium grid, the agreement between the predicted and measured averaged velocity profiles is satisfactory. Figure 3 (left) illustrates the time–averaged U–velocity field in the centre plane $y = 0.5m$. Note that unsteady simulation is done using 3 pressure corrections per outer iteration. Three outer iterations per time step are typically required to satisfy the convergence criterion of $1. \times 10^{-6}$ within each time step.

Flow in a rotating tube. The incompressible air flow through a tube rotating around axial axis with the constant angular velocity $\omega = 103rad/s$ is the next test case, also used by [15]. The tube diameter is $0.1m$, the length $0.5m$, and axial velocity of $5m/s$ is specified to get a Reynolds number of 50000. As the whole solution domain rotates, the governing equations are solved in the non–inertial rotating reference frame using absolute velocity formulation i.e. the velocity relative to the inertial reference frame. Note that a special treatment of body forces which results in modification of the mass conserving cell–face velocity, see for example [2, p. 362], is not used here.

As Figure 5 (left) shows, the effect of additional pressure corrections on the convergence rate is very beneficial. In the same figure (right), the computed relative pressure distribution along two radial

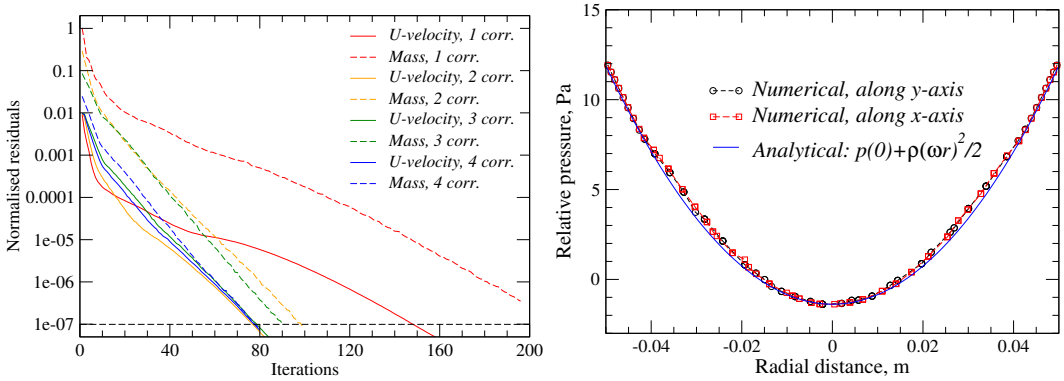

Figure 5: Rotating tube. Convergence rate (left) and radial pressure distributions in the mid–tube cross–section (right).

x and y–axes in the tube mid–plane cross–section can be compared with an approximate analytical solution [15]. The relative pressure at the tube walls is under–predicted by 1.5% with reference to the analytical solution.

Transonic channel bump. This nominally 2D flow over a channel bump is accompanied by the strong shock, Figure 6 (left), which provokes the flow separation. It has been used as a benchmark for turbulence modelling in shock–wave/boundary–layer interactions, see [5]. The channel geometry consists of a bump on the lower wall and of slightly sloping upper wall, [16]. At the inlet, subsonic stagnation conditions are known: $p_t = 96\ kPa$, $T_t = 300\ K$ which correspond to the Mach number $Ma = 0.39$. The turbulent kinetic energy and its dissipation are set to $1.5\ m^2/s^2$ and $1000\ m^2/s^3$, respectively. The uniform static pressure of $63.36\ kPa$ is prescribed at the outlet.

With reference to Figure 7 (left), and following initial steep reductions, the residuals achieve lo-

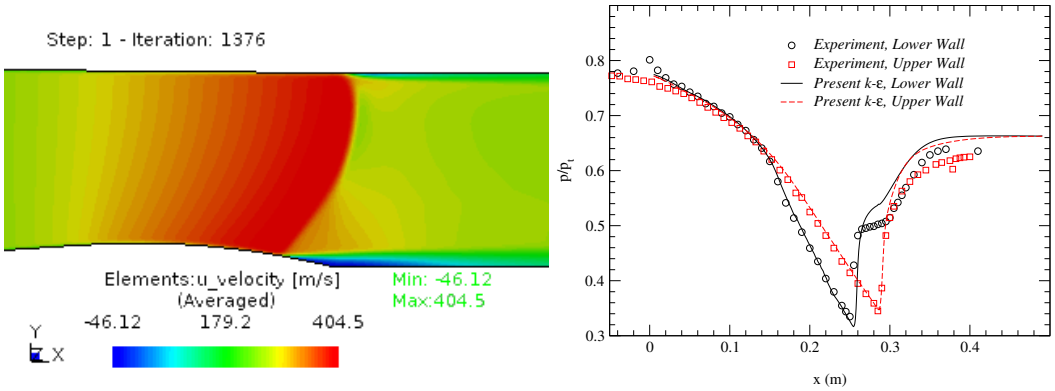

Figure 6: Transonic channel bump. Axial velocity distribution showing the λ–shock pattern (left) and comparison of the predicted wall pressure with measured data from [16] (right).

cal minimum and maximum values before having near–linear reduction trend. This residual behaviour

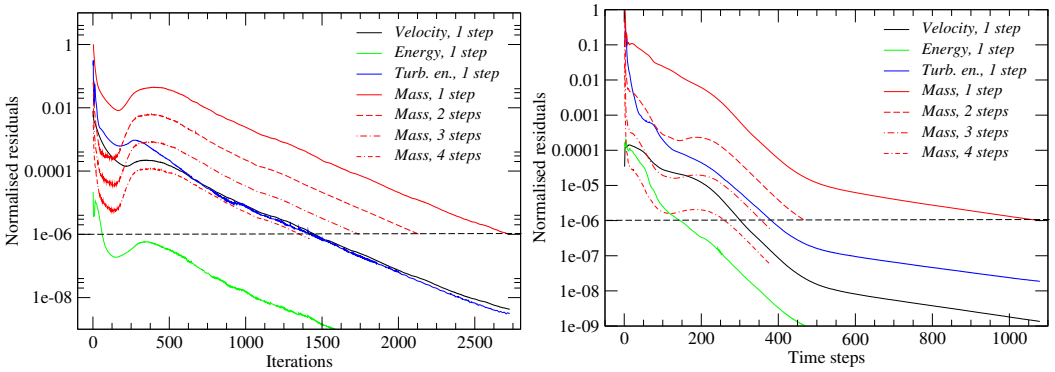

Figure 7: Residuals history for the transonic channel bump (left) and for the super–sonic cylinder–flare juncture (right).

probably corresponds to the time required by the solver to deal with the shock–wave development. Regardless of this behaviour, the usage of additional pressure corrections significantly reduces the number of required iterations. Turning back to Figure 6 (right), the computed wall pressure distributions indicate a correct prediction of the shock positions at the walls. However, the measured 'pressure plateau' over the lower wall is not reproduced by the current turbulence model. Also, the model over–predicts the wall pressure after the shock which is also deficiency of other two–equation turbulence models [5].

Super–sonic flow over cylinder–flare. The geometry of the final test case consists of a sting supported cylinder aligned with the free-stream, and a 20° flare, displaced by $1.27\ cm$ from the cylinder centreline, Figure 8 (left). The nominal Mach number 3 turbulent flow over this configuration was experimentally investigated by [17]. The axial velocity distribution in the above figure indicate the formation of the oblique shocks and the expansion fans due to abrupt changes in the geometry. On the bottom of the geometry and upstream of the cylinder–flare juncture, the flow separates, forming a small recirculation region.

The top and bottom of the cylinder will be identified by angles 0° and 180°, respectively. Since the

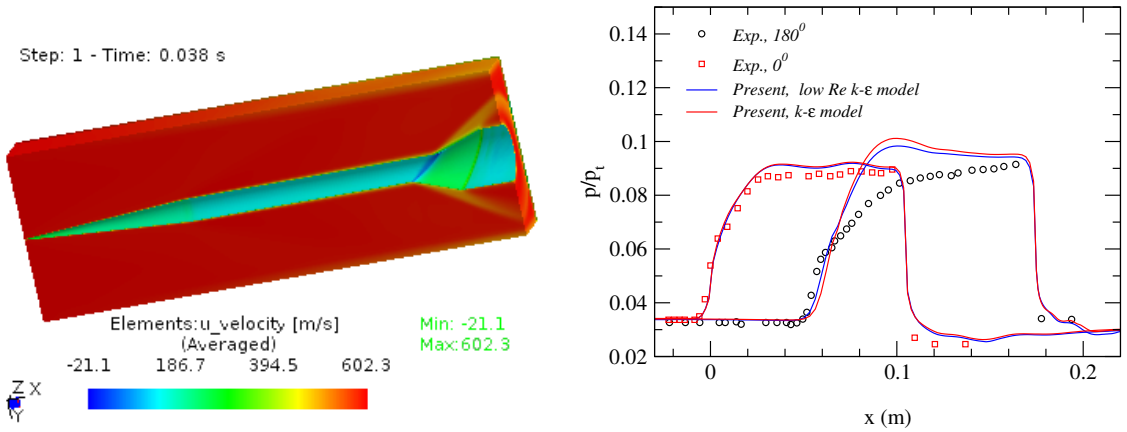

Figure 8: Cylinder–flare juncture. Axial velocity distribution indicating oblique shocks and expansion fans (left) and comparison of the pressure profiles along the top/bottom walls with measured data from [17] (right).

flow is steady the solution domain is taken to be half of the original geometry. The test–section operating conditions at the inlet: $p_t = 172370\ Pa$, $T_t = 280\ K$, $p_\infty = 5540\ Pa$, $\rho_\infty = 0.184,\ kg/m^3$, $U_\infty = 593\ m/s$ are used for the present simulations. For this case, the steady–state solution is obtained using unsteady procedure.

As Figure 7 (right) shows, the convergence behaviour is similar to the previous transonic test case. However, the differences between residuals extrema are not significant any more. Following Figure 8 (right), the calculated pressure distribution along the top wall is in satisfactory agreement with the measured one. However, the pressure profile along the bottom wall does not capture well the experimental distribution.

Correction steps versus computing time. Evidently, the convergence rate of all computed cases has been significantly improved by introduction of two or more pressure correction steps. However, the computing time required to solve additional pressure correction equations can negatively impact overall computational efficiency. To illustrate this, the computing efficiency norm defined as $1 - cpu(n)/cpu(1)$ ($cpu(1)$ is the solver CPU time for 1 correction step, while n is the number of correction steps) is plotted against the number of correction steps in Figure 4 (right). Considering all test cases, only the cubic cavity does not benefit from additional correction steps. For this case but without generalisation, the reason is the usage of a uniform hexahedral (orthogonal) grid with highly optimised under–relaxation factors. Thus in addition to optimal under–relaxation factors, an optimal number of correction steps can be established for each problem.

5 CLOSURE

Within the framework of an unstructured grid cell–centred finite–volume method, a general pressure–based sequential algorithm is formulated. The algorithm is applicable to any type of numerical grid with co-located variable arrangement as well as to all speed flows. In comparison to the SIMPLE–like algorithms, the present algorithm offers the convergence acceleration which is an outcome of the novel treatment of the pressure correction equation. A relatively straightforward inclusion of cell neighbours velocity corrections and pressure corrections due to grid non–orthogonality is provided by introduction

of two or more correction steps akin to the PISO method. The above corrections are typically neglected in the SIMPLE–like methods. The satisfactory convergence rate and accuracy of the algorithm have been established by computing several benchmark test cases. For all test cases, the convergence rate can be significantly improved by performing two or more pressure correction steps. An optimal number of pressure corrections exists for which a meaningful reduction of computing time is possible. This number varies between two and four – four pressure corrections are typically used for complex industrial applications where poor quality numerical grids are frequently employed. Notably, higher under–relaxation factors for the momentum and pressure can be used. If the sum of these factors is close to $(1.0 - 1.1)$ it is a good indicator for their optimal choice which is in agreement with the recommendation from [1].

References

[1] Ferziger, J. and Peric, M., *Computational Methods for Fluid Dynamics*. Springer, Berlin (1997).

[2] Moukalled, F. and Darwish, M., Pressure–based algorithms for single–fluid and multifluid flows. In W.J. Minkowycz, E.M. Sparrow and J.Y. Murthy (eds.), *Handbook of Numerical Heat Transfer*, chap. 10, John Wiley & Sons, Inc., Hoboken, New Jersey, second ed., pp. 326–367 (2006).

[3] van Doormal, J.P. and Raithby, G.D., Enhancement of the SIMPLE method for predicting incompressible fluid flow. *Numerical Heat Transfer* **7**:147–163 (1984).

[4] Issa, R.I., Solution of the implicitly discretized fluid flow equations by operator–splitting. *Journal of Computational Physics* **62**:40–65 (1986).

[5] Przulj, V. and Basara, B., A SIMPLE–based control volume method for compressible flows on arbitrary grids. *AIAA 2002 –3289* (2002).

[6] Przulj, V., Birkby, P. and Mason, P., Finite volume method for conjugate heat transfer in complex geometries using Cartesian cut–cell grids. In *CHT-08*, Marrakech, Morocco (2008).

[7] Demirdzic, I., Lilek, Z. and Peric, M., A collocated finite volume method for predicting flows at all speeds. *Int. Journal for Numerical Methods in Fluids* **16**:1029–1050 (1993).

[8] Rhie, C.M. and Chow, W.L., Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* **21**:1525–1532 (1983).

[9] Przulj, V. and Basara, B., Bounded convection schemes for unstructured grids. *AIAA 2001–2593* (2001).

[10] Mathur, S.R. and Murthy, J.Y., A pressure based method for unstructured meshes. *Numerical Heat Transfer B* **31**:195–215 (1997).

[11] Przulj, V., Tiney, N., Shapiro, E., Penning, R. and Shala, M., The time scale bounded $k - \varepsilon$ turbulence model and its assessment for automotive applications. In K. Hanjalic and Y. Nagano (eds.), *7th Int. Symp. Turbulence, Heat and Mass Transfer*, Palermo, Italy (2012).

[12] Demirdzic, I., Lilek, Z. and Peric, M., Fluid flow and heat transfer test problems for non–orthogonal grids: Bench–mark solutions. *Int. Journal for Numerical Methods in Fluids* **15**:329–354 (1992).

[13] Lilek, Z., Muzaferija, S. and Peric, M., Efficiency and accuracy aspects of a full–multigrid SIMPLE algorithm for three–dimensional flows. *Numerical Heat Transfer, Part B* **31**:23–42 (1997).

[14] Prasad, A.K. and Koseff, J.R., Reynolds number and end–wall effects on a lid–driven cavity flow. *Physics of Fluids A* **1**(2):208–218 (1989).

[15] Chen, Z.J. and Przekwas, A.J., A coupled pressure–based computational method for incompressible /compressible flows. *Journal of Computational Physics* **229**(24):9150–9165 (2010).

[16] Delery, J.M., Experimental investigation of turbulence properties in transonic shock–wave/boundary–layer interactions. *AIAA Journal* **21**. 180. (1983).

[17] Wideman, J.K., Brown, J.L., Miles, J.B. and Ozcan, O., Skin–friction measurements in three–dimensional, supersonic shock–wave/boundary–layer interaction. *AIAA Journal* **33**:805–811 (1995).