INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **44**:621–644 (DOI: 10.1002/fld.662)

Development of a fully coupled control-volume finite element method for the incompressible Navier–Stokes equations

Idriss Ammara[‡] and Christian Masson[*,†,§]

Department of Mechanical Engineering, École de Technologie Supérieure, Montréal, Canada

SUMMARY

This paper proposes and investigates fully coupled control-volume finite element method (CVFEM) for solving the two-dimensional incompressible Navier–Stokes equations. The proposed method borrows many of its features from the segregated CVFEM described by Baliga *et al*. Thus finite-volume discretization is employed on a colocated grid using either the MAW or the FLO schemes and an element-by-element assembling procedure is applied for the construction of the discretizations equations. In this paper, and unlike the case for most fully coupled formulations available in the literature, the Poisson pressure equation has been retained from the segregated approach. The use of a pressure equation leads to an unfavourable size increase of the fully coupled linear system, but significantly improves the system's conditioning. The fully coupled system obtained is solved using an ILUT preconditioned GMRES algorithm. The other important element in this paper is the proposal of a Newton linearization of the convection terms in lieu of the common Picard iteration procedure. A systematic comparison between two segregated and four fully coupled fomulations has been presented which has allowed for an evaluation of the individual benefits and strengths of the coupling and linearization procedure by studying lid-driven cavity problems and flows past a circular cylinder. All coupled formulations have proven to be significantly superior both in robustness and efficiency, as compared with the segregated formulation. In some circumstances, the coupled methods yield a converged solution of the system of discretized equations constructed using the FLO scheme, while the segregated formulations diverge. Compared to Picard's linearization, Newton's linearization is more efficient at reducing the number of iterations needed to converge, but requires more computational effort per iteration from the linear equation solver. Furthermore, the Jacobian matrix should include contributions from the nonlinearity appearing at both the governing-equation level and the interpolation-scheme level to ensure Newton's method convergence. The key element in guaranteeing successful, fully coupled solutions lies in the use of an efficient linear equation solver and preconditioner. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: control-volume finite element method; incompressible Navier–Stokes; Poisson pressure equation

*Correspondence to: C. Masson, École de Technologie Supérieure, 1100 Notre-Dame West, Montréal, Québec, Canada H3C 1K3.
†Ph.D. student.
§Professor, Holder of the Canada Research Chair on the Aerodynamics of Wind Turbines in Nordic Environment.
‡E-mail: christian.masson@etsmtl.ca

Contract/grant sponsor: Canada Research Chair
Contract/grant sponsor: Canadian Natural Resources Ministry
Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada (NSERC)

Received 4 April 2003
Revised 30 September 2003

1. INTRODUCTION

In the incompressible Navier–Stokes equations, pressure appears only through its gradient in the momentum equations and is only indirectly specified *via* the continuity equation. This implies that there is no transport equation describing the evolution of the pressure field and that only its relative value is relevant. The lack of a dedicated pressure equation is responsible for the difficulties encountered in solving the incompressible Navier–Stokes equations. Numerous approaches have been suggested to overcome this problem. Apart from the vorticity-based and pseudocompressibility methods, the most commonly used methods use the so-called Poisson pressure equation approach. In this approach, the incompressibility constraint is satisfied through a pressure equation derived from the continuity equation by expressing velocity derivatives as a function of the pressure gradient using the momentum equations. The most common approach for solving this system of equations, known as the segregated solution procedure, consists in solving these equations independently, in sequence. The pressure–velocity coupling is solved iteratively, successively updating the velocity variables in the momentum equations and the pressure in the Poisson pressure equation. SIMPLE (semi-implicit method for pressure-linked equations) [1] and other derived methods such as SIMPLER [1] and PISO [2], using an iterative treatment of the pressure–velocity coupling, have demonstrated their ability to solve complex problems. However, not only do these methods require some form of underrelaxation and solver tuning to ensure convergence, they may also fail to reach a solution for some problems. Weak pressure–velocity coupling is one of the key components of these methods, in that it partly controls the convergence rate of the overall solution procedure. This aspect becomes dominant as the number of grid points increases. Rapid advances in computer speed and available memory combined with recent developments in non-stationary iterative solvers and preconditioners [3–5] for non-symmetric matrices have enabled the development of fully coupled algorithms for the solution of the Navier–Stokes equations. Although research in the finite volume community has been ambivalent on the efficiency of fully coupled methods compared to the more common segregated solution procedures [6, 7], there is common agreement on the added robustness of fully coupled methods. This improved robustness is attributed to the implicit treatment of the pressure–velocity coupling [6, 8].

In this paper, the development of a fully coupled method for solving the two-dimensional incompressible Navier–Stokes equations is presented. The proposed method borrows many of its features from the segregated control volume finite element method (CVFEM) described by Baliga *et al.* [9]. Thus, finite volume discretization is employed on a colocated grid using either the MAW or the FLO scheme described by Masson *et al.* [10], and using an element-by-element assembling procedure. Unlike the case for most fully coupled formulations, in this paper, the Poisson pressure equation of the segregated approach has been retained. Although the use of a Poisson pressure equation is not essential in fully coupled solution contexts and leads to the unfavourable increase in size of the fully coupled linear system, it does significantly improve matrix system conditioning. Straightforward discretization of the Navier–Stokes equations (i.e. when no pressure equation is introduced through a treatment of the continuity equation) leads to very ill-conditioned systems resulting from the presence of zeros in the main diagonal for the continuity equation since pressure is not appearing explicitly. These zeros in the main diagonal can be eliminated by reordering the variables or by using a penalty formulation [6, 11]. In either case, these treatments are palliative, whereas the use of

a poisson pressure equation eliminates the source of the problem. Thus, efficiency problems associated with difficulties in solving the ill-conditioned linear systems encountered in previous fully coupled efforts [6, 11] can be alleviated by using a Poisson pressure equation formulation. This approach allows full benefits to be obtained from the implicit pressure–velocity coupling and this is what motivates its use in this paper. The resulting fully coupled system can be solved by two different methods. The most common is to use a direct solver such as a Gaussian elimination method. However, the use of direct methods is limited by both storage requirements and computing time, which prevent their use in large problems. This drawback has been countered by recent developments in non-stationary iterative solution algorithms [5, 4] such as Krylov subspace methods. Besides requiring fewer computer resources, using an iterative solver provides flexibility in controlling the extent to which the linear system is solved, contrary to direct methods, which yield the exact solutions. Hence, the tolerance of the iterative linear equation solver can be relaxed when far from the converged solution and tightened as convergence approaches, leading to a more efficient algorithm. In this paper, a restarted version of Saad's GMRES [12] is employed using an incomplete LUT decomposition as a preconditioner. Although this algorithm may not be the fastest among all of the iterative solvers, this preconditioned Krylov subspace method constitutes a robust and efficient solver guaranteeing monotonic convergence.

Apart from the implicit treatment of the pressure velocity coupling, the use of a fully coupled formulation has additional benefits for the algorithm, because it is capable of implicitly treating the nonlinearity of the momentum equations' convection terms. Thus, the more efficient Newton linearization can be applied in place of the commonly used Picard iteration procedure, leading to a faster solution algorithm.

The paper is arranged as follows: The governing equations are first presented in Section 2. The ingredients for the numerical discretization of these equations are then detailed in Section 3. Starting from the SIMPLER segregated algorithm, the proposed fully coupled formulation is presented. Particular emphasis is placed on the description of the linearization of the convective terms, presenting both Picard's and Newton's linearizations. Other significant aspects include the derivation of a pressure equation allowing the formulation of a co-located equal order CVFEM that does not suffer from checkerboard pressure fields in the simulation of incompressible fluid flows. In the final section, the performance of the proposed fully coupled CVFEM methods are compared to that of their segregated counterparts through lid-driven cavity problems and flows past circular cylinders.

2. GOVERNING EQUATIONS

Within the finite volume framework, the most appropriate basis for the discretization process is the integral form of the conservation laws. In conservative integral form, the incompressible laminar, steady-state Navier–Stokes equations over a control volume $\mathscr{V}$ enclosed by the surface $\mathscr{A}$ read as follows:

Continuity equation

$$\int_A \rho u_j n_j \, \mathrm{d}\mathscr{A} = 0 \tag{1}$$

Table I. Specific forms of the general transport equation.

	Γ_ϕ	ϕ	S_ϕ
x-Momentum	μ	u	$-(\partial p/\partial x)$
y-Momentum	μ	v	$-(\partial p/\partial y)$
Continuity	0	1	0

Momentum equations

$$\int_{\mathscr{A}} (\rho u_j n_j) u_i \, \mathrm{d}\mathscr{A} = -\int_{\mathscr{V}} \frac{\partial p}{\partial x_i} \, \mathrm{d}\mathscr{V} + \int_A \mu \frac{\partial u_i}{\partial x_j} n_j \, \mathrm{d}\mathscr{A} \tag{2}$$

where u_i is a Cartesian component of the velocity vector $\mathbf{V}$, ρ and μ represent the fluid density and viscosity, respectively, and p is the pressure. n_j is a component of the outward unit vector normal to the elementary surface $\mathrm{d}\mathscr{A}$. A general description of a transport phenomenon involving the scalar dependent variable ϕ is given by

$$\int_{\mathscr{A}} (\rho u_j n_j) \phi \, \mathrm{d}\mathscr{A} = \int_A \Gamma_\phi \frac{\partial \phi}{\partial x_j} n_j \, \mathrm{d}\mathscr{A} + \int_{\mathscr{V}} S_\phi \, \mathrm{d}\mathscr{V} \tag{3}$$

The momentum and the continuity equations, Equations (1) and (2), can be obtained from Equation (3) by defining the independent variable, ϕ, the diffusion coefficient, Γ_ϕ, and source term, S_ϕ, according to Table I.

3. NUMERICAL APPROACH

The proposed numerical method is a two-dimensional CVFEM based on a primitive-variable, co-located, equal-order formulation that shares some similarities with the work of Baliga, Saabas and Masson [9, 10], Prakash and Patankar [13] and Schneider and Raw [14]. The two main aspects that dissociate this method from prior formulations are the implicit treatment of the pressure–velocity coupling and the proposal of a Newton linearization of the convection terms. Therefore, for the sake of clarity, a concise description of the various common building blocks involved in the formulation of a CVFEM are first presented in this section. Then, building upon this framework, a SIMPLER-based CVFEM and the proposed fully coupled method are successively constructed, placing some emphasis on the aspects distinguishing the two approaches. This section ends by extending the ideas proposed in the fully coupled method by implementing a Newton linearization of the convection terms.

3.1. Building blocks

3.1.1. Domain discretization. The spatial discretization procedure begins by dividing the two-dimensional computational domain into triangular elements upon the vertices at which flow variables are stored. Polygonal control volumes are then formed by joining the midpoints of the sides of each element to the centroid as shown in Figure 1; the solid lines denote domain and element boundaries; the dashed lines represent control-volume faces and the shaded areas show the control volumes associated with one internal node and one boundary node.

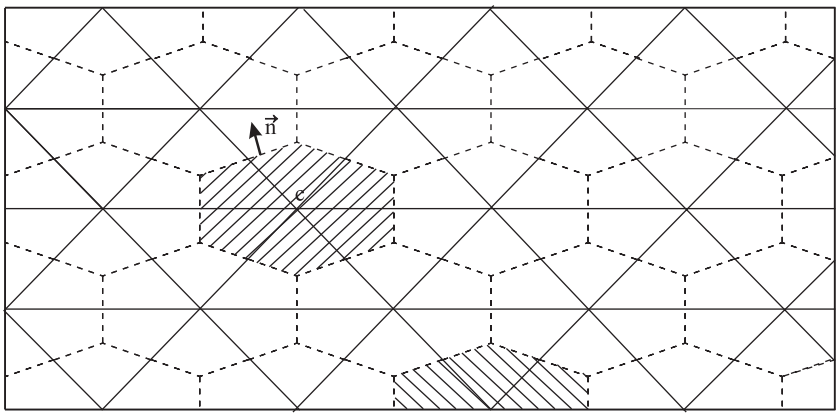

Figure 1. Example of a calculation domain and its discretization into triangular elements and polygonal control volumes.

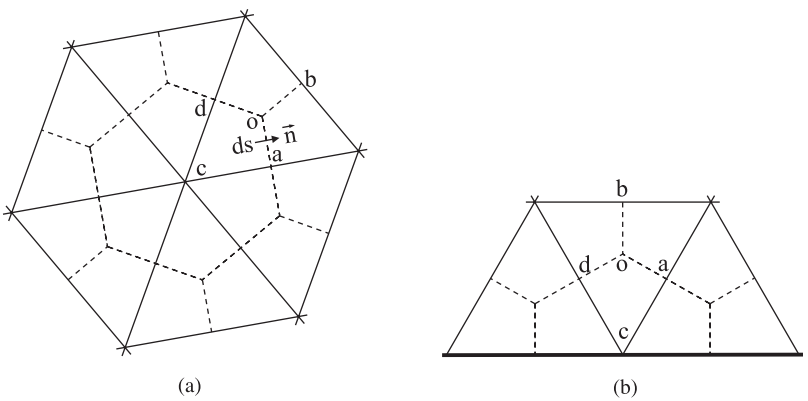

Figure 2. Typical control volumes associated with (a) an internal node, (b) a boundary node.

3.1.2. Integral conservation equation. Consider a typical node c in the calculation domain: it could be an internal node, such as the one shown in Figure 2(a) or a boundary node, similar to the one shown in Figure 2(b). The general integral conservation equation (Equation (3)), when applied to the control volume surrounding node c in Figure 2, can be written as follows:

$$\sum_{k=1}^{M}\left[\int_a^o \mathcal{J}_j n_j \,\mathrm{d}s + \int_o^d \mathcal{J}_j n_j \,\mathrm{d}s - \int_{caod} S_\phi \,\mathrm{d}\mathcal{V}\right] + [\text{boundary contribution if applicable}] = 0 \quad (4)$$

where M is the number of triangular elements having node c as a vertex, n_j is a unit outward vector normal to the differential length element ds, and $\mathcal{J}_j$ is the combined convection–diffusion flux of ϕ:

$$\mathcal{J}_j = \mathcal{J}_{\mathrm{D}j} + \mathcal{J}_{\mathrm{C}j} \quad (5)$$

$$\mathscr{J}_{\mathrm{D}j} = -\Gamma_\phi \frac{\partial \phi}{\partial x_j} \tag{6}$$

$$\mathscr{J}_{\mathrm{C}j} = \rho u_j^m \phi \tag{7}$$

With reference to Equation (7), the superscript is attached to the velocity vector in order to emphasize its connection to the mass flux: this velocity is interpolated in a special way, as discussed in the next section. The form of Equation (4) emphasizes its ability to be assembled using an element-by-element procedure.

3.1.3. Interpolation functions. The derivation of algebraic approximations to the integral conservation equations, Equation (3), requires the specification of element-based interpolation functions for dependent variables and thermophysical properties.

In each element, all thermophysical properties, such as density, ρ and viscosity, μ, are evaluated at the triangular element's centroid o and assumed to prevail over the element.

To obtain an algebraic approximation of the diffusive fluxes, Equation (6), the transported scalar ϕ is interpolated linearly within each element.

In the derivation of algebraic approximations of the convective fluxes, Equation (7), two different interpolation schemes for ϕ have been investigated: a flow oriented upwind scheme (FLO) and a mass weighted upwind scheme (MAW). The FLO scheme is based on the earlier work of Baliga and Patankar [15, 16]. The interpolation function used in this scheme responds appropriately to an element-based Peclet number and to the direction of the element-average velocity vector. In planar two-dimensional problems involving acute-angled triangular elements and relatively low-element Peclet numbers, the FLO scheme has proved quite successful [16, 17]. The MAW scheme [10] ensures, at the element level, that the extent to which the dependent variable at a node exterior to a control volume contributes to the convective outflow is less than or equal to its contribution to the inflow by convection. Thus it is a sufficient condition for ensuring that the algebraic approximations to the convective terms in Equation (4) add positively to the discretized equation. Furthermore, the MAW scheme takes better account of the influence of the direction of flow than Prakash's donor–cell scheme [18]. Thus the MAW scheme produces less false diffusion than the donor-cell scheme. In problems with acute-angled triangular elements and relatively low-element Peclet numbers, the FLO scheme is more accurate than the MAW scheme. However, when high-element Peclet numbers are involved, especially in conjunction with obtuse-angled elements, the FLO scheme produces negative coefficients in the discretized equations which can lead to numerical difficulties and physically meaningless solutions [10]. When such difficulties are encountered, the MAW scheme is recommended. As demonstrated and discussed in the Results section, the convergence difficulties of the FLO scheme seem to be less severe when a fully coupled formulation is used.

Pressure is interpolated linearly within each element.

The proposed formulation uses co-located grids; hence, to avoid spurious pressure modes, Prakash and Patankar's approach [13] has been adopted. A distinction is made between mass velocities, denoted u^m in Equation (7) and corresponding to u_j in Equations (1)–(3) and convected velocities corresponding to u_i in Equations (2) and (3). Mass fluxes, (ρu_j), in the momentum and continuity equations, Equations (1)–(2), are interpolated using a particular form of the discretized momentum equations according to the approach of Saabas

and Baliga [17]. The mass velocities' $(u_i^m)_c$ definitions are very similar to the expressions of the convected velocities $(u_i)_c$ obtained by rewriting the discretized momentum equations, Equation (13), with the exception that the volume-averaged pressure gradient, $\overline{\partial p/\partial x_i}$, is now evaluated for the element level instead of the control volume. Following Prakash and Patankar [13], the mass velocities u_i^m can be considered as the sum of a pseudo-velocity, $(\hat{u}_i)_c$, and a pressure gradient term, $-(d^{u_i})_c(\overline{\partial p/\partial x_i})_{\text{ele}}$:

$$(u_i^m)_c = (\hat{u}_i)_c - d_c^{u_i}\left(\frac{\overline{\partial p}}{\partial x_i}\right)_{\text{ele}} \tag{8}$$

where

$$(\hat{u}_i)_c = \frac{\sum_{\text{nb}} a_{\text{nb}}^{u_i}(u_i)_{\text{nb}} + b^{u_i}}{a_c^{u_i}} \tag{9}$$

$$d_c^{u_i} = \frac{\mathscr{V}_{\text{CV}}}{a_c^{u_i}} \tag{10}$$

The subscript nb refers to the nodes neighbouring to the grid point of interest c. Substituting the pseudo-velocities and the pressure terms into the continuity equation leads to a Poisson equation for the pressure. On each element, mass fluxes are interpolated by assuming linear variation of the pseudo-velocities, $(\hat{u}_i)_c$ and the pressure coefficient, $d_c^{u_i}$, in conjunction with the local element pressure gradient.

3.2. Segregated formulation

Using the previously described interpolation functions, the Navier–Stokes equations, Equations (1)–(2), can be discretized into the following sets of equations for any particular node of interest c:

Continuity equation

$$a_c^p p_c = \sum_{\text{nb}} a_{\text{nb}}^p p_{\text{nb}} + b^p \tag{11}$$

where

$$b^p = \sum_{k=1}^{M}\left[\int_a^o \rho\hat{u}_j n_j \,\mathrm{d}s + \int_o^d \rho\hat{u}_j n_j \,\mathrm{d}s\right] \tag{12}$$

Momentum equations

$$a_c^{u_i}(u_i)_c = \sum_{\text{nb}} a_{\text{nb}}^{u_i}(u_i)_{\text{nb}} + b^{u_i} + \left(\frac{\overline{\partial p}}{\partial x_i}\right)_{\text{CV}}\mathscr{V}_{\text{CV}} \tag{13}$$

where

$$\left(\frac{\overline{\partial p}}{\partial x_i}\right)_{\text{CV}}\mathscr{V}_{\text{CV}} = \sum_{k=1}^{M}\int_{\text{caod}}\left(\frac{\partial p}{\partial x_i}\right)_k \mathrm{d}\mathscr{V} \tag{14}$$

Written in this form, these coupled sets of non-linear discretized equations can be solved in a segregated manner using an iterative sequential variable adjustment scheme [17] similar to the well-known SIMPLER [1] algorithm without a pressure correction equation:

1. Start with a guessed velocity field.
2. Calculate the coefficients in the momentum equations, Equation (13), without the contributions of the pressure gradient terms.
3. Calculate $\hat{u}_i$ and d^{u_i}.
4. Calculate the coefficients of the pressure equations, Equation (11).
5. Solve the pressure equations.
6. Complete the momentum equations by adding the pressure gradient terms, under-relax these equations, and solve.
7. Return to step 2 and repeat until convergence.

For this paper, steps 2–7 of this procedure were repeated until the non-dimensional sum of the absolute values of the residues for each set of discretization equations was less than 10^{-8}.

This segregated algorithm is based on successive substitution, or Picard, linearization of the convective transport terms. In each cycle of this algorithm, the linearized sets of discretized equations for p, u, v are solved sequentially. The discretized equations for u and v are under-relaxed, using Patankar's implicit under-relaxation procedure [1] just before they are solved. For all computations, the values of the under-relaxation parameters has been set to 0.5 for the momentum equations. The discretized equations for p are *not* under-relaxed.

In order to facilitate the implementation and testing of the proposed CVFEM, structured grids were used in this study: the nodes in the finite element mesh lie along non-orthogonal lines that allow (I,J) indexing and each node has a maximum of eight neighbours. Thus, in the structured-grid implementation used here, each of the discretized equations follows a nine-diagonal matrix structure as presented in Figure 3. This system can be cast into three tridiagonal matrix systems and solved independently in sequence for $u_i(u,v)$ and p, using an iterative line-by-line Gauss–Seidel method based on the Thomas tridiagonal matrix algorithm [1]. As subsequently detailed in Section 3.3, a similar block tridiagonal solution procedure cannot be used for solving the matrix system corresponding to the proposed fully coupled formulation. In this study, the coupled-equation system, which strictly requires the simultaneous solution update of all independent variables over the whole computational domain, is solved using Saad's GMRES(m) algorithm [12] preconditioned by an incomplete LUT decomposition. Therefore, in order to properly evaluate the sole effect of the proposed pressure–velocity coupling procedure of the fully coupled formulation on the efficiency of the overall algorithm, the same ILUT preconditioned GMRES(m) algorithm was also implemented in the segregated formulation.

3.3. Fully coupled formulation

Although not explicitly expressed, the segregated discretized equations, Equations (11)–(13), already contain the necessary elements for their coupled solution. In the momentum equations, the volume-averaged pressure gradient over the control volume surrounding node c can be

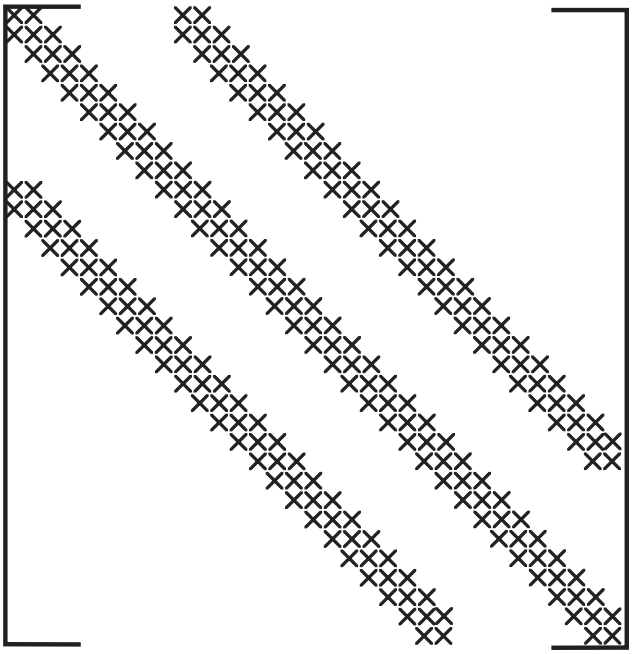

Figure 3. Matrix pattern for the segregated method.

expressed in terms of the pressure at its neighbouring grid points:

$$\left(\frac{\overline{\partial p}}{\partial x_i}\right)_{\mathrm{CV}} \mathscr{V}_{\mathrm{CV}} = \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{u_i,p}\, p_{\mathrm{nb}} \tag{15}$$

Similarly, in the continuity equation, the term b^p can be expressed explicitly in terms of the pseudo-velocity components at the neighbouring grid points:

$$b^p = -a_c^{p,\hat{u}}\hat{u}_c + \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{p,\hat{u}}\hat{u}_{\mathrm{nb}} - a_c^{p,\hat{v}}\hat{v}_c + \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{p,\hat{v}}\hat{v}_{\mathrm{nb}} \tag{16}$$

Having recognized the implicit dependence of the discretized momentum and continuity equations in p, $\hat{u}$ and $\hat{v}$, one can obtain the following set of equations:
Continuity equation:

$$a_c^p\, p_c + a_c^{p,\hat{u}}\hat{u}_c + a_c^{p,\hat{v}}\hat{v}_c = \sum_{\mathrm{nb}} a_{\mathrm{nb}}^p\, p_{\mathrm{nb}} + \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{p,\hat{u}}\hat{u}_{\mathrm{nb}} + \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{p,\hat{v}}\hat{v}_{\mathrm{nb}} \tag{17}$$

Momentum equations:

$$a_c^{u_i}(u_i)_c + a_c^{u_i,p}\, p_c = \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{u_i}(u_i)_{\mathrm{nb}} + \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{u_i,p}\, p_{\mathrm{nb}} \tag{18}$$

Finally, the expression of $\hat{u}_i$ closes the fully coupled system:

$$a_c^{u_i}(\hat{u}_i)_c = \sum_{\mathrm{nb}} a_{\mathrm{nb}}^{u_i}(u_i)_{\mathrm{nb}} \tag{19}$$

which can be written in matrix form

$$F(X) = A_c X_c + \sum_{\text{nb}} A_{\text{nb}} X_{\text{nb}} - B = 0 \tag{20}$$

where $X_c = [u_c, v_c, \hat{u}_c, \hat{v}_c, p_c]^{\text{T}}$ and $X_{\text{nb}} = [u_{\text{nb}}, v_{\text{nb}}, \hat{u}_{\text{nb}}, \hat{v}_{\text{nb}}, p_{\text{nb}}]^{\text{T}}$. In 2D flows, the structures of the matrix A_c and A_{nb} are given by

$$A_c = \begin{pmatrix} a_c^u & 0 & 0 & 0 & a_c^{u,p} \\ 0 & a_c^v & 0 & 0 & a_c^{v,p} \\ 0 & 0 & a_c^u & 0 & 0 \\ 0 & 0 & 0 & a_c^v & 0 \\ 0 & 0 & a_c^{p,\hat{u}} & a_c^{p,\hat{v}} & a_c^p \end{pmatrix} \tag{21}$$

$$A_{\text{nb}} = \begin{pmatrix} -a_{\text{nb}}^u & 0 & 0 & 0 & -a_{\text{nb}}^{u,p} \\ 0 & -a_{\text{nb}}^v & 0 & 0 & -a_{\text{nb}}^{v,p} \\ -a_{\text{nb}}^u & 0 & 0 & 0 & 0 \\ 0 & -a_{\text{nb}}^v & 0 & 0 & 0 \\ 0 & 0 & -a_{\text{nb}}^{p,\hat{u}} & -a_{\text{nb}}^{p,\hat{v}} & -a_{\text{nb}}^p \end{pmatrix}$$

The vector B contains contributions related only to Dirichlet boundary conditions since all other boundary condition contributions are treated implicitly. Thus, for two-dimensional structured grids, the linear system involves a 5×5 block nine-diagonal matrix. In this formulation, the pseudo-velocities are treated as primary unknowns, which ensures a nine diagonal matrix structure. The pseudo-velocities could have been expressed directly in terms of velocities in the discretized equations leading to a more compact 3×3 block matrix structure [19]. However, with the mass flux interpolation procedure adopted in this study, the continuity equations would have involved 24 velocity neighbours of point c instead of eight in the present formulation. This would have greatly increased the complexity of the assembling procedure and the implementation of a Newton-type linearization without any significant storage savings.

The following sequence of operations outlines the coupled solution procedure for the discretized Navier–Stokes equations:

1. Start with guessed velocity and pressure fields.
2. Calculate the coefficients in the momentum equations, Equation (18), and the pseudo-velocity equations (Equation (19)).
3. Calculate d^{u_i}.
4. Calculate the coefficients of the pressure equation (Equation (17)).
5. Simultaneously solve the momentum, pressure and pseudo-velocity equations.
6. Return to step 2 and repeat until convergence.

In a procedure that is similar to the one used for the segregated algorithm, steps 2–6 of the coupled solution procedure are repeated until the non-dimensional sum of the absolute values

of the residues for the discretized continuity and momentum equations are less than 10^{-8}. In the fully coupled formulation, the discretized equations are solved simultaneously for velocity components, u_i, pseudo-velocity components, $\hat{u}_i$, and pressure, p. No relaxation or special treatment is required to ensure convergence, as equation coupling is done implicitly.

The 5×5 block nine-diagonal structure of the fully coupled linear system is very similar to the one established using the segregated approach (see Figure 3), with a 5×5 block now replacing the single segregated equation coefficient. Hence, it would be natural to cast this system into a 5×5 block tridiagonal system and solve it using line-by-line Gauss–Seidel iterations. However, this approach systematically fails to converge regardless of the number and order of the sweeps. When solving for pressure in the segregated procedure, the pseudo-velocity field is kept constant within an overall iteration. This is not the case in the proposed fully coupled formulation, where pseudo-velocities, $\hat{u}_i$, are primary unknowns. Therefore, successively solving and updating the pseudo-velocities using the line-by-line Gauss–Seidel algorithm creates significant local mass imbalances and leads to divergence. As a result, the linear solver must simultaneously update all the unknowns over the whole computational domain. The fully coupled system of discretized equations can be solved using direct methods. However, direct methods such as Gaussian elimination can be extremely costly in terms of both CPU time and memory requirements owing to large matrix bandwidths, and therefore limit the size and complexity of tractable problems. The future extension of this research to complex problems involving two- and three-dimensional fine grids motivates the use of preconditioned sparse matrix iterative solution algorithms for solving the fully coupled system. These algorithms are implemented to take advantage of sparse, banded matrix structures such as the ones encountered in CFD problems. In this study, the restarted version of Saad's GMRES algorithm [12] is employed using an ILUT decomposition as a preconditioner. A well-known limitation of GMRES is that the amount of computational work and storage required per iteration rises linearly with the iteration count. GMRES quickly becomes impractical for large grids. GMRES(m) offers a remedy to this problem by restarting the algorithm after m iteration. Accumulated data are cleared and intermediate results are used as the initial data for the following m iterations. This procedure is repeated until convergence is achieved and leads to a more efficient solver. Although significant CPU savings can be realized by fine tuning the solver's parameters (such as the number of ILUT fill in, the size of Krylov subspace before restart and the tolerance stopping criteria), this issue has not been addressed in this paper. For the calculations presented in Section 4, the control parameters were set to baseline recommended values [12] and were modified only when convergence difficulties occurred. The only solver modification implemented in this study to speed up the solution of the linear system was the freezing of the preconditioner [20] for some iterations after an approximate, but realistic, solution was obtained. This strategy proved to be both quite successful and easy to implement.

3.4. Newton's linearization

Apart from the implicit pressure–velocity coupling, the use of a fully coupled formulation also allows for the implicit treatment of the convection terms' non-linearities in the momentum equations. Newton's method is an efficient technique for solving systems of non-linear equations with this form

$$F(X) = [f_1(X), f_2(X), \ldots, f_N(X)]^{\mathrm{T}} = 0 \tag{22}$$

where X can be expressed as $X=[x_1,x_2,\ldots,x_N]^{\mathrm{T}}$. The application of Newton's method requires the solution of the linear system:

$$J^n\delta X^n=-F(X^n) \tag{23}$$

where the elements of the Jacobian matrix J are defined by $j_{ij}=\partial f_i/\partial x_j$ and the new solution approximation is obtained from

$$X^{n+1}=X^n+d\delta X^n \tag{24}$$

The constant $d(0\leqslant d\leqslant 1)$ in the above equation is used to damp the Newton updates when far from the converged solution. In this study however, no relaxation was necessary in order to obtain the solutions presented in Section 4. Forming the Jacobian matrix J can be done either by numerically evaluating the derivatives applying finite difference approximations [7] to Equation (20), or *via* direct algebraic differentiation. Although more tedious to implement, algebraic differentiation eliminates the errors associated with finite differencing at a fraction of computing effort.

The non-linearities in the Navier–Stokes equations appear in the momentum equations only through the convection terms from the product $(u_j^m)(u_i)$. However, neglecting the dependence of u_i in u^m through the interpolation convection scheme leads to significant errors in the Jacobian matrix and lowers the efficiency of the algorithm. Thus, the interpolated value of u_i must be differentiated with respect to the five primary unknowns in order for the algorithm to benefit from Newton's linearization. In this study, the MAW and the FLO scheme where both differentiated. Non-linearities related to the mass velocity definition (see Equation (19)) are present in the fully coupled system of equations (Equation (20)). Their contributions are not included in the proposed Jacobian matrix. The Jacobian matrix structures remains a 5×5 block nine-diagonal matrix taking the form

$$J^n\delta X^n=(A_c+J_c)\delta X_c+(A_{\mathrm{nb}}+J_{\mathrm{nb}})\delta X_{\mathrm{nb}} \tag{25}$$

with

$$J_c=\begin{pmatrix} 0 & 0 & j_c^{\hat{u}} & j_c^{\hat{v}} & j_c^{u,p} \\ 0 & 0 & j_c^{\hat{u}} & j_c^{\hat{v}} & j_c^{v,p} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{26}$$

$$J_{\mathrm{nb}}=\begin{pmatrix} 0 & 0 & -j_{\mathrm{nb}}^{\hat{u}} & -j_{\mathrm{nb}}^{\hat{u}} & -j_{\mathrm{nb}}^{u,p} \\ 0 & 0 & -j_{\mathrm{nb}}^{\hat{v}} & -j_{\mathrm{nb}}^{\hat{v}} & -j_{\mathrm{nb}}^{v,p} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \tag{27}$$

As the basic structure of the equation system Equation (23) associated with Newton's linearization is similar to that of system Equation (20) as related to Picard's, the same coupled solution procedure has been used for both linearization schemes. In Newton's method, the only special treatment needed is to set J_c and J_{nb} to zero for a few iterations to initiate the algorithm, until an approximate but realistic solution is established.

4. RESULTS

In this section, six specific algorithms are compared systematically in terms of their computational efficiency for an internal and external flow problem. There are two segregated algorithms which differ only with respect to the chosen linear solver. The first one uses a line-by-line Gauss–Seidel method based on a Thomas tridiagonal matrix solver, while the second formulation uses Saad's GMRES(m) solver. There are four fully coupled algorithms which all use the GMRES(m) solver. Two of them use Picard's linearization while the last two are based on Newton's linearization. For a given type of linearization (Picard's or Newton's), there is one algorithm with and another algorithm without the freezing of the ILUT preconditioner. For a given convection scheme (FLO or MAW), the six algorithms produce exactly the same converged solution, since they solve the same set of discretized equations. The solution accuracy varies only in terms of the selected convection scheme (FLO or MAW). The proposed fully coupled algorithm necessitate about 20 times the memory requirements of its segregated counterpart making a possible extension to practical three-dimensionnal problems untractable on a personal computer.

4.1. Lid-driven cavity flow

The classical 2D square-driven cavity problem $0 \leqslant x \leqslant L$, $0 \leqslant y \leqslant L$ with $u(x, L) = U_{wall}$ is first used to demonstrate the performance of the proposed fully coupled methods (see Figure 4). To qualify the accuracy of the proposed formulation, calculations have been performed on four different uniform grids using both the MAW and the FLO schemes. Tables II and III compare values of minimal and maximal centerline velocities at $Re = 400$ and 1000, respectively. The Reynolds number for this problem has been defined as $Re = \rho U_{wall} L/\mu$. In Tables II and III, no distinction is made among the calculation algorithms (segregated, fully coupled or Newton) since they do not affect the values of the converged solutions.

Using the FLO scheme, the proposed CVFEM yields results that are in good agreement with the benchmark calculations made by Ghia *et al.* [21]. However, the MAW scheme computations systematically display lower accuracy, confirming the observations of Saabas *et al.* [9, 10]. The benefits of the fully coupled and the Newton formulations are quantifiable in terms of added efficiency and robustness. Inferences as to the robustness of the fully coupled procedure can be drawn from the fact that no form of relaxation is needed to obtain the results presented here, contrary to the segregated algorithm, where underrelaxation is necessary to offset the weak iterative pressure–velocity coupling. In Newton's method, the only special treatment needed is to hold back the introduction of the linearized convection terms of the Jacobian matrix for a few iterations to initiate the algorithm, until an approximate but realistic solution is established.

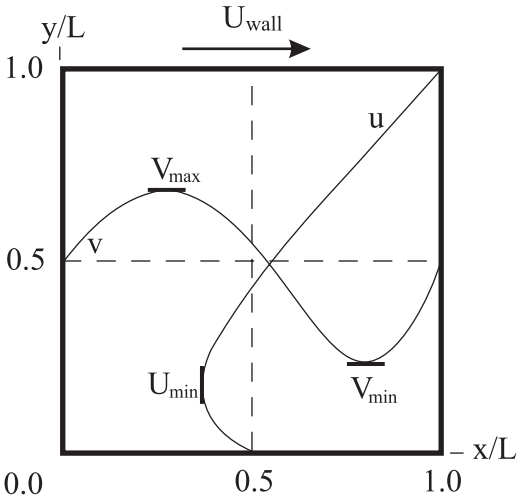

Figure 4. Definitions for lid-driven cavity flows.

Table II. Maximum and minimum centrelinevelocities—$Re = 400$.

Method	Grid	$\frac{U_{min}}{U_{wall}}$	$\frac{V_{min}}{U_{wall}}$	$\frac{V_{max}}{U_{wall}}$
MAW	32×32	−0.2290	−0.3610	0.2196
MAW	64×64	−0.2608	−0.3902	0.2449
MAW	96×96	−0.2780	−0.4079	0.2597
MAW	128×128	−0.2884	−0.4180	0.2687
FLO	32×32	−0.2796	−0.3994	0.2608
FLO	64×64	−0.3097	−0.4336	0.2867
FLO	96×96	−0.3191	−0.4436	0.2951
FLO	128×128	−0.3231	−0.4477	0.2987
Ghia [21]	129×129	−0.3273	−0.4499	0.3020

Table III. Maximum and minimum centreline velocities—$Re = 1000$.

Method	Grid	$\frac{U_{min}}{U_{wall}}$	$\frac{V_{min}}{U_{wall}}$	$\frac{V_{max}}{U_{wall}}$
MAW	32×32	−0.2561	−0.4230	0.2362
MAW	64×64	−0.2798	−0.4385	0.2664
MAW	96×96	−0.3015	−0.4571	0.2885
MAW	128×128	−0.3162	−0.4699	0.3036
FLO	32×32	−0.3061	−0.4551	0.2898
FLO	64×64	−0.3419	−0.4828	0.3271
FLO	96×96	−0.3596	−0.4988	0.3458
FLO	128×128	−0.3692	−0.5079	0.3562
Ghia [21]	129×129	−0.3829	−0.5155	0.3709

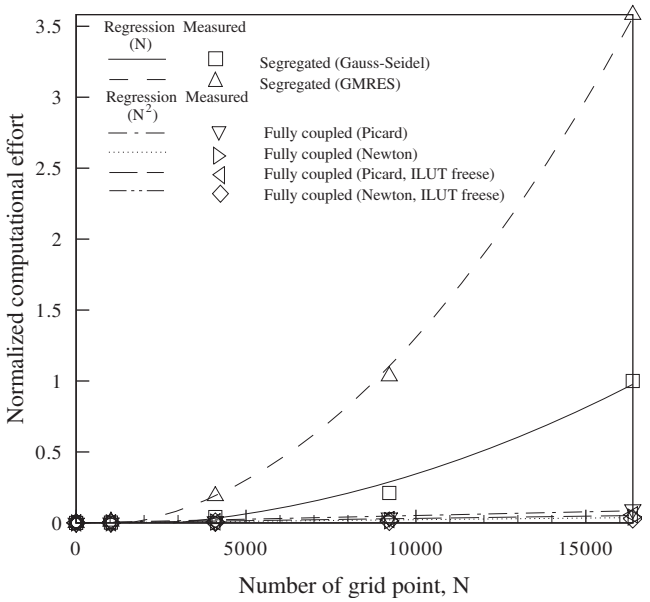

Figure 5. Evolution of normalized computational effort with number N of points, lid-driven cavity flow—$Re = 400$, FLO scheme.

To assess the efficiency of each of the proposed formulations, Figure 5 shows the evolution of the normalized computational effort needed to reach a value of 10^{-8} for the non-dimensional sum of the absolute values of the residues for each set of discretization equations. The residuals have been non-dimensionalized using reference length L and velocity U_{wall}. The normalized computational effort refers to the CPU computational time of a process divided by the CPU computational time required by the segregated formulation with the Gauss–Seidel solver when solving the lid-driven cavity at $Re = 400$ on a 128×128 grid. This reference computational time by which all computational efforts are normalized has a numerical value of 5196 seconds as obtained on an Intel Pentium IV 1.9 GHz running Linux Mandrake 8.1 and using GNU fortran compiler(v0.5.24). Results were obtained using the FLO scheme for solving the lid-driven cavity problem at $Re = 400$. The two segregated solutions presented in Figure 5 differ solely in the type of linear equation solver used. In a segregated-solution context with structured grid, a line-by-line Gauss–Seidel solver clearly proves to be more efficient than an ILUT preconditioned GMRES(m) strategy. Moreover, the difference in efficiency between the two methods increases with the number of grid points. Although much simpler to implement, line-by-line Gauss–Seidel takes better advantage of the multi-diagonal structure of the system and is therefore more efficient. In a structured-grid context, no efficiency gain seems to be drawn from the use of a non-stationary iterative solver such as GMRES(m). As previously mentioned, for mass conservation reasons, coupled solutions require the simultaneous updating of all flow variables over the whole computational domain, making a non-stationary solver mandatory. Hence, any improvement from the coupled method over the segregated method should be attributed solely to the implicit coupling algorithm rather than the solver. Regardless

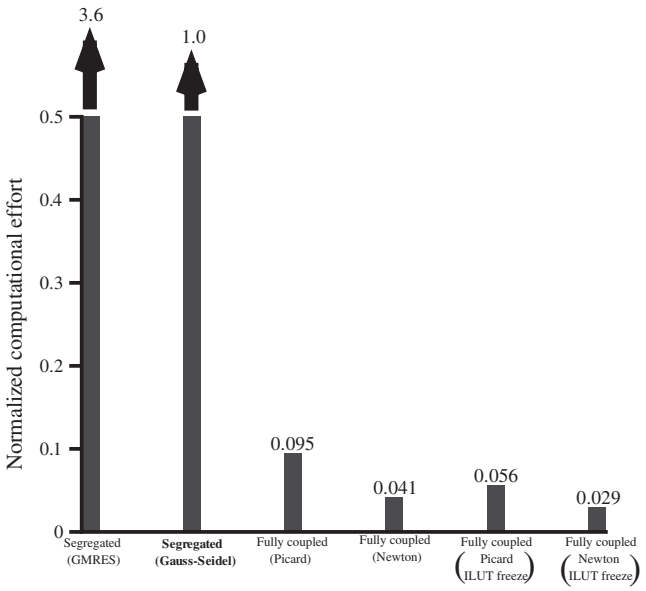

Figure 6. Normalized computational effort for a 128×128 grid, lid-driven cavity flow—$Re = 400$, FLO scheme.

of the number of grid points, all the proposed fully coupled formulations are more efficient than the segregated approaches. The evolution of the latter's normalized computational effort seems to follow a quadratic (N^2) behaviour, while their fully coupled counterparts display a linear form (N). The quadratic and linear regressions shown in Figure 5 have been fitted to the results with a R^2 value over 95%. Thus, in comparison with the segregated approach, the relative efficiency of the fully coupled methods increases with the number of grid points. One can see in Figure 6 that for a 128×128 grid, the segregated Gauss–Seidel method is more than three times faster than its GMRES(m) counterpart. Figure 6 also shows that the proposed fully coupled formulation using Newton's linearization and preconditioner freezing strategy is the more efficient one and proves to be 34 times faster than the segregated approach using Gauss–Seidel. Since most practical CFD problems require more than 16 000 grid points, significant computational effort savings can be made especially within the context of an optimization procedure requiring numerous computations. The convergence acceleration achieved through Newton's linearization of the convection terms represents about 40–50% in normalized computational effort savings over that using Picard's linearization.

Some additional insights can be had from studying the convergence histories of the algorithms presented in Figure 7 for the lid-driven cavity problem at $Re = 400$ on a 128×128 grid. Contrary to the segregated solution, both coupled methods display monotonic convergence. The most significant difference lies in the rate of convergence per iteration; 14 300 iterations are necessary in order to reach a residual under 10^{-8} for the segregated method, while only 27 and 13 iterations are required to achieve similar accuracy for the fully coupled methods using Picard's and Newton's linearizations, respectively. This result highlights the highly iterative nature of segregated algorithms where iterations are needed for both solving non-linearities

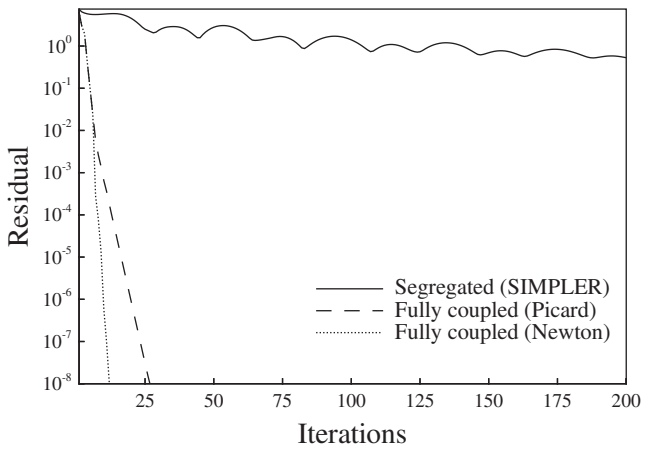

Figure 7. Convergence histories, lid-driven cavity flow—$Re = 400$, 128×128 grid, FLO scheme.

in the PDEs and coupling these equations. In light of these results, it appears that for segregated solutions, most of the iterating work is done for coupling the equations, while in fully coupled algorithms, the sole purpose for iterating is to solve non-linearities. However, these reductions in the number of iterations do not translate into proportional computational time savings. This result arises from the geometric growth of the computational effort of the linear equation solver with the size of the fully coupled system. The matrix conditioning of the fully coupled system is another element that contributes to the slowing down of the coupled solutions: the number of ILUT fill-ins and the maximum size of Krylov subspace have to be increased to ensure convergence of Newton's method. Thus, the key element for successfully guaranteeing fully coupled solutions lies in the use of an efficient linear equation solver and preconditioner.

4.2. Flow past a circular cylinder immersed in a freestream

Simulation of lid-driven cavity flows has confirmed the accuracy and efficiency of the proposed coupled method for internal flow problems involving Dirichlet boundary conditions. Here, the coupled method is validated for laminar steady external unbounded flows past a circular cylinder. The Reynolds number is defined as $Re_d = U_\infty d/\nu$, with d the cylinder diameter and U_∞ the freestream velocity. Simulations have been undertaken for Reynolds numbers below the threshold value of $Re_d = 46 \pm 1$, above which the wake behind the cylinder becomes unsteady and Karman vortex shedding appears [22]. Thus, for the simulated Reynolds numbers $Re_d = 20$ and 40, the flow around a cylinder is steady and symmetric with respect to a transversal axis parallel to the freestream direction ($\mathbf{U}_\infty = U_\infty \mathbf{i}$). Consequently, by taking advantage of the problem's symmetry, the calculation domain can be reduced by half. All simulations have been performed on the half O-grid shown in Figure 8. Two quantities have been used to evaluate the accuracy of the solutions: the drag coefficient, defined by $C_D = D/(1/2)\rho U_\infty^2 d$, and the length of the recirculating zone, L_W. A grid study has revealed that a $130d$ radius domain is necessary to minimize external boundary effects on the flow and to ensure that the drag coefficient reaches Fornberg's benchmark values [23] within a 0.2%

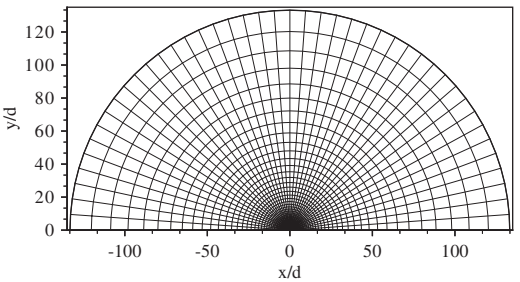

Figure 8. Half O-grid used for the simulations of the flow past a circular cylinder. Only every other grid point is shown.

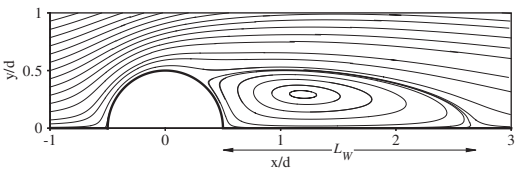

Figure 9. Streamline plot of the flow past a circular cylinder at $Re = 40$, FLO scheme.

Table IV. Comparison of mean drag coefficient C_D and length of wake bubble L_W (measured from the rear end of cylinder)—$Re = 20$ and 40.

Reynolds number→		20		40	
Method	Grid	C_D	L_W/d	C_D	L_W/d
MAW	88×138	2.08	0.83	1.59	1.80
FLO	88×138	2.00	0.90	1.50	2.20
Tritton [22]	Exp.	2.22	N/A	1.48	N/A
Fornberg [23]†	129×132 65×52	2.00	0.91	1.50	2.24

†Two superposed grids were used for the computations.

error margin. Based on the same precision criteria, it has also been demonstrated that 88 grid points along the cylinder surface and 148 grid points in the radial direction are required to guarantee grid-independent solutions. The streamline plot in Figure 9 shows the simulated main recirculating region behind a circular cylinder at $Re = 40$. Table IV compares computed mean drag coefficient C_D and length of wake bubble L_W at $Re = 20$ and 40 with published results [23, 22]. As expected, FLO scheme simulations are in very good agreement with Fornberg's benchmark calculations [23] for both drag coefficient C_D and length of recirculating

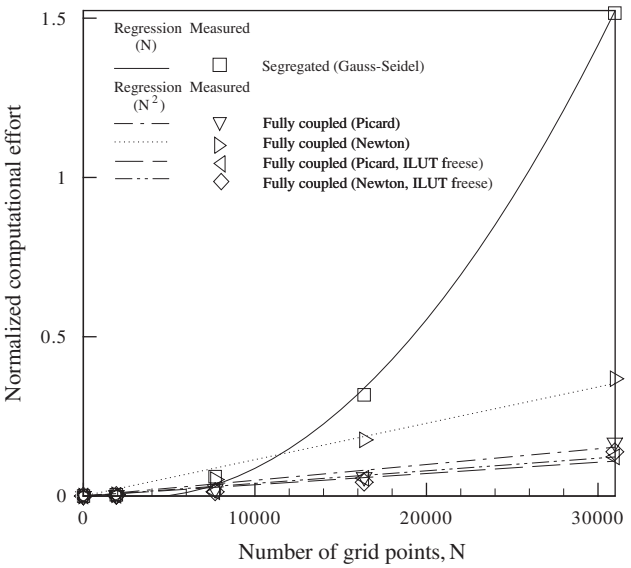

Figure 10. Evolution of normalized computational effort with number N of points, flow past a circular cylinder—$Re = 40$, MAW scheme.

zone L_W. The drag coefficients obtained also compare well to Tritton's experimental results [22], especially for the higher Reynolds number. MAW-scheme computations systematically display lower accuracy, which confirms previous results. All FLO-scheme calculations have been undertaken using fully coupled algorithms as segregated procedure invariably *failed to converge* regardless of the relaxation factors and solver parameters. Convergence difficulties have been previously reported by Saabas and Baliga [17] when using the FLO scheme in segregated solution contexts. These shortcomings might be attributable to the high element-based Peclet numbers of the large grid elements located on the periphery of the domain (see Figure 8). Convergence difficulties of the linear equation solver were also observed when using Newton's linearization, but were easily alleviated by increasing the number of ILUT fill-ins for the preconditioner. These observations are a clear indication of the superior robustness of fully coupled methods over segregated methods.

Figure 10 shows the evolution of normalized computational effort needed to reach a value of 10^{-8} for the non-dimensional sum of absolute values for the residues for each set of discretization equations as a function of the number of grid points. The residuals were non-dimensionalized using the reference length d and velocity U_∞. The computational effort has been normalized with respect to the computational effort required by the segregated formulation with the Gauss–Seidel solver when solving the lid-driven cavity at $Re = 400$ on a 128×128 grid. Results were obtained using the MAW scheme for solving the flow past a circular cylinder at $Re = 40$. As for lid-driven cavity flow, regardless of the number of grid points, all the proposed fully coupled formulations are more efficient than the segregated approach. The evolution of normalized computational effort for the segregated method displays a quadratic (N^2) behaviour while the fully coupled formulations follow a linear form (N). Quadratic and linear regressions are fitted to the results, as shown on Figure 10. Figure 11 shows that for

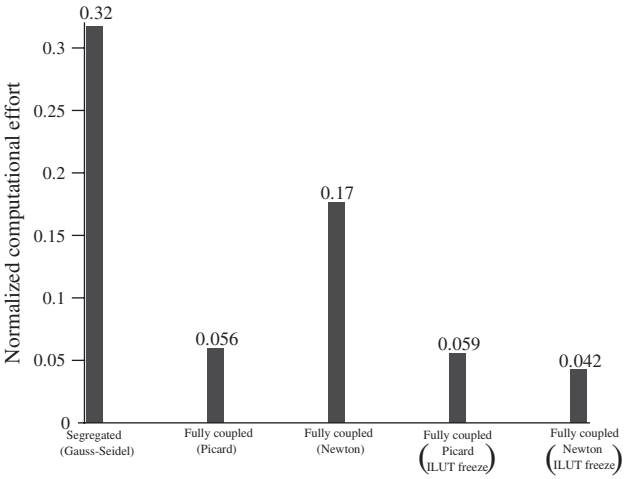

Figure 11. Normalized computational effort, flow past a circular cylinder—$Re = 40$, 98×167 grid, MAW scheme.

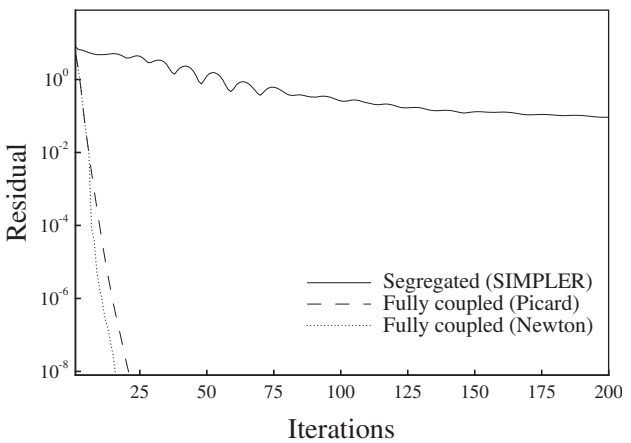

Figure 12. Convergence histories, flow past a circular cylinder—$Re = 40$, 98×167 grid, MAW scheme.

the circular cylinder at $Re = 40$ with 98 grid points along the cylinder surface and 167 grid points in the radial direction (i.e. 16 366 points), the proposed fully coupled formulation using Newton's linearization and preconditioner freezing strategy is the more efficient one and proves to be 7.6 times faster than the segregated approach with the Gauss–Seidel solver. It is interesting to note that Newton's linearization needs more than twice the computational effort of Picard's linearization when no preconditioner freezing strategy is applied. For the flow around a cylinder, Newton's method suffers from convergence difficulties of the linear equation solver which are cured by increasing the number of ILUT fill-ins and the maximum

size of Krylov subspace. However, this remedy has the drawback of increasing the computation effort of the linear equation solver up to non-competitive levels. Newton's linearization requires an economical combination of preconditioner and linear equation solver in order to be advantageous. The use of a preconditioner freezing strategy greatly reduces the computational effort of Newton's method.

Some inferences on these results can be drawn from investigating the convergence histories presented in Figure 12 for the circular cylinder at $Re = 40$ with 98 grid points along the cylinder surface and 167 grid points in the radial direction. Compared to the lid-driven cavity case at $Re = 400$ on a 128×128 grid, the number of iterations required to converge is divided by three for the segregated method; it increases by a third (to 16) and cut by a fifth (to 21) for the fully coupled methods using Newton's and Picard's linearizations, respectively. This might suggest that for the segregated method, the iterative work done for coupling purposes is proportionally lower for the flow around a circular cylinder. It would explain the lower efficiency gain of the fully coupled formulation over the segregated method for this test case.

5. CONCLUSIONS

A fully coupled control-volume finite element method for solving the incompressible Navier–Stokes equations has been implemented and validated using internal and external flow problems. A systematic comparison between two segregated and four fully coupled fomulations has been presented which has allowed the evaluation of the individual benefits and strengths of the coupling and linearization procedure. The proposed method is based on a primitive-variable co-located equal order formulation using either the MAW or the FLO scheme and following an element-by-element assembling procedure. The coupled set of discretized equations obtained is solved using an ILUT preconditioned GMRES(m) algorithm. The special features of the proposed method lie in the prescription of a Poisson pressure equation and a Newton-type linearization. Through lid-driven cavity problems and flows past a circular cylinder, the following comparative remarks can be inferred:

1. The proposed coupled methods are more robust than segregated methods. In some circumstances, the coupled methods yield a converged solution of the system of discretized equations constructed using the FLO scheme, while the segregated formulations diverge.
2. Regardless of the problem, all the proposed fully coupled formulations are more efficient than the segregated approach. This gain in efficiency grows quadratically with the number of grid points, thus yielding significant time savings for large grids.
3. Newton's linearization requires fewer iterations than Picard's to reach convergence.
4. Newton's linearization is less robust than Picard's and requires more work per iteration from the linear equation solver. This is attributed to the ill-conditioning of the Newton's linearization matrix.

All preceding conclusions only hold when solving laminar Navier–Stokes equations problems. The behaviour of a similar fully coupled method involving additional equations related to variables such as turbulence properties or temperature cannot be extrapolated from the present work due to the particular nature of the coupling in these equations which only takes place

through non-linear terms. For laminar Navier–Stokes equations problems, coupled methods prove to be superior in both robustness and efficiency, when compared to segregated methods. The key element in guaranteeing successful fully coupled solutions lies in the use of an efficient linear equation solver and preconditioner, particularly in the context of Newton's linearization. Newton's method is efficient at reducing the number of iterations required for convergence, but it is sensitive to the linear equation solver's control parameters. In order to establish a definitive statement about the performance of Newton's method, a complete evaluation of available preconditioner and linear equation solvers must be undertaken. A possible compromise could be to alternate between Newton's and Picard's linearizations within the same algorithm, based upon the convergence rate per iteration.

NOMENCLATURE

a	discretized equation coefficient
A	block matrix
$\mathscr{A}$	control-volume surface (m^2)
b	right hand side term of the discretized equation
C_D	drag coefficient
d	circular cylinder diameter (m)
d^{u_i}	Pressure-gradient coefficient for u_i
D	drag force per unit length (N/m)
f	non-linear equation
F	non-linear equation system
j	Jacobian matrix element
J	Jacobian matrix
$\mathscr{J}$	combined convection–diffusion flux
$\mathscr{J}_C$	convection flux
$\mathscr{J}_D$	diffusion flux
m	maximum number of Krylov subspace before restart
M	number of triangular elements having node c as vertex
$\dot{M}$	interpolated point value of ρu^m
n_i	outward unit normal component in the i-direction
p	pressure (Pa)
Re	Reynolds number
S	volumetric source term
u_i	velocity in the i-direction (m/s)
u	x-component of the velocity (m/s)
u_i^m	mass-flow related velocity in the i-direction (m/s)
U_∞	freestream velocity (m/s)
$\hat{u}_i$	pseudo-velocity in the i-direction (m/s)
$\hat{u}$	x-component of the pseudo-velocity (m/s)
v	y-component of the velocity (m/s)
$\hat{v}$	y-component of the pseudo-velocity (m/s)
$\mathscr{V}$	control volume (m^3)
x_i	co-ordinate in the i-direction (m)

Greek symbols

Γ_ϕ diffusion coefficient of ϕ
μ dynamic viscosity ($\mathrm{N\,s/m^2}$)
ϕ dependent variable
ρ fluid density ($\mathrm{kg/m^3}$)

Superscript

m pertains to the mass-flow related velocity
n pertains to the n iteration
p pertains to the equation for p
$p, \hat{u}$ pertains to the $\hat{u}$ contribution to the p equation
$p, \hat{v}$ pertains to the $\hat{v}$ contribution to the p equation
u_i pertains to the equation for u_i
u, p pertains to the pressure-gradient contribution to the u equation
u_i, p pertains to the pressure-gradient contribution to the u_i equation
v, p pertains to the pressure-gradient contribution to the v equation

Subscript

c pertains to node c
ele pertains to the element
i index used in tensor notation
j index used in tensor notation
nb pertains to a neighbouring node
ϕ pertains to the dependent variable ϕ

ACKNOWLEDGEMENTS

This study has received support from the Canada Research Chair Program, the Ministère des ressources naturelles du Québec through the Programme d'aide au développement des technologies de l'énergie, and the Canadian Natural Resources Ministry through the Efficiency and Alternative Energy Program. Support from the Natural Sciences and Engineering Research Council of Canada (NSERC) in the form of research grants is gratefully acknowledged.

REFERENCES

1. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: New York, 1980.
2. Issa RI. Solution of the implicitly discretized fluid flow equations by operator-splitting. *Journal of Computational Physics* 1985; **62**:40–65.
3. Saad Y. *Iterative Methods for Sparse Linear Systems*, 2000.
4. Trefethen LN, III, Bau D. *Numerical Linear Algebra*. SIAM: Philadelphia, 1997.
5. Dutto LC. On the iterative methods for solving linear systems of equations. *Technical Report*, Concordia University, 1997.
6. Pascau A, Pérez C, Serón FJ. A comparison of segregated and coupled methods for the solution of the imcompressible Navier–Stokes equations. *Communications in Numerical Methods in Engineering* 1996; **12**:617–630.

7. McHugh PR, Knoll DA. Fully-coupled finite volume solution of incompressible Navier–Stokes and energy equations using an inexact Newton method. *International Journal for Numerical Methods in Fluids* 1994; **19**:439–455.
8. Van Santen H, Lathouwers D, Kleijn CR, Van Den Akker HEA. Influence of segregation on the efficiency of finite volume methods for the incompressible Navier–Stokes equations. In *Proceedings of the Fluids Engineering Division Conference ASME*, vol. 238, 1996; 151–157.
9. Baliga BR. *Advances in Numerical Heat Transfer*, vol. 1, Chapter 3. Hemisphere Publishing Corp., Washington DC, 1996.
10. Masson C, Saabas HJ, Baliga BR. Co-located equal-order control volume finite element method for two-dimensional axisymmetric incompressible fluid flow. *International Journal for Numerical Methods in Fluids* 1994; **18**:1–26.
11. Vanka SP. Block-implicit calculation of steady turbulent recirculating flows. *International Journal of Heat and Mass Transfer* 1985; **28**:2093–2103.
12. Saad Y. Sparskit: a basic tool for sparce matrix computation. *Technical Report*, 90.20, RIACS, NASA Ames Research Center, 1990.
13. Prakash C, Patankar SV. A control-volume-based-finite-element method for solving the Navier–Stokes equations using equal-order velocity–pressure interpolation. *Numerical Heat Transfer* 1985; **8**:259–280.
14. Schneider GE, Raw MJ. A skewed positive influence coefficient upwinding procedure for control volume-based finite-element convection–diffusion computation. *Numerical Heat Transfer* 1986; **9**:1–26.
15. Baliga BR, Patankar SV. A new finite-element formulation for convection–diffusion problems. *Numerical Heat Transfer* 1980; **3**:393–409.
16. Baliga BR, Patankar SV. *Hand Book of Numerical Heat Transfer*, Chapter 11. Wiley: New York, 1988; 421–461.
17. Saabas HJ Baliga BR. Co-located equal-order control-volume finite element method for multidimensional, incompressible, fluid flow-part I. *Numerical Heat Transfer* 1994; **26B**:21–32.
18. Prakash C. Examination of the upwind (donor-cell) formulation in control volume finite-element methods for fluid flow and heat transfer. *Numerical Heat Transfer* 1987; **11**:401–416.
19. Deng GB, Piquet J, Queutey P, Visonneau M. A new fully coupled solution of the Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1994; **19**:605–639.
20. Knoll DA, McHugh PR. A fully implicit direct Newton solver for Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1993; **17**:449–461.
21. Ghia U, Ghia KN, Shin CT. High-Re solution for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
22. Tritton DJ. Experiments on the flow past a circular cylinder at low Reynolds number. *Journal of Fluid Mechanics* 1959; **6**:547.
23. Fornberg B. A numerical study of steady viscous flow past a circular cylinder. *Journal of Fluid Mechanics* 1980; **98**:819.

