

A coupled finite volume solver for the solution of incompressible flows on unstructured grids

M. Darwish, I. Sraj, F. Moukalled *

Department of Mechanical Engineering, American University of Beirut, P.O. Box 11-0236, Riad El Solh, Beirut 1107 2020, Lebanon

ARTICLE INFO

Article history:

Received 7 March 2008

Received in revised form 19 August 2008

Accepted 27 August 2008

Available online 19 September 2008

Keywords:

Finite volume method

Pressure-based method

Coupled solver

Segregated solver

Acceleration techniques

ABSTRACT

This paper reports on a newly developed fully coupled pressure-based algorithm for the solution of laminar incompressible flow problems on collocated unstructured grids. The implicit pressure-velocity coupling is accomplished by deriving a pressure equation in a procedure similar to a segregated SIMPLE algorithm using the Rhie–Chow interpolation technique and assembling the coefficients of the momentum and continuity equations into one diagonally dominant matrix. The extended systems of continuity and momentum equations are solved simultaneously and their convergence is accelerated by using an algebraic multigrid solver. The performance of the coupled approach as compared to the segregated approach, exemplified by SIMPLE, is tested by solving five laminar flow problems using both methodologies and comparing their computational costs. Results indicate that the number of iterations needed by the coupled solver for the solution to converge to a desired level on both structured and unstructured meshes is grid independent. For relatively coarse meshes, the CPU time required by the coupled solver on structured grid is lower than the CPU time required on unstructured grid. On dense meshes however, this is no longer true. For low and moderate values of the grid aspect ratio, the number of iterations required by the coupled solver remains unchanged, while the computational cost slightly increases. For structured and unstructured grid systems, the required number of iterations is almost independent of the grid size at any value of the grid expansion ratio. Recorded CPU time values show that the coupled approach substantially reduces the computational cost as compared to the segregated approach with the reduction rate increasing as the grid size increases.

© 2008 Elsevier Inc. All rights reserved.

1. Introduction

At the heart of computational fluid dynamics (CFD) is the velocity-pressure coupling algorithm that drives the fluid flow simulations to convergence. Over the past decades efforts to develop more robust and efficient velocity-pressure algorithms have resulted in a better understanding of the numerical issues affecting the performance of these algorithms, such as the choice of primitive variables (density-based versus pressure-based [1]), the type of variable arrangement (staggered versus collocated [2]), and the kind of solution approach (coupled versus segregated), to cite a few. While consensus regarding best practices of many issues have been reached within the CFD community, for pressure-based algorithms the coupled versus segregated approach dichotomy has not been completely resolved. This was clearly indicated in a recent review of pressure-based algorithms for single and multiphase flow conducted by the authors [3], in which it was mentioned that even though the situation is currently in favor of the segregated approach, recent work seems to indicate that this might be changing.

* Corresponding author. Tel.: +961 1 347952; fax: +961 1 744462.

E-mail address: memouk@aub.edu.lb (F. Moukalled).

Nomenclature

$a_P^\phi, a_F^\phi, \dots$	coefficients in the discretized equation for ϕ
b_P^ϕ	source term in the discretized equation for ϕ
$\mathbf{d}_{PF}$	space vector joining the grid points P and F
$\mathbf{D}_P$	the matrix $\mathbf{D}$ operator
g	geometric interpolation factor
$H[\phi]$	the H operator
$\mathbf{H}[\mathbf{v}]$	the vector form of the H operator
$\mathbf{I}$	the identity matrix
$\dot{m}_f$	mass flow rate at control volume face f
p	pressure
P	main grid point
Q	general source term
$\mathbf{S}$	surface vector
u, v	velocity components in x- and y-directions, respectively
$\mathbf{v}$	velocity vector
Ω_P	volume of the P cell

Greek symbols

ϕ	general scalar quantity
ε	grid expansion ratio
Γ	diffusion coefficient
μ	dynamic viscosity
ρ	fluid density

Subscripts

f	refers to control volume face
F	refers to the F grid point
nb	refers to values at the faces obtained by interpolation between P and its neighbors
NB	refers to the neighbors of the P grid point
P	refers to the P grid point
x, y	refers to x and y directions

Superscripts

p	refers to pressure
u	refers to the u-velocity component
v	refers to the v-velocity component
x	refers to x-direction
y	refers to y-direction
.	refers to an interpolated value

This renewed interest in the development of coupled solvers [4,5] is due to the tremendous increase in computer memory and to the convergence problem experienced by segregated solvers when used with dense computational meshes [6]. Even though the convergence issue has been addressed successfully through multigrid, parallel processing, and domain decomposition the convergence issue has not been directly resolved. It is worthwhile in this respect to point out that density-based Euler methods have been using coupled solvers quite successfully for solving highly compressible flow problems. In the coupled approach, the conservation equations are discretized and solved as one system of equations as opposed to the segregated approach where the equation of each variable is solved separately using, previously computed, best estimate values of the other dependent variables. Although the coupled versus segregated issue is not directly related to the method used, traditionally it has been the case that pressure-based methods follow a segregated approach. This state of affairs owes more to the development history of pressure-based algorithms than to any algorithmic limitation.

Pressure-based algorithms originated with the work of Harlow and Welch [7] and Chorin [8]. However the real thrust to this group of algorithms was generated in the early 1970s by the CFD group at Imperial College through the development of the well-known segregated SIMPLE algorithm (semi-implicit method for pressure linked equations) [9] for incompressible flows. The CFD research community widely adopted the SIMPLE algorithm which led to the development of a number of SIMPLE-like algorithms, a review of which is reported in [10]. Furthermore, the work of Rhie and Chow [11] provided a solution to the checkerboard problem and expanded the application area of the SIMPLE-like algorithms by enabling the use of a collocated variable arrangement and setting the ground for a geometric flexibility similar to that of the finite element method

(FEM) [12]. Additional developments resulted in extending the applicability of the SIMPLE-like algorithms to a wide range of fluid physics such as compressible [1], free-surface [13], and multiphase flows [14,15].

Several pressure-based coupled algorithms have also been reported in the literature. These algorithms followed two approaches in their development. In the first, no pressure equation is introduced and the momentum and continuity equations are discretized in a straightforward manner. Examples of these algorithms include the SIVA (simultaneous variable arrangement) algorithm of Carretto et al. [16], the SCGS (symmetric coupled Gauss Seidel) algorithm of Vanka [17], the UVP method of Karki and Mongia [18], the method of Braaten [19], and more recently the BIP (Body Implicit Procedure) of Mazhar [20], among others. Since no pressure equation is derived, zeros are present in the main diagonal of the discretized continuity equation leading to an ill conditioned system of equations. This problem has been addressed, with various degrees of success, through the use of pre-conditioning [6], penalty formulations [21], or by algebraic manipulations [22].

In the second approach a pressure equation is derived either through the addition of pseudo-velocities [23] as in the segregated SIMPLER algorithm [24] or without the addition of new variables [25] as in the segregated SIMPLE algorithm [9]. Following either method, a set of diagonally dominant equations is obtained. Using the control volume finite element method (CVFEM), Lonsdale [26] and Webster [5] followed this approach and reported impressive convergence rates and good scaling behavior with dense meshes. However Lonsdale's algorithm did not prove to be robust [5].

In a recent article [27], the authors reported on a pressure-based coupled algorithm for the solution of incompressible flow problems over structured grid systems developed within the context of a finite volume formulation. Their results showed that, for the problems presented, the CPU time per control volume is nearly independent of the grid size. The objective of this work is to extend the method to unstructured grid topology and compare the performance of the coupled algorithm with the segregated SIMPLE algorithm for both structured and unstructured grids by solving a series of test problems showing the effects of grid size, grid non-uniformity, mesh skewness, large pressure gradients, and large source terms on the convergence rate. Extension into unstructured grid systems entails substantial changes to the algorithm (e.g. using different numbers of control volume faces depending on the type of elements that compose the mesh, connectivity of the grid, the algebraic equation solver, and the algebraic multigrid solver for convergence acceleration).

In the remainder of this article a brief description of the Finite Volume discretization process is presented, followed by a short review of the segregated algorithm. Then the formulation of the coupled algorithm, the most frequently encountered boundary conditions, the algebraic multigrid solver, and some implementation tips are detailed. Finally, the performance of the coupled algorithm is illustrated by solving several problems.

2. Finite volume discretization

The conservation equations governing steady, laminar incompressible Newtonian fluid flow are given by

$$\nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

$$\nabla \cdot (\rho \mathbf{v} \mathbf{v}) = \nabla \cdot (\mu \nabla \mathbf{v}) - \nabla \cdot (p \mathbf{I}) \quad (2)$$

These equations can be expressed in the general conservative form as

$$\nabla \cdot (\rho \mathbf{v} \phi) = \nabla \cdot (\Gamma \nabla \phi) + Q \quad (3)$$

where the values of ϕ and Γ differ depending on the equation represented.

Integrating the general transport equation over the control volume displayed in Fig. 1(a) and transforming the volume integrals of the diffusion and convection terms into surface integrals through the use of the divergence theorem, the semi-discretized form of the governing conservation equation using the finite volume method is obtained as

$$\oint_{\partial \Omega} (\rho \mathbf{v} \phi) \cdot d\mathbf{S} = \oint_{\partial \Omega} (\Gamma \nabla \phi) \cdot d\mathbf{S} + \iiint_{\Omega} Q d\Omega \quad (4)$$

Evaluating these integrals using a second order integration scheme yields

$$\sum_{f \in \text{nb}(P)} (\rho \mathbf{v} \phi - \Gamma \nabla \phi)_f \cdot \mathbf{S}_f = Q_P \Omega_P \quad (5)$$

Finally, the equation is expressed in algebraic form by representing the variables at the control volume faces in terms of nodal values. The resulting equation is written as

$$a_P^\phi \phi_P + \sum_{F \in \text{NB}(P)} a_F^\phi \phi_F = b_P^\phi \quad (6)$$

The above equation could equivalently be written as

$$\phi_P + \sum_{F \in \text{NB}(P)} \frac{a_F^\phi}{a_P^\phi} \phi_F = \frac{b_P^\phi}{a_P^\phi} \quad \text{or} \quad \phi_P + \sum_{F \in \text{NB}(P)} A_F^\phi \phi_F = B_P^\phi \quad (7)$$

while for the continuity equation the following discrete form is used:

$$\sum_{f=nb(P)} \dot{m}_f = 0 \text{ with } \dot{m}_f = \rho \mathbf{v}_f \cdot \mathbf{S}_f \quad (9)$$

3. The collocated SIMPLE algorithm

In the segregated SIMPLE algorithm, the solution is obtained by iteratively solving the momentum equations and a pressure correction equation, derived from the continuity equation, while accounting for the effects of the pressure field on the momentum equations through a correction to the velocity field. Denoting corrections with a prime, the corrected fields are written as

$$p = p^{(n)} + p' \text{ and } \mathbf{v} = \mathbf{v}^* + \mathbf{v}' \quad (10)$$

where p' and $\mathbf{v}'$ are the pressure and velocity corrections, respectively. Thus, before the pressure field is known, the velocity obtained from the solution of the momentum equation is actually $\mathbf{v}^*$ rather than $\mathbf{v}$. Hence the equations to be solved are

$$\mathbf{v}_p^* + \sum_{F=NB(P)} \mathbf{A}_F^v \mathbf{v}_F^* = \mathbf{B}_p^v - \mathbf{D}_p \nabla p_p^{(n)} \quad (11)$$

where the final solution satisfies

$$\mathbf{v}_p + \sum_{F=NB(P)} \mathbf{A}_F^v \mathbf{v}_F = \mathbf{B}_p^v - \mathbf{D}_p \nabla p_p \quad (12)$$

Subtracting the two sets of Eqs. (12) and (11) from each other yields the following equation involving the correction terms:

$$\mathbf{v}_p' + \sum_{F=NB(P)} \mathbf{A}_F^v \mathbf{v}_F' = -\mathbf{D}_p \nabla p_p' \quad (13)$$

Using the Rhie–Chow interpolation [11], the velocity correction along the control volume face, is written as

$$\mathbf{v}_f' = \overline{\mathbf{v}_f'} - \overline{\mathbf{D}_f} (\nabla p_f' - \overline{\nabla p_f'}) = -\overline{\mathbf{D}_f} \nabla p_f' + \overline{\mathbf{v}_f'} + \overline{\mathbf{D}_f} \nabla p_f' = -\overline{\mathbf{D}_f} \nabla p_f' - \overline{\sum_{f=nb(P)} \mathbf{A}_f^v \mathbf{v}_f'} \quad (14)$$

To derive the pressure correction equation, the following expanded form of the continuity equation [Eq. (9)] is used:

$$\sum_{f=nb(P)} \dot{m}_f^* + \sum_{f=nb(P)} (\rho \mathbf{v}')_f \cdot \mathbf{S}_f = 0 \quad (15)$$

By substituting $\mathbf{v}_f'$ from Eq. (14) into the continuity equation (Eq. (15)), the pressure correction equation is obtained as

$$\sum_{f=nb(P)} (-\rho_f \overline{\mathbf{D}_f} \nabla p_f' \cdot \mathbf{S}_f) = - \sum_{f=nb(P)} \dot{m}_f^* + \overline{\sum_{f=nb(P)} \rho_f \mathbf{A}_f^v \mathbf{v}_f' \cdot \mathbf{S}_f} \quad (16)$$

Neglecting the last term in Eq. (16) as done in SIMPLE [9], the algebraic form of the pressure correction equation is written as

$$\begin{aligned} a_p^{p'} p_p' + \sum_{F=NB(P)} a_F^{p'} p_F' &= b_p^{p'} \\ a_F^{p'} &= \rho_f \frac{(\overline{\mathbf{D}_f} \mathbf{S}_f) \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \\ a_p^{p'} &= \sum_{F=NB(P)} a_F^{p'} \\ b_p^{p'} &= - \sum_{f=nb(P)} \dot{m}_f^* \end{aligned} \quad (17)$$

Using the Rhie–Chow interpolation, the mass flow rate $\dot{m}_f^*$ at a control volume face is computed from

$$\dot{m}_f^* = \rho_f \mathbf{v}_f^* \cdot \mathbf{S}_f = \rho_f \overline{\mathbf{v}_f^*} \cdot \mathbf{S}_f - \rho_f \overline{\mathbf{D}_f} (\nabla p_f^{(n)} - \overline{\nabla p_f^{(n)}}) \cdot \mathbf{S}_f \quad (18)$$

Moreover, neglecting the last term in Eq. (14), the velocity correction is found to be

$$\mathbf{v}_f' = -\overline{\mathbf{D}_f} \nabla p_f' \quad (19)$$

The overall SIMPLE algorithm can be summarized as follows:

1. Solve the momentum equations implicitly for $\mathbf{v}$ using the available pressure field.
2. Calculate the $\mathbf{D}$ field.
3. Solve the pressure correction equation.
4. Correct $\mathbf{v}$ and p .
5. Solve sequentially all other scalar equations (if any).
6. Return to the first step and repeat until convergence.

4. The coupled algorithm

The convergence of the SIMPLE algorithm is highly affected by the explicit treatment of the pressure gradient in the momentum equation and the velocity field in the continuity equation. Treating both terms in an implicit manner is in essence the aim of any coupled algorithm. This is achieved here by coupling the momentum equation and the pressure equation form of the continuity equation through a set of coefficients that represent the mutual influence of the continuity and momentum equations on the pressure and the velocity fields, as described below.

Starting with the semi-discretized momentum equation given by

$$\sum_{f=\text{nb}(P)} (\rho \mathbf{v} \mathbf{v} - \mu \nabla \mathbf{v})_f \cdot \mathbf{S}_f + \sum_{f=\text{nb}(P)} p_f \mathbf{S}_f = \mathbf{b}_p \Omega_P \quad (20)$$

where the pressure gradient term has been integrated over the faces of the control volume and the pressure at each face is evaluated using

$$p_f = g_f p_P + (1 - g_f) p_F \quad (21)$$

substituting Eq. (21) into Eq. (20), and manipulating, the final form of the discretized momentum equations is obtained as

$$\begin{cases} \underline{a_P^{uu}} u_P + \underline{a_P^{uv}} v_P + \underline{a_P^{up}} p_P + \sum_{F=\text{NB}(P)} \underline{a_F^{uu}} u_F + \sum_{F=\text{NB}(P)} \underline{a_F^{uv}} v_F + \sum_{F=\text{NB}(P)} \underline{a_F^{up}} p_F = b_P^u \\ \underline{a_P^{vv}} v_P + \underline{a_P^{vu}} u_P + \underline{a_P^{vp}} p_P + \sum_{F=\text{NB}(P)} \underline{a_F^{vv}} v_F + \sum_{F=\text{NB}(P)} \underline{a_F^{vu}} u_F + \sum_{F=\text{NB}(P)} \underline{a_F^{vp}} p_F = b_P^v \end{cases} \quad (22)$$

where the coefficients are given by

$$\begin{aligned} a_F^{uu} &= a_F^{vv} = \mu_f \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} + \|\dot{m}_f, 0\| \\ a_P^{uu} &= \sum_{F=\text{NB}(P)} a_F^{uu} & a_P^{vv} &= \sum_{F=\text{NB}(P)} a_F^{vv} \\ a_F^{uv} &= 0 & a_F^{vu} &= 0 \\ a_P^{uv} &= \sum_{F=\text{NB}(P)} a_F^{uv} & a_P^{vu} &= \sum_{F=\text{NB}(P)} a_F^{vu} \\ a_F^{up} &= (1 - g_f) S_f^x & a_F^{vp} &= (1 - g_f) S_f^y \\ a_P^{up} &= \sum_{f=\text{nb}(P)} g_f S_f^x & a_P^{vp} &= \sum_{f=\text{nb}(P)} g_f S_f^y \\ b_P^u &= \sum_{f=\text{nb}(P)} \left[\nabla u \cdot (\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF}) \right] & b_P^v &= \sum_{f=\text{nb}(P)} \left[\nabla v \cdot (\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF}) \right] \end{aligned} \quad (23)$$

It should be noted that the single underlined terms in Eq. (22) represent the pressure gradient in its implicit form; while the double underlined terms account for the velocity component interactions with their values being zero except at wall boundaries. Even though their values are set at zero in Eq. (23), their inclusion is necessary for the proper implementation of the algebraic solver.

To derive the pressure equation, the semi-discretized form of the continuity equation, given by

$$\sum_{f=\text{nb}(P)} \rho_f \mathbf{v}_f \cdot \mathbf{S}_f = 0 \quad (24)$$

is combined with the Rhie–Chow interpolation to yield

$$\sum_{f=\text{nb}(P)} \rho_f [\bar{\mathbf{v}}_f - \bar{\mathbf{D}}_f (\nabla p_f - \nabla \bar{p}_f)] \cdot \mathbf{S}_f = 0 \quad (25)$$

Eq. (25) can be expanded into

$$\sum_{f=nb(P)} \rho_f (-\bar{\mathbf{D}}_f \nabla p_f) \cdot \mathbf{S}_f + \sum_{f=nb(P)} \rho_f \bar{\mathbf{v}}_f \cdot \mathbf{S}_f = \sum_{f=nb(P)} \rho_f (-\bar{\mathbf{D}}_f \nabla p_f) \cdot \mathbf{S}_f \quad (26)$$

where

$$\bar{\mathbf{v}}_f = g_f \mathbf{v}_P + (1 - g_f) \mathbf{v}_F \quad (27)$$

Substituting Eq. (27) into Eq. (26), the algebraic form of the pressure equation is obtained as

$$a_P^{pp} p_P + a_P^{pu} u_P + a_P^{pv} v_P + \sum_{F=NB(P)} a_F^{pp} p_F + \sum_{F=NB(P)} a_F^{pu} u_F + \sum_{F=NB(P)} a_F^{pv} v_F = b_P^p \quad (28)$$

with the coefficients evaluated as

$$\begin{aligned} a_F^{pp} &= \rho_f \frac{(\bar{\mathbf{D}}_f \mathbf{S}_f) \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \\ a_P^{pp} &= \sum_{F=NB(P)} a_F^{pp} \\ a_F^{pu} &= (1 - g_f) S_f^x \quad a_F^{pv} = (1 - g_f) S_f^y \\ a_P^{pu} &= \sum_{f=nb(P)} g_f S_f^x \quad a_P^{pv} = \sum_{f=nb(P)} g_f S_f^y \\ b_P^p &= \sum_{f=nb(P)} \rho_f (-\bar{\mathbf{D}}_f \nabla p_f) \cdot \mathbf{S}_f - \sum_{f=nb(P)} \rho_f (-\bar{\mathbf{D}}_f \nabla p_f) \cdot \left(\mathbf{S}_f - \frac{\mathbf{S}_f \cdot \mathbf{S}_f}{\mathbf{S}_f \cdot \mathbf{d}_{PF}} \mathbf{d}_{PF} \right) \end{aligned} \quad (29)$$

Combining the discretized momentum and continuity equations [Eqs. (22) and (28)], the following system of equations is obtained for each control volume:

$$\begin{bmatrix} a_P^{uu} & a_P^{uv} & a_P^{up} \\ a_P^{vu} & a_P^{vv} & a_P^{vp} \\ a_P^{pu} & a_P^{pv} & a_P^{pp} \end{bmatrix} \begin{bmatrix} u_P \\ v_P \\ p_P \end{bmatrix} + \sum_{F=NP(P)} \begin{bmatrix} a_F^{uu} & a_F^{uv} & a_F^{up} \\ a_F^{vu} & a_F^{vv} & a_F^{vp} \\ a_F^{pu} & a_F^{pv} & a_F^{pp} \end{bmatrix} \begin{bmatrix} u_F \\ v_F \\ p_F \end{bmatrix} = \begin{bmatrix} b_P^u \\ b_P^v \\ b_P^p \end{bmatrix} \quad (30)$$

The above set of equations expressed over the entire computational domain yields a system of equations in the form of

$$\mathbf{A}\Phi = \mathbf{B} \quad (31)$$

where all variables ($\mathbf{v}$, p) are now solved simultaneously. Note that the continuity equation is now written in terms of pressure rather than pressure correction.

The overall coupled algorithm can be summarized as follows:

1. Start with the latest available values ($\hat{m}_f^{(n)}, \mathbf{v}^{(n)}, p^{(n)}$).
2. Assemble and solve the momentum and continuity equation for $\mathbf{v}^*$ and p^* .
3. Assemble $\hat{m}_f$ using the Rhie–Chow interpolation.
4. Solve sequentially all other scalar equations (if any).
5. Return to step 2 and repeat until convergence.

5. Boundary conditions

The proper treatment of the boundary conditions is critical to the success of the proposed algorithm because of the coupling between the governing equations. The contribution of a boundary face to the algebraic equation of the control volume concerned depends on the type of the boundary condition. The details for implementing the most frequently encountered boundary conditions at a wall, inlet, and outlet are given next.

5.1. The no-slip boundary condition at a moving wall

The general case of a wall moving with a velocity $\mathbf{v}_w (= u_w \mathbf{i} + v_w \mathbf{j})$ is considered; the special case of a stationary wall is obtained by setting $\mathbf{v}_w$ to zero. The convection term has no effect on the momentum equation because no flow crosses the wall. The shear stress is accounted for using the method described next.

The velocity vector at the first interior grid point (Fig. 1(b)), designated by $\mathbf{v}_p (= u_p \mathbf{i} + v_p \mathbf{j})$, is decomposed into two vectors one tangential ($\mathbf{v}_t$) and one normal ($\mathbf{v}_n$) to the wall. The wall velocity ($\mathbf{v}_w$) being in the tangential direction, the wall shear stress can be calculated as

$$\tau_w = \mu \frac{\mathbf{v}_t - \mathbf{v}_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} \quad (32)$$

where $\mathbf{d}_{pw}$ is the distance vector between the internal and boundary grid point, $\mathbf{n}_w$ is the outward unit vector normal to the wall ($\mathbf{n}_w = n_{w,x}\mathbf{i} + n_{w,y}\mathbf{j} = \mathbf{S}_w/S_w$), and $(\mathbf{d}_{pw} \cdot \mathbf{n}_w)$ is the normal distance to the wall. Then the shear force $\mathbf{F}_s$ is given by

$$\mathbf{F}_s = -\tau_w \mathbf{S}_w \quad (33)$$

while the tangential velocity $\mathbf{v}_t$ is computed from

$$\mathbf{v}_t = \mathbf{v}_p - (\mathbf{v}_p \cdot \mathbf{n}_w) \mathbf{n}_w \quad (34)$$

Combining Eqs. (32) and (34), the expanded form of the shear force is written as

$$\mathbf{F}_s = \begin{bmatrix} F_{s,x} \\ F_{s,y} \end{bmatrix} = -\frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} \begin{bmatrix} u_p(1 - n_{w,x}^2) - v_p n_{w,x} n_{w,y} - u_w \\ v_p(1 - n_{w,y}^2) - u_p n_{w,x} n_{w,y} - v_w \end{bmatrix} \quad (35)$$

The contribution of the wall shear stress can now be incorporated into the coefficients to obtain

$$\begin{aligned} a_p^{uu} &= a_p^{uu} + \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} (1 - n_{w,x}^2) & a_p^{vv} &= a_p^{vv} + \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} (1 - n_{w,y}^2) \\ a_p^{uv} &= a_p^{uv} - \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} n_{w,x} n_{w,y} & a_p^{vu} &= a_p^{vu} - \frac{\mu S_w}{\mathbf{d}_{pw} \cdot \mathbf{n}_w} n_{w,x} n_{w,y} \end{aligned} \quad (36)$$

Further, the pressure at the wall is extrapolated from the pressure at the main grid point using a zero order profile to yield

$$p_w = p_p \quad (37)$$

and its contribution to the momentum equations is therefore written as

$$\begin{aligned} a_p^{up} &= a_p^{up} + S_{w,x} \\ a_p^{vp} &= a_p^{vp} + S_{w,y} \end{aligned} \quad (38)$$

Because at a wall the mass flow rate is zero, no modification is needed for the pressure equation so its coefficients remain unchanged.

5.2. Inlet boundary condition

For an incompressible flow, either the velocity or the pressure can be specified at inlet. Both cases are presented next.

5.2.1. Specified static pressure

The flux at an interior control volume face is a function of the two control volumes straddling the face, while at a boundary face the flux becomes a function of the control volume and the boundary face itself. When the value of the dependent variable is specified, the corresponding boundary flux can be computed and moved to the source term. In a case with a specified static pressure at the inlet the pressure is known. However the velocity, being unknown, has to be interpolated. In addition, the velocity direction should be specified, because it cannot be predicted. By splitting the surface vector into two components $\mathbf{E}$ and $\mathbf{T}$ (i.e. $\mathbf{S} = \mathbf{E} + \mathbf{T}$), with $\mathbf{E}$ being aligned with the distance vector and $\mathbf{T}$ normal to the $\mathbf{S}$ vector (Fig. 1(c)), the modified coefficients of the momentum equations at the inlet boundary are written as

$$\begin{aligned} a_p^{uu} &= a_p^{uu} + \|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} & b_p^u &= b_p^u + (\mu \nabla u \cdot \mathbf{T})_i + \left[\|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \right] u_i - p_i S_{i,x} \\ a_p^{vv} &= a_p^{vv} + \|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} & b_p^v &= b_p^v + (\mu \nabla v \cdot \mathbf{T})_i + \left[\|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \right] v_i - p_i S_{i,y} \end{aligned} \quad (39)$$

For the pressure equation the velocity is extrapolated from the nearest control volume and the pressure gradient term is computed with the known inlet pressure term considered explicitly. The modified coefficients of the pressure equation are given by

$$\begin{aligned} a_p^{up} &= a_p^{up} + \rho S_{i,x} \\ a_p^{vp} &= a_p^{vp} + \rho S_{i,y} \\ a_p^{pp} &= a_p^{pp} + \rho_i \frac{(\overline{\mathbf{D}_i \mathbf{S}_i}) \cdot \mathbf{S}_i}{\mathbf{S}_i \cdot \mathbf{d}_{pi}} \\ b_p^p &= \overline{D_i \nabla p_i^*} \cdot \mathbf{T}_i + \rho_i \frac{(\overline{\mathbf{D}_i \mathbf{S}_i}) \cdot \mathbf{S}_i}{\mathbf{S}_i \cdot \mathbf{d}_{pi}} p_i - \overline{D_i \nabla p_i^*} \cdot \mathbf{S}_i \end{aligned} \quad (40)$$

5.2.2. Specified velocity

Because the velocity is known at inlet, the convection term can be treated explicitly. The contribution of the stress term affects the coefficient of the interior control volume, as well as the boundary itself, and a source term appears. For the pres-

sure gradient term in the momentum equations, the pressure is extrapolated from the interior, as in the case of a wall. The set of coefficients for the momentum equations at the inlet are modified as

$$\begin{aligned} a_p^{uu} &= a_p^{uu} + \|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} & a_p^{vv} &= a_p^{vv} + \|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \\ b_p^u &= b_p^u + (\mu \nabla u \cdot \mathbf{T})_i + \left[\|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \right] u_i & a_p^{up} &= a_p^{up} + S_{i,x} \\ b_p^v &= b_p^v + (\mu \nabla v \cdot \mathbf{T})_i + \left[\|\dot{m}_i, 0\| + \mu \frac{\mathbf{S}_i \cdot \mathbf{S}_i}{\mathbf{d}_{pi} \cdot \mathbf{S}_i} \right] v_i & a_p^{vp} &= a_p^{vp} + S_{i,y} \end{aligned} \quad (41)$$

Because the flow is incompressible, the mass flow rate at inlet is known. Therefore no modification is needed to the pressure equation and its coefficients remain unchanged.

5.3. Outlet boundary condition

A common boundary condition at an outlet is a specified value for the static pressure. This boundary condition is similar to the specified static pressure at an inlet boundary condition. The modified coefficients are those given by Eqs. (39) and (40).

6. Linear multigrid solver

Many methods exist for the solution of large systems of linear equations and these can be categorized as being either direct or indirect iterative methods. The use of a direct method is not appropriate in the present context because direct methods require by far more storage than iterative methods and are usually more time consuming. This is further magnified by the non-linearity encountered in fluid flow calculations. The algorithm used in this work is a combination of the ILU(0) [28] algorithm with an additive corrective multigrid method [29]. Surprisingly, this combination is found to provide the simplicity and low storage needs of the basic ILU algorithm with the high convergence rate of multigrid methods.

The simplest form of incomplete factorization is based on taking a subset Σ of nonzero elements from the original coefficient matrix $\mathbf{A}$ while keeping all positions outside this set equal to zero. If Σ is chosen to coincide with the non zero elements of $\mathbf{A}$, then the factorization is called the *ILU(0)* [28]. For the *ILU(0)* method, the factorization does not produce any non zero elements beyond the sparsity of $\mathbf{A}$ so that the pre-conditioner requires at worst as much storage as $\mathbf{A}$. To remedy the deterioration of the convergence rate with increasing mesh size, the *ILU(0)* is used as a smoother for an algebraic multigrid solver.

Multigrid algorithms were independently introduced by Federenko [30] (Geometric Multigrid) and Poussin [31] (Algebraic Multigrid) in the 60s, and later gained popularity with the work of Brandt [32]. They are considered one of the most efficient techniques for the numerical solution of PDEs, at least for sequential computers. While standard iterative solvers (e.g. SOR and ILU) are efficient in removing high frequency errors, they are inefficient in removing the remaining low frequency or smooth errors. Multigrid methods overcome the decay in the convergence rate by using a hierarchy of coarse grids in addition to the one on which the solution is sought. The fundamental idea is that by restricting the problem to a coarser grid, the lower frequency errors now appear more oscillatory.

Without going into details, the implementation of a multigrid method involves two stages. In the first stage, the coarse grids and their connectivity are setup using an agglomeration or coarsening algorithm [33]. In the second stage, a multigrid cycling procedure is used with a smoother to yield the solution at the finest desired grid. All segregated and coupled results presented in this work are generated using an algebraic multigrid with an *ILU(0)* solver as a smoother.

7. Efficient implementation of the coupled solver

In addition to an appropriate fully implicit discretization of the Navier–Stokes' equations, the performance of the coupled algorithm is critically dependent on the proper implementation of an iterative solver to ensure that the increase in computational time incurred in the solution of the enlarged system of equations does not counter balance the advantage of the higher convergence rate. In one-component systems, the coefficients represent the influences between neighboring elements, i.e. spatial influences. For a coupled system, in addition to the spatial connectivity, inter-component connections arise. This renders the use of the algebraic multigrid iterative solver described above unsuitable. To circumvent this hurdle and efficiently employ the one-component algebraic multigrid algorithm to solving the coupled system, the original spatial connectivity array describing the topology of the mesh is retained for use in the agglomeration procedure of the multigrid algorithm, while an expanded connectivity array that accounts for the inter-variable influences is constructed for the iterative solver as described below.

The algebraic system of equations for a single variable φ , over a computational domain of size n , has the following form:

$$\begin{bmatrix} a_{11} & \cdot & \cdot & a_{1n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{n1} & & & a_{nn} \end{bmatrix} \begin{bmatrix} \varphi_1 \\ \cdot \\ \cdot \\ \varphi_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ b_n \end{bmatrix} \quad (42)$$

where $[a]$ is an $n \times n$ matrix, and $[\phi]$ and $[b]$ are vectors of size n (n being the number of elements in the computational mesh). The two vector $[\phi]$ and $[b]$ are stored in arrays of size n , while the sparse matrix $[a]$ is usually stored in two vectors $[a_p]$ containing the diagonal elements represented by the array $a_p(i)$ of size n and $[a_{nb}]$ containing the neighboring coefficients. The size of $[a_{nb}]$ depends on the number of neighbors associated with the mesh elements and is equal to the sum over all the mesh elements of the number of neighboring elements (i.e. $\text{sizeof}[a_{nb}] = \sum_{i=1}^n n_b(i)$). Access to an element of $[a_{nb}]$ is done via an offset array, as depicted in Fig. 2, whereby the coefficients of the neighbors of the control volume i are stored in $\text{coeff}[\text{offset}(i)]$ to $\text{coeff}[\text{offset}(i) - 1]$.

For a coupled algorithm the equations to be solved can be written for the case of a three component system in the following form:

$$\begin{bmatrix} \begin{bmatrix} a_{11}^{11} & a_{11}^{12} & a_{11}^{13} \\ a_{11}^{21} & a_{11}^{22} & a_{11}^{23} \\ a_{11}^{31} & a_{11}^{32} & a_{11}^{33} \end{bmatrix} & \dots & \begin{bmatrix} a_{1n}^{11} & a_{1n}^{12} & a_{1n}^{13} \\ a_{1n}^{21} & a_{1n}^{22} & a_{1n}^{23} \\ a_{1n}^{31} & a_{1n}^{32} & a_{1n}^{33} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} a_{n1}^{11} & a_{n1}^{12} & a_{n1}^{13} \\ a_{n1}^{21} & a_{n1}^{22} & a_{n1}^{23} \\ a_{n1}^{31} & a_{n1}^{32} & a_{n1}^{33} \end{bmatrix} & \dots & \begin{bmatrix} a_{nn}^{11} & a_{nn}^{12} & a_{nn}^{13} \\ a_{nn}^{21} & a_{nn}^{22} & a_{nn}^{23} \\ a_{nn}^{31} & a_{nn}^{32} & a_{nn}^{33} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \phi_1^1 \\ \phi_1^2 \\ \phi_1^3 \\ \vdots \\ \phi_n^1 \\ \phi_n^2 \\ \phi_n^3 \end{bmatrix} = \begin{bmatrix} b_1^1 \\ b_1^2 \\ b_1^3 \\ \vdots \\ b_n^1 \\ b_n^2 \\ b_n^3 \end{bmatrix} \quad (43)$$

From this perspective $[a]$ is now an $n \times n$ matrix of sub-matrices of size $n_c \times n_c$ (n_c being the number of components), while $[\phi]$ and $[b]$ are arrays of vectors of size n_c . The sparse matrix $[a]$ of the multi-component system, which now accounts for the inter-component influence in addition to the spatial influence between the elements, is again decomposed into the two vectors $[a_p]$ containing the diagonal elements represented by the array $a_p(i)$ of sub-matrices of size n and $[a_{nb}]$ containing the neighboring sub-matrix coefficients. The number of connections for a control volume i in this case becomes $n_c \times n_c \times n_b(i)$ and the size of $[a_{nb}]$ becomes $n_c \times n_c \times \sum_{i=1}^n n_b(i)$. The storage of a_p , b_p , and the neighboring coefficients is shown in Fig. 3(a). To solve this system using the standard iterative solver, the sub-matrices are first unraveled and transformed into an $N \times N$ system of scalar equations ($N = n \times n_c$) through the formulation of an extended connectivity matrix. The process followed in constructing the coupled connectivity matrix is explained next by referring to the element and its neighbors displayed in Fig. 3(b and c).

As shown in Fig. 3(b), the chosen element (element 5) is connected to elements 2, 3 and 7. The original spatial or geometric connectivity matrix is summarized in the upper part of Table 1, which displays the indices for a_p , a_{nb} , and b_p . These indices are suitable for solving a one-component system, which is the case for a segregated solution algorithm. For a coupled system, each coefficient is transformed into a 3×3 matrix (Fig. 3(c)). The connectivity is maintained by renumbering the elements of the matrix $[a]$ according to $(i \times n_c + 1, i \times n_c + 2, i \times n_c + 3)$ for the three components, where i is the element number under consideration (5 in this case) and the 1, 2, and 3 refers to the component u, v, and p, respectively. In a similar manner, the elements of the vector $[b]$ are renumbered as $(i \times n_c + i_c)$, where i_c refers to the component number (1 for u, 2 for v, and 3 for p). The connectivity for the $[a_{nb}]$ coefficients is now given by $(N(i_c) \times n_c + i_c)$, ($N(i_c)$ being the value in the old connectivity of the chosen element (element 5 in this case) and i_c the component under consideration. The connectivity arrays obtained by applying the above relations are depicted in Table 1. With this approach, the original algebraic multigrid solver is used with minor modifications.

8. Results and discussion

The performance of the coupled algorithm is assessed in this section by presenting solutions to the following five laminar incompressible fluid flow problems: (i) lid-driven flow in a rectangular and a skew cavity, (ii) flow behind a backward facing step, (iii) sudden expansion in a rectangular cavity, (iv) flow in a Planar Tee-Junction, and (v) natural convection in a trapezoidal cavity. For all problems, results are generated using both triangular and quadrilateral control volumes on three grid sizes with cell values of 10^4 , 5×10^4 , and 3×10^5 . The largest grid used was limited by the computational resources available and not because of any algorithmic limitation. The same initial guess was used for all grid sizes and for both coupled and segregated methods and the computations were stopped when the maximum residual of all variables, defined as,

$$\left\{ \begin{array}{l} (\text{RES})^\phi = \max_{i=1}^N \frac{|a_p^\phi \phi_p + \sum_{F \in \text{NB}(P)} a_F^\phi \phi_F - b_p^\phi|}{a_p^\phi \phi_{\text{scale}}} \\ \text{where} \\ \phi_{\text{scale}} = \max(\phi_{p,\text{max}} - \phi_{p,\text{min}}, \phi_{p,\text{max}}) \quad \phi_{p,\text{max}} = \max_{i=1}^N (\phi_p) \quad \phi_{p,\text{min}} = \min_{i=1}^N (\phi_p) \end{array} \right. \quad (44)$$

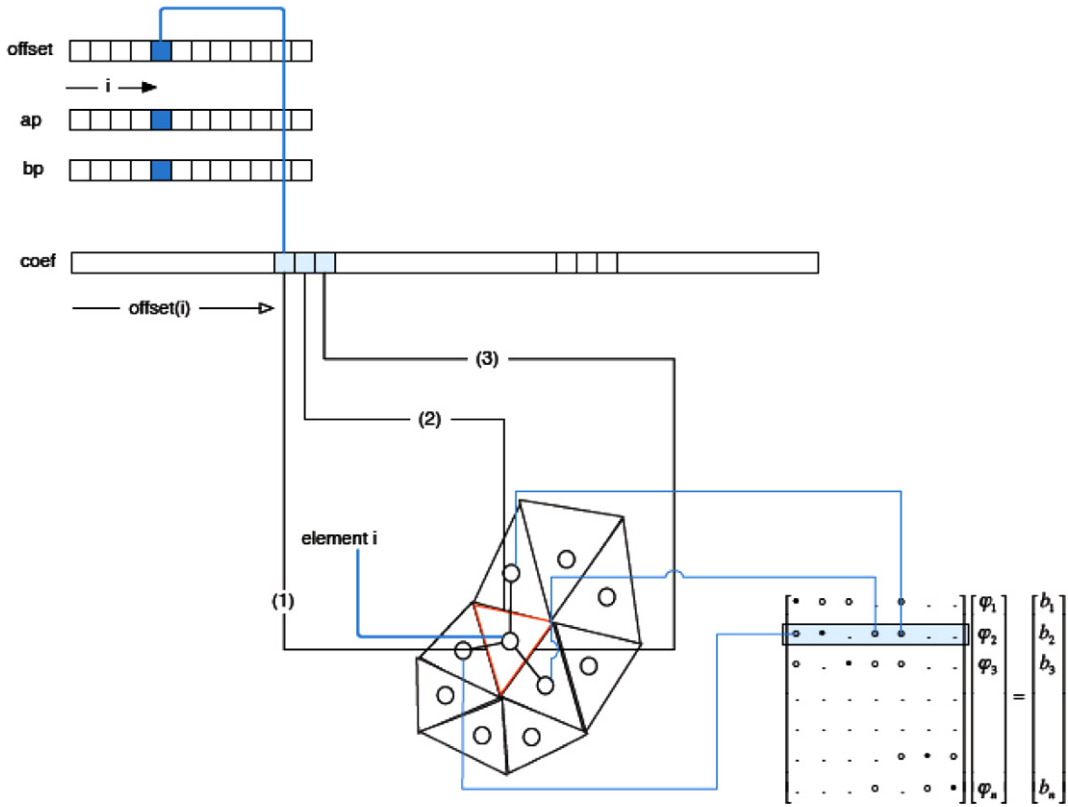


Fig. 2. Storage of coefficients for a single component system of equations.

became smaller than a vanishing quantity, which was set at 10^{-5} . All computations were performed on a “MacBook Pro” computer with a 2.16 GHz Intel Core Duo processor and 2 GB of RAM.

All problems were solved using both the coupled and segregated approach and the efficiency of the proposed coupled algorithm is demonstrated by comparing the number of iterations and CPU time required by each method on the various grids. No under relaxation was used with the coupled approach but it was needed to obtain converged solutions with the segregated method ($\alpha_u = \alpha_v = 0.7$ and $\alpha_p = 0.3$).

8.1. Comparison of solutions generated using the coupled and segregated solvers

The physical situations for the various problems solved, along with illustrative portions of the quadrilateral and triangular meshes used are depicted in Fig. 4. The first problem considered, which involves two configurations, is the standard CFD test case of lid-driven flow in a square (Fig. 4(a)) and a skew (Fig. 4(b)) cavity. It is used here to check the performance of the coupled approach in predicting recirculating flows on orthogonal and non-orthogonal unstructured grids. The second problem (Fig. 4(c)) is concerned with separated flows behind steps, which arise in many applications such as in electronic equipment and combustors and is used here to check the effect of a high-pressure gradient on the performance of the coupled approach. The third problem, depicted in Fig. 4(d), represents a sudden expansion of a flow entering a square cavity with a side of L from a vertical section with a width of $W = L/5$ located in the lower left corner of the domain. The problem is solved for a value of Reynolds number ($Re = \rho v_{in} L / \mu$) of 1000 with the velocity vector at the inlet set at $\mathbf{v}_{in}(1, 1)$. The geometry and boundary conditions of the fourth problem, which deals with the flow split in a Planar Tee-Junction (Fig. 4(e)), are those used by Hayes et al. [34] with the gauge pressure at the outlets set to zero. The flow enters the domain from its lower part moving vertically upward with a parabolic velocity profile of $\mathbf{v}(0, 4x - 4x^2)$. The problem is solved for a Reynolds number value ($Re = \rho V_c W / \mu$, V_c is the centerline velocity at inlet) of 500. The width of the domain W is set at 1 m and the length L at 3 m. The buoyancy-driven flow in a trapezoidal cavity problem, illustrated schematically in Fig. 4(f), is the one analyzed by Moukalled and Darwish [35] and is used here to check the performance of the new algorithm for sequentially solving the energy equation with the coupled hydrodynamic equations in the presence of a large source term on non-orthogonal unstructured grids.

Solutions for the various problems are generated using the coupled and segregated solvers by assuming the flow to be steady, laminar, and two-dimensional and the resulting flow fields in the domains are visualized by the streamline maps

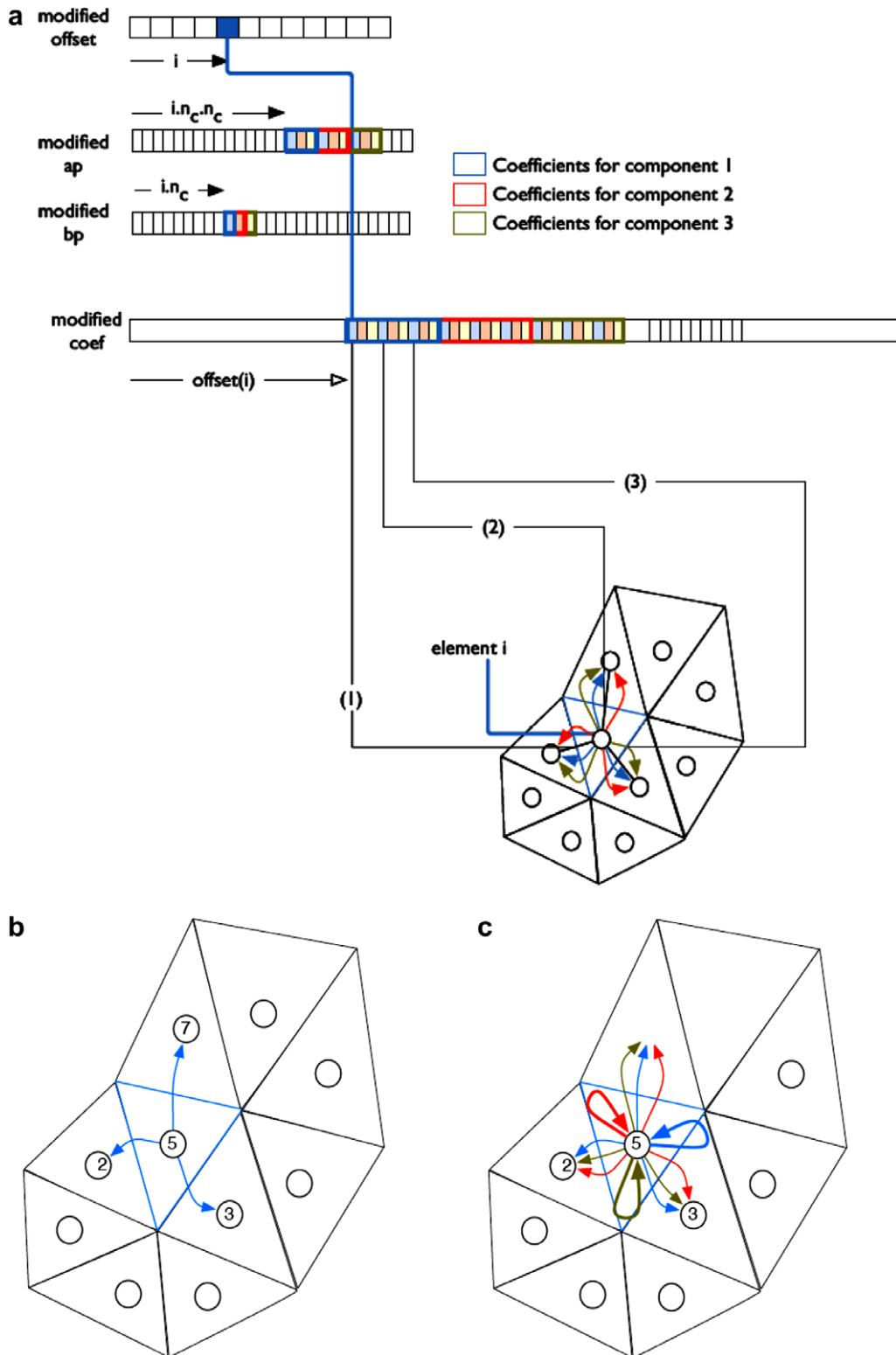


Fig. 3. (a) Storage vectors for a coupled system with three variables, (b) single component connectivity, and (c) multi-component connectivity.

presented in Fig. 5(a–f). Differences between the segregated and coupled solutions can be inferred from the u - and v -velocity contours displayed in Figs. 6 and 7, respectively. As shown, the two sets of contours are on top of each other, indicating that both solvers produce the same solution.

Table 1

Example of geometric and multi-component connectivity

Geometric connectivity						
Element		Coefficients				
5		a_p	a_{N1}	a_{N2}	a_{N3}	b_p
5		Connectivity	2	3	7	5
Multi-component connectivity						
Element/scalar		Coefficients				
5	1	$a_p^{11} a_p^{12} a_p^{13}$	$a_{N1}^{11} a_{N1}^{12} a_{N1}^{13}$	$a_{N2}^{11} a_{N2}^{12} a_{N2}^{13}$	$a_{N3}^{11} a_{N3}^{12} a_{N3}^{13}$	b_p^1
5	2	$a_p^{21} a_p^{22} a_p^{23}$	$a_{N1}^{21} a_{N1}^{22} a_{N1}^{23}$	$a_{N2}^{21} a_{N2}^{22} a_{N2}^{23}$	$a_{N3}^{21} a_{N3}^{22} a_{N3}^{23}$	b_p^2
5	3	$a_p^{31} a_p^{32} a_p^{33}$	$a_{N1}^{31} a_{N1}^{32} a_{N1}^{33}$	$a_{N2}^{31} a_{N2}^{32} a_{N2}^{33}$	$a_{N3}^{31} a_{N3}^{32} a_{N3}^{33}$	b_p^3
Element/scalar		Connectivity (scalar, $i_v = 1, 2, 3$)				
		$i^*3 + i_v$	$N1^*3 + i_v$	$N2^*3 + i_v$	$N3^*3 + i_v$	$i^*3 + i_v$
5	1	16, 17, 18	7, 8, 9	10, 11, 12	22, 23, 24	16
5	2	16, 17, 18	7, 8, 9	10, 11, 12	22, 23, 24	17
5	3	16, 17, 18	7, 8, 9	10, 11, 12	22, 23, 24	18

As a further validation check, pressure and velocity profiles along the vertical centerline of the main channel and the centerline of the horizontal branch for the Tee-Junction problem generated using both solvers are compared and the results are presented in Fig. 8(a–d). As shown, the profiles fall almost on top of each other confirming the correctness of the developed method.

8.2. Performance of the coupled solver on unstructured meshes

A summary of the number of iterations, the CPU time, and the CPU time per control volume are presented in Table 2 for the various problems solved on grids with triangular control volumes. Except for the flow in a Planar Tee-Junction, the number of iterations required to solve a problem is independent of the grid size. The increase in the number of iterations for the flow in a Planar Tee-Junction problem is attributed to intermediate flow reversal at the exit section of the horizontal branch (Fig. 5(e)) before convergence is reached causing larger changes in the coefficients between two consecutive iterations.

As expected, the CPU time increases with the number of the control volumes. A more indicative performance parameter is the CPU per control volume, which is nearly constant (its percent variation is trivial as compared to the percent variation in the grid size) for all problems except for the flow in a Planar Tee-Junction (for the reasons stated above).

The above findings are in line with results reported in [27], for the performance of the coupled solver on structured quadrilateral control volumes, and a clear indication of a successful extension of the coupled solver to unstructured grid.

8.3. Comparison of performance of the coupled solver with the segregated solver

A summary of the number of iterations and CPU time needed by both segregated and coupled approaches using quadrilateral and triangular elements are presented for all problems and grid sizes in Table 3. Except for the flow in a Planar Tee-Junction problem, the number of iterations required by the coupled solver for both types of control volumes is nearly independent of the grid size. For the segregated solver this number increases with increasing the number of cells in the domain. The ratio of the number of iterations required by the segregated algorithm to the number required by the coupled algorithm (S/C) for quadrilateral (triangular) elements increases from 45 to 546 (78–642), 76 to 342 (46–296), 15 to 149 (17–185), 24 to 204 (28–261), 26 to 92 (31–134), and 12 to 154 (14–143) for the driven flow in a square cavity, driven flow in a skew cavity, backward facing step, sudden expansion in a square cavity, flow in a Planar Tee-Junction, and natural convection in a trapezoidal cavity problem, respectively. Because the cost per iteration is higher for the coupled solver, it is more meaningful to compare the CPU time consumed by both solvers. Results in Table 3 indicate that as the grid size increases from 10^4 to 3×10^5 quadrilateral (triangular) control volumes, the corresponding ratio of the CPU time needed by the segregated solver to the CPU time required by the coupled algorithm increases from 13 to 115 (13–104), 18 to 71 (8–58), 4 to 31 (3–33), 8 to 56 (6–54), 8 to 22 (6–25), and 5 to 38 (3–37) for the above problems. This represents a tremendous savings as the total time required by the coupled approach to solve all problems on the coarsest and densest quadrilateral (triangular) grids used are 209.6 and 11660.1 (339.7 and 12849.3) seconds while the times required by the segregated method are 1844.89 and 525911.74 (2113.11 and 564206) seconds with the average S/C ratio varying from 8.8 to 45.1 (6.22–43.9). This clearly demonstrates the virtues of the coupled approach.

8.4. Effects of the structured and unstructured grid systems on performance

Results presented in Table 3 also reveal that on the coarsest grid used (10^4 control volumes) the CPU time required by the coupled solver on structured grid is lower than the CPU time required on unstructured grid. On the densest grid (3×10^5 control volumes) however, the CPU time required on structured grid could be lower or higher than that required on unstruc-

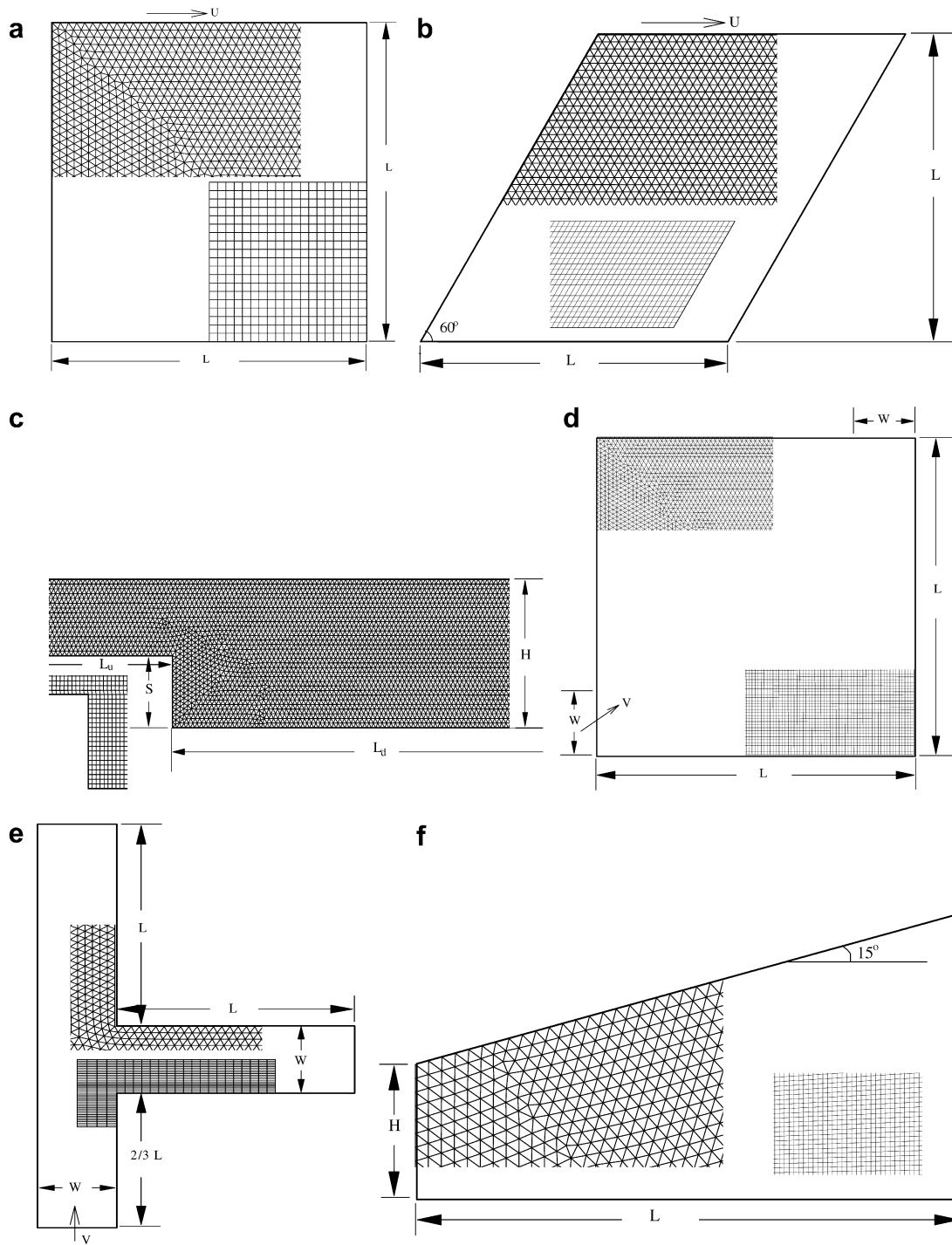


Fig. 4. Physical domain and illustrative triangular and quadrilateral grids used for the (a) driven flow in a rectangular cavity, (b) driven flow in a skew cavity, (c) flow behind a backward facing step, (d) sudden expansion in a square cavity, (e) flow in a Tee-Junction, and (f) natural convection in a trapezoidal cavity problems.

tured grid. While the connectivity of the grid is cheaper to establish on structured meshes in comparison with its connectivity on unstructured grid networks, the higher number of control volume faces associated with quadrilateral elements increases the computational cost. Since the same ILU(0) solver with an additive corrective multigrid method is used for both structured and unstructured solvers, the competing effects of the grid connectivity and number of control volume faces decide on whether the use of the coupled solver on a structured grid system results in an increase or a decrease in CPU time in comparison with its use on an unstructured mesh and results in the CPU times reported in Table 3.

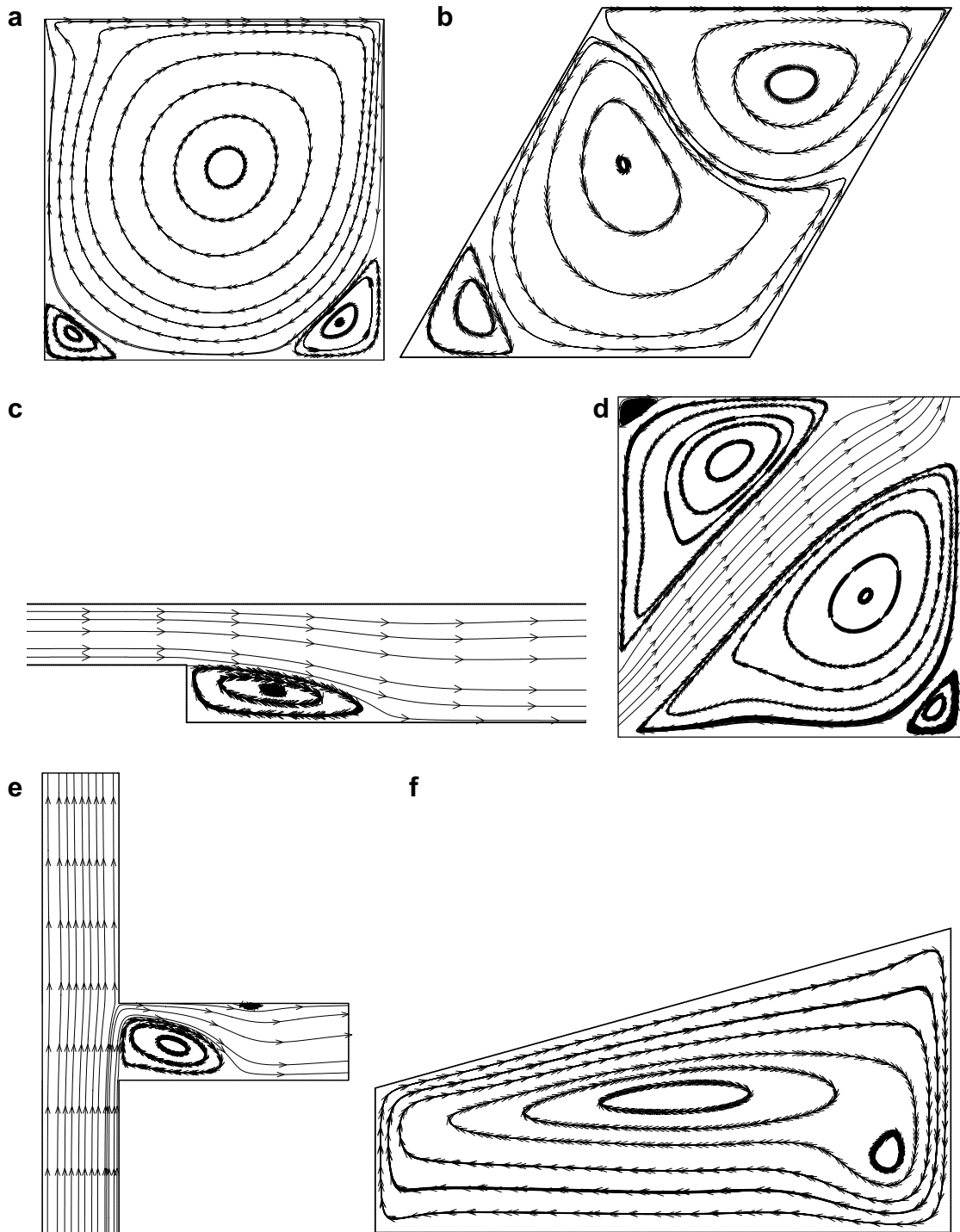


Fig. 5. Computed streamlines for the (a) driven flow in a rectangular cavity, (b) driven flow in a skew cavity, (c) flow behind a backward facing step, (d) sudden expansion in a square cavity, (e) flow in a Tee-junction, and (f) natural convection in a trapezoidal cavity problems.

8.5. Effect of the grid aspect ratio on performance

All results presented in the previous sections were generated on uniform structured and unstructured grid systems. To study the effect of grid non-uniformity (i.e. variable aspect ratio of the grid) on the performance of the coupled solver, the driven flow in a skew cavity problem is solved on a series of structured and unstructured grid systems of different degree

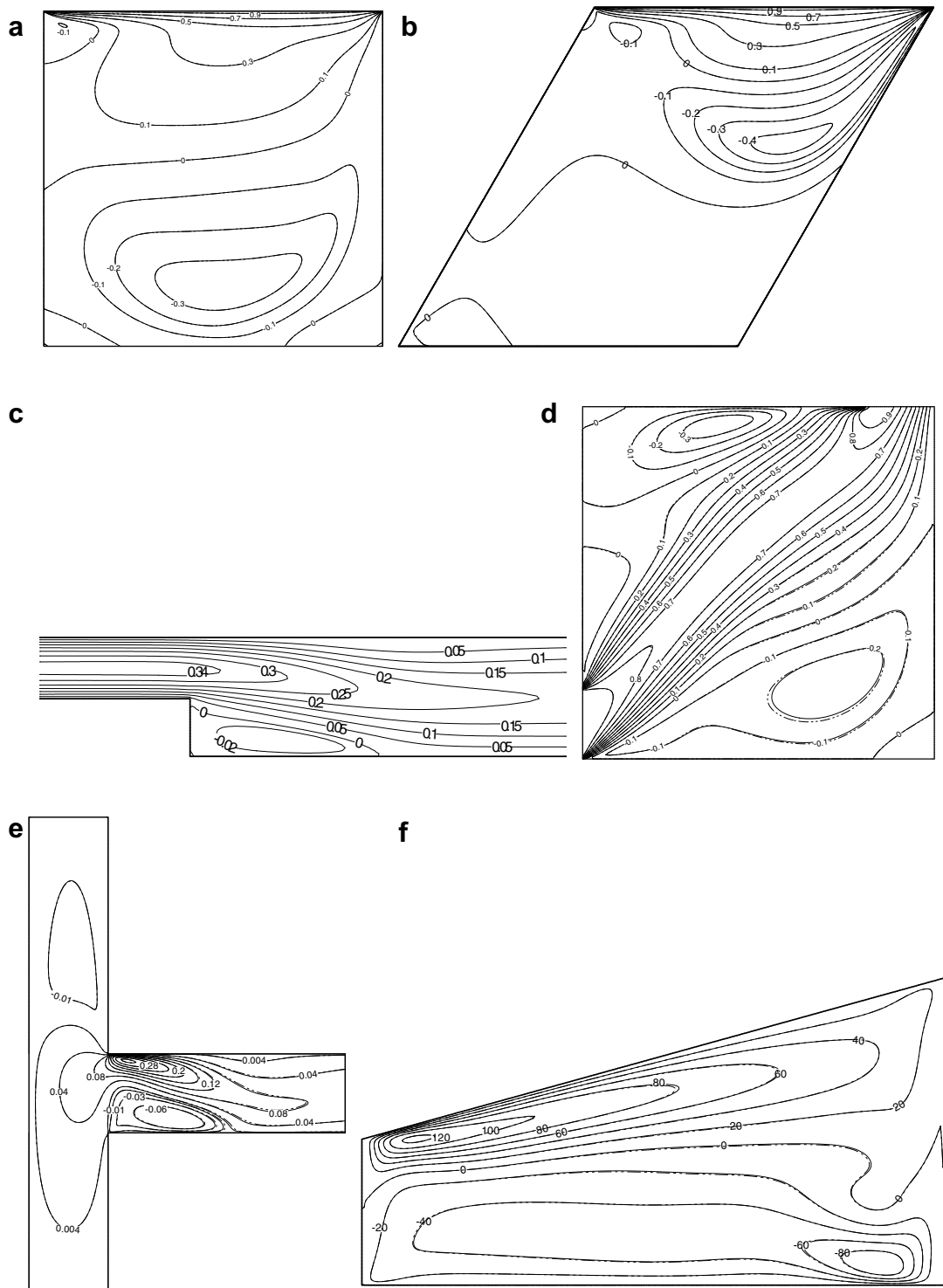


Fig. 6. Comparison of contours of constant u-velocity generated using the coupled and segregated solvers for the (a) driven flow in a rectangular cavity, (b) driven flow in a skew cavity, (c) flow behind a backward facing step, (d) sudden expansion in a square cavity, (e) flow in a Tee-Junction, and (f) natural convection in a trapezoidal cavity problems.

of non-uniformity and results (number of iterations and CPU times) are displayed in Table 4. The degree of non-uniformity is defined by the expansion ratio of the grid at the boundaries. Designating by Δx_i and Δx_{i+1} the lengths of the faces of the boundary elements i and $i + 1$, the expansion ratio is defined as $\varepsilon = \Delta x_{i+1} / \Delta x_i$. If expansion is performed in both directions

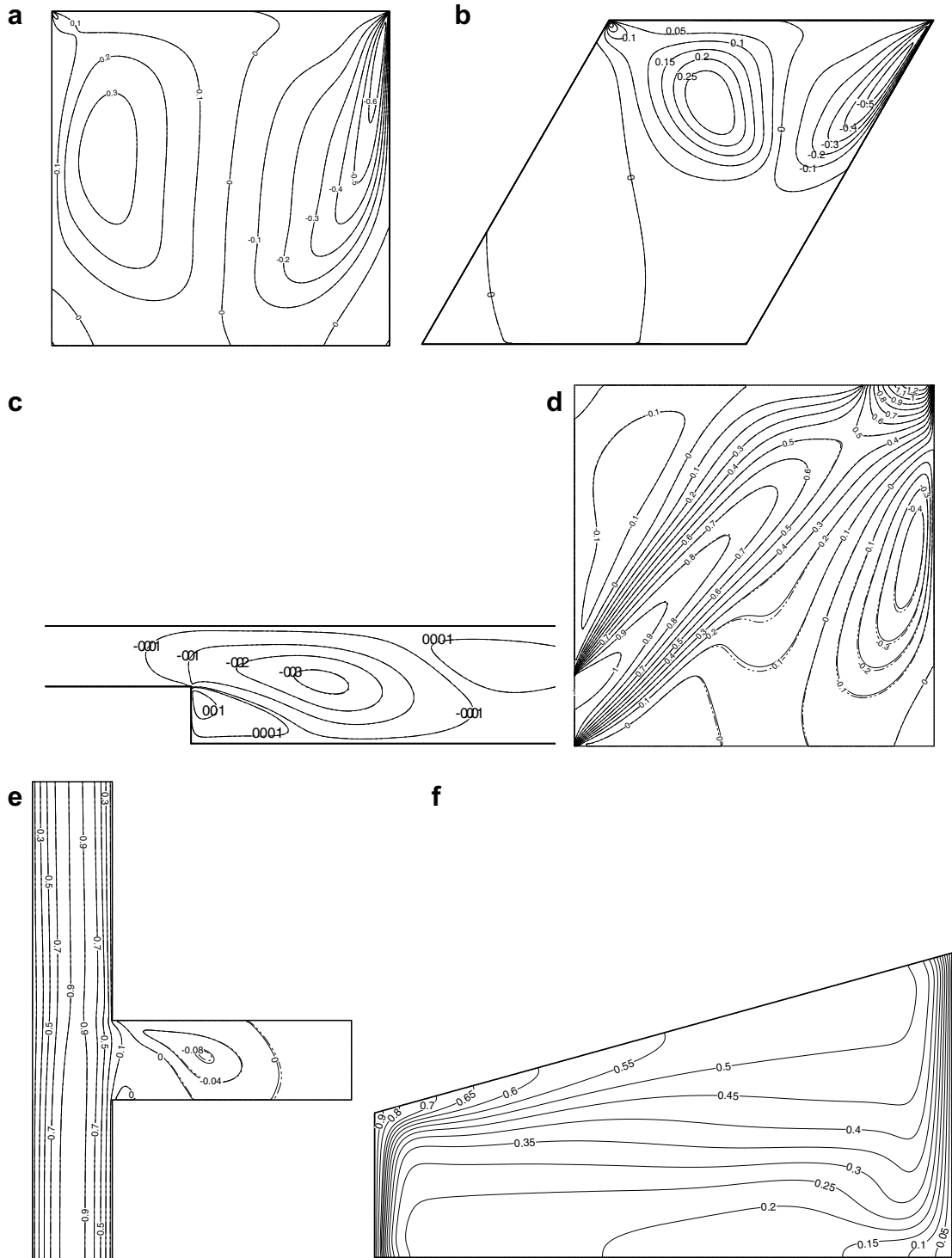


Fig. 7. Comparison of contours of constant v-velocity generated using the coupled and segregated solvers for the (a) driven flow in a rectangular cavity, (b) driven flow in a skew cavity, (c) flow behind a backward facing step, (d) sudden expansion in a square cavity, (e) flow in a Tee-junction, and (f) natural convection in a trapezoidal cavity problems.

(similar to concentrating grid points near both walls of a channel) and if the total number of elements along a boundary side is $2N$, then the ratio between the maximum and minimum length of a control volume face will be the expansion ratio raised to the power N (ε^N). The problem is solved using three grid sizes of values 10^4 , 5×10^4 , and 3×10^5 and for each grid size six

Table 3

Comparison of the number of iterations and CPU time required by the segregated and coupled flow solvers for the various problems on meshes of different sizes

Size	Quadrilateral elements						Triangular elements					
	# of iterations			CPU time			# of iterations			CPU time		
	C	S	S/C	C	S	S/C	C	S	S/C	C	S	S/C
<i>Driven flow in a square cavity</i>												
10^4	17	768	45	26.5	351.2	13	18	1400	78	45.0	582.3	13
5×10^4	17	2444	144	150.5	5621.8	37	17	3412	201	205.9	6249.	30
3×10^5	17	9280	546	1135.0	130890.4	115	17	10917	642	1211.5	125521	104
<i>Driven flow in a skew cavity</i>												
10^4	17	1286	76	30.5	553.39	18	23	1068	46	52.7	436.75	8
5×10^4	18	2599	144	170.5	5672.76	33	24	2191	91	278.9	4873.8	17
3×10^5	21	7190	342	1304.2	92114.74	71	24	7107	296	1691.3	97410	58
<i>Backward facing step</i>												
10^4	25	385	15	48.3	196.9	4	22	377	17	57.6	171.9	3
5×10^4	24	1090	45	240.8	2471.7	10	21	1052	50	246.9	2164.9	9
3×10^5	26	3869	149	1725.6	54289.2	31	23	4257	185	1766.9	58628	33
<i>Sudden expansion in a square cavity</i>												
10^4	23	547	24	31.0	262.6	8	27	761	28	61.1	366.16	6
5×10^4	23	1665	72	198.5	4292.6	22	26	2049	79	317.6	5005.6	16
3×10^5	27	5521	204	1560.7	86619.6	56	26	6791	261	1919.5	104195	54
<i>Flow in a Planar Tee-junction</i>												
10^4	26	671	26	35.3	295.3	8	27	824	31	59.7	349.2	6
5×10^4	42	1876	45	285.4	4347.8	15	31	2504	81	346.0	5647.8	16
3×10^5	74	6773	92	4045.6	89951.4	22	61	8165	134	4483.8	113552	25
<i>Natural convection in a Trapezoidal cavity</i>												
10^4	25	291	12	38.0	185.5	5	25	345	14	63.6	206.8	3
5×10^4	23	999	43	220.26	3422.9	16	25	977	39	303.0	2566.7	8
3×10^5	23	3536	154	1889.0	72046.4	38	23	3292	143	1776.3	64900	37

different values of the expansion ratio are considered: 1 (corresponding to uniform grid), 1.001, 1.01, 1.02, 1.05, and 1.1. The grid generation routine resulted in cells with negative volumes for a grid of size of 3×10^5 control volumes and an expansion ratio with value of 1.1 for quadrilateral elements (the ratio of the maximum to minimum length of a control volume face is nearly 2.2×10^{11}) and with values of 1.05 and 1.1 for triangular elements. Because of that it was not possible to generate solutions for these three cases.

Fig. 9 presents examples of non-uniform structured (Fig. 9(a and b)) and unstructured (Fig. 9(c and d)) grid systems for a grid with size of 5×10^4 control volumes for expansion ratios with values of 1.05 (Fig. 9(a and c)) and 1.1 (Fig. 9(b and d)). The large variation in the grid aspect ratio is apparent in the figures.

Table 4 presents the number of iterations and CPU time required by the coupled algorithm. For both structured and unstructured non-uniform grid systems, the number of iterations required to solve the problem is almost independent of the grid expansion ratio for values of $\varepsilon \leq 1.02$ on structured and $\varepsilon \leq 1.01$ on unstructured grid systems for all grid sizes considered.

For structured grid, the number of iterations almost doubles as the expansion ratio increases from 1 to 1.1. At the highest value of ε , the ratio of the maximum to minimum length of a boundary control volume face increases from 117 to 39, 317 as the grid size increase from 10^4 to 5×10^4 . This implies that the number of iterations only doubles on such a highly non-uniform grid. The CPU time follows a similar trend with its value increasing as the grid expansion ratio increases. As ε increases from 1 to 1.1, the CPU time almost doubles for a grid of size of 10^4 control volumes, while its value is almost 2.5 times higher on a grid of size of 5×10^4 cells. A similar behavior is observed with the densest grid used.

Table 4

The effect of the grid distribution on the number of iterations and CPU time required by the coupled flow solver for the driven flow in a skew cavity problem

Type	Quadrilateral elements						Triangular elements					
	10^4 CV		5×10^4 CV		3×10^5 CV		10^4 CV		5×10^4 CV		3×10^5 CV	
ε	Iteration	CPU	Iteration	CPU	Iteration	CPU	Iteration	CPU	Iteration	CPU	Iteration	CPU
1	17	30.5	18	170.5	21	1304.5	23	52.7	24	278.9	24	1691.3
1.001	16	27.0	18	171.9	21	1490.3	23	47.1	23	279.0	23	1593.4
1.01	18	34.3	18	262.1	22	1731.1	23	45.8	23	274.1	24	1697.8
1.02	17	33.0	16	242.2	21	1784.9	28	57.2	28	331.7	27	1874.2
1.05	24	45.3	23	321.7	26	2203.7	31	58.9	29	285.6		
1.1	37	60.6	37	434.4			30	42.4	31	310.5		

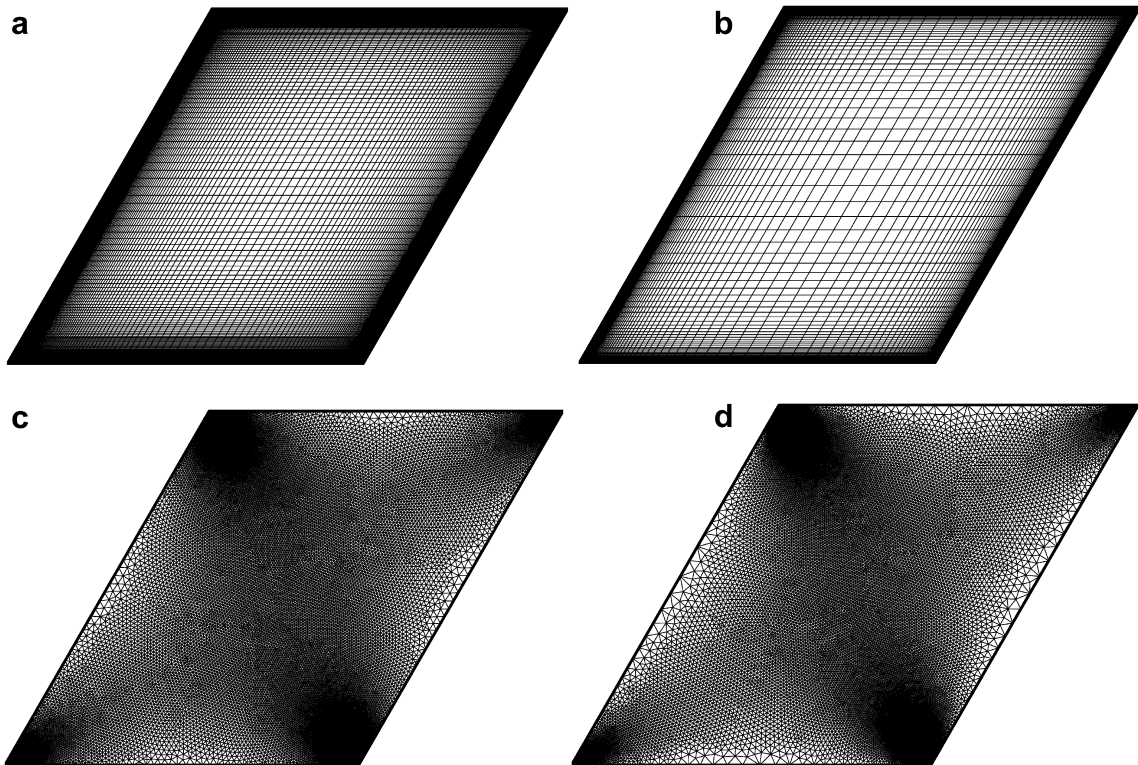


Fig. 9. Non-uniform grid systems with size of 5×10^4 cells for the driven flow in a skew cavity problem using (a and b) quadrilateral and (c and d) triangular elements and expansion ratios of values of (a and c) 1.05 and (b and d) 1.1.

For unstructured grid, the number of iterations varies slightly as the expansion ratio increases, with this number increasing from 23 to a maximum of 31 for $\varepsilon = 1.05$ ($\approx 35\%$ increase in the number of iterations). The variation of CPU time with ε is relatively small with values being slightly dependent on the grid aspect ratio. This difference in performance on structured and unstructured grids is attributed to the difference in the grid distribution over the computational domain as ε increases (compare grid networks in Fig. 9(a–d)).

The decrease, for some cases, in the required CPU time for a given grid size as ε increases even though the number of iterations required for convergence may be higher is due to the increase in the number of inner iterations needed to satisfy the local convergence criteria. For all computations, the residual reduction factor (RRF), defined as the ratio of the root mean square of the residuals of the algebraic system at the end of an inner iteration to the root mean square of the residuals at the start of the iteration process is set at 0.01 with the maximum number of iterations of the multigrid solver during any global iteration set at 10. The inner iterative process is stopped when either of the above two conditions is reached, which explains the variations in the CPU time.

Finally it should be noted that for structured and unstructured grid systems, the required number of iterations is almost independent of the grid size at any value of the grid expansion ratio, which is an important characteristic of a coupled solver.

9. Issues and future extensions

Extensions to the above described coupled algorithm could proceed in several directions. In the present work an algebraic multigrid method was used to accelerate the convergence of the linearized systems of equations. Further improvement to performance could be achieved by embedding the current algorithm within a Full Multigrid framework (denoted in the literature by the Full Approximation Storage, FAS) whereby inter-equations coupling and non-linearity are dealt with by applying multigrid to the outer iterations [36].

The convective terms have been discretized using the first order upwind scheme [24]. Higher accuracy will be sought by implementing, within the framework of the normalized variable and space formulation methodology [37], high resolution schemes following either the deferred correction approach [38] or the more implicit normalized weighting factor method [39].

Two issues that warrant further considerations are turbulence and flow compressibility. With regard to turbulence, appending turbulence models to the system of equations bring additional scalar transport equations (with the number depending on the model used) and cause the diffusion coefficients to vary several orders of magnitude within the solution

domain, which is bound to affect the convergence rate. As for flow compressibility, its inclusion will influence the discretization of the continuity equation and will require investigating whether to add the energy equation to the coupled system of continuity and momentum equations or to keep it as a separate equation to be solved independently. Both issues will have to be addressed in future work.

Another matter that deserves attention is the application of the coupled solver to unsteady flows. In most practical situations temporal accuracy can be achieved at a high Courant number. In these cases, multiple iterations of the segregated solver are needed for convergence at any one transient step. The performance advantage of the coupled is expected to carry to these situations. However for applications where the time scale necessitates a low courant number, the segregated approach and even the explicit scheme will have an advantage over the coupled approach.

10. Closing remarks

A pressure-based fully coupled method for the solution of laminar incompressible flow problems on unstructured grid was developed. The method was tested by solving the following five problems: (i) driven flow in a square and a skew cavity, (ii) flow behind a backward facing step, (iii) sudden expansion in a square cavity, (iv) flow in a Planar Tee-Junction, and (v) buoyancy-induced flow in a trapezoidal cavity. The performance of the coupled algorithm was assessed by comparing the number of iteration and CPU time required to produce a solution that converged to a desired level with those required using the segregated approach. It was found that for problems in which intermediate flow reversals at an outlet boundary do not occur during the solution, the number of iterations needed by the coupled algorithm is grid independent. Moreover, results showed a substantial decrease in computational time using the coupled approach when compared to using the segregated method with the reduction rate increasing as the grid size increases. The CPU times required by the coupled solver on structured and unstructured grids depended on the grid size with the time required on relatively coarse structured meshes being lower. Low and moderate variation in the grid aspect ratio did not alter the required number of outer iterations but, rather, affected the time consumed by the inner iterations with the CPU time increasing as the grid expansion ratio increases. At any grid expansion ratio, the required number of iterations was almost independent of the grid size.

Acknowledgments

The financial support provided by the University Research Board of the American University of Beirut through Grant # 888322 is gratefully acknowledged.

References

- [1] F. Moukalled, M. Darwish, A high resolution pressure-based algorithm for fluid flow at all speeds, *Journal of Computational Physics* 169 (1) (2001) 101–133.
- [2] W. Rodi, S. Majumdar, B. Schonung, Finite volume methods for two-dimensional incompressible flows with complex boundaries, *Computer Methods in Applied Mechanics and Engineering* 75 (1989) 369–392.
- [3] F. Moukalled, M. Darwish, Pressure-based algorithms for single-fluid and multifluid flows, in: W.J. Minkowycz, E.M. Sparrow, J.Y. Murthy (Eds.), *Handbook of Numerical Heat Transfer*, second ed., Wiley, 2006, pp. 325–367.
- [4] G.B. Deng, J. Piquet, X. Vasseur, M. Visonneau, A new fully coupled method for computing turbulent flows, *Computers and Fluids* 30 (2001) 445–472.
- [5] R. Webster, An algebraic multigrid solver for Navier–Stokes problems, *International Journal for Numerical Methods in Fluids* 18 (1994) 761–780.
- [6] R.F. Hanby, D.J. Silvester, A comparison of coupled and segregated iterative solution techniques for incompressible swirling flow, *International Journal for Numerical Methods in Fluids* 22 (1996) 353–373.
- [7] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Physics of Fluids* 8 (1965) 2182–2189.
- [8] A.J. Chorin, Numerical solution of the Navier–Stokes equation, *Mathematics of Computation* 22 (1971) 745–762.
- [9] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *International Journal of Heat and Mass Transfer* 15 (1972) 1787–1806.
- [10] F. Moukalled, M. Darwish, A unified formulation of the segregated class of algorithms for fluid flow at all speeds, *Numerical Heat Transfer, Part B* 37 (1) (2000) 103–139.
- [11] C.M. Rhie, W.L. Chow, A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation, *AIAA Journal* 21 (1983) 1525–1532.
- [12] G.E. Schneider, G.D. Raithby, M.M. Yonavovich, Finite element analysis of incompressible fluid flow incorporating equal order pressure and velocity interpolation, in: C. Taylor, K. Morgan, C.A. Brebbia (Eds.), *Numerical Methods in Laminar and Turbulent Flow*, Pentech Press, London, 1978.
- [13] P. Brufau, P. García-Navarro, Unsteady free surface flow simulation over complex topography with a multidimensional upwind technique, *Journal of Computational Physics* 186 (2) (2003) 503–526.
- [14] M. Darwish, F. Moukalled, B. Sekar, A unified formulation of the segregated class of algorithms for multi-fluid flow at all speeds, *Numerical Heat Transfer, Part B* 40 (2) (2001) 99–137.
- [15] F. Moukalled, M. Darwish, B. Sekar, A high resolution pressure-based algorithm for multi-phase flow at all speeds, *Journal of Computational Physics* 190 (2003) 550–571.
- [16] L.S. Caretto, R.M. Curr, D.B. Spalding, Two numerical methods for three-dimensional boundary layers, *Computer Methods in Applied Mechanics and Engineering* 1 (1972) 39–57.
- [17] S.P. Vanka, Block-implicit multigrid solution of Navier–Stokes equations in primitive variables, *Journal of Computational Physics* 65 (1) (1986) 138–158.
- [18] K.C. Karki, H.C. Mongia, Evaluation of a coupled solution approach for fluid flow calculations in body fitted co-ordinates, *International Journal for Numerical Methods in Fluids* 11 (1990) 1–20.
- [19] Braaten, M.E., Development and evaluation of iterative and direct methods for the solution of the equations governing recirculating flows, Ph.D. Thesis, University of Minnesota, May 1985.
- [20] Z. Mazhar, A procedure for the treatment of the velocity-pressure coupling problem in incompressible fluid flow, *Numerical Heat Transfer, Part B* 39 (2001) 91–100.

- [21] A. Pascau, C. Pérez, F.J. Serón, A comparison of segregated and coupled methods for the solution of the incompressible Navier–Stokes equations, *Communications in Numerical Methods in Engineering* 12 (1996) 617–630.
- [22] P.F. Galpin, G.D. Raithby, Numerical solution of problems in incompressible fluid flow: treatment of the temperature-velocity coupling, *Numerical Heat Transfer* 10 (1986) 105–129.
- [23] I. Ammara, C. Masson, Development of a fully coupled control-volume finite element method for the incompressible Navier–Stokes equations, *International Journal for Numerical Methods in Fluids* 44 (2004) 621–644.
- [24] S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, NY, 1981.
- [25] Van Doormal, J.P., *Numerical methods for the solution of incompressible and compressible fluid flows*, PhD Thesis, University of Waterloo, Ontario, Canada, 1985.
- [26] R.D. Lonsdale, An algebraic multigrid scheme for solving the Navier–Stokes equations on unstructured meshes, in: C. Taylor, J.H. Chin, G.M. Homsy (Eds.), *Numerical Methods in Laminar and Turbulent Flow*, vol. 7, Pineridge Press, Swansea, UK, 1991, pp. 1432–1442. part 2.
- [27] M. Darwish, I. Sraj, F. Moukalled, A coupled incompressible flow solver on structured grids, *Numerical Heat Transfer, Part B* 52 (2007) 353–371.
- [28] C. Pommerell, W. Fichtner, Memory aspects and performance of iterative solvers, *SIAM Journal of Scientific and Statistical Computing* 15 (1994) 460–473.
- [29] B.R. Hutchinson, G.D. Raithby, A multigrid method based on the additive correction strategy, *Numerical Heat Transfer* 9 (1986) 511–537.
- [30] R.P. Fedorenko, A relaxation method for solving elliptic difference equations, *USSR Computational Mathematics and Mathematical Physics* 1 (1962) 1092–1096.
- [31] F.V. Poussin, An accelerated relaxation algorithm for iterative solution of elliptic equations, *SIAM Journal of Numerical Analysis* 5 (1968) 340–351.
- [32] A. Brandt, Multi-level adaptive solutions to boundary value problems, *Mathematics of Computations* 31 (138) (1977) 333–390.
- [33] S.R. Elias, G.D. Stubley, G.D. Raithby, An adaptive agglomeration method for additive correction multigrid, *International Journal for Numerical Methods in Engineering* 40 (1997) 887–903.
- [34] R.E. Hayes, K. Nandakumar, H. Nasr-El-Din, Steady laminar flow in a 90 degree planar branch, *Computers and Fluids* 17 (1989) 537–553.
- [35] F. Moukalled, M. Darwish, Natural Convection in a Partitioned Trapezoidal Cavity Heated from the Side, *Numerical Heat Transfer, Part A* 43 (5) (2003) 543–563.
- [36] M. Darwish, F. Moukalled, B. Sekar, A Robust multi-grid algorithm for multifluid flow at all speeds, *International Journal for Numerical Methods in Fluids* 41 (2003) 1221–1251.
- [37] M. Darwish, F. Moukalled, Normalized variable and space formulation methodology for high-resolution schemes, *Numerical Heat Transfer, Part B* 26 (1) (1994) 79–96.
- [38] S.G. Rubin, P.K. Khosla, Polynomial interpolation method for viscous flow calculations, *Journal of Computational Physics* 27 (1982) 153–168.
- [39] M. Darwish, F. Moukalled, The normalized weighting factor method: a novel technique for accelerating the convergence of high-resolution convective schemes, *Numerical Heat Transfer, Part B* 30 (2) (1996) 217–237.