Journal of Computational Physics **167,** 217–243 (2001)
doi:10.1006/jcph.2000.6673, available online at http://www.idealibrary.com on IDEAL®

A Fourth-Order-Accurate Finite Volume Compact Method for the Incompressible Navier–Stokes Solutions

J. M. C. Pereira,* M. H. Kobayashi,† and J. C. F. Pereira‡

Department of Mechanical Engineering/LASEF, Instituto Superior Técnico/Technical University of Lisbon, Av. Rovisco Pais, 1049-001 Lisbon, Portugal
E-mail: *zeman@navier.ist.utl.pt, †marcelo@popsrv.ist.utl.pt, ‡jose@navier.ist.utl.pt

Received November 20, 1999; revised August 14, 2000

This paper presents a finite volume fourth-order-accurate compact scheme for discretization of the incompressible Navier–Stokes equations in primitive variable formulation. The numerical method of integrating the Navier–Stokes equations comprises a compact finite volume formulation of the average convective and diffusive fluxes. The pressure–velocity coupling is achieved via the coupled solution of the resulting system of equations. The solution of the coupled set of equations is performed with an implicit Newton–Krylov matrix-free method for stationary problems. For simulation of unsteady flows, a standard fourth-order Runge–Kutta method was used for temporal discretization and the velocity–pressure coupling was ensured at each stage also using the matrix-free method. Several incompressible viscous steady and unsteady flow problems have been computed to assess the robustness and accuracy of the proposed method.

Key Words: high-order schemes; finite volume; compact schemes; navier–stokes equations.

1. INTRODUCTION

Compact finite difference schemes have recently become popular and they are often called Padé schemes because of their similarity to schemes obtained from Padé approximations. Lele [1] has shown that high-order compact schemes require narrower computational grid stencils, have better fine-scale resolution, and yield better global accuracy than standard finite difference schemes with the same formal order of accuracy.

Several compact schemes have been proposed that can be cast into symmetric or non-symmetric stencils; see, e.g., Lele [1], Mahesh [2], and Tolstykh and Lipavskii [3]. Adams and Shariff [4] and Yee [5] present high-order compact methods aimed at problems with

0021-9991/01 $35.00

shock waves. Steady or unsteady Navier–Stokes solutions have been obtained with compact schemes; see, e.g., Gupta [6], Tang and Fornberg [7], Spotz and Carey [8], Wilson and Demuren [9], or Wilson *et al*. [10].

Compact schemes have been used primarily in conjunction with the finite difference formulation. The incorporation of compact schemes into the finite volume formulation is more complex and has recently been considered by Gaitonde and Shang [11] and Kobayashi [12]. Gaitonde and Shang [11] developed a range of fourth-order compact difference-based finite volume schemes for linear wave phenomena. The formulation combines the primitive function approach with five-point stencil of sixth- and fourth-order methods. Kobayashi [12] has formulated and examined a wide range of Padé finite volume formulations based on sliding averages of the variables and has investigated their properties related with accuracy, spectral resolution, boundary conditions, and stability.

To the authors' knowledge the particular implementation problems related with high-order compact finite volume schemes for multidimensional Navier–Stokes equations were not previously addressed. Hence, the main objective of the present work is to introduce a fourth-order-accurate numerical method of integrating the incompressible form of the steady or unsteady Navier–Stokes equations in primitive variable formulation. A compact fourth-order-accurate scheme for the discretization of the averaged convective and diffusive cell face fluxes is developed and implemented.

Special effort is dedicated to the numerical treatment of the nonlinear cell face averaged convective fluxes and the pressure–velocity coupling. The resulting set of equations was implicitly solved with the so-called Newton–Krylov matrix-free method. These techniques were studied in Marques and Pereira [13] in the context of the compressible Navier–Stokes equations, using ENO methods for the reconstruction of the primitive variables. These authors presented an implicit Newton–Krylov method, which uses the GMRES method together with various preconditioning techniques, such as Jacobi, polynomial approximations to the eigenvalues, or the spectrum. The present paper proposes an implicit Newton–Krylov method for the incompressible Navier–Stokes equations using the compact finite volume method.

For unsteady flow problems the fourth-order-accurate Runge–Kutta method is used. Four different flow test cases are considered in order to assess the robustness and accuracy of the method.

In the next section we present the main features of the numerical method together with the method used to solve the coupled set of resulting equations. The section closes with the deconvolution procedure, used to compute the point values of the variables, and the pressure–velocity coupling. Section 3 is devoted to the presentation of the flow test cases that demonstrate the method accuracy. The paper ends with summarizing conclusions.

2. NUMERICAL METHOD

The continuity and Navier–Stokes equations that describe the incompressible flow of a Newtonian fluid can be represented in the intrinsic form as

$$\text{(a)} \quad \operatorname{div}(\mathbf{v}) = 0 \tag{2.1}$$

$$\text{(b)} \quad \frac{\partial \mathbf{v}}{\partial t} + \operatorname{div}(\mathbf{v} \otimes \mathbf{v} - \quad _{\nu}(\mathbf{v})) = -\operatorname{grad} p, \tag{2.2}$$

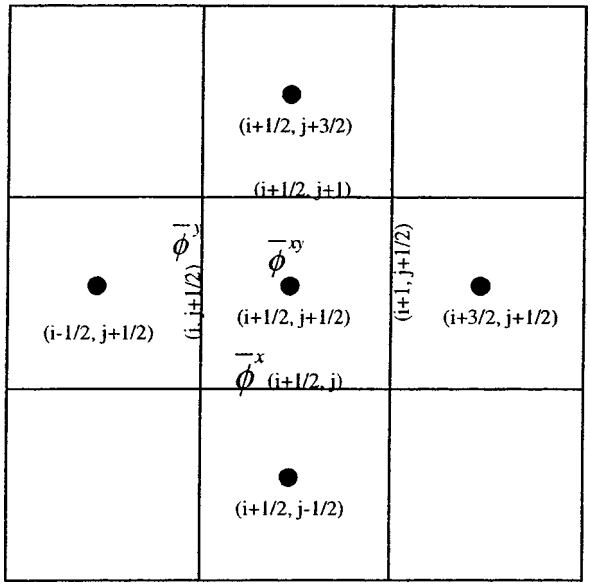


FIG. 1. Mesh example and notation.

where $\mathbf{v}$ is the velocity vector field, p is the specific pressure scalar field, and

$$\tau_\nu(\mathbf{v}) = \nu(\operatorname{grad}\mathbf{v} + (\operatorname{grad}\mathbf{v})^T)$$

is the deviatoric stress tensor according to the Stokes model.

To develop the finite volume formulation the equations are integrated over each control volume (see Fig. 1). With the application of the Gauss divergence theorem, Eqs. (2.1) and (2.2) become

(a) Continuity equation

$$\int_{\partial\theta} \mathbf{v}\cdot\mathbf{n} = 0 \tag{2.1a}$$

(b) Navier–Stokes equations

$$\int_{\partial\theta} (\mathbf{v}\cdot\mathbf{n})\mathbf{v} - \tau_\nu(\mathbf{v})\cdot\mathbf{n} = -\int_{\partial\theta} p\mathbf{n}, \tag{2.2a}$$

where $\partial\theta$ denotes the boundary of a control volume θ, and $\mathbf{n}$ is the unit outward pointing normal vector to $\partial\theta$. In the finite volume approach each control volume boundary is usually further decomposed into piecewise linear elements; $\partial\theta = \bigcup_{i=1}^{f} \gamma_*$, where γ_* are line segments such that the intersection of two adjacent elements are two vertice points. Equations (2.1a) and (2.2a) are general integral equations that are valid for any coordinate system. For simplicity, we present the fourth-order method for a Cartesian grid, with coordinates (x, y) (see Remark 2.1 on the formulation in curvilinear coordinates). Hence, let $\{\theta_{ij}\}$, where $\theta_{ij} = [x_i, x_{i+1}] \times [y_j, y_{j+1}]$, with $\Delta x = x_{i+1} - x_i$, $\Delta y = y_{j+1} - y_j$, be a uniform Cartesian partition of a rectangular domain $\Omega \subset \mathbf{R}^2$. Then, Eqs. (2.1a) and (2.2a) can be written in the averaged flux balance form

(a) Continuity equation

$$([\overline{u}^y]_{i+1} - [\overline{u}^y]_i)\Delta y + ([\overline{v}^x]_{j+1} - [\overline{v}^x]_j)\Delta x = 0, \tag{2.3}$$

where $\mathbf{v} = (u, v)$, and, for instance,

$$[\overline{u}^y]_i \equiv \frac{1}{\Delta y}\int_{y_j}^{y_{j+1}} u(x_i, y)\, dy, \tag{2.4}$$

(b) Navier–Stokes equation

$$\mathsf{C}(\mathbf{v}) - \mathsf{D}(\mathbf{v}) = -\mathsf{G}p. \tag{2.5}$$

The letters C, D, and G denote the integral convective, diffusive, and gradient maps, respectively. That is, locally we have

$$[\mathsf{C}_{ij}(\mathbf{v})]^x = ([\overline{uu}^y]_{i+1} - [\overline{uu}^y]_i)\Delta y + ([\overline{vu}^x]_{j+1} - [\overline{vu}^x]_j)\Delta x \tag{2.6}$$

$$[\mathsf{C}_{ij}(\mathbf{v})]^y = ([\overline{uv}^y]_{i+1} - [\overline{uv}^y]_i)\Delta y + ([\overline{vv}^x]_{j+1} - [\overline{vv}^x]_j)\Delta x \tag{2.7}$$

$$[\mathsf{D}_{ij}(\mathbf{v})]^x = \nu\left[\left(\left[\overline{\frac{\partial u}{\partial x}}^y\right]_{i+1} - \left[\overline{\frac{\partial u}{\partial x}}^y\right]_i\right)\Delta y + \left(\left[\overline{\frac{\partial u}{\partial y}}^x\right]_{j+1} - \left[\overline{\frac{\partial u}{\partial y}}^x\right]_j\right)\Delta x\right] \tag{2.8}$$

$$[\mathsf{D}_{ij}(\mathbf{v})]^y = \nu\left[\left(\left[\overline{\frac{\partial v}{\partial x}}^y\right]_{i+1} - \left[\overline{\frac{\partial v}{\partial x}}^y\right]_i\right)\Delta y + \left(\left[\overline{\frac{\partial v}{\partial y}}^x\right]_{j+1} - \left[\overline{\frac{\partial v}{\partial y}}^x\right]_j\right)\Delta x\right] \tag{2.9}$$

$$\mathsf{G}_{ij}\, p = (([\overline{p}^y]_{i+1} - [\overline{p}^y]_i)\Delta y, ([\overline{p}^x]_{j+1} - [\overline{p}^x]_j)\Delta x). \tag{2.10}$$

We proceed in the next section with the derivation of the compact method for discretizing the cell face fluxes appearing in Eqs. (2.6) to (2.10).

2.1. A Compact Fourth-Order Finite Volume Method for the Navier–Stokes Equations

To compute the fluxes of the finite volume discretization of the Navier–Stokes equations, we look for a Padé type relation between the fluxes and the cell average of the primitive variables. The relations can be obtained in several ways; we use the common Taylor series approach.

The high-order finite volume discretization should be associated with the variable cell averages instead of its point values. However, in many reconstruction procedures (see, for example, [14]), it is common to recover the point values, which are represented by a piecewise reconstruction polynomial. To obtain the fluxes along the cell faces, it is then necessary to use a high-order numerical integration method (Gauss quadrature, Simpson, etc.) that integrates the flux from a set of point values previously selected.

In the present approach, we store and use the cell averages during all the processes (the point values, if necessary, are recovered at the end of the computation by a deconvolution procedure; see Section 2.5). This strategy, for the same order of accuracy, simplifies the computations by reducing the size of the stencil and avoiding the integration of the reconstruction polynomial associated with point-value reconstruction mentioned above.

2.1.1. *Linear Convective Fluxes*

Because of the nondissipating nature of their truncation error, we consider centered schemes. So, by symmetry, the coefficients on the left and right sides are equal.

Let us consider the $\overline{u}^y$ as an example. The problem can be stated as follows: Find coefficients a and b that satisfy the relationship

$$a\tau_{\Delta x}\overline{u}^y + \overline{u}^y + a\tau_{-\Delta x}\overline{u}^y = b\left(\tau_{\frac{1}{2}\Delta x}\overline{u}^{xy} + \tau_{-\frac{1}{2}\Delta x}\overline{u}^{xy}\right) + \mathrm{O}(h^4), \tag{2.11}$$

where h is a grid parameter, for example, $h = \frac{4A}{P}$, where A is the area, and P the perimeter of the control volume, Δx as well as Δy are the grid spacing in the x and y directions respectively, τ is the shift operator, and

$$\overline{u}^{xy} \equiv \frac{\int_{-\Delta x/2}^{\Delta x/2}\int_{-\Delta y/2}^{\Delta y/2} u(x+\xi, y+\eta)\, d\xi\, d\eta}{\Delta x\, \Delta y} \tag{2.12}$$

is the sliding average. Since we only need the fluxes at the cell faces, taking into account Eq. (2.11) it is only necessary to store the values of the sliding averages at the centroids of the control volumes.

With the use of truncated Taylor series (TTS), one can expand $u(x, y)$ in the vicinity of (x_0, y_0) as

$$\begin{aligned} u(x,y) = {} & u_0 + (x-x_0)u_0^{(1,0)} + (y-y_0)u_0^{(0,1)} + (x-x_0)(y-y_0)u_0^{(1,1)} \\ & + (x-x_0)^2\frac{u_0^{(2,0)}}{2} + (y-y_0)^2\frac{u_0^{(0,2)}}{2} + (x-x_0)^2(y-y_0)\frac{u_0^{(2,1)}}{2} \\ & + (x-x_0)(y-y_0)^2\frac{u_0^{(1,2)}}{2} + (x-x_0)^3\frac{u_0^{(3,0)}}{6} + (y-y_0)^3\frac{u_0^{(0,3)}}{6} + \mathrm{O}(h^4). \end{aligned} \tag{2.13}$$

Because we are interested in the mean values only, taking into account that the integral is a linear operator and the origin at $(x0, y0)$ is the middle point of the face for which we want to compute the flux, we have

$$TTS(\overline{u}^y) \equiv \frac{1}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2} TTS(u(x0, y0+\eta))\, d\eta = u_0 + \Delta y^2\frac{u_0^{(0,2)}}{24} + \mathrm{O}(h^4) \tag{2.14}$$

$$\begin{aligned} TTS(\tau_{-\Delta x}\overline{u}^y) \equiv {} & \frac{1}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2} TTS(u(x0+\Delta x, y0+\eta))\, d\eta \\ = {} & u_0 + \Delta y u_0^{(1,0)} + \Delta x^2\frac{u_0^{(2,0)}}{2} + \Delta y^2\frac{24u_0^{(0,2)}}{576} + \Delta x^3\frac{u_0^{(3,0)}}{6} \\ & + \Delta x\Delta y^2\frac{24u_0^{(1,2)}}{576} + \mathrm{O}(h^4) \end{aligned} \tag{2.15}$$

$$\begin{aligned} TTS\left(\tau_{-\frac{1}{2}\Delta x}\overline{u}^{xy}\right) \equiv {} & \frac{\int_0^{\Delta x}\int_{-\Delta y/2}^{\Delta y/2} TTS(u(x0+\xi, y0+\eta))\, d\xi\, d\eta}{h^2} \\ = {} & u_0 + \Delta x\frac{u_0^{(1,0)}}{2} + \Delta x^2\frac{u_0^{(2,0)}}{6} + \Delta y^2\frac{u_0^{(0,2)}}{24} + \Delta x\Delta y^2\frac{u_0^{(1,2)}}{48} \\ & + \Delta x^3\frac{u_0^{(3,0)}}{24} + \mathrm{O}(h^4), \end{aligned} \tag{2.16}$$

and similarly for the other terms.

Replacing these expressions in Eq. (2.11) and nullifying the coefficients of the derivatives, one obtains a linear system of equations. Solution of this problem gives $a = 1/4$ and $b = 3/4$.

Therefore, to compute the edge averages at the control-volume faces, it is only necessary to solve one direction at a time. This requires the solution of only tri-diagonal solutions. Indeed, for a fixed index j (which corresponds to a horizontal strip of the domain), we solve the tri-diagonal system of equations resulting from Eq. (2.11) to obtain $(\overline{u}^y_{i,j})$ in that strip. Then repeating this process to all strips we obtain the values of $(\overline{u}^y_{i,j})$ for all control-volume faces. The values for other cell face averages are computed analogously. Note, in passing, that the strip-by-strip computation of the cell averages is the same procedure used to compute the derivatives in the compact finite difference method [1].

To facilitate the description of the method of solution of the Navier–Stokes equations, it is convenient to express the previous procedure in matrix form,

$$A_x^{Cu}\overline{u}^y = B_x^{Cu}\overline{u}^{xy}, \tag{2.17}$$

and for the remaining terms we have

$$A_y^{Cu}\overline{u}^x = B_y^{Cu}\overline{u}^{xy} \tag{2.18}$$

$$A_x^{Cv}\overline{v}^y = B_x^{Cv}\overline{v}^{xy} \tag{2.19}$$

$$A_y^{Cv}\overline{v}^x = B_y^{Cv}\overline{v}^{xy} \tag{2.20}$$

$$A_x^{p}\overline{p}^y = B_x^{p}\overline{p}^{xy} \tag{2.21}$$

$$A_y^{p}\overline{p}^x = B_y^{p}\overline{p}^{xy}, \tag{2.22}$$

where, for example,

$$A_x(nfx \times nfx),\ B_x(nel \times nfx)$$

and

$$\overline{\phi}^{xy}(nel),\ \overline{\phi}^{y}(nfx),\ \overline{\phi}^{x}(nfy),$$

with ϕ standing for the vector quantities (cell averages of the primitive variables and its cell face averaged values) and

$$\begin{aligned} nel &= (ni-1)(nj-1) \\ nf &= ni\ nj \\ nfx &= ni\ (nj-1) \\ nfy &= (ni-1)\ nj \end{aligned}$$

ni, nj being the computational grid points in the x and y directions, respectively.

Note that matrices A and B, apart from points close to boundaries, are the same for all variables. Also, all matrices have a block diagonal form reflecting the decoupling of the approximation of the cell face averages in one direction from the other. For instance, the

matrix A can be represented as

$$A = \begin{bmatrix} \tilde{A} & & & & \\ & \tilde{A} & & 0 & \\ & & \ddots & & \\ & & & \ddots & \\ & 0 & & \tilde{A} & \\ & & & & \tilde{A} \end{bmatrix}$$

with $\tilde{A}_x(nfx \times nfx)$ or $\tilde{A}_y(nfy \times nfy)$ being the tri-diagonal matrix corresponding to a horizontal or vertical strip, respectively, of the Cartesian grid.

2.1.2. *Diffusive Fluxes*

For the diffusive fluxes, for example, $\overline{\frac{\partial u}{\partial x}}^{y}$ coefficients a and b are obtained from

$$a\tau_{\Delta x}\overline{\frac{\partial u}{\partial x}}^{y} + \overline{\frac{\partial u}{\partial x}}^{y} + a\tau_{-\Delta x}\overline{\frac{\partial u}{\partial x}}^{y} = \frac{b}{\Delta x}\left(\tau_{-\frac{1}{2}\Delta x}\overline{u}^{xy} - \tau_{\frac{1}{2}\Delta x}\overline{u}^{xy}\right) + \mathrm{O}(h^4), \tag{2.23}$$

with

$$\overline{\frac{\partial u}{\partial x}}^{y} \equiv \frac{1}{\Delta y}\int_{-\Delta y/2}^{\Delta y/2} \frac{\partial u}{\partial x}(x, y+\eta)\, d\eta. \tag{2.24}$$

The solution of this problem gives $a = 1/10$ and $b = 6/5$.

In matrix form and for all the existing terms we have

$$A_x^D \overline{\frac{\partial u}{\partial x}}^{y} = B_x^D \overline{u}^{xy} \tag{2.25}$$

$$A_y^D \overline{\frac{\partial u}{\partial y}}^{x} = B_y^D \overline{u}^{xy} \tag{2.26}$$

$$A_x^D \overline{\frac{\partial v}{\partial x}}^{y} = B_x^D \overline{v}^{xy} \tag{2.27}$$

$$A_y^D \overline{\frac{\partial v}{\partial y}}^{x} = B_y^D \overline{v}^{xy}. \tag{2.28}$$

Generalization of the compact finite volume representations of convective or diffusive fluxes can be found in [12].

2.1.3. *Nonlinear Convective Fluxes*

The quadratic terms, $\overline{uu}^{y}$, $\overline{uv}^{y}$, $\overline{vu}^{x}$, $\overline{vv}^{x}$ that appear in the momentum equations need more careful treatment. One way to extend the method used above to handle such terms is as follows (for example for $\overline{uu}^{y}$): Compute coefficients a and $\{b_i\}_{i=1,2}$ such that

$$\begin{aligned} &a\tau_{\Delta x}\overline{uu}^{y} + \overline{uu}^{y} + a\tau_{-\Delta x}\overline{uu}^{y} \\ &\quad = \sum_{i=1}^{2} b_i \tau_{(-i+\frac{1}{2})\Delta x}(\overline{u}^{xy})^2 + \sum_{i=1}^{2} b_i \tau_{(i-\frac{1}{2})\Delta x}(\overline{u}^{xy})^2 + \mathrm{O}(h^4). \end{aligned} \tag{2.29}$$

Solution of this problem gives $a = -1/2$, and $b_1 = 1/4, b_2 = -1/4$. However, this is a fourth-order approximation for the square of the sliding averages. Taking square roots leads to a second-order approximation for the sliding averages. This difficulty should be overcome by taking an eighth-order compact method for the squares. We proceed in a different direction, however.

Indeed, instead of trying to approximate the square of a cell face average, for example, $[\overline{uu}^y]_{i+1}$, by the squares of cell averages like $([\overline{u}^{xy}]_{i+1/2})^2$, we approximate the former by the squares of the cell face average $([\overline{u}^y]_{i+1})^2$ together with some cell averages. For example, if we compare the Taylor series expansion of $[\overline{uu}^y]_{i+1}$ with $([\overline{u}^y]_{i+1})^2$, up to fourth order, the remaining term is

$$\frac{\Delta y^2}{12}\left(u_0^{(0,1)}\right)^2 + \mathrm{O}(h^4),$$

which vanishes for a field that is independent of the coordinate y. This means that for the nonlinear terms, additional information must be supplied to take into account the variation of the function along the cell face.

Thus, a second-order approximation of $u_0^{(0,1)}$ is enough to recover the desired accuracy for the approximation of the nonlinear flux. A simple computation shows that if

$$\begin{aligned}\Delta y u_0^{(0,1)} = {} & a_1[\overline{u}^{xy}]_{i+1/2,\,j-1/2} + a_2[\overline{u}^{xy}]_{i+1/2,\,j+3/2} + a_3[\overline{u}^{xy}]_{i+3/2,\,j-1/2} \\ & + a_4[\overline{u}^{xy}]_{i+3/2,\,j+3/2} + \mathrm{O}(\Delta x^2, \Delta y^2),\end{aligned} \tag{2.30}$$

then

$$a_1 = a_3 = -\frac{1}{4}; \quad a_2 = a_4 = \frac{1}{4},$$

and consequently,

$$\begin{aligned}[\overline{uu}^y]_{i+1} = {} & ([\overline{u}^y]_{i+1})^2 + \frac{1}{192}(-[\overline{u}^{xy}]_{i+1/2,\,j-1/2} + [\overline{u}^{xy}]_{i+1/2,\,j+3/2} \\ & - [\overline{u}^{xy}]_{i+3/2,\,j-1/2} + [\overline{u}^{xy}]_{i+3/2,\,j+3/2})^2 + \mathrm{O}(h^4).\end{aligned} \tag{2.31}$$

The remaining terms, $[\overline{vu}^y]_{i+1}$, $[\overline{vv}^x]_{j+1}$, and $[\overline{uv}^x]_{j+1}$, are discretized analogously. Again, to simplify the exposition, the above discretization procedure is stated in the form of maps,

$$\overline{uu}^y = B_x^{C^2}(\overline{u}^{xy}, \overline{u}^y) \tag{2.32}$$

$$\overline{uv}^y = B_x^{C^2}(\overline{u}^{xy}, \overline{v}^{xy}, \overline{u}^y, \overline{v}^y) \tag{2.33}$$

$$\overline{vu}^x = B_y^{C^2}(\overline{u}^{xy}, \overline{v}^{xy}, \overline{u}^x, \overline{v}^x) \tag{2.34}$$

$$\overline{vv}^x = B_y^{C^2}(\overline{v}^{xy}, \overline{v}^x), \tag{2.35}$$

where B is a nonlinear map.

2.1.4. *Boundary Conditions*

The boundary conditions at the control-volume faces which intersect the domain boundary need to be implemented in the Padé finite volume compact operator. Dirichlet or von Neumann conditions require the prescription of flux values in the respective Padé operator.

As proved in [12], fourth-order boundary conditions are stable and necessary to keep the global accuracy of the method. They differ from the common boundary conditions used in compact finite differences [15], in that the finite difference scheme requires a downwind (very unstable) approach for the inlet convective term while under the compact finite volume method we can use the inlet convective flux without approximation. We briefly discuss this issue below.

Take, for instance, a prescribed Dirichlet boundary condition. This type of boundary condition abounds in fluid problems. The no-slip condition on a wall for the components of the velocity, an inlet boundary condition on any variable, the no-flux condition on a symmetry line, and the far field condition are examples of this type of boundary condition.

Let us write the balance equation of a control volume close to a left boundary. To fix ideas, we consider the problem of handling the convective flux at a prescribed left boundary. The balance equation near the boundary can be written as

$$C^x_{2,\,j+1/2} - C^x_{1,\,j+1/2} + C^y_{3/2,\,j+1} - C^y_{3/2,\,j} + \cdots = 0, \tag{2.36}$$

where $C^x_{2,\,j+1/2}$, $C^x_{1,\,j+1/2}$, $C^y_{3/2,\,j+1}$, and $C^y_{3/2,\,j}$ denote the convective flux crossing the respective faces. Now, because the value of the variable is prescribed at the inlet (face $1,\ j+1/2$), we insert the exact value of $C^x_{1,\,j+1/2}$ in this equation without any approximation. The remaining $C^x_{2,\,j+1/2}$, $C^y_{3/2,\,j+1}$, and $C^y_{3/2,\,j}$ fluxes are evaluated using the compact method as explained above. For instance, to compute the flux

$$C^x_{2,\,j+1/2} = \overline{uu}^y_{2,\,j+1/2}\Delta y, \tag{2.37}$$

we use the approximation

$$[\overline{u\overline{u}^y}]_{2,\,j+1/2} = \left([\overline{u}^y]_{2,\,j+1/2}\right)^2 + \frac{1}{192}\left(-[\overline{u}^{xy}]_{5/2,\,j-1/2} + [\overline{u}^{xy}]_{5/2,\,j+3/2} - [\overline{u}^{xy}]_{7/2,\,j-1/2} + [\overline{u}^{xy}]_{7/2,\,j+3/2}\right)^2,$$

where $[\overline{u}^y]_{2,\,j+1/2}$ is evaluated using the compact finite volume method as

$$\frac{1}{4}[\overline{u}^y]_{1,\,j+1/2} + [\overline{u}^y]_{2,\,j+1/2} + \frac{1}{4}[\overline{u}^y]_{3,\,j+1/2} = \frac{3}{4}\left([\overline{u}^{xy}]_{3/2,\,j+1/2} + [\overline{u}^{xy}]_{5/2,\,j+1/2}\right)$$

and so involves $[\overline{u}^y]_{1,\,j+1/2}$, which is known, and $[\overline{u}^y]_{3,\,j+1/2}$, $[\overline{u}^{xy}]_{3/2,\,j+1/2}$, and $[\overline{u}^{xy}]_{5/2,\,j+1/2}$, which are to be determined in the solution process. Since the exact value of the convective flux is available and enters directly in the balance equation and also the exact value of the cell face average $[\overline{u}^y]_{1,\,j+1/2}$ is prescribed and enters directly in the compact reconstruction, the compact finite volume method does not require any downwind extrapolation.

In contrast, the corresponding finite difference treatment requires an equation for $[\partial u/\partial_x]_{1,\,j}$ at the inlet boundary $1,\ j$. The approximation is asymmetric and in its simplest

form (for fourth-order accuracy) can be written as [15]

$$\left[\frac{\partial u}{\partial x}\right]_{1,j} + 3\left[\frac{\partial u}{\partial x}\right]_{2,j} = \frac{1}{6\Delta x}(-17u_1 + 9u_2 + 9u_3 - u_4). \tag{2.38}$$

Now this last equation is a downwind extrapolation, which, as reported in Carpenter *et al*. [15], is unstable. Other types of boundary conditions are handled analogously.

Remark 2.1. The discretization procedure was presented for a Cartesian mesh. In this remark we briefly discuss the extension of the formulation for curvilinear coordinates. The latter is indeed deceptively simple.

Consider the conservative form of the Navier–Stokes equations written in a general curvilinear coordinate system (x^i),

$$\frac{1}{\sqrt{g}}\frac{\partial}{\partial x^\alpha}C_i^\alpha u^i u^j + \cdots = 0, \tag{2.39}$$

where we have used the summation convection of repeated indexes, the Latin superscripts indicate the Cartesian components of vectors and tensors, $g = \det g_{\alpha\beta}$, with $g_{\alpha\beta}$ the metric matrix, J is the Jacobian, C is the cofactor matrix, with y^i the Cartesian coordinate system. Again, to fix ideas we concentrate in the convective flux. Integration over a control volume θ then leads to

$$\int_\theta \left(\frac{\partial}{\partial x^\alpha}C_i^\alpha u^i u^j + \cdots\right) dx = 0, \tag{2.40}$$

where in two dimensions $dx = dx^1 dx^2$. After using the Gauss theorem, we are left with the problem of approximating $\overline{C_1^1 u^1 u^1}^2, \overline{C_2^1 u^2 u^1}^2, \ldots, \overline{C_2^2 u^2 u^2}^1$, where, for example,

$$\overline{C_1^1 u^1 u^1}^2 = \int_{x^2}^{x^2+\Delta x^2} C_1^1 u^1 u^1 \, dx^2.$$

The problem is solved if we are able to discretize a triple product. For instance, consider the product $\overline{C_1^1 u^1 u^2}^2$. Then, to discretize the latter we recursively apply the procedure for double products to the subproducts

$$\overline{(C_1^1 u^1) u^2}^2, \overline{C_1^1 (u^1 u^2)}^2, \quad \text{and } \overline{(C_1^1 u^2) u^1}^2$$

yielding

$$\begin{aligned}
\left[\overline{C_1^1 u^1 u^2}^2\right]_{i+1} = {} & \left[\overline{C_1^1}^2\right]_{i+1}[\overline{u^1}^2][\overline{u^2}^2] + \frac{1}{192}\left[\overline{C_1^1}^2\right]_{i+1}\left(-[\overline{u^1}^{12}]_{i+1/2,j-1/2} + [\overline{u^1}^{12}]_{i+1/2,j+3/2}\right. \\
& \left. - [\overline{u^1}^{12}]_{i+3/2,j-1/2} + [\overline{u^1}^{12}]_{i+3/2,j+3/2}\right)\left(-[\overline{u^2}^{12}]_{i+1/2,j-1/2}\right. \\
& \left. + [\overline{u^2}^{12}]_{i+1/2,j+3/2} - [\overline{u^2}^{12}]_{i+3/2,j-1/2} + [\overline{u^2}^{12}]_{i+3/2,j+3/2}\right) \\
& + \frac{1}{192}[\overline{u^1}^2]_{i+1}\left(-\left[\overline{C_1^1}^{12}\right]_{i+1/2,j-1/2} + \left[\overline{C_1^1}^{12}\right]_{i+1/2,j+3/2}\right. \\
& \left. - \left[\overline{C_1^1}^{12}\right]_{i+3/2,j-1/2} + \left[\overline{C_1^1}^{12}\right]_{i+3/2,j+3/2}\right)\left(-[\overline{u^2}^{12}]_{i+1/2,j-1/2}\right.
\end{aligned}$$

$$+[\overline{u^2}^{12}]_{i+1/2,j+3/2}-[\overline{u^2}^{12}]_{i+3/2,j-1/2}+[\overline{u^2}^{12}]_{i+3/2,j+3/2}\big)$$
$$+\frac{1}{192}[\overline{u^2}^{2}]_{i+1}\big(-\big[\overline{C_1^1}^{12}\big]_{i+1/2,j-1/2}+\big[\overline{C_1^1}^{12}\big]_{i+1/2,j+3/2}$$
$$-\big[\overline{C_1^1}^{12}\big]_{i+3/2,j-1/2}+\big[\overline{C_1^1}^{12}\big]_{i+3/2,j+3/2}\big)\big(-[\overline{u^1}^{12}]_{i+1/2,j-1/2}$$
$$+[\overline{u^1}^{12}]_{i+1/2,j+3/2}-[\overline{u^1}^{12}]_{i+3/2,j-1/2}+[\overline{u^1}^{12}]_{i+3/2,j+3/2}\big)+\mathrm{O}(h^4). \quad (2.41)$$

Thus, to close the approximation it is only necessary to choose an interpolation method, of the same order of accuracy as the interpolation of the variables, for the geometry and compute the geometrical data appearing in the discrete equations. For instance, to save computing time, the cofactor

$$\overline{C_1^1}^{2}$$

can be computed using the Lagrange method for the cell averages,

$$\overline{C_1^1}^{12}=\frac{1}{\Delta x^1}\frac{1}{\Delta x^2}\int_\theta \frac{\partial y^2}{\partial x^2}. \quad (2.42)$$

The latter can be computed in the preprocessing stage and stored for later use, by using an interpolation method for the Cartesian coordinates and numerically integrating the resulting interpolating function. The resulting approximation of the cell-averaged cofactor should be of the same order of accuracy as the interpolation for the variables. The remaining terms are discretized analogously.

Although the length of the expressions increases, the general ideas developed for the Cartesian grid can still be applied. In particular, we solve for one direction at a time, and for each strip in the computational space we invert only tri-diagonal systems of equations.

After approximation of the continuum problem with a discrete one, the next issue is related to the procedure for solving the resulting set of nonlinear equations. This is the subject of the next section.

2.2. The Stationary Navier–Stokes Equations

Substituting the discretized terms, Eqs. (2.6)–(2.10), into Eqs. (2.3) and (2.5) yields the following nonlinear coupled system of equations:

$$([\overline{u}^{y}]_{i+1}-[\overline{u}^{y}]_i)\Delta y+([\overline{v}^{x}]_{j+1}-[\overline{v}^{x}]_j)\Delta x=0 \quad (2.43)$$

$$([\overline{uu}^{y}]_{i+1}-[\overline{uu}^{y}]_i)\Delta y+([\overline{vu}^{x}]_{j+1}-[\overline{vu}^{x}]_j)\Delta x$$
$$=\nu\left[\left(\left[\frac{\overline{\partial u}^{y}}{\partial x}\right]_{i+1}-\left[\frac{\overline{\partial u}^{y}}{\partial x}\right]_i\right)\Delta y+\left(\left[\frac{\overline{\partial u}^{x}}{\partial y}\right]_{j+1}-\left[\frac{\overline{\partial u}^{x}}{\partial y}\right]_j\right)\Delta x\right]$$
$$-([\overline{p}^{y}]_{i+1}-[\overline{p}^{y}]_i)\Delta y \quad (2.44)$$

$$([\overline{uv}^{y}]_{i+1}-[\overline{uv}^{y}]_i)\Delta y+([\overline{vv}^{x}]_{j+1}-[\overline{vv}^{x}]_j)\Delta x$$
$$=\nu\left[\left(\left[\frac{\overline{\partial v}^{y}}{\partial x}\right]_{i+1}-\left[\frac{\overline{\partial v}^{y}}{\partial x}\right]_i\right)\Delta y+\left(\left[\frac{\overline{\partial v}^{x}}{\partial y}\right]_{j+1}-\left[\frac{\overline{\partial v}^{x}}{\partial y}\right]_j\right)\Delta x\right]$$
$$-([\overline{p}^{x}]_{j+1}-[\overline{p}^{x}]_j)\Delta x. \quad (2.45)$$

These must be solved together with the compact representation of the fluxes.

The previous system of equations is nonlinear, coupled, and degenerate—in the sense that p is not unique.

This nonlinear system of equations can be solved in several ways. In the present work we consider a coupled implicit method. So, because of the complexity of the relations between the fluxes and the primitive cell averaged variables and to avoid inverting the relatively dense pattern of the resulting system of equations the Newton–Krylov method with a matrix-free technique is selected. This is explained in the next section.

2.3. The Newton–Krylov Matrix-Free Method

Roughly speaking, Krylov methods (such as GMRES and BI-CGSTAB) are based on the minimization of the residual in a Krylov space. These methods are appropriate for general sparse matrices. One important feature of them is the fact that only matrix–vector products and obviously vector–vector products are required.

Given the discretization above, which we write symbolically as

$$\mathbf{N}(\mathbf{x}) = 0, \tag{2.46}$$

the corresponding Newton's method proceeds as

$$\mathbf{J}(\mathbf{x_i})\delta\mathbf{x_i} = -\mathbf{N}(\mathbf{x_i}), \tag{2.47}$$

where $\mathbf{J}(\mathbf{x_i})$ is the Fréchet derivative of $\mathbf{N}$ at the point $\mathbf{x_i}$ and

$$\mathbf{x_{i+1}} = \mathbf{x_i} + \delta\mathbf{x_i}. \tag{2.48}$$

The matrix–vector products appearing in the Krylov process are of the type $\mathbf{J}(\mathbf{x_i})\ \mathbf{p}$, for some vector $\mathbf{p}$. The product $\mathbf{J}(\mathbf{x_i})\ \mathbf{p}$ can be represented as the Gateaux derivative in the direction of the vector $\mathbf{p}$, that is,

$$\mathbf{J}(\mathbf{x_i})\mathbf{p} = \frac{\partial \mathbf{N}}{\partial \mathbf{p}}(\mathbf{x_i}), \tag{2.49}$$

where, by definition, the Gateaux derivative $\frac{\partial \mathbf{N}}{\partial \mathbf{p}}(\mathbf{x_i})$ is given by

$$\frac{\partial \mathbf{N}}{\partial \mathbf{p}}(\mathbf{x_i}) = \lim_{t\to 0} \frac{\mathbf{N}(\mathbf{x_i} + t\mathbf{p}) - \mathbf{N}(\mathbf{x_i})}{t}. \tag{2.50}$$

Note that to avoid introducing rounding errors in the computation we normalize the direction and compute $\mathbf{J}(\mathbf{x_i})\ \mathbf{p} = \frac{\partial \mathbf{N}}{\partial \hat{\mathbf{p}}}(\mathbf{x_i})\ \|\mathbf{p}\|_2$ with $\hat{\mathbf{p}} = \mathbf{p}/\|\mathbf{p}\|_2$.

Grosso modo, the Fréchet derivative, conveys the information on the variation of vector function $\mathbf{N}$ in any direction, whereas the Gateaux derivative measures the rate of variation of this function in a given direction only.

We can use these observations to devise a matrix–free Newton–Krylov method, that is, a Newton-Krylov method where the Fréchet derivative is not needed and consequently not stored. Thus, the equation is linearized using the Newton method. Then, we solve the resulting linear system of equations with a Krylov method. To bypass the computation of the Fréchet derivative we note that: (a) formally in any Krylov method, what is required is not the Fréchet matrix, but rather its product with some vector; (b) this product is equivalent

to the Gateaux derivative times the norm of the given vector, as mentioned previously; and, finally, (c) for practical purposes, the Gateaux derivative can be approximated by the unilateral finite difference

$$\frac{\partial \mathbf{N}}{\partial \hat{\mathbf{p}}}(\mathbf{x_i}) \cong \frac{\mathbf{N}(\mathbf{x_i} + \varepsilon \hat{\mathbf{p}}) - \mathbf{N}(\mathbf{x_i})}{\varepsilon} \tag{2.51}$$

for some small $\varepsilon > 0$. So, every time a product $\mathbf{J}(\mathbf{x_i})\ \mathbf{p}$ is needed in the Krylov method, we estimate the Gateaux derivative by using Eq. (2.51) and compute the product by using Eq. (2.49).

Notwithstanding the fact that the procedure described above can be used to find the desired root of the nonlinear problem, it is well known that to be useful for a wide range of problems, the Krylov methods require a preconditioning method. Moreover, this preconditioning method determines the overall efficiency of the Krylov method, and the usual ILU preconditioning uses the matrix structure.

Recently, however, Brown and Saad [16] introduced the notion of Flexible-GMRES (FGMRES). These methods consist of the minimization of the residual on a convenient subspace, which allows the use of any partial solver as preconditioning. For instance, they considered the FGMRES method together with the GMRES method as the preconditioning (FGMRES/GMRES). They compared the FGMRES/GMRES method with the conventional GMRES/ILU(0) and in the problems studied, both methods displayed a similar convergence history. In general, the method will converge if the steepest descent direction belongs to the subspace where the minimization takes place.

The use of a Krylov method as the preconditioning method for the FGMRES is the key for an efficient Newton–Krylov matrix-free method. In the present work, we solve the nonlinear system of equations (2.43) to (2.45) by using the Newton–Krylov matrix-free method, with the FGMRES/GMRES. Next, we briefly describe the method and the preconditioning technique used in the computations.

Let us consider the GMRES method for the solution of a general linear system of equations,

$$AM^{-1}(Mx) = b, \tag{2.52}$$

where A stands for the matrix of coefficients, M is the preconditioner, x is the solution vector, and b is the nonhomogeneous term. In the Arnoldi process of the GMRES, an orthonormal basis is constructed for the Krylov subspace

$$\text{span}\{\mathrm{r}_0, AM^{-1}\mathrm{r}_0, \ldots, (AM^{-1})^{\mathrm{m}}\mathrm{r}_0\}$$

by using a modified Gram–Schmidt method. In this process some vectors $z_j = M^{-1}v_j$, for some convenient v_j are generated and the solution is sought as a linear combination of them. Note that the preconditioner is the same for all z_j. Now, in the FGMRES to allow for a greater flexibility in the preconditioning process within the GMRES, the preconditioning may vary from one z_j to the other; that is, in the FGMRES we have $z_j = M_j^{-1}v_j$. Indeed, the residual will be minimized in this space. So, in the FGMRES/GMRES used in the present work to compute the z_j, we solve the following system of equations

$$Az_j = v_j$$

using the GMRES for some prescribed convergence level.

At this point it is useful to summarize the procedure as an algorithm.

1. **Start**. Initialize the variables $\mathbf{x_0} = (\mathbf{u_0}, \mathbf{v_0}, \mathbf{p_0})$. Choose the dimension of the Krylov space m. Set $i \leftarrow 0$.
2. **Newton–Krylov Method:**
   (a) Set $_0 \leftarrow 0$.
   (b) **Arnoldi process:**
      i. Compute $\mathbf{r}_0 = -\mathbf{N}(\mathbf{x_i} + \xi_\mathbf{0})$, $\beta = \|\mathbf{r_0}\|_2$, and $\mathbf{v_1} = \frac{\mathbf{r_0}}{\beta}$. Define an $(m+1) \times$ m matrix $\bar{\mathbf{H}}_\mathbf{m}$ and initialize all its entries $h_{k,j} = 0$.
      ii. For j $= 1, \ldots,$ m, do
      - Solve to a prescribed level $\mathbf{J}(\mathbf{x_i})\mathbf{z_j} = \mathbf{v_j}$;
      - Compute $\mathbf{w} = \mathbf{J}(\mathbf{x_i})\mathbf{z_j}$;
      - For k $= 1, \ldots,$ j, do $\begin{cases} h_{k,j} = (\mathbf{w}, \mathbf{v_k}), \\ \mathbf{w} = \mathbf{w} - h_{k,j}\mathbf{v_k}; \end{cases}$
      - Compute $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v_{j+1}} = \frac{\mathbf{w}}{h_{j+1,j}}$.

      iii. Define $\mathbf{Z}_m = [\mathbf{z_1}, \ldots, \mathbf{z_m}]$.
   (c) **Form the approximate solution**: Compute $\xi_\mathbf{m} = \xi_\mathbf{0} + \mathbf{Z_m}\mathbf{y_m}$, where $\mathbf{y_m} = \arg\min_y \|\beta\mathbf{e_1} - \bar{\mathbf{H}}_\mathbf{m}\mathbf{y}\|_2$ and $e_1 = [1, 0, \ldots, 0]^T$.
   (d) **Restart**. If satisfied exit the Arnoldi process, else set $\xi_\mathbf{0} \leftarrow \xi_\mathbf{m}$ and go to step b.
3. **Update**. Set $(\mathbf{u_{i+1}}, \mathbf{v_{i+1}}, \mathbf{p_{i+1}}) = \mathbf{x_{i+1}} \leftarrow \mathbf{x_i} + \xi_\mathbf{m}$. If satisfied stop, else go to step 2.

At this point we should stress that the vector function **N** used in the Newton method is nothing but the discretized continuity and Navier–Stokes equations. Therefore, each product $\mathbf{J}(\mathbf{x_i})\,\mathbf{q}$, for some vector **q**, in the Newton–Krylov step requires only the computation of the balance equation with the vector $\mathbf{x_i} + \varepsilon\mathbf{q}$ (compare with Eq. (2.51)).

Because all equations in the full coupled system are forced to be simultaneously satisfied, the method is very robust and can be applied to any Reynolds number. As we said, the matrix representation of the Fréchet derivative is dense. Hence, it is very hard to implicitly solve the Newton's equations without the present techniques.

Remark 2.2. The Newton method can be directly applied to the stationary equations. However, as is well known, the Newton method requires a "satisfactory" guess of the solution as the starting point. There are several ways to overcome this problem, for instance, a line minimization of the residual and the iterative increment of the Reynolds number. In the present work, we obviate this problem by using the implicit first-order Euler method to solve a pseudo-temporal problem. That is, starting with $\boldsymbol{v}^0$ we solve the nonlinear (NL) problem,

Find $(\bar{\mathbf{v}}^{n+1}, \bar{p}) \in \mathbf{R}^{2N} \times \mathbf{R}^N$, *where N is the number of unknowns in the mesh, such that*

$$\left([\bar{u}^y]_{i+1}^{n+1} - [\bar{u}^y]_i^{n+1}\right)\Delta y + \left([\bar{v}^x]_{j+1}^{n+1} - [\bar{v}^x]_j^{n+1}\right)\Delta x = 0 \tag{2.53}$$

$$\bar{\mathbf{v}}_{ij}^{n+1} = \bar{\mathbf{v}}_{ij}^{n} + \frac{\Delta t}{m_V(\theta_{ij})}(-\mathsf{C}_{ij}(\mathbf{v}^{n+1}) + \mathsf{D}_{ij}(\mathbf{v}^{n+1}) - \mathsf{G}_{ij}\,p^{n+1}) \tag{2.54}$$

and $(\bar{\mathbf{v}}^{n+1}, \bar{p}) \in \mathbf{R}^{2N} \times \mathbf{R}^N$ *satisfies some boundary conditions.*

In this expression, $m_V(\theta_{ij})$ denotes the Lebesgue (volume) measure of the control volume θ_{ij}. For a sufficiently small Δt, the NL problem has a unique solution and the Newton method converges quadratically toward it. We prove it by using the Banach fixed point theorem for complete metric spaces and Theorem 6.3 in Girault and Raviart [17].

Thus, let V denote the Banach space of discrete averaged vector fields associated with a given grid, and V_0 the subspace of V that are zero at the domain boundary $\partial\Omega$. Let S denote the linear space of discrete scalar fields associated with a given grid. Define the conservative divergence operator $\mathsf{M}: V \to S$ as

$$\mathsf{M}_{ij}(\mathbf{v}) = \frac{([\overline{u}^y]_{i+1} - [\overline{u}^y]_i)\Delta y + ([\overline{v}^x]_{j+1} - [\overline{v}^x]_j)\Delta x}{m_V(\theta_{ij})} \tag{2.55}$$

for all $\mathbf{v} = (\overline{u}^y, \overline{v}^x) \in V$. Similarly, define the conservative gradient operator $\mathsf{G_m}: S \to V$ as

$$(\mathsf{G_m})_{ij}(\phi) = \frac{(([\overline{\phi}^y]_{i+1} - [\overline{\phi}^y]_i)\Delta y, ([\overline{\phi}^x]_{j+1} - [\overline{\phi}^x]_j)\Delta x)}{m_V(\theta_{ij})}$$

for all $\phi \in S$. Then, we have the following decomposition: $V_0 = (V_0 \cap \ker\mathsf{M}) \oplus (V_0 \cap \mathrm{im}\mathsf{G_m})$. Indeed, given a $\nu_0 \in (V_0 \cap \ker\mathsf{M}) \oplus (V_0 \cap \mathrm{im}\mathsf{G_m})$, it follows that there exists a $\phi \in S$, such that $\nu_0 = \mathsf{G_m}\phi$ and $\mathsf{MG_m}\phi = 0$, or $((\mathsf{D}_x)^2 + (\mathsf{D}_y)^2)(\phi) = 0$, where, for instance,

$$(\mathsf{D}_x)_{ij}(\phi) = \frac{([\overline{\phi}^y]_{i+1} - [\overline{\phi}^y]_i)}{\Delta x},$$

for all $\phi \in S$. A discrete version of the separation of variables approach yields the eigenvalue problems $(\mathsf{D}_x)^2(\varphi) = \lambda\varphi$ and $(\mathsf{D}_y)^2(\tilde{\varphi}) = -\lambda\tilde{\varphi}$, where $\phi(ij) = \varphi(i)\tilde{\varphi}(j)$. Now, the eigenvalues of, for instance, $(\mathsf{D}_x)^2(\varphi) = \lambda\varphi$ are all negative because it is the square of an antisymmetric operator. Then $\nu_0 = 0$. Now, given a $\nu \in V_0$, we write $\nu = w + \mathsf{G_m}\phi$, and determine $\phi \in S$ from $\mathsf{M}\nu = \mathsf{MG_m}\phi$. Then, $w = \nu - \mathsf{G_m}\phi$ as required.

Let $P: V_0 \to V_0 \cap \ker\mathsf{M}$ be the (oblique) projection onto $V_0 \cap \ker\mathsf{M}$. Let $\mathbf{v}_d \in \ker\mathsf{M}$ be a discrete solenoidal vector field that satisfies the boundary conditions. Define on V_0 the function $\mathsf{F}: V_0 \to V_0$

$$\mathsf{F} = \mathrm{id}_{V_0} + \Delta t\mathrm{P}(-\mathsf{C_m} + \mathsf{D_m} + g) - \mathbf{v}_0^n \tag{2.56}$$

for some function $g: \mathrm{V}_0 \to \mathrm{V}_0$, and where the maps $\mathsf{C_m}$ and $\mathsf{D_m}$ are the conservative convection and diffusion finite differences maps; for example,

$$(\mathsf{C_m})_{ij} = \frac{\mathsf{C_{ij}}}{m_V(\theta_{ij})}.$$

Hence, by putting $\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_d$ we conclude that the NL problem is equivalent to finding a root $\mathbf{v}_0 \in \mathrm{V}_0$ of F with a function g given by the remaining terms in (2.54) after substituting $\mathbf{v} = \mathbf{v}_0 + \mathbf{v}_d$ in this equation. Now, applying the Newton method yields

$$\mathbf{v}_0^{k+1} = \mathbf{v}_0^k - (D\mathsf{F})^{-1}_{\nu_0^k}\big(\mathsf{F}\big(\mathbf{v}_0^k\big)\big), \tag{2.57}$$

where the superscript k denotes a Newton iteration (not a time step). Solution of (2.57) is

equivalent to finding a fixed point for the function

$$\mathsf{H} = \mathrm{id}_{V_0} - (D\mathsf{F})^{-1}(\mathsf{F}). \tag{2.58}$$

To prove the existence and uniqueness of the fixed point we show that H is a contraction on the closure of the open ball $B_r(\mathbf{v}_0^n)$ for some $0 < r$. From the definition of F we have

$$D\mathsf{F} = \mathrm{id}_{V_0} + \Delta t\, D\mathsf{K},$$

where

$$\mathsf{K} = \mathsf{P}(-\mathsf{C}_\mathsf{m} + \mathsf{D}_\mathsf{m} + g).$$

For sufficiently small Δt, it follows that

$$\left\| \Delta t\, D\mathsf{K}_{|B_r(\nu_0^n)} \right\| < 1$$

and so $D\mathsf{F}$ is invertible in the Banach algebra of linear operators on V_0 for all $\nu \in B_r(\mathbf{v}_0^n)$. Moreover, we have

$$(D\mathsf{F})^{-1} = \mathrm{id}_{V_0} - \Delta t\, D\mathsf{K} + \mathrm{o}(\Delta t\, D\mathsf{K}).$$

Then, for arbitrary $\mathbf{u}, \mathbf{v} \in B_r(\mathbf{v}_0^n)$ we have

$$\begin{aligned} \|\mathsf{H}(\mathbf{u}) - \mathsf{H}(\mathbf{v})\| &= \|\mathbf{u} - \mathbf{v} - (\mathsf{F}(\mathbf{u}) - \mathsf{F}(\mathbf{v}) - \Delta t\, D\mathsf{K}(\mathsf{F}(\mathbf{u}) - \mathsf{F}(\mathbf{v}))) + \mathrm{o}(\Delta t(\mathbf{u} - \mathbf{v}))\| \\ &\leq C\Delta t \|\mathbf{u} - \mathbf{v}\|, \end{aligned}$$

where C is a positive real constant, and we have used the smoothness of F, H, and K (they are vector functions with polynomial entries). For a sufficiently small Δt, it follows that $C\Delta t < 1$. Also, given a $\mathbf{u} \in B_r(\mathbf{v}_0^n)$ we have

$$\begin{aligned} \left\|\mathsf{H}(\mathbf{u}) - \mathbf{v}_0^n\right\| &\leq \left\|\mathsf{H}(\mathbf{u}) - \mathsf{H}\left(\mathbf{v}_0^n\right)\right\| + \left\|\mathsf{H}\left(\mathbf{v}_0^n\right) - \mathbf{v}_0^n\right\| \\ &\leq \left\|\mathbf{u} - \mathbf{v}_0^n\right\| + \Delta t L \end{aligned}$$

for some positive constant L. For a sufficiently small Δt, this shows that $\mathsf{H}(B_r(\mathbf{v}_0^n)) \subset B_r(\mathbf{v}_0^n)$ and, so by continuity of H it follows that the closure of the ball $\mathrm{cl}(B_r(\mathbf{v}_0^n))$ is invariant for H. Now the Banach fixed point theorem gives the existence and uniqueness of the fixed point in $\mathrm{cl}(B_r(\mathbf{v}_0^n))$. Finally, the convergence characteristics of the Newton method follow at once from the smoothness of F (for instance, apply Theorem 6.3 in Girault and Raviart [17]). So, by controlling the CFL (possibly during the pseudo-time evolution) we can ensure convergence as well as accelerate it.

2.4. Pressure–Velocity Coupling

Since the maxtrix-free approach allows a coupled solution of the discrete Navier–Stokes and continuity equations to be taken without too many memory requirements, the natural

option for solving the complete coupled system of equations was followed. Pressure–velocity coupling is, then, automatically guaranteed.

We use a centered method in a collocated grid. Therefore, it is natural to inquire about the possibility of the existence of a solution with a spurious pressure oscillation pattern. Clearly, the elimination of the spurious pressure oscillation is equivalent to $\dim\ker G = 1$. For then, the only solution of $Gp = 0$ is p constant. It is easy to see that with the fourth-order compact finite volume method, $\dim\ker G = 1$ on uniform grids with an odd number of control volumes and periodic or Dirichlet boundary conditions. Also, for all the types of boundary conditions and meshes that we have used in the computations, the geometric multiplicity of the null eigenvalue was 1, which in turn yields $\dim\ker G = 1$. Finally, it should be noted that for all test cases that follow, the pressure fields revealed a pattern free from spurious oscillations.

2.5. Deconvolution of the Mean Fields

As mentioned above, we store and solve for the cell average values of the primitive variables. In a fine grid, the latter are close to point values. However, sometimes we need the local point values of the physical quantities, for example, to plot the pressure distribution. We recover the point values from the predicted mean values by a "deconvolution" technique. For simplicity, we use an explicit deconvolution where the point values, stored in the mesh vertices, are obtained from the existing mean values at the cell faces γ_i. So, to obtain fourth-order-accurate point values of $(u,\ v,\ p)$, the general procedure, for example, for u in the x direction, consists of computing coefficients $\{b_i\}_{i=1,2}$ such that

$$u_i = \sum_{i=1}^{2} b_i \tau_{(-i+\frac{1}{2})\Delta x}\overline{u}^x + \sum_{i=1}^{2} b_i \tau_{(i-\frac{1}{2})\Delta x}\overline{u}^x + \mathrm{O}(h^4). \tag{2.59}$$

Solution of this problem gives $b_1 = 7/12$, $b_2 = -1/12$. We remark that this is done in postprocessing, that is, after the computation of the flow field is finished.

2.6. The Time-Dependent Navier–Stokes Equations

Simulation of an unsteady laminar flow is carried out using the fourth-order Runge–Kutta method for the semidiscrete form of the equations. That is, starting with $\mathbf{v}^0$ we compute

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4) - \Delta t \mathsf{G} p^{n+1} \tag{2.60}$$

$$\mathsf{M}\mathbf{v}^{n+1} = 0, \tag{2.61}$$

where

$$k_i = (-\mathsf{C} + \mathsf{D})(\mathbf{v}_i), \quad i = 1, \ldots, 4 \tag{2.62}$$

and

$$\mathbf{v}_1 = \mathbf{v}^n \tag{2.63}$$

$$\begin{cases} \mathbf{v}_2 = \mathbf{v}^n + \dfrac{\Delta t}{2}(-\mathsf{C}+\mathsf{D})(\mathbf{v}_1) - \Delta t \mathsf{G} p_2 \\ \mathsf{M}\mathbf{v}_2 = 0 \end{cases} \tag{2.64}$$

$$\begin{cases} \mathbf{v}_3 = \mathbf{v}^n + \dfrac{\Delta t}{2}(-\mathsf{C}+\mathsf{D})(\mathbf{v}_2) - \Delta t \mathsf{G} p_3 \\ \mathsf{M}\mathbf{v}_3 = 0 \end{cases} \tag{2.65}$$

$$\begin{cases} \mathbf{v}_4 = \mathbf{v}^n + \Delta t(-\mathsf{C}+\mathsf{D})(\mathbf{v}_3) - \Delta t \mathsf{G} p_4 \\ \mathsf{M}\mathbf{v}_4 = 0, \end{cases} \tag{2.66}$$

where

$$(\mathbf{v}_i, p_i) \quad i = 2, 3, 4, n+1$$

are solved in a coupled way with the Krylov matrix-free method explained above.

3. NUMERICAL RESULTS

This section presents numerical results for several test cases aimed at assessing the accuracy and efficiency of the proposed method for steady or unsteady flow problems.

3.1. Analytical 2D Cavity

In this test case, we consider the recirculating viscous flow in a square cavity driven by combined shear and body forces. Figure 2 shows schematically the geometry of the problem and boundary conditions for the velocity field. This benchmark test case appeared in Shih

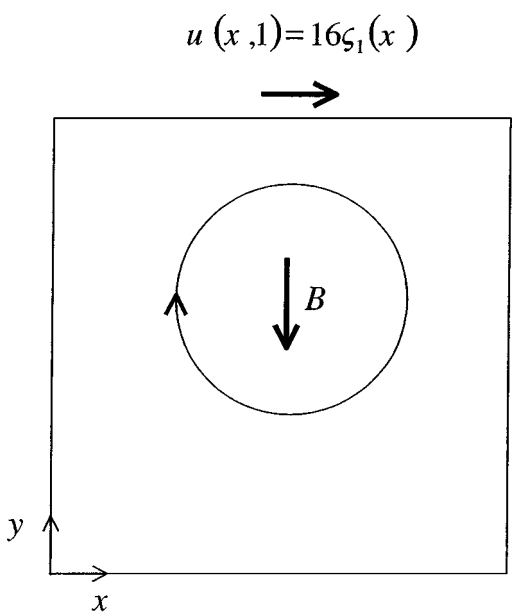


FIG. 2. Geometry of Problem 3.1 and boundary conditions.

et al. [18], where the details of the problem can also be found. Here we summarize the relevant information. The vertical body force is given by the expression

$$B(x, y; \mathrm{Re}) = \frac{8}{\mathrm{Re}}\left[24 \int \zeta_1(x) + 2\zeta_1'(x)\zeta_2''(y) + \zeta_1'''(x)\zeta_2(y)\right] - 64[\mathrm{Y}_2(x)\mathrm{Y}_3(y) - \zeta_2(y)\zeta_2'(y)\mathrm{Y}_1(x)],$$

where

$$\begin{aligned}\zeta_1(x) &= x^4 - 2x^3 + x^2\\ \zeta_2(y) &= y^4 - y^2\\ \mathrm{Y}_1(x) &= \zeta_1(x)\zeta_1''(x) - [\zeta_1'(x)]^2\\ \mathrm{Y}_2(x) &= \int \zeta_1(x)\zeta_1'(x)\\ \mathrm{Y}_3(y) &= \zeta_2(y)\zeta_2'''(y) - \zeta_2'(y)\zeta_2''(y)\end{aligned}$$

for all $(x, y) \in \Omega \equiv [0, 1]^2$. The Dirichlet boundary conditions correspond to zero velocity at all boundaries except for the top surface where

$$u(x, 1) = 16\zeta_1(x), \quad x \in [0, 1].$$

An exact solution for this problem exists and is known to be

$$\begin{aligned}u(x, y) &= 8\zeta_1(x)\zeta_2'(y)\\ v(x, y) &= -8\zeta_1'(x)\zeta_2(y)\end{aligned}$$

and

$$p(x, y; \mathrm{Re}) = \frac{8}{\mathrm{Re}}\left[24\left(\int \zeta_1(x)\right)\zeta_2'''(y) + 2\zeta_1'(x)\zeta_2'(y)\right] + 64\mathrm{Y}_2(x)\{\zeta_2(y)\zeta_2''(y) - [\zeta_2(y)]^2\}.$$

We have considered four meshes, and the parameters for the simulation are $\mathrm{Re} = 1$ and $u_{\mathrm{ref}} = \ell_{\mathrm{ref}} = 1$. Note that this benchmark has the particularity of having an equal flow field pattern independent of the Reynolds number. Tables I and II list the errors and numerical

TABLE I
Error Norm(L_1) Dependence on Mesh Size for Test Case 3.1: Pontual Values

	U		V		P	
Grid	Error (L_1)	Order	Error (L_1)	Order	Error (L_1)	Order
7×7	6.87E-4	—	1.94E-4	—	5.92E-3	—
15×15	3.10E-5	4.07	6.80E-6	4.40	2.56E-4	4.12
31×31	1.66E-6	4.03	3.28E-7	4.18	1.45E-5	3.95
63×63	9.64E-8	4.01	1.81E-8	4.09	8.85E-7	3.94

TABLE II
Error Norm (L_1) Dependence on Mesh Size for Test Case 3.1: Mean Values

	U		V		P	
Grid	Error (L_1)	Order	Error (L_1)	Order	Error (L_1)	Order
7×7	2.02E-4	—	1.03E-4	—	7.23E-3	—
15×15	8.10E-6	4.22	4.76E-6	4.03	3.63E-4	3.93
31×31	4.48E-7	3.99	2.58E-7	4.02	2.03E-5	3.97
63×63	2.63E-8	4.00	1.52E-8	3.99	1.33E-6	3.84

order of accuracy for point and mean values of the variables, respectively. Fourth-order accuracy is achieved for both the cell averages and the deconvoluted point values.

3.2. Lamb–Oseen Vortex

The Lamb–Oseen vortices are very often used to model aircraft wake vortices decay and motion in the atmosphere. Because of the large disparity of scales between the vortex core radius and computational domain, a numerical method of high-order accuracy is required, or otherwise numerical dissipation corrupts the solution.

The exact inviscid tangential velocity and pressure solutions are

$$v_\theta(r) = \frac{\Gamma_0}{2\pi r}\left(1 - e^{-\beta\left(\frac{r}{r_{c0}}\right)^2}\right)$$

$$p(r) = -\frac{1}{8\pi^2 r^2 {r_{c0}}^2}\left(r_{c0}{}^2\Gamma_0^2\rho\left(-1 - e^{\frac{-2\beta r^2}{r_{c0}{}^2}} + 2e^{\frac{-\beta r^2}{r_{c0}{}^2}}\right)\right.$$
$$\left. - 2\beta\Gamma_0^2 r^2\rho\left(\mathrm{EI}\left(\frac{-2\beta r^2}{r_{c0}{}^2}\right) - \mathrm{EI}\left(\frac{-\beta r^2}{r_{c0}{}^2}\right)\right)\right),$$

where (r, θ) are the polar coordinates, and EI is the exponential integral defined as

$$\mathrm{EI}(z) = -\int_{-z}^{\infty} \frac{e^{-t}}{t}\, dt.$$

The vortex parameters used in the computations are $\Gamma_0 = 250$, $r_{c0} = 3$, and $\beta = 1.25643$, respectively, the circulation, core radius, and a constant. A low Reynolds number

$$\mathrm{Re} = \frac{\Gamma_0}{\nu} = \frac{250}{56.8 \times 10^{-3}} = 4.4 \times 10^3$$

was selected. A source term equivalent to the viscous fluxes was added to the momentum equations. The computational domain is $\Omega \equiv [-10, 10]^2$, and Dirichlet conditions are used in all boundaries.

TABLE III
Error Norm (L_1) Dependence on Mesh Size for Test Case 3.2

	U		V		P	
Grid	Error (L_1)	Order	Error (L_1)	Order	Error (L_1)	Order
5×5	2.80E-1	—	2.80E-1	—	3.81	—
11×11	4.44E-2	2.34	4.44E-2	2.34	2.20E-1	3.62
21×21	1.20E-3	5.58	1.20E-3	5.58	1.10E-2	4.63
41×41	7.50E-5	4.14	7.50E-5	4.14	7.05E-4	4.11

Table III lists the error norms of the predicted velocity and pressure fields for different grids showing that the numerical solution is fourth-order accurate. To see the implications of the relative magnitude of the errors, it is convenient to say that maximum velocity and pressure are equal to $v_\theta = 9.47$ m/s and $p = 153.2$ N/m^2, respectively. Relative to these values, the errors (L_1 norm) are less than 0.5% on an 11×11 mesh, which comprises only three control volumes in the vortex core radius. Figure 3 shows the velocity profiles for all the meshes considered. The prediction suggests that, using the fourth-order compact scheme, a minimum of three points in the vortex core radius is necessary to get accurate solutions.

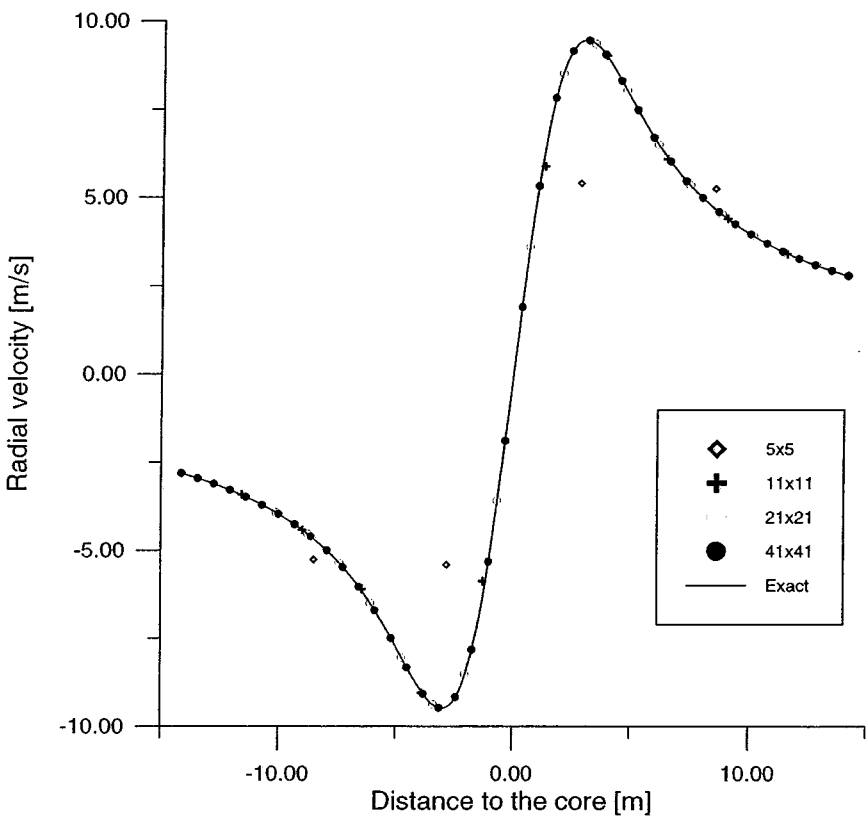


FIG. 3. Lamb–Oseen vortex radial velocity predictions.

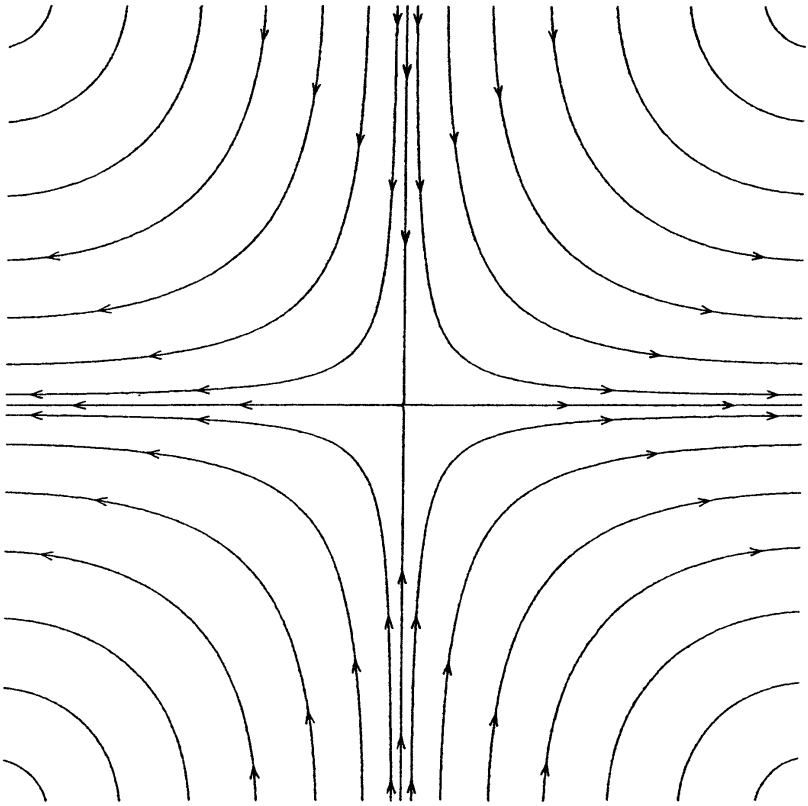


FIG. 4. Streamlines for test case 3.3.

3.3. Analytical Vortex Decay

The purpose of this test case is to validate unsteady Navier–Stokes solutions in a 2-D domain. The exact solution of the Taylor vortex appears as

$$\mu(x, y, t; \mathrm{Re}) = -\cos(x)\sin(y)e^{\frac{-2t}{\mathrm{Re}}}$$

$$v(x, y, t; \mathrm{Re}) = \sin(x)\cos(y)e^{\frac{-2t}{\mathrm{Re}}}$$

$$p(x, y, t; \mathrm{Re}) = -\frac{1}{4}(\cos(2x) + \cos(2y))e^{\frac{-4t}{\mathrm{Re}}}$$

for which the domain $\Omega \equiv [0, \pi]^2$ was used. This domain provides inflow and outflow in all boundaries and Dirichlet boundary conditions are prescribed.

The final time is $T = 0.34657\,\mathrm{Re}$. This corresponds to a decay in the velocity field equal to half their initial values.

The parameters for the simulation are $\mathrm{Re} = 100$ and $u_{\mathrm{ref}} = \ell_{\mathrm{ref}} = 1$. The CFL is kept constant; $\mathrm{CFL} = 1/8$ for all grids. As an illustration, Fig. 4 shows the velocity field topology in which the streamlines maintain its position with the time evolution.

Table IV lists the error norms of the predicted velocity and pressure variables for the different grids and clearly shows that the numerical solution is fourth-order accurate in space and time.

TABLE IV
Error Norm (L_1) Dependence on Mesh Size for Test Case 3.3

	U		V		P	
Grid	Error (L_1)	Order	Error (L_1)	Order	Error (L_1)	Order
7×7	1.11E-4	—	1.23E-4	—	7.44E-4	—
15×15	3.77E-6	4.44	4.06E-6	4.48	7.73E-5	2.97
31×31	1.51E-7	4.43	1.53E-7	4.52	4.80E-6	3.83

3.4. Classical Lid-Driven Cavity Flow

This test case is selected to evaluate the method performance under recirculating flows. This benchmark is classical (see, e.g., Ghia *et al.* [19]). Recently, a spectral solution appeared [20] of this flow test case, which we use as the reference solution.

The square domain $\Omega \equiv [0, 1]^2$ was discretized with several uniform meshes and Dirichlet boundary conditions and corresponds to zero velocity at all boundaries except for the top surface where $u = 1$. The parameters for the simulation are $\mathrm{Re} = 1000$ and $u_{\mathrm{ref}} = \ell_{\mathrm{ref}} = 1$.

For the purpose of comparison, alongside the fourth-order compact finite volume method, additional results were obtained with the third-order Quick scheme [21] for convective terms and central second-order discretization for the remaining ones. The latter combination is commonly used in engineering problems involving incompressible fluid flows. This method was implemented in the same way as the compact method.

Figures 5a and 5b display the u and v profiles in the vertical and horizontal middle lines, respectively. An excellent agreement with the spectral results is observed even for the coarse grids. The Quick scheme requires 120×120 mesh control volumes to obtain solutions similar to those obtained with the proposed compact method with only 30×30 grid control volumes. However, the solution with the 30×30 mesh of the present method requires only 5% of the CPU time required by Quick (120×120) for the same parameters. Furthermore, the compact method with the 80×80 mesh still requires less computing time than the Quick scheme (120×120).

For the purpose of analyzing the convergence properties of the present Newton–Krylov matrix-free compact method we solve the lid-cavity flow with $\mathrm{Re} = 1000$ for CFL numbers equal to 10, 100, and 1000. Figures 6a, 6b, and 6c present the convergence history of the method for these CFL numbers and for a combination of Krylov subspace dimensions in the solver and preconditioner.

For the low CFL number, the time evolution controls the convergence process, and a quasi-physical evolution is obtained. So, for all combinations of dimensions of the Krylov subspace in the solver or in the preconditioning the same evolution is obtained. As the CFL grows, the required number of iterations for convergence decreases, as long as the linear system of equations is well resolved. The latter depends upon the CFL also. As expected, the preconditioning process greatly influences the solver, making the convergence history strongly dependent on it.

For optimal results, a CFL increasing strategy (linear or inversely proportional to the residual decay) should be used. Starting with a small CFL avoids the initial oscillations associated with high CFL numbers. As the steady state is approximated, super-linear

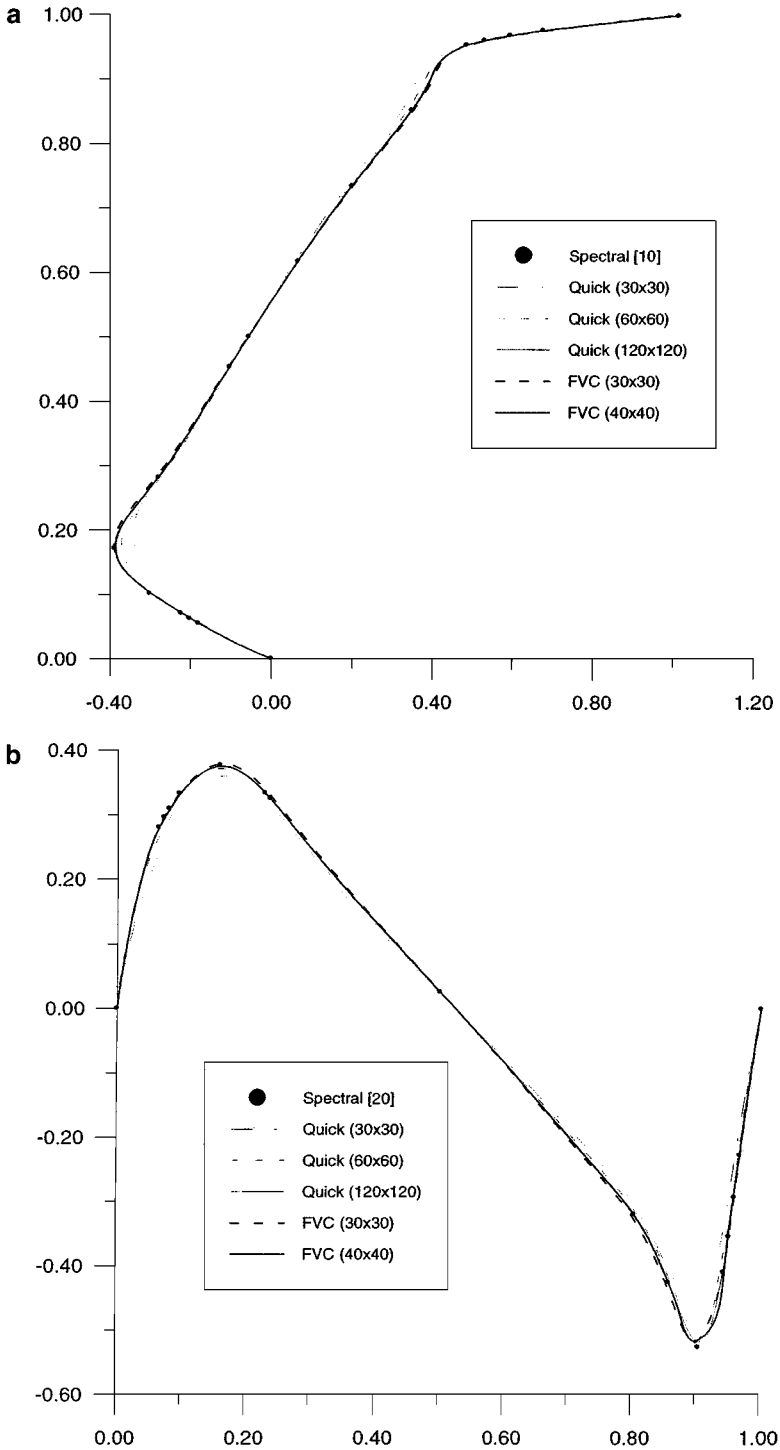
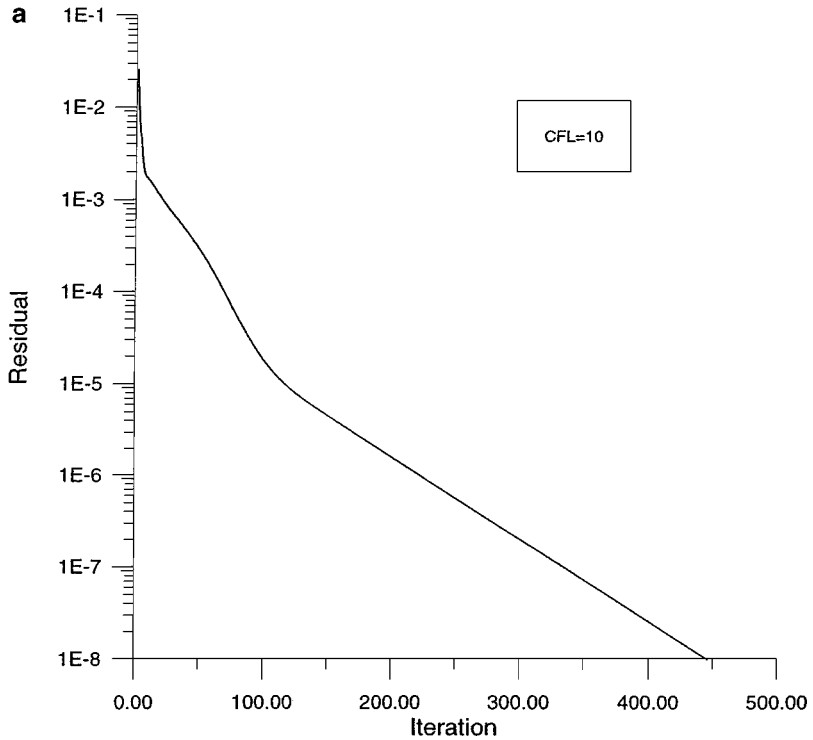


FIG. 5. (a) Vertical u-velocity profile for test case 3.4. (b) Horizontal v-velocity profile for test case 3.4.



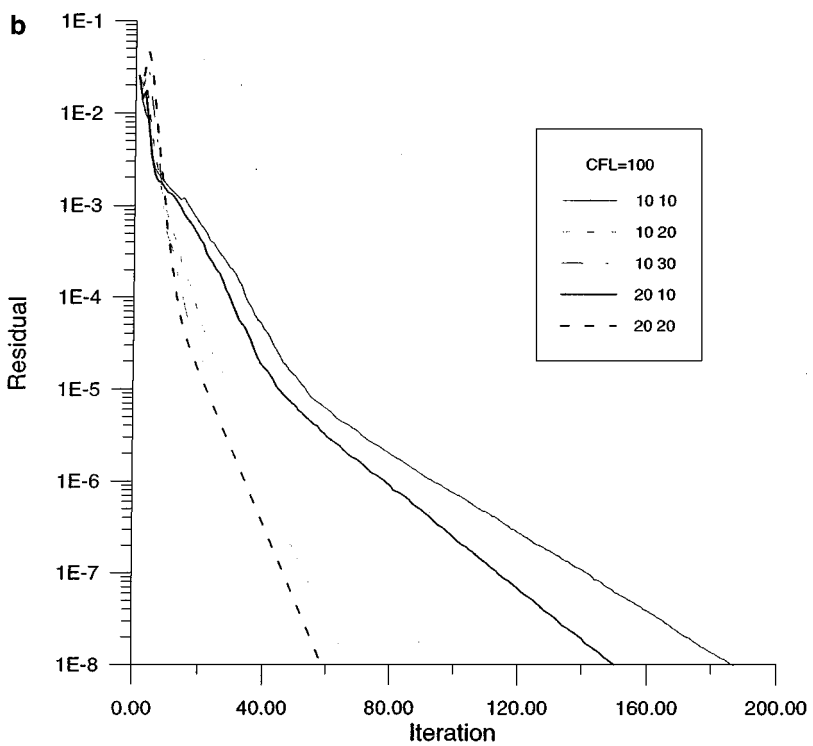


FIG. 6. (a) Convergence history for test case 3.4: CFL = 10. (b) Convergence history for test case 3.4: CFL = 100. (c) Convergence history for test case 3.4: CFL = 1000.

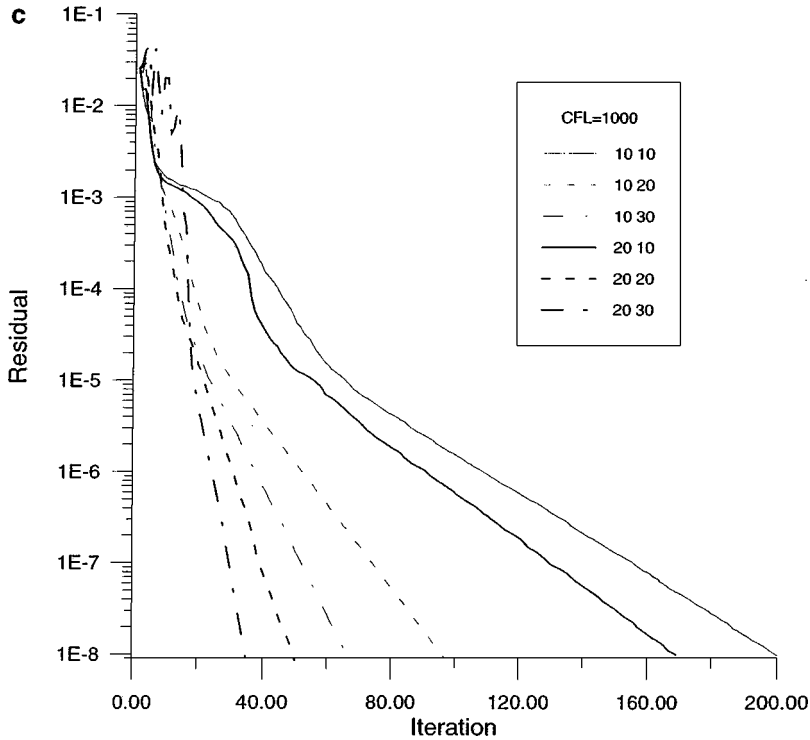


FIG. 6.—*Continued*

convergence can be obtained if the (inner) linear system of equations remains well resolved. Other factors, such as the level of approximation in the Gateaux derivative, appear to have only a minor effect in the convergence process.

4. CONCLUSIONS

A fourth-order compact finite volume method has been developed for the incompressible Navier–Stokes equations. The stencil for convective and diffusive flux approximations consists of a one-direction 3-point stencil. The Navier–Stokes equations poses new problems in the framework of high-order compact finite volume schemes, and they are mainly related with the average value of the nonlinear convective fluxes at the control volume faces and its coupling with the continuity equation. An original procedure was developed that ensures the implicit treatment of the average of the convective fluxes. The coupled solution of the discretized continuity and momentum equations is performed with an implicit Newton–Krylov matrix-free method. Because of the coupled solution strategy, no special Pressure–velocity method was required. In addition, the existence of a solution for the intermediate Newton iteration in the coupled solution of the equations is proved.

The unsteady form of the governing equations was treated with the standard fourth-order Runge–Kutta method in which, at each time step, the implicit solution of the corresponding coupled system of equations for the divergence constraint is obtained with a matrix-free Krylov method.

The performance of the numerical method was assessed in four problems. They show that the present method is fourth-order accurate in space and time, having promising capabilities.

ACKNOWLEDGMENTS

The authors are grateful to Eng. Nelson Marques for his help and suggestions related to matrix-free techniques.

REFERENCES

1. S. K. Lele, Compact finite difference schemes with spectral-like resolution, *J. Comput. Phys.* **103**, 16 (1992).
2. K. Mahesh, A family of high order finite difference schemes with good spectral resolution, *J. Comput. Phys.* **145**, 332 (1998).
3. A. I. Tolstykh and M. V. Lipavskii, On performance of methods with third- and fifth-order compact upwind differencing, *J. Comput. Phys.* **140**, 205 (1998).
4. N. A. Adams and K. Shariff, A high-resolution hybrid compact-ENO scheme for shock-turbulence interaction problems, *J. Comput. Phys.* **127**, 27 (1996).
5. H. C. Yee, Explicit and implicit multidimensional compact high-resolution shock-capturing methods: Formulation, *J. Comput. Phys.* **131**, 216 (1997).
6. M. M. Gupta, High accuracy solutions of incompressible Navier–Stokes equations, *J. Comput. Phys.* **93**, 343 (1991).
7. M. Li, T. Tang, and B. Fornberg, A compact fourth-order finite difference scheme for the steady incompressible Navier–Stokes equations, *Int. J. Numer. Methods Fluids* **20**, 1137 (1995).
8. W. F. Spotz and G. F. Carey, High-order compact scheme for the steady stream-function vorticity equations, *Int. J. Numer. Methods Eng.* **38**, 3497 (1995).
9. R. V. Wilson and A. O. Demuren, Numerical simulation of turbulent jets With rectangular cross-section, *J. Fluids Eng.* **120**, 285, ICASE, NASA Langley Research Center, Hampton, VA (1998).
10. R. V. Wilson, A. O. Demuren, and M. Carpenter, *Higher-Order Compact Schemes for Numerical Simulation of Incompressible Flows*, ICASE Technical Report 98-13 (1998).
11. D. Gaitonde and J. S. Shang, Optimized compact-difference-based finite-volume schemes for linear wave phenomena, *J. Comput. Phys.* **138**, 617 (1997).
12. M. H. Kobayashi, On a class of Padé finite volume methods, *J. Comput. Phys.* **156**, 137 (1999).
13. N. P. Marques and J. C. Pereira, Comparison of Matrix-Free Acceleration Techniques in Compressible Navier–Stokes Calculations, in *Proceedings of 1999 International Conference on Preconditioning Techniques for Large Sparse Matrix Problems in Industrial Applications, Minneapolis, Minnesota, June 10–12*, 1999.
14. A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, Uniformly high order accuracy essentially non-oscillatory schemes III, *J. Comput. Phys.* **71**, 231 (1987).
15. M. H. Carpenter, D. Gottlieb, and S. Abarbanel, The stability of numerical boundary treatments for compact high-order finite-difference schemes, *J. Comput. Phys.* **108**, 272 (1993).
16. P. N. Brown and Y. Saad, Hybrid Krylov methods for nonlinear systems of equations, *SIAM J. Sci. Stat. Comput.* **11**, 450 (1990).
17. V. Girault and P. Raviart, *Finite Element Methods for Navier–Stokes Equations* (Springer-Verlag, Berlin, 1986).
18. T. M. Shih, C. H. Tan, and B. C. Hwang, Effects of grid staggering on numerical schemes, *Int. J. Numer. Methods Fluids* **9**, 193 (1989).
19. U. Ghia, K. N. Ghia, and C. T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, *J. Comput. Phys.* **48**, 387 (1982).
20. O. Botella and R. Peyret, Benchmark spectral results on the lid-driven cavity flow, *Comput. Fluids* **27**, 421 (1998).
21. B. P. Leonard, A stable and accurate convective modeling procedure based on quadratic upstream interpolation, *Comput. Meth. Appl. Mech. Eng.* **19**, 59 (1979).