Proceedings of the World Congress on Engineering 2013 Vol I,
WCE 2013, July 3 - 5, 2013, London, U.K.

A High Order Accurate MultiGrid Pressure Correction Algorithm for Incompressible Navier-Stokes Equations

V. Mandikas, E. Mathioudakis, E. Papadopoulou and N. Kampanis

***Abstract*—A fourth-order accurate finite-difference compact numerical scheme coupled with a geometric MultiGrid technique is introduced for an efficient incompressible Navier-Stokes solver on staggered meshes. Incompressibility condition is enforced iteratively by solving a Poisson-type equation performing global pressure correction. Its application is the most computationally demanding part of the algorithm and in order to save computation cost a MultiGrid technique is applied to accelerate the iterative procedure of pressure updates at each time step. Appropriate boundary closure formulas are developed in case of the cell-centered approximations which implement the boundary conditions. Geometric MultiGrid techniques are not straightforward in this case, because the coarse grids do not constitute part of the fine grids. Suitable classical restriction and extension operators are modified for the efficient application of MultiGrid procedure. The performance investigation of realistic applications on non-high performance computing environments resulted that the MultiGrid technique can accelerate significantly the numerical solution process, holding at the same time the high accuracy of the numerical method even for high Reynolds numbers.**

***Index Terms*—Incompressible Navier-Stokes equations, Compact Finite Differences schemes, staggered grids, Geometric MultiGrid techniques.**

I. Introduction

In many practical applications, high resolution and fast solvers for incompressible Navier-Stokes equations are needed, such as in fine structured engineering on flows with high Reynolds number, where detailed flowfield information is required. The incompressibility constraint can be applied through various procedures, [1], [2]. In [3] it was proposed an iteratively procedure at each time step, until velocities computed satisfy the continuity equation to machine precision. A traditional approach is that of cell-by-cell pressure corrections applied iteratively until velocity practically satisfies the continuity equation, [4]. Numerical experiments showed that by increasing the Reynolds number, the number of iterations required by the local pressure correction method per time increase fast, making the associated computational cost intolerable. An alternative approach is to solve the Poisson equation satisfied by the pressure corrections which are obtained from the momentum equations, and obtain the pressure field distribution in terms of the velocities. Pressure correction computation using the Poisson equation is employed in several effective methods proposed for the numerical solution of the incompressible Navier-Stokes equations, [5], [6], [7]. An important observation for the pressure correction computation, using the Poisson equation is that, for high Reynolds number, the number of iterations required for incompressibility remains lower, [8], than that for the cell-by-cell pressure correction procedure. The application of the Poisson-type pressure correction procedure produces a large and sparse linear system of algebraic equations, suggesting the use of iterative solvers to save computational cost. Execution time for the solution of the linear system remains a forbidding factor for realistic applications on non-high performance computing environments. But significant convergence acceleration can achieved with the incorporation of geometric MultiGrid techniques into the iterative solver [9], [10], [11], [12], [13], [14].

We develop an algorithm based on a high order accurate finite difference method, more specifically on the fourth-order compact numerical scheme, which subsequently is applied to a cell-centered grid and the resulting algorithm is implemented for the pressure correction procedure in two dimensions as an extension of the numerical method introduced in [8]. In the proposed scheme a major innovation amounts to the treatment of realistic boundary conditions, since we focus to actual applications in fluid mechanics. Boundary closure formulas, for Dirichlet boundary conditions valid on the physical boundary, are derived and presented here in. In order to accelerate the pressure correction procedure a MultiGrid technique is applied and appropriate intergrid transfer operators with special treatments for boundary closures are constructed for this purpose. This paper is organized as follows: In Section 2, the overall numerical method based on a fourth order cell-centered compact difference scheme, for the pressure correction based on the Poisson-type equation is described. In Section 3, the MultiGrid technique is considered for accelerating the update pressure-correction procedure. Finally, Sections 4 and 5, present the numerical results and the performance investigation of the algorithm implementation and the conclusions respectively.

II. The incompressible Navier-Stokes solver

In this section a presentation of the proposed numerical method is given, with emphasis in description of the numerical treatment of realistic type boundary conditions for the application of the incompressibility condition.

Manuscript received February 20, 2013; revised April 16, 2013.

Corresponding author's e-mail: manolis@science.tuc.gr.

This research was supported in part by the special grant GZF030/2009-10 for Scholar's Association of Alexander S. Onassis Public Benefit Foundation.

V. Mandikas, E. Mathioudakis and E. Papadopoulou are with the Department of Sciences, Division of Mathematics, Technical University of Crete, University Campus, 73132 Chania, Crete, Greece email: bmandikas@science.tuc.gr, manolis@science.tuc.gr and elena@science.tuc.gr

N. Kampanis is with the Institute of Applied and Computational Mathematics, Foundation for Research and Technology - Hellas, 70013 Heraklion, Greece email:kampanis@iacm.forth.gr

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The incompressible Navier-Stokes equations in Cartesian (x, y) coordinates can be expressed as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = -\nabla p + \frac{1}{Re}\Big(\frac{\partial \mathbf{F_v}}{\partial x} + \frac{\partial \mathbf{G_v}}{\partial y}\Big), \tag{2}$$

where $\mathbf{u} = [u, v,]^T$ and p are the velocity vector and pressure, respectively, and $Re = UL/\nu$ is the Reynolds number based on a characteristic velocity U and a characteristic length L. Furthermore, $\mathbf{F}$ and $\mathbf{G}$ are the inviscid flux vectors, while $\mathbf{F_v}$ and $\mathbf{G_v}$ are the viscous fluxes given by

$$\mathbf{F} = [u^2, uv]^T,\ \mathbf{G} = [vu, v^2]^T,$$

$$\mathbf{F_v} = \Big[\frac{\partial u}{\partial x}, \frac{\partial v}{\partial x}\Big]^T, \quad \mathbf{G_v} = \Big[\frac{\partial u}{\partial y}, \frac{\partial v}{\partial y}\Big]^T.$$

Using appropriate finite compact centered schemes on staggered grids, a fourth-order accuracy in space was obtained in [8]. The overall fourth order accurate numerical method, is not able to model properly incompressible flows at high Reynolds numbers, due to the increase of the computational cost for incompressibility condition. This limitation can be overcame by using MultiGrid techniques in order to accelerate the numerical solution of the Poisson-type equation discretized by high order finite difference methods. Specifically, a fourth order compact scheme is used to discretize the Poisson-type equation. The use of compact descritizations additionally serves the treatment of boundary conditions with one layer of fictitious cells in outward along the boundary. Unlike compact methods, classical finite differences require wider fictitious layer making the use in curvilinear domains inappropriate. The explicit Runge-Kutta [15], time-marching scheme fourth-order accuracy is employed for the time semidiscretization. This amounts to the time stepping through the following intermediate steps:

$$\mathbf{U}^{n,1} = \mathbf{U}^n, \qquad P^{n,1} = P^n \tag{3}$$

$$\mathbf{U}^{n,2} = \mathbf{U}^n + \frac{\Delta t}{2}\mathbf{R}^{n,1}, \tag{4}$$

$$\mathbf{U}^{n,3} = \mathbf{U}^n + \frac{\Delta t}{2}\mathbf{R}^{n,2}, \tag{5}$$

$$\mathbf{U}^{n,4} = \mathbf{U}^n + \Delta t\mathbf{R}^{n,3}, \tag{6}$$

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \frac{\Delta t}{6}(\mathbf{R}^{n,1} + 2\mathbf{R}^{n,2} + 2\mathbf{R}^{n,3} + \mathbf{R}^{n,4}) \tag{7}$$

with $t^n = n\Delta t$, $t^{n,1} = t^n$, $t^{n,2} = t^{n,3} = t^n + \Delta t/2$, $t^{n,4} = t^n + \Delta t$, and $\mathbf{R}^{n,\ell} = \mathbf{R}(\mathbf{U}^{n,\ell}, P^{n,\ell}; t^{n,\ell})$, for $\ell = 2, 3, 4$, where **R**(**U**,P;t) is the right-hand side of (2).

The quantities $P^{n,\ell}$, $\ell = 2, 3, 4$ and P^{n+1}, appearing in (3)-(7), are determined by enforcing the incompressibility on the velocity vectors $\mathbf{U}^{n,\ell}$, $\ell = 2, 3, 4$ and $\mathbf{U}^{n+1}$. Incompressibility is enforced on the velocity vectors $\mathbf{U}^{n,\ell}$, $\ell = 2, 3, 4$, and $\mathbf{U}^{n+1}$ by the pressure update procedures described below.

The velocities vectors $\mathbf{U}^{n+1}$(as well as the intermediate stages $\mathbf{U}^{n,\ell}$, $\ell = 2, 3, 4$) do not (necessarily) satisfy the incompressibility condition (1). The computed velocity field at each time step (and the intermediate stages) should be corrected to satisfy (1) usually iteratively through suitable pressure updates. Pressure updates at each time step may be corrected for all cells simultaneously obtained by computing pressures corrections for all cells locally computed in a cell by cell advancing procedure. This however results in a low order, lowly convergence procedure with poor effectiveness especially at higher Reynolds numbers where flow patterns are significantly complex. As a consequence, the Poisson-type equation must be solved several times making the application of a MultiGrid technique necessary. The spatial derivatives appearing in the continuity equation during the pressure update procedures, are computed using fourth-order accurate compact formulas.

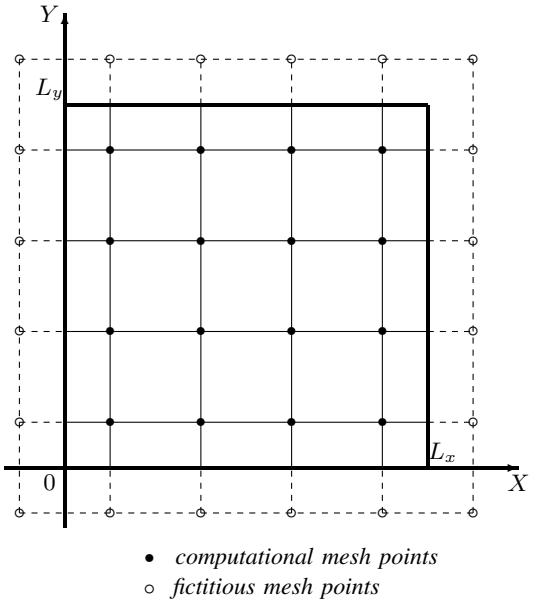

Fig. 1. The pressure correction computational grid.

As an alternative, [8], the following Poisson-type equation for the pressure correction Δp defined globally on $\Omega \equiv [0, L_x] \times [0, L_y]$, and valid at the cell centers M_{ij}, $i = 1, ..., N_x$, $j = 1, ...N_y$:

$$\left(\frac{\partial^2(\Delta p)}{\partial x^2}\right)_{ij} + \left(\frac{\partial^2(\Delta p)}{\partial y^2}\right)_{ij} = f_{ij} \tag{8}$$

with $f_{ij} = f(M_{ij}) = \frac{1}{a_{\ell,\ell-1}\Delta t}(\nabla \cdot \mathbf{u}^{n,\ell}_{old})$.

The boundary conditions for the pressure correction Δp and pressure are associated. For Dirichlet boundary conditions, p is constant, therefore $\Delta p = 0$. For Neumann boundary conditions, $\partial p/\partial n = 0$, that implies $\partial(\Delta p)/\partial n = 0$, with n the outward normal on the boundary. The solution $\Delta p(x, y)$ and the right-hand side function $f(x, y)$ are assumed to be sufficiently smooth and have the required continuous partial derivatives. The fourth order finite difference compact numerical method is seeking for an approximation of pressure corrections solution at the center node of each computational cell of Ω, since our computational grid is a staggered one. There are $n = N_x N_y$ centered mesh points $(x_{i-1/2}, y_{i-1/2})$, $i = 0, 1, \ldots, N_x + 1$, $j = 0, 1, \ldots, N_y + 1$ inside Ω, and they can be assumed as grid points of a new shifted computational grid (x_i, y_i), $i = 0, 1, \ldots, N_x + 1$, $j = 0, 1, \ldots, N_y + 1$, where mesh points with $i = 0, N_x + 1$

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

and $j = 0, N_y + 1$ are the fictitious mesh points, Fig. 2. The fourth order compact approximation scheme, which is mathematically equivalent to [16], can be expressed as

$$\begin{aligned} &d(\Delta p_{i+1,j+1} + \Delta p_{i+1,j-1} + \Delta p_{i-1,j+1} + \Delta p_{i-1,j-1}) \\ &+ c(\Delta p_{i+1,j} + \Delta p_{i-1,j}) + b(\Delta p_{i,j+1} + \Delta p_{i,j-1}) \\ &- a\Delta p_{i,j} \\ &= \frac{\Delta x^2}{2}(8f_{i,j} + f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}), \end{aligned} \tag{9}$$

with $a = 10(1+\gamma^2)$, $\quad b = 5 - \gamma^2$, $\quad c = 5\gamma^2 - 1$, $d = (1+\gamma^2)/2$ and $\gamma = \Delta x/\Delta y$ the mesh ratio.

Since the new shifted computational grid coincides with the center nodes of the staggered grid, the incorporation of the actual boundary conditions is not straightforward. To this end, the following treatment procedure of boundary conditions is introduced by the authors. Following [17], the interpolation formula

$$\phi_{i+\frac{1}{2}} + \alpha\phi_{i+\frac{3}{2}} = a\phi_i + b\phi_{i+1} + c\phi_{i+2} + d\phi_{i+3}, \tag{10}$$

with $\alpha = free$, $a = \frac{5}{16} - \frac{a}{16}$, $b = \frac{15}{16} + \frac{9a}{16}$, $c = -\frac{5}{16} + \frac{9a}{16}$ and $d = \frac{1}{16} - \frac{a}{16}$, is used. This procedure relates values on the original structure grid $(i+1, i+2, i+3)$ with values on the new shifted grid $(i+1/2, i+3/2)$, and is used for the implementation of Dirichlet boundary conditions.
Accordingly, [17],

$$\phi'_{i+\frac{1}{2}} + \alpha\phi'_{i+\frac{3}{2}} = a\phi_i + b\phi_{i+1} + c\phi_{i+2} + d\phi_{i+3} + e\phi_{i+4}, \tag{11}$$

with $\alpha = free$, $a = -\frac{22}{24} + \frac{a}{24}$, $b = \frac{17}{24} - \frac{9a}{8}$, $c = \frac{3}{8} + \frac{9a}{8}$, $d = -\frac{5}{24} - \frac{a}{24}$ and $e = \frac{1}{24}$, is used for the case of Neumann boundary conditions. The above formulas (10) and (11) simplify to

$$\phi_i = \frac{16}{5}\phi_{i+\frac{1}{2}} - 3\phi_{i+1} + \phi_{i+2} - \frac{1}{5}\phi_{i+3} \tag{12}$$

$$\phi_i = -\frac{12}{11}\phi'_{i+\frac{1}{2}} + \frac{17}{22}\phi_{i+1} + \frac{9}{22}\phi_{i+2} - \frac{5}{22}\phi_{i+3} + \frac{1}{22}\phi_{i+4} \tag{13}$$

for $\alpha = 0$.

The zebra coloring scheme for numbering unknowns and equations is adopted for the linear system in (9) , which guarantees a desirable potential high degree of parallelism and scalability [18], [19], [20]. This type of coloring scheme produces the coefficient matrix $A \in \mathbb{R}^{n\times n}$ of the fourth order cell-centered compact difference scheme in the following block form

$$A = \begin{bmatrix} A_1 & A_2 & O & \dots & O & O & O & A_3 & A_4 & O & \dots & O & O & O \\ O & A_5 & O & \dots & O & O & O & A_6 & A_6 & O & \dots & O & O & O \\ O & O & A_5 & \dots & O & O & O & O & A_6 & A_6 & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & A_5 & O & O & O & O & O & \dots & A_6 & O & O \\ O & O & O & \dots & O & A_5 & O & O & O & O & \dots & A_6 & A_6 & O \\ O & O & O & \dots & O & O & A_5 & O & O & O & \dots & O & A_6 & A_6 \\ A_6 & A_6 & O & \dots & O & O & O & A_5 & O & O & \dots & O & O & O \\ O & A_6 & A_6 & \dots & O & O & O & O & A_5 & O & \dots & O & O & O \\ O & O & A_6 & \dots & O & O & O & O & O & A_5 & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & A_6 & A_6 & O & O & O & O & \dots & A_5 & O & O \\ O & O & O & \dots & O & A_6 & A_6 & O & O & O & \dots & O & A_5 & O \\ O & O & O & \dots & O & A_4 & A_3 & O & O & O & \dots & O & A_2 & A_1 \end{bmatrix}, \tag{14}$$

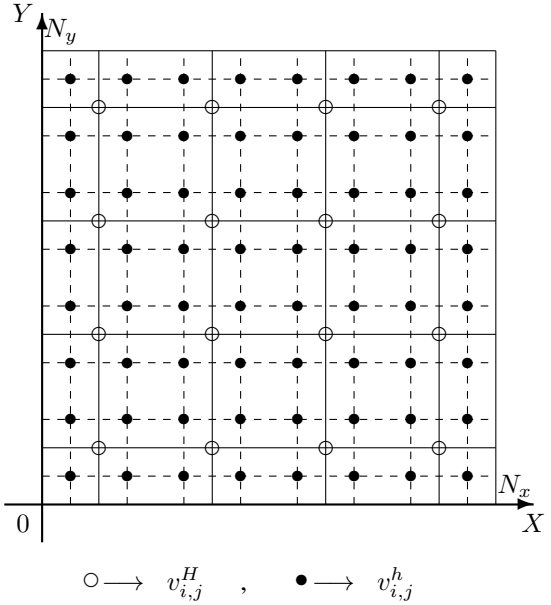

Fig. 2. Fine and Coarse grid discretization and residual nodes.

with the order of submatrices A_i for $i = 1, \dots, 4$ been N_x. Robin or mixed boundary conditions can be treated similarly, eliminating the fictitious nodes combining the approximation formulas (12) and (13).

III. MultiGrid solver for pressure correction

MultiGrid methods usually achieve high rates of convergence and are considered among the fastest methods for solving large sparse linear systems arising from the discretization of multi-dimensional boundary value problems. The multigrid method solves the error correction sub-problem (coarse grid correction) on a coarse grid and interpolates the error correction solution back to fine grid. All major computational operations are performed on coarse grids where the size of the problem is small in order to save computational time. A multigrid method consists of a smoother, which is a relaxation scheme for the error, and two grid-transfer operators, the restriction for mapping residual vectors from the fine to the coarse grid, and the prolongation (interpolation) for returning the corrected error vectors back to the fine grid [9], [11], [12], [13], [14]. The classical intergrid operations cannot be applied to the pressure correction grid without suitable modifications, because in every grid level the mesh does not constitute part of one in the previous level after the grid transferring procedure. For this reason, the corresponding residuals for these nodes are unknown, and all node values in the case of the cell-centered grid have to be approximated (Fig. 2). Performance investigation of several grid operators resulted that a bilinear interpolation operator and the corresponding restriction operator for cell-centered discretizations are an efficient choice for the present work.
Suppose that v^H is the error vector corresponding to step-size $H = 2h$ of the coarse grid Ω_H, the respective extended vector v^h corresponding to step-size h for the fine grid Ω_h is defined by the following relations

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$\begin{aligned} v^h_{2i,2j} &= \tfrac{1}{16}(9v^H_{i,j} + 3v^H_{i+1,j} + 3v^H_{i,j+1} + v^H_{i+1,j+1}) \\ v^h_{2i+1,2j} &= \tfrac{1}{16}(3v^H_{i,j} + 9v^H_{i+1,j} + v^H_{i,j+1} + 3v^H_{i+1,j+1}) \\ v^h_{2i,2j+1} &= \tfrac{1}{16}(3v^H_{i,j} + v^H_{i+1,j} + 9v^H_{i,j+1} + 3v^H_{i+1,j+1}) \\ v^h_{2i+1,2j+1} &= \tfrac{1}{16}(v^H_{i,j} + 3v^H_{i+1,j} + 3v^H_{i,j+1} + 9v^H_{i+1,j+1}) \end{aligned} \quad (15)$$

for $i = 1, ..., \frac{N_x}{2} - 1$ and $j = 1, ..., \frac{N_y}{2} - 1$ and is called bilinear interpolation (I^H_h) operator. For all components of v^h corresponding to points close to the boundary, fictitious coarse points values are involved lying outside the boundary area. These values can be eliminated using the boundary conditions. For the case of Dirichlet with zero valued function and Neumann boundary condition cases the components of fine vector close to the $y = 0$ boundary are giver by

$$v^h_{2i+1,1} = \tfrac{1}{8}(v^H_{i,1} + 3v^H_{i+1,1}) \, , v^h_{2i,1} = \tfrac{1}{8}(3v^H_{i,1} + v^H_{i+1,1}) \quad (16)$$

and

$$v^h_{2i+1,1} = \tfrac{1}{4}(v^H_{i,1} + 3v^H_{i+1,1}) \, , v^h_{2i,1} = \tfrac{1}{4}(3v^H_{i,1} + v^H_{i+1,1}) \quad (17)$$

respectively, for $1 \leq i \leq \frac{N_x}{2} - 1$. Other points close to the boundary are similarly treated.

Choosing as a restriction operator I^H_h the one that satisfies the relation

$$I^h_H = \frac{1}{4} I^H_h \quad ,$$

then in stencil notation it is defined as

$$I^h_H = \frac{1}{64} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 3 & 9 & 9 & 3 \\ 3 & 9 & 9 & 3 \\ 1 & 3 & 3 & 1 \end{bmatrix} . \quad (18)$$

In the MultiGrid algorithm the V-cycle scheme is chosen which can decrypted with the following compact recursive algorithm

V-cycle algorithm

$\Delta p^{m+1}_k = mgvcycle(k, \Delta p^m_k, A_k, f_k, v_1, v_2)$

Presmoothing:

$\Delta p^m_k = smooth^{v_1}(\Delta p^m_k, A_k, f_k)$

Restriction:

$f^m_{k-1} = I^k_{k-1}(f_k - A_k \Delta p_k m)$

Recursion:

if $k = 1$ use a fast iterative solver for $A_{k-1}\Delta p^m_{k-1} = f^m_{k-1}$

if $k > 1$ $\Delta p^m_{k-1} = mgvcycle(k-1, 0, A_{k-1}, f^m_{k-1}, v_1, v_2)$

Interpolation:

$\Delta p^m_k = \Delta p^m_k + I^{k-1}_k \Delta p^m_{k-1}$

Postsmoothing:

$\Delta p^{m+1}_k = smooth^{v_2}(\Delta p^m_k, A_k, f_k)$

The above algorithm calculates the new approximation Δp^{m+1}_k at iteration step $m+1$ of the linear system solution Δp_k from the previous one Δp^m_k. The subscript k indicates the grid level with $k = 1$ corresponding to finest grid. In the description of the V-cycle algorithm, performing v smoothing steps with an iterative solver e.g Gauss-Seidel [21], [11], [14], [12], applied to any discrete problem of the form $A_k \Delta p_k = f_k$ with initial approximation Δp_k is denoted by the $smooth^v$ procedure. Usually the number of pre- and postsmoothing iterations in the descent and ascent phase of V-cycle are both equal to 2. For the multigrid levels of V-cycle, the matrices A_k at the coarse levels are determined by discretizing the Poisson-type equation on the corresponding coarse grid. The smoother Gauss-Seidel iterative solver is based on the following slitting of the coefficient matrix $A = D_A - L_A - U_A$ where

$$D_A := \begin{bmatrix} A_1 & A_2 & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & A_5 & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & A_5 & \dots & O & O & O & O & O & O & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & A_5 & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & A_5 & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & A_5 & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & A_5 & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & A_5 & O & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & O & O & O & O & O & O & \dots & A_5 & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & A_5 & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & A_2 & A_1 \end{bmatrix} ,$$

$$-L_A := \begin{bmatrix} O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ A_6 & A_6 & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & A_6 & A_6 & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & A_6 & \dots & O & O & O & O & O & O & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & A_6 & A_6 & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & A_6 & A_6 & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & A_4 & A_3 & O & O & O & \dots & O & O & O \end{bmatrix}$$

and

$$-U_A := \begin{bmatrix} O & O & O & \dots & O & O & O & A_3 & A_4 & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & A_6 & A_6 & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & A_6 & A_6 & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & O & O & O & O & O & O & \dots & A_6 & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & A_6 & A_6 & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & A_6 & A_6 \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \\ O & O & O & \dots & O & O & O & O & O & O & \dots & O & O & O \end{bmatrix} .$$

IV. Numerical Experiments

This section presents numerical results obtained from the realization of the fourth order accurate numerical algorithm. The convergence and accuracy of the method is investigated for steady and unsteady flow problems. The total number of Poisson-type equation required to be solved, as well average cpu cost per Poisson-type equation are recorded. All numerical tests were performed on a SunFire X2200M2 machine with 4GB memory and two dual core Opteron@3.0GHz processors with one core been enabled. The implementation was developed in double precision Fortran code and all basic linear algebra operations were performed using the Lapack [20] scientific library.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

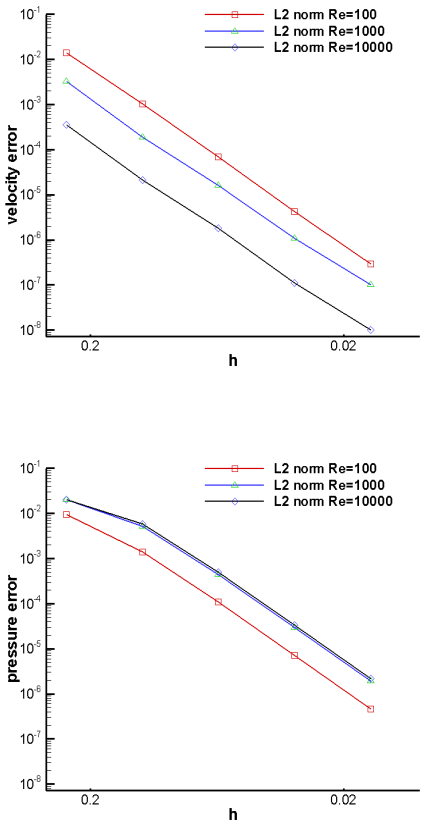

Fig. 3. Spatial accuracy against the analytical solution for the Taylor Vortex problem with Re=100, 1000 and 10000.

Problem 1: **Taylor vortex.**

In order to confirm the spatial accuracy of the algorithm implementation for every choice of Reynolds number, we solved the Taylor vortex problem [22] on the domain $[-1,1] \times [-1,1]$ with Dirichlet boundary conditions. The initial condition is based on the exact solution:

$$\mathbf{u} = (-\cos(\pi x)\sin(\pi y)\mathbf{i} + \sin(\pi x)\cos(\pi y)\mathbf{j}) \exp\left(\frac{-2\pi^2 t}{Re}\right)$$
$$p = -\frac{1}{4}(cos(2\pi x) + \cos(2\pi y)) \exp\left(\frac{-2\pi^2 t}{Re}\right) \quad (19)$$

where Re=1/v been the Reynolds number and considered as $Re \in \{10, 10^2, 10^3\}$. The time step size $\Delta t = 10^{-5}$ was chosen to satisfy the CFL condition and to ensure that the dominant error was the spatial error. The results are summarized in Fig. 3. The convergence rate of L^2 norm for velocity and pressure error evaluated at the final time measurement T=5, is 4 for all choices of Re.

Problem 2: **Driven cavity flow.**

We also considered the steady-state driven cavity flow problem with well-defined boundary conditions on the unit square. Numerical solutions were obtained at different Reynolds numbers and compared with the computations of Ref. [23]. Fig. 4 presents the agreement with the solution approximation of Ghia et al for $Re = 1000$ and $Re = 3200$. On a 256×256 grid or finer for $Re = 1000$ the solution can be approximated sufficiently. Nonetheless, increasing the Reynolds numbers to 3200 a finer grid must be applied as shown on the two bottom plots. Tables 1 and 2 present the total number of Poisson-type equations required to be solved in all stages of Runge-Kutta time marching procedure, until time reaches the end in order to enforce incompressibility condition iteratively. Additionally, the average cpu time in seconds for approximating the solution of every Poisson-type equation is shown. These time measurements are also presented for the corresponding method proposed in [8] . As shown, our New method's algorithm has reduced significantly the number of Poisson-type equations required to be solved in all cases. An important outcome of this performance investigation is that every Poisson-equation is solved faster as the Reynolds number increases for the new algorithm. This is due to the application of the fourth order compact scheme with the multigrid technique, because high Reynolds numbers produce more oscillatory solutions corresponding to more oscillatory errors. It is well-known that a multigrid technique quickly eliminates the oscillatory error components.

Table 1
Total number of Poisson-type equations for $Re = 1000$ and T=35.

grid size	Method in [8]		New Method	
	No. Poisson	Time/Poisson	No. Poisson	Time/Poisson
64	224710	0.015	121705	0.006
128	613226	0.147	131124	0.025
256	1756002	1.513	219869	0.105

Table 2
Total number of Poisson-type equations for $Re = 3200$ and T=75.

grid size	Method in [8]		New Method	
	No. Poisson	Time/Poisson	No. Poisson	Time/Poisson
64	300150	0.015	155817	0.006
128	709023	0.149	265375	0.015
256	1600310	1.515	430163	0.062
512	-	-	996746	0.318

V. Conclusion

An important performance improvement has been achieved for the high accurate incompressible Navier-Stokes solver proposed in [8]. The application of a fourth order finite difference compact discretization method coupled with a multigrid technique on the spatial staggered grid made readily available the pressure values approximations on the required grid nodes by the incompressibility condition. This leads avoiding a high order two dimensional interpolation for every time step. Moreover, a significantly reduction of the number of Poisson-type equations has been observed, which has accordingly reduced the execution time of the algorithm. At the same time the high accuracy of the solver is sustained. The proposed method is currently being extended to the more complicated, general elliptic type equation satisfied by pressure correction at each time step, for the Navier-Stokes equations expressed in curvilinear coordinates. The method employed can also be extended in the three dimensional case in a straightforward manner on parallel architectures.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

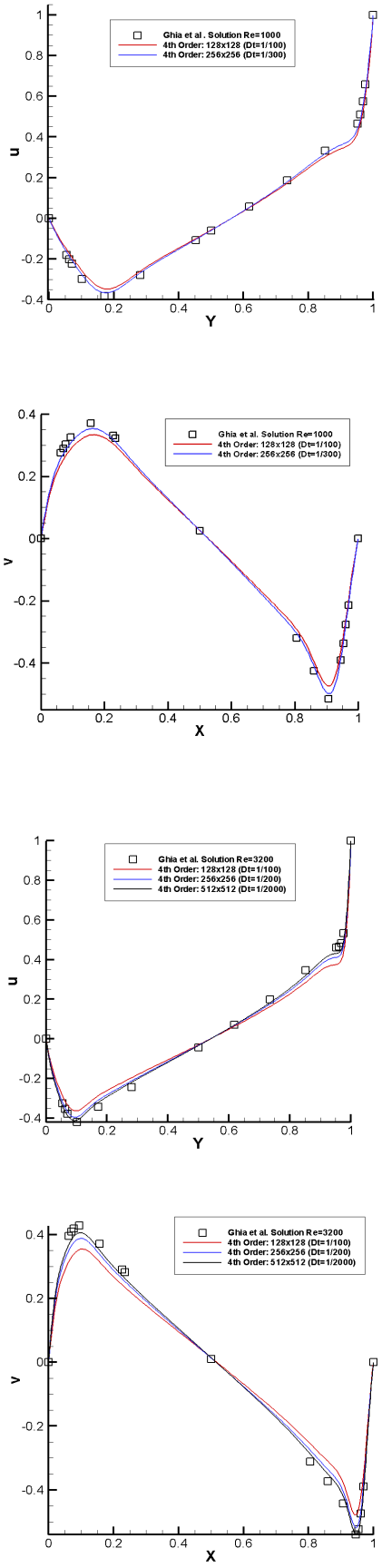

Fig. 4. Comparison of the horizontal and vertical velocity components computed with the reference solution by Ghia et al.

REFERENCES

[1] A. Chorin, "A numerical method for solving incompressible, viscous flow problems," *J. Comp. Physics*, vol. 2(1), pp. 12–26, 1967.
[2] C. Merkle and M. Athavale, "Time-accurate unsteady incompressible flow algorithms based on artificial compressibility," *AIAA Paper*, pp. 87–1137, 1987.
[3] J. D. Anderson, *Computational Fluid Dynamics*. New York: McGrawHill, 1995.
[4] C. Hirt, B. Nichols, and N. Romero, "SOLA: a numerical solution algorithm for transient fluid flows," in *Los Alamos Report LA-5852*, 1975.
[5] A. Chorin, "Numerical solution of the Navier-Stokes equations," *Math. Comput.*, vol. 22, pp. 745–762, 1968.
[6] R. Issa, "Solution of the implicitly discretised fluid flow equations by operator-splitting," *J. Comp. Physics*, vol. 65, pp. 40–65, 1985.
[7] S. Patankar, *Numerical Heat Transfer and Fluid Flow*. Washington, DC: Hemisphere Publishing Co., 1980.
[8] N. Kampanis and J. Ekaterinaris, "A staggered grid, high-order accurate method for the incompressible Navier-Stokes equations," *J. Comp. Physics*, vol. 215, pp. 589–613, 2006.
[9] A. Brandt, "Multi-level adaptive solutions to boundary value problems," *Mathematics of Computation*, vol. 31, pp. 333–390, 1997.
[10] A.Brandt, "A guide to multigrid development in multigrid methods," W.Hackbusch and U. Trottenberg, Eds. Berlin: Springer Verlag, 1982, pp. 220–312.
[11] W. L. Briggs, V. E. Henson, and S. McCormick, *A Multigrid Tutorial*. Philadelphia: SIAM, 2000.
[12] U. Trottenberg, C. Oosterlee, and A. Schüller, *Multigrid*. New York: Elsevier Academic Press, 2001.
[13] P. Wesseling, *An Introduction to Multigrid Methods*. Chichester, U.K.: J. Wiley and Sons, 1992.
[14] S. McCormick, *Multigrid Methods*. Philadelphia: SIAM, 1987.
[15] J. Butcher, *The Numerical Analysis of Ordinary Differential Equations: Runge-Kutta Methods and General Linear Methods*. Chichester: J. Wiley and Sons, 1987.
[16] J. Zhang, "Multigrid method and fourth-order compact scheme for 2D Poisson equation with unequal mesh-size discretization," *J. Comp. Physics*, vol. 179, pp. 170–179, 2002.
[17] D. Gaitonde and M. Visbal, "High order schemes for Navier-Stokes equations: Algorithm and implementation into FDL3DI," in *NASA, AFRL-VA-WP-TR-1998-3060*, 1998.
[18] E. Mathioudakis, E. Papadopoulou, and Y. Saridakis, "Iterative solution of elliptic collocation systems on a cognitive parallel computer," *Computers and Maths with Appl.*, vol. 48, pp. 951–970, 2004.
[19] Y. Saad, *Iterative methods for sparse linear systems*. Philadelphia: SIAM, 2003.
[20] J. Dongarra, I. Duff, D. Sorensen, and H. van der Vorst, *Numerical Linear Algebra for high-performance computers*. Philadelphia: SIAM, 1998.
[21] R. Varga, *Matrix Iterative Analysis*. New York: Springer Verlag, 2000.
[22] C. R. E. K. Shahbazi, P. F Fischer, "A high-order discontinuous galerkin method for the unsteady incompressible navierstokes equations." *J. Comput. Physics*, vol. 222, 2007.
[23] G. K. N. Ghia, U. and C. T. Shin, "High-re solutions for incompressible flow using the navier-stokes equations and a multigrid method," *J. Comput. Physics*, vol. 48, pp. 387–411, 1982.

ISBN: 978-988-19251-0-7
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)