

AN ADAPTIVE SECOND-ORDER PROJECTION METHOD FOR UNSTEADY FLOWS

Millena M. Villar

mmvillar@mecanica.ufu.br

Faculdade de Engenharia Mecânica - Universidade Federal de Uberlândia

Av. João Naves de Ávila 2121, Campus Santa Mônica, 38400-902, Uberlândia - MG - Brasil

Alexandre M. Roma

roma@ime.usp.br

Instituto de Matemática e Estatística - Universidade de São Paulo

Rua do Matão, 1010, Cidade Universitária, 05508-090, São Paulo - SP - Brasil

Aristeu da Silveira Neto

aristeus@mecanica.ufu.br

Faculdade de Engenharia Mecânica - Universidade Federal de Uberlândia - Brasil

Hector D. Cenicerós

hdc@math.ucsb.edu

Department of Mathematics - University of California - Santa Barbara - USA

Abstract. *A new methodology based on projection method which allow to simulate incompressible flows is presented in this work. The integration domain is discretized on a block-structured adaptive mesh. This approach consist on a sequence of nested grids with successively finer spacing with refinement ratio equal to 2. This new implementation of projection method is based on Gear method where the forcing term is extrapolated in two time-steps assuring second-order to the velocities ($O(dt^2, dx^2)$) but at least first-order to the pressure and CFL condition to the time-step constraint. The multigrid-multilevel method is applied to advection-diffusion equation and to the Poisson equation solution of pressure correction, both of which are spatially discretized with second-order centered finite differences method. The turbulent viscosity proposed by Smagorinsky which is a linear model is introduced in an explicit time way. The method is checked trough some examples where the fluid properties might be constant or variable and the integration meshes might be as well as not be locally refined.*

Keywords. *Adaptive mesh, Gear method, Projection method.*

1. INTRODUCTION

Detailed modeling of unsteady flows which gives realistic results poses severe requirements on the computational resources. The computational costs can be dramatically reduced by combining implicit methods, which allow for less stringent time-step selections, with locally refined meshes, which focus the computational work on the regions of the domain where it is most needed (e.g. near boundary layers and/or turbulent regions).

The objective of this paper is to present an efficient and robust numerical method for the unsteady Navier-Stokes equations, which employs a local adaptive mesh refinement (AMR) in the space discretization, and a semi-implicit time discretization which stably handles the non-linear viscous term explicitly, the method being restricted only by a linear CFL time-step constraint ($\Delta t = O(\Delta x)$).

The AMR algorithm employs the hierarchical block-structured grid approach developed by Berger and coworkers (Berger and Oliger, 1984; Berger and Colella, 1989 and Berger and Leveque, 1998), initially introduced for hyperbolic partial differential equations. The AMR is based on a collection of logically rectangular meshes that make up the coarse grid; refinements cover a subset of the domain and use smaller rectangular grid patches. These finer patches can be recursively nested until a given level of accuracy is attained. Accordingly to the problem, several criteria identifying the regions where refinement is to be applied may be used. Here, high turbulence and/or vorticity will be considered.

The application of the AMR has been extended to incompressible flows (Howel and Bell, 1997), and to incompressible multi-phase flows (Ceniceros e Roma, 2004) where a fluid indicator is obtained at optimal computational cost by applying ideas from *Computational Geometry*. Trangenstein (2002) used the same approach to the flow simulation in a porous media where the localized phenomena was treated with hybrid mixed finite element and adaptive mesh. In order to simulate the flow around deformable boundaries, Roma (1999) used the Immersed Boundary Method and adaptive mesh refinement to dynamically follow the immersed boundary motion.

In the present work, the flow equations are solved employing a projection method based on the work by Badalassi *et al.* (2003). The time discretization of the method is a semi-implicit one based on an extraction of constant coefficient leading order terms that are time-step split. This efficient semi-implicit time discretization based on Gear's Method is employed along with centered finite differences for the discretization in space. Special multilevel-multigrid methods are applied to solve the *linear systems* arising from the convection-diffusion equation and from the pressure-correction Poisson equation. Although the non-linearity, introduced in the problem by the Smagorinsky turbulent viscosity model (Smagorinsky, 1963), is treated explicitly (thus preventing any non-linear systems of equations of being solved), the overall time discretization is subjected only to a mild time-step constraint ($\Delta t = O(\Delta x)$).

The paper is organized as follows. Section 2 gives a brief introduction to the mathematical model, presenting the space and time discretizations adopted. Section 3 presents the block-structured, adaptive mesh refinement technique, and the multilevel-multigrid method. Results and conclusions are presented in Sections 4 and 5, respectively.

2. MATHEMATICAL AND NUMERICAL METHOD

In this section, the projection method for the unsteady, two-dimensional incompressible flow, which is second-order for the velocities, is presented. This scheme may be applied on both uniform and locally refined meshes. Both the time/space discretizations and the turbulent modeling are also briefly described.

Let the governing equations of motion be given by the Navier-Stokes equations

$$\rho(\mathbf{u}_t + \mathbf{u} \cdot \nabla \mathbf{u}) = -\nabla p + \nabla \cdot [\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)] + \mathbf{f}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

with appropriate boundary and initial conditions on the domain Ω , where $\mathbf{u}$ is the velocity field, p is the pressure and $\mathbf{f}$ are the external forces. The fluid material properties may be variable or constant, where ρ and μ is the density and viscosity, respectively.

2.1. Time Discretization

A semi-implicit time discretization combined with a time-split strategy is applied. The discretization, based on the numerical scheme employed by Badalassi *et al.* (2003), yields an efficient and robust modular scheme. To introduce the main conceptual ideas, for simplicity, consider the diffusion equation

$$\mathbf{u}_t = \nabla \cdot (\theta \nabla \mathbf{u}), \quad (3)$$

where $\theta > 0$ at any given instant of time. Eq. (3) can be rewritten as

$$\mathbf{u}_t = \gamma \nabla^2 \mathbf{u} + f(u), \quad (4)$$

where $f(u) = \nabla \cdot (\theta \nabla \mathbf{u}) - \gamma \nabla^2 \mathbf{u}$ and γ is constant. If the first term on the right-hand side of Eq. (4) is treated implicitly and $f(u)$ is treated explicitly, a semi-implicit discretization is obtained. With energy estimates, one can show that the first order Euler discretization is unconditionally stable if $\gamma \geq (1/2) \max\{\theta(t, x)\}$, $0 \leq t \leq T$, $x \in \Omega$ (Douglas and Dupont, 1971).

A second-order, extrapolated multi-step Gear's Method (SBDF) is chosen for the discretization in time. Among the second-order multi-step methods, the second-order SBDF has the strongest high-modal decay. This provides the required damping for the very high frequencies in the diffusion equation without a harsh time-step constraint. Employing Gear's Method in conjunction with the ideas presented above for the diffusion term, one obtains for Eq. (1)

$$\frac{\rho^{n+1}}{\Delta t} [\alpha_2 \mathbf{u}^{n+1} - \alpha_1 \mathbf{u}^n + \alpha_0 \mathbf{u}^{n-1}] = \gamma \nabla^2 \mathbf{u}^{n+1} + \beta_1 \mathbf{g}^n + \beta_0 \mathbf{g}^{n-1} - \nabla p^n, \quad (5)$$

where the forcing term g is given by

$$g = -\gamma \nabla^2 \mathbf{u} + \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^T)] - \frac{\rho}{\Delta t} (\mathbf{u} \cdot \nabla \mathbf{u}) + \mathbf{f}, \quad (6)$$

and, from experience, the constant γ must be chosen as $\gamma \geq 2.0 \|\mu\|_\infty$ for Gear's Method. The parameters α and β are time dependent. The indices 2, 1, and 0, represent the time levels $n+1$, n , and $n-1$, respectively. Those coefficients are given by

$$\alpha_2 = \frac{\Delta t_0 + 2\Delta t_1}{\Delta t_0 + \Delta t_1}, \quad \alpha_1 = \frac{\Delta t_0 + \Delta t_1}{\Delta t_0}, \quad \alpha_0 = \frac{\Delta t_1^2}{\Delta t(\Delta t_0 + \Delta t_1)}, \quad (7)$$

$$\beta_1 = \frac{\Delta t_2}{2\Delta t_2 - \Delta t_1}, \quad \beta_0 = \frac{-\Delta t_1}{\Delta t_2 - \Delta t_1}. \quad (8)$$

The semi-implicit scheme above is solved numerically with a multigrid-multilevel method for each time step. When the time-step is constant, the coefficients above are the usual ones found in the extrapolated Gear's Method, that is, $\alpha_2 = \frac{3}{2}$, $\alpha_1 = \frac{4}{2}$, $\alpha_0 = \frac{1}{2}$, $\beta_1 = 2$, and $\beta_0 = -1$. Just a linear CFL condition must be applied for the overall stability of the scheme. The time-step is selected such that

$$\Delta t \geq \min \left\{ \min \left\{ \frac{\Delta x}{|u_{i,j}|}, \frac{\Delta y}{|v_{i,j}|} \right\}, \min(\Delta x, \Delta y) \right\}, \quad (9)$$

where the first is the advection restriction and the second guarantees that the error in the time discretization is not greater than the discretization in space.

2.2. Discretization of the Projection Step

A fractional-step projection method is applied to the Navier-Stokes equations, resulting a scheme which is second-order accurate for the velocities (the current implementation is only first-order for the pressure). Next, for the sake of the simplicity, the main ideas are presented for homogeneous Dirichlet boundary conditions, and no external forces considered.

The projection method relies on the Hodge decomposition of the velocity field (Popinet, 2003). Classically, the Hodge decomposition is a splitting of an arbitrary vector field into two orthogonal components, one divergence-free, and the other the gradient of a scalar field. If $\mathbf{w} = \mathbf{w}(\mathbf{x})$ is a smooth vector field defined on a region Ω , with

$$\int_{\partial\Omega} \mathbf{w} \cdot \mathbf{n} \, dS = 0, \quad (10)$$

then $\mathbf{w}$ can be written as

$$\mathbf{w} = \mathbf{w}_d + \nabla\phi, \quad (11)$$

where $\nabla \cdot \mathbf{w}_d = 0$ and $\mathbf{w}_d \cdot \mathbf{n} = 0$ on the boundary of Ω . Taking the divergence of Eq. (11) yields the Poisson equation

$$\nabla^2\phi = \nabla \cdot \mathbf{w}, \quad (12)$$

while the normal component of $\nabla \cdot \mathbf{w}_d = 0$ yields the boundary condition

$$\frac{\partial\phi}{\partial\mathbf{n}} = \mathbf{w} \cdot \mathbf{n} \quad \text{on } \partial\Omega. \quad (13)$$

The divergence-free field is then defined as

$$\mathbf{w}_d = \mathbf{w} - \nabla\phi, \quad (14)$$

where ϕ is obtained as the solution of the Poisson problem (Eq. 12). This defines the projection of $\mathbf{w}$ onto the space of the divergence-free fields (Chorin and Marsden, 1979).

In practice, the solution of Eqs. (5)-(6) is computed in three steps by a projection method specially designed for composite grids. The following development assumes that the pressure and other scalar functions are placed at the centers of the computational cells, while velocity and other vector function components are placed at cell edges. First, in the *parabolic step*, one must solve, for a provisional velocity $\mathbf{u}_{i,j}^*$, the implicit parabolic equation (Eq. 5),

$$\frac{\rho_{i,j}^{n+1}}{\Delta t} [\alpha_2 \mathbf{u}_{i,j}^* - \alpha_1 \mathbf{u}_{i,j}^n + \alpha_0 \mathbf{u}_{i,j}^{n-1}] = \gamma \nabla^2 \mathbf{u}_{i,j}^* + \beta_1 \mathbf{g}_{i,j}^n + \beta_0 \mathbf{g}_{i,j}^{n-1} - \nabla p_{i,j}^n, \quad (15)$$

where

$$\mathbf{g} = -\gamma \nabla^2 \mathbf{u}_{i,j} + \nabla [\mu_{i,j} (\nabla \mathbf{u}_{i,j} + \nabla \mathbf{u}_{i,j}^T)] - \frac{\rho_{i,j}}{\Delta t} (\mathbf{u}_{i,j} \cdot \nabla \mathbf{u}_{i,j}) + \mathbf{f}_{i,j}$$

with $\gamma = 2.0 \parallel \mu_{i,j} \parallel_{\infty}$. Second, in the *elliptic step*, one must solve the pressure-correction Poisson equation

$$\nabla \cdot \left(\frac{1}{\rho_{i,j}} \nabla \phi \right)^{n+1} = \frac{\alpha_2}{\Delta t} \nabla \cdot \mathbf{u}_{i,j}^* \quad (16)$$

with homogeneous Neumann boundary conditions.

In the third step, to complete the projection, the provisional velocity field $\mathbf{u}_{i,j}^*$ is decomposed using the pressure correction obtained in the second step,

$$\mathbf{u}_{i,j}^{n+1} = \mathbf{u}_{i,j}^* - \frac{\Delta t}{\alpha_2 \rho_{i,j}^{n+1}} \nabla \phi^{n+1}, \quad (17)$$

giving as a result $\mathbf{u}^{n+1}$, a discretely divergence-free vector field defined on the entire composite grid. This step is referred to as the *decomposition step*.

2.3. Space Discretization

The unknowns $\mathbf{u} = (u, v)$ and p are placed on the composite grid in a MAC-fashioned manner. In this discretization, the cells covering the domain are “mass control volumes” where scalar quantities (e.g., pressure and divergence) are defined at the cell centers and vector quantities (e.g., velocity and pressure gradient) have their vertical component defined at the middle of the vertical cell edges. Given a computational cell (i, j) , its left and right mid edges will either have indices (i, j) as show the Fig. 1.

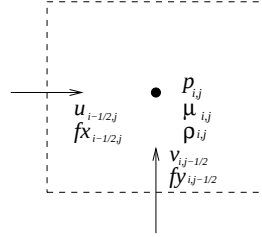


Figure 1: Schematic representation of a MAC cell.

Second- or third-order interpolation schemes are employed at coarse/fine grid interfaces to provide boundary data for *ghost cells* appended at fine grid borders, formally preventing the difference operators from being defined there (Roma, 1996).

2.4. Turbulence Model

The sub-grid turbulent viscosity ($\mu_{i,j}$) is implemented and calculated with the Smagorinsky model. The Smagorinsky model depends only on the deformation ratio and can introduce artificial turbulent viscosity in the laminar portion of the flow (Matos et al., 1998). In the Smagorinsky’s model, the turbulent viscosity is given by

$$\mu_t = \mu + \rho [L(\Delta)]^2 \sqrt{2S_{ij}S_{ij}}, \quad (18)$$

where $S_{i,j}$ is the ratio-of-strain tensor and $L(\Delta)$, a factor that has units of length, depends on the size of the computational mesh and of its anisotropy. With different spacing in the horizontal and in the vertical direction, one has

$$L(\Delta) = C_s \sqrt{\Delta x \Delta y}, \quad (19)$$

where C_s is the Smagorinsky constant (usually taken between 0.1 and 0.2). The filtered ratio-of-strain tensor is given by

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \quad (20)$$

3. ADAPTIVE MESH

In this Section, the algorithm previously described is extended for an adaptive hierarchy of nested rectangular grids. In the first subsection, it is described the creation of the grid hierarchy. In the second one, it is shown the multilevel-multigrid method used to solve the Poisson equation and the use of the ghost cells.

3.1. Composite Grid Creation

The existence of local phenomena and the need for more grid points to accurately capture them suggest the application of local mesh refinement techniques. One such technique will be briefly described here. For more details, see Berger and Oliger (1984), Berger and Colella (1989), Berger and Rigoutsos (1991), Roma *et al.* (1999), Roma (1996), Almgren *et al.* (1998), and Martin and Colella (2000).

The implementation of the adaptive mesh refinement (AMR) is based on a sequence of nested grids with successively finer spacing in space. The fine grids are formed by evenly dividing coarse cells by a refinement ratio, r ($r = 2$), in each direction. Specific selection criteria dynamically create or remove rectangular fine patches as resolution requirements change. If the finer levels are defined as $l = 1, 2, \dots, l_{finest}$ and the set of disjoint rectangular grids $G_{l,k}$, $k = 1, 2, \dots, n_l$, one has

$$\{\text{level } l\} = \bigcup_k G_{l,k}, \quad (21)$$

with $G_{l,j} \cap G_{l,k} = \emptyset$, $j \neq k$ (two different grids in the same level do not overlap).

Cells tagged for refinement are grouped into rectangular patches using the clustering algorithm developed by Berger and Rigoutsos (1991). These rectangular patches form the grids and the process is repeated until all the levels have been created. Grids must be *properly nested*. This means that they must satisfy the following two properties:

- A fine starts and ends at the corner of a cell in the coarser level;
- There must be at least one level ($l - 1$) cell in some level ($l - 1$) grid separating a grid cell at level l from a cell at level ($l - 2$), in the north, south, east, and west directions.

These properties assure that any refined grid must be bordered by either a physical boundary, or another grid at the same level, or be surrounded by grids in the next coarser level (Martin and Cartwright, 1996). In the first case, physical boundary conditions will be applied. In the second case, the boundary conditions will simply be transferred from the neighboring, same-level grid. The later case, boundary conditions are interpolated employing coarse and fine cell values. Fig. 2 illustrates grids which are properly nested.

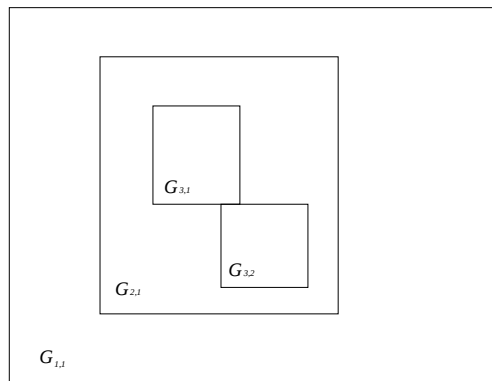


Figure 2: Properly nested grids.

An important note on defining the initial condition on composite grids at time $t = 0$ consists in the fact that one must project the initial velocity field to discretely enforce the divergence constraint. Usually, few iterates are also computed in order to define an initial pressure field. At every time step, after performing the projection, velocity components (but not the pressure) are averaged down from fine onto underlying coarser grids to have their values defined in all the levels (Almgren, 1998).

3.2. Multilevel-Multigrid

The multigrid method is the fastest and most efficient iterative algorithm for solving linear systems arising from elliptic differential equations (Eq. 16). This method offers convergence rates independently of the size of the problem and is therefore very effective for solving large scale computational-intensive problems. One iteration of a simple multigrid V-cycle consists of smoothing the error using a relaxation technique (e.g. Red-Black Gauss-Seidel), solving an approximation to the smooth error equation on a coarse grid, interpolating the error correction to the fine grid, and finally adding the error correction to the approximation. In this Section, a brief description of the multigrid technique (MG) for the adaptive mesh (AMR) is presented.

The same general idea behind the standard multigrid technique still holds for the multilevel approach. Special care must be taken only at coarse-fine level interfaces.

In the multilevel technique the residual equation is solved in the coarse grid instead of the source term itself so that the coarse and fine grids can be taken into account simultaneously (this method is usual called *correction storage* scheme - Miyashita and Yamada, 2005).

The problem to be solved on a composite MAC-grid is:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \phi \right) = rhs, \quad (22)$$

where rhs is just the right hand side determined by the needs of the time-stepping algorithm. All indices were suppressed for a simpler notation.

The residual-correction problem on a grid at the finest level is defined by

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \psi \right) = R \doteq rhs - \nabla \cdot \left(\frac{1}{\rho} \nabla \phi \right) \quad (23)$$

where R is the residual and ψ the correction.

In order to transfer the value from fine grids to coarse grids, the restriction operator ($\mathfrak{R}$) that takes an simple average of all the fine grids that corresponds to one coarse grids is defined. In this way, the coarse residual problem is given by:

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \psi \right) = \mathfrak{R}_{l-1}^l R \quad (24)$$

In order for a solution to be unique when pure Neumann or Periodic boundary conditions exist, the compatibility condition to the discrete right-hand side needs to be assured. The compatibility condition then just says that the sum of all right-hand side have to be zero ($\int_{\Omega} rhs = \sum (rhs)_{i,j} = 0$).

After the residual restriction, the Red-Black Gauss-Seidel is used to perform a point relaxation for Poisson's Equation. Before each relaxation or residual computation, it is necessary to update ghost cells around the border of each grid employing either boundary conditions or neighboring grids or both. After relaxations, it is also necessary to synchronize the cells shared by adjacent grids.

In order to apply the correction to the fine grids, is defined the *prolongation operator* ($\wp$) that makes values in the fine grids from the ones in the coarse grids by something like a linear interpolation (Miyashita and Yamada, 2005). In this way, is possible to apply the correction to ϕ to update the solution

$$\phi_l = \phi_l + \wp_l^{l-1} \psi \quad (25)$$

these processes are usually called coarse-grid correction and is iterated until R is sufficiently small.

On composite grids, mainly at coarse/fine interfaces the residual computation must preserve the *solvability condition*

$$\int_{\Omega} \nabla \left(\frac{1}{\rho} \nabla \phi \right) \cdot \mathbf{n} = \int_{\partial\Omega} rhs \cdot \mathbf{n}, \quad (26)$$

which leads to a flux correction for the cell-centered variable ϕ .

In the flux correction step, instead of simply averaging the residual, the restriction is performed on the cell-edge values by simply averaging them onto the parent cell edges they cover. For example, considering a fine grid to the left of a coarse grid (Fig. 3) and placing the control volume around the coarse cell adjacent to the coarse/fine and summing the fluxes passing in and out of the box the Laplacian operator is now written as

$$L\phi_{i,j} = \frac{F_{i+\frac{1}{2},j}^c - F_{i-\frac{1}{2},j}^{ave}}{\Delta x^c} + \frac{F_{i,j+\frac{1}{2}}^c - F_{i,j-\frac{1}{2}}^c}{\Delta y^c}, \quad (27)$$

$$F_{i-\frac{1}{2},j}^{ave} = \frac{1}{2}(F_{top}^f + F_{bot}^f), \quad (28)$$

and,

$$F_{i+\frac{1}{2},j} = \frac{1}{\Delta x}(\phi_{i+1,j} - \phi_{i,j}), \quad (29)$$

$$F_{i-\frac{1}{2},j} = \frac{1}{\Delta x}(\phi_{i,j} - \phi_{i-1,j}), \quad (30)$$

$$F_{i,j+\frac{1}{2}} = \frac{1}{\Delta y}(\phi_{i,j+1} - \phi_{i,j}), \quad (31)$$

$$F_{i,j-\frac{1}{2}} = \frac{1}{\Delta y}(\phi_{i,j} - \phi_{i,j-1}), \quad (32)$$

where F is the flux in a volume control, the indices c and f denote the coarse and fine mesh respectively.

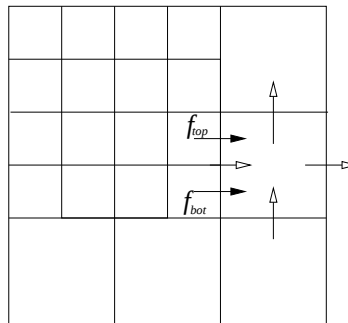


Figure 3: The black arrows indicate the fluxes from the coarse grids and white arrows indicate fluxes from the fine grids where the new evaluation to the Laplacian on composite grids is obtained through the flux control

In the solution of parabolic equations, the multilevel-multigrid method is also applied, though the flux-correction step is not needed. More details on the multilevel-multigrid method employed, including details on the interpolation stencils can be found in the works by Roma (1999), Martin and Cartwright (1996), Martin and Colella (2000) and Miyashita and Yamada (2005).

4. RESULTS

This section presents the numerical results obtained in computations performed for the Navier-Stokes equations related to the model problem described in the Section 2. Here, it is provided evidence that this methodology yields to a second-order accurate approximation to the velocities and at least first order for the pressure. To estimate the order of the algorithm, the test problem (Eq. 35) is solved on uniform and on composite grids of progressively increasing resolution. For each grid size, the norm of the error is calculated using the computed solution and the known exact solution. The L_2 norm of the error, which is used to verify the convergence order, is defined by

$$\|e\|_2 = \|\theta_e - \theta_d\|_2 = \frac{\sum \sqrt{(\theta_e - \theta_d)^2 \Delta x \Delta y}}{A}, \quad (33)$$

where A is the area of the domain, and θ represents the variable that is being evaluated exactly (subindex e) and discretely (subindex d). The ratio between two successive meshes (n and $2n$) gives an approximation of the convergence order q

$$q = \frac{(\|e\|_2)_n}{(\|e\|_2)_{2n}}. \quad (34)$$

Note that, when calculating norms of the error over the entire domain, coarse cells covered by fine cells are not included.

Following Brown *et al.* (2000), two convergence tests illustrating second-order accuracy of the method with *no flow* through the boundaries in a square domain are presented. The solution is advanced up to $t = 0.1$. In the first problem, consider constant fluid properties, with the density and viscosity equal to 1.0. This test is ran on uniform and on composite grids doubling the number of cells in each case. Base levels are 64×64 , 128×128 , 256×256 , and 512×512 . The exact solution of the test problem is given by:

$$\begin{aligned} p &= -\frac{w'(t)}{2\pi} \sin(2\pi(x - w(t))) (\sin(2\pi y) - 2\pi y + \pi) \\ &\quad - \mu \cos(2\pi(x - w(t))) (-2\sin(2\pi y) + 2\pi y - \pi) \\ u &= \cos(2\pi(x - w(t))) (3y^2 - 2y) \\ v &= 2\pi \sin(2\pi(x - w(t))) y^2 (y - 1), \end{aligned} \quad (35)$$

with $w(t) = 1 + \sin(2\pi t^2)$ on $\Omega = [-1.0, 0.8] \times [-1.0, 0.8]$. For the composite grid case only two levels are considered and the grids are strategically located to verify the exchange information between sibling grids, physical domain and coarse-fine interfaces. On a base level 32×32 , the subgrids are 48×24 , 48×40 and 24×32 . In the subsequent meshes, these numbers of cells are doubled. Fig. 4 shows the grid patches position on the composite mesh test.

Tables 1 and 2 show the convergence order results for uniform and for composite grids, with constant fluid material properties. The second-order convergence is obtained in both cases for the velocity field, and at least first-order convergence is obtained for the pressure field.

In the second case, the fluid properties are variable and tests are ran on both uniform and composite meshes with the same configuration previously shown. Smagorinsky's model is employed (see Section 2.4) with constant $C_s = 0.1$. Since the exact velocity field is known from (Eq. 35) the "exact" turbulent viscosity (μ_t) can be easily obtained:

$$\mu_t = \mu_o + \rho C_s^2 \Delta x \Delta y \sqrt{2 \left[\left(\frac{\partial u}{\partial x} \right)^2 + \frac{1}{2} \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right]}, \quad (36)$$

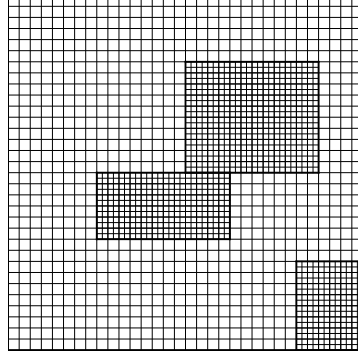


Figure 4: Grid patches locations on the composite-mesh test.

Domain	L_2						
	$n = 64$	Ratio	$n = 128$	Ratio	$n = 256$	Ratio	$n = 512$
u	4.2145e-03	3.78	1.1142e-03	3.87	2.8757e-04	3.96	7.2607e-05
v	4.5268e-03	3.82	1.1845e-03	3.90	3.0397e-04	3.96	7.6836e-05
p	1.5454e-01	3.13	4.9349e-02	3.57	1.3825e-02	3.58	3.9741e-03

Table 1: L_2 errors and convergence ratios on uniform meshes for constant fluid properties.

where the $\mu_o = 1.0$ and the variable density (ρ) is given by:

$$\rho = 1.0 + [\sin(2\pi x)\sin(2\pi y)]^2 e^{-t}. \quad (37)$$

The convergence order in the variable fluid property case are displayed in Tables 3 and 4, for both uniform and composite grids. They clearly show second-order accuracy for the velocity field, and at least first-order accuracy for the pressure in the L_2 norm.

5. CONCLUSIONS

An accurate and efficient numerical method for solving the Navier-Stokes equations was presented. The numerical method is a time-split scheme that combines a novel semi-implicit discretization of Gear's method with a time-discretization of the projection for the Navier-Stokes equations. These equations were solved on a locally refined grids using multilevel algorithm for the elliptic and parabolic equations in a MAC-grid cell. The numerical method is robust and has minimal cost. Some of the capabilities of the method were illustrated with numerical examples in two dimensions on uniform and composite grids with constant and variable fluid properties.

Domain	L_2						
	$n_{bot} = 64$	Ratio	$n_{bot} = 128$	Ratio	$n_{bot} = 256$	Ratio	$n_{bot} = 512$
$\Delta \cdot \mathbf{u}$	9.6413e-06	4.0	2.4101e-06	4.0	6.0250e-07	4.0	1.5065e-07
u	1.0970e-03	3.87	2.8362e-04	3.95	7.1842e-05	3.98	1.8057e-05
v	1.1443e-03	3.89	2.9444e-04	3.95	7.4606e-05	3.98	1.8727e-05
p	4.1720e-02	3.58	1.1645e-02	3.29	3.5391e-03	2.89	1.2230e-03

Table 2: L_2 errors and convergence ratios on composite meshes for constant fluid properties.

Domain	L_2						
	$n = 64$	Ratio	$n = 128$	Ratio	$n = 256$	Ratio	$n = 512$
u	4.1282e-03	3.84	1.0739e-03	3.87	2.7764e-04	3.95	7.0208e-05
v	4.3618e-03	3.81	1.1446e-03	3.88	2.9485e-04	3.95	7.4654e-05
p	1.4229e-01	3.15	4.5194e-02	3.59	1.2595e-02	3.51	3.5840e-03

Table 3: L_2 errors and convergence ratios on uniform meshes for variable fluid properties.

Domain	L_2						
	$n_{bot} = 64$	ratio	$n_{bot} = 128$	ratio	$n_{bot} = 256$	ratio	$n_{bot} = 512$
$\Delta \cdot \mathbf{u}$	9.6413e-06	4.0	2.4101e-6	4.0	6.0250e-07	4.0	1.5065e-07
u	1.0641e-03	3.87	2.7522e-04	3.96	6,9570e-05	3.99	1.5062e-05
v	1.1168e-03	3.88	2.8796e-04	3.96	7.2779e-05	4.0	1.8180e-05
p	3.8148e-02	3.59	1.0622e-02	3.29	3.2242e-03	2.85	1.1328e-03

Table 4: L_2 errors and convergence ratios on composite meshes for variable fluid properties.

The algorithm presented here assure second-order accuracy in space and in time for the velocity field, and at least first-order for the pressure field. The method can be extended to three dimensions problem and to other boundary conditions besides no flow condition (e.g. periodic conditions) retaining the same efficiency characteristics.

Acknowledgements

This works was supported by research grant CNPq (National Research Council), Brazil.

References

- Almgren, A. S., Bell, J. B., Colella, P., Howell, L. H. and Welcome, M. L., 1998. *A Conservative Adaptive Projection Method for the Variable Density Incompressible Navier-Stokes Equations*. Journal of Computational Physics, Vol. 142 pp 1-46.
- Badalassi, V. E., Cenicerros, H. D. and Banerjee S., 2003. *Computation of Multiphase Systems with Phase Field Models*. Journal of Computational Physics, Vol.190, pp 371-397.
- Bell, J. B., Berger, M. J., Saltzman, J. S., and Welcome M.,1991. *Three-Dimensional Adaptive Mesh Refinement for hyperbolic Conservation Laws*. Technical Report UCRL-JC-108794, LLNL.
- Bell, J. B., Colella, P., and Glaz, H. M., 1989. *A Second-Order Projection Method for Incompressible Navier-Stokes Equations*. Journal of Computational Physics, Vol.85, pp 257.
- Berger, M. J. and Colella, P., 1989. *Local adaptive mesh refinement for shock hydrodynamics*. Journal of Computational Physics, Vol. 82, pp. 67-84.
- Berger, M. J. and Leveque, R. J., 1998. *Adaptive Mesh Refinement Using Wave-Propagation Algorithms for Hyperbolic Systems*. SIAM Journal on Numerical Analalysys, Vol. 35 pp 2298-2316.
- Berger, M. J. and Oliger, J., 1984. *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*. Journal of Computational Physics, Vol. 53, pp 484.
- Berger, M. J. and Rigoutsos, I., 1991. *An Algorithm for Hyperbolic Partial Differential Equations*. IEE Trans. Systems Man and Cybernet, Vol. 21 pp 1278-1286.
- Cenicerros, H. D. and Roma, A. M., 2004. *A Multi-Phase Flow Method with a Fast, Geometry-Based Fluid Indicator*. Journal of computational Physics, Vol. 205 pp 391-400.

- Chorin, A. and Marsden, J., 1979. *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag. New York, Heidelberg, Berlin.
- Douglas, J. J. and Dupont, T., 1971. *Alternating-Direction Galerkin Methods on Rectangles*. Numerical Solution of Partial Differential Equations-II, Academic Press. New York, pp 133-213.
- Howell, L. H. and Bell, J. B., 1997. *An Adaptive-Mesh Projection Method for Viscous Incompressible Flow*. SIAM Journal on Scientific Computing, Vol. 18 pp 996.
- Martin D. and Colella P., 2000. *A Cell-Centered Adaptive Projection Method for the Incompressible Euler Equations*. Journal of Computational Physics, Vol. 163 pp 271-312.
- Martin, D. and Cartwright, K., 1996. *Solving Poisson's Equation Using Adaptive Mesh Refinement*. Technical Report UCB/ERL M96/66, UC Berkeley.
- Matos, A., Pinho F. A. A. and Silveira-Neto, A., 1998. *Large-Eddy Simulation of Turbulent Flow over a Two-Dimensional Cavity with Temperature Fluctuations*. International Journal of Heat and Mass Transfer, Vol. 42 pp 49-59.
- Miyashita H. and Yamada Y., 2005. *Practical Improvements of Multi-Grid Iteration for Adaptive Mesh Refinement Method*. Fluid Dynamics Research, Vol. 36 pp 137-152.
- Popinet, S., 2003. *Gerris: Tree-Based Adaptive Solver for the Incompressible Euler Equations in Complex Geometries*. Journal of Computational Physics, Vol. 190, pp 572-600.
- Roma A. M., 1996. *A Multilevel Self-Adaptive Version of the Immersed Boundary Method*. Ph.D. thesis, Courant Institute of Mathematical Sciences - New York University.
- Roma, A. M., Peskin C. S., and Berger M. J., 1999. *An Adaptive Version of the Immersed Boundary Method*. Journal of Computational Physics, Vol. 153, pp 509-534.
- Smagorinsky, J., 1963 *General Circulation Experiments with Primitive Equations*. Monthly Weather Review, Vol. 91, pp 99-164.
- Trangenstein, J. A., 2002. *Multi-Scale Iterative Techniques and Adaptive Mesh Refinement for Flow in Porous Media*. Advances in Water Resources, Vol. 25 pp 1175-1213.