A Second Order Accurate Projection Method for the Incompressible Navier-Stokes Equations on Fully Adaptive Grids

Chohong Min and Frédéric Gibou

13th November 2005

Abstract

We present an unconditionally stable second order accurate projection method for the incompressible Navier-Stokes equations on fully adaptive Cartesian grids, allowing for constraint-free mesh generation. We employ quadtree and octree data structures as an efficient means to represent the grid. We use the supra-convergent Poisson solver of Min *et al.* [18], a second order accurate semi-Lagrangian method to update the momentum equation and an unconditionally stable backward difference scheme to treat the diffusion term. We sample all the variables at the grid nodes, producing a scheme that is straightforward to implement. We propose two and three dimensional examples to demonstrate second order accuracy for the velocity field and the divergence free condition in the L^1 and L^∞ norms.

1 Introduction

The incompressible Navier-Stokes equations describe the motion of fluid flows and are therefore used in countless applications in science and engineering. In non-dimensional form these equations read:

$$\begin{aligned} &U_t + (U\cdot\nabla)U + \nabla p = \mu\Delta U + F && \text{in } \Omega,\\ &\nabla\cdot U = 0 && \text{in } \Omega,\\ &U|_{\partial\Omega} = U_b && \text{on } \partial\Omega, \end{aligned}$$

where p is the pressure, F is the sum of the external forces and μ is the viscosity coefficient. Ω represents the domain in which the velocity field U is to be found and $\partial\Omega$ denotes the boundary of the domain, where the velocity field can be prescribed. These equations lack an evolution equation for pressure, which thus only plays a role in ensuring that the velocity field is divergence free. As a consequence, most numerical methods in the primitive variables are fractional methods, i.e. they first solve the momentum equation ignoring the effects of pressure, and then project the velocity onto the divergence free vector space. Starting with the seminal work of Chorin [7], several projection methods have been introduced, see e.g. the work of Kim and Moin [12], Kan [11], Bell *et al.* [3] and the references therein. The MAC grid configuration [10], where the pressure is stored at the cells' center and where the velocity components are stored at their respective cells' faces, is often the preferred arrangement. This is mainly due to the fact that it produces methods that offer a straightforward mechanism to enforce *discretely* the incompressibility condition $\nabla\cdot u = 0$. However, other arrangements have been shown to produce high order accurate schemes for the velocity field, without enforcing the incompressibility condition at the discrete level (see e.g. the work of E *et al.* [9], Almgren *et al.* [2], the review by Brown *et al.* [6] and the references therein).

Physical phenomena have differences in length scales and numerical approximations on uniform grids are in such cases extremely inefficient in terms of C.P.U. and memory requirement. This stems from the fact that only a small fraction of the domain needs high grid resolution to correctly approximate the solution, while other parts of the domain can produce accurate solutions on coarser grids (for example in regions where the solution experiences smooth variations). As a consequence adaptive mesh refinement strategies, starting with the work of Berger and Oliger [5] for compressible flows, have been proposed in order to concentrate the computational effort where it is most needed. In the original work of Berger *et al.* [5, 4], a fine Cartesian grid is hierarchically embedded into a coarser grid. Almgren *et al.* [1] then introduced a projection method for the variable density incompressible Navier-Stokes equations on nested grids. Sussman *et al.* extended this method to two-phase flows [25]. Within this block structured grid approach, a multigrid approach is necessary to solve the Poisson equation. In contrast, the methods on quadtrees/octrees presented in [17, 16, 18, 20] build *one* linear system of equations that is then solved with standard iterative linear solvers [21].

One of the main reasons behind considering block structured grids was to avoid spurious shock reflections occurring when the solution sweeps portions of the grid with different levels of resolution. However, in the case of incompressible flows, shocks do not exist and a more optimal fully adaptive grid structure can be used. One of the main difficulties in solving the Navier-Stokes equations on irregular grids is in solving the Poisson equation associated with the incompressibility condition. Rather recently, Popinet [20] introduced a Navier-Stokes solver using an octree data structure. In this work, the discretization of the Poisson equation at one cell's center involves cells that are not necessarily adjacent to it. As a consequence, a nonsymmetric linear system of equations was obtained and graded octrees only were considered in order to ease the implementation. Later, Losasso *et al.* [17] introduced a symmetric discretization of the Poisson equation in the context of free surface flows. In this case, the discretization at one cell's center only involves adjacent cells, therefore producing a symmetric linear system of equations. Moreover, this method is straightforward to implement and does not require any constraint on the grid. This approach produces first order accurate solutions in the case of a fully adaptive mesh and is found to be second order accurate in the case of a graded mesh. In this case, the pressure fluxes defined at the faces are the same for a large cell and its adjacent smaller cells. Using ideas introduced in [15], Losasso *et al.* then extended this method to second order accuracy. In [18], Min *et al.* introduced a second order accurate method to solve the Poisson equation on fully adaptive grids as well. A hallmark of this approach is that the solution's gradients are found to second order accuracy as well. In this case, the linear system is nonsymmetric but is proven to be diagonally dominant. In this paper, we propose a second order accurate Navier-Stokes solver on fully adaptive grids, making use of the Poisson solver introduced in [18].

2 Spatial Discretization

The physical domain in two (resp. three) spatial dimensions is discretized into squares (resp. cubes), and we use a standard quadtree (resp. octree) data structure to represent this partitioning. For example, consider the case depicted in figure 1 in the case of two spatial dimensions: The root of the tree is associated with the entire domain that is then split into four cells of equal sizes, called the

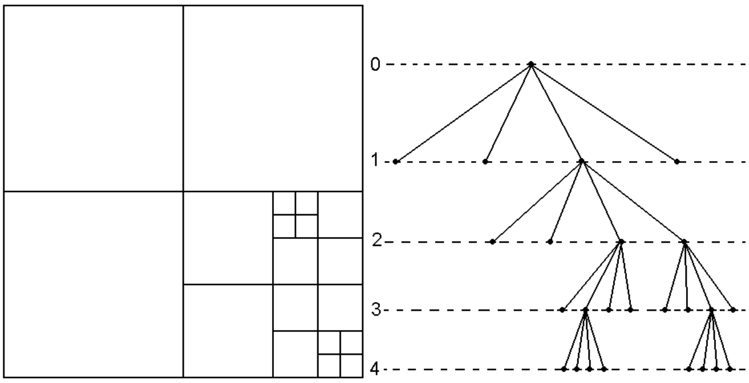

Figure 1: Discretization of a two dimensional domain (left) and its quadtree representation (right). The entire domain corresponds to the root of the tree (level 0). Then each cell can be recursively subdivided further to four children. In this example, this tree is ungraded, since the difference of level between cells exceeds one.

children of the root. The discretization proceeds recursively, i.e. each cell can be in turn split into four children until the desired level of detail is achieved. In three spatial dimensions, the domain (root) is split in eight cubes (children) and each cell can be recursively split in the same manner. We refer the interested reader to the books of Samet [23, 22] for more details on quadtree/octree data structures.

The level of a cell is set to be zero if it is associated with the root and is incremented by one for each new generation of children. A tree in which the difference of level between adjacent cells is at most one is called a graded tree. Meshes associated with graded trees are often used in the case of finite element methods in order to produce procedures that are easier to implement. In [20], Popinet also uses a graded grid to simplify the finite difference formulas associated with his discretizations. As a consequence, such methods must introduce a large amount of extra grid cells in regions where they are not necessarily needed, consuming some computational resources that cannot be spent elsewhere, eventually limiting the highest level of detail that can be achieved. In fact, Moore [19] demonstrates that the cost of transforming an arbitrary quadtree into a graded quadtree could involve 8 times as many grid nodes. Weiser [26] proposed a rough estimate for the three dimensional case and concluded that as much as 71 times as many grid nodes could be needed for balancing octrees. In contrast, we do not impose any constraint on the difference of level between two adjacent cells in the proposed method, allowing for a fully adaptive mesh generation.

3 Numerical Methods

In this section, we present an unconditionally stable second order accurate projection method for the incompressible Navier-Stokes equations. All the variables are stored at the nodes, producing a scheme that is straightforward to implement. We use the quadtree and octree data structures described in section 2 and we allow for fully adaptive grids, hence removing the difficulties associated with grid generations. We use the supra-convergent Poisson solver of Min *et al.* [18], a second order

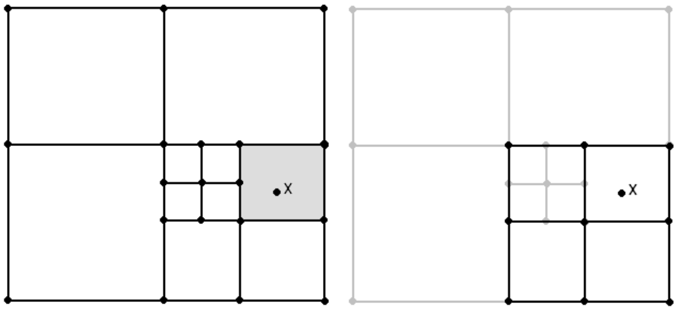

Figure 2: Quadratic interpolation in quadtree: The shaded cell is the smallest cell containing the location x where the data must be interpolated at. The parent cell of the shaded cell has a 3×3 locally uniform grid that enables a straightforward quadratic Hermite interpolation.

accurate semi-Lagrangian method to update the momentum equation and an unconditionally stable backward difference scheme to treat the diffusion term.

3.1 Second Order Accurate Semi-Lagrangian Method

Semi-Lagrangian schemes are extensions of the Courant-Isaacson-Rees [8] method for hyperbolic equations. They are unconditionally stable and therefore allow for large time steps, which is a particularly desirable feature in an adaptive setting since for standard explicit schemes the time step restriction imposed by the CFL condition is proportional to the smallest grid cell. The general idea behind semi-Lagrangian methods is to reconstruct the solution by integrating numerically the equation along characteristic curves, starting from any grid point x_i and tracing back the departure point x_d in the upwind direction. Interpolation formulas are then used to recover the value of the solution at such points. Consider for example the linear advection equation

$$\phi_t + U \cdot \nabla \phi = 0,$$

where U is an externally generated velocity field (i.e. does not depend on ϕ). Then $\phi^{n+1}(x_i) = \phi^n(x_d)$, where x_i is any grid point and x_d is the corresponding departure point from which the characteristic curve originates from. In this work, we use the following second order explicit mid-point rule for locating the departure point, as in [27]:

$$\begin{aligned} \hat{x} &= x^{n+1} - \tfrac{\Delta t}{2} \cdot U^n(x^{n+1}), \\ x_d^n &= x^{n+1} - \Delta t \cdot U^{n+\frac{1}{2}}(\hat{x}), \end{aligned}$$

where we define the velocity at the mid time step $t^{n+1/2}$ as a linear combination of the velocities at the two previous time steps t^n and t^{n-1}, i.e. $U^{n+\frac{1}{2}} = \frac{3}{2}U^n - \frac{1}{2}U^{n-1}$.

Since $\hat{x}$ is not guaranteed to be on a grid node, a procedure must be provided to interpolate the value of $U^{n+\frac{1}{2}}(\hat{x})$ from the values of $U^{n+1/2}$ defined at the nodes. Likewise, $\phi^n(x_d^n)$ must be interpolated from the values of ϕ^n defined at the nodes. Piecewise multilinear interpolation schemes on nonuniform grids are often used in conjunction with semi-Lagrangian methods (see e.g. [17, 24]). In this work, we use a quadratic Hermite interpolation [13], which is constructed from the solution's

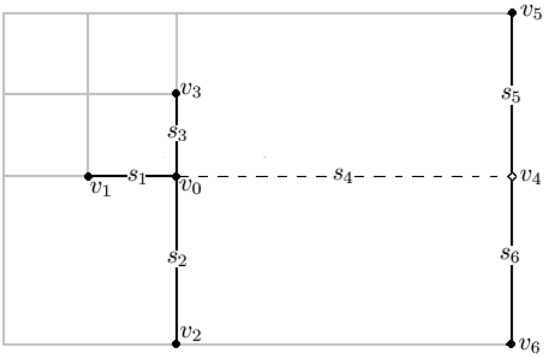

Figure 3: Local structure around a node v_0 in a quadtree mesh: At most one node in the two Cartesian directions might not exist. In this case, we define a ghost node (here v_4) to be used in the discretizations.

values at nine distinct nodes in two spatial dimensions (27 in three spatial dimensions). Since the local structure of a nonuniform cell is arbitrary, we use the four children of the parent cell to select a uniform grid of 3 × 3 nodes in two spatial dimensions (3 × 3 × 3 in three spatial dimensions) as illustrated in figure 2. Similarly, the discretization of the momentum equation in the projection method of section 3.4 requires the definition of x_d^{n-1}, which is given by

$$\begin{aligned} \hat{x} &= x^{n+1} - \Delta t \cdot U^n(x^{n+1}), \\ x_d^{n-1} &= x^{n+1} - 2\Delta t \cdot U^n(\hat{x}). \end{aligned}$$

3.2 Basic Finite Differences on Nonuniform Cartesian Grids

In this section, we provide the formulas we use to compute the first and second order derivatives on a nonuniform mesh.

3.2.1 Two Spatial Dimensions

Consider a node v_0 in a two dimensional nonuniform grid as depicted in figure 3. Standard finite difference approximations for the first and second order derivatives in the y-direction, namely

$$D_y f(v_0) = \tfrac{f_3 - f_0}{s_3} \tfrac{s_2}{s_2+s_3} + \tfrac{f_0 - f_2}{s_2} \tfrac{s_2}{s_2+s_3},$$

and

$$D_{yy} f(v_0) = \tfrac{f_3 - f_0}{s_3} \tfrac{2}{s_2+s_3} - \tfrac{f_0 - f_2}{s_2} \tfrac{2}{s_2+s_3},$$

where $f_i = f(v_i)$, are respectively second and first order accurate since all the nodes involved in these discretizations are aligned on the same axis.

On the other hand, standard finite difference approximations for the first and second order derivatives in the x-direction first require the definition of a ghost node in order to construct a locally Cartesian stencil[1]. Given a discrete function f sampled at the nodes of a quadtree grid, the

[1] We emphasize that in two spatial dimensions at most one ghost node must be defined in order to construct a locally Cartesian stencil.

function's value at the ghost node v_4 is defined as a linear combination of the function's values at the nodes v_5 and v_6:

$$f(v_4) = \frac{s_6 f_5 + s_5 f_6}{s_5 + s_6}.$$

Then the standard finite difference approximations for the first and second order derivatives in the x-direction read:

$$\tilde{D}_x f(v_0) = \frac{f_4 - f_0}{s_4} \frac{s_1}{s_1 + s_4} + \frac{f_0 - f_1}{s_1} \frac{s_4}{s_1 + s_4},$$

and

$$\tilde{D}_{xx} f(v_0) = \frac{f_4 - f_0}{s_4} \frac{2}{s_1 + s_4} - \frac{f_0 - f_1}{s_1} \frac{2}{s_1 + s_4}.$$

A straightforward Taylor analysis demonstrates that $\tilde{D}_x$ is locally[2] first order accurate and that $\tilde{D}_{xx}$ is locally inconsistent because the linear interpolation defining the ghost node v_4 introduces errors in the transversal direction. These spurious terms can be removed by a weighted average of the approximations in the y-direction, i.e. we define

$$D_x f(v_0) = \tilde{D}_x f(v_0) - \frac{s_1 s_5 s_6}{2 s_4 (s_1 + s_4)} D_{yy} f(v_0),$$

and

$$D_{xx} f(v_0) = \tilde{D}_{xx} f(v_0) - \frac{s_5 s_6}{s_4 (s_1 + s_4)} D_{yy} f(v_0).$$

In this case, $D_x f(v_0)$ is locally second order accurate. Likewise, $D_{xx} f(v_0)$ is locally first order accurate, and becomes second order accurate if the grid is locally uniform at v_0, i.e. $s_1 = s_4$ and $s_2 = s_3$. We note that one-sided difference formulas are used at the wall's boundaries. Similarly, formulas can be derived for D_y and D_{yy} in the case where a ghost node is needed in their discretizations.

3.2.2 Three Spatial Dimensions

Consider a node v_0 in a three dimensional nonuniform grid as depicted in figure 4. Standard finite difference approximations for the first and second order derivatives in the z-direction, namely

$$D_z f(v_0) \quad = \quad \tfrac{f_6 - f_0}{s_6} \tfrac{s_3}{s_3 + s_6} + \tfrac{f_0 - f_3}{s_3} \tfrac{s_6}{s_3 + s_6},$$

and

$$D_{zz} f(v_0) \quad = \quad \tfrac{f_6 - f_0}{s_6} \tfrac{2}{s_3 + s_6} - \tfrac{f_0 - f_3}{s_3} \tfrac{2}{s_3 + s_6},$$

where $f_i = f(v_i)$, are respectively second and first order accurate since all the nodes involved in these discretizations are aligned on the same axis.

On the other hand, standard finite difference approximations for the first and second order derivatives in the x- and y- directions first require the definition of two ghost nodes in order to construct a locally Cartesian stencil[3]. Referring to figure 4, the function's value at the ghost nodes

[2] By definition, a cell is locally nonuniform if its size is different from the size of at least one of its neighbors. A cell is locally uniform if its size is equal to that of all of its neighbors.

[3] We emphasize that in three spatial dimensions at most two ghost nodes must be defined in order to construct a locally Cartesian stencil.

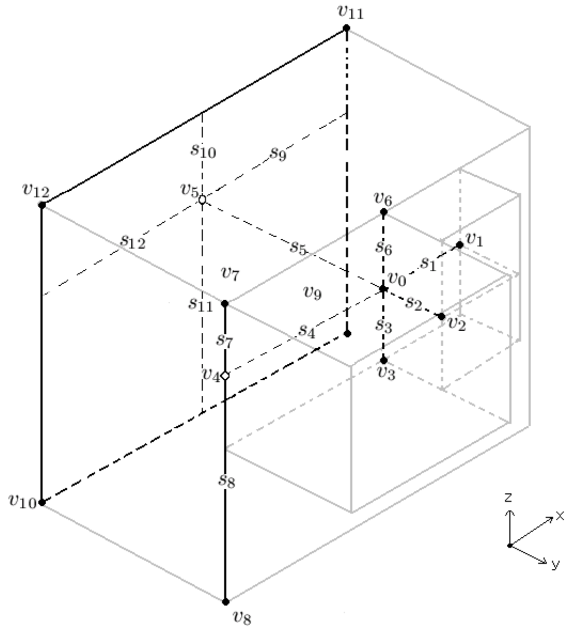

Figure 4: Neighboring vertices of a vertex three spatial dimensions.

v_4 and v_5 are defined as a linear combination of the function's values at the nodes v_7, v_8 v_9, v_{10}, v_{11}, and v_{12}:

$$f(v_4) = \frac{s_7 f_8 + s_8 f_7}{s_7 + s_8},$$

and

$$f(v_5) = \frac{s_{11}s_{12}f_{11} + s_{11}s_9 f_{12} + s_{10}s_{12}f_9 + s_{10}s_9 f_{10}}{(s_{10}+s_{11})(s_9+s_{12})}.$$

Then the standard finite difference approximations for the first and second order derivatives in the x- and y- directions read:

$$\tilde{D}_x f(v_0) = \frac{f_4 - f_0}{s_4}\frac{s_1}{s_1+s_4} + \frac{f_0 - f_1}{s_1}\frac{s_4}{s_1+s_4},$$

$$\tilde{D}_y f(v_0) = \frac{f_2 - f_0}{s_2}\frac{s_5}{s_2+s_5} + \frac{f_0 - f_5}{s_5}\frac{s_2}{s_2+s_5},$$

$$\tilde{D}_{xx} f(v_0) = \frac{f_4 - f_0}{s_4}\frac{2}{s_1+s_4} - \frac{f_0 - f_1}{s_1}\frac{2}{s_1+s_4},$$

and

$$\tilde{D}_{yy} f(v_0) = \frac{f_2 - f_0}{s_2}\frac{2}{s_2+s_5} - \frac{f_0 - f_5}{s_5}\frac{2}{s_2+s_5}.$$

As in the two dimensional case, a Taylor analysis reveals that $\tilde{D}_x$ and $\tilde{D}_y$ are locally first order accurate and that $\tilde{D}_{xx}$ and $\tilde{D}_{yy}$ are locally inconsistent because the linear interpolations defining the ghost nodes v_4 and v_5 introduce errors in the transversal directions. These spurious terms can be removed by a weighted average of the approximations in the x-, y- and z- directions, i.e. we define

$$\begin{aligned}
D_x f(v_0) &= \tilde{D}_x f(v_0) + \tfrac{s_7 s_8}{2} \tfrac{s_1}{s_4(s_1+s_4)} D_{zz} f(v_0), \\
D_y f(v_0) &= \tilde{D}_y f(v_0) + \tfrac{s_{10} s_{11}}{2} \tfrac{s_2}{s_5(s_2+s_5)} D_{zz} f(v_0) + \tfrac{s_9 s_{12}}{2} \tfrac{s_2}{s_5(s_2+s_5)} D_{xx} f(v_0), \\
D_{xx} f(v_0) &= \tilde{D}_{xx} f(v_0) - \tfrac{s_7 s_8}{s_4(s_1+s_4)} D_{zz} f(v_0), \\
D_{yy} f(v_0) &= \tilde{D}_{yy} f(v_0) - \tfrac{s_{10} s_{11}}{s_5(s_2+s_5)} D_{zz} f(v_0) - \tfrac{s_9 s_{12}}{s_5(s_2+s_5)} D_{xx} f(v_0).
\end{aligned}$$

In this case, $D_x f(v_0)$ and $D_y f(v_0)$ are locally second order accurate. Likewise, $D_{xx} f(v_0)$ and $D_{yy} f(v_0)$ are locally first order accurate, and become second order accurate if the grid is locally uniform at v_0, i.e. $s_1 = s_4$, $s_2 = s_3$ and $s_3 = s_6$. We note that one-sided difference formulas are used at the wall's boundaries. Similarly, formulas can be derived for D_z and D_{zz} in the case where a ghost node is needed in their discretizations.

3.3 Supra-Convergent Poisson Solver

Consider a Cartesian domain $\Omega \in \mathbb{R}^n$ with boundary $\partial\Omega$ and the Poisson equation with Dirichlet boundary conditions:

$$\begin{aligned}
\Delta u = f \quad \text{on } \Omega, \\
u_{|\partial\Omega} = g.
\end{aligned}$$

3.3.1 Discretization and Analysis

We have presented in Min *et al.* [18] a Poisson solver on fully adaptive grids that produces second order accurate solutions with second order accurate gradients. Moreover, the discretization associated with one grid nodes involves only two (resp. three) adjacent cells in two (resp. three) spatial dimensions, producing a scheme straightforward to implement. Finally, the linear system associated with this solver is an M-matrix, which allows the application of the following theorem:

Theorem 1 *(supra-convergence - from [18]) Let u_h be the solution of $\Delta_h u_h = f$ with a Dirichlet boundary condition on at least one node, where Δ_h is a discretization of the Laplace operator that is m^{th} order accurate at locally uniform cells and n^{th} order accurate at locally non-uniform cells, with $m, n \geq 1$. Suppose that the matrix associated with $-\Delta_h$ is an M-matrix. Then, the approximation is globally* $\min(m, n+1)^{th}$ *order accurate in L^∞ norm.*

This theorem implies that second order accuracy in the maximum norm can be achieved with discretizations that are only first order accurate at locally nonuniform cells, but second order accurate at locally uniform cells. Based on this fact, we proposed in [18] a second order accurate discretization of the Poisson equation making use of the finite difference formulas defined in section 3.2 for $D_{xx}u$, $D_{yy}u$ and $D_{zz}u$. The solutions' gradients are also computed using the formulas in section 3.2 (i.e. $\nabla u = (D_x u, D_y u, Du)$) and are found to be second order accurate in the L^∞ norm.

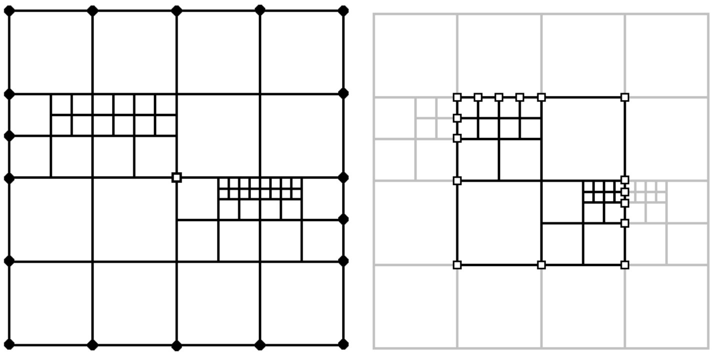

Figure 5: Two-step procedure to solve the Poisson equation with Neumann boundary conditions. The squares (□) represent Dirichlet boundary conditions while the bullet points (•) represent Neumann boundary conditions. Left: In the first step the Poisson equation is solved in the entire domain with Neumann boundary conditions on the domain's boundary and a single Dirichlet boundary condition (here at the center of the domain). Right: In the second step, the Poisson equation is solved on a small portion of the domain with Dirichlet boundary conditions on that domain's boundary.

3.3.2 Neumann Boundary Condition

Projection methods for the Navier-Stokes equations require the solution of a Poisson equation with Neumann boundary conditions on the domain's boundary. Since the corresponding linear system is singular, a solution must be picked out of the infinite possibilities, for example by adding a Dirichlet boundary condition at one node. In this case however, we observed in [18] that this procedure lowers the accuracy of the solution's gradients to first order. We then proposed a simple two-step procedure illustrated in figure 5 that produces solutions with second order accurate gradients: First, in order to select one particular solution, we impose a Dirichlet boundary condition at one node (for example, the center of the domain). This introduces spurious errors in the solution's gradients localized near that center node as depicted in figure 6. These errors are removed by considering a small portion of the domain containing the center node. Letting Ω_p be that portion, we solve the Poisson equation on Ω_p with Dirichlet boundary conditions on its boundary $\partial\Omega_p$. The Dirichlet boundary conditions are the solution' values obtained in the first step. Numerical experiments demonstrate that the gradients are second order accurate in the L^∞ norm (see [18]). We note that the cost of solving the Poisson equation in the second step is negligible since it only uses a small portion of the domain.

3.4 Projection Method

Consider momentum equation:

$$U_t + (U \cdot \nabla)U + \nabla p = \mu\Delta U + F.$$

The Crank-Nicholson scheme has often been used for discretizing implicitly the viscosity term [3, 12]. However, in the case where the convection term is treated with a semi-Lagrangian method, a difficulty arises: The corresponding pressure is not defined at the grid nodes, making the projection step complicated to implement in conjunction with a Crank-Nicholson scheme. The backward

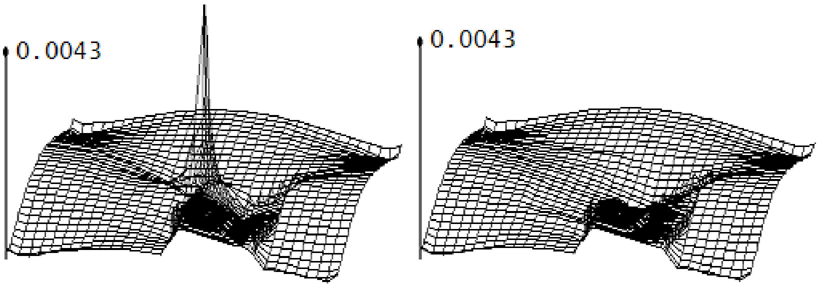

Figure 6: Solving $-\Delta u = f$ with $u = \cos(x)\cos(y)$. Plot of $||\nabla u - \nabla u_h||_\infty$ before (left) and after (right) applying the procedure in section 3.3.2

differentiation formula offers a more convenient choice, since in this case the corresponding pressure is defined *at* the grid nodes [27]. The discretization of the momentum equation using a backward differentiation formula and a semi-Lagrangian method for the convection term can be written as:

$$\frac{1}{\Delta t}\left(\frac{3}{2}U^{n+1} - 2U_d^n + \frac{1}{2}U_d^{n-1}\right) + \nabla p^{n+1} = \mu\Delta U^{n+1} + F^{n+1} \tag{1}$$

This equation is solved using a three-step projection method approach: First, given the velocity field U^n at time t^n, an intermediate velocity U^* is calculated by ignoring the pressure component:

$$\frac{1}{\Delta t}\left(\frac{3}{2}U^* - 2U_d^n + \frac{1}{2}U_d^{n-1}\right) = \mu\Delta_h U^* + F^{n+1}. \tag{2}$$

Second, in order for the velocity U^{n+1} at time t^{n+1} to satisfy the incompressibility condition $\nabla \cdot U^{n+1} = 0$ the second step defines a potential function ϕ^{n+1} through the solution of the following Poisson equation:

$$\Delta_h \phi^{n+1} = \frac{1}{\Delta t}\left(\nabla_h \cdot U^*\right). \tag{3}$$

In the last step, the fluid velocity U^{n+1} at the new time step is projected to the divergence free field:

$$U^{n+1} = U^* - \Delta t \cdot \nabla_h \phi^{n+1}. \tag{4}$$

Taking the divergence of equation 4 and using the relation given by equation 3 yields a velocity field U^{n+1} that is indeed divergence free. The relation between ϕ^{n+1} and the p^{n+1} is found by first combining equation 2 and equation 4 to get the following expression relating U^{n+1} to ϕ^{n+1}:

$$\frac{1}{\Delta t}\left(\frac{3}{2}U^{n+1} - 2U_d^n + \frac{1}{2}U_d^{n-1}\right) + \frac{3}{2}\nabla\phi^{n+1} = \mu\Delta U^{n+1} + F^{n+1} + \Delta t\mu\Delta\nabla\phi^{n+1}, \tag{5}$$

and then comparing this expression to equation 1. This leads the following expression relating ϕ^{n+1} to p^{n+1}:

$$\nabla p^{n+1} = \frac{3}{2}\nabla\phi^{n+1} - \Delta t\mu\Delta\nabla\phi^{n+1}.$$

An analysis similar to that in [6, 12] demonstrates that the following boundary conditions for U^* and ϕ^{n+1} are sufficient to ensure second order accuracy for the velocity field:

$$\begin{array}{rcl} N \cdot U^*|_{\partial\Omega} & = & N \cdot U^{n+1}|_{\partial\Omega}, \\ T \cdot U^*|_{\partial\Omega} & = & T \cdot U^{n+1}|_{\partial\Omega} + \Delta t \cdot T \cdot \nabla\phi^n, \\ \nabla\phi \cdot n|_{\partial\Omega} & = & 0, \end{array}$$

where N and T denote the normal and tangent vectors at the boundary, respectively.

We note that the first step of the projection method computes the intermediate velocity U^* by solving the following convection-diffusion equation:

$$\left(\frac{3}{2}Id - \Delta t \mu \Delta_h\right) U^* = 2U_d^n - \frac{1}{2}U_d^{n-1} + \Delta t F^{n+1},$$

with Dirichlet boundary conditions at the domain's boundary. As demonstrated in [18], the matrix associated with $-\Delta_h$ is an M-matrix, implying that $\frac{3}{2}Id - \Delta t \mu \Delta_h$ is also an M-matrix. In turn, the supra-convergence theorem of [18] thus guarantees that U^* is second order accurate in the L^∞ norm. We also note that we use a starting routine to guess the initial value $\nabla\phi^0$ as described in Brown *et al.* [6].

4 Examples

In this section, we present numerical evidences that the proposed projection method yields second order accuracy for the velocity field and the divergence free condition in the L_1 and the L_∞ norms. All the examples were tested on highly arbitrary grids to demonstrate that this scheme is applicable to fully adaptive grids. The linear systems are strictly diagonally dominant, and are solved using the BiCGSTAB algorithm with an ILU preconditioner [21].

4.1 Single Vortex in Two Spatial Dimensions

Consider a domain $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}]^2$ and a single vortex flow with an exact solution of:

$$\begin{array}{rcr} u(x, y, t) & = & -\cos(x)\sin(y)\cos(t), \\ v(x, y, t) & = & \sin(x)\cos(y)\cos(t), \\ p(x, y, t) & = & -\frac{1}{4}\cos^2(t)\left(\cos(2x) + \cos(2y)\right), \end{array}$$

We use the grid depicted in figure 7 and impose Dirichlet boundary conditions on the domain's boundary. We emphasize that the difference of level between some cells and their neighbors exceeds one, demonstrating the ability of our method to produce second order accurate solutions on arbitrary grids. The time step is chosen as $\Delta t = 5 \times \Delta x_s$, where Δx_s is the size of the finest grid cell. Table 1 demonstrates the second order accuracy of the velocity field in the L^1 and L^∞ norms while table 2 demonstrates the second order accuracy for the divergence free condition in the L^1 and L^∞ norms.

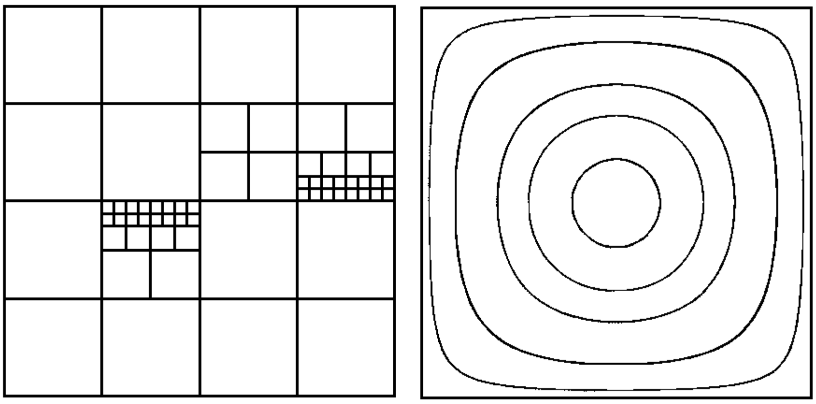

Figure 7: Arbitrarily generated quadtree (left) and streamlines of the numerical solution (right) for example 4.1.

Effective Resolution	$\|\|U - U_h\|\|_\infty$	Order	$\|\|U - U_h\|\|_1$	order
32^2	5.71×10^{-2}		2.91×10^{-2}	
64^2	1.57×10^{-2}	1.85	7.06×10^{-3}	2.04
128^2	2.54×10^{-3}	2.63	1.17×10^{-3}	2.59
256^2	3.92×10^{-4}	2.69	1.82×10^{-4}	2.68
512^2	8.38×10^{-5}	2.22	3.42×10^{-5}	2.41

Table 1: Accuracy of the velocity field in the L^1 and L^∞ norms for example 4.1.

Effective Resolution	$\|\|\nabla \cdot U_h\|\|_\infty$	Order	$\|\|\nabla \cdot U_h\|\|_1$	order
32^2	1.50×10^{-1}		2.31×10^{-2}	
64^2	5.59×10^{-2}	1.42	5.55×10^{-3}	2.05
128^2	1.34×10^{-2}	2.05	7.30×10^{-4}	2.92
256^2	3.44×10^{-3}	1.96	1.21×10^{-4}	2.58
512^2	8.35×10^{-4}	2.04	1.84×10^{-5}	2.72

Table 2: Accuracy of the divergence free condition in the L^1 and L^∞ norms for example 4.1.

Effective Resolution	$\|\|U - U_h\|\|_\infty$	Order	$\|\|U - U_h\|\|_1$	order
32^2	4.21×10^{-1}		2.78×10^{-1}	
64^2	2.04×10^{-1}	1.03	3.97×10^{-2}	2.80
128^2	2.23×10^{-2}	3.19	5.58×10^{-3}	2.83
256^2	2.92×10^{-3}	2.93	9.16×10^{-4}	2.60
512^2	8.17×10^{-4}	1.83	1.86×10^{-4}	2.29

Table 3: Accuracy of the velocity field in the L^1 and L^∞ norms for example 4.2.

4.2 Influx/Outflux Flow in Two Spatial Dimensions

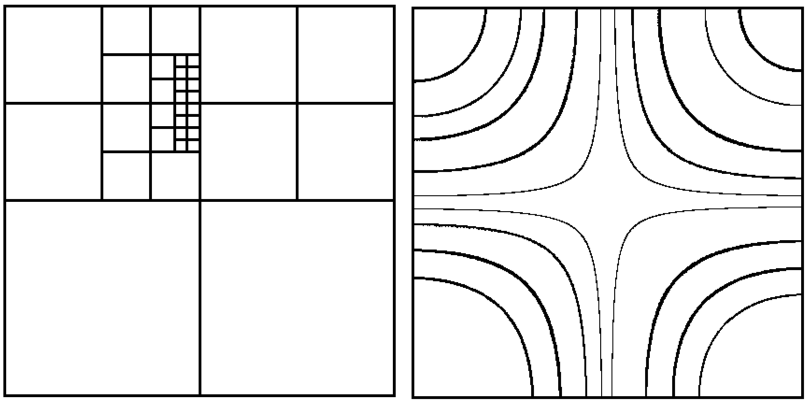

Figure 8: Arbitrarily generated two dimensional grid (left) and streamlines of the numerical solution (right) for example 4.2.

This example is taken from [14]. Consider a domain $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}]^2$ and a flow with a nonzero normal velocity field at the domain's boundary with an exact solution is given of:

$$\begin{aligned} u(x,y,t) &= -\cos(t)\sin^2(x)\sin(2y), \\ v(x,y,t) &= \cos(t)\sin(2x)\sin^2(y), \\ p(x,y,t) &= \tfrac{1}{4}\cos(t)\,(2+\cos(x))\,(2+\cos(y))\,. \end{aligned}$$

We use the grid depicted in figure 8 and impose Dirichlet boundary conditions on the domain's boundary. We emphasize that the difference of level between some cells and their neighbors exceeds one as in the previsous example. The time step is chosen as $\Delta t = 5 \times \Delta x_s$, where Δx_s is the size of the finest grid cell. Table 3 demonstrates the second order accuracy of the velocity field in the L^1 and L^∞ norms while table 4 demonstrates the second order accuracy for the divergence free condition in the L^1 and L^∞ norms.

Effective Resolution	$\lVert \nabla \cdot U_h \rVert_\infty$	Order	$\lVert \nabla \cdot U_h \rVert_1$	order
32^2	5.23×10^{-1}		8.07×10^{-2}	
64^2	2.80×10^{-1}	0.89	1.29×10^{-2}	2.64
128^2	3.25×10^{-2}	3.11	2.15×10^{-3}	2.58
256^2	1.17×10^{-2}	1.46	1.74×10^{-4}	3.62
512^2	3.26×10^{-3}	1.84	2.01×10^{-5}	3.11

Table 4: Accuracy of the divergence free condition field in the L^1 and L^∞ norms for example 4.2.

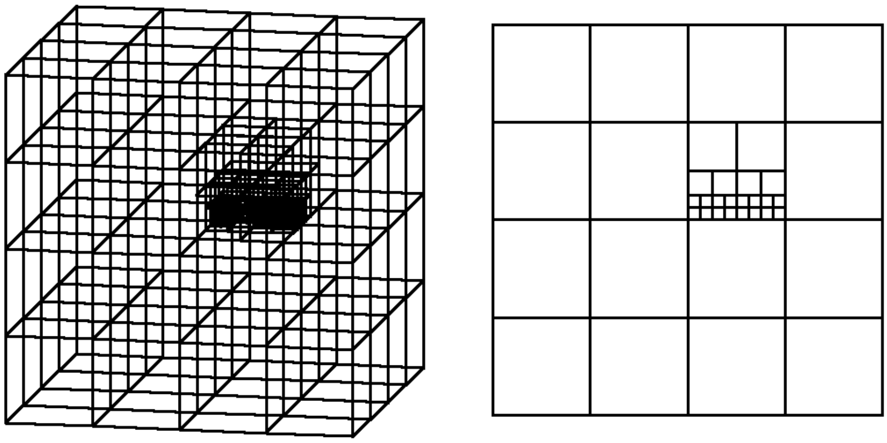

Figure 9: Arbitrarily generated three dimensional grid (left) and one of its cross section (right) used in example 4.3.

Effective Resolution	$\|\|U - U_h\|\|_\infty$	Order	$\|\|U - U_h\|\|_1$	order
32^2	6.92×10^{-2}		2.60×10^{-2}	
64^2	2.64×10^{-2}	1.38	9.73×10^{-3}	1.41
128^2	6.28×10^{-3}	2.07	2.49×10^{-3}	1.96
256^2	1.07×10^{-3}	2.54	4.98×10^{-4}	2.32
512^2	2.23×10^{-4}	2.26	3.94×10^{-5}	2.90

Table 5: Accuracy of the velocity field in the L^1 and L^∞ norms for example 4.3.

Effective Resolution	$\|\|\nabla \cdot U_h\|\|_\infty$	Order	$\|\|\nabla \cdot U_h\|\|_1$	order
32^2	3.56×10^{-1}		4.82×10^{-2}	
64^2	1.36×10^{-1}	1.38	1.59×10^{-2}	1.60
128^2	3.74×10^{-2}	1.87	2.30×10^{-3}	2.78
256^2	9.55×10^{-3}	1.96	2.94×10^{-4}	2.96
512^2	2.56×10^{-3}	1.90	3.94×10^{-5}	2.90

Table 6: Accuracy of the divergence free condition in the L^1 and L^∞ norms for example 4.3.

4.3 Three Spatial Dimensions

Consider a domain $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}]^3$ and an exact solution defined by:

$$\begin{array}{rcr} u(x,y,z,t) & = & -2\cos(t)\cos(x)\sin(y)\sin(z) \\ v(x,y,z,t) & = & \cos(t)\sin(x)\cos(y)\sin(z) \\ w(x,y,z,t) & = & \cos(t)\sin(x)\sin(y)\cos(z) \\ p(x,y,z,t) & = & \frac{1}{4}\cos^2(t)\left(2\cos(2x)+\cos(2y)+\cos(2z)\right) \end{array}$$

The time step is chosen as $\Delta t = 5 \times \Delta x_s$, where Δx_s is the size of the finest grid cell. Figure 9 depicts the grid used. In particular, the level difference between some cells and their neighbors is larger than one, illustrating the ability of our method to retain second order accuracy on fully adaptive grids. Table 5 demonstrates the second order accuracy of the velocity field in the L^1 and L^∞ norms while table 6 demonstrates the second order accuracy for the divergence free condition in the L^1 and L^∞ norms.

5 Conclusions

We have presented an unconditionally stable second order accurate projection method for the incompressible Navier-Stokes equations on fully adaptive Cartesian grids. Quadtree and octree data structures are used to provide an optimal representation of the mesh. We use the supra-convergent Poisson solver of Min *et al.* [18] to account for the incompressibility condition, a second order accurate semi-Lagrangian method to update the momentum equation and a stiffly stable backward difference scheme to treat the diffusion term. All the variables are sampled at the nodes, producing a scheme that is straightforward to implement. Two and three dimensional examples have been presented to demonstrate second order accuracy for the velocity field and the divergence free condition in the L^1 and the L^∞ norms.

References

[1] A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome. A conservative adaptive projection method for the variable density incompressible navier-stokes equations. *J. Comput. Phys.*, 142:1–46, 1998.

[2] A. S. Almgren, J. B. Bell, and W. G. Syzmczak. A numerical method for the incompressible navier-stokes equations based on an approximate projection. *SIAM J. Sci. Comput.*, 17:358–369, 1996.

[3] J. B. Bell, P. Colella, and H. M. Glaz. A second order projection method for the incompressible navier-stokes equations. *J. Comput. Phys*, 85:257–283, 1989.

[4] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64–84, 1989.

[5] M. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[6] D. Brown, R. Cortez, and M. Minion. Accurate projection methods for the incompressible navier-stokes equations. *J. Comput. Phys.*, 168:464–499, 2001.

[7] A. Chorin. A Numerical Method for Solving Incompressible Viscous Flow Problems. *J. Comput. Phys.*, 2:12–26, 1967.

[8] R. Courant, E. Isaacson, and M. Rees. On the solution of nonlinear hyperbolic differential equations by finite differences. *Comm. Pure Appl. Math.*, 5:243–255, 1952.

[9] W. E and J. G. Liu. Gauge method for viscous incompressible flows. *Comm. Math. Sci.*, 1:317–332, 2003.

[10] F. Harlow and J. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, 8:2182–2189, 1965.

[11] J. V. Kan. A second-order-accurate pressure-correction scheme for viscous incompressible flow. *SIAM J. Sci. Stat. Comput.*, 7:870–891, 1986.

[12] J. Kim and P Moin. Application of a fractional-step method to incompressible navier-stokes equations. *J. Comput. Phys.*, 59:308–323, 1985.

[13] D. Kincaid and W. Cheney. *Numerical Analysis: Mathematics of Scientific Computing.* Brooks/Cole Publishing Co., Pacific Grove, CA, USA, 2002.

[14] R. Leveque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33:627–665, 1996.

[15] K. Lipnikov, J. Morel, and M. Shashkov. Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes. *J. Comput. Phys.*, 199:589–597, 2004.

[16] F. Losasso, R. Fedkiw, and S. Osher. Spatially adaptive techniques for level set methods and incompressible flow. *Computers and Fluids (in press).*

[17] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *ACM Trans. Graph. (SIGGRAPH Proc.)*, pages 457–462, 2004.

[18] C. Min, F. Gibou, and H. Ceniceros. A simple second order accurate finite difference scheme for the variable coefficient poisson equation on fully adaptive grids. *CAM report 05-29, Submitted to J. Comput. Phys.*

[19] D. Moore. The cost of balancing generalized quadtrees. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 305–312, 1995.

[20] S. Popinet. Gerris: A tree-based adaptive solver for the incompressible euler equations in complex geometries. *J. Comput. Phys.*, 190:572–600, 2003.

[21] Y. Saad. *Iterative methods for sparse linear systems.* PWS Publishing, 1996. New York, NY.

[22] H. Samet. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, New York, 1989.

[23] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing and GIS.* Addison-Wesley, New York, 1990.

[24] J. Strain. Tree methods for moving interfaces. *J. Comput. Phys.*, 151:616–648, 1999.

[25] M. Sussman, A. S. Algrem, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome. An adaptive level set approach for incompressible two-phase flow. *J. Comput. Phys*, 148:81–124, 1999.

[26] A. Weiser. *Local-Mesh, Local-Order, Adaptive Finite Element Methods with a Posteriori Error Estimators for Elliptic Parital Differential Equations.* PhD thesis, Yale University, June 1981.

[27] D. Xiu and G. Karniadakis. A semi-lagrangian high-order method for navier-stokes equations. *J. Comput. Phys*, 172:658–684, 2001.