Non-Graded Adaptive Grid Approaches to the Incompressible Navier-Stokes Equations

Frédéric Gibou [1], Chohong Min [2], Hector D. Ceniceros [3]

(*Communicated by John Lowengrub and Mark Sussman*)

Abstract: We describe two finite difference schemes for simulating incompressible flows on nonuniform meshes using quadtree/octree data structures. The first one uses a cell-centered Poisson solver that yields first-order accurate solutions, while producing symmetric linear systems. The second uses a node-based Poisson solver that produces second-order accurate solutions and second-order accurate gradients, while producing nonsymmetric linear systems as the basis for a second-order accurate Navier-Stokes solver. The grids considered can be non-graded, i.e. the difference of level between two adjacent cells can be arbitrary. In both cases semi-Lagrangian methods are used to update the intermediate fluid velocity in a standard projection framework. Numerical results are presented in two and three spatial dimensions.

1 Introduction

Incompressible flows are at the center of countless applications in physical and biological sciences. Uniform Cartesian grids used in numerical simulations are limited in their ability to resolve small scale details and as a consequence nonuniform meshes are often desirable in practice. Since the work of Berger and Oliger (1984) on compressible flows, adaptive mesh refinement techniques have been widely used, see e.g the approach of Almgren et al (1998) (and the references therein) for the Navier-Stokes equations on block structured grids. In the case of incompressible flows, adaptive mesh strategies are quite common (see e.g. Ham (2002); Sussman (1999); Ceniceros and Roma (2004); Cristini et al (2001); Yang et al (2006); Zheng et al (2005); Anderson et al (2005); Cristini and Renardy (2006) and the references therein), but implementations based on the optimal quadtree/octree data structure is less common.

In the case of a standard projection method (see e.g. Chorin (1967); Brown et al (2001)), the most computationally expensive part comes from solving a Poisson equation for the pressure. This is also the limiting part in terms of accuracy, since high order accurate (and unconditionally stable) semi-Lagrangian methods exist for the convective part. In Popinet (2003), Popinet proposed a second-order nonsymmetric numerical method to study the incompressible Navier-Stokes equations using an octree data structure. In Losasso (2004), Losasso *et al.* proposed a symmetric solution of the Poisson equation for non-graded adaptive grids, i.e. grids for which the size's ratio between adjacent cells is not constrained. This work relies on the observation that, in the case of the Poisson equation, first-order perturbations in the location of the solution yield consistent schemes (see Gibou et al (2002)). Losasso *et al.* then extended the work of Lipnikov et al (2004) to the case of arbitrary grids to propose a second-order accurate symmetric discretization of the Poisson equation Losasso (in press). In Min et al (2006b), Min *et al.* proposed a second-order accurate scheme for the Poisson equation that also yields second-order accurate gradients. In this case the linear system is nonsymmetric, but diagonally dominant, i.e. for

[1] Mechanical Engineering Department & Computer Science Department, University of California, Santa Barbara, CA 93106.

[2] Mathematics Department, University of California, Santa Barbara, CA 93106.

[3] Mathematics Department, University of California, Santa Barbara, CA 93106.

each row the diagonal element is greater or equal to the sum of the nondiagonal elements. This Poisson solver was used for solving the Navier-Stokes equations to second-order accuracy in Min et al (2006a) using the projection methods described in Brown et al (2001). In this paper, we describe two finite difference schemes for simulating incompressible flows on nonuniform meshes using quadtree/octree data structures. The first one uses a cell-centered Poisson solver that yields first-order accurate solutions, while producing symmetric linear systems (see Losasso (2004)). The second uses a node-based Poisson solver that produces second-order accurate solutions and second-order accurate gradients, while producing nonsymmetric linear systems (see Min et al (2006b) for a supra-convergent Poisson solver and Min et al (2006a) for a second-order accurate Navier-Stokes solver). The grids considered can be non-graded, i.e. the difference of level between two adjacent cells can be arbitrary. In both cases semi-Lagrangian methods are used to update the intermediate fluid velocity in a standard projection framework. We present numerical results in two and three spatial dimensions to complement the analysis of Losasso (2004); Min et al (2006b,a).

2 The Navier-Stokes Equations

The motion of fluids is described by the incompressible Navier-Stokes equations for the conservation of momentum and mass:

$$\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \mathbf{f} + \mu\Delta u, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{2}$$

where $\mathbf{u} = (u, v, w)$ is the velocity field, $\mathbf{f}$ accounts for the external forces such as gravity and where the spatially constant density of the fluid has been absorbed in the pressure p. We assume the viscosity parameter μ to be constant.

3 First-Order Accurate Symmetric Navier-Stokes Solver on Octrees

3.1 Cell-Centered Arrangement

In Losasso (2004), Losasso *et al.* proposed a solver for the incompressible Euler equations on non-graded adaptive grids. The domain is tiled with cells as depicted in figure 1 and the mesh is refined automatically in order to capture the local details critical to realistic simulations and coarsened elsewhere. An octree data structure is used (see Samet (1989)) for efficient processing and the different variables are stored as depicted in figure 1: The velocity components u, v and w are stored at the cell faces while the pressure is stored at the center of the cell. This is the standard MAC grid arrangement used in previous works (see e.g. Harlow and Welch (1965)). However, in the case of nonuniform meshes it is more convenient to store the other scalar quantities such as the density ρ at the *nodes* of each cell. This stems from the fact that interpolations are more difficult with cell-centered data as discussed in Strain (1999).

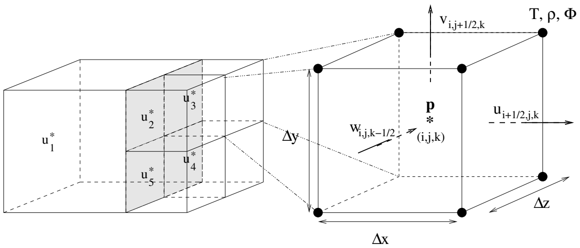

Figure 1: Left: the domain is tiled with cells of sizes varying according to the refinement criterion. Right: Zoom of one computational cell. The velocity components u, v and w are defined on the cell faces while the pressure p is defined at the center of the cell. The other scalar quantities are stored at the nodes.

3.2 Projection Method

A standard first-order accurate projection method Chorin (1967) (see also Brown et al (2001)) is used to solve equations (1) and (2): First an intermediate velocity $\mathbf{u}^*$ is computed over a time step $\triangle t$, ignoring the pressure term

$$\frac{\mathbf{u}^* - \mathbf{u}}{\triangle t} + \mathbf{u}\cdot\nabla\mathbf{u} \quad = \quad \mathbf{f}. \tag{3}$$

This step, accounting for the convection and the external forces, is followed by a projection step to account for incompressibility and boils down to solving the Poisson equation

$$\nabla^2 p = \frac{1}{\triangle t} \nabla \cdot \mathbf{u}^*. \tag{4}$$

Finally, the new velocity field $\mathbf{u}$ is defined as:

$$\mathbf{u} = \mathbf{u}^* - \Delta t \nabla p. \tag{5}$$

The reader is referred to Losasso (2004) and the references therein for the application of this scheme to the simulations of free surface flows.

3.2.1 Computing the Intermediate Velocity

The intermediate velocity $\mathbf{u}^*$ is found by solving equation (3) using a first-order accurate semi-Lagrangian method. In the case of nonuniform grids, the standard high order accurate upwind methods (see e.g. Harten (1987); Shu and Osher (1989); Liu et al (1996)) traditionally used in the case of uniform grids are not well suited due to their stringent time step restrictions and the complexity of their implementations. On the other hand, semi-Lagrangian methods (see e.g. Staniforth and Cote (1991)) are unconditionally stable and are straightforward to implement.

3.2.2 The Intermediate Velocity Divergence

Equation (14) is solved by first evaluating the right hand side at every grid point in the domain. Then, a linear system for the pressure is constructed and inverted. Consider the discretization of equation (14) for a large cell with dimensions $\triangle x$, $\triangle y$ and $\triangle z$ neighboring small cells as depicted in figure 1 (left). Since the discretization is closely related to the second vector form of Green's theorem that relates a volume integral to a surface integral, we first rescale equation (14) by the volume of the large cell to obtain

$$\mathrm{V_{cell}} \triangle t \nabla^2 p = \mathrm{V_{cell}} \nabla \cdot \mathbf{u}^*. \tag{6}$$

The right hand side of equation (6) now represents the quantity of mass flowing in and out of the large cell within a time step $\triangle t$ in $m^3 s^{-1}$. This can be further rewritten as

$$\mathrm{V_{cell}} \nabla \cdot (\mathbf{u}^* - \triangle t \nabla p) = 0. \tag{7}$$

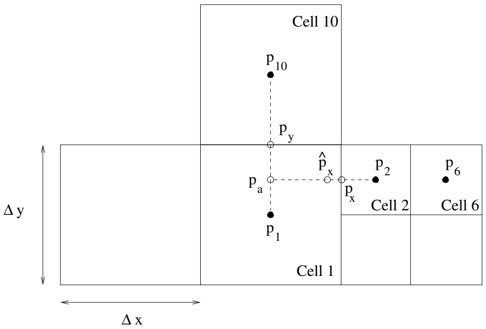

Figure 2: Discretization of the pressure gradient. The pressure values p_1, p_2, p_6, and p_{10} are defined at the center of the cells. p_a represents a weighted average pressure value. p_y defines the y component of the pressure gradient between Cell 1 and Cell 10 defined by standard central differencing. $\hat{p}_x$ represents the discretization of the x component of the pressure gradient between Cell 1 and Cell 2, whereas p_x is a $O(\triangle x)$ perturbation of $\hat{p}_x$.

This equation implies that the term ∇p is most naturally evaluated at the same location as $\mathbf{u}^*$, namely at the cell faces, and that there is a direct correspondence between the components of the vectors ∇p and $\mathbf{u}^*$. That is, there is a direct correspondence between p_x and u, p_y and v, p_z and w, which live on the right and left faces, top and bottom faces, front and back faces, respectively. Moreover, substituting equation (15) into equation (7) implies $\mathrm{V_{cell}} \nabla \cdot \mathbf{u} = 0$ or $\nabla \cdot \mathbf{u} = 0$ as desired.

Invoking the second vector form of Green's theorem, one can write

$$\mathrm{V_{cell}} \nabla \cdot \mathbf{u}^* = \sum_{\text{faces}} (\mathbf{u}^*{}_{\text{face}} \cdot \mathbf{n}) \, \mathrm{A_{face}}, \tag{8}$$

where $\mathbf{n}$ is the *outward* unit normal of the large cell and where A_{face} represents the area of a cell face. In the case of figure 1 (left), the discretization of the x-derivative of the x-component u^* of the velocity field $\mathbf{u}^*$ reads

$$\triangle x \triangle y \triangle z \frac{\partial u^*}{\partial x} = u_2^* \mathrm{A}_2 + u_3^* \mathrm{A}_3 + u_4^* \mathrm{A}_4 + u_5^* \mathrm{A}_5 - u_1^* \mathrm{A}_1, \tag{9}$$

where the *minus* sign in front of $u_1^*\mathrm{A}_1$ accounts for the fact that the unit normal points to the left. In this example, the discretization of $\partial u^*/\partial x$ amounts to

$$\frac{\partial u^*}{\partial x} = \frac{1}{\triangle x}\left(\frac{u_2^*+u_3^*+u_4^*+u_5^*}{4} - u_1^*\right). \tag{10}$$

The y- and z- directions are treated similarly.

3.2.3 Defining the Pressure Derivative to Obtain a Symmetric Linear System

Once, the divergence is computed at the grid nodes, equation (14) is used to construct a linear system of equations for the pressure. Invoking again the second vector form of Green's theorem, one can write

$$\mathrm{V}_{\mathrm{cell}}\nabla\cdot(\triangle t\nabla p) = \sum_{\mathrm{faces}} \left((\triangle t\nabla p)_{\mathrm{face}}\cdot\mathbf{n}\right)\mathrm{A}_{\mathrm{face}}. \tag{11}$$

Therefore, once the pressure gradient is computed at every face, we can carry out the computation in a similar manner as for the divergence of the velocity described above.

In Gibou et al (2002), we showed that $O(\triangle x)$ perturbations in the location of the solution sampling still yield consistent approximations. This was then exploited in Losasso (2004) to define ∇p in order to construct a symmetric linear system. We simply define

$$p_x = \frac{p_2 - p_1}{\triangle},$$

where $\triangle$ can be defined as $\triangle = \triangle x$, which is the size of the large cell or $\triangle = \frac{1}{2}\triangle x$, which is the size of the small cell, or as the Euclidean distance between the locations of p_1 and p_2 or as the distance along the x direction between the locations of p_1 and p_2 etc. We have used the distance along the x direction between the locations of p_1 and p_2.

4 Second-Order Accurate Navier-Stokes Solver on Octrees

4.1 Projection Method

In this case, we choose to store all the variable at the grid nodes in order to develop a simple supra-convergent scheme for the Poisson equation as well as a second-order accurate Navier-Stokes solver. Backward differentiation formulas offer a convenient choice to obtain second-order accuracy in time. In this case, the discretization of the momentum equation is written as:

$$\frac{1}{\Delta t}\left(\frac{3}{2}\mathbf{u}^{n+1} - 2\mathbf{u}_d^n + \frac{1}{2}\mathbf{u}_d^{n-1}\right) + \nabla p^{n+1} = \mu\Delta\mathbf{u}^{n+1} + \mathbf{f}^{n+1}, \tag{12}$$

where $\mathbf{u}_d$ is the velocity at the "departure" point found by tracing back the characteristic curves and interpolated using quadratic interpolation procedures. Equation (12) can be solved using the pressure-free three-step projection method approach of Brown et al (2001): In this method, the intermediate velocity $\mathbf{u}^*$ is first computed by ignoring the pressure component:

$$\frac{1}{\Delta t}\left(\frac{3}{2}\mathbf{u}^* - 2\mathbf{u}_d^n + \frac{1}{2}\mathbf{u}_d^{n-1}\right) = \mu\Delta_h\mathbf{u}^* + \mathbf{f}^{n+1}. \tag{13}$$

Second, a potential function ϕ^{n+1} satisfying the Poisson equation:

$$\Delta_h\phi^{n+1} = \frac{1}{\Delta t}\left(\nabla_h\cdot\mathbf{u}^*\right). \tag{14}$$

is computed to project $\mathbf{u}^*$ onto the divergence free field:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t\alpha\nabla_h\phi^{n+1}, \tag{15}$$

where

$$\alpha = \frac{\mathbf{u}^*\cdot\nabla_h\phi^{n+1}}{\nabla_h\phi^{n+1}\cdot\nabla_h\phi^{n+1}}$$

guarantees that the projection step is an Hodge decomposition at the discrete level Min et al (2006a). The inner product of two functions f and g is computed cell-wise by multiplying the average value for $f\times g$ using the cell's nodes with the volume of the cell.

Taking the divergence of equation (15) and using the relation given by equation (14) yields a velocity field $\mathbf{u}^{n+1}$ that is indeed divergence free (up to the accuracy of the scheme). The relation between ϕ^{n+1} and the p^{n+1} is given by:

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

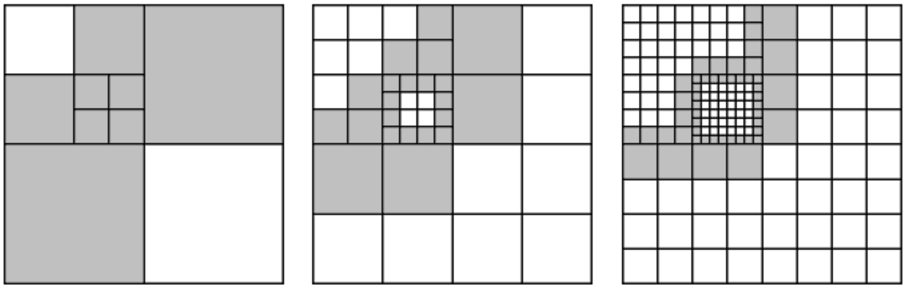

Figure 3: Example of refinement in two spatial dimensions. The total number of cells increases quadratically whereas the number of locally nonuniform cells (shaded) increases linearly. The contribution of nonuniform cells decreases relatively to that of uniform cells.

The boundary conditions on $\mathbf{u}^*$:

$$\begin{aligned}\mathbf{n}\cdot\mathbf{u}^*|_{\partial\Omega} &= \mathbf{n}\cdot\mathbf{u}^{n+1}|_{\partial\Omega},\\ \mathbf{t}\cdot\mathbf{u}^*|_{\partial\Omega} &= \mathbf{t}\cdot\mathbf{u}^{n+1}|_{\partial\Omega}+\Delta t\cdot\mathbf{t}\cdot\nabla\phi^n,\\ \nabla\phi\cdot\mathbf{n}|_{\partial\Omega} &= 0,\end{aligned}$$

where **n** and **t** denote the normal and tangent vectors at the boundary, respectively are sufficient to ensure second-order accuracy for the velocity field (see Brown et al (2001); Kim and Moin (1985)).

4.2 Supra-Convergent Poisson Solver

The Poisson solver presented in section 3.2.3 is globally first-order accurate (consistent), even though the discretization at nonuniform mesh points is inconsistent. In fact, the different approximations of the pressure gradients in Losasso (2004) result in consistent schemes, regardless of how the distance between the two adjacent cells involved in the discretization of the pressure gradients is accounted for. In this case, the scheme is therefore locally inconsistent on nonuniform meshes but still leads consistent solutions. This was explained by the fact that first-order perturbations in the location produce a consistent method as demonstrated in Gibou and Fedkiw (2005); Gibou et al (2002). This can be related to the work of Johansen and Colella (1998) who provided a heuristic argument based on potential theory as to why schemes that are only first-order accurate at locally nonuniform grid nodes can be globally second-order accurate (see also the related work by Manteuffel and White (1986) as well as Kreiss et al (1986)). One of the basic reason is that the set of locally non-uniform cells is one-dimension lower than the set of locally uniform cells (see figure 3). Here, we say that a cell is locally nonuniform if its size is different from the size of at least one of its neighbors whereas a cell is locally uniform if its size is equal to that of all of its neighbors. In turn, the influence of nonuniform cells is absorbed by that of the uniform one through the inversion of the elliptic solver, yielding a second-order accurate scheme. Based on this argument and on numerical evidence, Min *et al.* hypothesized in Min et al (2006b) that a strategy for deriving p^{th} order accurate finite difference schemes in the L^∞ norm, is to focus on designing schemes that are $(p-1)^{th}$ order accurate at locally nonuniform cells, which reduce to at least p^{th} order accurate schemes at locally uniform cells. In particular, in order to derive second-order accurate schemes, it is enough to focus on finding a consistent approximation at non-uniform cells.

Consider a Cartesian domain $\Omega\in\mathbb{R}^n$ with boundary $\partial\Omega$ and the variable Poisson equation $\nabla\cdot(\rho\nabla u)=f$ on Ω with Dirichlet boundary condition $u_{|\partial\Omega}=g$. We assume that the variable coefficient ρ is bounded from below by a positive constant. In one spatial dimension, standard central

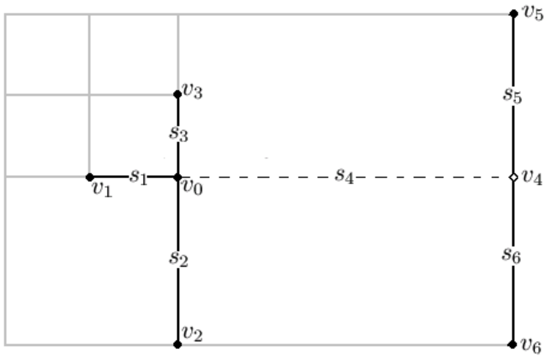

Figure 4: Local structure around a node v_0 in a quadtree mesh: At most one node in the two Cartesian directions might not exist. In this case, we define a ghost node (here v_4) to be used in the discretizations.

differencing formulaes read:

$$\left(\frac{u_{i-1}-u_i}{s_{i-\frac{1}{2}}}\cdot\frac{\rho_{i-1}+\rho_i}{2}+\frac{u_{i+1}-u_i}{s_{i+\frac{1}{2}}}\cdot\frac{\rho_{i+1}+\rho_i}{2}\right)\cdot\frac{2}{s_{i-\frac{1}{2}}+s_{i+\frac{1}{2}}}=f_i,$$

where $s_{i-1/2}$ is the distance between nodes $i-1$ and i. This discretization is second-order accurate and can be applied in a dimension by dimension framework. However, special care needs to be taken when vertices are no longer aligned (see, e.g. figure 4). In this case, Min et al (2006b) proposed to use the truncation error in linear interpolation in the transverse direction as part of the stencil for the derivative in the other direction, leading to a more compact stencil, and an M-matrix. For example, referring to figure 4 the discretizations for $(\rho u_x)_x$ and $(\rho u_y)_y$ along with their Taylor analysis are given respectively by

$$\left(\frac{u_1-u_0}{s_1}\cdot\frac{\rho_1+\rho_0}{2}+\frac{s_6D_5+s_5D_6}{s_5+s_6}\right)\cdot\frac{2}{s_1+s_4}=(\rho u_x)_x+\frac{s_5s_6}{(s_1+s_4)s_4}(\rho u_y)_y+O(h),\quad (16)$$

and

$$\left(\frac{u_2-u_0}{s_2}\cdot\frac{\rho_2+\rho_0}{2}+\frac{u_3-u_0}{s_3}\cdot\frac{\rho_3+\rho_0}{2}\right)\cdot\frac{2}{s_2+s_3}=(\rho u_y)_y+O(h),\quad (17)$$

with

$$\begin{aligned} D_5 &= \frac{u_5-u_0}{s_4}\cdot\frac{\rho_5+\rho_0}{2}, \\ D_6 &= \frac{u_6-u_0}{s_4}\cdot\frac{\rho_6+\rho_0}{2}. \end{aligned}$$

The spurious term $\frac{s_5s_6}{(s_1+s_4)s_4}(\rho u_y)_y$ is cancelled by weighting appropriately equations (16) and (17) as

$$\left(\frac{u_1-u_0}{s_1}\cdot\frac{\rho_1+\rho_0}{2}+\frac{s_6a_5+s_5a_6}{s_5+s_6}\right)\cdot\frac{2}{s_1+s_4}+\left(\frac{u_2-u_0}{s_2}\cdot\frac{\rho_2+\rho_0}{2}+\frac{u_3-u_0}{s_3}\cdot\frac{\rho_3+\rho_0}{2}\right)\cdot\frac{2}{s_2+s_3}\cdot\left(1-\frac{s_5s_6}{(s_1+s_4)s_4}\right)=f_0+O(h).$$

The discretization obtained is now first-order accurate at locally nonuniform points and second-order accurate at locally uniform points, hence yields a globally second-order accurate scheme in the maximum norm.

5 Numerical Results

We report numerical evidences that confirm the schemes described in section 4 yield second-order accuracy in the L^1 and L^∞ norms, on highly irregular grids. In particular the difference of level between cells can be greater that one, illustrating that the method preserves its second-order accuracy on *non-graded* adaptive meshes. In the case of the Poisson scheme derived in section 4.2, we demonstrate that both the solution *and* its gradients are second-order accurate. The linear systems of equations are solved using a bi-conjugate gradient method with an incomplete Cholesky preconditioner.

5.1 Accuracy for the Supra-Convergent Poisson Solver

Consider a domain $\Omega=[0,2]\times[0,1]$ and a grid depicted in figure 5 and $\Delta u=f$ with an exact solution of $u(x,y)=\sin(x)\cos(y)$. Dirichlet boundary conditions are imposed on the boundary. Tables 1 and 2 demonstrate second-order accuracy in the L^1 and L^∞ norms for the solution and its gradient, respectively.

Table 1: Convergence rate of u for example 5.1.

Finest Resolution	L^∞ error on u	rate	$L^1 error$ on u	rate
64^2	2.73×10^{-3}		3.75×10^{-2}	
128^2	6.99×10^{-4}	1.91	4.18×10^{-4}	2.33
256^2	1.76×10^{-4}	1.97	2.75×10^{-5}	1.89
512^2	4.38×10^{-5}	1.98	6.87×10^{-6}	2.05
1024^2	1.09×10^{-5}	2.00	1.71×10^{-6}	2.00

Table 2: Convergence rate of ∇u for example 5.1.

Finest Resolution	L^∞ error on ∇u	rate	$L^1 error$ on ∇u	rate
64^2	2.28×10^{-1}		5.04×10^{-2}	
128^2	7.63×10^{-2}	1.58	1.07×10^{-2}	1.24
256^2	2.05×10^{-2}	1.89	2.71×10^{-3}	1.99
512^2	5.21×10^{-3}	1.98	6.06×10^{-4}	2.16
1024^2	1.31×10^{-3}	1.99	1.46×10^{-4}	2.04

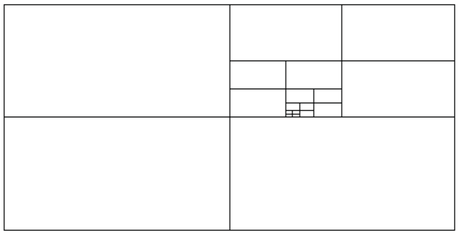

Figure 5: Original mesh used in example 5.1 illustrating high size ratios between adjacent cells.

5.2 Accuracy for the Navier-Stokes Equation

5.2.1 Unconditional Stability

Unconditional schemes are not bound to respect the CFL condition $\Delta t < \Delta x_s$, which can lead to very severe time step restriction. Here, we demonstrate that our solver allows unconstrained time steps. Consider a domain $\Omega = [-\pi/2, \pi/2] \times [-\pi/2, \pi/2]$ and a grid depicted in figure 6. We consider the Navier-Stokes equations with an exact solution of:

$$
\begin{aligned}
&u(x,y,t) = -\cos(t) * \cos(x) * \sin(y),\\
&v(x,y,t) = \cos(t) * \sin(x) * \cos(y),\\
&p(x,y,t) = \sin(x) * \sin(y) * (\sin(t) - 2 * \cos(t)).
\end{aligned}
$$

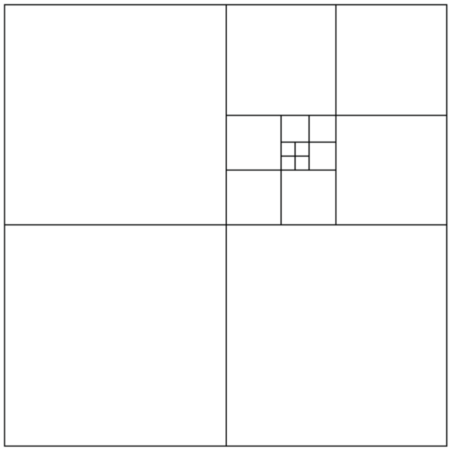

Figure 6: Original mesh used in example 5.2.

The viscosity is set to $\mu = 1$ and Dirichlet boundary conditions are imposed on the boundary. Tables 3 and 4 demonstrate second-order accuracy in the L^1 and L^∞ norms for the solution when the time step is given by $\Delta t = \Delta x_s$ and $\Delta t = 3\Delta x_s$, respectively, where Δx_s refers to the size of the most refined grid cell.

Table 3: Convergence rate of the x-component u of the velocity field $\mathbf{u}$ for example 5.2.1 in the case of a time step $\Delta t = \Delta x_s$.

Finest Resolution	L^∞ error on u	rate	L^1 *error* on u	rate
64^2	1.49×10^{-1}		2.71×10^{-2}	
128^2	1.91×10^{-2}	2.96	3.57×10^{-3}	2.92
256^2	3.86×10^{-3}	2.31	7.58×10^{-4}	2.23
512^2	2.81×10^{-4}	3.78	7.92×10^{-5}	3.25

Table 4: Convergence rate of the x-component u of the velocity field $\mathbf{u}$ for example 5.2.1 in the case of a time step $\Delta t = 3\Delta x_s$.

Finest Resolution	L^∞ error on u	rate	L^1 *error* on u	rate
64^2	7.10×10^{-2}		1.89×10^{-2}	
128^2	1.30×10^{-2}	2.44	2.33×10^{-3}	3.02
256^2	4.98×10^{-3}	1.38	6.72×10^{-4}	1.79
512^2	1.29×10^{-3}	1.98	1.94×10^{-4}	2.26

Table 5: Accuracy of the velocity field in the L^1 and L^∞ norms for example 5.2.3.

Size of the Finest Grid	$\|\|U - U_h\|\|_\infty$	Order	$\|\|U - U_h\|\|_1$	order
32^2	$6.92E-2$		$2.60E-2$	
64^2	$2.64E-2$	1.38	$9.73E-3$	1.41
128^2	$6.28E-3$	2.07	$2.49E-3$	1.96
256^2	$1.07E-3$	2.54	$4.98E-4$	2.32
512^2	$2.23E-4$	2.26	$3.94E-5$	2.90

Table 6: Accuracy of the divergence free condition in the L^1 and L^∞ norms for example 5.2.3.

Size of the Finest Grid	$\|\|\nabla \cdot U_h\|\|_\infty$	Order	$\|\|\nabla \cdot U_h\|\|_1$	order
32^2	$3.56E-1$		$4.82E-2$	
64^2	$1.36E-1$	1.38	$1.59E-2$	1.60
128^2	$3.74E-2$	1.87	$2.30E-3$	2.78
256^2	$9.55E-3$	1.96	$2.94E-4$	2.96
512^2	$2.56E-3$	1.90	$3.94E-5$	2.90

5.2.2 Lid-Driven Cavity

We test our Navier-Stokes solver on the well-known lid-driven cavity problem studied extensively by Ghia (1982): Consider a domain $\Omega = [0,1]^2$, with the top wall moving with unit velocity. We impose no-slip boundary conditions on the four walls. In this example, we take a Reynolds number $Re = 1000$, i.e. a viscosity coefficient $\mu = 1/1000$. For the refinement criteria, there exist various choices including a posteriori error control based on Richardson extrapolation Berger and Colella (1989), or a simple criteria based on the magnitude of local vorticity Popinet (2003); Min et al (2006a). The criterion for mesh refinement we use is that proposed in Blom et al

(1996), i.e. a cell C is refined whenever

$$\min(\triangle x, \triangle y)^2 \times \max_{x \in C} \left(|u_{xx}|, |u_{yy}|, |v_{xx}|, |v_{yy}|\right) > \tau, \tag{18}$$

where τ is an empirically chosen threshold taken to be .01. More precisely, consider a grid structure G^n at time t^n on which the velocity field is updated from $\mathbf{u}^n$ to $\mathbf{u}^{n+1}$. The grid G^{n+1} at t^{n+1} is constructed in the following way: First, we compute the second-order derivatives at every nodes of G^n. Second, starting from the root of G^{n+1} split the cell if (18) is satisfied. Finally, $\mathbf{u}^{n+1}$ is defined on the new grid G^{n+1} from the values of $\mathbf{u}^{n+1}$ on G^n using the quadratic interpolation.

Figure 7 depicts the evolution of the streamlines and the evolution of the adaptive grid until steady state, while figure 8 demonstrates the convergence of the velocity at steady state to the benchmark solution of Ghia (1982). We note that these simulation results are comparable to the results of Min et al (2006a) that utilized a different refinement criteria, a different CFL number and a different level difference between coarsest and finest cells.

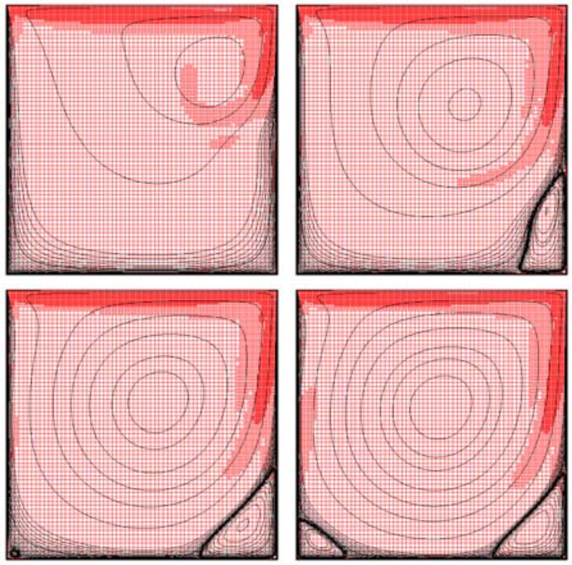

Figure 7: Adaptive grids and streamlines for the driven cavity example 5.2.2. From top to bottom and left to right: $t = 3.12$, 7.50, 13.75 and 37.50. The coarsest grid has level 6, and the finest has level 8.

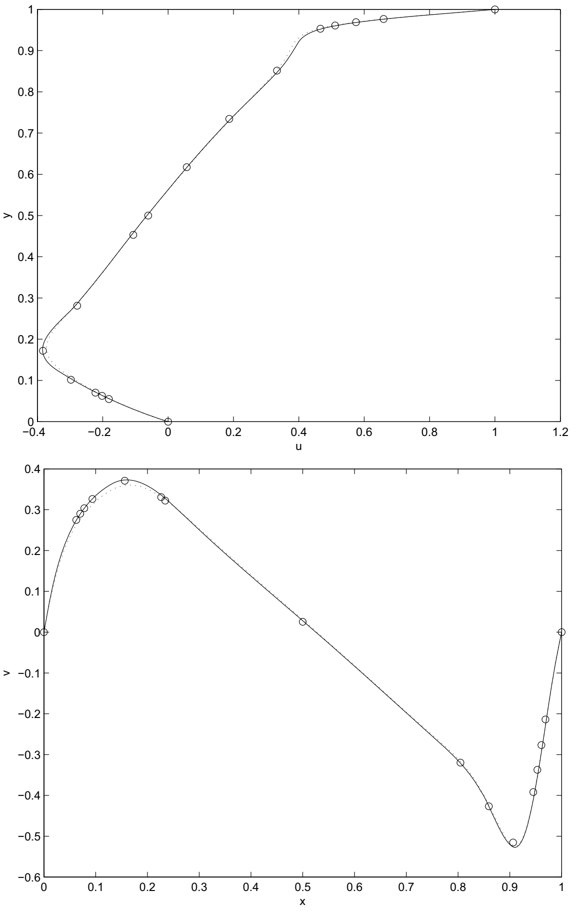

Figure 8: x- and y- components of the velocity field in the driven cavity example 5.2.2. The domain is $[0,1]$, $Re = 1000$ and the time step is $\triangle t = 5\triangle x_s$, where $\triangle x_s$ is the size of the smallest grid cell. The symbols are the experimental results of Ghia (1982), the dotted line depicts the numerical results obtained with an adaptive quadtree with levels ranging from 6 to 8, whereas the solid line depicts the numerical results obtained with an adaptive quadtree with levels ranging from 7 to 9.

5.2.3 Three Spatial Dimensions

In three spatial dimensions, we consider a domain $\Omega = [-\frac{\pi}{2}, \frac{\pi}{2}]^3$ and a flow with viscosity $\mu = 1$ and

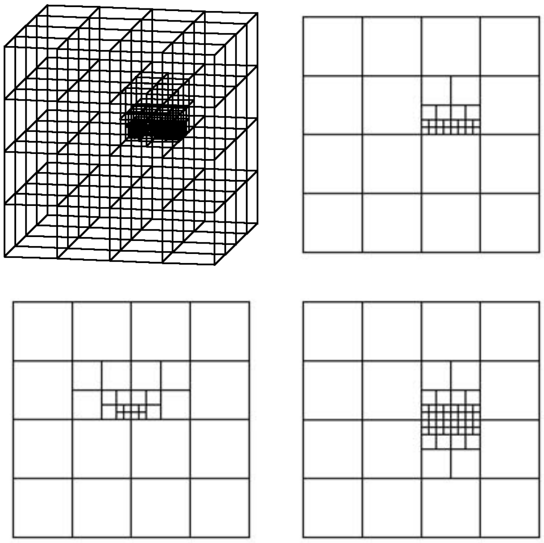

Figure 9: From top to bottom and from left to right: Arbitrarily generated three dimensional grid used in example 5.2.3, its front view, side view and top view. In particular, note that the difference of level between adjacent grid cells can exceed one.

with an exact solution defined by:

$$u(x,y,z,t) = -2\cos(t)\cos(x)\sin(y)\sin(z)$$
$$v(x,y,z,t) = \cos(t)\sin(x)\cos(y)\sin(z)$$
$$w(x,y,z,t) = \cos(t)\sin(x)\sin(y)\cos(z)$$
$$p(x,y,z,t) = \frac{1}{4}\cos^2(t)\Big(2\cos(2x)+\cos(2y) + \cos(2z)\Big)$$

The time step is chosen as $\Delta t = 5 \times \Delta x_s$, where Δx_s is the size of the finest grid cell and we run the simulation up to a final time of $t = \pi$. 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.

6 Conclusion

We have described two finite difference schemes for simulating incompressible flows on nonuniform meshes using quadtree/octree data structures. The first one uses a cell-centered Poisson solver that yields first-order accurate solutions, while producing symmetric linear systems (see Losasso (2004)). The second uses a node-based Poisson solver that produces second-order accurate solutions and second-order accurate gradients, while producing nonsymmetric linear systems (see Min et al (2006b) for a supra-convergent Poisson solver and Min et al (2006a) for a second-order accurate Navier-Stokes solver). The grids considered can be non-graded, i.e. the difference of level between two adjacent cells can be arbitrary, which facilitates grid generations. In both cases semi-Lagrangian methods were used to update the intermediate fluid velocity in a standard projection framework. Numerical results were reported in two and three spatial dimensions to demonstrate the accuracy of the methods.

Acknowledgement: The research of F. Gibou was supported in part by the Alfred P. Sloan Foundation through a research fellowship in Mathematics. The research of H. D. Ceniceros was supported in part by NSF DMS-0311911.

References

Almgren, A.; Bell, J.; Colella, P.; Howell, L.; and Welcome, M. (1998) A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, Vol. 142, pp.1–46.

Anderson, A.; Zheng, X.; and Cristini, V. (2005) Adaptive unstructured volume remeshing i: The method. *J. Comput. Phys.*, Vol. 208, pp.616–625.

Berger, M. and Colella, P. (1989) Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, Vol. 82, pp.64–84.

Berger, M. and Oliger, J. (1984) Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, Vol. 53, pp.484–512.

Blom, J.G.;Trompert, R.A.; and Verwer, J.G. (1996) A vectorizable adaptive grid solver for pdes in 2d. *ACM Trans. Math. Softw.*, Vol. 22, pp.302–328.

Brown, D.; Cortez, R.; and Minion, M. (2001) Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, Vol. 168, pp.464–499.

Ceniceros, H.D.; and Roma, A.M. (2004) Study of long-time dynamics of a viscous vortex sheet with a fully adaptive non-stiff method. *Phys. Fluids*, Vol. 16, No. 12, pp.4285–4318.

Chorin, A. (1967) A Numerical Method for Solving Incompressible Viscous Flow Problems. *J. Comput. Phys.*, Vol. 2, pp.12–26.

Cristini, V.; and Renardy, Y. (2006) Scalings for droplet sizes in shear-driven breakup: Non-microfluidic ways to monodisperse emulsions. *FDMP: Fluid Dynamics & Materials Processing*, Vol. 2, pp.77–94.

Ghia, U.; Ghia, K.N.; and Shin, C.T. (1982) High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *J. Comput. Phys.*, Vol. 48, pp.387–411.

Gibou, F.; and Fedkiw, R. (2005) A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem. *J, Comput. Phys.*, Vol. 202, pp.577–601.

Gibou,F.; Fedkiw, R.; Cheng, L.-T.; and Kang, M. (2002) A second–order–accurate symmetric discretization of the Poisson equation on irregular domains. *J. Comput. Phys.*, Vol. 176, pp.205–227.

Ham, F.; Lien, F.; and Strong, A. (2002) A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids. *J. Comput. Phys.*, Vol. 117, pp.117–133.

Harlow, F.; and Welch, J. (1965) Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Phys. Fluids*, Vol. 8, pp.2182–2189.

Harten, A.; Enquist, B.; Osher, S.; and Chakravarthy, S. (1987) Uniformly high-order accurate essentially non-oscillatory schemes III. *J. Comput. Phys.*, Vol. 71, pp.231–303.

Johansen, H.; and Colella, P. (1998) A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.*, Vol. 147, pp.60–85.

Kim, J.; and Moin, P. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comput. Phys.*, Vol. 59, pp.308–323.

Kreiss, H.O.; Manteuffel, H.-O.; Schwartz, T.A.; Wendroff, B.; and White Jr., A.B. Supra-convergent schemes on irregular grids. *Math. Comp.*, Vol. 47, pp.537–554.

Lipnikov, K.; Morel, J.; and Shashkov, M. (2004) Mimetic finite difference methods for diffusion equations on non-orthogonal non-conformal meshes. *J. Comput. Phys.*, Vol. 199, pp.589–597.

Liu, X.-D.;Osher, S.; and Chan, T. (1996) Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, Vol. 126, pp.202–212.

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

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

Manteuffel, T.; and White, A. (1986) The numerical solution of second-order boundary value problems on nonuniform meshes. *Math. Comput.*, Vol. 47, No. 176, pp.511–535.

Min, C.; and Gibou, F. (2006a) A second order accurate projection method for the incompressible Navier-Stokes equation on non-graded adaptive grids. *J. Comput. Phys.*, Vol. 219, pp.912–929.

Min, C.; Gibou, F.; and Ceniceros, H. (2006b) A supra-convergent finite difference scheme for the variable coefficient Poisson equation on non-graded grids. *J. Comput. Phys.*, Vol. 218, pp.123–140.

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

Samet, H. (1989) *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, New York.

Shu, C.-W.; and Osher, S. (1989) Efficient implementation of essentially non-oscillatory shock capturing schemes II (two). *J. Comput. Phys.*, Vol. 83, pp.32–78.

Staniforth, A.; and Cote, J. (1991) Semi-Lagrangian Integration Schemes for Atmospheric Models: A Review. *Monthly Weather Review*, Vol. 119, pp.2206–2223.

Strain, J. (1999) Fast tree-based redistancing for level set computations. *J. Comput. Phys.*, Vol. 152, pp.664–686.

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

Cristini, V.; Blawzdziewicz, J.; and Lowenberg, M. (2001) An adaptive mesh algorithm for evolving surfaces: simulations of drop breakup and coalescence. *J. Comput. Phys.*, Vol. 168, pp.445–463.

Yang, X.; James, A.; Lowengrub, J.; Zheng, X.; and Cristini, V. (2006) An adaptive coupled level-set/volume-of-fluid interface capturing method for unstructured triangular grids. *J. Comput. Phys.*, Vol. 217, pp.364–394.

Zheng, X.; Lowengrub, J.; Anderson, A.; and Cristini, V. (2005) Adaptive unstructured volume remeshing ii: Application to two- and three-dimensional level-set simulations of multiphase flow. *J. Comput. Phys.*, Vol. 208, pp.626–650.