A Cartesian Grid Projection Method for the Incompressible Euler Equations in Complex Geometries

Ann S. Almgren *

John B. Bell *

Center for Computational Sciences and Engineering

Lawrence Livermore National Laboratory

Livermore, CA 94550

Phillip Colella †

Tyler Marthaler†

Mechanical Engineering Dept.

University of California at Berkeley

Berkeley, CA 94720

*This work of these authors was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. Support under contract No. W-7405-Eng-48 was provided by the Applied Mathematical Sciences Program and the High Performance Computing and Communications Program of the DOE Office of Scientific Computing and by the Defense Nuclear Agency under IACRO 93-817 and IACRO 94-831.

†Research supported at UC Berkeley by the US Department of Energy Office of Scientific Computing under grants FDDE-FG03-94-ER25205 and FDDE-FG03-92-ER25140, and by by a National Science Foundation Presidential Young Investigator award under grant ACS-8958522.

ABSTRACT

Many problems in fluid dynamics require the representation of complicated internal or external boundaries of the flow. Here we present a method for calculating time-dependent incompressible inviscid flow which combines a projection method with a "Cartesian grid" approach for representing geometry. In this approach, the body is represented as an interface embedded in a regular Cartesian mesh. The advection step is based on a Cartesian grid algorithm for compressible flow, in which the discretization of the body near the flow uses a volume-of-fluid representation. A redistribution procedure is used to eliminate time-step restrictions due to small cells where the boundary intersects the mesh. The projection step uses an approximate projection based on a Cartesian grid method for potential flow. The method incorporates knowledge of the body through volume and area fractions along with certain other integrals over the mixed cells. Convergence results are given for the projection itself and for the time-dependent algorithm in two dimensions. The method is also demonstrated on flow past a half-cylinder with vortex shedding.

1 Introduction

In this paper, we present a numerical method for solving the unsteady incompressible Euler equations in domains with irregular boundaries. The underlying discretization method is a projection method [22, 5]. Discretizations of the nonlinear convective terms and lagged pressure gradient are first used to construct an approximate update to the velocity field; the divergence constraint is subsequently imposed to define the velocity and pressure at the new time. The irregular boundary is represented using the Cartesian mesh approach [58], i.e. by intersecting the boundary with a uniform Cartesian grid, with irregular cells appearing only adjacent to the boundary. The extension of the basic projection methodology to the Cartesian grid setting exploits the separation of hyperbolic and elliptic terms of the method in [5] to allow us to use previous work on discretization of hyperbolic and elliptic PDE's on Cartesian grids. The treatment of the hyperbolic terms is based on algorithms developed for gas dynamics, and closely follows the algorithm of Pember *et al.* [56, 57]. The Cartesian grid projection uses the techniques developed by Young et al [72] for full potential transonic flow to discretize the elliptic equation that is used to enforce the incompressibility constraint. The overall design goals of the method are to be able to use the high-resolution finite difference methods based on higher-order Godunov methods for the advective terms in the presence of irregular boundaries that effectively add a potential flow component to the solution.

Cartesian grid methods were first used by Purvis and Burkhalter [58] for solving the equations of transonic potential flow; see also [71, 40, 62, 72]. Clarke *et al.* [23] extended the methodology to steady compressible flow; see also [30, 29, 28, 52]. Zeeuw and Powell [74], Coirier and Powell [24], Coirier [25], Melton *et al.* [51], and Aftosmis *et al.* [1] have developed adaptive methods for the steady Euler and Navier-Stokes equations.

For time-dependent hyperbolic problems, the primary difficulty in using the Cartesian grid approach lies in the treatment of the cells created by the intersection of the irregular boundary with the uniform mesh. There are no restrictions on how the boundary intersects the Cartesian grid (unlike the "stair step" approach which defines the body as aligned with cell edges), and as a result cells with arbitrarily small volumes can be created. A standard finite-volume approach using conservative differencing requires division by the volume of each cell; this is unstable unless the time step is reduced proportionally to the volume. Although in the projection method convective differencing is used for the hyperbolic terms in the momentum equation, we base our methodology for incompressible flow on the experience gained for compressible flow in the handling of small cells. (In addition, we will wish to update other quantities conservatively.) The major issues, then, in designing such a method are how to maintain accuracy, stability, and conservation in the irregular cells at the fluid-body interface while using a time step restricted by CFL considerations using the regular cell size alone. We refer to this as the "small cell problem."

Noh [54] did early work in this area in which he used both cell merging techniques and redistribution. LeVeque [42, 43] and Berger and LeVeque [12] have developed explicit methods which use the large time step approach developed by LeVeque [41] to overcome the small cell problem. Berger and LeVeque [13, 14] have also studied approaches in which the small cell problem is avoided by the use of a rotated difference scheme in the cells cut by the fluid-body interface. Both the large time step and the rotated difference schemes are globally second-order and better than first-order but less than second-order at the fluid-body interface. Cell merging techniques were also used by Chiang *et al.* [20], Bayyuk *et al.* [4], and Quirk [60, 59] to overcome the small cell problem. Results for this method suggest it is globally second order accurate and first order at the boundary. Two other approaches for unsteady compressible flow are based on flux-vector splitting (Choi and Grossman [21]) and state-vector splitting (Öksüzoğlu [55], Gooch and Öksüzoğlu [31]), but neither of these approaches avoids the small cell problem.

The advection step of the method presented here uses a different approach for handling irregular cells based on ideas previously developed for shock tracking by Chern and Colella

[17] and Bell, Colella, and Welcome [8], and extended by Pember *et al.* [56, 57]. In this approach the boundary is viewed as a "tracked front" in a regular Cartesian grid with the fluid dynamics at the boundary governed by the boundary conditions of a stationary reflecting wall. The basic integration scheme for hyperbolic problems consists of two steps. In the first step, a reference state is computed using fluxes generated by a higher-order Godunov method in which the fluid-body boundary is essentially "ignored". In the second step, a correction is computed to the state in each irregular cell. A stable, but nonconservative, portion of this correction is applied to the irregular cell. Conservation is then maintained by a variation of the algebraic redistribution algorithm in [17] which distributes the remainder of the correction to those regular and irregular cells that are immediate neighbors of the irregular cell. This redistribution procedure allows the scheme to use time steps computed from CFL considerations on the uniform grid alone. We adapt this scheme for the advection step of the projection method.

The projection step requires solution of an elliptic equation on an irregular grid. The finite-element-based projection developed by Almgren *et al.* [3] for a regular grid is extended here to accommodate embedded boundaries using the techniques developed by Young *et al.* [72]. The approach of Young *et al.* defines the approximating space using standard bilinear elements on an enlarged domain that includes nodes of mixed cells lying outside the domain. Quadratures in the weak form are restricted to the actual domain to define the discretization. The resulting linear system can be readily solved with a straightforward extension of multigrid.

A second-order Cartesian grid projection method has been developed recently by Tau [67] for the incompressible Navier-Stokes equations. In this formulation velocities are defined on a staggered grid (which is incompatible with the cell-centered form of the Godunov method), and the tangential velocity must vanish at the boundaries (i.e. the formulation is not extensible to the Euler equations). Comments and results in [67] indicate that the accuracy of the method again is in general first-order at the boundary but globally second-order. No mention is made of the small cell problem for advecting quantities other than momentum.

In addition to the approach taken here, there are two other basic approaches to the treatment of irregular domains: and locally body-fitted grids obtained from mesh mappings, and unstructured grids. For body-fitted structured or block-structured grids for compressible flow, the literature is quite extensive. Typical examples include [37, 35, 16, 11, 27, 7, 65, 68, 9, 61, 73]. The primary objection to mapped grids is the difficulty in generating them for complex geometries, particularly in three dimensions. There have been a number of approaches to making mapped grids more flexible, such as using combinations of of unstructured and structured grids [69, 70, 49, 64, 53, 36, 48], multiblock-structured grids [2, 66, 63] and overlapping composite grids [10, 19, 34]. On the topic of unstructured grids, we refer the reader to [39, 38, 45, 46, 50] for compressible flow. Löhner et al [44] have developed an unstructured finite element algorithm for high-Reynolds number and inviscid incompressible flows.

In comparison with the other approaches discussed above, the advantages of the Cartesian grid approach include not requiring special grid generation techniques to fit arbitrary boundaries, the ability to use a standard time-stepping algorithm at all cells with no additional work other than at cells at or near the boundary, and the ability to incorporate efficient solvers for the elliptic equation. The most serious disadvantage is a loss of accuracy at the boundary (numerical results show a reduction to first-order accuracy at the boundary). The loss of second-order accuracy at the boundary indicates that the Cartesian grid approach is not ideal for calculations designed to resolve viscous boundary layers, but should be satisfactory for flows where the primary effect of the boundary is to introduce a potential flow component to the velocity corresponding to the effect of the no-normal flow condition at the boundary. Problems where this is the case include flows in large containers, such as utility burners and boilers, and atmospheric flows over irregular orography.

In the next section we review the basic fractional step algorithm, and introduce the notation of the Cartesian grid method. The subsequent two sections contain descriptions of the advection step and the projection step, respectively, for flows with embedded boundaries. In the final two sections we present numerical results and conclusions. All results and detailed discussion will be for two spatial dimensions; the extension to three dimensions is straightforward and will be presented in later work.

2 Basic Algorithm

2.1 Overview of Fractional Step Formulation

The incompressible Euler equations written in convective form are

$$U_t + (U \cdot \nabla)U + \nabla p = 0 \tag{2.1}$$

and

$$\nabla \cdot U = 0. \tag{2.2}$$

Alternatively, (2.1) could be written in conservative form as

$$U_t + \nabla \cdot (UU) + \nabla p = 0 \tag{2.3}$$

The projection method is a fractional step scheme for solving these equations, composed of an advection step followed by a projection. In the advection step for cells entirely in the flow domain we solve the discretization of (2.1),

$$\frac{U^* - U^n}{\Delta t} + [(U \cdot \nabla)U]^{n+\frac{1}{2}} + \nabla p^{n-\frac{1}{2}} = 0 \tag{2.4}$$

for the intermediate velocity U^*. For small cells adjoining the body we modify the velocity update using the conservative formulation of the nonlinear terms (see section 3). The pressure gradient at $t^{n-\frac{1}{2}}$ was computed in the previous time step and is treated as a source term in (2.4). The advection terms in (2.4), namely $[(U \cdot \nabla)U]^{n+\frac{1}{2}}$, are approximated at time $t^{n+\frac{1}{2}}$ to second-order in space and time using an explicit predictor-corrector scheme; their construction is described in section 3.

The velocity field U^* is not, in general, divergence-free. The projection step of the algorithm decomposes the result of the first step into a discrete gradient of a scalar potential and an approximately divergence-free vector field. They correspond to the update to the pressure gradient and the update to the velocity, respectively. In particular, if $\mathbf{P}$ represents the projection operator then

$$\frac{U^{n+1} - U^n}{\Delta t} = \mathbf{P}\left(\frac{U^* - U^n}{\Delta t}\right) \tag{2.5}$$

$$\nabla p^{n+\frac{1}{2}} = \nabla p^{n-\frac{1}{2}} + (\mathbf{I} - \mathbf{P})\left(\frac{U^* - U^n}{\Delta t}\right)$$

Note that the pressure gradient is defined at the same time as the time derivative of velocity, and therefore at half-time levels.

2.2 Notation

We first introduce the notation used to describe how the body intersects the computational domain. The volume fraction $\Lambda_{i,j}$ for each cell is defined as the fraction of the computational cell $B_{i,j}$ that is inside the flow domain. The area fractions $a_{i+\frac{1}{2},j}$ and $a_{i,j+\frac{1}{2}}$ specify the fractions of the $(i+\frac{1}{2},j)$ and $(i,j+\frac{1}{2})$ edges, respectively, that lie inside the flow domain. We label a cell entirely within the fluid ($\Lambda = 1$) as a *full cell* or *fluid cell*, a cell entirely outside of the flow domain ($\Lambda = 0$) as a *body cell*, and a cell partially in the fluid ($0 < \Lambda < 1$) as a *mixed cell*. A *small* cell is a mixed cell with a small volume fraction. We note that it is possible for the area fraction of an edge of a fluid cell to be less than one if the fluid cell abuts a body or mixed cell.

In the method presented here, the state of the fluid at time t^n is defined by $U^n_{ij} = (u^n_{ij}, v^n_{ij})$, the velocity field in cell $B_{i,j}$ at time t^n, and $p^{n-\frac{1}{2}}_{i-\frac{1}{2},j-\frac{1}{2}}$, the pressure at node $(i-\frac{1}{2}, j-\frac{1}{2})$ at $t^{n-\frac{1}{2}}$. $Gp^{n-\frac{1}{2}}_{ij}$ is the pressure gradient in cell $B_{i,j}$ at time $t^{n-\frac{1}{2}}$. For the construction of the nonlinear advective terms at $t^{n+\frac{1}{2}}$, velocities are also defined on all edges of full and mixed cells at $t^{n+\frac{1}{2}}$; this process requires values of the velocity and pressure gradients in the cells on either side of an edge at t^n. We must therefore define, at each time step, *extended states* in the body cells adjoining mixed or full cells. We do this in a volume-weighted fashion:

$$U_{i,j}^{ext} = \frac{\sum_{k,\ell \in nbhd(B_{i,j})} \Lambda_{k,\ell} U_{k,\ell}}{\sum_{k,\ell \in nbhd(B_{i,j})} \Lambda_{k,\ell}}$$

The pressure gradient is extended in the same manner. Here $nbhd(B_{i,j})$, defined as the neighborhood of $B_{i,j}$, refers to the eight cells (in two dimensions) that share an edge or corner with $B_{i,j}$.

3 Advection step

3.1 Momentum

The algorithm is a predictor-corrector method, similar to that used in [5], but with some modifications as discussed in [6]. The details of the current version without geometry are given in [3]. For simplicity we will assume that the normal velocity on the embedded boundary is zero; the treatment of a more general Dirichlet boundary condition such as inflow is straightforward.

In the predictor we extrapolate the velocity to the cell edges at $t^{n+\frac{1}{2}}$ using a second-order Taylor series expansion in space and time. The time derivative is replaced using (2.1). For edge $(i+\frac{1}{2}, j)$ this gives

$$U_{i+\frac{1}{2},j}^{n+\frac{1}{2},L} = U_{ij}^n + \left(\frac{\Delta x}{2} - \frac{u_{ij}\Delta t}{2}\right) U_{x,ij}^{n,lim} - \frac{\Delta t}{2}(\widehat{vU_y})_{ij} - \frac{\Delta t}{2} Gp_{ij}^{n-\frac{1}{2}}, \tag{3.1}$$

extrapolated from $B_{i,j}$, and

$$U_{i+\frac{1}{2},j}^{n+\frac{1}{2},R} = U_{i+1,j}^n - \left(\frac{\Delta x}{2} + \frac{u_{i+1,j}\Delta t}{2}\right) U_{x,i+1,j}^{n,lim} - \frac{\Delta t}{2}(\widehat{vU_y})_{i+1,j} - \frac{\Delta t}{2} Gp_{i+1,j}^{n-\frac{1}{2}}, \tag{3.2}$$

extrapolated from $B_{i+1,j}$.

Analogous formulae are used to predict values at each of the other edges of the cell. In evaluating these terms the first-order derivatives normal to the edge (in this case $U_x^{n,lim}$) are evaluated using a monotonicity-limited fourth-order slope approximation [26]. The limiting is done on the components of the velocity individually, with modifications as discussed below in cells near the body.

The transverse derivative terms ($\widehat{vU_y}$ in this case) are evaluated as in [6], by first extrapolating from above and below to construct edge states, using normal derivatives only, and then choosing between these states using the upwinding procedure defined below. In particular, we define

$$\widehat{U}^B_{i,j+\frac{1}{2}} = U^n_{ij} + \left(\frac{\Delta y}{2} - \frac{v_{ij}\Delta t}{2}\right) U^{n,lim}_{y,ij},$$
$$\widehat{U}^T_{i,j+\frac{1}{2}} = U^n_{i,j+1} - \left(\frac{\Delta y}{2} + \frac{v_{i,j+1}\Delta t}{2}\right) U^{n,lim}_{y,i,j+1},$$

where $U_y^{n,lim}$ are limited slopes in the y direction, with similar formulae for the lower edge of $B_{i,j}$. In this upwinding procedure we first define the normal advective velocity on the edge:

$$\widehat{v}^{adv}_{i,j+\frac{1}{2}} = \begin{cases} \widehat{v}^B & \text{if } \widehat{v}^B > 0,\ \ \widehat{v}^B + \widehat{v}^T > 0 \\ 0 & \text{if } \widehat{v}^B \leq 0, \widehat{v}^T \geq 0 \text{ or } \widehat{v}^B + \widehat{v}^T = 0 \\ \widehat{v}^T & \text{if } \widehat{v}^T < 0,\ \ \widehat{v}^B + \widehat{v}^T < 0 \end{cases}$$

(We suppress the $i, j+\frac{1}{2}$ spatial indices on bottom and top states here and in the next equation.) We now upwind $\widehat{U}$ based on $\widehat{v}^{adv}_{i,j+\frac{1}{2}}$:

$$\widehat{U}_{i,j+\frac{1}{2}} = \begin{cases} \widehat{U}^B & \text{if } \widehat{v}^{adv}_{i,j+\frac{1}{2}} > 0 \\ \frac{1}{2}(\widehat{U}^B + \widehat{U}^T) & \text{if } \widehat{v}^{adv}_{i,j+\frac{1}{2}} = 0 \\ \widehat{U}^T & \text{if } \widehat{v}^{adv}_{i,j+\frac{1}{2}} < 0 \end{cases}$$

After constructing $\widehat{U}_{i,j-\frac{1}{2}}$ in a similar manner, we use these upwind values to form the transverse derivative in (3.1):

$$(\widehat{vU_y})_{ij} = \frac{1}{2\Delta y}(\widehat{v}^{adv}_{i,j+\frac{1}{2}} + \widehat{v}^{adv}_{i,j-\frac{1}{2}})(\widehat{U}_{i,j+\frac{1}{2}} - \widehat{U}_{i,j-\frac{1}{2}}).$$

We use a similar upwinding procedure to choose the appropriate states $U_{i+\frac{1}{2},j}$ given the left and right states, $U^{n+\frac{1}{2},L}_{i+\frac{1}{2},j}$ and $U^{n+\frac{1}{2},R}_{i+\frac{1}{2},j}$:

$$U_{i+\frac{1}{2},j} = \begin{cases} U^L & \text{if } U^L > 0 \text{ and } U^L + U^R > 0 \\ \frac{1}{2}(U^L + U^R) & \text{if } U^L \le 0, U^R \ge 0 \text{ or } U^L + U^R = 0 \\ U^R & \text{if } U^R < 0 \text{ and } U^L + U^R < 0 \end{cases}$$

We follow a similar procedure to construct $U_{i-\frac{1}{2},j}$, $U_{i,j+\frac{1}{2}}$, and $U_{i,j-\frac{1}{2}}$.

In general the normal velocities at the edges are not divergence-free. In order to make these velocities divergence-free we apply a MAC projection [6] before the construction of the convective derivatives. The equation

$$D^{MAC} G^{MAC} \phi = D^{MAC} U^{n+\frac{1}{2}}$$

is solved for ϕ in all full or mixed cells, with

$$D^{MAC} U^{n+\frac{1}{2}} = \frac{1}{\Lambda_{i,j}} \left(\frac{a_{i+\frac{1}{2},j} u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j} u^{n+\frac{1}{2}}_{i-\frac{1}{2},j}}{\Delta x} + \frac{a_{i,j+\frac{1}{2}} v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - a_{i,j-\frac{1}{2}} v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}}{\Delta y} \right),$$

$$(G^{MAC}\phi)^x_{i+\frac{1}{2},j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \quad (G^{MAC}\phi)^y_{i,j+\frac{1}{2}} = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}.$$

Note that in regions of full cells the $D^{MAC}G^{MAC}$ stencil is simply a standard five-point cell-centered stencil for the Laplacian. For edges within the body that are needed for the corrector step below, the MAC gradient is linearly extrapolated from edges that lie partially or fully in the fluid. For example, if edge $(i+\frac{1}{2}, j)$ lies completely within the body, but edges $(i-\frac{1}{2}, j)$ and $(i-\frac{3}{2}, j)$ lie at least partially within the fluid, then we define

$$(G^{MAC}\phi)^x_{i+\frac{1}{2},j} = 2\ (G^{MAC}\phi)^x_{i-\frac{1}{2},j} - (G^{MAC}\phi)^x_{i-\frac{3}{2},j}.$$

While it is possible this could lead to instabilities, none have yet been observed in numerical testing. We solve this linear system using standard multigrid methods (see [15]), specifically, V-cycles with Jacobi relaxation.

In the corrector step, we form an approximation to the convective derivatives in (2.1)

$$[(U\cdot\nabla)U]^{n+\frac{1}{2}} = \frac{1}{2\Delta x}(u^{MAC}_{i+\frac{1}{2},j}+u^{MAC}_{i-\frac{1}{2},j})(U^{n+\frac{1}{2}}_{i+\frac{1}{2},j}-U^{n+\frac{1}{2}}_{i-\frac{1}{2},j})+\frac{1}{2\Delta y}(v^{MAC}_{i,j+\frac{1}{2}}+v^{MAC}_{i,j-\frac{1}{2}})(U^{n+\frac{1}{2}}_{i,j+\frac{1}{2}}-U^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}),$$

where

$$u^{MAC}_{i+\frac{1}{2},j} = u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - (G^{MAC}\phi)^x_{i+\frac{1}{2},j}, \quad v^{MAC}_{i,j+\frac{1}{2}} = v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - (G^{MAC}\phi)^y_{i,j+\frac{1}{2}}.$$

The intermediate velocity U^* at time $n+1$ is then defined on all full and mixed cells as

$$U^* = U^n - \Delta t([(U\cdot\nabla)U]^{n+\frac{1}{2}} + Gp^{n-\frac{1}{2}}).$$

The upwind method is an explicit difference scheme and, as such, requires a time-step restriction. When all cells are full cells, a linear, constant-coefficient analysis shows that for stability we require

$$\max_{ij}\left(\frac{|u_{ij}|\Delta t}{\Delta x}, \frac{|v_{ij}|\Delta t}{\Delta y}\right) = \sigma \le 1,$$

where σ is the CFL number. The time-step restriction of the upwind method is used to set the time step for the overall algorithm.

As mentioned earlier, for incompressible flow the two forms of the momentum equation, (2.1) and (2.3), are analytically equivalent. For flows without geometry we have successfully used the convective difference form of the equations, (2.4). By contrast, we could use the conservative form of the equations,

$$\frac{U^* - U^n}{\Delta t} + \nabla\cdot\mathbf{F}^{n+\frac{1}{2}} + \nabla p^{n-\frac{1}{2}} = 0,$$

where $\mathbf{F}^{n+\frac{1}{2}} = (u^{MAC}U^{n+\frac{1}{2}}, v^{MAC}U^{n+\frac{1}{2}})$, and evaluate the conservative update as $\int \mathbf{F}^{n+\frac{1}{2}}\cdot\mathbf{n}\,dA$. In the presence of embedded boundaries, the convective update form of the equations is stable even for very small cells, but this update is calculated as if the cell were a full cell, i.e. it doesn't "see" the body other than through the MAC-projected normal advection velocities (and the modification of the limited slopes as discussed at the end of this section). To compute the conservative update, however, one ignores entirely the portion of the cell not in the fluid, and integrates the flux only along the parts of the edges of the cell that lie within the fluid. Thus the presence of the body is correctly accounted for, but using the conservative form requires that the time step approach zero as the cell volume goes to zero, which is the small-cell time step restriction which we seek to avoid.

A solution in this case is to use a weighted average of the convective and conservative updates, effectively allowing as much momentum to pass into a small cell in a time step as will keep the scheme stable. The momentum that does not pass into the small cell is redistributed to the neighboring cells, to maintain conservation in the advection step. This approach is modeled on the algorithm of [57, 56], and is based on the algebraic redistribution scheme of Chern and Colella [18]. The algorithm is as follows:

(1) First, in all cells construct the *reference state*, $\widetilde{U}^*$, defined as

$$\widetilde{U}^* = U^n - \Delta t[(U \cdot \nabla)U]^{n+\frac{1}{2}},$$

using the advective algorithm described for full cells.

(2) On mixed cells only construct an alternative $\widetilde{\widetilde{U}}^*$ using a conservative approach:

$$\widetilde{\widetilde{U}}^* = U^n - \frac{\Delta t}{\Lambda_{i,j}}\left(\frac{(a_{i+\frac{1}{2},j}F^x_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j}F^x_{i-\frac{1}{2},j})}{\Delta x} + \frac{(a_{i,j+\frac{1}{2}}F^y_{i,j+\frac{1}{2}} - a_{i,j-\frac{1}{2}}F^y_{i,j-\frac{1}{2}})}{\Delta y}\right),$$

where the fluxes are defined as

$$F^x_{i+\frac{1}{2},j} = u^{MAC}_{i+\frac{1}{2},j}U_{i+\frac{1}{2},j}, \qquad F^y_{i,j+\frac{1}{2}} = v^{MAC}_{i,j+\frac{1}{2}}U_{i,j+\frac{1}{2}}.$$

This solution enforces no-flow across the boundary of the body, but is not necessarily stable. Define the difference $\delta M_{i,j} = \Lambda_{i,j}(\widetilde{\widetilde{U}}^*_{i,j} - \widetilde{U}^*_{i,j})$.

(3) The conservative solution can be written $\widetilde{\widetilde{U}}^*_{i,j} = \widetilde{U}^*_{i,j} + \frac{\delta M_{i,j}}{\Lambda_{i,j}}$; however, this solution is not stable for $\Lambda \ll 1$. Define instead $\widehat{U}^*_{i,j} = \widetilde{U}^*_{i,j} + \delta M_{i,j}$ on mixed cells, $\widehat{U}^*_{i,j} = \widetilde{U}^*_{i,j}$ on full cells. This allows the mixed cell state to keep the fraction of δM which will keep the scheme stable given that the time step is set by the full-cell CFL constraint.

(4) Now redistribute the remaining fraction of δM from each mixed cell, the amount $(1 -$

$\Lambda_{i,j})\delta M_{i,j}$, to the fluid and mixed cells among the eight neighbors of $B_{i,j}$ in a volume-weighted fashion. Since we redistribute the extensive rather than intensive quantity (e.g. momentum rather than momentum density), the resulting redistribution has the form

$$\widehat{\widehat{U^*}}_{i,j} = \widehat{U^*}_{i,j} + \sum_{k,\ell \in nbhd(B_{i,j})} \frac{(1-\Lambda_{k,\ell})\delta M_{k,\ell}}{m_{k,\ell}},$$

where

$$m_{k,\ell} = \sum_{n,p \in nbhd(B_{k,\ell})} \Lambda_{n,p}.$$

(5) Subtract the pressure gradient term from the solution for all full and mixed cells, treating it as a source term:

$$U^*_{i,j} = \widehat{\widehat{U^*}}_{i,j} - \Delta t \; (Gp)^{n-\frac{1}{2}}_{i,j}.$$

A second modification to the algorithm for cells at or near the boundary is to the slope calculations; this is based on the principle that no state information from cells entirely within the body should be used in calculating slopes. Despite the use of extended states in the creation of edge states, the slope calculation uses state information only from full and mixed cells. In a mixed or full cell for which the fourth-order stencil would require using an extended state, the slope calculation reduces to the second-order monotonicity-limited formula. If the second-order formula would require an extended state, then the slope is set to zero. This has the effect of adding dissipation to the scheme; if the slopes throughout the domain were all set to zero the method would reduce to first-order.

Two comments about the algorithm are in order here. First, we note that while the use of extended states seems to be a low-order approach to representing the body, numerical results show that it is adequate to maintain first-order accuracy at the boundary. An alternative to extended states is the so-called "thin wall approximation" of [57]; with this approximation extended states are not defined, rather upwinding always chooses the values coming from the fluid side (rather than the body side) of the fluid-body interface. In our numerical testing, however, we have found that the extended states version of the algorithm performs slightly better.

Second, we are unable to provide a proof that the redistribution algorithm is stable, but in extensive numerical testing of the redistribution algorithm for compressible flow (see [57, 32, 47]) no stability problems have been observed, and none have been observed in our incompressible flow calculations. It is clear that the algorithm removes the original small-cell stability problem by replacing division by arbitrarily small volume fractions with division by one in step (3). The redistribution algorithm is such that the amount redistributed into a cell is proportional to the volume of that cell; this feature combined with the observation in our calculations that the values of δM are small relative to the values of U in the mixed cells gives a heuristic argument as to why redistribution would be stable.

3.2 Passive Scalars

The algorithm for advecting momentum extends naturally to linear scalar advection. In the third numerical example shown in this paper, a passive scalar enters the domain at the inflow boundary and is advected around the body. We now briefly describe the scalar advection routine, assuming the scalar s is a conserved quantity, i.e.

$$s_t + \nabla \cdot (Us) = 0. \tag{3.3}$$

First, at each time step extended states are defined for the passive scalar just as for velocity. In the predictor we extrapolate the scalar to the cell edges at $t^{n+\frac{1}{2}}$ using a second-order Taylor series expansion in space and time. For edge $(i+\frac{1}{2}, j)$ this gives

$$s_{i+\frac{1}{2},j}^{n+\frac{1}{2},L} = s_{ij}^n + \left(\frac{\Delta x}{2} - \frac{u_{ij}\Delta t}{2}\right) s_{x,ij}^{n,lim} - \frac{\Delta t}{2} v_{i,j} \widehat{s_y}_{ij} \tag{3.4}$$

extrapolated from $B_{i,j}$, and

$$s_{i+\frac{1}{2},j}^{n+\frac{1}{2},R} = s_{i+1,j}^n - \left(\frac{\Delta x}{2} + \frac{u_{i+1,j}\Delta t}{2}\right) s_{x,i+1,j}^{n,lim} - \frac{\Delta t}{2} v_{i,j} \widehat{s_y}_{i+1,j} \tag{3.5}$$

extrapolated from $B_{i+1,j}$.

Analogous formulae are used to predict values at each of the other edges of the cell. As with velocity, derivatives normal to the edge (in this case $s_x^{n,lim}$) are evaluated using the monotonicity-limited fourth-order slope approximation [26], with modifications due to geometry.

The transverse derivative terms ($v\widehat{s_y}$ in this case) are defined using $\widehat{s_y}_{i,j} = (\widehat{s}^T - \widehat{s}^B)/\Delta y$ where

$$\widehat{s}^T = \begin{cases} s^n_{i,j} & \text{if } v^{MAC}_{i,j+\frac{1}{2}} > 0 \\ \frac{1}{2}(s^n_{i,j} + s^n_{i,j+1}) & \text{if } v^{MAC}_{i,j+\frac{1}{2}} = 0 \\ s^n_{i,j+1} & \text{if } v^{MAC}_{i,j+\frac{1}{2}} < 0 \end{cases}$$

and

$$\widehat{s}^B = \begin{cases} s^n_{i,j-1} & \text{if } v^{MAC}_{i,j-\frac{1}{2}} > 0 \\ \frac{1}{2}(s^n_{i,j-1} + s^n_{i,j}) & \text{if } v^{MAC}_{i,j-\frac{1}{2}} = 0 \\ s^n_{i,j} & \text{if } v^{MAC}_{i,j-\frac{1}{2}} < 0 \end{cases}$$

We use a similar procedure to choose $s^{n+\frac{1}{2}}_{i+\frac{1}{2},j}$ given $s^{n+\frac{1}{2},L}_{i+\frac{1}{2},j}$ and $s^{n+\frac{1}{2},R}_{i+\frac{1}{2},j}$,

$$s^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \begin{cases} s^{n+\frac{1}{2},L}_{i+\frac{1}{2},j} & \text{if } u^{MAC}_{i+\frac{1}{2},j} > 0 \\ \frac{1}{2}(s^{n+\frac{1}{2},L}_{i+\frac{1}{2},j} + s^{n+\frac{1}{2},R}_{i+\frac{1}{2},j}) & \text{if } u^{MAC}_{i+\frac{1}{2},j} = 0 \\ s^{n+\frac{1}{2},R}_{i+\frac{1}{2},j} & \text{if } u^{MAC}_{i+\frac{1}{2},j} < 0 \end{cases}$$

In the corrector step for full cells, we update s conservatively:

$$\widehat{s^*}_{i,j} = s^n_{i,j} - \frac{\Delta t}{\Delta x}(u^{MAC}_{i+\frac{1}{2},j} s^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - u^{MAC}_{i-\frac{1}{2},j} s^{n+\frac{1}{2}}_{i-\frac{1}{2},j}) - \frac{\Delta t}{\Delta y}(v^{MAC}_{i,j+\frac{1}{2}} s^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - v^{MAC}_{i,j-\frac{1}{2}} s^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}).$$

In the corrector step for mixed cells, we follow the three steps below:

(1) Construct the reference state $\widetilde{s}^*$ using the convective formulation:

$$\widetilde{s}^* = s^n - \frac{\Delta t}{2\Delta x}(u^{MAC}_{i+\frac{1}{2},j} + u^{MAC}_{i-\frac{1}{2},j})(s^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - s^{n+\frac{1}{2}}_{i-\frac{1}{2},j}) - \frac{\Delta t}{2\Delta y}(v^{MAC}_{i,j+\frac{1}{2}} + v^{MAC}_{i,j-\frac{1}{2}})(s^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - s^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}).$$

(2) Construct $\widetilde{\widetilde{s}}^*$ using the conservative formulation:

$$\widetilde{\widetilde{s}}^* = s^n - \frac{\Delta t}{\Lambda_{i,j}} \left(\frac{(a_{i+\frac{1}{2},j} u^{MAC}_{i+\frac{1}{2},j} s_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j} u^{MAC}_{i-\frac{1}{2},j} s_{i-\frac{1}{2},j})}{\Delta x} + \frac{(a_{i,j+\frac{1}{2}} u^{MAC}_{i,j+\frac{1}{2}} s_{i,j+\frac{1}{2}} - a_{i,j-\frac{1}{2}} u^{MAC}_{i,j-\frac{1}{2}} s_{i,j-\frac{1}{2}})}{\Delta y} \right).$$

Define the difference $\delta s_{i,j} = \Lambda_{i,j} (\widetilde{\widetilde{s}}^*{}_{i,j} - \widetilde{s}^*{}_{i,j})$.

(3) Construct $\widehat{s}^*_{i,j} = \widetilde{s}^*_{i,j} + \delta s_{i,j}$.

Finally, redistribute the remaining fraction of δs from each mixed cell, the amount $(1 - \Lambda_{i,j}) \delta s_{i,j}$, to the full and mixed cells among the eight neighbors of $B_{i,j}$ in a volume-weighted fashion:

$$s^{n+1}_{i,j} = \widehat{s}^*{}_{i,j} + \sum_{k,\ell \in nbhd(B_{i,j})} \frac{(1 - \Lambda_{k,\ell}) \delta s_{k,\ell}}{m_{k,\ell}}.$$

In full cells not adjacent to any mixed cells, set $s^{n+1}_{i,j} = \widehat{s}^*{}_{i,j}$.

4 Projection Step

Since U^* is defined on cell centers, the projection used to enforce incompressibility at time t^{n+1} must include a divergence operator that acts on cell-centered quantities, unlike the MAC projection. The projection we use here is approximate; i.e. $\mathbf{P^2} \neq \mathbf{P}$. The operator, as well as the motivation for using an approximate rather than exact projection, is described in detail in [3]. Here we outline the algorithm for the case of no-flow physical boundaries with no geometry, and for the case of embedded boundaries. The basic approach for the embedded boundaries utilizes the same discretization as was used by Young et al [72] for full potential transonic flow. The necessary modifications for inflow-outflow conditions are described in [3] and their use is demonstrated in the numerical examples.

This projection is based on a finite element formulation. In particular, we consider the scalar pressure field to be a C^0 function that is bilinear over each cell; i.e., the pressure is in

$$S^h = M^1_0(x) \otimes M^1_0(y)$$

where $M_s^t(x)$ is the space of polynomials of degree t in the x direction on each cell with C^s continuity at x-edges. For the velocity space we define

$$\mathbf{V}^h = \mathbf{V}^{h,x} \ \times \ \mathbf{V}^{h,y} \tag{4.1}$$

where $\mathbf{V}^{h,x} = M_{-1}^0(x) \otimes M_{-1}^1(y)$ and $\mathbf{V}^{h,y} = M_{-1}^1(x) \otimes M_{-1}^0(y)$, i.e., u is piecewise constant in x and a discontinuous linear function of y in each cell, with a similar form for v. For mixed cells, we think of the representations as only being defined on the portion of each cell within the flow domain, Ω, although the spaces implicitly define an extension of the solution over the entire cell. Note that in the integral used to define the projection (see (4.3) below) the domain of integration is limited to the actual flow domain; we do not integrate over the portion of mixed cells outside the domain.

For use in the predictor and corrector, the velocity and pressure gradient are considered to be average values over each cell. The vector space, $\mathbf{V}^h$, contains additional functions that represent the linear variation within each cell. These additional degrees of freedom make $\mathbf{V}^h$ large enough to contain $\nabla\phi$ for $\phi \in S^h$. We establish a correspondence between these two representations by introducing an orthogonal decomposition of $\mathbf{V}^h$. In particular, for each $V \in \mathbf{V}^h$ we define a piecewise constant component $\overline{V}$ and the variation $V^\perp = V - \overline{V}$ so that for each cell $B_{i,j}$, $\int_{B_{i,j}\cap\Omega} V^\perp d\mathbf{x} = 0$. By construction these two components are orthogonal in L^2 so they can be used to define a decomposition of $\mathbf{V}^h$ into two subspaces

$$\mathbf{V}^h = \overline{\mathbf{V}}^h \oplus \mathbf{V}^{h^\perp}$$

where $\overline{\mathbf{V}}^h$ and $\mathbf{V}^{h^\perp}$ represent the cell averages and the orthogonal linear variation, respectively. The decomposition of $\mathbf{V}^h$ induces a decomposition of $\nabla\phi$ for all $\phi \in S^h$; namely,

$$(\nabla\phi)_{ij} = (\overline{\nabla\phi})_{ij} + (\nabla\phi)_{ij}^\perp.$$

We now define a weak form of the projection on $\mathbf{V}^h$, based on a weak divergence on $\mathbf{V}^h$. In particular, we define a vector field V^d in $\mathbf{V}^h$ to be divergence-free in the domain Ω if

$$\int_\Omega V^d \cdot \nabla\phi \, d\mathbf{x} = 0 \qquad \forall \phi \in S^h. \tag{4.2}$$

Using the definition (4.2) we can then project any vector field V into a gradient $\nabla\phi$ and weakly divergence-free field V^d (with vanishing normal velocities on boundaries) by solving

$$\int_\Omega \nabla\phi(\mathbf{x}) \cdot \nabla\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})\, d\mathbf{x} = \int_\Omega V \cdot \nabla\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})\, d\mathbf{x} \qquad \forall\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x}) \tag{4.3}$$

for $\phi(\mathbf{x}) = \sum \phi_{i+\frac{1}{2},j+\frac{1}{2}}\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})$, and setting $V^d = V - \nabla\phi$. Here we define the ψ's to be the standard basis functions for S^h, namely $\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})$ is the piecewise bilinear function having node values $\psi_{i+1/2,j+1/2}(\mathbf{x}_{k+1/2,\ell+1/2}) = \delta_{ik}\delta_{j\ell}$.

For the purposes of the fractional step scheme we wish to decompose the vector field

$$V = \overline{V} = \frac{U^* - U^n}{\Delta t}$$

into its approximately divergence free-part

$$\overline{V^d} = \frac{U^{n+1} - U^n}{\Delta t}$$

and the update to the pressure,

$$\phi = \delta p^n \equiv p^{n+\frac{1}{2}} - p^{n-\frac{1}{2}}.$$

Since the finite-difference advection scheme is designed to handle cell-based quantities that are considered to be average values over each cell, the quantity $(G\phi)^\perp$ is discarded at the end of the projection step. This makes the projection approximate, i.e. if $D(V - G\phi) = 0$ then $D(V - (\overline{G\phi})) = O(h^2)$. So, in practice, we solve the system defined by (4.3) (again using standard multigrid techniques with V-cycles and Jacobi relaxation), and set

$$\overline{V}^d_{ij} = \overline{V}_{ij} - \frac{1}{\Lambda_{ij}} \int_{B_{i,j}\cap\Omega} (G\phi)_{ij}\, dx\, dy.$$

as the approximation to $\frac{\overline{U}^{n+1} - \overline{U}^n}{\Delta t}$ in (2.5). (See [3] for more details.)

The left-hand-side of equation (4.3) is, in discrete form, a nine-point stencil approximating the Laplacian of ϕ and the right hand side, for $V = \overline{V}$, is a standard four-point divergence stencil. Without embedded boundaries, the method reduces to constant coefficient difference stencils for divergence, gradient, and the Laplacian operator. These stencils are, for $V = (V^x, V^y)$,

$$
\begin{aligned}
(D\overline{V})_{i+\frac{1}{2},j+\frac{1}{2}} = \quad & (V^x_{i+1,j} + V^x_{i+1,j+1} - V^x_{i,j} - V^x_{i,j+1})/(2\Delta x) + \\
& (V^y_{i,j+1} + V^y_{i+1,j+1} - V^y_{i,j} - V^y_{i+1,j})/(2\Delta y),
\end{aligned}
$$

$$
\begin{aligned}
(\overline{G\phi})_{i,j} = (&(\phi_{i+\frac{1}{2},j-\frac{1}{2}} + \phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}})/(2\Delta x), \\
&(\phi_{i-\frac{1}{2},j+\frac{1}{2}} + \phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}})/(2\Delta y)),
\end{aligned}
$$

and, letting $\Delta x = \Delta y$,

$$
\begin{aligned}
(DG\phi)_{i+\frac{1}{2},j+\frac{1}{2}} = (\ & \phi_{i+\frac{3}{2},j+\frac{3}{2}} + \phi_{i+\frac{3}{2},j+\frac{1}{2}} + \phi_{i+\frac{3}{2},j-\frac{1}{2}} + \phi_{i+\frac{1}{2},j+\frac{3}{2}} + \\
& \phi_{i+\frac{1}{2},j-\frac{1}{2}} + \phi_{i-\frac{1}{2},j+\frac{3}{2}} + \phi_{i-\frac{1}{2},j+\frac{1}{2}} + \phi_{i-\frac{1}{2},j-\frac{1}{2}} - 8\phi_{i+\frac{1}{2},j+\frac{1}{2}})/(3\Delta x^2)
\end{aligned}
$$

Body cells contribute nothing to the integrals in (4.3); in mixed cells the integrals are computed only over the portion of each cell that lies in the fluid. This calculation can be optimized by pre-computing the following integrals in each mixed cell; these integrands result from the products of the gradients of the basis functions in (4.3). We define

$$
\begin{aligned}
I^a &= \int_{B_{i,j}\cap\Omega} x_{ij}\ dx\ dy, \\
I^b &= \int_{B_{i,j}\cap\Omega} y_{ij}\ dx\ dy, \\
I^c &= \int_{B_{i,j}\cap\Omega} (x_{ij}^2 + y_{ij}^2)\ dx\ dy
\end{aligned}
$$

where x_{ij} and y_{ij} are functions defined on each cell such that $x_{ij} = y_{ij} = 0$ at the center of $B_{i,j}$, $x_{ij} = \pm 1/2$ at the left and right edges of $B_{i,j}$, and $y_{ij} = \pm 1/2$ at the top and bottom edges of $B_{i,j}$.

Linear combinations of these integrals form the coefficients used in the divergence, gradient and Laplacian operators. For example, the divergence of a barred vector becomes

$$
(D\overline{V})_{i+\frac{1}{2},j+\frac{1}{2}} = \quad (\ (\tfrac{1}{2}\Lambda_{i+1,j+1} - I^b_{i+1,j+1})V^x_{i+1,j+1} + (\tfrac{1}{2}\Lambda_{i+1,j} + I^b_{i+1,j})V^x_{i+1,j} +
$$

$$
\begin{aligned}
&(\tfrac{1}{2}\Lambda_{i,j+1} - I^b_{i,j+1})V^x_{i,j+1} + (\tfrac{1}{2}\Lambda_{i,j} + I^b_{i,j})V^x_{i,j})/\Delta x + \\
(\;&(\tfrac{1}{2}\Lambda_{i+1,j+1} - I^a_{i+1,j+1})V^y_{i+1,j+1} + (\tfrac{1}{2}\Lambda_{i+1,j} + I^a_{i,j+1})V^y_{i,j+1} + \\
&(\tfrac{1}{2}\Lambda_{i,j+1} - I^a_{i+1,j})V^y_{i+1,j} + (\tfrac{1}{2}\Lambda_{i,j} + I^a_{i,j})V^y_{i,j})/\Delta y
\end{aligned}
$$

If we let $(\overline{G\phi})_{full}$ be $(\overline{G\phi})$ as it would be defined if the cell were a full cell, then in a mixed cell the average of the gradient over the fluid portion of the cell is

$$
(\overline{G\phi})_{i,j} = ((\overline{G\phi})^x_{full} + \frac{I_b}{\Lambda}(G\phi)^{\perp}, \;\; (\overline{G\phi})^y_{full} + \frac{I_a}{\Lambda}(G\phi)^{\perp}),
$$

where, for $\Delta x = \Delta y$,

$$
(G\phi)^{\perp}_{i,j} = (\phi_{i+\frac{1}{2},j+\frac{1}{2}} + \phi_{i-\frac{1}{2},j-\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}})/\Delta x.
$$

The full Laplacian operator requires the third integral I^c because of the quadratic terms which results from the inner products of $\phi_{i+\frac{1}{2},j+\frac{1}{2}}$ and $\phi_{k+1/2,\ell+1/2}$; we leave the derivation to the reader. Note that for a full cell $\Lambda = 1, I^a = I^b = 0$, and so the stencils above for mixed cells reduce to those for full cells.

Note that solving (4.3) defines $\phi_{i+\frac{1}{2},j+\frac{1}{2}}$ at each node $(i+\frac{1}{2}, j+\frac{1}{2})$ for which the support of $\psi_{i+\frac{1}{2},j+\frac{1}{2}}(\mathbf{x})$ intersects the fluid region. Thus, the pressure is defined even at nodes contained in the body region, as long as they are within one cell of the fluid region.

5 Numerical Results

In this section we present results of calculations done using the Cartesian grid representation of bodies and/or boundaries of the domain. The first two sets of results are convergence studies; the first tests the accuracy of the projection alone, the second tests the accuracy of the full method for the Euler equations. In both cases, the results are presented in the form of tables which show the norms of the errors, as well as the calculated rates of convergence. The error for a calculation on a grid of spacing h is defined as the difference between the solution on that grid and the averaged solution from the same calculation on a grid of spacing $h/2$. In the first convergence study, the column 128-256 refers to the errors of the solution on the 128^2 grid as calculated by comparing the solution on the 128^2 grid to the solution on the 256^2 grid; similarly for 256-512. Once the errors are computed pointwise for each calculation, the L_1, L_2, and L_∞ norms of the errors are calculated. The convergence rates of the method can be estimated by taking the $\log_2$ of the ratio of these norms. This provides a heuristic estimate of the convergence rate assuming that the method is operating in its asymptotic range. We present two separate measures of the error. In the first we compute the norms over the entire domain ("All cells"). In the second we examine the error in an interior subdomain in order to measure errors away from the boundary. For the subdomain we have selected the region covered by full cells in the 128^2 grid ("Full 128 cells").

The first convergence study tests the accuracy of the projection alone. The domain is the unit square, and three bodies are placed in the interior. A circle is centered at $(.75, .75)$ and has radius .1; ellipses are centered at $(.25, .625)$ and $(.5, .25)$ and have axes $(.15, .1)$ and $(.2, .1)$, respectively. The boundary conditions are inflow at $x = 0$, outflow ($p = 0$) at $x = 1$, and no-flow boundaries at $y = 0$ and $y = 1$.

In this study, the initial data inside the numerical domain is $(u, v) = (1, 0)$; this data is then projected to define a velocity field which is approximately divergence-free in the region of the domain not covered by the bodies. The pressure gradient which is used to correct the initial data represents the deviation of the potential flow solution from uniform flow.

In Tables 1 and 2 we present the results of this convergence study. The almost-second-order convergence in the L_1 norm, first-order convergence in the L_∞ norm, and approximately $h^{1.5}$ convergence in the L_2 norm, are what we would expect of a solution which is second-order in most of the domain and first-order at boundaries. This is consistent with the observation that the maximum absolute error on the mixed cells is an order of magnitude higher than the maximum error on the full cells for the finer grids. Figure 1 shows a contour plot of the magnitude of the error of the 256^2 calculation.

The second convergence study is of flow through a diverging channel. In this case we evaluate the velocity field at time $t = 1.0$ in order to demonstrate the order of the complete algorithm for flow that is smooth in and near the mixed cells. The problem domain is 4 x 1, and the fluid is restricted to flow between the curves y_{top} and y_{bot}, defined as

$$y_{bot} = \begin{cases} y_1 & \text{if } 0 \le x \le 1. \\ y_2 + .5(y_1 - y_2)(1. + \cos(\frac{\pi}{2}(x - 1.)) & \text{if } 1. < x < 3. \\ y_2 & \text{if } 3. \le x \le 4. \end{cases}$$

and

$$y_{top} = 1. - y_{bot},$$

with $y_1 = .2$ and $y_2 = .000001$. The calculation is run at CFL number 0.9, and the flow is initialized with the potential flow field corresponding to inflow velocity $u = 1.$ at the left edge, as computed by the initial projection. Tables 3 and 4 show the errors and convergence rates for the velocity components; here 128-256 refers to the errors of the solution on the 128x32 grid as calculated by comparing the solution on the 128x32 grid to the solution on the 256x64 grid; similarly for 256-512. As before, we compute the errors both on the full domain and on a subdomain defined as the region covered by full cells on the 128x32 grid. (In this case, the norms are scaled, as appropriate, by the area of the domain.) Again we see rates corresponding to global second-order accuracy but first-order accuracy near the boundaries. Figures 2a-b are contour plots of the $\log_{10}$ of the error in the velocity components. The error is clearly concentrated along the fluid-body boundaries; for the sake of the figures we have defined all errors less than 1% of the maximum error in each (including the values in the body cells) to be equal to 1% of the maximum error. There are ten contour intervals in each figure spanning these two orders of magnitude. In time the error advects further along $y = y_{bot}$ and $y = y_{top}$, but does not contaminate the interior flow. We note that for this particular flow the velocity field near the boundary is essentially parallel to the boundary, and in a more general case we might expect to see more contamination of the interior flow. Note, however, that the maximum error is less than .3% of the magnitude of the velocity.

The third example we present is that of flow past a half-cylinder. There is an extensive experimental and computational literature on the subject of flow past a cylinder in an infinite domain at low to moderate Reynolds number; see, for example [33] and [61] for recent experimental and computational results, respectively. Since the methodology presented here is for inviscid flow, we compute flow past a half-cylinder rather than a full cylinder so as to force the separation point to occur at the trailing edge. However, we present this calculation to demonstrate the type of application for which the Cartesian grid methodology would be most useful: a calculation in which one might be interested most in the flow features away from the body (i.e. the shed vortices downstream from the cylinder), but which requires the presence of the body in order to generate or modify those features.

The resolution of our calculation is 256 x 64, the domain is 4 x 1, and the diameter of the half-cylinder is .25. The inflow velocity at the left edge is $u = 1$; the boundary conditions are outflow at the right edge, no-flow boundaries at the top and bottom. The initial conditions are the potential flow with uniform inflow as calculated from the initial projection, combined with a small vortical perturbation to break the symmetry of the problem. Shown here are "snapshots" of the vorticity (Figure 3a) and a passively advected scalar (Figure 3b) at late enough time that the flow is periodic, and that the perturbation has been advected through the domain. The scalar was advected in from the center of the inflow edge. The Strouhal number is calculated to be approximately $D/(U_\infty T) \approx (.25)/(1.)(.80) \approx .31$, where $T \approx .80$ is the observed period of vortex shedding, $D = .25$ is the cylinder diameter, and $U_\infty = 1.$ is the free-stream (i.e. inflow) velocity. One would expect a value between .2 and .4 (see [33] and the references cited there), so this seems a reasonable value given the limitations of the comparison.

	All cells			Full 128 cells		
	128-256	Rate	256-512	128-256	Rate	256-512
L_1	1.05e-3	1.92	2.76e-4	8.83e-4	1.92	2.32e-4
L_2	1.52e-3	1.52	5.31e-4	7.90e-4	1.87	2.16e-4
L_∞	6.69e-2	1.10	3.09e-2	5.55e-3	1.41	2.09e-3

Table 1: Errors and convergence rates for the x-component of the pressure gradient.

	All cells			Full 128 cells		
	128-256	Rate	256-512	128-256	Rate	256-512
L_1	7.14e-4	1.92	1.89e-4	5.81e-4	1.93	1.53e-4
L_2	1.25e-3	1.46	4.57e-4	6.42e-4	1.81	1.83e-4
L_∞	5.55e-2	1.08	2.63e-2	5.55e-3	1.23	2.37e-3

Table 2: Errors and convergence rates for the y-component of the pressure gradient.

	All cells			Full 128 cells		
	128-256	Rate	256-512	128-256	Rate	256-512
L_1	6.13e-5	1.64	1.97e-5	4.30e-5	2.18	9.49e-6
L_2	2.08e-4	1.21	8.99e-5	1.45e-4	1.79	4.20e-5
L_∞	2.36e-3	.80	1.35e-3	1.28e-3	1.08	6.04e-4

Table 3: Errors and convergence rates for the x-component of the velocity.

	All cells			Full 128 cells		
	128-256	Rate	256-512	128-256	Rate	256-512
L_1	1.01e-5	1.95	2.59e-6	8.28e-6	2.12	1.91e-6
L_2	2.67e-5	1.71	8.13e-6	1.90e-5	1.94	4.95e-6
L_∞	1.34e-3	1.55	4.56e-4	1.54e-4	1.10	7.72e-5

Table 4: Errors and convergence rates for the y-component of the velocity.

6 Conclusion

We have presented a method for calculation of time-dependent incompressible inviscid flow in a domain with embedded boundaries. This approach combines the basic projection method, using an approximate projection, with the Cartesian grid representation of geometry. In this approach, the body is represented as an interface embedded in a regular Cartesian mesh. The adaptation of the higher-order upwind method to include geometry is modeled on the Cartesian grid method for compressible flow. The discretization of the body near the flow uses a volume-of-fluid representation with a redistribution procedure. The approximate projection incorporates knowledge of the body through volume and area fractions, and certain integrals over the mixed cells. Convergence results indicate that the method is first-order at and near the body, and globally second-order away from the body. The method is demonstrated on flow past a half-cylinder with vortex shedding, and is shown to give a reasonable Strouhal number.

The method here is presented in two dimensions; the extension to $r - z$ and three dimensions and to variable density flows, and the inclusion of this representation with an adaptive mesh refinement algorithm for incompressible flow are being developed. The techniques developed here are also being modified for use in more general low Mach number models. In particular, we are using this methodology to model low Mach number combustion in realistic industrial burner geometries and to represent terrain in an anelastic model of the atmosphere.

7 Acknowledgements

We would like to thank Rick Pember for the many conversations on the subtleties of the Cartesian grid method.

References

[1] M. J. Aftosmis, M. J. Berger, and J. E. Melton, *Adaptation and surface modeling for cartesian mesh methods*, in 12th AIAA Computational Fluid Dynamics Conference, San Diego, June 19–22, 1995.

[2] S. E. Allwright, *Techniques in multiblock domain decomposition and surface grid generation*, in Numerical Grid Generation for Computational Fluid Mechanics '88, S. Sengupta, J. Häuser, P. R. Eiseman, and J. F. Thompson, eds., Pineridge Press, 1988.

[3] A. S. Almgren, J. B. Bell, and W. G. Szymczak, *A numerical method for the incompressible Navier-Stokes equations based on an approximate projection*, SIAM J. Sci. Comput., (to appear).

[4] S. Bayyuk, K. Powell, and B. van Leer, *A simulation technique for 2-d unsteady inviscid flows around arbitrarily moving and deforming bodies of arbitrary geometry*, in Proceedings, AIAA 11th Computational Fluid Dynamics Conference. Orlando, FL., July, 1993, p.423-437, AIAA Paper 93-3391-CP.

[5] J. B. Bell, P. Colella, and H. M. Glaz, *A second-order projection method for the incompressible Navier-Stokes equations*, J. Comput. Phys., 85 (1989), pp. 257–283.

[6] J. B. Bell, P. Colella, and L. H. Howell, *An efficient second-order projection method for viscous incompressible flow*, in 10th AIAA Computational Fluid Dynamics Conference, Honolulu, June 24–27, 1991.

[7] J. B. Bell, P. Colella, J. A. Trangenstein, and M. Welcome, *Adaptive mesh refinement on moving quadrilateral grids*, in 9th AIAA Computational Fluid Dynamics Conference, Buffalo, 1989.

[8] J. B. Bell, P. Colella, and M. Welcome, *Conservative front-tracking for inviscid compressible flow*, in 10th AIAA Computational Fluid Dynamics Conference, Honolulu, June 24–27, 1991.

[9] J. B. Bell, J. M. Solomon, and W. G. Szymczak, *A projection method for viscous incompressible flow on quadrilateral grids*, AIAA Journal, (to appear).

[10] J. Benek, J. Steger, and F. Dougherty, *A flexible grid embedding technique with application to the Euler equations*, in Proceedings of the 6th AIAA Computation Fluid Dynamics Conference, July 1983.

[11] M. Berger and A. Jameson, *Automatic adaptive refinement for the Euler equations*, AIAA Journal, 23 (1984), pp. 561–568.

[12] M. Berger and R. LeVeque, *An adaptive Cartesian mesh algorithm for the Euler equations in arbit rary geometries*, in Proceedings, AIAA 9th Computational Fluid Dynamics Conference. Buffalo, New York, June 14-16, 1989, p.1-7.

[13] ——, *A rotated difference scheme for Cartesian grids in complex geometries*, in Proceedings, AIAA 10th Computational Fluid Dynamics Conference. Honolulu, Hawaii, June 24-27, 1991, p.1-7, AIAA Paper 91-1602.

[14] ——, *Stable boundary conditions for Cartesian grid calculations*, Comp. Sys. Eng., 1 (1990), pp. 305–311.

[15] W. L. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, PA, 1987.

[16] D. Caughey and A. Jameson, *Progress in finite-volume calculations for wind-fuselage combinations*, AIAA Journal, 18 (1980), pp. 1281–1288.

[17] I. Chern and P. Colella, *A conservative front tracking method for hyperbolic conservation laws*, Tech. Rep. UCRL JC-97200, Lawrence Livermore National Laboratory, July, 1987.

[18] I.-L. Chern and P. Colella, *A conservative front tracking method for hyperbolic conservations laws*, Tech. Rep. UCRL-97200, LLNL, July 1987.

[19] G. Chessire and W. D. Henshaw, *Composite overlapping meshes for the solution of partial differential equations*, J. Comput. Phys., 90 (1990), pp. 1–64.

[20] Y. Chiang, B. van Leer, and K. Powell, *Simulation of unsteady inviscid flow on an adaptively refined Cartesian grid*, Tech. Rep. AIAA Paper 92-0443-CP, 1992.

[21] S. Choi and B. Grossman, *A flux-vector split finite-volume method for Euler's equations on non-m apped grids.* AIAA 26th Aerospaces Meeting, AIAA Paper 88-0227, Reno, NV, Jan., 1988.

[22] A. J. Chorin, *Numerical solution of the navier-stokes equations*, J. Math. Comp., 22 (1968), pp. 745–762.

[23] D. Clarke, M. Salas, and H. Hassan, *Euler calculations for multielement airfoils using Cartesian grids*, AIAA Journal, 24 (1986), pp. 353–358.

[24] W. Coirier and K. Powell, *An accuracy assessment of Cartesian-mesh approaches for the Euler equ ations*, in Proceedings, AIAA 11th Computational Fluid Dynamics Conference. Orlando, FL., July, 1993, p.423-437.

[25] W. J. Coirier, *An adaptively-refined, Cartesian, cell-based scheme for the Euler and Navier-Stokes equations*, Tech. Rep. NASA Technical Memorandum 106754, NASA Lewis Research Center, Cleveland, OH, October 1994. PhD thesis.

[26] P. Colella, *A direct Eulerian MUSCL scheme for gas dynamics*, SIAM J. Comput., 6 (1985), pp. 104–117.

[27] ——, *Multidimensional upwind methods for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 171–200.

[28] B. Epstein, A. Luntz, and A. Nachson, *Cartesian Euler methods for arbitrary aircraft configurations*, AIAA Journal, 30 (1992), pp. 679–687.

[29] S. Falle and J. Giddings, *An adaptive multigrid applied to supersonic blunt body flow*, in Numerical Methods in Fluid Dynamics, III, K. Morton and M. Baines, eds., Clarendon Press, 1988.

[30] R. Gaffney, H. Hassan, and M. Salas, *Euler calculations for wings using Cartesian grids.* AIAA 25th Aerospaces Meeting, AIAA Paper 87-0356, Reno, NV, Jan., 1987.

[31] C. Gooch and H. Öksüzoğlu, *Extension of state vector splitting to the Navier-Stokes equations*, in Proceedings, AIAA 11nd Computational Fluid Dynamics Conference. Orlando, FL., July, 1993, p.832-840.

[32] J. A. Greenough, V. Beckner, R. B. Pember, W. Y. Crutchfield, J. B. Bell, and P. Colella, *An adaptive multifluid interface-capturing method for compressible flow in complex geometries*, in 11th AIAA Computational Fluid Dynamics Conference, San Diego, June 19–22, 1995.

[33] M. Hammache and M. Gharib, *An experimental study of the parallel and oblique vortex shedding from circular cylinders*, J. Fluid Mech., 232 (1991), pp. 567–590.

[34] W. D. Henshaw, *A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids*, J. Comput. Phys., 113 (1994), pp. 13–25.

[35] T. Holst and W. Balhaus, *Fast, conservative schemes for the full potential equation applied to transonic flows*, AIAA Journal, 17 (1979), pp. 145–152.

[36] C. J. Hwang and J. Liu, *Inviscid and viscous solutions for airfoil/cascade flows using a locally implicit algorithm on adaptive meshes*, Journal of Turbomachinery, 113 (1991), pp. 553–560.

[37] A. Jameson, *Transonic potential flow calculations using conservative form*, in Proceedings, AIAA 2nd Computational Fluid Dynamics Conference. June, 1975, p.148-155.

[38] A. Jameson and T. Baker, *Improvements to the aircraft Euler method.* AIAA 25th Aerospaces Meeting, AIAA Paper 87-0452, Reno, NV, Jan., 1987.

[39] A. Jameson, T. Baker, and N. Weatherill, *Calculation of inviscid transonic flow over a complete aircraft.* AIAA 24th Aerospaces Meeting, AIAA Paper 86-0103, Reno, NV, Jan., 1986.

[40] F. Johnson, R. James, J. Bussoletti, , A. Woo, and D. Young, *A transonic rectangular panel method.* AIAA Paper 82-0953, 1982.

[41] R. LeVeque, *A large time step generalization of Godunov's method for systems of conservation laws*, SIAM J. Numer. Anal., 22 (1985), pp. 1051–1073.

[42] ——, *Cartesian grid methods for flow in irregular regions*, in Numerical Methods in Fluid Dynamics, III, K. Morton and M. Baines, eds., Clarendon Press, 1988, pp. 375–382.

[43] ——, *High resolution finite volume methods on arbitrary grids via wave propaga tion*, J. Comput. Phys., 78 (1988), pp. 36–63.

[44] R. Löhner, D. Martin, and R. Ramamurti, *An implicit, linelet-based solver for incompressible flows*, Intl.J. Num. Methods. for Fluids, (to appear).

[45] R. Löhner, K. Morgan, and O. Zienkiewicz, *Adaptive grid refinement for the compressible Euler equations*, in Accuracy Estimates and Adaptivity for Finite Elements, I. Babuška, O. Zienkiewicz, J. Gago, and E. d. A. Oliveira, eds., Wiley, New York, 1986, p. 281.

[46] R. LÖHNER AND P. PARIKH, *Generation of three-dimensional unstructured grids by the advancing-fron t method*. AIAA Paper 88-0515, 1988.

[47] D. L. MARCUS, R. B. PEMBER, V. BECKNER, J. B. BELL, D. SIMKINS, AND M. WELCOME, *Multidimensional numerical simulation of a pulse combustor*, in AIAA Paper 94-2315, 1994.

[48] A. MATHUR, R. SANJAY, B. MADAVAN, AND R. G. RAJAGOPALAN, *A hybrid structured-unstructured grid method for unsteady turbomachinery flow compuations*, AIAA Paper, 93-0387 (1993).

[49] S. R. MATHUR, N. K. MADAVAN, AND R. G. RAJAGOPALAN, *A solution-adaptive hybrid-grid method for the unsteady analysis of turbomachinery*, AIAA Paper, 94-0645 (1993).

[50] D. MAVRIPLIS, *Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes*, AIAA Journal, 28 (1990), pp. 213–221.

[51] J. E. MELTON, M. J. BERGER, M. J. AFTOSMIS, AND M. D. WONG, *3d applications of a cartesian grid euler method*, in AIAA Paper 95-0853, 1995.

[52] K. MORINISHI, *A finite difference solution of the Euler equations on non-body fitted grids*, Comp. Fluids, 21 (1992), pp. 331–344.

[53] K. NAKAHASHI, O. NOZAKI, K. KIKUCHI, AND A. TAMURA, *Navier-Stokes computations of two- and three-dimensional cascade flowfields*, Journal of Propulsion and Power, 5 (1989), pp. 320–326.

[54] W. NOH, *Cel: A time-dependent, two-space-dimensional, coupled Eulerian-Lagran gian code*, in Fundamental Methods of Hydrodynamics, (Methods of Computational Physics, Volume 3), 1964, pp. 117–179.

[55] H. Öksüzoğlu, *State vector splitting to the Euler equations of gas dynamics.* AIAA paper, 92-0326, 1992.

[56] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, *An adaptive Cartesian grid method for unsteady compressible flow in complex geometries*, submitted for publication, (1993).

[57] ——, *Adaptive Cartesian grid methods for representing geometry in inviscid compressible flow*, in 11th AIAA Computational Fluid Dynamics Conference, Orlando, FL, July 6–9, 1993.

[58] J. Purvis and J. Burkhalter, *Prediction of critical Mach number for store configurations*, AIAA Journal, 17 (1979), pp. 1170–1177.

[59] J. Quirk, *A Cartesian grid approach with hierarchical refinement for compressible flows*, Tech. Rep. ICASE Report 94-51.

[60] ——, *An alternative to unstructured grids for computing gas dynamic flows arou nd arbitrarily complex two-dimensional bodies*, Comput. Fluids, 23 (1994), pp. 125–142.

[61] M. Rosenfeld, D. Kwak, and M. Vinokur, *A fractional step solution method for the unsteady incompressible Navier-Stokes equations in generalized coordinate systems*, JCP, 94 (1991), pp. 102–137.

[62] P. Rubbert, J. Bussoletti, F. Johnson, W. Sidwell, K.W. a nd Rowe, S. Samant, G. SenGupta, W. Weatherill, R. Burkhart, B. Everson, D. Young, and A. Woo, *A new approach to the solution of boundary value problems involving compl ex configurations*, in Computational Mechanics— Advances and Trends AMD vol. 75, A.K.Noor, ed., Clarendon Press, 1986, pp. 49–84.

[63] T. Schönfeld and P. Weinerfelt, *Automatic generation of quadrilateral multiblock grids by the advancing front technique*, in Third International Conference on Nu-

merical Grid Generation, A. S.-Arcilla, J. Häuser, P. R. Eisemann, and J. F. Thompson, eds., Elsevier Science Publishers, North Holland, 1991, pp. 743–754.

[64] M. Soetrisno, S. T. Imlay, and D. W. Roberts, *A zonal implicit procedure for hybrid structured-unstructured grids*, AIAA Paper, 94-0645 (1994).

[65] E. Steinthorsson, D. Modiano, and P. Colella, *Computations of unsteady viscous compressible flows using adaptive mesh refinement in curvilinear body-fitted grid systems*, Tech. Rep. NASA TM-106704, NASA Lewis Research Center, June 1994.

[66] M. Stewart, *A general decomposition algorithm applied to multi-element airfoil grids*, AIAA Paper, 90-1606 (1990).

[67] E. Y. Tau, *A second-order projection method for the incompressible Navier-Stokes equations in arbitrary domains*, JCP, 115 (1994), pp. 147–152.

[68] J. Thompson, Z. Warsi, and C. Mastin, *Boundary fitted coordinate systems for numerical solution of partial differential equations – A review*, J. Comput. Phys., 47 (1982), pp. 1–108.

[69] F. L. Tsung, J. Loellbach, O. Kwon, and C. Hah, *A three-dimensional structured/unstructured hybrid Navier-Stokes method for turbine blade rows*, Tech. Rep. NASA Technical Memorandum 106813, NASA Lewis Research Center, Cleveland, OH, December 1994.

[70] N. P. Weatherill, *Mixed structured-unstructured meshes for aerodynamic flow simulation*, Aeronautical Journal, (1990), pp. 111–123.

[71] B. Wedan and J. South, *A method for solving the transonic full-potential equations for general configurations*, in Proceedings, AIAA 6th Computational Fluid Dynamics Conference. Danvers, Mass., July, 1993, AIAA Paper 83-1889.

[72] D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussoletti, *A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics*, J. Comput. Phys., 62 (1991), pp. 1–66.

[73] Y. Zang, R. L. Street, and J. R. Koseff, *A non-staggered grid, fractional step method for time-dependent incompressible Navier-Stokes equations in curvilinear coordinates*, J. Comput. Phys., 114 (1994), pp. 18–33.

[74] D. Zeeuw and K. Powell, *An adaptively refined Cartesian mesh solver for the Euler equations*, J. Comput. Phys., 104 (1993), pp. 56–68.