A Cell-Centered 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 †
Dept. of Mechanical Engineering
University of California at Berkeley
Berkeley, CA 94720

1 Abstract

Many problems in fluid dynamics have domains with complicated internal or external boundaries of the flow. Here we present a method for calculating time-dependent incompressible inviscid flow using a "Cartesian grid" approach for representing geometry. In this approach, the body is represented as an interface embedded in a regular Cartesian mesh. The basic algorithm is a fractional-step projection method based on an approximate projection. 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 with a redistribution procedure to eliminate time-step restrictions due to small cells where the boundary intersects the mesh. The approximate projection incorporates knowledge of the body through volume and area fractions. The method is demonstrated on flow past a half-cylinder with vortex shedding.

2 Introduction

Modeling of low Mach number flows in complex geometries is often required in engineering applications. In this paper we present a Cartesian grid algorithm for the unsteady incompressible Euler equations. in which the problem geometry is represented as a "tracked front" embedded in a uniform Cartesian grid. The incompressible Euler equations provide a prototype for more general low Mach number flows such as low speed combustion ([7, 9, 10]). The basic integration scheme uses a fractional step approach in which the nonlinear convection equations are approximated to construct a velocity field without enforcing the divergence constraint. In the second step of the algorithm a discrete projection is applied to the intermediate velocity field computed in the first step to enforce the incompressibility

*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 a National Science Foundation Presiditional Young Investigator award under grant ACS-8958522.

constraint. The adaptation of the basic projection methodology to the Cartesian grid setting combines two different types of techniques developed for compressible flows. The key issue in the advection step is eliminating time-step restrictions due to small cells where the boundary intersects the mesh, and to address this we have adapted the redistribution techniques developed by Chern and Colella [5] and Pember et al [11] for gas dynamics. The work present4ed here is similar to earlier work of Almgren et al [1] who present a similar algorithm. The algorithm presented here uses a different projection which requires less information about the geometry so that it is more amenable to automating the construction of the geometric description.

In the next section we review the basic fractional step algorithm, introduce the notation of the Cartesian grid method, and describe the advection and projection steps 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 will be presented in later work.

3 Basic Algorithm

3.1 Overview of Fractional Step Formulation

The incompressible Euler equations for constant density flows can be written as

$$U_t + (U \cdot \nabla)U + \nabla p = 0 \tag{3.1.1}$$

$$\nabla \cdot U = 0. \tag{3.1.2}$$

Alternatively, (3.1.1) could be written as

$$U_t + \nabla \cdot (UU) + \nabla p = 0 \tag{3.1.3}$$

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 (3.1.3),

$$\frac{U^* - U^n}{\Delta t} + (U \cdot (\nabla U))^{n+\frac{1}{2}} + \nabla p^{n-\frac{1}{2}} = 0 \tag{3.1.4}$$

for the intermediate velocity U^*; as will be discussed in section 3.1.5, for small cells adjoining the body we modify the velocity update using the convective formulation of the nonlinear terms. The pressure gradient at $t^{n-\frac{1}{2}}$ was computed in the previous time step and is treated as a source term in (3.1.4). The advection terms $(\nabla \cdot (UU))^{n+\frac{1}{2}}$, 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 3.1.6.

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 which correspond, respectively, to the update to the pressure gradient and the update to the velocity. In particular, if $\mathbf{P}$ represents the projection operator then

$$\frac{U^{n+1} - U^n}{\Delta t} = \mathbf{P}\left(\frac{U^* - U^n}{\Delta t}\right) \tag{3.1.5}$$

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

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

3.2 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, which lie inside the flow domain, also known as edge apertures. A cell entirely within the fluid but which shares an edge with a full body cell would have a volume fraction of $\Lambda = 1$ but the area fraction of the edge corresponding with the body would be zero. Additionally, external domain boundaries can be described using zero aperture edges. We label a cell entirely within the fluid

($\Lambda = 1$) as a *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*.

In the method presented here, the state of the fluid at time t^n is defined by $U_{ij}^n = (u_{ij}^n, v_{ij}^n)$, the velocity field in cell $B_{i,j}$ at time t^n, and $p_{i,j}^{n-\frac{1}{2}}$, the pressure in cell (i,j) at $t^{n-\frac{1}{2}}$. The pressure gradient $Gp_{ij}^{n-\frac{1}{2}}$ represents the average value of ∇p in cell $B_{i,j}$ at time $t^{n-\frac{1}{2}}$.

3.3 Discretization of Projection Operators

The original discretization of the exact projection operator has the form

$$\mathbf{P} = I - G(L)^{-1}D. \tag{3.3.1}$$

where D and G represent divergence and gradient operators, respectively and $L = DG$. For this algorithm, we utilize an approximate projection, i.e., $\mathbf{P}^2 \neq \mathbf{P}$ which is described in detail in [8]. The motivation for using an approximate projection instead of an exact projection comes from the simplicity of the linear algebra associated with the approximate projection and is described in detail in [2]. The projection operator requires solution of the equation

$$L\phi = \rho, \tag{3.3.2}$$

where ϕ is a scalar potential, which in the context of (3.1.5) is the update to the pressure, and ρ is the divergence of the vector field being projected. In order to calculate the Laplacian L we define here the traditional MAC divergence and gradient operators. We first define a divergence operator D^{MAC} which operates on edge-based vectors:

$$D^{MAC}F = \frac{1}{\Lambda_{i,j}} \cdot \left(\frac{a_{i+\frac{1}{2},j}F^x_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j}F^x_{i-\frac{1}{2},j}}{\Delta x} + \frac{a_{i,j+\frac{1}{2}}F^y_{i,j+\frac{1}{2}} - a_{i,j-\frac{1}{2}}F^y_{i,j-\frac{1}{2}}}{\Delta y} \right), \tag{3.3.3}$$

Next we define the MAC gradient operators as follows:

$$\begin{aligned} (G^{MAC,x}\phi)_{i+\frac{1}{2},j} &= \frac{(\phi_{i+1,j} - \phi_{i,j})}{\Delta x} \\ (G^{MAC,y}\phi)_{i,j+\frac{1}{2}} &= \frac{(\phi_{i,j+1} - \phi_{i,j})}{\Delta y}. \end{aligned} \tag{3.3.4}$$

Now we have what we need to define the Laplacian for use in (3.3.2):

$$L = D^{MAC}G^{MAC}.$$

In the case of no embedded boundaries the resulting stencil for the Laplacian is the standard five-point stencil, resulting in straightforward linear algebra to solve (3.3.2). Note that because of the multiplication of fluxes by areas in defining the divergence operator, the Laplacian operator L does not require that the gradient be defined on edges with zero length, i.e. ϕ need only be defined in fluid or mixed cells.

In order to solve (3.3.2), point-relaxation is used with a multigrid accelerator. At the coarsest level of multigrid, a conjugate gradient solver is employed to further speed convergence. Typical multigrid convergence acceleration characteristics are exhibited by the solver.

3.4 Cell-Centered Projection

In our fractional step formulation, we project the vector field $V = (V^x, V^y) \equiv \frac{U^* - U^n}{\Delta t}$ as defined in (3.1.4) onto a subspace of approximately divergence-free vector fields. To do this according to (3.3.1), we first must define a divergence given cell-based values of the temporal derivatives of the velocity field V. In order to calculate this divergence we construct an edge-centered vector field F from the cell-centered field as follows:

$$\begin{aligned} F^x_{i+\frac{1}{2},j}(V) &= \frac{1}{2}(V^x_{i+1,j} + V^x_{i,j}) \\ F^y_{i,j+\frac{1}{2}}(V) &= \frac{1}{2}(V^y_{i,j+1} + V^y_{i,j}). \end{aligned} \tag{3.4.1}$$

Then the divergence is found by applying (3.3.3) to the field F.

The cell-centered gradient operator G must define gradients on all mixed or fluid cells, even

those adjacent to body cells, and to do this must extrapolate values of the gradient from cells that do not border body cells into cells bordering body cells. The following protocol is used to calculate this gradient:

if $(a_{i+\frac{1}{2},j} > 0)$ and $(a_{i-\frac{1}{2},j} > 0)$ then
  $(G^{*x}\phi)_{i,j} = \frac{1}{2\Delta x}(\phi_{i+1,j} - \phi_{i-1,j})$
else if $(a_{i+\frac{1}{2},j} = 0)$ then
  $= 2 * (G^{*x}\phi)_{i-1,j} - (G^{*x}\phi)_{i-2,j}$
else if $(a_{i-\frac{1}{2},j} = 0)$ then
  $= 2 * (G^{*x}\phi)_{i+1,j} - (G^{*x}\phi)_{i+2,j}$

if $(a_{i,j+\frac{1}{2}} > 0)$ and $(a_{i,j-\frac{1}{2}} > 0)$ then
  $(G^{*y}\phi)_{i,j} = \frac{1}{2\Delta y}(\phi_{i,j+1} - \phi_{i,j-1})$
else if $(a_{i,j+\frac{1}{2}} = 0)$ then
  $= 2 * (G^{*y}\phi)_{i,j-1} - (G^{*y}\phi)_{i,j-2}$
else if $(a_{i,j-\frac{1}{2}} = 0)$ then
  $= 2 * (G^{*y}\phi)_{i,j+1} - (G^{*y}\phi)_{i,j+2}$.

It is important to note that for cells with four nonzero apertures, the gradient operator in the projection is just the standard cell-centered gradient operator, which can be thought of as the average of the appropriate two edge gradients G.

3.5 Advective Derivatives

This section will discuss the steps required to calculate the term $[\nabla \cdot (UU)]_{i,j}^{n+\frac{1}{2}}$ for use in (3.1.4), given a set of nearly divergence-free edge-centered velocities centered in time at $t^{n+\frac{1}{2}}$. The algorithm is a predictor-corrector method, similar to that used in [3], but with some modifications as discussed in [4]. The details of the current version without geometry are given in [2]. 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 straight forward.

For the construction of the nonlinear advective terms at $t^{n+\frac{1}{2}}$, velocities are defined on all edges of fluid 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 fluid 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 intermediate velocity U^* at time $n+1$ is then defined on all fluid and mixed cells from (3.1.4) as

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

The time-step restriction of the upwind method as given by the full-cell stability analysis,

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

is used to set the time step for the overall algorithm; here σ is the CFL number.

As mentioned earlier, for incompressible flow the two forms of the momentum equation, (3.1.1) and (3.1.3), are analytically equivalent. In the presence of embedded boundaries, the approach using convective derivatives is stable even for very small cells, but the convective update and the conservative update are no longer equivalent. To construct the convective difference, we must in some cases extrapolate onto edges which lie fully or partially within the body, and so the convective update does not "see" the body other than through the MAC-projected normal advection velocities (and the limited slopes).

The conservative update, by contrast, more correctly represents the body, but is not, in general, stable for small cells without a reduction in time step. In fact, the time step restriction is such that as the cell volume goes to zero the time step must also approach zero. We define the conservative update by

$$\frac{U^* - U^n}{\Delta t} + \nabla \cdot \mathbf{F}^{n+\frac{1}{2}} + \nabla p^{n-\frac{1}{2}} = 0, \qquad (3.5.1)$$

where $\mathbf{F}^{n+\frac{1}{2}} = (u_{i+\frac{1}{2},j}^{n+\frac{1}{2}} U_{i+\frac{1}{2},j}^{n+\frac{1}{2}}, v_{i,j+\frac{1}{2}}^{n+\frac{1}{2}} U_{i,j+\frac{1}{2}}^{n+\frac{1}{2}})$, the MAC projected edge velocities as defined later, and evaluate the divergence as $\int \mathbf{F}^{n+\frac{1}{2}} \cdot \mathbf{n}\, dA$, where the flux is only integrated on the parts of the edges of the cell that lie within the fluid.

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 with respect to the maximum cell size. To maintain conservation in the advection step, the momentum which does not pass into the small cell is redistributed to the neighboring cells in a volume weighted fashion. This approach is modeled on the algorithm of [11, 12], and is based on the algebraic redistribution scheme of Chern and Colella [5]. The algorithm is as follows:

1. In all cells construct $\widetilde{\widetilde{U}}^*$ using the conservative form of the advective terms:

$$\begin{aligned}\Lambda_{i,j}\widetilde{\widetilde{U}}^* &= \Lambda_{i,j}U^n \\ &\quad -\tfrac{\Delta t}{\Delta x}(a_{i+\frac{1}{2},j}F^x_{i+\frac{1}{2},j} - a_{i-\frac{1}{2},j}F^x_{i-\frac{1}{2},j}) \\ &\quad -\tfrac{\Delta t}{\Delta y}(a_{i,j+\frac{1}{2}}F^y_{i,j+\frac{1}{2}} - a_{i,j-\frac{1}{2}}F^y_{i,j-\frac{1}{2}}),\end{aligned}$$

   where the fluxes are defined as in (3.5.1). This solution enforces no-flow across the boundary of the body, but is not necessarily stable.

2. In mixed cells only construct $\widetilde{U}^*$, defined as

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

   The advective term $[(U\cdot\nabla)U]^{n+\frac{1}{2}}$ can be found from the edge- and time-centered velocities as:

$$\begin{aligned}[(U\cdot\nabla)U]^{n+\frac{1}{2}}_{i,j} &= \frac{1}{2\Delta x}\left(u^{n+\frac{1}{2}}_{i+\frac{1}{2},j} + u^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right) * \\ &\qquad \left(U^{n+\frac{1}{2}}_{i+\frac{1}{2},j} - U^{n+\frac{1}{2}}_{i-\frac{1}{2},j}\right) + \\ &\qquad \frac{1}{2\Delta y}\left(v^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} + v^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\right) * \\ &\qquad \left(U^{n+\frac{1}{2}}_{i,j+\frac{1}{2}} - U^{n+\frac{1}{2}}_{i,j-\frac{1}{2}}\right)\end{aligned}$$

   The technique for calculating the time- and edge-centered velocities is outlined below in section 3.6. We will refer to $\widetilde{U}^*$ as the *reference state.*

3. In order to address the stability issue for mixed cells, define the difference

$$\delta M_{i,j} = \Lambda_{i,j}(\widetilde{\widetilde{U}}^*_{i,j} - \widetilde{U}^*_{i,j}) \qquad (3.5.2)$$

   in mixed cells only. This is the flux to be redistributed.

4. The conservative solution can be written on mixed cells as $\widetilde{\widetilde{U}}^*_{i,j} = \widetilde{U}^*_{i,j} + \frac{\delta M_{i,j}}{\Lambda_{i,j}}$. However, this solution is not stable for $\Lambda \ll 1$. Define instead $\widehat{U}^*_{i,j} = \widetilde{U}^*_{i,j} + \delta M_{i,j}$; in other words allow 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. Note that for fluid cells $\widehat{U}^*_{i,j} = \widetilde{\widetilde{U}}^*$.

5. Redistribute the remaining fraction of δM from mixed cells, that is $(1-\Lambda_{i,j})\delta M_{i,j}$, 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

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

6. Finally, subtract the lagged pressure gradient term from the solution for all fluid and mixed cells, treating it as a source term:

$$U^*_{i,j} = \widehat{\widehat{U}}^*_{i,j} - \Delta t\,(Gp)^{n-\frac{1}{2}}_{i,j}. \qquad (3.5.3)$$

   This vector field U^* is the approximation to the velocity field at t^{n+1}; all that remains to define U^{n+1} is the projection step.

3.6 Calculation of Edge-Centered Velocities

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 in the expansion is replaced using (3.1.1), neglecting for the time being the pressure gradient. For edge $(i+\frac{1}{2},j)$ this gives

$$\widehat{U}^L_{i+\frac{1}{2},j} = U^n_{ij} + \left(\frac{\Delta x}{2} - \frac{u_{ij}\Delta t}{2}\right) U^n_{x,ij} - \frac{\Delta t}{2}(\widehat{vU_y})_{ij}, \qquad (3.6.1)$$

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

$$\widehat{U}^R_{i+\frac{1}{2},j} = U^n_{i+1,j} - \left(\frac{\Delta x}{2} + \frac{u_{i+1,j}\Delta t}{2}\right) U^n_{x,i+1,j} - \frac{\Delta t}{2}(\widehat{vU_y})_{i+1,j}, \qquad (3.6.2)$$

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) are evaluated using a monotonicity-limited second-order slope approximation [6]. These limited slopes are calculated by first evaluating the central, 'plus,' and 'minus,' differences as follows

$$\begin{aligned} \Delta^- &= 2*(U_{i,j} - U_{i-1,j}) \\ \Delta^0 &= .5*(U_{i+1,j} - U_{i-1,j}) \\ \Delta^+ &= 2*(U_{i+1,j} - U_{i,j}). \end{aligned}$$

The limited slope is then defined as

$$U_x = \text{sign}(\Delta^0)\min(|\Delta^0|, |\Delta^-|, |\Delta^+|).$$

To account for the embedded boundaries, Δ^- or Δ^+ are set to zero if the edge over which the difference is outside the domain. This effectively reduces the slope calculation to first order near embedded boundaries, but ensures that only values from fluid or mixed cells, not extended states, are used in the slope calculation.

The transverse derivative terms as used in (3.6.1) and (3.6.2) are evaluated as in [1, 4], by choosing a single sided difference in the upwind direction.

For every edge in the fluid domain, (3.6.1) and (3.6.2) each give a time-centered estimate of the velocity vector. A unique edge- and time-centered value for the velocity is obtained from the edge states by using an upwinding procedure defined below. The procedure to choose the unique state $\widehat{U}_{i+\frac{1}{2},j}$ given the left and right states, $U^L \equiv \widehat{U}^L_{i+\frac{1}{2},j}$ and $U^R \equiv \widehat{U}^R_{i+\frac{1}{2},j}$ is given as:

$$\widehat{U}_{i+\frac{1}{2},j} = \begin{cases} U^L & \text{if } u_{i,j} \text{ and } u_{i+1,j} > 0 \\ \frac{1}{2}(U^L + U^R) & \text{if } u_{i,j} * u_{i+1,j} < 0 \\ U^R & \text{if } u_{i,j} \text{ and } u_{i+1,j} < 0 \end{cases}.$$

An additional condition on the upwinding is that the upwind state must be chosen from a fluid cell. We follow a similar procedure to construct $U_{i,j+\frac{1}{2}}$.

In general these normal velocities at the edges are not divergence-free. In order to make these velocities divergence-free we apply a MAC projection [4] to the edge-centered velocity field.

$$\mathbf{P}^{MAC} = I - G^{MAC*}(L)^{-1}D^{MAC}. \qquad (3.6.3)$$

The MAC projection operates on the edge-centered velocities, $\widehat{U}_{i+\frac{1}{2},j}, \widehat{U}_{i,j+\frac{1}{2}}$. The generic divergence operator (3.3.3) operates on these velocities directly, and the same standard gradient operator is used in the construction of DG. However, the gradient operator G^{MAC*} differs from G^{MAC} in that G^{MAC} yields only the normal edge-centered gradients from a cell-centered scalar field, but G^{MAC*} defines both tangential and normal gradients. This requires additional extrapolation in the case of mixed or fluid cells adjacent to body cells. Calculating this gradient is done in a series of steps:

- Calculate all normal edge gradients using the following logic.

  if $(a_{i+\frac{1}{2},j} > 0)$ then
  $(G^{MAC*x}\phi)_{i+\frac{1}{2},j} = (\phi_{i+1,j} - \phi_{i,j})\frac{1}{\Delta x}$
  else if $(\Lambda_{i,j} < 0)$ then
  $= 2*(G^x\phi)_{i+1,j} - (G^x\phi)_{i+2,j}$
  else
  $= 2*(G^x\phi)_{i-1,j} - (G^x\phi)_{i-2,j}$

  In order for these gradients to be well defined as above, the normal gradients as found

by the first above conditional must be calculated throughout the domain before any gradients are found by extrapolation. This calculation includes boundary normal gradients if any. Similar formulae yield the y direction normal gradients.

- Calculate the tangential gradients at the horizontal edges as follows.

$$
\begin{aligned}
&\text{if } (a_{i,j+\frac{1}{2}} = 1) \text{ then} \\
&\quad (G^{MAC*x}\phi)_{i,j+\frac{1}{2}} = .25 * (\\
&\qquad (G^x\phi)_{i-\frac{1}{2},j+1} + (G^x\phi)_{i+\frac{1}{2},j+1} + \\
&\qquad (G^x\phi)_{i-\frac{1}{2},j} + (G^x\phi)_{i+\frac{1}{2},j}) \\
&\text{else if } (a_{i-1,j+\frac{1}{2}} > 0) \text{ then} \\
&\quad = 2 * (G^x\phi)_{i-1,j+\frac{1}{2}} - (G^x\phi)_{i-2,j+\frac{1}{2}} \\
&\text{else} \\
&\quad = 2 * (G^x\phi)_{i+1,j+\frac{1}{2}} - (G^x\phi)_{i+2,j+\frac{1}{2}}
\end{aligned}
$$

  Analogous formulae yield the y direction tangential gradients.

- Calculate the tangential MAC gradients on the boundaries where needed as follows:

$$
\begin{aligned}
&G^{MAC*x}\phi_{0,j} = \\
&\quad 2 * (G^x\phi)_{1,j} - (G^x\phi)_{2,j} \\
&G^{MAC*x}\phi_{nx,j} = \\
&\quad 2 * (G^x\phi)_{nx-1,j} - (G^x\phi)_{nx-2,j}
\end{aligned}
$$

With these definitions of the gradient operators and the divergence operator we define the following MAC velocities:

$$U^{n+\frac{1}{2}}_{i+\frac{1}{2},j} = \widehat{U}_{i+\frac{1}{2},j} - (G\phi)_{i+\frac{1}{2},j} \qquad (3.6.4)$$

4 Numerical Results

Figure 1 shows results of a calculation done using the algorithm presented in this paper. In this figure are time series of u,v, and the vorticity. The test problem is a half circle facing to the left in a 4x1 channel. The center of the half-circle is located at $(1, .5)$, and the diameter of the half-circle is 0.25. The flow is moving left to right over the half circle and between rigid walls top and bottom. An outflow boundary condition is imposed on the right edge. The initial conditions are defined by the projection of a uniform inlet velocity and quiescent fluid with a slight asymmetric perturbation upstream of the obstruction.

5 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. Convergence results for the projection itself are given as is a result from the projection calculation.

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.

6 References

[1] A. S. Almgren, J. B. Bell, P. Colella, and T. C. Marthaler. Cartesian mesh projection methods for the incompressible euler equations in complex geometries.

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

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

[4] 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.

[5] I.-L. Chern and P. Colella. A conservative front tracking method for hyperbolic conservations laws. Technical Report UCRL-97200, LLNL, July 1987.

[6] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM Journal on Computing*, 6:104–117, January 1985.

[7] M. Lai, J. B. Bell, and P. Colella. An approximate projection for reacting flow in the zero Mach number limit. *in preparation.*

[8] M. Lai and P. Colella. An approximate projection for the incompressible Navier-Stokes equations. *in preparation.*

[9] M. Fruchtman Lai. *A Projection Method for Reacting Flow in the Zero Mach Number Limit.* PhD thesis, University of California at Berkeley, 1993.

[10] R. Pember, J. B. Bell, and M. Lai. Numerical simulation of an industrial burner. *in preparation.*

[11] Richard B. Pember, John B. Bell, Phillip Colella, William Y. Crutchfield, and Michael L. Welcome. Adaptive Cartesian grid methods for representing geometry in inviscid compressible flow. In *11th AIAA Computational Fluid Dynamics Conference*, Orlando, FL, July 6–9, 1993.

[12] Richard B. Pember, John B. Bell, Phillip Colella, William Y. Crutchfield, and Michael L. Welcome. An adaptive Cartesian grid method for unsteady compressible flow in complex geometries. *submitted for publication*, 1993.