517-34
197149
p. 15

A Multilevel Adaptive Projection Method for Unsteady Incompressible Flow*

Louis H. Howell
Lawrence Livermore National Laboratory
Livermore, CA 94550

ABSTRACT

There are two main requirements for practical simulation of unsteady flow at high Reynolds number: the algorithm must accurately propagate discontinuous flow fields without excessive artificial viscosity, and it must have some adaptive capability to concentrate computational effort where it is most needed. We satisfy the first of these requirements with a second-order Godunov method similar to those used for high-speed flows with shocks, and the second with a grid-based refinement scheme which avoids some of the drawbacks associated with unstructured meshes.

These two features of our algorithm place certain constraints on the projection method used to enforce incompressibility. Velocities are cell-based, leading to a Laplacian stencil for the projection which decouples adjacent grid points. We discuss features of the multigrid and multilevel iteration schemes required for solution of the resulting decoupled problem. Variable-density flows require use of a modified projection operator—we have found a multigrid method for this modified projection that successfully handles density jumps of thousands to one. Numerical results are shown for the 2D adaptive and 3D variable-density algorithms.

INTRODUCTION

The incompressible flow algorithm presented by Bell, Colella and Glaz [3] combines the original projection method of Chorin [9, 10] with the Godunov methodology developed by Colella [11] to yield a robust scheme which is second-order in both space and time. In [5] Bell and Marcus extend this method to handle flows involving spatial density variations.

Originally developed for gas dynamics problems with strong shocks, the second-order Godunov technology gives the algorithm the ability to propagate discontinuous

---

*This work 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 was provided by the Applied Mathematical Sciences Program of the Office of Energy Research under contract No. W-7405-Eng-48, and by the Defense Nuclear Agency under IACRO 93-817.

flow fields or density jumps without introducing nonphysical oscillations, violating conservation laws, or employing unnecessary dissipation. The resulting schemes are therefore appropriate for studying unsteady flows with little or no viscosity. The projection portion of the algorithm enforces incompressibility without the need for an artificial pressure boundary condition.

The most natural discretization for Godunov methods involves storing all velocity components at the centers of grid cells. Node-based variants are not difficult to obtain, but the requirement that all components be stored at the same points is a fairly strong one. Formulations of the projection using the staggered grid system of Harlow and Welsh [13] are thus largely incompatible with the Godunov approach. Use of collocated velocities, however, leads to unusual difference stencils for the projection which decouple adjacent grid cells.

We have developed extensions to the algorithms of [3] and [5], the most important of which are a reformulation of the methods on an adaptive hierarchy of grids, and the use of multigrid and multilevel iteration techniques to speed up computation of the projection. While we have made some attempt to keep separate the questions of how to formulate the projection versus how to solve it, there has inevitably been some interplay between these two halves of the problem. The decoupled difference stencils used by the projection in uniform parts of the grid place certain requirements on the multigrid scheme, while the need for efficient convergence of the multilevel iteration influences the choice of derivative stencils across coarse-fine grid interfaces.

These issues, concerning the formulation of the projection and its solution via multigrid methods, are the primary concern of this paper. Most of this material is new, though the need for a decoupled multigrid stencil was discussed briefly in [4]. The detailed formulation of the Godunov module, methods for error estimation and regridding, and the addition of viscous terms to the equations are all discussed in another paper, currently in preparation. These subjects will therefore be given only the most cursory attention in the present work. We will, however, describe the time-stepping procedure, so as to place the projection in its proper context as a component of the algorithm. This will be part of the general overview given in the next section. The section after that discusses the multigrid projection, while the final section presents some examples and numerical results.

OVERVIEW OF THE METHOD

The equations we are attempting to solve are the incompressible Euler equations with finite-amplitude density variation,

$$
\begin{aligned}
U_t + (U \cdot \nabla)U &= -\frac{\nabla p}{\rho}, && (1)\\
\rho_t + (U \cdot \nabla)\rho &= 0, && (2)\\
\nabla \cdot U &= 0, && (3)
\end{aligned}
$$

where U represents the velocity field, p represents the hydrodynamic pressure and ρ represents the local mass density. We will denote the x and y components of velocity by u and v, respectively.

The range of density variation in a problem may be moderate, as in the case of two or more different gases mixing in a combustion chamber, or may be relatively large, as in the 800-to-1 density jump at a water-air interface. Of course, many flows of interest do not involve density variations at all—for these problems (2) may be discarded, or similar equations may be used to advect passive quantities which do not affect the flow field. (Our implementation of the adaptive scheme currently handles only constant-density flows.) Flows with very small density variations are an intermediate case, as they may not require the full variable-density formulation. As described in [5], these flows may be modeled using what amounts to a constant-density projection method with a Boussinesq forcing term added to (1).

From a computational point of view the most problematic term in (1–3) is the pressure gradient. In contrast to the compressible case, pressure in incompressible flow plays no thermodynamic role, and cannot be determined from an equation of state. Its only function in the equations is to indirectly enforce the incompressibility constraint (3). The essential idea of projection methods is to eliminate the pressure entirely, by use of an operator which projects the velocity U onto the space of divergence-free vector fields.

The theory behind the projection operator is based on the Hodge decomposition, which provides that any vector field V can be decomposed into a divergence-free component V^d and the gradient of some scalar ϕ. This decomposition can be made unique through imposition of appropriate boundary conditions, e.g., no flow through boundaries. It is also orthogonal, since divergence and gradient are skew-adjoint with respect to the usual inner products on scalar and vector fields.

Given operators D for divergence and G for gradient, either continuous or discrete, a projection onto the space of divergence-free fields can be written as

$$\mathbf{P} = I - G(DG)^{-1}D. \tag{4}$$

(The numerical inversion of DG takes the place of solving the "pressure Poisson equation" that often appears in incompressible flow algorithms.) A modification of this projection is required for variable-density flows. We want to decompose a field into a divergence-free component and $1/\rho$ times the gradient of a scalar. The appropriate form is

$$\mathbf{P}_\sigma = I - \sigma G(D\sigma G)^{-1}D, \tag{5}$$

where $\sigma = 1/\rho$ and orthogonality is now with respect to a ρ-weighted inner product. In terms of this weighted projection, (1) can be written as

$$U_t = \mathbf{P}_\sigma\left[(-U \cdot \nabla)U\right]. \tag{6}$$

To obtain a second-order temporal discretization of this equation (and (2)), we use a fractional step process. First, the Godunov advection procedure is used to compute $(U \cdot \nabla)U$ and $(U \cdot \nabla)\rho$ at the $n + 1/2$ time level. The density equation can then be advanced immediately, while the projection is applied to $(U \cdot \nabla)U^{n+1/2}$ to give a divergence-free approximation to U_t:

$$\frac{\rho^{n+1} - \rho^n}{\Delta t} = -(U \cdot \nabla)\rho^{n+1/2}, \tag{7}$$

$$\frac{U^{n+1} - U^n}{\Delta t} = \mathbf{P}_\sigma \left[-(U \cdot \nabla)U^{n+1/2}\right]. \tag{8}$$

Since the ρ equation can be advanced first, $\rho^{n+1/2}$ is available for use in the projection. The Godunov method uses $(1/\rho)\nabla p^{n-1/2}$ to approximate the effect of the incompressibility constraint on U_t; the projection in (8) then yields an updated approximation to $(1/\rho)\nabla p^{n+1/2}$ to be used at the next time step.

We will not go into detail on the internal workings of the Godunov procedure here. Suffice it to say that using approximations to time derivatives and limited slopes (U_x, etc.) at cell centers at time n, U and ρ are extrapolated to cell edges (faces in 3D) at time $n + 1/2$. Upwinding rules resolve the choices between values coming from either side of an edge, then these edge values are differenced to yield the $(U \cdot \nabla)$ terms at cell centers at time $n + 1/2$. The detailed procedure we use is very similar to that described in [3], with the variable-density enhancements given in [5], and an improved treatment of the transverse derivative terms (vU_y, etc.) as described in [4].

For a more thorough discussion of the Hodge decomposition, the incompressible Godunov algorithm, and the time-stepping procedure, we refer the reader to [3] and [5]. These papers deal exclusively with the single-grid case, but the adaptive case requires no changes to the time-stepping method and only minimal modification to the Godunov method, e.g., interpolation into ghost cells around the edges of fine grids. An adaptive Godunov method for gas dynamics that is similar to our approach is described in [7]. We describe the adaptive projection at the end of the next section; other aspects of our adaptive incompressible algorithm will be addressed in a forthcoming paper.

MULTIGRID PROJECTION

We now discuss a multigrid algorithm for computing the variable-density projection (5). For simplicity we restrict the notation to two dimensions, but the methods presented are immediately extensible to 3D. A three-dimensional flow example is included in the following section.

Given appropriate divergence and gradient stencils, a projection of the form (5) will yield a velocity field which is discretely divergence-free to the limit imposed by roundoff error. The projection will therefore be idempotent, i.e., repeated application will not further modify the projected vector field. This is a valuable property for an unsteady flow algorithm since the projection will be applied at every time step. If $D = -G^T$ then the projection will also be orthogonal, yielding the nearest—in a ρ-weighted sense—divergence-free field.

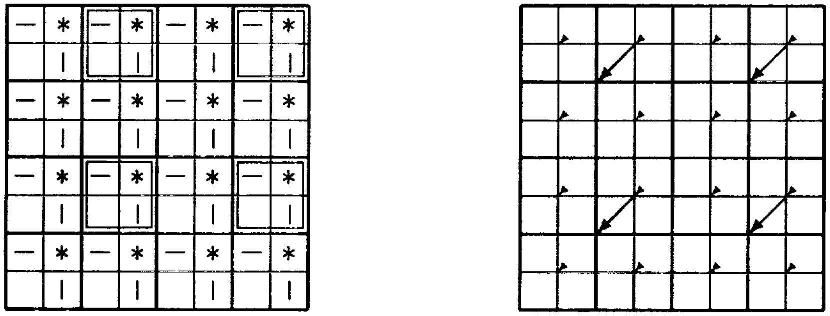

Figure 1: Decoupled grid structure: $D\sigma G\phi$ at cells marked '$*$' depends on ϕ at '$*$' cells, σ at '–' cells for x-differences, and σ at '|' cells for y-differences. Residuals from '$*$' cells are restricted by averaging to the cells marked with boxes on the next coarser grid. For purposes of restriction and interpolation, these coarse and fine values behave as if they were located at the points indicated by the arrows, rather than at the centers of their respective cells.

The simplest choice is to use centered differences for both divergence and gradient:

$$(DU)_{i,j} = \frac{1}{\Delta x}(u_{i+1,j} - u_{i-1,j}) + \frac{1}{\Delta y}(v_{i,j+1} - v_{i,j-1}), \tag{9}$$

$$(G\phi)_{i,j} = \left(\frac{1}{\Delta x}(\phi_{i+1,j} - \phi_{i-1,j}), \frac{1}{\Delta y}(\phi_{i,j+1} - \phi_{i,j-1})\right). \tag{10}$$

Composition of these then yields the elliptic stencil

$$\begin{aligned}(D\sigma G\phi)_{i,j} &= \frac{1}{(\Delta x)^2}[\sigma_{i-1,j}(\phi_{i-2,j} - \phi_{i,j}) + \sigma_{i+1,j}(\phi_{i+2,j} - \phi_{i,j})] + \\ &\quad \frac{1}{(\Delta y)^2}[\sigma_{i,j-1}(\phi_{i,j-2} - \phi_{i,j}) + \sigma_{i,j+1}(\phi_{i,j+2} - \phi_{i,j})]\end{aligned} \tag{11}$$

which appears in the projection. The main calculation we have to perform is the inversion of this expression—we have to solve $D\sigma G\phi = DV$ for ϕ given an input vector field V. Boundary conditions for ϕ are determined by those for the velocity field. Slip walls (inviscid flow) yield Neumann boundary conditions for ϕ, while in periodic problems all quantities are, naturally, periodic. Though the linear system is singular, solvability is provided by the special structure of the problem: if $D = -G^T$, then the range of G is orthogonal to the null space of D; therefore, any field in the range of D is also in the range of $D\sigma G$.

Ignoring the σ's for the moment, we see that (11) looks like a stretched version of the familiar 5-point stencil for the Laplacian. The difference is that (11) provides for no communication between adjacent grid points. Except for the effect of boundary conditions, four distinct sets of grid points participate in four distinct linear systems. Grids couple in pairs at wall boundaries, but the only local coupling comes from the smoothness of the right hand side DV. Figure 1 illustrates the decoupling pattern, including the role of the σ's.

However smooth the initial right hand side, later residuals in a multigrid scheme tend to have significant components at all wavenumbers. Multigrid depends on the fact that a solution to a coarsened system provides a good approximation to the desired fine solution. It is not surprising, therefore, that every experiment we have tried where the coarsening procedure combined components from decoupled grids proved to be wildly divergent. On the other hand, coarsening schemes which respect the decoupling lead to systems analogous to those arising from the usual 5-point Laplacian, for which multigrid is quite effective.

Let us define transformations between coarse and fine index spaces as follows,

$$\begin{aligned} I &= 2 \cdot \lfloor i/4 \rfloor + i \bmod 2, \qquad (12) \\ i &= 4 \cdot \lfloor I/2 \rfloor + I \bmod 2 \qquad (13) \end{aligned}$$

and similarly for J, j. Capitals denote indices on the coarse grid, lower case on the fine grid, and $\lfloor \; \rfloor$ reduces its argument to the next lower (or equal) integer. Each coarse point (I, J) then has four fine points associated with it: (i, j), $(i, j+2)$, $(i+2, j)$, $(i+2, j+2)$. These fine points do not appear to be quite centered around the coarse point, which would complicate restriction and interpolation formulas. We observe, however, that a centered pattern results if the points in question are each shifted to the center of their local 2x2 blocks, as illustrated in Figure 1. This shifting does not change the spatial relationship of any coupled points, even at the boundary, so for multigrid purposes we can treat each coarse point as if it were centered among its four associated fine points.

The simplest restriction formula gives a coarse cell the average of the values from its associated fine cells, while the simplest interpolation formula distributes the coarse value to each of the four fine cells (piecewise-constant interpolation). There are both theoretical results and experiments, discussed in [17], which suggest that for second-degree problems at least one of these must be replaced by a higher-order formula in order to give satisfactory convergence rates. Our own experience does not bear out this assertion. However, for difficult problems involving large density jumps we have observed an improvement in robustness from use of a bilinear stencil for interpolation,

$$\phi_{i,j} = \frac{1}{16}(9\phi_{I,J} + 3\phi_{I-2,J} + 3\phi_{I,J-2} + \phi_{I-2,J-2}) \qquad (14)$$

and similarly for $\phi_{i,j+2}$, etc. A smaller improvement resulted from the opposite choice, bilinear restriction with piecewise-constant interpolation. Problems without difficult density configurations did not show a consistent improvement in convergence rate with either stencil. We use (14) routinely in our variable-density code, but use the piecewise-constant formula in the constant-density adaptive code. Restriction is by simple averaging in both cases.

We have now satisfactorily dealt with the decoupling problem for ϕ, but what about σ, i.e., how to we form the elliptic stencil on coarser grids? It is apparent from Figure 1 that σ values do not occupy the same decoupled component of the grid as ϕ and the residuals. Moreover, σ values used for x-differences are on a different component from those used for y-differences.

One possibility is to redefine the problem to place σ's at the same points as ϕ's:

$$\begin{aligned}(D\sigma G\phi)_{i,j} = {} & \frac{1}{2(\Delta x)^2}[(\sigma_{i-2,j}+\sigma_{i,j})(\phi_{i-2,j}-\phi_{i,j})+(\sigma_{i+2,j}+\sigma_{i,j})(\phi_{i+2,j}-\phi_{i,j})]+ \\ & \frac{1}{2(\Delta y)^2}[(\sigma_{i,j-2}+\sigma_{i,j})(\phi_{i,j-2}-\phi_{i,j})+(\sigma_{i,j+2}+\sigma_{i,j})(\phi_{i,j+2}-\phi_{i,j})]. \quad (15)\end{aligned}$$

The hope is that σ could be coarsened by averaging over associated cells, just as ϕ is. Unfortunately, this scheme gives somewhat degraded accuracy, and more importantly, horrible multigrid convergence rates for problems with large density variations.

The convergence rate of the multigrid cycle seems more strongly dependent on the proper coarsening pattern for σ than on any other single feature of the method. The following procedure is in fact the *only* scheme we have tried that gave anything approaching satisfactory results. We keep two different arrays of σ values on coarser grids, one for x-differences and one for y-differences. These are coarsened as follows:

$$\begin{aligned}\sigma^x_{I,J} &= \frac{1}{2}(\sigma^x_{i',j}+\sigma^x_{i',j+2}), \\ \sigma^y_{I,J} &= \frac{1}{2}(\sigma^y_{i,j'}+\sigma^y_{i+2,j'}), \qquad (16)\end{aligned}$$

where $i' = 2I + I \bmod 2$, $j' = 2J + J \bmod 2$ and $\sigma^x = \sigma^y = \sigma$ on the fine grid. Coarse stencils based on (11) and formed with these values perform well even in the presence of sharp density interfaces. They only begin to fail when presented with such nonphysical effects as large sawtooth variations in the density field.

One common approach to deriving coarse grid equations is to use the form RAP, where R is the restriction operator, A is the elliptic stencil, and P is the interpolation operator. Unfortunately, this approach does not give a usable stencil when applied with piecewise-constant formulas for R and P, and higher-order transfer stencils give rise to larger, more complicated coarse grid operators. Use of (16) can be motivated in two ways, however. First, patterns like this one do appear in the RAP stencils, even though those formulas have other drawbacks. Second, if we confine our attention to one decoupled component of the grid, the σ locations can be interpreted as the edges between its cells. An analogy to a diffusion problem with ϕ as heat content and σ as conductivity then suggests an averaging along edges equivalent to (16).

A detailed discussion of multigrid for problems with difficult coefficients can be found in [1]. Our approach seems adequate for configurations likely to arise in practical projection problems, however, and the authors of [1] acknowledge certain pathological cases where even their more complicated schemes will fail.

For our multigrid schedule we use the pattern called FMV in [8]—the F-cycle in [17]—with smoothing by point Gauss-Seidel. Two smoothing steps before each grid transfer operation, up or down, seems to give the best performance. In problems with large density variations the Gauss-Seidel method alone does not give rapid convergence on the coarsest grid, so we have replaced it at that level with an exact solver. A direct method could be used here, but we have found it more convenient to employ

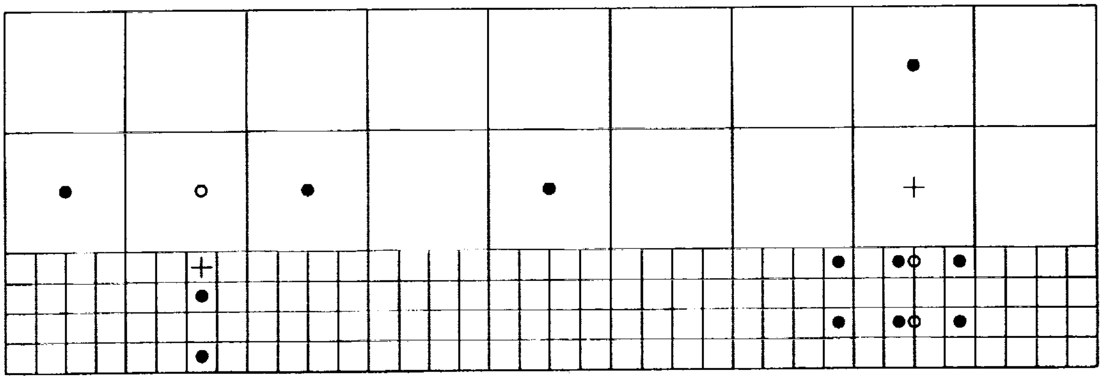

Figure 2: Examples of decoupled derivative stencils across a coarse-fine interface. The crosses indicate a fine cell (left) and a coarse cell (right) at which y-derivatives are evaluated. Bullets show which cells participate in the stencils. In each case, values on the opposite side of the interface are interpolated in the transverse direction to the circled points, giving three values on a line normal to the interface from which the derivative can be computed.

a simple diagonally-preconditioned conjugate gradient method based on algorithm 10.3-1 and equation 10.3-3 from [12]. The conjugate gradient approach has the advantage in that it neither requires explicit storage of a matrix, nor special treatment of the singular linear system.

This completes our description of the variable-density multigrid projection. One variation should be noted in passing. To reformulate the 2D projection in cylindrical (r-z) coordinates, it suffices to redefine σ as x/ρ, where $x = r$ becomes the radial coordinate. No other change is required in the projection portion of the algorithm.

– – –

An adaptive version of the projection method can be described, at least roughly, in terms of a few relatively minor additions to the single-grid algorithm. The details of the implementation, however, are considerably more complicated, and we only have a working program for the 2D constant-density flow case. Our purpose here is not to give a step-by-step breakdown of the entire adaptive procedure, but rather to highlight the ways in which a decoupled Laplacian stencil affects the multilevel projection calculation. For the sake of brevity, we have decided not to burden this discussion with explicit formulas—we trust that all necessary expressions can be easily derived from the descriptions given in the text.

The structure of the grid hierarchy is similar to that used in [7]. A single rectangular grid covers the entire computational domain at the coarsest level. In "interesting" regions of the flow, finer grid patches are laid down, refined from the coarse level by a fixed ratio r. These finer grids are themselves rectangular, both to minimize program overhead and to improve performance on vector architectures. If necessary, more levels of grids can be created, but we impose a "proper-nesting" requirement that each refined level l have a border of cells at level $l-1$ separating it from still coarser levels. The simplest choice for a refinement ratio is 2, but we often use 4 instead in order to reduce both the number of refined levels and the amount of wasted storage allocated to coarse grids underlying fine grids.

In contrast to approaches like that of [15], we have maintained a logical separation between the multilevel iteration for the adaptive scheme, and the multigrid solvers on individual grids. Our multilevel iteration proceeds as follows, where we assume familiarity with the residual-correction formulation discussed in [8] and [17]:

- Start with an initial approximation to ϕ, either 0 or the value obtained at the previous time step.
- Repeat until residuals satisfy tolerance:
    - Compute residual on all grids, including coarse-fine interfaces.
    - Restrict residuals from fine to coarse grids.
    - Set correction array to 0 at coarse level.
    - For each level l, from coarse to fine, do:
        - Execute FMV cycle for residual equation on each grid of level l, using values from adjacent grids as boundary conditions if necessary.
        - Add correction into ϕ at level l.
        - Interpolate correction to next finer level, if any.

The convergence properties of this method depend on a coarse grid solution being a satisfactory approximation to the solution on the composite grid. In order for this to be the case, all interpolation, restriction, and difference stencils have to respect the decoupling pattern. For the grid transfer operations, these formulas are like those we have already discussed. Restriction is by simple averaging of associated cells. For interpolation we have had best results with a higher-order method, a biquadratic formula using coarse cells from the appropriate decoupled grid component. Unlike the single-grid case, effective position shifts like those shown in Figure 1 are no longer valid, so we use the actual positions of cell centers to derive the interpolation stencil.

Difference formulas across the grid interfaces are more problematic. Whereas restriction and interpolation schemes affect only the convergence rate of the iteration, the difference stencils determine the actual converged solution. Stencil outlines for both fine and coarse points near the interface are shown in Figure 2. In both cases we use quadratic interpolation to obtain third-order accurate values on the opposite side of the interface, then a three-point difference formula to give a second-order accurate derivative at the desired point. Composition of second-order derivatives in D and G gives a Laplacian approximation that is first-order accurate along the interface, sufficient for global second-order accuracy of the projected velocity field.

These derivative stencils are used for computing residuals and for obtaining divergence and gradient in the projection formula. Note that D is no longer equal to

	32	64	128	256
1	3.54926 8	0.907518 (3.91) 7	0.228655 (3.97) 7	0.0573795 (3.98) 7
2	4.16279 8	1.14104 (3.65) 7	0.293781 (3.88) 7	0.074023 (3.97) 7
3	9.84036 13	4.7626 (2.07) 9	2.14014 (2.23) 11	0.876724 (2.44) 18
4	1.29866 19	0.378418 (3.43) 16	0.097241 (3.89) 15	0.024058 (4.04) 14
5	7.26845 19	1.84558 (3.94) 20	0.476259 (3.88) 21	0.123554 (3.85) 22
	0.802074	0.196431 (4.08)	0.0487401 (4.03)	0.0121474 (4.01)

Table 1: Convergence results for both variable-density and adaptive implementations of the decoupled projection. For each case the problem was run with square base grids of four different sizes—32x32 through 256x256—to a final residual less than 10^{-10}. The numbers given for each run are the final ∞-norm error in the velocity field (times 1000), the factor of improvement from the next coarser grid, and the number of multigrid cycles required. For the last run (adaptive code), 2-norm error data is also given. A description of each problem is given in the text.

$-G^T$. This means that the adaptive projection is no longer quite orthogonal, and we have to add a slight correction to DV to make the system solvable. The alternative, however, would be to use less accurate stencils for either D or G at the interface, which would seriously degrade the performance of the algorithm.

NUMERICAL EXAMPLES

Table 1 summarizes the convergence behavior of the projection for five different problems. The domain is the unit square with no flow through the boundaries. In each case we start with the divergence-free vector field

$$\begin{aligned} u &= (+0.2)(x+1)(\pi(y+1)\cos\pi y + \sin\pi y)\sin\pi x, \\ v &= (-0.2)(y+1)(\pi(x+1)\cos\pi x + \sin\pi x)\sin\pi y, \end{aligned} \tag{17}$$

add to it $1/\rho$ times the gradient of

$$\phi = \frac{-1}{\pi}\cos\left(\frac{\pi}{2}(x+x^3)\right)\cos\pi y, \tag{18}$$

then apply the projection. This should strip off the gradient portion of each field and return the divergence-free portion (17). The five cases considered are: (1) constant density, (2) mild density variation—$\rho = 1 + 100\sin^2\pi x\sin^2\pi y$, (3) extreme density variation—$\rho = 1 + 100000\sin^2\pi x\sin^2\pi y$, (4) discontinuous jump in density—$\rho = 1$ inside a radius 0.1 circle centered at (0.4, 0.4), $\rho = 10001$ elsewhere, (5) constant density adaptive—the square from 0.25 to 0.75 in x and y is refined by a factor of four from the base grid.

Cases (1) and (2) are smooth, so the multigrid scheme converges rapidly and gives unambiguous second-order convergence. Cases (3) and (4) are more difficult, but the scheme is still clearly better than first order. In the adaptive case (5) the errors are concentrated along the coarse-fine grid interface, where the discretization of DG is only first-order accurate. Convergence is still second-order in the 2-norm, but may be slightly degraded in the ∞-norm. Note that this example is not representative of the intended use of the adaptive method. In normal operation the interfaces are well-separated from complicated regions of the flow field, which dominate the error behavior of the scheme. Slower convergence for the adaptive scheme appears to be due the mismatch between coarse grid stencils and the residuals computed at the interface. Relaxation at interfaces and/or closer integration of the multigrid and multilevel iterations might yield a faster algorithm.

Quantitative analysis of the the flow solver as a whole is beyond the scope of this paper. Our remaining two examples are intended mainly as illustrations, to demonstrate the power of the algorithm for modeling unsteady flow fields with finely detailed structure. In Figure 3 we show an image from a 3D variable-density calculation set up and run by Dan Marcus. A bubble of helium was initially started at rest near the bottom of the domain. The ambient fluid is air, giving a density ratio of 7.25. The calculation was performed on a 64x64x128 grid occupying one quarter of the volume shown—this was filled out to 128^3 for rendering by reflection through the two symmetry planes. At the time of the picture the bubble has risen and developed into a torus, with more complicated flow patterns visible in the outer mixed region. We do not claim that this calculation accurately models a turbulent flow field. However, a more detailed examination of transition to turbulence, using a projection method similar to the one presented here, can be found in [6].

Figure 4 illustrates the adaptive algorithm. A 64x64 base grid is refined twice, by a factor of four each time, so the finest level has resolution equivalent to a single 1024x1024 grid. Every 10 time steps grids are re-allocated according to a procedure based on second derivatives of the velocity field. In the initial conditions, four patches of vorticity with radii 0.025 are placed in the unit square at (0.5, 0.5), (0.5, 0.575), and the two 120° rotations of this position. Each patch has uniform vorticity except for a linear ramp 3/256 wide down to zero vorticity at the edge—the radius of the patch is the distance from the center to the halfway point of the ramp. The initial velocity field is obtained by solving for the stream function associated with the given vorticity field. This is identical to the projection calculation, except that the stream function satisfies a Dirichlet boundary condition. Note how well the Godunov advection scheme preserves fine details of the flow field, even in the highly stretched regions near the vortex core.

CONCLUSIONS AND FUTURE PLANS

Centered difference stencils are the simplest choice for implementing the discrete divergence and gradient, subject to the requirement that velocity components must all be defined at the same points. The decoupled projection stencils arising from this choice require various contortions in the solution algorithm, which raises doubts as to

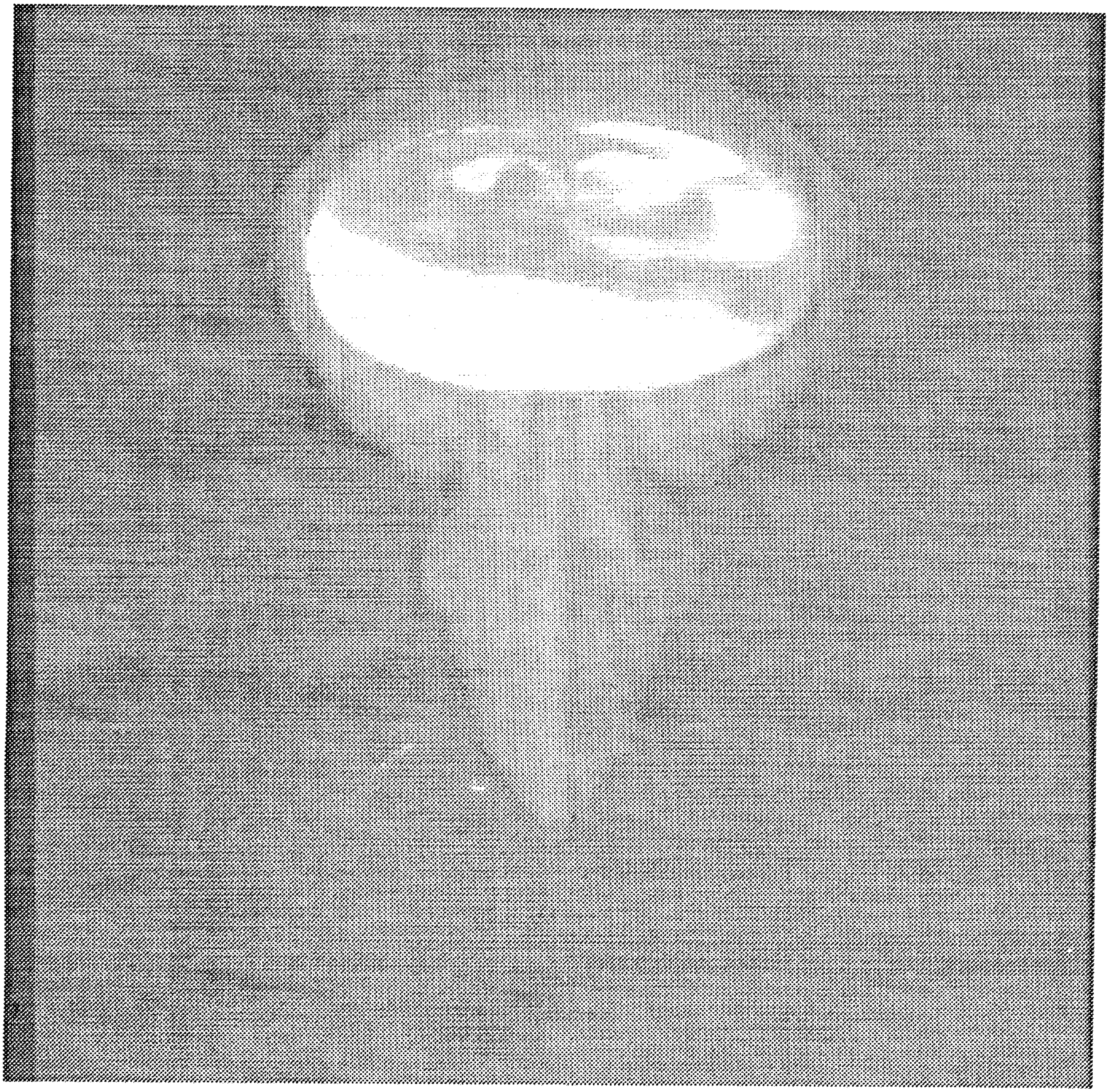

Figure 3: Volume-rendering of a helium bubble rising through air. The central part of the bubble has taken on a simple toroidal shape, but the outlying mixed regions show more complicated flow patterns.

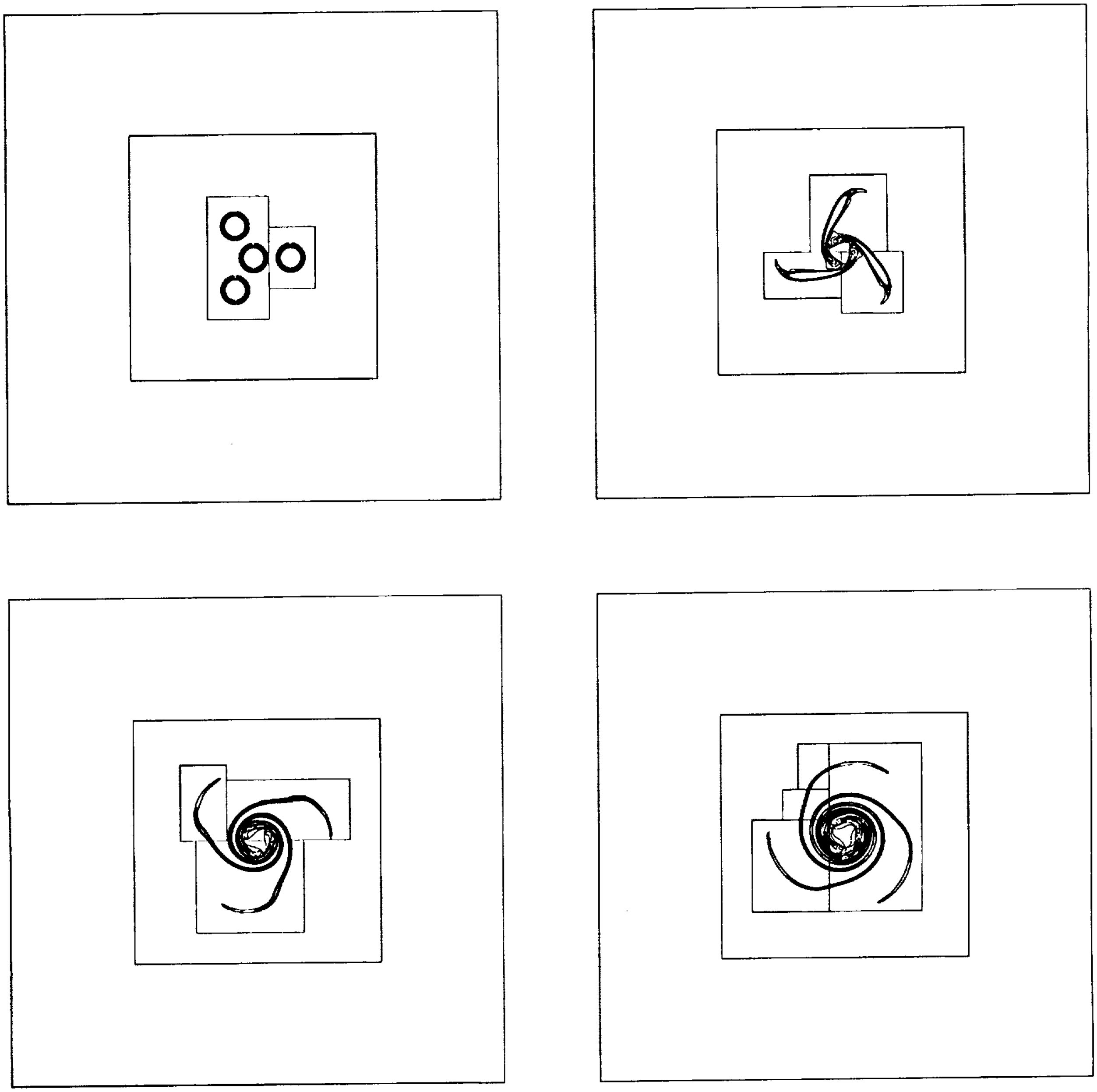

Figure 4: Adaptive simulation of a four-way vortex merger problem, showing contours of vorticity.

the practical utility of the results thus obtained. Despite the unusual behavior of the projection, however, the difficulties have been overcome and the method successfully models a variety of incompressible flow problems.

It seems likely that some flow problems will not be suitable for this type of algorithm. Though the projection does not directly cause high-wavenumber instabilities, neither does it do anything to suppress them when they are excited by other parts of a flow solver. Lai, for example, reports having difficulty using this type of projection for certain combustion problems [14]. We have seen stability problems ourselves in an adaptive version of the algorithm of [4], where a staggered-mesh projection is applied to the edge velocities computed in the Godunov predictor.

While we believe the decoupled method is a worthy contender, these difficulties beg for comparative studies with other types of projections. One alternative is the regularization given by Strikwerda [16]. Though coupled, however, the stencils derived in this work are both large and asymmetrical. A newer approach is that of Almgren, Bell and Szymczak in [2], which is coupled and symmetrical but not quite idempotent. We have recently completed an adaptive version of this projection, early results from which seem quite promising.

ACKNOWLEDGEMENTS

The author wishes to thank John Bell for his help with the design of the Godunov module, Dan Simkins for writing the streamlined Godunov integrator for 3D, and Dan Marcus for supplying the data from his helium bubble calculation.

REFERENCES

[1] R. E. Alcouffe, A. Brandt, J. E. Dendy, and J. W. Painter. The multi-grid method for the diffusion equation with strongly discontinuous coefficients. *SIAM J. Sci. Stat. Comput.*, 2:430–454, 1981.

[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. Technical Report UCRL-JC-112842, LLNL, January 1993.

[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] J. B. Bell and D. L. Marcus. A second-order projection method for variable-density flows. *J. Comput. Phys.*, 101:334–348, 1992.

[6] J. B. Bell and D. L. Marcus. Vorticity intensification and transition to turbulence in the three-dimensional Euler equations. *Commun. Math. Phys.*, 147:371–394, 1992.

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

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

[9] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comput.*, 22:742–762, October 1968.

[10] A. J. Chorin. On the convergence of discrete approximations to the Navier-Stokes equations. *Math. Comput.*, 23:341–353, 1969.

[11] P. Colella. A multidimensional second order Godunov scheme for conservation laws. *J. Comput. Phys.*, 87:171–200, 1990.

[12] G. H. Golub and C. F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, 1983.

[13] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluids with free surfaces. *Physics of Fluids*, 8:2182–2189, 1965.

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

[15] S. F. McCormick. *Multilevel Adaptive Methods for Partial Differential Equations.* SIAM, Philadelphia, 1989.

[16] J. C. Strikwerda. Finite difference methods for the Stokes and Navier-Stokes equations. *SIAM J. Sci. Stat. Comput.*, 5:56–67, 1984.

[17] P. Wesseling. *An Introduction to Multigrid Methods.* Wiley, New York, 1992.