

# A Second-Order Projection Method for the Incompressible Navier–Stokes Equations

JOHN B. BELL\* AND PHILLIP COLELLA\*

*Lawrence Livermore National Laboratory, Livermore, California 94550*

AND

HARLAND M. GLAZ†

*University of Maryland College Park, Maryland 20742*

Received March 16, 1988; revised November 21, 1988

In this paper we describe a second-order projection method for the time-dependent, incompressible Navier–Stokes equations. As in the original projection method developed by Chorin, we first solve diffusion-convection equations to predict intermediate velocities which are then projected onto the space of divergence-free vector fields. By introducing more coupling between the diffusion-convection step and the projection step we obtain a temporal discretization that is second-order accurate. Our treatment of the diffusion-convection step uses a specialized higher order Godunov method for differencing the nonlinear convective terms that provides a robust treatment of these terms at high Reynolds number. The Godunov procedure is second-order accurate for smooth flow and remains stable for discontinuous initial data, even in the zero-viscosity limit. We approximate the projection directly using a Galerkin procedure that uses a local basis for discretely divergence-free vector fields. Numerical results are presented validating the convergence properties of the method. We also apply the method to doubly periodic shear-layers to assess the performance of the method on more difficult applications © 1989 Academic Press, Inc.

## 1. INTRODUCTION

In this paper we develop a second-order projection method for the incompressible Navier–Stokes equations

$$U_t + (U \cdot \nabla) U = \varepsilon \Delta U - \nabla p + F \tag{1.1}$$

$$\nabla \cdot U = 0 \tag{1.2}$$

The submitted manuscript has been authored by a contractor of the U.S. Government under contract number W-7405-ENG-48. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

\* This work was performed, in part, under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract W-7405-Eng-48. Partial support under Contract W-7405-Eng-48 was provided by the Applied Mathematical Sciences Program of the Office of Energy Research and by the Air Force Office of Scientific Research under Grant AFOSR-ISSA-870016.

† The work of this author was partially supported by the National Science Foundation under Grant DMS-87-3971.

on a domain  $\Omega$ , where  $U$  represents the velocity field,  $p$  represents the hydrodynamic pressure, and  $F$  represents any external forces. We denote the  $x$  and  $y$  components of velocity by  $u$  and  $v$ , respectively. Typical initial and boundary conditions for (1.1)–(1.2) involve specifying an initial velocity field and specifying Dirichlet or Neumann conditions for velocity; no boundary conditions are required for pressure. Orthogonality of the pressure gradient with divergence-free vector fields effectively eliminates pressure from the system while enforcing (1.2); in fact, specifying pressure boundary conditions overdetermines the system. (See Ladyzhenskaya [1] and Fujita and Kato [2] for a discussion of the theory.) This orthogonality property is reflected in the Hodge decomposition which states, in its simplest form, that a vector field  $V$  can be uniquely decomposed into a divergence-free component  $V^d$  that satisfies  $V^d \cdot n = 0$  and the gradient of some scalar  $\phi$ . Furthermore, if we integrate by parts we find that

$$\int_{\Omega} V^d \cdot \nabla \phi \, dx \, dy = 0.$$

Consequently, the Hodge decomposition defines, by means of suitable density arguments, an orthogonal projection  $\mathbf{P}$  on  $L^2$  such that  $\mathbf{P}V = V^d$ . (See Temam [3] for a more detailed description of the structure of the projection.) By using this projection we can interpret (1.1)–(1.2) as an evolution equation for velocity within the space of divergence-free vector fields.

One of the central issues in the design of numerical methods for the Navier–Stokes equations is the development of an appropriate discrete form of the incompressibility constraint (1.2). The first primitive-variable numerical method, developed by Harlow and Welch [4], attempts to enforce the incompressibility constraint (1.2) by deriving a Poisson equation for the pressure, taking the divergence of (1.1). Unfortunately, since there are no physical boundary conditions for pressure, some type of artificial boundary condition is required. The subsequent literature contains considerable discussion of possible pressure boundary conditions. Gresho and Sani [5] provide a critical survey of these issues and discuss the relationship of different pressure boundary conditions to the original system of equations.

Krzywicki and Ladyzhenskaya [6] proposed a difference scheme that more closely reflects the analytic structure of the equations. In essence they show that by simultaneously discretizing (1.1) and (1.2) they obtain a method that converges to weak solutions of the Navier–Stokes equations while avoiding any artificial pressure boundary conditions. Chorin, in a series of papers [7–9], developed a practical numerical method based on a discrete form of the Hodge decomposition. This method, known as the projection method, computes an intermediate vector field that is then projected onto divergence-free fields to recover the velocity in (1.1). The calculation of the discrete Hodge projection involves discretizing (1.2) and using the resulting discrete operator and its adjoint to form the projection. The accuracy with which the divergence-free condition is represented depends only on

the truncation error in the approximation to (1.2). In particular, the determination of an explicit pressure boundary condition is not required. Similar ideas were also discussed by Temam [10]. Chorin's error estimate [8] shows that his method is  $O(\Delta t + \Delta x^2)$  for a periodic box in two and three dimensions. Since the method is only first-order accurate in time, it requires a fairly restrictive time step to achieve acceptable accuracy. Furthermore, the situation is considerably complicated by the presence of boundaries. This complication arises because the projection does not commute with the Laplacian appearing in the viscous terms in the presence of boundaries. Chorin analyzes a version of his scheme that introduces a first-order, local error at the boundaries by imposing homogeneous boundary data for the intermediate field. For this modified scheme he proves  $O(\Delta t^{1/4})$  accuracy. (The heuristic arguments in Section 2 as well as computational evidence suggest that the modified scheme is, in fact, first-order accurate in time for velocity.)

Chorin, in [7, 8], also introduces a more practical version of his scheme for problems with boundaries. This version uses an inhomogeneous boundary condition for the intermediate velocity field. Kim and Moin [11] use this version of Chorin's algorithm, replacing the treatment of the nonlinear terms with a second-order explicit Adams–Bashforth scheme and using the staggered grid system of Harlow and Welch [4]. They provide computational evidence that their scheme is second-order accurate. Van Kan [12] proposes another second-order generalization of the projection method. His method is based on first discretizing the spatial terms using the Harlow and Welch staggered grid. This reduces the partial differential equations to a system of differential algebraic equations. He then develops a second-order integration technique of projection-type for this system. Van Kan's analysis assumes fixed bounds on the operators that represent the spatial discretization of the differential equations so that his analysis is not valid if the spatial and temporal discretizations are simultaneously refined.

In this paper we present a new projection method that is second-order accurate in space and time provided  $\Delta t = O(\Delta x, \Delta y)$ . This method is motivated by a desire to apply higher order upwind methods developed for inviscid, compressible flow to the incompressible Navier–Stokes equations. In particular, the method incorporates a specialized version of the unsplit second-order Godunov methodology introduced for gas dynamics by Colella [13]. The Godunov method provides a robust discretization of the convective terms that avoids any cell Reynolds number stability restriction for high Reynolds number flow. The overall algorithm is based on a time-step procedure that is similar to van Kan's method although it is motivated by different considerations. (Van Kan's time-stepping strategy is not directly applicable because of some technical incompatibilities with the Godunov procedure.)

For the description of the algorithm in this paper, we will restrict our attention to homogeneous Dirichlet boundary conditions and assume that there are no external forces. We will also assume that the mesh spacing is uniform in the  $x$  and  $y$  directions. These restrictions are not inherent limitations of the method; they have been adopted here for clarity of exposition. In the next section we discuss the time-

stepping procedure in a semi-discrete setting and provide a heuristic argument that the semidiscrete form is second-order in time under suitable assumptions. In Section 3 we describe the approximation of the nonlinear diffusion-convection equation (1.1). The diffusion terms are discretized using a standard cell-centered difference approximation. The nonlinear convection terms are treated using an explicit second-order Godunov method based on ideas similar to those discussed in Colella [13], van Leer [14], and Bell, Dawson, and Shubin [15]. The method provides a second-order discretization for smooth flow and selectively introduces dissipation near discontinuities by means of a “monotonized” slope computation. This avoids any cell Reynolds number restrictions and allows the method to be applied to the incompressible Euler equations without introducing spurious oscillations or instabilities, even for discontinuous data. The description of the method is completed in Section 4 where we discuss the numerical approximation of the projection  $\mathbf{P}$ . The discrete form of the projection is based on a finite-difference Galerkin formalism introduced for the steady state Navier–Stokes equations by Stephens *et al.* [16]. The fifth section of the paper contains numerical results. The first set of results validates the second-order convergence of the method when applied to smooth flow for Stokes flow, Reynolds number 100, and the incompressible Euler equations. The method is then applied to the study of doubly periodic shear layers. These computational results serve to demonstrate the capabilities of the method on more difficult problems.

## 2. TEMPORAL DISCRETIZATION

In this section we describe, in a semi-discrete form, the time stepping strategy for the projection method. Before discussing the semi-discrete form, we first review the basic analytic underpinning of projection methods. The projection form of the equations is suggested by reorganizing the terms in (1.1) to obtain

$$U_t + \nabla p = \varepsilon \Delta U - (U \cdot \nabla) U. \quad (2.1)$$

If we assume temporal smoothness, (1.2) implies that  $U_t$  is divergence free. Therefore, if we apply the projection  $\mathbf{P}$  to (2.1) we find that

$$U_t = \mathbf{P}(\varepsilon \Delta U - (U \cdot \nabla) U). \quad (2.2)$$

In this form, we have completely eliminated the pressure from the equations. The absence of initial or boundary conditions for pressure is not surprising since the mathematical theory of the Navier–Stokes equations says that they are not needed; in fact, specifying initial or boundary data for pressure overdetermines the problem.

As in the original projection algorithm, the numerical form of the projection method involves a two-step process in which an auxiliary vector field is computed

and then some type of discrete projection is applied. In the original algorithm [8] the auxiliary vector field is a linear combination of the velocity at the new time level and the gradient of the pressure. In van Kan's algorithm [12], the auxiliary field contains a pressure update rather than the pressure itself. Although operationally similar to van Kan's approach in the sense that our time discretization can be interpreted as a pressure correction, our time-stepping strategy is actually based on an iterative procedure that converges to a modified Crank–Nicholson scheme. The modification arises because we assume, for the purposes of the temporal discretization, that the nonlinear convection terms at the half time-level are explicitly computable to second-order accuracy from velocity data at  $t^n$  and the current approximation to  $\nabla p$ ; they are *not* the average of the convective terms evaluated at  $t^n$  and  $t^{n+1}$ . (See Section 3 for a discussion of this procedure.)

Ideally, what we would like to compute is

$$\frac{U^{n+1} - U^n}{\Delta t} = \mathbf{P} \left( \frac{\varepsilon}{2} \Delta(U^n + U^{n+1}) - [(U \cdot \nabla) U]^{n+1/2} \right) \quad (2.3)$$

or, equivalently,

$$\frac{U^{n+1} - U^n}{\Delta t} + \nabla p^{n+1/2} = \frac{\varepsilon}{2} \Delta(U^n + U^{n+1}) - [(U \cdot \nabla) U]^{n+1/2} \quad (2.3a)$$

$$\nabla \cdot U^{n+1} = 0. \quad (2.3b)$$

Equation (2.3) is a Crank–Nicholson approximation to (2.2) which is clearly second-order accurate in time. While it is possible to actually compute this directly using the discrete Galerkin ideas discussed in Section 4, the resulting linear system that must be solved is poorly conditioned. Instead, we have adopted an iterative strategy that, from a linear algebraic standpoint, involves the solution of symmetric, positive-definite systems corresponding to discretizations of second-order elliptic operators.

The iterative procedure describes an iteration for determining the gradient part of the right-hand side of (2.3a), namely,

$$(\mathbf{I} - \mathbf{P}) \left( \frac{\varepsilon}{2} \Delta(U^n + U^{n+1}) - [(U \cdot \nabla) U]^{n+1/2} \right) = \nabla p^{n+1/2}. \quad (2.4)$$

If we denote by  $\nabla p^{n+1/2,k}$  the  $k$ th iterate approximating (2.4), we then define  $U^{*,k}$  by

$$\frac{U^{*,k} - U^n}{\Delta t} = \frac{\varepsilon}{2} \Delta(U^n + U^{*,k}) - [(U \cdot \nabla) U]^{n+1/2} - \nabla p^{n+1/2,k}. \quad (2.5)$$

Once we have computed  $U^{*,k}$ , we obtain the next iterate for (2.4) by evaluating the right-hand side of (2.3a) with  $U^{n+1}$  replaced by  $U^{*,k}$ . We now apply the projection

to separate this vector field into an update of the approximations of  $U_i$  and  $\nabla p^{n+1/2}$ . More precisely, we update the iteration by computing

$$\frac{U^{n+1,k+1} - U^n}{\Delta t} + \nabla p^{n+1/2,k+1} = \frac{\varepsilon}{2} \Delta(U^n + U^{*,k}) - [(U \cdot \nabla) U]^{n+1/2}, \quad (2.6)$$

where  $U^{n+1,k+1}$  is divergence free; or, equivalently

$$\frac{U^{n+1,k+1} - U^n}{\Delta t} = \mathbf{P} \left( \frac{\varepsilon}{2} \Delta(U^n + U^{*,k}) - [(U \cdot \nabla) U]^{n+1/2} \right) \quad (2.6a)$$

$$\nabla p^{n+1/2,k+1} = (\mathbf{I} - \mathbf{P}) \left( \frac{\varepsilon}{2} \Delta(U^n + U^{*,k}) - [(U \cdot \nabla) U]^{n+1/2} \right). \quad (2.6b)$$

We next want to examine the behavior of the iteration scheme defined by (2.5) and (2.6). The basic approach is to compare the iterates to the Crank–Nicholson solution defined by (2.3). We first discuss the convergence of the iteration scheme for one time step and show that the iterates converge to a single step of the Crank–Nicholson scheme (2.3). We then discuss the global properties associated with using a single step of the iteration for each time step. Our discussion will be essentially a formal one. In particular, we will not address issues such as smoothness and other considerations needed to construct actual proofs.

Before discussing the convergence properties in detail, we first state our assumptions on the Godunov procedure used to compute  $[(U \cdot \nabla) U]^{n+1/2}$ . This is complicated by the need for a projection in the predictor step of the Godunov procedure. However, since the required projection need only be approximated to first-order accuracy, we avoid the computational expense of a second projection by approximating its effect using the computed pressure at  $t^{n+1/2}$ . For the purposes of discussing the temporal accuracy of the time-step strategy we assume that the Godunov procedure satisfies the estimate

$$\|[(U \cdot \nabla) U]^{n+1/2} - [(V \cdot \nabla) V]^{n+1/2}\| \leq c \|U^n - V^n\| + c \Delta t \|\nabla p^{n-1/2} - \nabla q^{n-1/2}\|, \quad (2.7)$$

where  $\nabla p$  and  $\nabla q$  are the pressures associated with  $U$  and  $V$ , respectively. A straightforward truncation error analysis of the method presented in the next section can be used to verify the required formal error estimate.

We now compare the behavior of the iterates defined by (2.5), (2.6) with the result of one step of Crank–Nicholson. We use the same initial data for each scheme which we denote by  $U^n$  and  $\nabla p^{n-1/2}$  and initialize the pressure iteration with  $\nabla p^{n+1/2,0} = \nabla p^{n-1/2}$ . Combining (2.3a) and (2.5) yields

$$U^{n+1} - U^{*,k} = \Delta t \left( I - \frac{\varepsilon \Delta t}{2} \Delta \right)^{-1} (\nabla p^{n+1/2,k} - \nabla p^{n+1/2}), \quad (2.8)$$

where  $U^{n+1}$  is the solution to (2.3). If we now combine (2.3a) and (2.6) and use (2.8) to eliminate  $U^{n+1} - U^{*,k}$ , we obtain

$$\begin{aligned} & \frac{U^{n+1} - U^{n+1,k+1}}{\Delta t} + \nabla p^{n+1/2} - \nabla p^{n+1/2,k+1} \\ &= \frac{\varepsilon \Delta t}{2} \Delta \left( I - \frac{\varepsilon \Delta t}{2} \Delta \right)^{-1} (\nabla p^{n+1/2,k} - \nabla p^{n+1/2}). \end{aligned} \tag{2.9}$$

If we apply the projection to remove the divergence-free component of (2.9), we obtain

$$\nabla p^{n+1/2} - \nabla p^{n+1/2,k+1} = (\mathbf{I} - \mathbf{P}) \frac{\varepsilon \Delta t}{2} \Delta \left( I - \frac{\varepsilon \Delta t}{2} \Delta \right)^{-1} (\nabla p^{n+1/2,k} - \nabla p^{n+1/2}). \tag{2.10}$$

The convergence properties of the iteration now depend on the operator on the right-hand side of (2.10). However, this operator is bounded with norm one and it is contractive so that  $\nabla p^{n+1/2,k} \rightarrow \nabla p^{n+1/2}$  as  $k \rightarrow \infty$ . (When discretized, the numerical approximation of the operator in (2.10) is a contraction mapping with norm strictly less than one.)

Although the iteration defined by (2.5) and (2.6) converges to a second-order temporal discretization, we do not want to iterate to convergence at each step. Computational evidence presented in Section 5 demonstrates that the method is second-order accurate with only one step of the iterative scheme for each time step. Standard truncation error analysis is not suitable for studying the behavior of the method because it fails to properly address potential accumulation of error in the pressure approximation. Instead, we will compare the solution obtained using (2.5)–(2.6) with the Crank–Nicholson solution. This approach is similar to van Kan’s [12]; however, we will not assume Lipschitz continuity of the differential operators so that our analysis remains valid if we refine spatially as well as temporally. In the remainder of this section we will show that, in the absence of boundaries, one step of the iteration scheme described above is sufficient to obtain second-order temporal accuracy for velocity, provided the solution is sufficiently smooth. We will also indicate where the difficulties in the analysis arise in the presence of boundaries.

For this discussion we let  $V^n$  and  $\nabla q^{n+1/2}$  denote the approximate solution obtained using the Crank–Nicholson equations (2.3). We also drop superfluous superscripts letting  $U^n$  represent  $U^{n,1}$ ,  $U^*$  represent  $U^{*,1}$ , and  $\nabla p^{n+1/2}$  represent  $\nabla p^{n+1/2,1}$  in (2.5) and (2.6). When using one step of the iteration we initialize the pressure at  $n + 1/2$  with the final approximation at  $n - 1/2$ . Thus,  $\nabla p^{n+1/2,0} = \nabla p^{n-1/2}$ .

Using these definitions and combining (2.5) and (2.6) we can show that

$$\begin{aligned} & \frac{U^{n+1} - U^n}{\Delta t} + \nabla p^{n+1/2} \\ &= \frac{\varepsilon}{2} \Delta (U^n + U^{n+1}) - [(U \cdot \nabla) U]^{n+1/2} + \frac{\varepsilon \Delta t}{2} \Delta (\nabla p^{n+1/2} - \nabla p^{n-1/2}). \end{aligned} \tag{2.11}$$

If we now let  $\eta^n = V^n - U^n$  and  $\nabla\delta^{n+1/2} = \nabla q^{n+1/2} - \nabla p^{n+1/2}$ , we can derive an error equation of the form

$$\begin{aligned} \frac{\eta^{n+1} - \eta^n}{\Delta t} + \nabla\delta^{n+1/2} &= \frac{\varepsilon}{2} \Delta(\eta^n + \eta^{n+1}) - ([U \cdot \nabla] U)^{n+1/2} - [(V \cdot \nabla) V]^{n+1/2} \\ &\quad + \frac{\varepsilon \Delta t}{2} \Delta(\nabla\delta^{n+1/2} - \nabla\delta^{n-1/2}) - \frac{\varepsilon \Delta t}{2} \Delta(\nabla q^{n+1/2} - \nabla q^{n-1/2}) \end{aligned}$$

or, rearranging terms,

$$\begin{aligned} \left(I - \frac{\varepsilon \Delta t}{2} \Delta\right) (\eta^{n+1} + \Delta t \nabla\delta^{n+1/2}) \\ = \left(I + \frac{\varepsilon \Delta t}{2} \Delta\right) \eta^n - \Delta t ([U \cdot \nabla] U)^{n+1/2} - [(V \cdot \nabla) V]^{n+1/2} \\ - \Delta t \frac{\varepsilon \Delta t}{2} \Delta \nabla\delta^{n-1/2} - \Delta t^2 \frac{\varepsilon}{2} \Delta(\nabla q^{n+1/2} - \nabla q^{n-1/2}). \end{aligned} \quad (2.12)$$

In the absence of boundaries we can invert  $(I - (\varepsilon \Delta t/2) \Delta)$  to obtain

$$\begin{aligned} \eta^{n+1} + \Delta t \nabla\delta^{n+1/2} &= \left(I - \frac{\varepsilon \Delta t}{2} \Delta\right)^{-1} \left(I + \frac{\varepsilon \Delta t}{2} \Delta\right) \eta^n - \Delta t \left(I - \frac{\varepsilon \Delta t}{2} \Delta\right)^{-1} \\ &\quad \times ([U \cdot \nabla] U)^{n+1/2} - [(V \cdot \nabla) V]^{n+1/2} \\ &\quad - \Delta t \left(I - \frac{\varepsilon \Delta t}{2} \Delta\right)^{-1} \frac{\varepsilon \Delta t}{2} \Delta \nabla\delta^{n-1/2} \\ &\quad - \Delta t^2 \left(I - \frac{\varepsilon \Delta t}{2} \Delta\right)^{-1} \frac{\varepsilon}{2} \Delta(\nabla q^{n+1/2} - \nabla q^{n-1/2}). \end{aligned}$$

If we now take norms of both sides exploiting the orthogonality of  $\eta^{k+1}$  and  $\nabla\delta^{n+1/2}$ , we can derive

$$\begin{aligned} \|\eta^{n+1}\| + \Delta t \|\nabla\delta^{n+1/2}\| &\leq \|\eta^n\| + \Delta t \|\nabla\delta^{n-1/2}\| + c \Delta t (\|\eta^n\| + \Delta t \|\nabla\delta^{n-1/2}\|) \\ &\quad + \frac{1}{2} \varepsilon \Delta t^2 \|\Delta(\nabla q^{n+1/2} - \nabla q^{n-1/2})\|. \end{aligned} \quad (2.13)$$

If we assume sufficient smoothness of the pressure then the last term in (2.13) is  $O(\Delta t^3)$  from which we conclude second-order accuracy of the velocity and first-order accuracy of the pressure.

Unfortunately, this analysis does not extend to the case in which boundaries are present. The basic source of difficulty, as observed by Chorin [8], is noncommutativity of the various operators appearing in the equations. In the particular form of the analysis we are using, the difficulty appears in the computation of  $(I - (\varepsilon \Delta t/2) \Delta)^{-1}$  in (2.12). Because  $\nabla\delta^{n+1/2}$  does not satisfy the boundary condi-



tions associated with the inversion of  $(I - (\varepsilon \Delta t / 2) \Delta)$ , the derivation of the analog of (2.13) from (2.12) in the case with boundaries introduces an additional boundary-layer term. Roughly speaking, the net effect of this term is to introduce an  $O((\varepsilon \Delta t)^{1/4})$  error (in  $L^2$ ) to the pressure approximation. This is essentially the same level of pollution that appears in Chorin's analysis [8]. The effect of this pollution on velocity is less severe. The boundary layer that is introduced is only large near boundaries where  $\eta^{n+1}$  vanishes. Thus, assuming suitable smoothness, the velocity errors are nearly orthogonal to the boundary layer term which reduces the error in velocity to  $O(\varepsilon \Delta t)$ .

The potential difficulties associated with boundary conditions appear to be an artifact of the analysis. There are several possible minor modifications to the algorithm that can be used to recover, formally, second-order accuracy; however, we have not found them to be necessary. As noted above, computational results reported in Section 5 clearly exhibit second-order convergence of velocity. Whether the analytical difficulty can be eliminated by a more careful treatment or is somehow "canceled" by the spatial discretization remains unknown.

Before discussing the spatial discretizations used in the algorithm, it will be useful to summarize the steps in the algorithm. First, we compute  $[(U \cdot \nabla) U]^{n+1/2}$  from  $U^n$  and  $\nabla p^{n-1/2}$ . Next, we solve the discrete form of the diffusion equations (2.5) with  $[(U \cdot \nabla) U]^{n+1/2}$  and  $\nabla p^{n+1/2,0} = \nabla p^{n-1/2}$  treated as source terms. Finally, we apply the projection to the auxiliary vector field  $U^*$  to decompose it into its divergence-free and gradient components. Initially, since we do not have initial data for the pressure we iterate (2.5)–(2.6) to convergence using the best available approximation to  $\nabla p$ , namely,  $\nabla p^{n+1/2,k}$  in computing  $[(U \cdot \nabla) U]^{1/2}$ . This introduces a perturbation term to (2.10) but does not substantially affect convergence or the analysis.

The bulk of the computational work associated with the method is spent on the linear algebra problems associated with the parabolic equations (2.5) and the projection. In each case the associated matrix problem is symmetric and positive definite with spectral properties corresponding to discretizations of second-order elliptic equations. Specific timings will be reported in Section 5.

### 3. APPROXIMATION OF DIFFUSION AND NONLINEAR CONVECTION TERMS

In this section we discuss the spatial discretization of the diffusion-convection equations that form the first step of the algorithm. In particular, we solve the diffusion-convection equations derived from (1.1) by treating the pressure term as a known forcing term (suitable lagged as described in the previous section) and by ignoring the incompressibility constraint (1.2). The spatial discretization is based on a cell-centered approximation that provides to most natural setting for Godunov-type methods. We let  $ij$  denote the cell whose center is located at  $((i-1/2) \Delta x, (j-1/2) \Delta y)$  for  $i=1, \dots, I; j=1, \dots, J$ . The right edge, top edge, and upper, right-hand corner of cell  $ij$  are denoted by  $i+1/2, j, i, j+1/2$ , and  $i+1/2, j+1/2$ , respec-

tively. Thus,  $1/2, j$  and  $I + 1/2, j$  refer to the left and right boundaries of the domain, etc. Velocity unknowns are specified at cell centers and velocity boundary conditions are specified at cell edges on the boundary.

There are essentially two pieces to the spatial approximation of (2.5): discretization of the Laplacian used to model the diffusion terms and the second-order Godunov procedure that is used to compute  $[(U \cdot \nabla) U]^{n+1/2}$ . The discretization of the Laplacian is done using standard, cell-centered finite difference approximations. It was observed by Russell and Wheeler [17] that for self-adjoint elliptic problems, cell-centered differences were equivalent to a mixed finite-element method using the lowest order Raviart–Thomas space and special quadrature rules. Weiser and Wheeler [18] exploit this relationship to show that the cell-centered difference approximation is second-order accurate. For the uniform grids considered in this paper the cell-centered approximation is equivalent to the standard five-point discretization of the Laplacian at interior cells; the only modifications occur for cells for which some edge lies on the boundary. For example, for cells bordering the left boundary (corresponding to index  $1, j$ ) we approximate  $u_{xx}$  by

$$u_{xx,1j} \approx \frac{1}{\Delta x} \left( \frac{u_{2,j} - u_{1,j}}{\Delta x} - \frac{2(u_{1,j} - u_{1/2,j})}{\Delta x} \right).$$

The reader is referred to the above papers for the construction of the approximation for more general grids and for details of the analysis.

The computation of  $[(U \cdot \nabla) U]^{n+1/2}$  is based on the construction of unsplit, second-order Godunov methods first proposed by Colella [13] and by van Leer [14]. Although considerable refinement is possible (see, for example, [15, 19]), we have, for this paper, used one of the simplest forms of this type of scheme. Unlike standard upwind differencing methods, these types of schemes couple the spatial and temporal discretization by propagating information along characteristics. This approach leads to a robust higher order discretization with excellent phase-error properties. The procedure can be broken down into three steps: reconstruction, characteristic extrapolation, and flux evaluation.

In the reconstruction step we compute a linear profile within each cell (discontinuous at boundaries). The slopes for these profiles are computed by monotonized difference approximations, e.g.,

$$u_x \approx \frac{\Delta_x u}{\Delta x}.$$

To compute  $\Delta_x u$  we first define an initial approximation

$$(\Delta_x u)^I = \frac{u_{i+1,j} - u_{i-1,j}}{2} \tag{3.1}$$

which will then be limited. The initial slope computation is modified at the boundary

to reflect the imposition of boundary conditions on the edges of boundary grid cells by taking, for example,

$$(\Delta_x u)^l = \frac{u_{2,j} + u_{1,j} - 2u_{1/2,j}}{2}.$$

The limiting step is designed to prevent the initial slopes from introducing new maxima and minima into the velocity field. This is accomplished by reducing the magnitude of  $\Delta_x u$  so that

$$|\Delta_x u| \leq 2 \max(|u_{i+1,j} - u_{i,j}|, |u_{i,j} - u_{i-1,j}|) \tag{3.2}$$

and by setting  $\Delta_x u = 0$  if

$$(u_{i+1,j} - u_{i,j}) \cdot (u_{i,j} - u_{i-1,j}) \leq 0. \tag{3.3}$$

At boundaries (3.2) is modified to ensure that the linear profile evaluated at the boundary point lies between the cell average and the specified boundary value.

The next step in the computation of  $[(U \cdot \nabla) U]^{n+1/2}$  is a characteristic extrapolation of  $U^n$  to predict values of  $U$  on cell edges at time  $t^{n+1/2}$ . The basis for the extrapolation is simply Taylor series. To second-order accuracy

$$U_{i+1/2,j}^{n+1/2,L} = U_{ij}^n + \frac{\Delta x}{2} U_{x,ij} + \frac{\Delta t}{2} U_{t,ij}, \tag{3.4a}$$

$$U_{i-1/2,j}^{n+1/2,R} = U_{ij}^n - \frac{\Delta x}{2} U_{x,ij} + \frac{\Delta t}{2} U_{t,ij}, \tag{3.4b}$$

$$U_{i,j+1/2}^{n+1/2,B} = U_{ij}^n + \frac{\Delta y}{2} U_{y,ij} + \frac{\Delta t}{2} U_{t,ij}, \tag{3.4c}$$

$$U_{i,j-1/2}^{n+1/2,T} = U_{ij}^n - \frac{\Delta y}{2} U_{y,ij} + \frac{\Delta t}{2} U_{t,ij}. \tag{3.4d}$$

The first two quantities denote the extrapolation of  $U$  to the left side of edge- $i+1/2, j$  and to the right side of edge- $i-1/2, j$ , respectively. The last two are the extrapolation of  $U$  to the bottom side of edge- $i, j+1/2$  and the top side of edge- $i, j-1/2$ . In these expressions  $U_{x,ij}$  and  $U_{y,ij}$  are evaluated using the limited slopes computed in the reconstruction phase. We now use the differential equations to express the time derivatives in terms of spatial derivatives. In particular, for each cell we evaluate  $U_t$  using our current approximation to the pressure to approximate the effect of the projection; i.e.,

$$U_t = \mathbf{P} \begin{pmatrix} \varepsilon \Delta u - uu_x - vv_y \\ \varepsilon \Delta v - uv_x - vv_y \end{pmatrix} \approx \begin{pmatrix} \varepsilon \Delta u - uu_x - vv_y \\ \varepsilon \Delta v - uv_x - vv_y \end{pmatrix} - \begin{pmatrix} p_x^{n-1/2} \\ p_y^{n-1/2} \end{pmatrix}. \tag{3.5}$$

(For the initial step we use  $\nabla p^{n+1/2,k}$  to approximate the effect of the projection instead of  $\nabla p^{n-1/2}$ .) When numerically evaluating the velocity derivatives in (3.5),

derivatives normal to the cell edge are treated differently than derivatives transverse to the edge for stability reasons (see Colella [13]). In particular, normal derivatives are evaluated using the limited slopes whereas transverse derivatives are treated in an upwind manner. More precisely, if  $v_{ij} \geq 0$ ,

$$U_{y,ij} = \frac{U_{ij} - U_{ij-1}}{\Delta y} + \frac{1}{2} \left( 1 - \frac{\Delta t v_{ij}}{\Delta y} \right) \frac{(\Delta_y U)_{ij} - (\Delta_y U)_{ij-1}}{\Delta y}$$

or, if  $v_{ij} < 0$ ,

$$U_{y,ij} = \frac{U_{ij+1} - U_{ij}}{\Delta y} - \frac{1}{2} \left( 1 + \frac{\Delta t v_{ij}}{\Delta y} \right) \frac{(\Delta_y U)_{ij+1} - (\Delta_y U)_{ij}}{\Delta y}.$$

This asymmetry in the treatment of the derivatives implies that there are actually two separate evaluations of the first-order derivatives in (3.5) corresponding to left-right edges and top-bottom edges, respectively.

The final step in the computation of  $[(U \cdot \nabla) U]^{n+1/2}$  is the actual computation of the flux differences. The exact form of the differencing is motivated by the relationship between the convective and conservative forms of the advection terms. Analytically, for divergence-free vector fields,

$$(U \cdot \nabla) U = \nabla \cdot (U \otimes U).$$

However, this relationship does not hold at  $t^{n+1/2}$  because of the effects of limiting and the approximation of the projection in (3.5) using old values of pressure. This introduces a variety of possibilities regarding the evaluation of  $[(U \cdot \nabla) U]^{n+1/2}$ . Our computational experiments indicate that the best approach is to difference  $[(U \cdot \nabla) U]^{n+1/2}$  in conservative form and then subtract off the requisite terms needed to modify the differencing to convective form. (If these terms are not subtracted from the differences the scheme generates considerable high-frequency noise that pollutes the numerical results.) Thus, we approximate  $(U \cdot \nabla) U$  by (suppressing  $n + 1/2$  superscripts)

$$\begin{aligned} uU_x + vU_y &\approx \frac{u_{i+1/2,j}U_{i+1/2,j} - u_{i-1/2,j}U_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2}U_{i,j+1/2} - v_{i,j-1/2}U_{i,j-1/2}}{\Delta y} \\ &\quad - \frac{1}{2} (U_{i+1/2,j} + U_{i-1/2,j}) \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \\ &\quad - \frac{1}{2} (U_{i,j+1/2} + U_{i,j-1/2}) \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \\ &= \frac{1}{2} (u_{i+1/2,j} + u_{i-1/2,j}) \frac{U_{i+1/2,j} - U_{i-1/2,j}}{\Delta x} \\ &\quad + \frac{1}{2} (v_{i,j+1/2} + v_{i,j-1/2}) \frac{U_{i,j+1/2} - U_{i,j-1/2}}{\Delta y}. \end{aligned} \tag{3.6}$$

Before evaluating the flux we must resolve the ambiguities in edge values introduced by (3.4). In particular, the characteristic extrapolation has defined double values of  $U$  for each edge corresponding to expansions from either side of the interface. We will restrict the discussion to the computation of  $U_{i+1/2,j}^{n+1/2}$  from the left and right states  $U_{i+1/2,j}^{n+1/2,L}$  and  $U_{i+1/2,j}^{n+1/2,R}$ . This is done by means of an upwinding procedure applied to the system that is obtained by dropping out the diffusion term, the pressure gradient, and the transverse flux, namely,

$$u_t + uu_x = 0 \tag{3.7a}$$

$$v_t + uv_x = 0. \tag{3.7b}$$

We see that  $u$  satisfies the quasilinear form of Burgers' equations and  $v$  is passively advected by  $u$ . This relationship to Burgers' equation motivates upwinding  $u$  based on

$$u_{i+1/2,j} = \begin{cases} u^L & \text{if } u^L \geq 0, u^L + u^R \geq 0 \\ 0 & \text{if } u^L < 0, u^R > 0 \\ u^R & \text{otherwise.} \end{cases}$$

(We suppress the temporal index and spatial indices on left and right states here and for the remainder of the discussion.) Note that this determination of  $u_{i+1/2,j}$  corresponds to the Riemann problem solution for the Burgers' equation (3.7a). The upwind determination for  $v$  is then given by

$$v_{i+1/2,j} = \begin{cases} v^L & \text{if } u_{i+1/2,j} > 0 \\ v^R & \text{if } u_{i+1/2,j} < 0 \\ 1/2(v^L + v^R) & \text{if } u_{i+1/2,j} = 0. \end{cases}$$

Note that the form of the differencing in (3.6) requires a value for  $v_{i+1/2,j}$  even in "sonic" cases in which  $u_{i+1/2,j} = 0$ . This upwind procedure is not used at boundaries; instead, we impose the specified boundary conditions at the  $\frac{1}{2}$ -time level.

The Godunov method is an explicit difference scheme and, as such, requires a time-step restriction. A linear, constant-coefficient analysis shows that we must require

$$\max_{ij} \left( \frac{u_{ij} \Delta t}{\Delta x}, \frac{v_{ij} \Delta t}{\Delta y} \right) \leq 1 \tag{3.8}$$

for stability. For some of the computations reported here we have had to reduce the time step by some safety factor in regions of high fluid strain. The time-step restriction of the Godunov method is used to set the time step for the overall algorithm.

As a final remark, we note that some minor modifications are required to the Godunov procedure when solving the Euler equations. These modifications stem from the fact that for the Euler equations we are only given  $U \cdot n = 0$  on the boundary; the tangential component of velocity is not specified. However, boundary values for these quantities are formally required in the slope computation

(3.1)–(3.3), in the evaluation of the transverse predictor terms (3.5) and in the corrector differencing (3.6), consequently some modification to the procedure is required. For slope computations and for the predictor, we have used first-order, one-sided difference approximations for cells adjacent to the boundary. In the corrector differencing, we have used characteristic extrapolation (e.g., the predictor step) to define the tangential velocity on the boundary.

#### 4. DISCRETIZATION OF THE PROJECTION

In this section we describe the numerical approximation of the projection. There are a number of possible approaches that can be used for this approximation. Chorin and van Kan each compute the gradient part,  $\nabla\phi$ , of a vector field  $V$ . The divergence-free part is then given by  $V^d = V - \nabla\phi$ . Alternatively, one may attempt to directly compute the divergence-free part of  $V$ . The latter approach was used by Stephens *et al.* [16] for the steady Navier–Stokes equations. Their approach is based on a discrete Galerkin formulation used in conjunction with a local basis for discretely divergence-free vector fields. For this paper we will adopt the Galerkin approach since the associated linear algebra problem is somewhat smaller. (The two approaches, of course, produce the same computational results.)

Before discussing the specific form of the Galerkin projection we have used, which depends on the definition of discrete divergence and gradient operators,  $D$  and  $G$ , we first discuss the requirements for the construction in a general context. Following Chorin [8], we require that the discrete operators  $D$  and  $G$  define approximations to the divergence and gradient satisfying the property that

$$(DV, \phi)_s = -(V, G\phi)_v, \quad (4.1)$$

where  $(\cdot, \cdot)_s$  and  $(\cdot, \cdot)_v$  represent appropriate inner products on discrete spaces of scalars and vectors, respectively. This condition represents a discrete form of integration by parts and guarantees that the numerical projection is orthogonal. (We note that the requirement (4.1) can be relaxed, typically near boundaries, to improve the accuracy of the difference formula without introducing any difficulties. See, for example, Chorin [7] and Solomon and Szymczak [20].)

To make this notion more precise we let  $\Omega_h$  represent the set of mesh locations at which we define velocity unknowns and  $\partial\Omega_h$  denote the boundary locations at which we specify velocity boundary conditions. The discrete divergence operator  $D$  maps vector fields defined on  $\bar{\Omega}_h \equiv \Omega_h \cup \partial\Omega_h$  to scalars defined at, possibly distinct, mesh locations  $W_h$ . The gradient operator,  $G$ , then maps from  $W_h$  back to  $\Omega_h$ . In terms of these definitions we require (4.1) to hold for all  $V$  defined on  $\bar{\Omega}_h$  that vanish on  $\partial\Omega_h$  and for all  $\phi$  defined on  $W_h$ . This condition then guarantees that for any vector field  $V_h$  defined on  $\Omega_h$ , there exists a scalar  $\phi_h$  defined on  $W_h$  and a vector field  $V_h^d$  defined on  $\Omega_h$  such that

$$V_h = V_h^d + G\phi_h \quad \text{on } \Omega_h \quad (4.2)$$

and

$$DV_h^{d,e} = 0 \quad \text{on } W_h, \tag{4.3}$$

where  $V_h^{d,e}$  is the extension of  $V_h^d$  to  $\partial\Omega_h$  by zero.

In order to construct a direct Galerkin approximation of the Hodge decomposition we want to find a local basis for

$$\mathbf{V}_h^d \equiv \{V_k^d : DV_h^{d,e} = 0\}.$$

The dimension of this space is given by

$$\dim \mathbf{V}_h^d = 2\mu(\Omega_k) - \mu(W_h) + \dim(\ker G), \tag{4.4}$$

where  $\mu(\cdot)$  is the number of mesh locations in its argument. Thus, to construct a local basis for  $\mathbf{V}_h^d$  we must find  $(\dim \mathbf{V}_h^d)$  linearly independent elements of  $\mathbf{V}_h^d$ .

Computation of a local basis for  $\mathbf{V}_h^d$  is not as formidable as it may appear. Although the situation is complicated somewhat by boundary conditions, it is generally the case that for any  $\phi$  defined on  $W_h$ ,  $G^\perp \phi \equiv (\phi_y, -\phi_x) \in \mathbf{V}_h^d$ . (In the present case  $\mathbf{V}_h^d$  is spanned by localized elements of this form; however, this is not always the case.) To see how this works in our particular case we first specify the grid system  $\Omega_h$ ,  $\partial\Omega_h$ , and  $W_h$ . As described in section three, we define velocity unknowns at cell centers and velocity boundary conditions at the midpoints of cell edges along the boundary. Thus,  $\Omega_h$  represents the cell centers, and  $\partial\Omega_h$  represents the midpoints of edges along the boundary. We define scalars and, consequently, divergence at the corners of grid cells which we denote by  $W_h$ . With this mesh structure,  $G$  is defined by

$$\phi_{x,ij} = \frac{\phi_{i+1/2,j+1/2} + \phi_{i+1/2,j-1/2} - \phi_{i-1/2,j+1/2} - \phi_{i-1/2,j-1/2}}{2\Delta x} \tag{4.5a}$$

$$\phi_{y,ij} = \frac{\phi_{i+1/2,j+1/2} + \phi_{i-1/2,j+1/2} - \phi_{i+1/2,j-1/2} - \phi_{i-1/2,j-1/2}}{2\Delta y}. \tag{4.5b}$$

We define the discrete divergence  $D$  from (4.1). In particular, for  $\phi$  defined on  $W_h$  and  $V_h = (u, v)$  defined on  $\bar{\Omega}_h$  with  $V_h = 0$  on  $\partial\Omega_h$ , we have

$$\begin{aligned} \sum_{i=1}^I \sum_{j=1}^J (\phi_{x,ij} u_{ij} + \phi_{y,ij} v_{ij}) &= - \sum_{i=1}^{I-1} \sum_{j=1}^{J-1} \phi_{i+1/2,j+1/2} \left( \frac{u_{i+1,j+1} + u_{i+1,j} - u_{i,j+1} - u_{ij}}{2\Delta x} \right. \\ &\quad \left. + \frac{v_{i+1,j+1} + v_{i,j+1} - v_{i+1,j} - v_{ij}}{2\Delta y} \right) \\ &\quad - \frac{1}{2} \sum_{j=1}^{J-1} \phi_{1/2,j+1/2} \left( \frac{u_{1,j+1} + u_{1,j}}{\Delta x} + \frac{v_{1,j+1} - v_{1,j}}{\Delta y} \right) \\ &\quad - \frac{1}{4} \phi_{1/2,1/2} \left( \frac{2u_{11}}{\Delta x} + \frac{2v_{11}}{\Delta y} \right) - \dots \end{aligned} \tag{4.6}$$

This discrete summation by parts formula defines the discrete divergence  $D$  and implicitly defines the inner products for  $\Omega_h$  and  $W_h$ . In particular, the term to the right of  $\phi$  in each of the three exhibited terms gives the divergence approximation in the interior, along the interior of the left edge and in the lower left-hand corner of the domain. (Analogous expressions define the divergence at other edges and corners.) The weights to the left of  $\phi$  in each term define the inner product on  $W_h$ ; namely, 1 in the interior, 1/2 on the interior nodes of boundary segments, and 1/4 at corners. The weights for the inner product on  $\Omega_h$  are all unity. We note that these inner products correspond to a tensor-product midpoint integration rule on  $\Omega_h$  and a tensor-product trapezoidal rule on  $W_h$ .

We can now define a basis for  $V_h^d$  using  $D$  and  $G$  defined by (4.5) and (4.6). We let

$$\Psi_{i+1/2, j+1/2} = G^{-1} \phi_{i+1/2, j+1/2},$$

where the indices range over elements of  $W_h$  not lying on the physical boundary. Here  $\phi_{i+1/2, j+1/2}$  is one at corner node  $i+1/2, j+1/2$  and zero elsewhere. (Note that  $D\Psi_{i+1/2, j+1/2} = 0$  follows trivially from (4.1)). We note that there are  $(I-1)(J-1)$   $\Psi$ 's. The sizes of  $\Omega_h$  and  $W_h$  are  $IJ$  and  $(I+1)(J+1)$ , respectively. Since the  $G$  defined by (4.5) has a 2-dimensional null space, the  $\Psi$ 's form a basis for  $V_h^d$ . (Linear independence of the  $\Psi$ 's follows from the invertibility of the discrete projection matrix which can be decomposed into a block-diagonal form where each block is an  $m$ -matrix. See Stephens *et al.* [16] for details.)

To actually compute the projection of a vector field  $V_h$  defined on  $\Omega_h$ , we note that the divergence-free part of  $V_h$  can be represented as a linear combination of basis elements; i.e.,  $\mathbf{P}V_h = \sum \alpha_{i+1/2, j+1/2} \Psi_{i+1/2, j+1/2}$  for some  $\alpha_{i+1/2, j+1/2}$ . To compute the  $\alpha_{i+1/2, j+1/2}$  we solve the linear system

$$\left( \sum \alpha_{i+1/2, j+1/2} \Psi_{i+1/2, j+1/2}, \Psi_{k+1/2, l+1/2} \right)_v = (V, \Psi_{k+1/2, l+1/2})_v. \quad (4.7)$$

When appropriately scaled, Eq. (4.7) has an interesting interpretation. If we use the definition of the  $\Psi$ 's to re-express Eq. (4.7) in terms of the  $\phi_{i+1/2, j+1/2}$  and then sum by parts, we find that the coefficients of the  $\alpha$ 's form a discretization of  $-\Delta$  and that the right-hand side is an approximation to the vorticity (the curl of  $V$ ). Thus, the  $\alpha$ 's computed during the projection define a discrete stream function for the velocity field.

## 5. NUMERICAL RESULTS

In this section we present numerical results designed to test the capabilities of the numerical method presented in the previous sections. For an application of the techniques described in this paper to a realistic flow problem see Bell *et al.* [21]. The first set of computations measure the numerical rate of convergence in the presence of boundaries. For these tests we define a smooth initial velocity inside the



unit square satisfying homogeneous Dirichlet boundary conditions. The initial velocity field is defined in terms of a stream function

$$\Psi = \pi^{-1} \sin^2(\pi x) \sin^2(\pi y).$$

The initial velocities are then defined by

$$u = -\Psi_y, \quad v = \Psi_x.$$

The normalization was chosen so that the maximum initial velocity was one. Computations were performed for Stokes flow ( $\epsilon = 0.1$  with the nonlinear terms dropped out), Reynolds number 100 ( $\epsilon = 0.01$ ), and for the incompressible Euler equations ( $\epsilon = 0$ ). For each case, we compute the solution on uniform grids with  $\Delta x = \Delta y = 1/2^n$  for  $n = 4, 5, \dots, 8$ . For each grid we integrate to time 0.5 with  $\Delta t = \frac{1}{2} \Delta x$ .

To estimate the convergence rate we compare the solution obtained on each grid with the solution obtained on the next finer grid. When the grid is refined by a factor of 2, each cell in the grid is split into four pieces. We let  $U_{h/2}^{ave}$  represent the values of the solution obtained with  $\Delta x = h/2$  averaged onto the grid with  $\Delta x = h$ . We then compute

$$\|U_{h/2}^{ave} - U_h\|_{l^2}$$

for each grid (except, of course, the finest). Asymptotically, these differences are proportional to the errors on the coarser grid. Table I contains these error measurements along with numerical rates of convergence for each of the cases. Second-order convergence for each case is apparent. The deviation from second-order for the nonzero Reynolds-number cases is most likely related to the effects of slope limiting in the Godunov scheme. Limiting drops the local accuracy to first order which can account for deviations from second-order behavior on coarse grids.

The second example is that of a jet in a doubly period geometry for the Euler equations. The initial data is given by

$$u = \begin{cases} \tanh((y - 0.25)/\rho) & \text{for } y \leq 0.5 \\ \tanh((0.75 - y)/\rho) & \text{for } y > 0.5 \end{cases}$$

$$v = \delta \sin(2\pi x), \tag{5.1}$$

TABLE I  
Convergence Results

	16-32	Rate	32-64	Rate	64-128	Rate	128-256
Stokes	1.64E-3	1.95	4.23E-4	1.97	1.08E-4	1.98	2.73E-5
Re 100	6.89E-3	1.85	1.90E-3	1.87	5.20E-4	1.92	1.36E-4
Euler	1.25E-2	2.09	2.94E-3	2.37	5.68E-4	2.16	1.27E-4

where  $\rho = 1/30$  and  $\delta = 0.05$ . Thus, the initial flow field consists of a horizontal shear layer of finite thickness, perturbed by a small amplitude vertical velocity. Each of the shear layers making up the boundaries of the jet evolves into a periodic array of large vortices, with the shear layer between the rolls being thinned by the large straining field there. Eventually, these thinned layers wrap around the large rolls. The evolution of the top and bottom layers are mirror images of one another, modulo translation by half the length of the box. In Figs. 1–3, we show results on  $128 \times 128$ ,  $256 \times 256$ , and  $512 \times 512$  grids, respectively. These computations were run with time-step equal to half that allowed by (4.6). Shown in the plots are contours of vorticity, computed by applying a second-order central difference formula

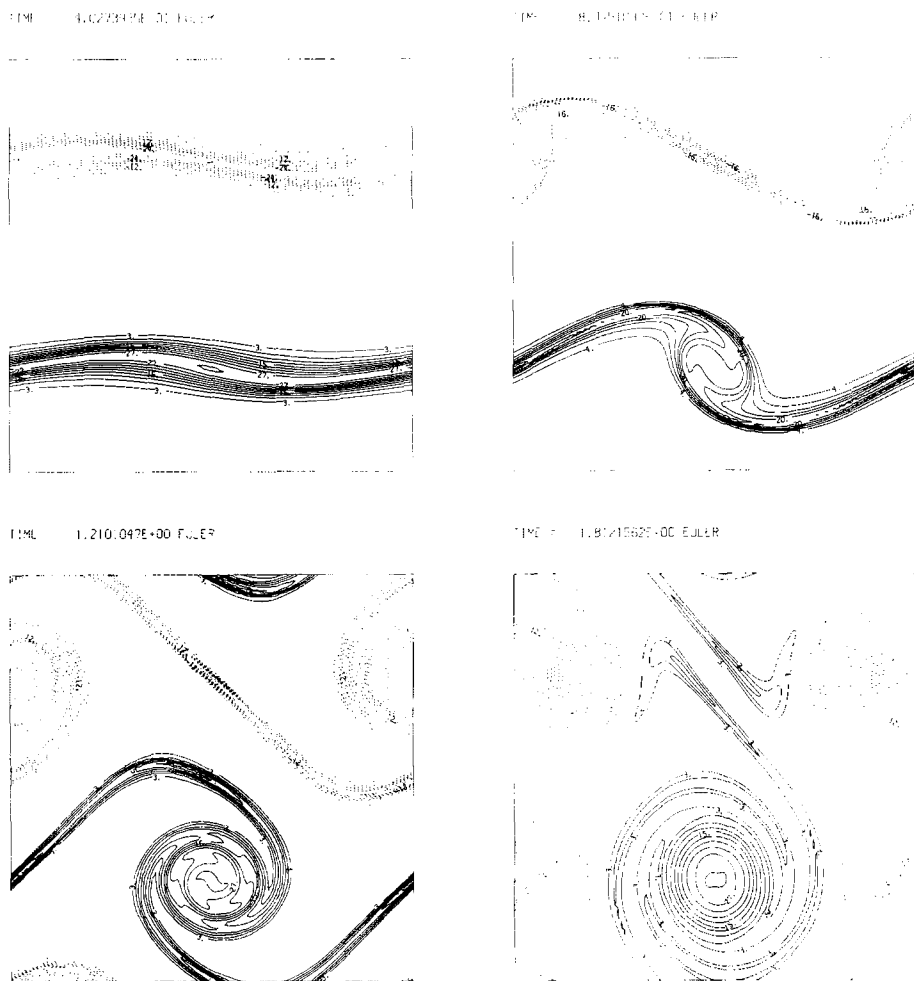
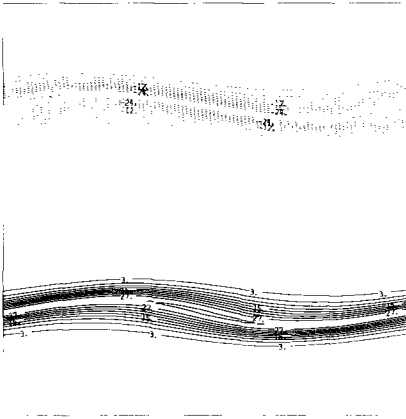
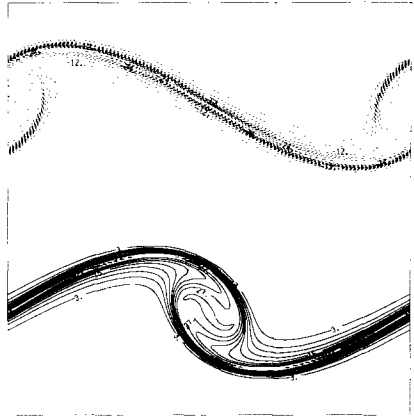


FIG. 1. Vorticity contours for smooth shear layer on  $128 \times 128$  grid.

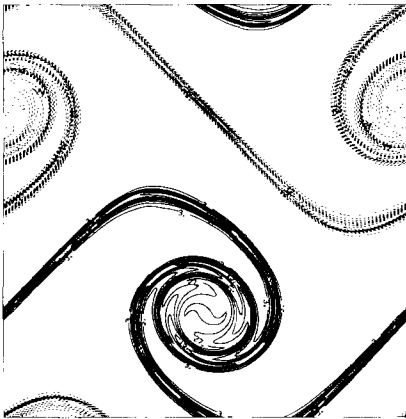
TIME = 1.0273812E+01 EULER



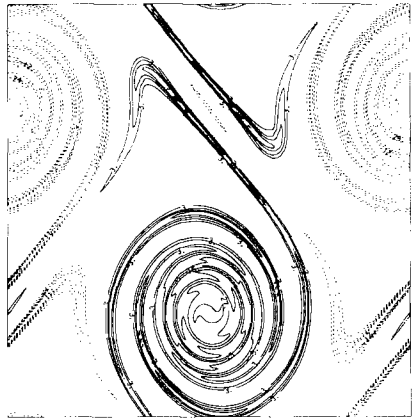
TIME = 9.3583895E+01 EULER



TIME = 1.2094665E+00 EULER



TIME = 1.8143086E+00 EULER

FIG. 2. Vorticity contours for smooth shear layer on  $256 \times 256$  grid.

to the velocity field. At time 0.8, the vorticity is well resolved to plotting accuracy even of the coarsest grid. As time progresses and the shear layers between the vortices become thinner, the coarser grids are unable to resolve the details of the solution. At time 1.8, the convergence of the vorticity to plotting accuracy on the  $512 \times 512$  grid is in doubt. Nonetheless, the computed solution degrades gracefully in the underresolved regions. The principal manifestation of the lack of grid resolution is the smearing of the high-frequency components of the vorticity field. There is no obvious evidence of problems with large dispersive errors in the presence of large gradients, such as oscillations or distortions in the shape of the large vortices

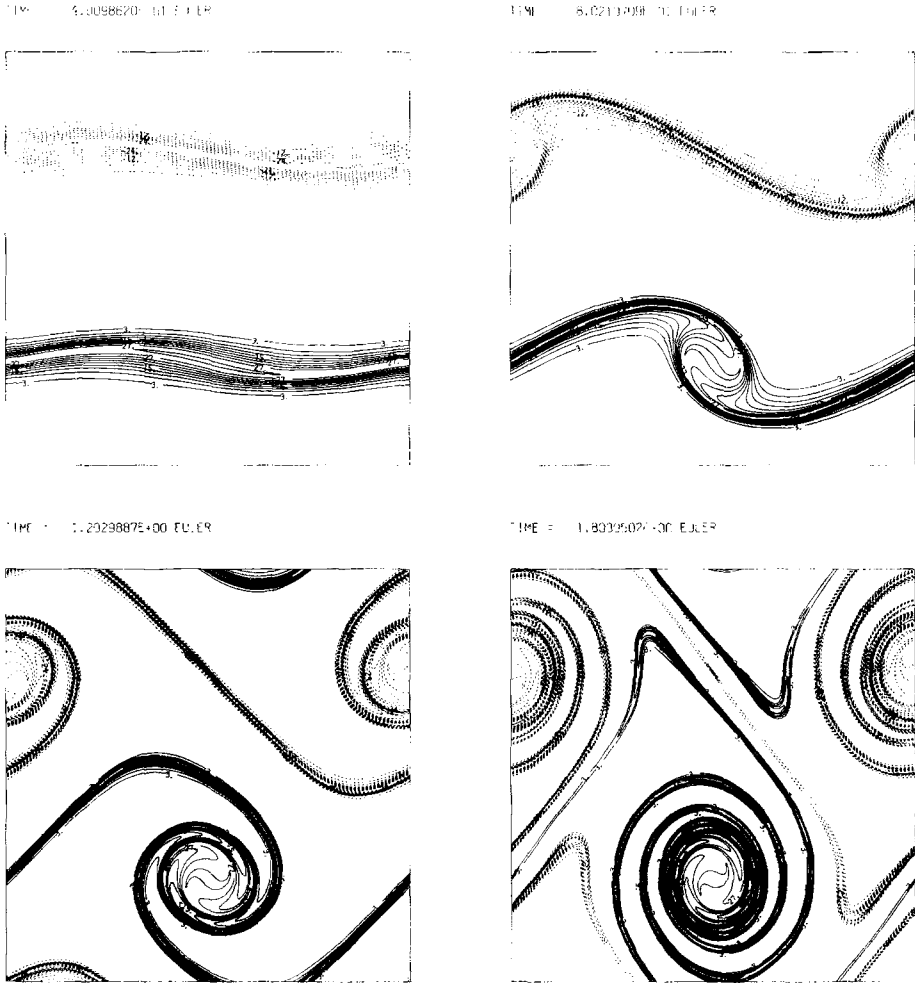


FIG. 3. Vorticity contours for smooth shear layer on  $512 \times 512$  grid.

on the coarse grids. In Fig. 4 we give time histories of the integral of the kinetic energy

$$K = \int U \cdot U \, dx \, dy$$

as a function of time. Since  $K$  is an invariant of solutions to the Euler equations, its variation in time for computed solutions is a gross measure of the accuracy of the method. It is evident from the calculations that  $K$  at any given time is converging to the exact answer at a rate of  $O(h^2)$  as one would expect from a second-

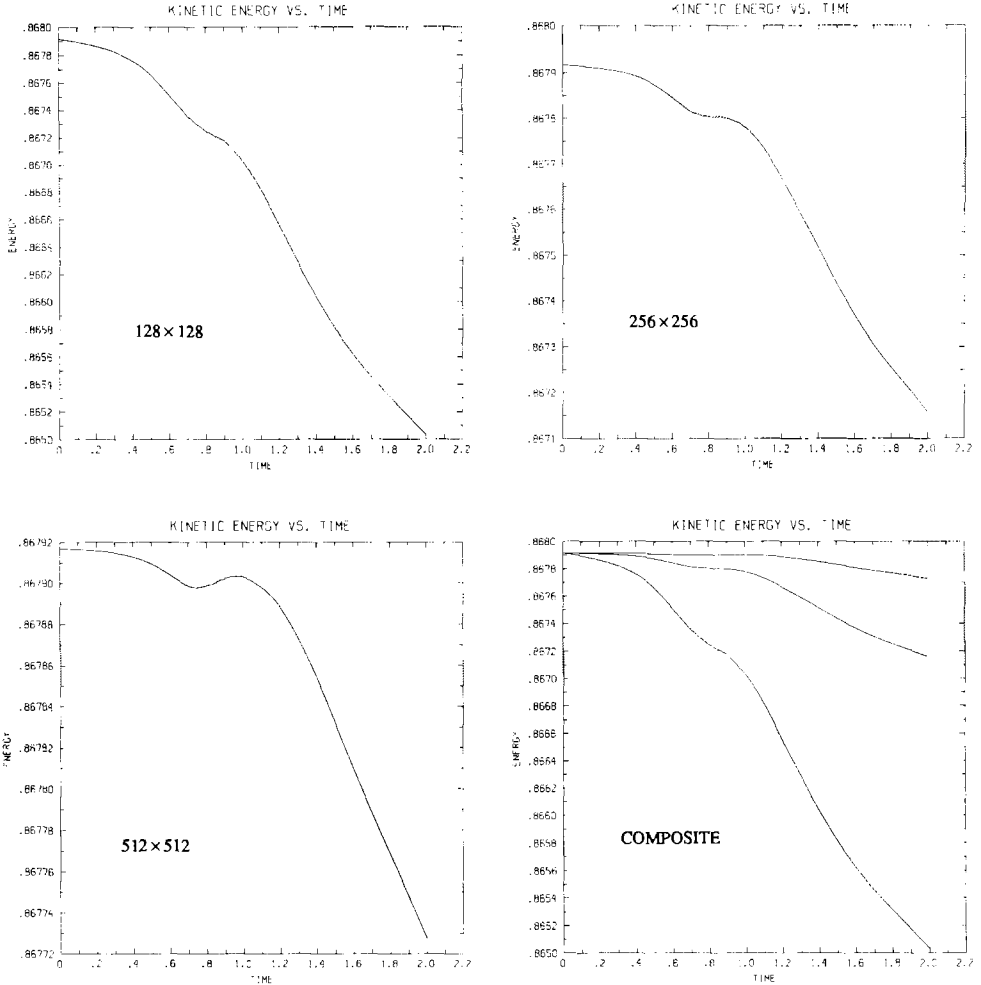


FIG. 4. Kinetic energy versus time for smooth shear layer.

order method. In addition, the variation of  $K$  for any of the grids is quite modest; on the coarsest grid, the maximum error in  $K$  is 0.35%. For the most part,  $K$  is decreasing as a function of time, although in the finer grid calculations, there is a slight increase in  $K$  around  $t=1$ . However, we do not find this particularly disturbing since the increase is small compared to the overall variation in  $K$ .

Our final set of test calculations are for the same initial data and boundary conditions as the previous ones, except that the shear layers bounding the jet are infinitely thin, i.e.,  $\rho \rightarrow 0$  in (5.1). In addition to providing a stringent test of the method, this example is also motivated by the widespread practice of using finite difference algorithms for the Euler equations to calculate the dynamics of sharp

shear layers for compressible flow. Such calculations have been performed for a variety of physical problems, including vortex sheets formed at the leading edge of a delta wing, and the slip line in Mach reflection. The justification usually given for the physical correctness of such calculations is that if the point at which the shear layer is created is computed correctly, then the numerical dissipation in the finite difference algorithm will mimic the physical dissipation mechanisms, leading to results which have the correct large-scale dynamics. The capability of the present algorithm to compute both high Reynolds number and Euler solutions for discontinuous data would permit one to examine the validity of this justification, albeit in a much simpler, incompressible setting. Such computations might shed some light

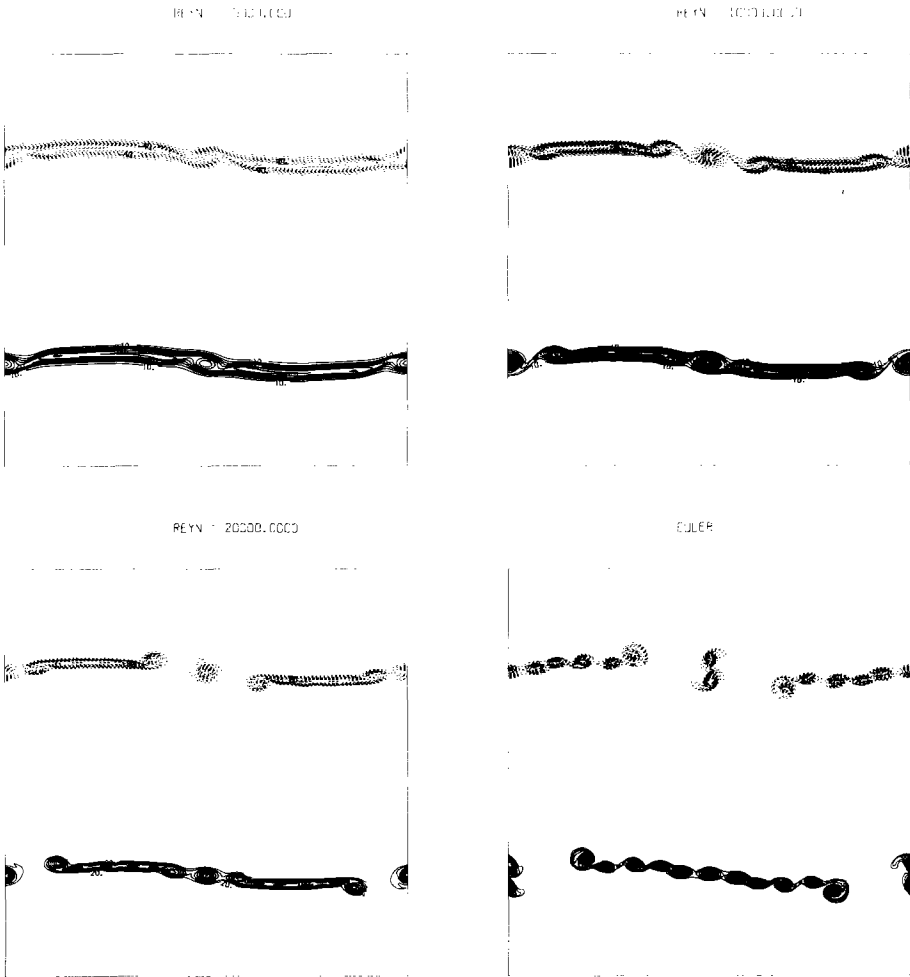


FIG. 5. Vorticity contours at  $t = 0.3$  on  $256 \times 256$  grid for varying viscosity.

on this issue in the compressible case, where many of the difference schemes used to compute solutions to the compressible Euler equations are closely related to our Godunov procedure for the nonlinear terms and for which the corresponding compressible Navier–Stokes calculations would be prohibitively expensive.

In Figs. 5–8, we show vorticity plots of solutions obtained using our method for both Navier–Stokes and Euler with the time-step selected to be 80% of the maximum allowed by (4.6). In each figure we show results for  $\text{Re} \equiv 1/\varepsilon = 5000, 10,000, 20,000$  and for Euler. The first two figures show early-time results ( $t = 0.3$ ) for  $256 \times 256$  and  $512 \times 512$  grids. Figs. 7–8 show results for these same grids at  $t = 0.6$ . At the lower Reynolds numbers, we have good agreement between the coarse and

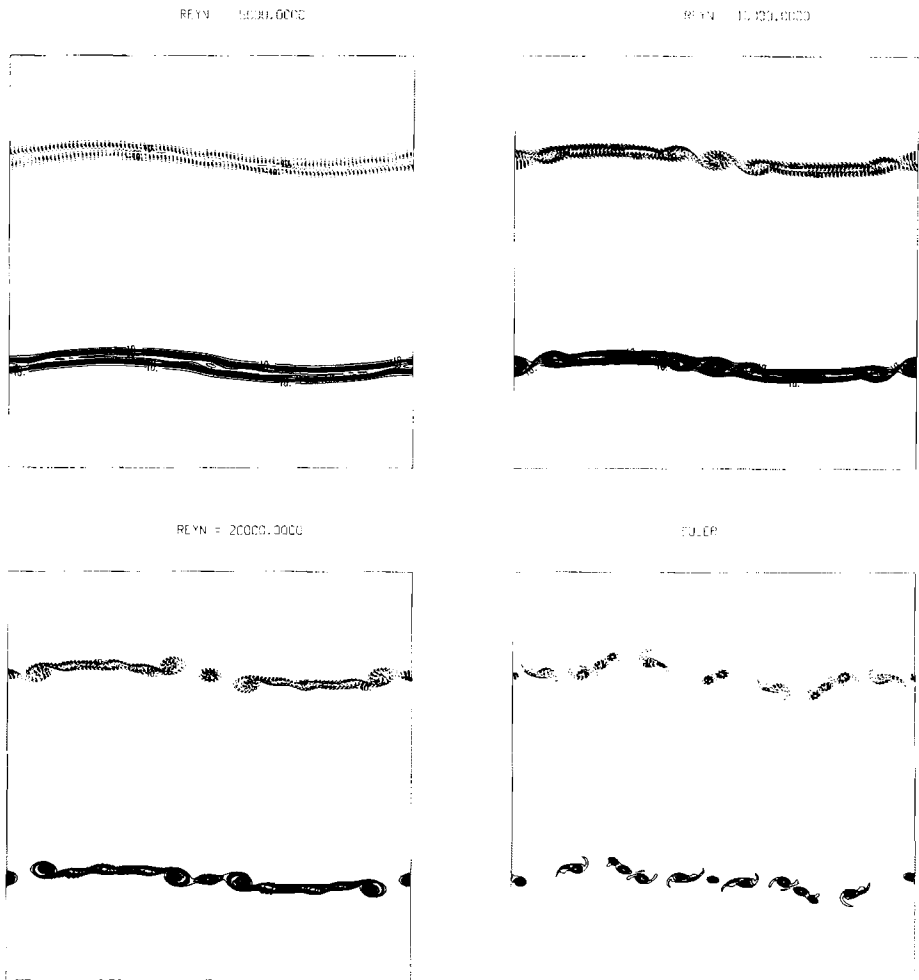


FIG. 6. Vorticity contours at  $t = 0.3$  on  $512 \times 512$  grid for varying viscosity.

fine grid solutions. At  $Re = 20,000$ , higher harmonics of the initial perturbation appear on the fine grid results which are not resolved on the coarse grid.

The Euler results, of course, are not expected to converge to anything as  $\Delta x \rightarrow 0$ . However, we note that for the Euler results, the characteristic wavelength of the small-scale structures appears to increase linearly with the mesh spacing, although one can still discern the original perturbation in the large-scale deflections of the jet which is roughly the same on both grids. In any case, the results serve to validate our claim that we have no cell Reynolds number stability restrictions; in particular, our method remains stable and non-oscillatory for  $\varepsilon = 0$ , even with discontinuous initial data.

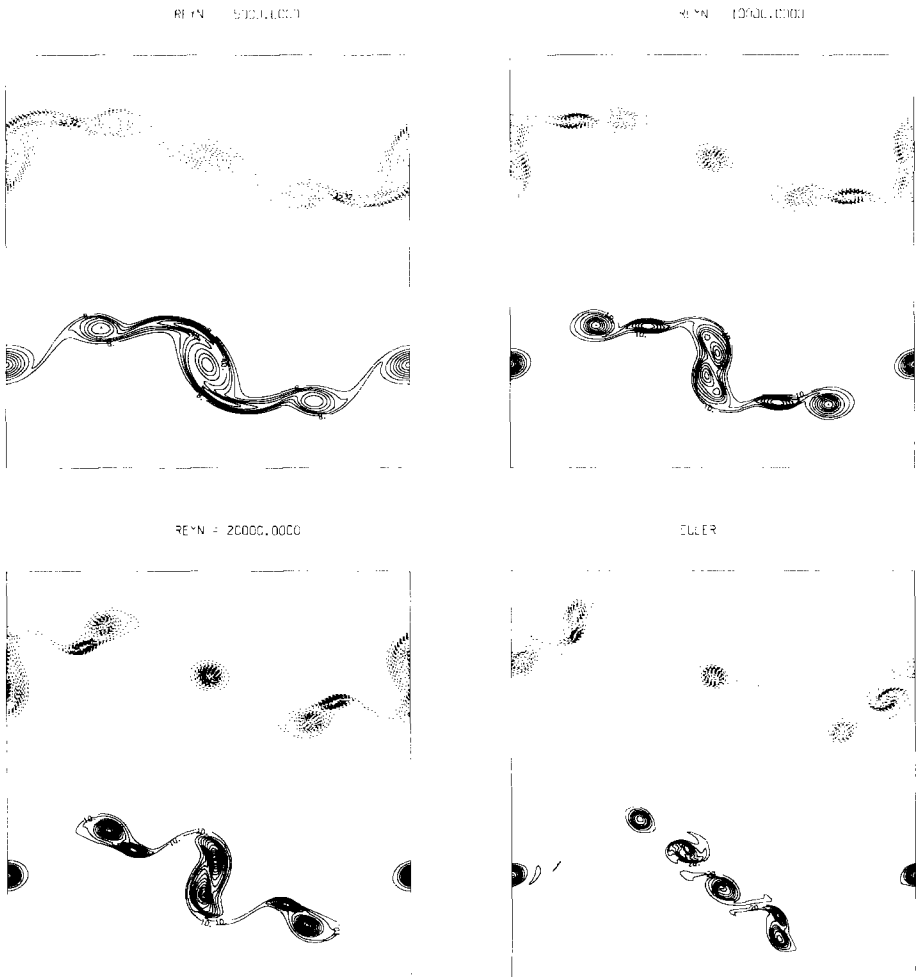


FIG. 7. Vorticity contours at  $t = 0.6$  on  $256 \times 256$  grid for varying viscosity.



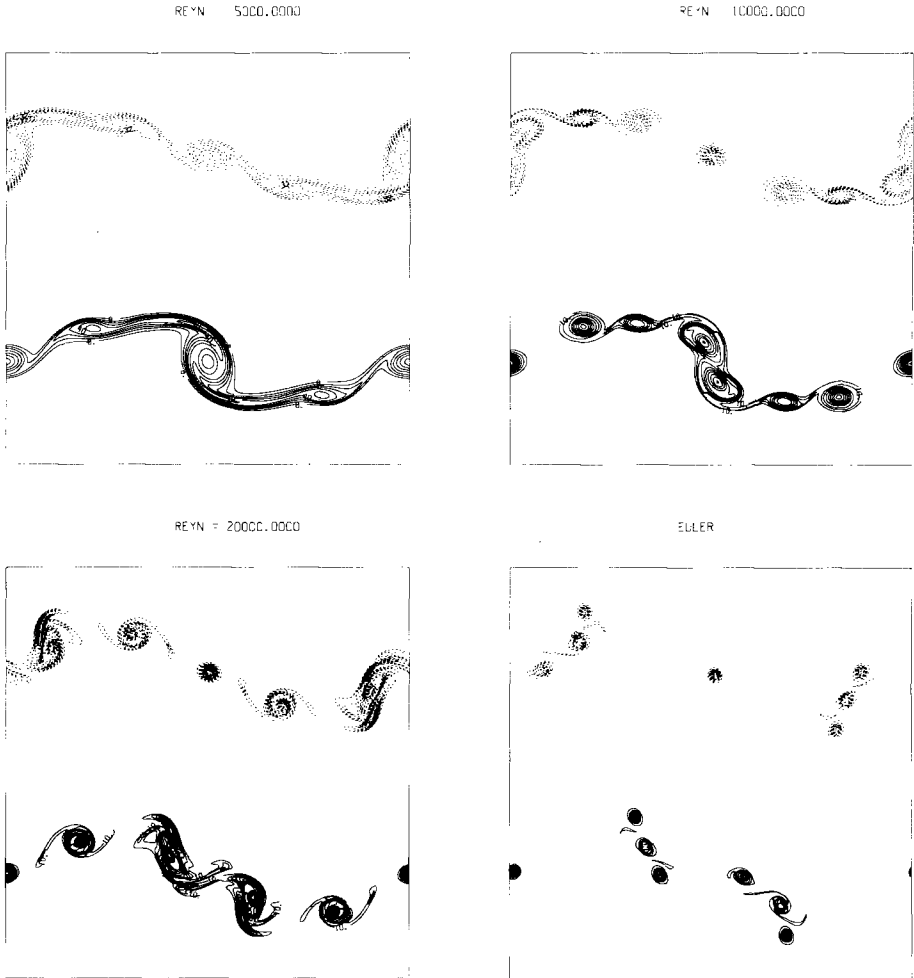


FIG. 8. Vorticity contours at  $t=0.6$  on  $512 \times 512$  grid for varying viscosity.

As a final comment concerning the performance of the method, we will briefly discuss some representative timings for the method on  $128 \times 128$  grids on a single-processor Cray XMP. Linear algebra is the dominant cost in the algorithm; consequently, the choice of iterative method can lead to considerable variation in results. In both the periodic case and the finite domain case we solved the linear systems required for the heat equations when viscous terms are present using a diagonally scaled preconditioned conjugate gradient. For the examples with boundary conditions, the projection was done using an MILU(0) preconditioned conjugate gradient algorithm. For the Euler equations the method required  $27.9 \mu\text{s}$  to advance a grid point with 76% of the cost being the projection linear algebra. Addition of

the viscous term with  $\varepsilon = 0.01$  increased the time per zone to  $40.8 \mu\text{s}$ . For the doubly periodic examples the projection was computed using FFTs which reduced the computational cost to  $12.5 \mu\text{s}$  per zone for the inviscid case. In this case, addition of the viscosity increased the time to  $23.7 \mu\text{s}$  per zone.

## 6. CONCLUSIONS

In this paper we have presented a new projection algorithm for the incompressible Navier–Stokes equations. The algorithm maintains sufficient coupling between the diffusion-convection step and the projection step so that the overall temporal discretization is second-order accurate. The method also incorporates an explicit Godunov-type discretization of the nonlinear terms. The Godunov method provides a robust, second-order accurate discretization for smooth flows that remains stable for singular initial data such as vortex sheets, even in the limit of vanishing viscosity. The method was tested on a simple model problem to validate the second-order convergence of the method. The method was also used to model the dynamics of shear layers in a doubly periodic domain. For a smooth initial layer the method performed well in the absence of viscosity. The solutions appeared to be quadratically convergent and provided reasonable resolution of the principal features of the flow on the coarsest grid. When applied to a discontinuous initial layer the method remained stable even in the Euler limit. Furthermore, for this initial data the method provided acceptable resolution of viscous effects for a cell Reynolds number of approximately 40.

Future work will address the extension of the method developed here to quadrilateral grids and to incorporate more general boundary conditions. This will allow us to model more complex geometries and to use boundary-layer zoning near solid walls to obtain better resolution of boundary-layer phenomena at high Reynolds number. The quadrilateral-grid version of the algorithm will then form the basis for the development of a local mesh refinement algorithm that will allow us to focus computational effort where it is required to resolve complex flow features.

## REFERENCES

1. O. A. LADYZHENSKAYA, *Mathematical Problems in the Dynamics of a Viscous Incompressible Flow* (Gordon & Breach, New York, 1963).
2. H. FUJITA AND T. KATO, *Arch. Rat. Mech. Anal.* **16**, 269 (1964).
3. R. TEMAM, *Navier–Stokes Equations* (Elsevier Science, Amsterdam, 1984).
4. F. H. HARLOW AND J. E. WELCH, *Phys. Fluids* **8**, 2182 (1965).
5. P. M. GRESHO AND R. L. SANI, *Int. J. Numer. Methods Fluids* **7**, 1111 (1987).
6. A. KRZYWICKI AND O. A. LADYZHENSKAYA, *Soviet Phys. Dokl.* **11**, 212 (1966).
7. A. J. CHORIN, *Math. Comput.* **22**, 745 (1968).
8. A. J. CHORIN, *Math. Comput.* **23**, 341 (1969).
9. A. J. CHORIN, *Stud. Num. Anal.* **2**, 64 (1968).
10. R. TEMAM, *Arch. Rat. Mech. Anal.* **32**, 135, 377 (1969).

11. J. KIM AND P. MOIN, *J. Comput. Phys.* **59**, 308 (1985).
12. J. VAN KAN, *SIAM J. Sci. Statist. Comput.* **7**, 870 (1986).
13. P. COLELLA, "A Multidimensional Second Order Godunov Scheme for Conservation Laws," LBL-17023, Lawrence Berkeley Laboratory, May 1984 (unpublished).
14. B. VAN LEER, "Multidimensional Explicit Difference Schemes for Hyperbolic Conservation Laws," in *Computing Methods in Applied Sciences an Engineering, VI*, p. 493 (Elsevier Science, Amsterdam, 1984).
15. J. B. BELL, C. N. DAWSON, AND G. R. SHUBIN, *J. Comput. Phys.* **74**, 1 (1988).
16. A. B. STEPHENS, J. B. BELL, J. M. SOLOMON, AND L. B. HACKERMAN, *J. Comput. Phys.* **53**, 152 (1984).
17. T. F. RUSSELL AND M. F. WHEELER, "Finite element and finite difference methods for continuous flows in porous media," in *Mathematics of Reservoir Simulation* (SIAM, Philadelphia, 1984).
18. A. WEISER AND M. WHEELER, *SIAM J. Num. Anal.* **25**, 351 (1988).
19. J. S. SALTZMAN, "Monotone Difference Schemes for the Linear Advection Equation in Two and Three Dimensions," LAUR 87-2479, Los Alamos National Laboratory, 1987 (unpublished).
20. J. M. SOLOMON AND W. G. SZYMCZAK, "Finite Difference Solutions for the Incompressible Navier-Stokes Equations using Galerkin Techniques," in *Fifth IMACS International Symposium on Computer Methods for Partial Differential Equations, Lehigh University, June 19-21, 1984*.
21. J. B. BELL, H. M. GLAZ, J. M. SOLOMON, AND W. G. SZYMCZAK, "Application of a Second-Order Projection Method to the Study of Shear Layers," in *11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburg, VA, June 27-July 1, 1988*.