AIAA-91-1560-CP

AN EFFICIENT SECOND-ORDER PROJECTION METHOD FOR VISCOUS INCOMPRESSIBLE FLOW*

A91-40734

John B. Bell†
Lawrence Livermore National Laboratory
Livermore, CA 94550

Phillip Colella§
University of California Berkeley
Berkeley, CA 94720

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

Abstract

In this paper we describe a second-order projection method for the time-dependent, incompressible Navier-Stokes equations. The method is a second-order fractional step scheme in which one first solves diffusion-convection equations to determine intermediate velocities which are then projected onto the space of divergence-free vector fields. The diffusion-convection step uses a specialized second-order Godunov method for differencing the nonlinear convective terms that is conservative and free-stream preserving and provides a robust treatment of the nonlinearities at high Reynolds number. The projection is based on cell-centered centered difference approximations to divergence and gradient operators with the resulting linear system solved using a multigrid relaxation scheme. We apply the method to vortex spindown in a box to validate the numerical convergence of the method and to measure its overall performance.

Introduction

This paper describes a second-order projection method for the incompressible Navier Stokes equations

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

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

Projection methods, which were first introduced by Chorin [1,2,3], are fractional step methods that compute an intermediate velocity field by solving (1.1), ignoring the incompressibility condition (1.2). This intermediate velocity field is then projected onto the space of divergence-free fields to recover the approximation to the velocity.

Several second-order generalizations of the original projection method have been proposed. Kim and Moin [4] use an inhomogeneous boundary condition for the intermediate velocity field, 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 [5]. They provide computational evidence that their scheme is second-order accurate. Van Kan [6] proposes another second-order generalization of the projection method. This method is based on first discretizing the spatial terms using the Harlow and Welch staggered grid, thereby reducing 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.

Another second-order projection method (BCG) was introduced by Bell, Colella and Glaz [7]. This method uses a strategy similar to van Kan's for the basic temporal discretization. It also incorporates an improved treatment of the nonlinear terms in (1) using a second order Godunov method that provides a robust treatment of the nonlinearity at high Reynolds number. The BCG algorithm has subsequently been extended to quadrilateral grids [8], three space dimensions [9], and variable density flows [10].

In the present work, we discuss a number of improvements to the BCG algorithm described in [7] to

* 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. Partial support under contract No. W-7405-Eng-48 was provided by the Applied Mathematical Sciences Program of the Office of Energy Research. Prof. Colella was supported by the Army Research Office under grant DAAL03-88-K-0197 ; by DARPA and the National Science Foundation under grant DMS-8919074; and by a National Science Foundation Presidential Young Investigator award under grant ACS-8958522.

† Group Leader, Applied Mathematics Group

§ Associate Professor, Dept. of Mechanical Engineering

‡ Staff Scientist, Applied Mathematics Group

improve its efficiency and robustness. We introduce a somewhat different treatment of the differencing of the advective terms that eliminates an instability observed for high Reynolds number and Euler calculations at advective CFL numbers ≥0.5. We also systematically use multigrid algorithms for the projection and the parabolic terms throughout; this is a nontrivial step for the projection operators, since they involve nonstandard, locally decoupled stencils for the discrete Laplacian. Combining these two ideas leads to an overall factor of 5 speedup over the method in [7].

The numerical method is a three-step process. In the first step, an unsplit, second-order Godunov method is used to compute time-centered conservative differences of the nonlinear flux terms $(U \cdot \nabla)U$. The Godunov method is a predictor-corrector method in which characteristics are used to extrapolate velocities to $t^{n+½}$ from the solution at time t^n. These predicted velocities are then used in a corrector step in which Riemann problems are solved to resolve ambiguities in the upwind direction and the resultant states are used to evaluate centered difference approximations to the advective derivatives. In the second step of the algorithm we solve (1.2) using a Crank-Nicholson discretization in which the pressure gradient and the nonlinear term (evaluated during the previous step) are treated as source terms. The pressure gradient is evaluated at time $t^{n-½}$. The explicit, second-order treatment of the nonlinear term in the first step provides a time-centered approximation of this term while maintaining a symmetric positive-definite system for the linear algebra associated with the Crank-Nicholson time discretization. The result of the second step of the algorithm would have been a second order approximation to u^{n+1} if a time centered pressure had been available. The lagging of the pressure term in (1.1) introduces a second-order perturbation into the result of step two. This perturbation is manifest in the failure of the incompressibility condition (1.2). In the final step of the algorithm, a discrete Hodge decomposition is performed to remove the non-divergence-free component from the solution at t^{n+1} and to update the pressure.

In the next section we describe the basic fractional step algorithm that specifies temporal discretization strategy. Details of the Godunov procedure for computing the time-centered approximation of $(U \cdot \nabla)U$ are described in section 3. The multigrid algorithm for computing the discrete projection is described in the fourth section. Finally, we present a numerical example with timings to illustrate the performance of the method.

Temporal discretization

In this section we review the second-order fractional step formulation from [7] used in the present work. Projection methods are based on the decomposition of vector fields into a divergence-free component and the gradient of a scalar field. More precisely, any vector field V can be uniquely written as $V = U_d + \nabla\phi$ where ϕ is a scalar and U_d is divergence free and satisfies specified boundary conditions. For the purposes of this paper, we shall assume that a solid boundary encloses the entire fluid, so that $U_d \cdot \mathbf{n} = 0$. One can define an orthogonal projection $\mathbf{P}$ such that $U_d = \mathbf{P}V$ and $\nabla\phi = (\mathbf{I}-\mathbf{P})V$. (See Temam [11] for a more detailed discussion of the projection.) Using the projection we can rewrite the Navier-Stokes equations (1.1)-(1.2) in the equivalent form

$$U_t = \mathbf{P}(\varepsilon \Delta U - (U \cdot \nabla) U) \quad . \qquad (2.1)$$

Equation (2.1) describes the evolution of U in terms of a nonlinear functional of U; the pressure has been eliminated from the system. The pressure in (1.1)-(1.2) represents the gradient component of the vector field that is projected in (2.1); i.e.,

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

For the basic temporal discretization, we assume that we are given an approximation to U^n. Furthermore, we assume that we have already computed a second-order, time-centered approximation to the nonlinear terms $[(U \cdot \nabla) U]^{n+½}$. (A Godunov-type procedure for computing this approximation is described in the next section.) A second-order discretization of (2.1) can be obtained using a Crank-Nicholson approximation

$$\frac{U^{n+1}-U^n}{\Delta t} = \mathbf{P}\left[\frac{\varepsilon}{2}\Delta(U^n+U^{n+1})-[(U \cdot \nabla) U]^{n+½}\right] \quad . \qquad (2.2)$$

However, the linear algebra problem associated with solving (2.2) would be extremely costly because of the nonlocal behavior of the projection.

As a less costly alternative, we construct a fractional step method that approximates (2.2) to second-order accuracy. To accomplish this we will assume that we are also given an approximation to $\nabla p^{n-½}$. We then compute an intermediate velocity field U^* using

$$\frac{U^*-U^n}{\Delta t}+\nabla p^{n-½} = \frac{\varepsilon}{2}\Delta(U^n+U^*)-[(U \cdot \nabla) U]^{n+½} \quad , \qquad (2.3)$$

where U^* satisfies the same boundary conditions as U. The role of the pressure gradient term in (2.3) is to approximate the effect of the projection in (2.2). We now apply the projection to decompose U^* into divergence-free and gradient components to obtain U^{n+1} and an update for ∇p

$$U^{n+1} = \mathbf{P}\, U^* \qquad (2.4a)$$

$$\nabla p^{n+½} = \nabla p^{n-½} + \Delta t^{-1}(\mathbf{I}-\mathbf{P})U^* \quad . \qquad (2.4b)$$

Equations (2.3)-(2.4) represent the fractional step scheme that we have used. The relationship between (2.3)-(2.4) and the Crank-Nicholson scheme (2.2) can be seen by first observing that (2.4) is equivalent to

$$\frac{U^{n+1} - U^{n}}{\Delta t} = \mathbf{P}\left[\frac{U^{*} - U^{n}}{\Delta t} + \nabla p^{n-1/2}\right] \qquad (2.5a)$$

$$\nabla p^{n+1/2} = (\mathbf{I}-\mathbf{P})\left[\frac{U^{*} - U^{n}}{\Delta t} + \nabla p^{n-1/2}\right] \quad . \qquad (2.5b)$$

If we use equation (2.3) to replace

$$\frac{U^{*} - U^{n}}{\Delta t} + \nabla p^{n-1/2}$$

in (2.5) we obtain

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

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

from which we can see that (2.4a) corresponds to (2.2) with U^{n+1} approximated by U^{*} on the right hand side and that $\nabla p^{n+1/2}$ represents the gradient component of the vector field being projected.

We note that since $\nabla p^{-1/2}$ is not available initially, we iterate (2.3) and (2.4) (with $\nabla p = 0$ initially) on the first step. The inclusion of the $\nabla p^{n-1/2}$ in (2.3) makes the algorithm second-order accurate in time (see [7]).

Before describing the spatial discretizations used in the algorithm we will summarize the basic approach. First we solve the diffusion-convection equations (2.3). This is a two-step process in which we first approximate $[(U\cdot\nabla)\,U]^{n+1/2}$ using a second-order Godunov procedure. Then, we solve the two parabolic equations represented by (2.3) with the nonlinear term treated as a source term. In the second step of the algorithm, we apply the projection to update U and ∇p. In the next section we discuss the spatial discretization of the diffusion-convection equations that forms the first step of the algorithm. In the following section we describe the approximation of the projection.

Spatial discretization

In this section we discuss the spatial discretization of the diffusion-convection equations (2.3) that form the first step of the algorithm. The spatial discretization is based on a cell-centered approximation that provides the most natural setting for Godunov-type methods. We let i,j denote the cell whose center is located at $((i-1/2)\Delta x, (j-1/2)\Delta y)$ for $i=1,\ldots,I$; $j=1,\ldots,J$. The right edge and top edge of cell i,j are denoted by $i+1/2,j$ and $i,j+1/2$, respectively. Thus, $1/2,j$ and $I+1/2,j$ refer to the left and right boundaries of the domain, etc. Velocity and pressure unknowns are specified at cell centers and velocity boundary conditions are specified at cell edges on the boundary.

The first step in solving (2.3) is the evaluation of $[(U\cdot\nabla)\,U]^{n+1/2}$ using a specialized second-order Godunov method. In [7], the construction of the time-centered predicted velocities is done by computing the leading order terms in the Taylor expansion of U at the cell center, replacing the time derivative with space derivatives using (1.1). The pressure gradient used for this purpose was that obtained on the previous time step. The use of this lagged pressure gradient in the predictor step for the advective derivatives leads to a nonlinear instability for advective CFL numbers ≥0.5. Analysis and computational experimentation suggested that this instability was related to the failure of the predicted velocities to satisfy (1.2). Here we present a modified version of the discretization of $(U\cdot\nabla)\,U$ that respects (1.2) which we find eliminates an instabilities at high advective CFL.

In the new version of the the predictor step, we proceed as before to compute the time-centered edge velocities using a Taylor series approximation.

$$U^{n+1/2,L}_{i+1/2,j} = U^{n}_{i,j} + \frac{\Delta x}{2}U^{n}_{x,ij} + \frac{\Delta t}{2}U^{n}_{t,ij} \qquad (3.1a)$$

$$U^{n+1/2,R}_{i-1/2,j} = U^{n}_{i,j} - \frac{\Delta x}{2}U^{n}_{x,ij} + \frac{\Delta t}{2}U^{n}_{t,ij} \qquad (3.1b)$$

$$U^{n+1/2,B}_{i,j+1/2} = U^{n}_{i,j} + \frac{\Delta y}{2}U^{n}_{y,ij} + \frac{\Delta t}{2}U^{n}_{t,ij} \qquad (3.1c)$$

$$U^{n+1/2,T}_{i,j-1/2} = U^{n}_{i,j} - \frac{\Delta y}{2}U^{n}_{y,ij} + \frac{\Delta t}{2}U^{n}_{t,ij} \quad . \qquad (3.1d)$$

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-$i,j-1/2$. We now approximate U_t using the differential equation (1.1). This approximation will be built in three stages. First, we define $\hat{U}$, which only includes the derivative terms normal to the edge. In the second phase, we add the viscous terms and an approximation to the transverse derivatives using $\hat{U}$ to compute the next approximation $\tilde{U}$. An upwinding procedure is applied to the $\tilde{U}$'s to resolve ambiguities at the interface and a MAC-type projection is used to incorporate the effect of the pressure gradient on the final edge values $U^{n+1/2}$. Thus, we have, for (3.1a)

$$\hat{U}^{L}_{i+1/2,j} = U^{n}_{i,j} + s_L\left[\frac{\Delta x}{2} - \frac{\Delta t}{2}u^{n}_{i,j}\right]U^{n}_{x,i,j} \qquad (3.2a)$$

where s_L is 1 if $u^{n}_{i,j} \geq 0$ and 0 otherwise; and

$$\tilde{U}^{L}_{i+1/2,j} = \hat{U}^{L}_{i,j} - \frac{\Delta t}{2}(\nu U_y)_{i,j} + \frac{\Delta t}{2}\varepsilon(\Delta U^{n})_{i,j} \qquad (3.2b)$$

Given left and right extrapolated values of the form, $\hat{U}^{L,R}_{i+1/2,j}$ or $\tilde{U}^{L,R}_{i+1/2,j}$, we assume that we have a means of extracting single edge-centered values $\hat{U}_{i+1/2,j}$, $\tilde{U}_{i+1/2,j}$, corresponding to an upwinding procedure; such a procedure is given below.

In equation (3.2), $\Delta U_{i,j}^{n}$ is evaluated using a standard five point finite difference approximation. The U_x term is evaluated using a fourth-order monotonicity limited slope approximation given by

$$\Delta x \, q_{x,i,j} \approx (\delta q)_{i,j} \;, q = u, v,$$

$$(\delta q)_{i,j} = \text{sign}(q_{i+1,j} - q_{i-1,j}) \times$$

$$\min\left(\left| \frac{2(q_{i+1,j} - q_{i-1,j})}{3} - \frac{((\delta^f q)_{i+1,j} + (\delta^f q)_{i-1,j})}{6} \right|, (\delta^{\text{lim}} q)_{i,j} \right)$$

$$(\delta^f q)_{i,j} = \min(2 | q_{i+1,j} - q_{i-1,j} |, \delta^{\text{lim}} q_{i,j}) \times \text{sign}(q_{i+1,j} - q_{i-1,j})$$

$$(\delta^{\text{lim}} q)_{i,j} = \min(2 | q_{i+1,j} - q_{i,j} |, 2 | q_{i,j} - q_{i-1,j} |) \quad \text{if } (q_{i+1,j} - q_{i,j})(q_{i,j} - q_{i-1,j}) > 0$$

$$= 0 \text{ otherwise} \;.$$

In [7] the slope formula corresponding to $\delta q = \delta^f q$ was used; the slope formula given here is preferred, as it has slightly less damping than the previous one (see [12]). The transverse derivative term vU_y is given by

$$vU_y \approx \frac{1}{2\Delta y} (\hat{v}_{i,j+\frac{1}{2}} + \hat{v}_{i,j-\frac{1}{2}})(\hat{U}_{i,j+\frac{1}{2}} - \hat{U}_{i,j-\frac{1}{2}})$$

This form of the transverse derivative term is slightly more elaborate than the one originally used in [7]; for linear advection, it is known to have somewhat lower phase error.

Our upwinding procedure is based on the Riemann problem for Burgers' equation. In particular, to uniquely define edge values $U_{i+\frac{1}{2},j}$ from left and right extrapolated values $U_{i+\frac{1}{2},j}^{L}$, $U_{i+\frac{1}{2},j}^{R}$, we first define the normal component (in this case u).

$$u_{i+\frac{1}{2},j} = \begin{cases} u^L & \text{if } u^L \geq 0 \,,\, u^L + u^R \geq 0 \\ 0 & \text{if } u^L < 0 \,,\, u^R > 0 \\ u^R & \text{otherwise} \end{cases} \tag{3.3}$$

(We suppress $i+\frac{1}{2},j$ spatial indices on left and right states for clarity.) Now we upwind v based on the normal velocity.

$$v_{i+\frac{1}{2},j} = \begin{cases} v^L & \text{if } u_{i+\frac{1}{2},j} > 0 \\ v^R & \text{if } u_{i+\frac{1}{2},j} < 0 \\ \frac{1}{2}(v^L + v^R) & \text{if } u_{i+\frac{1}{2},j} = 0 \end{cases} \tag{3.4}$$

To obtain a time-centered value for U at the cell edges, we must adjust $\tilde{U}_{i+\frac{1}{2},j}$ to account for the effect of the incompressibility condition to order Δt^2, i.e. by adding the pressure gradient term from (3.1):

$$U_{i+\frac{1}{2},j}^{n+\frac{1}{2}} = \tilde{U}_{i+\frac{1}{2},j} - \frac{\Delta t}{2} (\nabla p)_{i+\frac{1}{2},j} \tag{3.2c}$$

In [7], this was accomplished by adding to the right hand side of (3.2b) the quantity $-\Delta t / 2 \nabla p_{i,j}^{n-\frac{1}{2}}$, using the estimate of the pressure gradient obtained in the projection step (2.4) during the previous time step. We have found the use of this lagged pressure gradient introduces a mild instability for high Reynolds number and inviscid problems for advective CFL numbers greater than 0.5. For that reason, we have introduced the following alternative method, which eliminates that instability. Given $\tilde{U}_{i+\frac{1}{2},j}$, $\tilde{U}_{i,j+\frac{1}{2}}$, we compute the MAC divergence at the cell centers

$$(D^M \tilde{U})_{i,j} = \frac{1}{\Delta x} (\tilde{u}_{i+\frac{1}{2},j} - \tilde{u}_{i-\frac{1}{2},j}) + \frac{1}{\Delta y} (\tilde{v}_{i,j+\frac{1}{2}} - \tilde{v}_{i,j-\frac{1}{2}})$$

We then solve $\Delta_h \phi = D^M \tilde{U}$, where Δ_h is the standard five-point discretization of the Laplacian with cell-centered homogeneous Neumann boundary conditions. We use the field so obtained to compute the correction (3.2c).

$$-\frac{\Delta t}{2} p_{x,i+\frac{1}{2},j} \approx \frac{1}{\Delta x} (\phi_{i+1,j} - \phi_{i,j})$$

$$-\frac{\Delta t}{2} p_{y,i+\frac{1}{2},j} \approx \frac{1}{4\Delta y} (\phi_{i,j+1} + \phi_{i+1,j+1} - (\phi_{i,j-1} + \phi_{i+1,j-1}))$$

The resulting values for $U^{n+\frac{1}{2}}$ have the property that $D^M U^{n+\frac{1}{2}} = 0$; in particular, one can use the normal velocities at the cell edges to advect additional quantities in a way that is both conservative and leaves spatially constant fields unchanged.

Finally, we difference the $U^{n\frac{1}{2}}$'s to compute the approximation to $[(U \cdot \nabla) U]^{n+\frac{1}{2}}$.

$$(u\,U_x + v\,U_y)_{i,j} \approx \tfrac{1}{2}(u_{i+\frac{1}{2},j} + u_{i-\frac{1}{2},j})(U_{i+\frac{1}{2},j} - U_{i-\frac{1}{2},j}) + \tfrac{1}{2}(v_{i,j+\frac{1}{2}} + v_{i,j-\frac{1}{2}})(U_{i,j+\frac{1}{2}} - U_{i,j-\frac{1}{2}}) \tag{3.5}$$

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

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

for stability. The time-step restriction of the Godunov method is used to set the time step for the overall algorithm.

Once the evaluation of $[(U \cdot \nabla) U]^{n+\frac{1}{2}}$ is complete, we solve (2.3) with $[(U \cdot \nabla) U]^{n+\frac{1}{2}}$ and $\nabla p^{n-\frac{1}{2}}$ terms treated as source terms. Operationally, the equations are reduced to simple diffusion equations with source terms. The Laplacians in these equations are discretized using standard five-point finite difference equations. Both the linear system resulting from this discretization, and the one resulting from the MAC projection, are solved using using standard multigrid techniques. (See Briggs [13], for an introduction to multigrid techniques.)

Discretization of the projection

The final step of the algorithm involve application of the discrete projection to U^* to define the new velocity approximation and an update for the pressure as specified by (2.4). In this section we describe the discrete Hodge decomposition that is used to compute the divergence-free component of the velocity field U^*. The present formulation is different from the approach used in [7] in that both scalars (pressure) and vectors (velocity) are defined at cell centers. In addition, we use a multigrid scheme tailored to the particular properties of the projection operator.

In the following, we assume that $I = J = 2^N$ for some N. To simplify the formulas, we also assume that $\Delta x = \Delta y = h$, although that is not essential. To further simplify the notation we will develop the projection for an arbitrary given discrete vector field V defined on our grid, having component $(v^1_{i,j}, v^2_{i,j})$. We want to compute compute the decomposition

$$V = V^d + G\phi \tag{4.1}$$

where D and G are discrete divergence and gradient operators and $DV^d = 0$. D and G are assumed to be adjoints with respect to a pair of inner products $(\cdot,\cdot)_s$ and $(\cdot,\cdot)_v$ on discrete vector and scalar fields.

$$(DV,\phi)_s = -(V,G\phi)_v \tag{4.2}$$

$$(\phi,\psi)_s = \sum_{i,j} (\phi_{i,j}\psi_{i,j})h^2$$

$$(V,W)_v = \sum_{i,j} (V_{i,j}\cdot W_{i,j})h^2$$

Thus, only one of the operators D, G can be specified, the other being uniquely determined by (4.2) To define D in the interior of the grid, we use centered differences to approximate the derivatives appearing in the divergence operator.

$$v_x^1 \approx \frac{(v^1_{i+1,j} - v^1_{i-1,j})}{2h}$$

$$v_y^2 \approx \frac{(v^2_{i,j+1} - v^2_{i,j-1})}{2h}$$

At the solid-wall boundary, we use a one-sided difference approximation that incorporates the noflow boundary condition $V \cdot n = 0$ to define the divergence operator. For example, at the leftmost point on the grid, we use the approximation

$$v_x^1 \approx \frac{v^1_{1,j} + v^1_{2,j}}{2h}$$

With this definition of D, it is easy to derive the form of G from (4.2) In the interior, the gradient operator is also approximated by centered differences:

$$\phi_{x,i,j} \approx \frac{(\phi_{i+1,j} - \phi_{i-1,j})}{2h}$$

$$\phi_{y,i,j} \approx \frac{(\phi_{i,j+1} - \phi_{i,j-1})}{2h} .$$

At boundaries, we obtain an apparently inconsistent approximation to G.

$$\phi_{x,1,j} \approx \frac{\phi_{2,j} - \phi_{1,j}}{2h}$$

However, when one takes into account the boundary condition for the gradient field in the Hodge decomposition $\partial\phi/\partial n = 0$, we see that the definition of the discrete gradient at the boundaries is a combination of the interior formula and an extrapolation that sets

$$\phi_{0,j} = \phi_{1,j} . \tag{4.3}$$

Taking the divergence of equation (4.1) gives an equation

$$DG\phi = DV \tag{4.4}$$

to be solved for ϕ, after which V is easily obtained from (4.1).

There are two complications to this problem. Because we use centered differences for D and G, the resulting stencil for DG is not the usual 5-point approximation to the Laplacian, but rather is twice as large. Specifically, if we enforce (4.3) and the additional reflection

$$\phi_{-1,j} = \phi_{2,j} . \tag{4.5}$$

then DG is of the form

$$(DG\phi)_{i,j} = \frac{1}{4h^2}(\phi_{i-2,j} + \phi_{i+2,j} + \phi_{i,j-2} + \phi_{i,j+2} - 4\phi_{i,j}) \tag{4.6}$$

This stencil locally decouples the computational mesh into four disjoint subgrids that are globally coupled at the boundary by the boundary conditions (4.3) and (4.5); in particular, the nullspace of DG is the one-dimensional space of constant scalar fields. The idea, then, is to develop a version of multigrid that respects the local decoupling of the stencil (4.6) while accounting correctly for the boundary coupling.

Our basic multigrid relaxation cycle is the full multigrid V-cycle (FMV) (cf. [13].) Given ϕ, a current guess at the solution, ρ, the right-hand side, and h, the mesh spacing, we compute $MG(\phi,\rho,h)$, a new update to ϕ, as follows.

(i) Relax the solution with a point-relaxation scheme: $\phi := \phi + relax(\phi,\rho,h)$.

(ii) Form the residual $R = \rho - DG\phi$, and average onto the coarsened grid with spacing $2h$: $R^C = A(R)$

(iii) Apply multigrid relaxation on the $2h$ mesh: $\delta^C := MG(\delta^C, R^C, 2h)$, where δ^C is initialized to be identically zero. If $2h$ is equal to the size of the domain, step (ii),(iii) and (iv) are omitted.

(iv) Interpolate the correction onto the fine mesh, and add to the current guess ϕ: $\phi := \phi + I(\delta^C)$.

(v) Perform a second point-relaxation step: $\phi := \phi + relax(\phi,\rho,h)$.

We use Gauss-Seidel with red-black ordering (relative to the larger stencil), on each of the four locally-decoupled subgrids as our basic relaxation scheme. This choice insures that any high-frequency components in the residual on each subgrid are damped sufficiently prior to coarsening. Of course, any high-frequency component to the residual corresponding, for example, to different constant values on each of the four subgrids, will not be damped. Our averaging and interpolation operators are chosen to preserve the local decoupling. They are given as follows.

$$A(R)_{2(i-1)+r,2(j-1)+s} =$$

$$\frac{1}{4}(R_{4(i-1)+r,4(j-1)+s} + R_{4(i-1)+2+r,4(j-1)+s}$$

$$+ R_{4(i-1)+r,4(j-1)+2+s} + R_{4(i-1)+2+r,4(j-1)+2+s})$$

$$I(\delta^C)_{4(i-1)+r+r',4(j-1)+s+s'} = \delta^C{}_{2(i-1)+r,2(j-1)+s}$$

$$(r,s) = (1,1),(1,2),(2,1),(2,2)$$

$$(r',s') = (0,0),(0,2),(2,0),(2,2)$$

For these choices of relaxation, averaging, and interpolation operators, we obtain performance typcially found on more standard elliptic equations, with multigrid reducing the residual by five orders of magnitude in seven iterations.

Numerical results

We have timed the present method for a fully vectorized version of the code, and have compared the results to those reported in [7] All calculations were performed on a Cray XMP with 9.5 nsec. clock, using the cft77 compiler. For problems on a 128 x 128 mesh, the present code takes 9.5 μ secs. per cell per time step for inviscid problems, and 17.9 μ secs. per cell for Re = 100. The corresponding figures in [7] were 27.9 μ secs. and 41. μ secs., respectively. This is despite the introduction of an additional Poisson solve in the inviscid predictor. Thus, from the introduction of multigrid alone, we obtain a factor 2.3 - 2.9 speedup. Combined with the doubled time step, we get a factor of 4.6 - 5.8 increase in efficiency. These timings include the effect of the more elaborate slope and transverse derivative calculations; when we use the simpler ones described in [7] the speedups are slightly better. We also wish to emphasize that all the multigrid solves were iterated until the residual was less than 10^{-9} relative to the magnitude of the velocity field. In Figure 1, we show the vorticity contours for the viscous spindown of a vortex in a box for Re = 20000 on a 256×256 grid. This calculation was run at a time step that was continuously adjusted so that the maximum advective CFL number was exactly equal to 1.0; such a time step caused the algorithm in [7] to go unstable, generating $2\Delta x$ noise in the vorticity contours. We see no evidence of such instabilities in the figures below. In Figure 2, we show results on grids of size 64×64, 128×128 and 256×256. We observe that the method is able to resolve the formation of successive separation points in the boundary layer and the resulting large vortical structures even on the medium resolution grid. Finally, we note that the running time for 256×256 problem was reduced to 8.7 μ secs. per zone, due to the longer vector lengths and higher Reynolds number.

References

1. A. J. Chorin, "Numerical Solution of the Navier-Stokes Equations," *Math. Comp.*, vol. 22, pp. 745-762, Oct. 1968.
2. A. J. Chorin, "On the Convergence of Discrete Approximations to the Navier-Stokes Equations," *Math. Comp.*, vol. 23, pp. 341-353, April 1969.
3. A. J. Chorin, "Numerical Solution of Incompressible Flow Problems," *Studies in Num. Anal.*, vol. 2, pp. 64-71, 1968.
4. J. Kim and P. Moin, "Application of a Fractional-Step Method for Incomressible Navier-Stokes Equations," *J. Comput. Phys.*, vol. 59, pp. 308-323, 1985.
5. F. H. Harlow and J. E. Welch, "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluids with Free Surfaces," *Physics of Fluids*, vol. 8, pp. 2182-2189, 1965.
6. J. Van Kan, "A Second-Order Accurate Pressure-Correction Scheme for Viscous Incompressible Flow," *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 870-891, July 1986.
7. J. B. Bell, P. Colella, and H. M. Glaz, "A Second-Order Projection Method for the Incompressible Navier-Stokes Equations," *J. Comp. Phys.*, vol. 85, pp. 257-283, 1989.
8. J. B. Bell, J. M. Solomon, and W. G. Szymczak, "A Second-Order Projection Method for the Incompressible Navier Stokes Equations on Quadrilateral Grids," *paper no. 89-1967-CP, AIAA 9th Comp. Fluid Dynamics Conf.* , Buffalo, NY, June 14-16, 1989.
9. J. B. Bell, J. M. Solomon, and W. G. Szymczak, "A Second-Order Projection Method for the Three-Dimensional Euler and Navier-Stokes Equations," *to appear*, 1991.
10. J. B. Bell and D. L. Marcus, "A Second-Order Projection Method for Variable-Density Flows," UCRL-JC-104132, May 1990. submitted for publication
11. R. Temam, *Navier-Stokes Equations,* Elsevier Science Publishers, Amsterdam, 1984.
12. J. B. Bell, C. N. Dawson, and G. R. Shubin, "An Unsplit, Higher-Order Godunov Method for Scalar Conservation Laws in Multiple Dimensions," *J. Comp. Phys.*, vol. 74, pp. 1-24, 1988.
13. W. L. Briggs, in *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.

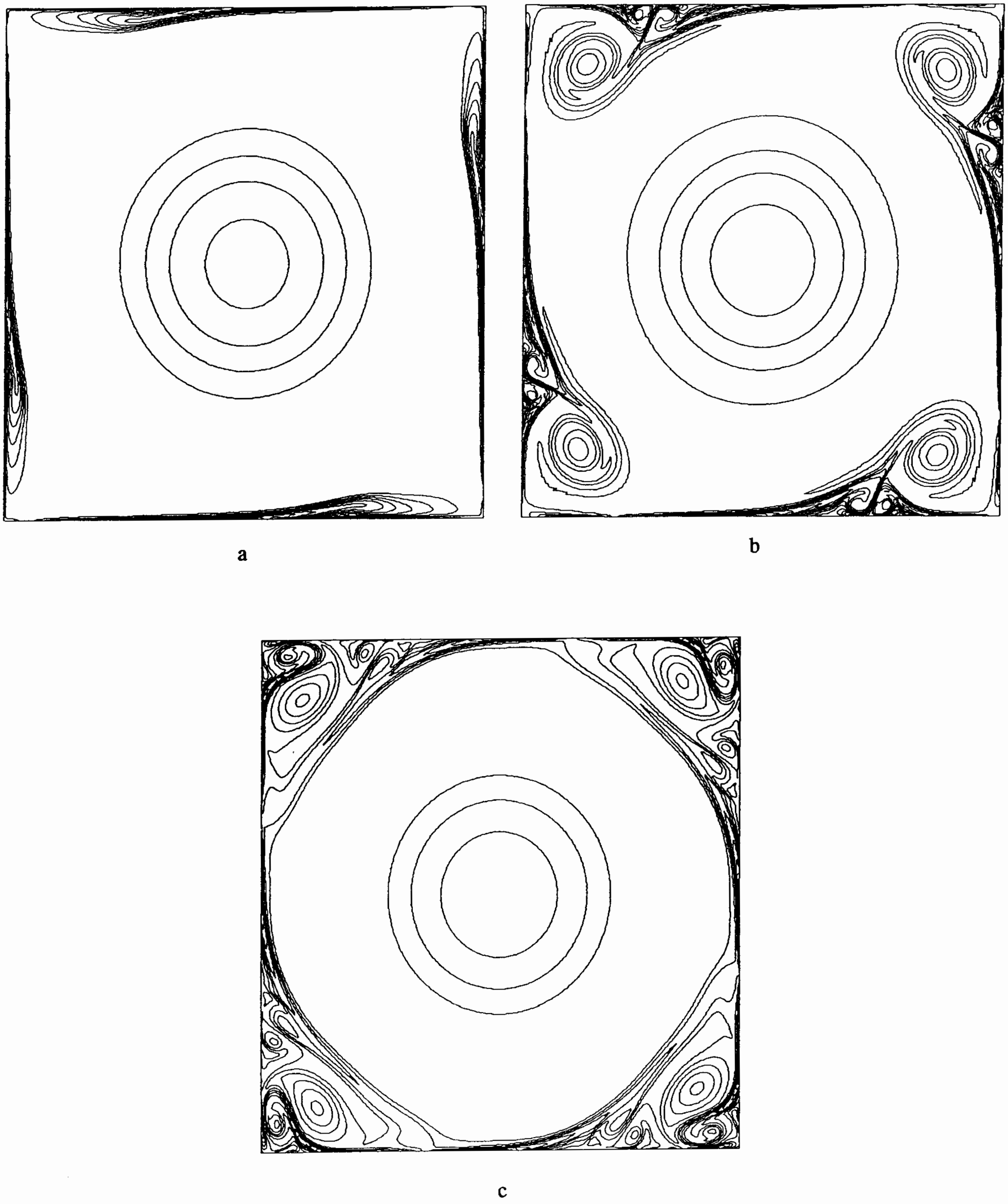

Figure 1. Spindown of a vortex in a box on a 256×256 grid. The maximum velocity in the initial data is .0686, and the length of the sides of the box is 1. Vorticity contours are shown at times 20 (a), 40 (b), and 60 (c). The total number of time steps taken is approximately 1100.

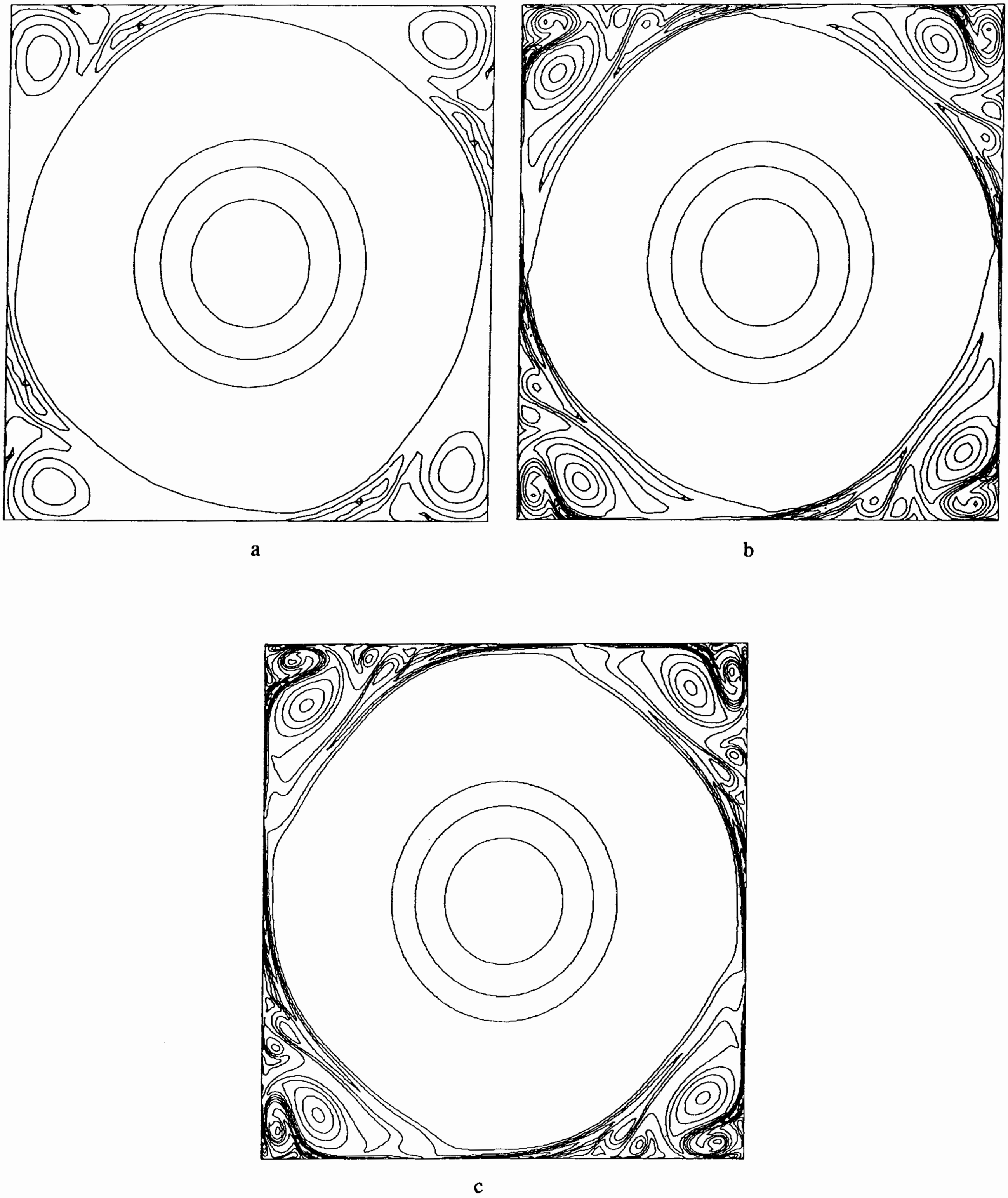

Figure 2. Spindown of a vortex in a box at time 60 on 64×64 (a), 128×128 (b) and 256×256 (c) grids.