INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **28**: 789–814 (1998)

FILTERING NON-SOLENOIDAL MODES IN NUMERICAL SOLUTIONS OF INCOMPRESSIBLE FLOWS

WILLIAM J. RIDER*

Applied Theoretical and Computational Physics Division, Hydrodynamic Methods Group,
Los Alamos National Laboratory, Mail Stop D413, Los Alamos, NM 87545, USA

SUMMARY

Solving the incompressible Navier–Stokes equations requires special care if the velocity field is not discretely divergence-free. Approximate projection methods and many pressure Poisson equation methods fall into this category. The approximate projection operator does not dampen high frequency modes that represent a local decoupling of the velocity field. For robust behavior, filtering is necessary. This is especially true in two instances that were studied: long-term integrations and large density jumps. Projection-based filters and velocity-based filters are derived and discussed. A cell-centered velocity filter, in conjunction with a vertex-projection filter, was found to be the most effective in the widest range of cases. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: filters; approximate projections; incompressible flow

1. INTRODUCTION

Among the primary challenges confronting computational fluid dynamics (CFD) will be the construction of robust solution algorithms. Here elements essential to achieving robustness for one category of flow solvers are discussed. The robustness sought implies that the method does not fail under extreme circumstances.

The concern is a class of solution algorithms known as approximate projection methods. These methods are extensions of the classic work of Chorin [1], and its modernization by Bell *et al.* [2] for solving the incompressible Navier–Stokes equations. These methods are also used to solve equations with varying density [3]. For a different viewpoint on modern projection methods, see Gresho [4]. The methods used in this paper will be semi-implicit in the sense that Gresho uses, the stability of the method is dependent upon the CFL condition on fluid velocity. These methods are applicable to problems that depend on the accurate temporal resolution of material motion.

In the course of this paper, references will be made to grid topologies. The most important of these are depicted in Figure 1. The primary grid will be collocated, but the filters will make use of both MAC and vertex staggering in their derivation.

Several comments should be made regarding some of the design choices made in the algorithms used. First, throughout the development of these methods, the robust computation

---

* Correspondence to: Applied Theoretical and Computational Physics Division, Hydrodynamic Methods Group, Los Alamos National Laboratory, Mail Stop B265, Los Alamos, NM 87545, USA. E-mail: wjr@lanl.gov

CCC 0271–2091/98/150789–26$17.50

Received 15 July 1996
Revised 28 April 1997

of flow irrespective of Reynolds number has been a requirement. To accomplish this, methods adapted from high-speed compressible flow have been adapted to the low-speed regime. An unsplit method for advection, introduced in Reference [5], was used and implemented in a manner similar to Reference [2]. The details of the actual implementation are quite close to that found in [6]. For viscous terms, a Crank–Nicholson discretization provides for a fully second-order treatment. These high-order Godunov (upwind) methods provide high resolution and robust solutions in all flow regimes. It has been found that these methods can resolve flows to a grid Reynolds number of 40, and gracefully degrade beyond this resolution, i.e. still produce physically reasonable results (see Brown and Minion for a discussion [7]). These methods provide solutions that are second-order in both time and space. No averaging is used for the convective differencing (as in most second-order MAC grid-based implementations), but rather upwinding (solving a Riemann problem) is used in conjunction with extrapolation, to control volume edges to different advection terms. These methods require that the velocities by collocated, which limits the potential grid topologies.

Additionally, the most favorable grid topology for projections, the MAC grid, is excluded because of the velocity variable location normal to cell edges. It should be noted that the author has experimented with a method similar to that developed by Tau [8] and found that the high-order Godunov methods can be extended to MAC-staggered grids. This point is not highlighted by Tau, and being contrary to conventional wisdom needs some amplification. Nonetheless, this method does not change the focus or the need for what follows. Secondly, the ultimate desire is to move these methods to curvilinear co-ordinates, and overlapping grids makes the choice of locating the other variables with the velocity quite appealing while the MAC grid is problematic.

Traditional projections on grids with collocated velocities are plagued by decoupling errors because of the discrete Laplacian (or the pressure Poisson equation) has a multidimensional null space. This corresponds to a loss of the discrete analog of elliptic regularity [9]. This decoupling also causes problems for modeling physics such as combustion [10,11] and iterative solution techniques such as multigrid [12].

A modification of the standard projection techniques are used where the velocity field is not discretely solenoidal. The term 'approximate projection' was coined by Almgren *et al.* [13] for a vertex-staggered grid (see Figure 1). The term 'approximate' refers to the characteristic that the flow field is not discretely divergence-free, but is approximately divergence-free. The divergence is a function of truncation error. Lai *et al.* [11] then extended the concept to

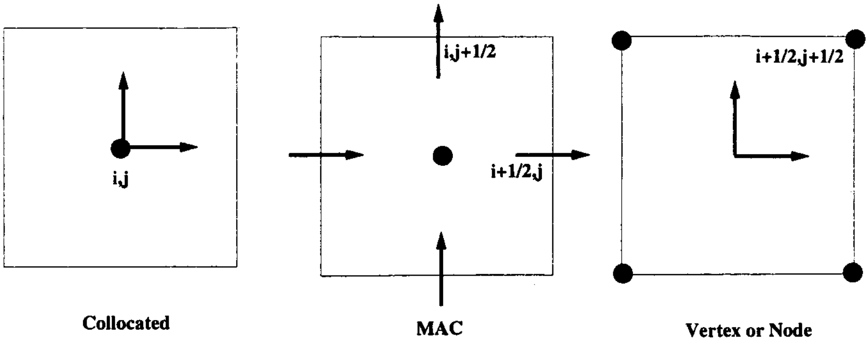


Figure 1. The major types of grids with positioning of variables is shown. The concentration will be on collocated grids, but the other two grids play an important role in the methodology.

collocated grids and combustion problems. In Reference [11], it was found that certain error modes hampered the solution through interacting with the combustion source terms. They derived a filter based on the vertex-staggered projection to damp these error modes. Since then, it has been found necessary to require more extensive filtering for a wider variety of flow problems.

Two cases are examined in this paper. First, an inviscid drop with a 1000:1 density ratio and secondly, a long-time integration of the Euler equations of constant density flow. The finding was that a cell-centered velocity filter is the single most effective filter, but it is much more effective when combined with other filters. These filters acting in concert greatly improve the results.

The remainder of the paper is organized into five sections. Next, the basics of projection methods, followed by their implementation in the form of approximate projections will be discussed. Following this, other details of the implementation are discussed. Then, the formulation and details of the various filters, followed by results detailing their relative effectiveness are discussed. Finally, the results will be summarized and their generalizations discussed.

2. PROJECTION METHODS

Projection methods advance a velocity field, $\mathbf{V} = (V^x, V^y)^T$ by some convenient means disregarding the solenoidal nature of $\mathbf{V}$, then recover the proper solenoidal velocity field, $\mathbf{V}^d$ ($\nabla \cdot \mathbf{V}^d = 0$). This uses a projection, $\mathscr{P}$, which has the effect

$$\mathbf{V}^d = \mathscr{P}(\mathbf{V}).$$

The projection accomplishes this through the decomposition of the velocity field into parts that are divergence-free and curl-free. This is also known as a Hodge, or Helmholtz decomposition [14]. The curl-free portion will be denoted by the gradient of a potential, $\nabla\varphi$. This decomposition can be written

$$\mathbf{V} = \mathbf{V}^d + \nabla\varphi. \tag{1}$$

This equation holds the key for computing solutions to incompressible flow equations.

Taking the divergence of Equation (1) gives

$$\nabla \cdot \mathbf{V} = \nabla \cdot \mathbf{V}^d + \nabla \cdot \nabla\varphi \rightarrow \nabla \cdot \mathbf{V} = \nabla \cdot \nabla\varphi.$$

Once φ has been computed, then the solution can be found through

$$\mathbf{V}^d = \mathbf{V} - \nabla\varphi. \tag{2}$$

Now the projection operator will be written based on practical steps toward the solution given above

$$\mathscr{P} = \mathbf{I} - \nabla(\nabla \cdot \nabla)^{-1}\nabla.$$

One can also write $\nabla\varphi = \mathscr{Q}(\mathbf{V})$ where $\mathscr{Q} = \mathbf{I} - \mathscr{P}$. One of the most important aspects of projections is that they are idempotent, i.e. $\mathscr{P}^2 = \mathscr{P}$. Also, it can be shown that the norm of the operator is less than or equal to one, $\|\mathscr{P}(\mathbf{V})\|_2 \leq \|\mathbf{V}\|_2$. Thus, it is easy to prove that the operator is stable [15].

It is straightforward to apply this to flows that are incompressible where the density varies. For variable density flows Equation (1) is written

$$\rho\mathbf{V} = \rho\mathbf{V}^d + \nabla\varphi,$$

or

$$\mathbf{V} = \mathbf{V}^d + \sigma\nabla\varphi,$$

where $\sigma = 1/\rho$. In a straightforward fashion, the elliptic equation for φ is

$$\nabla\cdot\mathbf{V} = \nabla\cdot\sigma\nabla\varphi. \tag{3}$$

The correction equation is

$$\mathbf{V}^d = \mathbf{V} - \sigma\nabla\varphi.$$

The projection operators now become

$$\mathscr{P}_\sigma = \mathbf{I} - \sigma\nabla(\nabla\cdot\sigma\nabla)^{-1}\nabla.$$

and

$$\mathscr{Q}_\sigma = \mathbf{I} - \mathscr{P}_\sigma.$$

In order for the above discussion to be useful for computing flows, a discrete analog to the operators above must be defined. Let D be the discrete divergence, and $\mathbf{G}$, the discrete gradient. It is required that the operators be skew adjoint, $D = -\mathbf{G}^T$, or in an inner product,

$$(D\mathbf{V}, \varphi)_\mathrm{s} = -(\sigma\mathbf{G}\varphi, \mathbf{V})_\mathrm{v},$$

where (,) is a discrete inner product over grid cells with s and v referring to scalar and vector products, respectively. The vector inner product is density weighted

$$(\mathbf{V}_1, \mathbf{V}_2)_\mathrm{v} = \sum_i \sum_j V_{1,i,j} V_{2,i,j} \rho_{i,j}.$$

It turns out that standard second-order finite difference forms for the divergence and gradient meet the above criteria. Unfortunately, the method experiences a fourfold decoupling.

At this stage approximate projections will be introduced. These methods will have more familiar discrete Laplacians, and will be more amenable to powerful solution techniques such as multigrid. The property that the solutions are discretely divergence-free will be lost.

3. APPROXIMATE PROJECTIONS

The exact discrete projections given in the previous section provide a good foundation in the numerical implementation of projection methods, but have some practical difficulties. While the exact discrete projections provide high quality solutions, they also have problems. The pressure solutions are decoupled for grids using collocated velocity fields (this is not a problem for the MAC grid). This decoupling can cause oscillations in problems involving local source terms such as in combustion [11,13]. Additionally, the local decoupling makes efficient linear algebra techniques cumbersome [12] as well as hampering the implementation of adaptive grid techniques [12,16].

Discrete methods based on the continuous projections are defined rather than demanding that the discrete system algebraically match the conditions for being a projection (as in an exact projection). Thus, the most straightforward means to discretize each operator ($\nabla\cdot$, ∇ and $\nabla\cdot\sigma\nabla$) will be chosen[1]. This makes these methods functionally quite similar to pressure

---

[1] This is nearly true, but some differences exist between the obvious form for $\nabla\cdot\sigma\nabla$ from the continuous case and the approximate projection as explained in [17].

Poisson equation methods, thus most of the techniques discussed here can be applied with appropriate generalizations. It is expected that a successful discretization will converge to the same solutions as the exact projection, and the discrete divergence will converge to zero as the grid is refined.

Because the same difference forms are still used (as for exact projections), divergence and gradient velocities can still experience decoupling. Controlling this decoupling is the purpose of filters.

4. PROJECTION ALGORITHMS FOR INCOMPRESSIBLE FLOW

Given the development above, projection algorithms can now be outlined. The algorithm will consist of two steps: a separate step where the solenoidal nature of the velocity field is ignored and a corrective step using the projection to force the velocity to be approximately solenoidal. This is often referred to as a 'fractional step method'. In the first step, the velocity is computed solving the motion equations as convection–diffusion equations

$$\mathbf{u}^{*,n+1}_{i,j} = \mathbf{u}^{n}_{i,j} - \Delta t\left[(\mathbf{u}\cdot\nabla\mathbf{u})^{n+1/2}_{i,j} + \sigma^{n+1/2}_{i,j}\mathbf{G}_{i,j}\phi^{n-1/2} - \frac{\nu\sigma^{n+1/2}_{i,j}}{2} L_{i,j}(\mathbf{u}^{n} + \mathbf{u}^{*,n+1}) - \mathbf{f}^{n+1/2}_{i,j}\right] \tag{4}$$

where $\sigma^{n+1/2}_{i,j} = (\sigma^{n}_{i,j} + \sigma^{n+1}_{i,j})/2$, $\mathbf{f}^{n+1/2}_{i,j} = (\mathbf{f}^{n}_{i,j} + \mathbf{f}^{n+1}_{i,j})/2$, or $\mathbf{f}^{n+1/2}_{i,j} = \mathbf{f}(x_{i,j}, t^{n+1/2})$, and $L_{i,j}$ is the discrete Laplacian (typically the standard five point operator in two dimensions). This general form of a projection was favored by Gresho in [4]. This form of a projection algorithm is also studied here because of its extensive use, [10,11,13,16].

The pressure equation can be formed using a projection formulation. The right-hand side of this equation involves taking the discrete divergence of the velocity field. As the results in Section 6 will demonstrate, the specific form chosen can influence the quality of the results. The 'standard' form chosen in References [10,11,13,16], forms the projection as follows

$$D\sigma^{n+1/2}\mathbf{G}\varphi = D\left(\frac{\mathbf{u}^{*,n+1} - \mathbf{u}^{n}}{\Delta t}\right). \tag{5}$$

The method becomes an approximate projection when $D\sigma^{n+1/2}\mathbf{G}\varphi$ is replaced by a variable coefficient Laplacian operator, $L_{\sigma}\varphi$ in Equation (5). The correction to the velocity field produced by the first step of the algorithm is,

$$\mathbf{u}^{n+1} = \mathbf{u}^{*,n+1} - \Delta t\sigma^{n+1/2}\mathbf{G}\varphi.$$

The pressure is then computed as

$$\phi^{n+1/2} = \phi^{n-1/2} + \frac{\varphi}{\Delta t}.$$

The filters developed and described in the next section are applied to the final velocity from the approximate projection, thus producing a more robust noise-free velocity field.

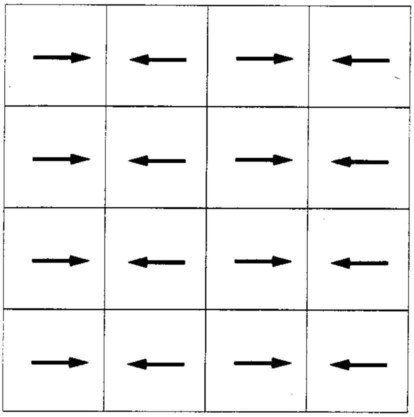


Figure 2. This shows the one-dimensional error mode that can persist with cell-centered divergence operator. This mode can also align itself in the other co-ordinate direction.

5. FILTERS

The filters given here are a solution to the new problems posed by the approximate satisfaction of the solenoidal condition. In another sense, the filters are also related to a problem left over from the exact projection, namely that the cell-centered divergence operator does not recognize a number of obvious non-divergence-free modes. These modes are commonly referred to as 'checkerboard' modes [18].

This can be restated as a mode in the null-space of the operator $D\mathbf{V}$. One remedy for this is to use other divergence operators to pick these modes out of solutions and diffuse them. You could certainly define an entire projection to remove these from the solution, but this would be overkill (and be first-order-accurate). The projection apparatus will be used here to define a procedure that will stably diffuse the non-divergence-free modes in the solution. Another remedy is to remove the errors via a velocity filter. In each case, the removal of these modes should not effect the accuracy or stability of the underlying method.

5.1. Motivation

First, the four major types of error modes seen in calculations will be described.

1. The simplest is primarily one-dimensional in nature and manifests itself as shown in Figure 2. This is called a line mode.
2. The second is shown in Figure 3 and naturally evokes a vertex-based divergence operator. This is a classic checkerboard mode.
3. The third mode considered is a diagonal mode as shown in Figure 4. It is notable that this mode cannot be seen with either the cell-centered or vertex-centered divergences. This is a diagonal line mode.
4. The final mode is shown in Figure 5. Modes of this type can be seen by the vertex divergence and one of the two edge-centered divergences (each of these discrete divergences will be given shortly).

The root-cause modes can all be described as linear combinations of the basic one-dimensional modes. These are shown in Figure 6. The modes given in Figures 2–5 are simply the same pattern for a mode repeated in some region of cells. A fundamental filter must recognize the one-dimensional basic modes in order to be completely effective.

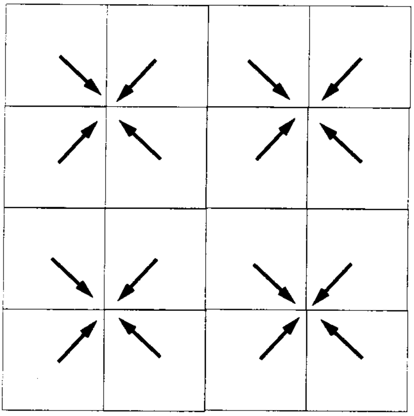


Figure 3. This shows the nature of the vertex mode that is not seen by the cell-centered divergence operator. This is picked up quite naturally by the vertex divergence operator.

The null-space of the operator can be seen with its Fourier symbol. The number of zeros of the symbol in the range $\alpha \in [0, \pi]$ shows the dimension of the null-space. Plotting the symbol of the cell-centered divergence operator (Figure 7) shows that the null-space has a dimension of four, with three being located in the highest frequency of the grid, thus adjacent grid points are decoupled (the other is at the lowest frequency on the grid).

5.2. Projection filters

Projection filters are designed to diffuse divergent modes shown in Figures 2–5 through the use of an unconverged projection operator. The projection operators used as the basis of the filter have the same order of accuracy as the primary projection operator, formal order of accuracy is not affected. Thus, with a consistent discretization of the filter, it should leave the accuracy of the underlying second-order method intact. Additionally, these operators will only effect the flow in cells where the discrete divergence is relatively large.

Symbolically, the filter will act on the advance time velocity field only,

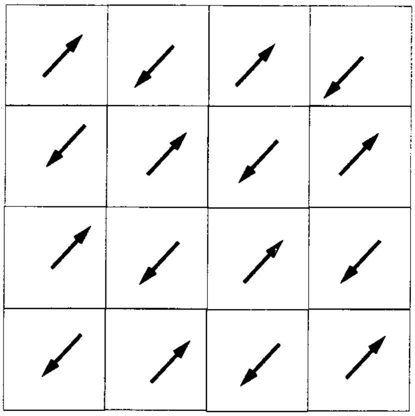


Figure 4. This shows the diagonal mode of that which is not seen by either the cell-centered or vertex-centered divergence operator. It is both visualized and diffused using an edge-centered divergence operator.

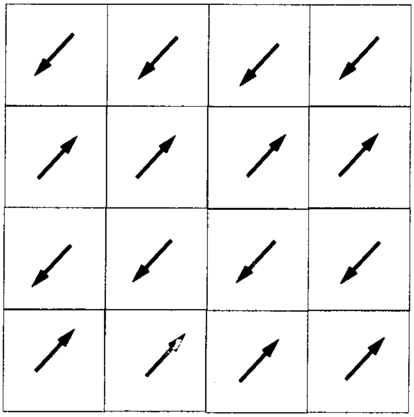


Figure 5. This mode can be seen with a vertex-based divergence and one of the two edge divergences. It is invisible to the other. In this case, the divergence is invisible to the x-edge-based divergence.

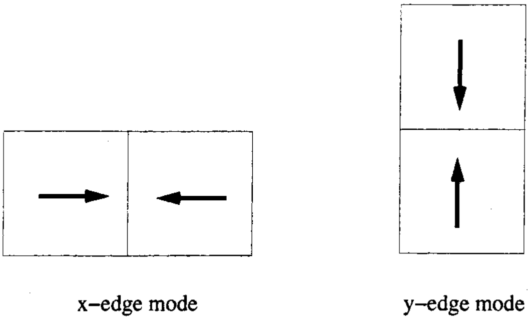


Figure 6. This shows the modes that can be used to define all non-divergence-free modes occurring with cell-centered projection as defined in this paper.

$$\mathbf{u}^{n+1} = \mathscr{F}(\tilde{\mathbf{u}}^{n+1}),$$

where $\tilde{\mathbf{u}}^{n+1}$ is the product of the preceding projection. The projection is then defined by a divergence–gradient operator pair, D and $\mathbf{G}$. Recalling that the projection can be written as $\mathbf{P} = \mathbf{I} - \sigma\mathbf{G}(D\sigma\mathbf{G})^{-1}D$, it will be written as a diffusion-like operator by replacing $(D\sigma\mathbf{G})^{-1}$ by

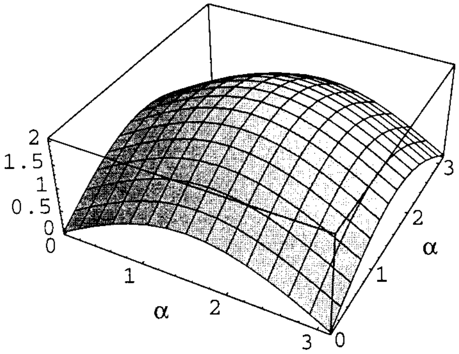


Figure 7. The symbol of the cell-centered divergence operator is plotted for α_x, $\alpha_y \in [0, \pi]$. The null-space shows itself where the symbol is zero at each of the nearest grid points.

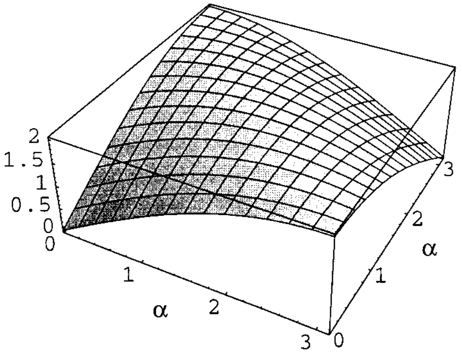


Figure 8. The symbol of the vertex-centered divergence operator is plotted for α_x, $\alpha_y \in [0, \pi]$. This divergence couples the adjacent cells in the x-and y-directions, but leaves the diagonal cell decoupled.

a diagonal operator. This reduces the projection to a single diffusion pass with a pseudo-time step set by the diagonal of the discretized elliptic operator $D\sigma\mathbf{G}$. This diagonal term will be called L^d_σ. The application of this form of a projection is like a Jacobian relaxation and has similar properties in that high frequency error components are effectively damped [19].

A filtering algorithm proceeds as follows.

Algorithm 1 (*projection filter*)

1. Initialize φ to zero.
2. Begin with a velocity, $\tilde{\mathbf{u}}^{n+1}$, and compute its divergence $D\tilde{\mathbf{u}}^{n+1}$.
3. Compute $\varphi = (L^d_\sigma)^{-1} D\tilde{\mathbf{u}}^{n+1}$.
4. Update velocity to get $\mathbf{u}^{n+1} = \tilde{\mathbf{u}}^{n+1} - \sigma^{n+12}\mathbf{G}\varphi$.

This can be given symbolically as $\mathscr{F} = \mathbf{I} - \sigma^{n+1/2}\mathbf{G}(L^d_\sigma)^{-1}D$.

The vertex-projection filter was originally developed to solve combustion problems, where the divergent mode interacted poorly with source terms [10,11]. This filter is based on divergence–gradient operators, where the velocities and gradients are cell-centered, but the divergence is vertex-centered. This requires that the filter potential field be vertex-centered. The discrete divergence is

$$D_{i+1/2,j+1/2}\mathbf{V} = \frac{V^x_{i+1,j} + V^x_{i+1,j+1} - V^x_{i,j} - V^x_{i,j+1}}{2\Delta x} + \frac{V^y_{i,j+1} + V^y_{i+1,j+1} - V^y_{i,j} - V^y_{i+1,j}}{2\Delta y}, \tag{6a}$$

and the discrete gradient is

$$\sigma\mathbf{G}_{i,j}\varphi = \begin{pmatrix} \dfrac{\varphi_{i+1/2,j+1/2} + \varphi_{i+1/2,j-1/2} - \varphi_{i-1/2,j-1/2} - \varphi_{i-1/2,j+1/2}}{2\rho_{i,j}\Delta x} \\ \dfrac{\varphi_{i+1/2,j+1/2} + \varphi_{i-1/2,j+1/2} - \varphi_{i+1/2,j-1/2} - \varphi_{i-1/2,j-1/2}}{2\rho_{i,j}\Delta y} \end{pmatrix}. \tag{6b}$$

The diagonal term is defined by the discrete Laplacian and is

$$L^{d}_{\sigma,i+1/2,j+1/2} = -\frac{\sigma_{i,j+1/2}+\sigma_{i+1,j+1/2}}{\Delta x^2} - \frac{\sigma_{i+1/2,j}+\sigma_{i+1/2,j+1}}{\Delta y^2}.$$

It is not difficult through Fourier to establish analysis that this makes for a stable algorithm [10]. With these operators specified, their application is straightforward.

The edge-centered filters should be implemented for both the z- or y-edges of the grid. The values of the velocities on the normal edges of the divergence cell are simple to compute, but the tangential velocities will require averaging. The filter potential field computed will be on edges, thus the normal velocity corrections will be simple compact stencils.

For the x-edges, the operators are

$$D_{i+1/2,j}\mathbf{V} = \frac{V^x_{i+1,j}-V^x_{i,j}}{\Delta x} + \frac{V^y_{i,j+1}+V^y_{i+1,j+1}-V^y_{i,j-1}-V^y_{i+1,j-1}}{4\Delta y},$$

and the gradient is

$$\sigma\mathbf{G}_{i,j}\varphi = \begin{pmatrix} \dfrac{\varphi_{i+1/2,j}-\varphi_{i-1/2,j}}{\rho_{i,j}\Delta x} \\ \dfrac{\varphi_{i+1/2,j+1}+\varphi_{i-1/2,j+1}-\varphi_{i-1/2,j-1}-\varphi_{i-1/2,j-1}}{4\rho_{i,j}\Delta y} \end{pmatrix}.$$

The diagonal operator is

$$L^{d}_{\sigma,i+1/2,j} = -\frac{\sigma_{i,j}+\sigma_{i+1,j}}{\Delta x^2} - \frac{\sigma_{i+1/2,j-1/2}+\sigma_{i+1/2,j+1/2}}{\Delta y^2},$$

For the y-edges, the operators are

$$D_{i,j+1/2}\mathbf{V} = \frac{V^x_{i+1,j}+V^x_{i+1,j+1}-V^x_{i-1,j}-V^x_{i-1,j+1}}{4\Delta x} + \frac{V^y_{i,j+1}-V^y_{i,j}}{\Delta y},$$

and the gradient is

$$\sigma\mathbf{G}_{i,j}\varphi = \begin{pmatrix} \dfrac{\varphi_{i+1,j+1/2}+\varphi_{i+1,j-1/2}-\varphi_{i-1,j-1/2}-\varphi_{i-1,j-1/2}}{4\rho_{i,j}\Delta x} \\ \dfrac{\varphi_{i,j+1/2}-\varphi_{i,j-1/2}}{\rho_{i,j}\Delta y} \end{pmatrix}.$$

The diagonal operator is

$$L^{d}_{\sigma,i,j+1/2} = -\frac{\sigma_{i-1/2,j+1/2}+\sigma_{i+1/2,j+1/2}}{\Delta x^2} - \frac{\sigma_{i,j}+\sigma_{i,j+1}}{\Delta y^2},$$

With a similar expression for the y-edge filter (with appropriate terms modified).

In applying the edge-projection filter, the correction should only be applied to the velocity in the normal direction of the edge divergence. This is because the symbol of $\sigma\mathbf{G}D$ in the transverse direction is quite similar to the cell-centered operator; one would expect this operation to be no good for the solution. Also, the edge-based filters should be applied to the same starting velocity field and the corrections should not be applied until both edge filters have had their velocity corrections computed (to preserve symmetry).

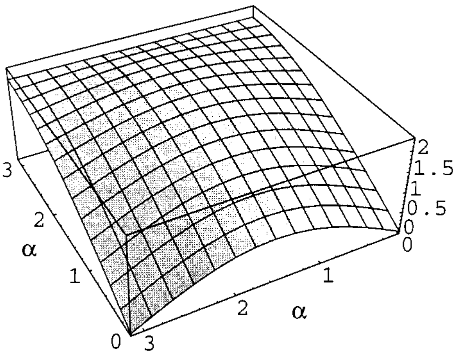


Figure 9. The symbol of the x-edge-centered divergence operator is plotted for α_x, $\alpha_y \in [0, \pi]$. The diagonal and x-direction cells are coupled, but the y-direction cells are decoupled. This shows the symbol for both x- and y-edge divergences by switching the α's.

By studying the symbols of the filter divergences, one can predict their impact on the solution. The symbols are plotted in Figures 8 and 9. The combination of these filters will couple all grid points, but the application of one of them will leave part of the decoupling intact. It should also be noted that Fourier analysis confirms that each of the projections given above is second-order-accurate.

5.3. Velocity filters

A completely different formulation for filters can be defined on criteria that uses physical intuition. These are motivated from work with Lagrangian hydrocodes, where elements can be deformed by 'hourglass' modes that are not physical. These modes are associated with degrees of freedom in an operator that are not associated with physical forces. This idea is based on the work of Margolin and Pyun [20] (a discussion is also given by Benson [21]).

The basic idea is to define a complete linear space that contains both physical and non-physical modes. The non-physical modes will be the null-space of the incomplete space defined by the physical modes. The non-physical modes can then be subtracted from the velocity field to filter the solution. This operation should leave the physical modes present in the solution undisturbed.

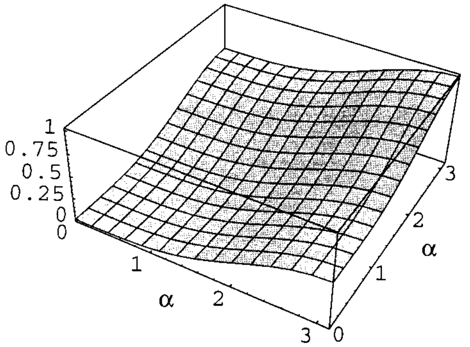


Figure 10. The symbol of the velocity filter operator is plotted for α_x, $\alpha_y \in [0, \pi]$. The filter effectively couples the adjacent grid points (i.e. has no destructive null-space associated with them).

The physical modes for two-dimensional flow are two translational modes, two shear modes and two stretching modes. These can be stated as operators, $\bar{u}$, $\bar{v}$, $\partial u/\partial x$, $\partial v/\partial y$, $\partial u/\partial y$, and $\partial v/\partial x$. The finite difference operators defining these operators discretely, will define a portion of the vector space, S.

The vertex-projection filter has a two-dimensional null-space that can be interpreted as hourglass modes in the solution. These modes can be removed with a filter like that defined by Reference [20]. A vertex-based divergence (cell) has four velocity vectors and thus eight degrees of freedom in two dimensions. With only six physical modes, two modes are left to define the hourglass modes.

Following [20], one has the velocities,

$$\mathbf{u}^T = (u_{i,j}, u_{i+1,j}, u_{i,j+1}, u_{i+1,j+1}, v_{i,j}, v_{i+1,j}, v_{i,j+1}, v_{i+1,j+1}),$$

defined on a divergence element. For example, the average velocity modes are

$$\mathbf{S}_1^T = \frac{1}{4}(1, 1, 1, 1, 0, 0, 0, 0), \qquad \mathbf{S}_2^T = \frac{1}{4}(0, 0, 0, 0, 1, 1, 1, 1).$$

Other modes can be given in a straightforward fashion, with two null-space modes. This gives a complete basis for $\mathbf{u}$. With appropriate normalization, the velocity can be expressed as

$$\mathbf{u} = \sum_{i=1}^{8} \mathbf{S}_i^T \mathbf{u} \mathbf{S}_i.$$

The filter then can be constructed to remove the hourglass modes

$$\tilde{\mathbf{u}} = \mathbf{u} - C \sum_{i=7}^{8} \mathbf{S}_i^T \mathbf{u} \mathbf{S}_i.$$

In Reference [20], C has a maximum value of 0.25 given by stability considerations. This filter has a differential analog of diffusing the velocities with the operators, u_{xyxy} and v_{xyxy}.

Unlike Lagrangian hydrodynamics, this paper has a cell-centered grid and does not have a readily defined element on which to define a set of velocities. This set of velocities will be taken as those needed to define a divergence operator. This gives ten degrees of freedom, with six physical modes, thus leaving four non-physical modes to be defined. These velocities are

$$\mathbf{u}^T = (u_{i-1,j}, u_{i,j}, u_{i+1,j}, u_{i,j-1}, u_{i,j+1}, v_{i-1,j}, v_{i,j}, v_{i+1,j}, v_{i,j-1}, v_{i,j+1}).$$

As before, various operators will then be defined by $\mathbf{S}_i^T \mathbf{u}$. The average velocity can be given by

$$\mathbf{S}_1^T = \frac{1}{5}(1, 1, 1, 1, 1, 0, 0, 0, 0, 0), \qquad \mathbf{S}_2^T = \frac{1}{5}(0, 0, 0, 0, 0, 1, 1, 1, 1, 1).$$

The shearing modes portion of S are

$$\mathbf{S}_3^T = \frac{1}{2\Delta x}(-1, 0, 1, 0, 0, 0, 0, 0, 0, 0), \qquad \mathbf{S}_4^T = \frac{1}{2\Delta y}(0, 0, 0, 0, 0, 0, 0, 0, -1, 1).$$

The stretching modes portion of S are

$$\mathbf{S}_5^T = \frac{1}{2\Delta y}(0, 0, 0, -1, 10, 0, 0, 0, 0), \qquad \mathbf{S}_6^T = \frac{1}{2\Delta x}(0, 0, 0, 0, 0, -1, 0, 1, 0, 0).$$

Once the null-space of S is found, the original velocity field can be expanded in terms of $\mathbf{S}_i$. The original set of velocities can be found from

$$\mathbf{u} = \sum_{i=1}^{10} \mathbf{S}_i^T \mathbf{u} \mathbf{S}_i.$$

with S being normalized so that $S^T S = I$ (i.e. $S^T = S^{-1}$). Because the components of velocity corresponding to the null-space are not physical (motions are not related to physical forces, thus are spurious) one should remove or lessen their participation in the solution. This can be done through subtracting them from the velocity field

$$\tilde{\mathbf{u}} = \mathbf{u} - C \sum_{i=7}^{10} \mathbf{S}_i^T \mathbf{u} \mathbf{S}_i. \tag{7}$$

For the cell-centered velocity field defined above, the null-space is

$$\mathbf{S}_7^T = (1, -2, 1, 0, 0, 0, 0, 0, 0, 0), \qquad \mathbf{S}_8^T = (0, -2, 0, 1, 1, 0, 0, 0, 0, 0),$$
$$\mathbf{S}_9^T = (0, 0, 0, 0, 0, 1, -2, 1, 0, 0), \qquad \mathbf{S}_{10}^T = (0, 0, 0, 0, 0, 0, -2, 0, 1, 1).$$

If these expressions are substituted into Equation (7), it is found that it looks like the finite difference expressions for u_{xxxx}, u_{yyyy}, v_{xxxx} and v_{yyyy}. Thus, this filter is equivalent to adding a fourth-order diffusion to the solution. Because of this finite difference form, this modification should not impact the order of accuracy for the overall scheme. Heuristic analysis of checkerboard errors suggests that a coefficient should multiply the diffusion terms of the form

$$C = \frac{h^4}{32},$$

where $h = \Delta x$ or Δy. Fourier analysis confirms that this coefficient is necessary for linear stability of this operation. The symbol of the operator (scaled by C) is shown in Figure 10.

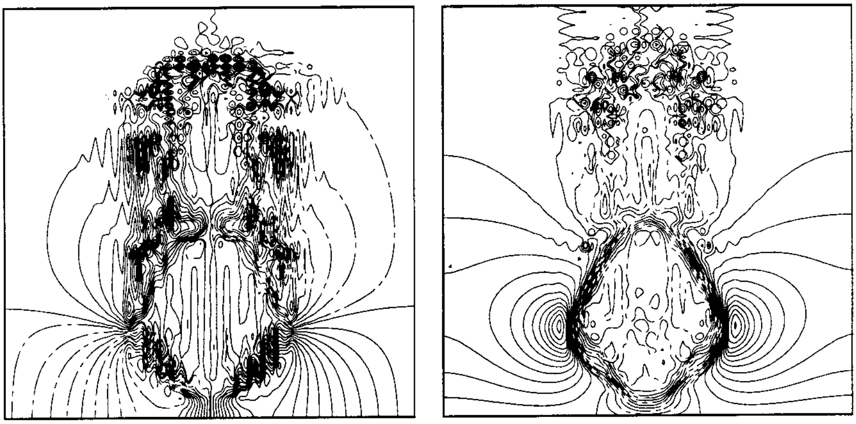
During the development of this filter, it was found that applying it to the entire flow field, while effective in controlling the discrete divergence, it also caused the solution quality to be degraded. As a consequence, this filter is applied only in the regions of the flow where the dimensionless divergence is large. A second coefficient is defined that gives the largest tolerable ratio of divergence to velocity in the flow field, C_D. Then, the edge-based divergences, introduced earlier, are used to define the local divergence. The local characteristic velocity is chosen to be the largest velocity in a cell. For the z-direction diffusion, z-edge divergences are used and for the y-direction, the y-edge divergences are used. The value of the coefficient is then limited for stability.

The leading coefficient on the velocity filter is then

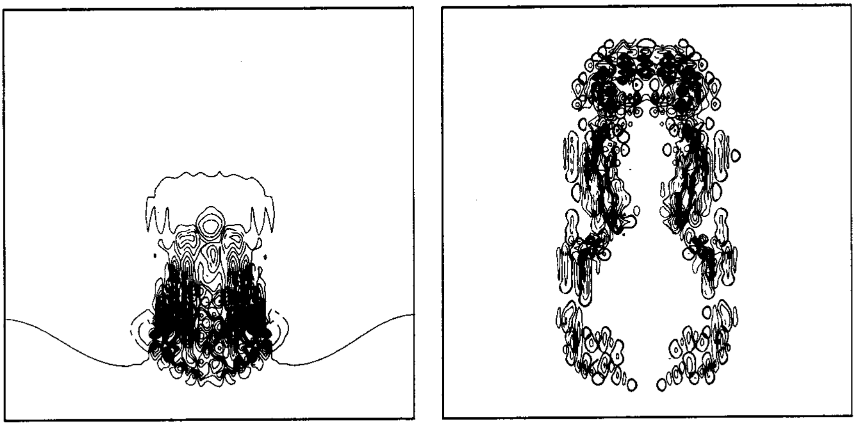
$$C_{i,j}^x = \frac{\Delta x^4}{32} \min\left[1,\, C_D \frac{(|D_{i-1/2,j}\mathbf{u}| + |D_{i+1/2,j}\mathbf{u}|)\Delta x}{\max(|u_{i,j}|, |v_{i,j}|)}\right],$$

with the y-direction given analogously. C_D is taken to be between 1 and 10 typically (4 is the choice for the calculations that follow). Just as for the projection-based filters, this step is applied at the end of a computational cycle to the advanced time velocities (it has been found best to use the velocity filters after the projection filters, should they be used in tandem).

There is also a need to consider the impact of physical viscosity on this filter. Physical viscosity should operate in a similar fashion to this filter through the coupling of adjacent grid cells effectively. When viscous forces dominate, one would like to limit the impact of this velocity filter accordingly. The velocity filter aims to couple cells that are decoupled by the divergence operator. The degree of damping applied by the viscous forces to the highest frequency errors (i.e. those operating locally) can be computed by looking at the symbol of the



(a) x-velocity (b) y-velocity



(c) pressure (d) Du^{n+1}

Figure 11. Inviscid discontinuous drop results with no filter. It should be noted that the discrete divergence is not on the same scale as the other figures. Thus the quality and position of this error should be compared with the other figures.

viscous discretization. In this case, the Crank–Nicholson differencing of diffusion is analyzed. For the highest frequency (α_x, $\alpha_y = \pi$) this is

$$\Lambda_\pi^v = \frac{1 - 2\mu_x - 2\mu_y}{1 + 2\mu_x + 2\mu_y},$$

where $\mu_x = v\Delta t/\mathrm{d}x^2\rho$ and $\mu_y = v\Delta t/\Delta y^2\rho$. Then the maximum damping applied by the filter is modified by

$$C_{i,j}^{x,\mathrm{mod}} = C_{i,j}^{x} \max(0, \Lambda_\pi^v).$$

The velocity filtering algorithm would proceed as follows

Algorithm 2 (*velocity filter*)

1. Given the advance time velocity field $(\tilde{u}^{n+1}, \tilde{v}^{n+1})^T$, compute edge divergences and $C^{x,y}_{i,j}$.
2. Take viscous forces into account, and compute $C^{x,y,\mathrm{mod}}_{i,j}$.
3. Compute u_{xxxx}, u_{yyyy}, v_{xxxx}, and v_{yyyy}.
4. Update x-velocity $u^{n+1}_{i,j} = \tilde{u}^{n+1}_{i,j} - C^{x,\mathrm{mod}}_{i,j} u_{xxxx} - C^{y,\mathrm{mod}}_{i,j} u_{yyyy}$.
5. Update y-velocity $v^{n+1}_{i,j} = \tilde{v}^{n+1}_{i,j} - C^{x,\mathrm{mod}}_{i,j} v_{xxxx} - C^{y,\mathrm{mod}}_{i,j} v_{yyyy}$.

This can be given symbolically as $\mathscr{F} = \mathbf{I} - C^{x,y,\mathrm{mod}}_{i,j} \Sigma^{10}_{i-7} \mathbf{S}^T_i \mathbf{I} \mathbf{S}_i$.

5.4. *Filters with a projection*

Weaving the filters with an approximate projection method is relatively straightforward.

Algorithm 3 (*projection method plus filters*)

1. Compute $\mathbf{u}^{*,n+1}$ with the equation of motion.
2. Perform the approximate projection to get $\tilde{\mathbf{u}}^{n+1}$.
3. Correct the velocity with any projection-type filters desired.
4. Further correct the velocity field with velocity type filters.
5. Begin the next computational cycle.

Now the impact of using these filters with approximate projection algorithms will be demonstrated.

6. RESULTS

In this section results will be given that will illuminate different properties of the filters described in the preceding section.

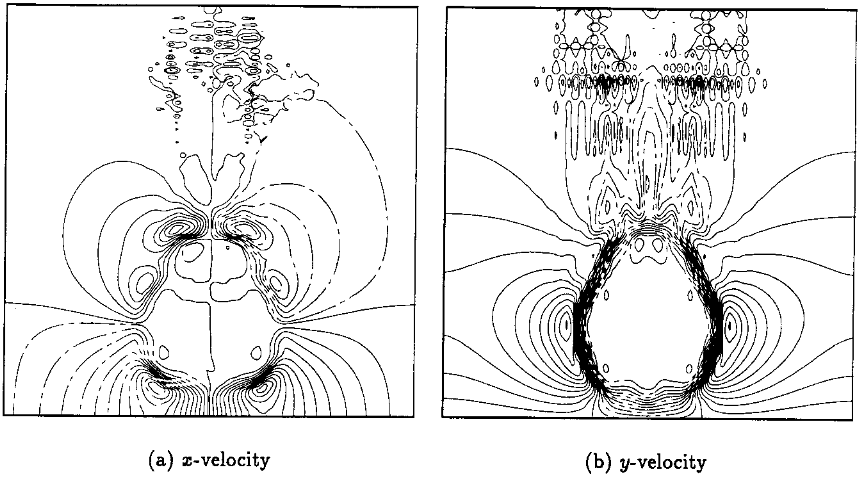
Here, an approximate projection method using collocated placement of variables, unsplit high-order Godunov advection, and the Crank-Nicholson method for diffusive processes, will be used. Some of the relevant details of the projection methods were discussed in Sections 3 and 4. A complete accounting of the methods used in this paper is given by the author in Reference [22]. The method is applicable to flows that have a time-dependent structure, best solved by the explicit solution of the convective portion of the equations. The method has been extensively tested and found to give second-order results in time and space for sufficiently resolved flows. The linear algebra is solved using a multigrid method that can handle discontinuous large jumps in density.

A set of calculations will be described that will show the performance and necessity of using the filters defined in Section 5. Next shown will be their necessity for variable density flows with large density variations and sharp density fronts and long-time integrations using the equations of constant density flow. Finally, it will also be shown that the filters have no negative impact on solution quality or accuracy (for resolved flows, the impact is positive).

The first problem used is a high density ratio drop. The drop is set in a unit square with wall boundary conditions, and has a radius of 0.15 units. Initially, it is centered at $(x_0, y_0) = (0.50, 0.75)$ initially at rest. The interface between the fluids is discontinuous (although it will smear as the computation progresses due to numerical diffusion in the density transport algorithm). The ratio of densities is 1000 (higher inside the drop) and the flow has no diffusive forces. There is a body force of -1 in the horizontal motion equation. The solutions are plotted at time, $t = 1.0$. The grid is 64^2, and the CFL number is set to 1/2. The nature of the comparison to be made is qualitative because of the discontinuous nature of the problem. This precludes the use of convergence rate estimates used on smoother problems.

With no filters, the basic approximate projection gives exceedingly poor results. This is shown in Figure 11. The resulting velocity field is shown in Figure 11(a) and (b). Both components of velocity have a 'halo' left behind where the drop was initially (before the density interface had a chance to smear). Both are noisy throughout. The pressure field (shown in Figure 11(c)) is extremely noisy inside the dense drop. The discrete divergence (i.e. error) is large and concentrated around the periphery of the original bubble position. The discrete divergence is not the largest where the velocity field is large (near the drop's current position). The divergence should only be large where the vorticity is present in the flow, which, in this case, is near the drop. Divergence errors are a combination of checkerboard and line-error modes. These results are unacceptable.

Figure 12 shows the impact of the vertex-projection filter. Again, the solution is improved, but the 'halo' effect is still prevalent, and the solution quality is poor. The divergence error is


(a) x-velocity (b) y-velocity

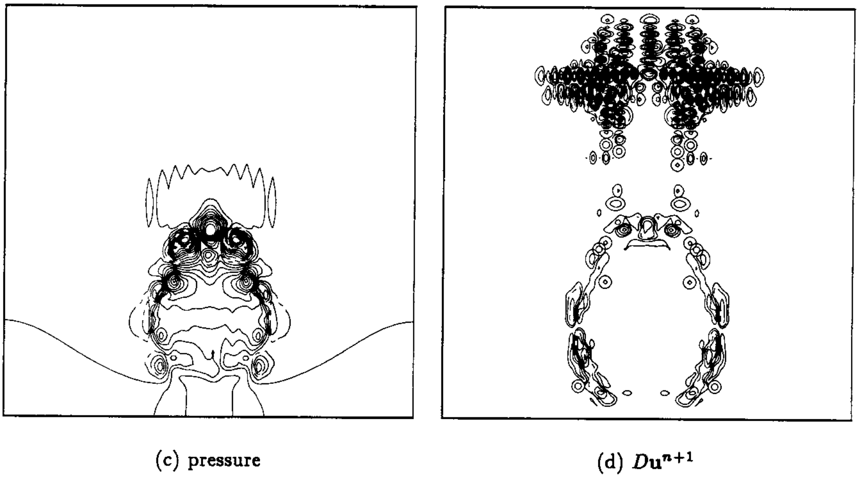
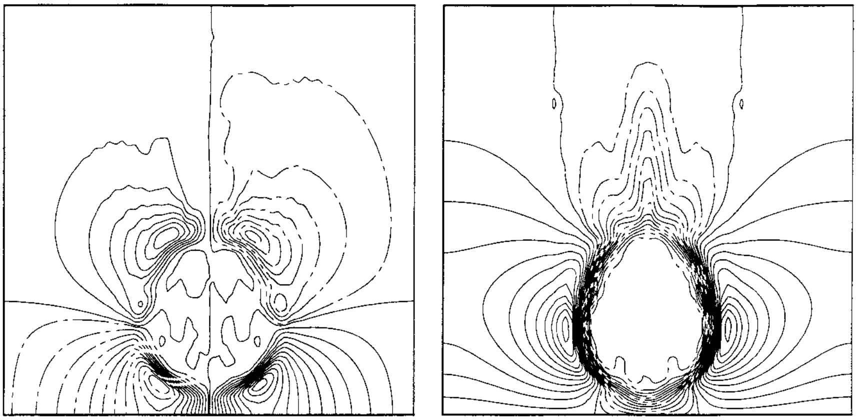
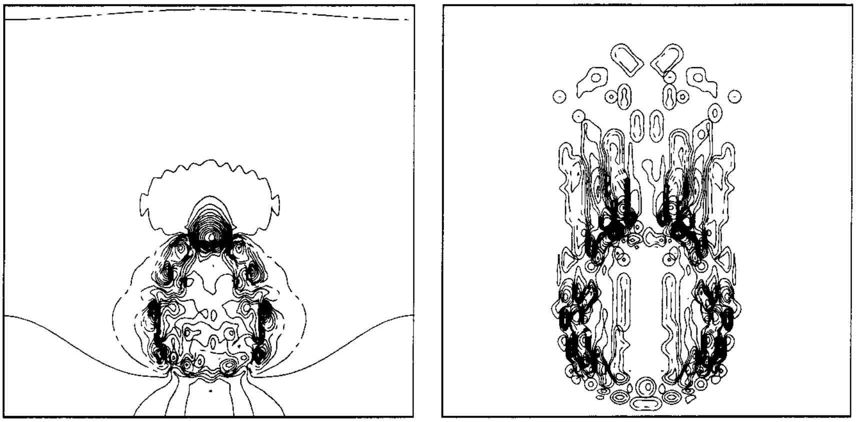

(c) pressure (d) $D\mathbf{u}^{n+1}$

Figure 12. Inviscid discontinuous drop results with a vertex-projection filter. It should be noted that the discrete divergence is not on the same scale as the other figures. Thus the quality and position of this error should be compared with the other figures. The divergences for each of the methods is shown in Figure 17



(a) x-velocity (b) y-velocity



(c) pressure (d) $D\mathbf{u}^{n+1}$

Figure 13. Inviscid discontinuous drop results with a cell-centered velocity filter.

again concentrated away from the drop, near the initial drop position. The line-error modes have been largely suppressed, but other checkerboard modes remain. These modes are related to those not detected by the vertex divergence.

The cell-centered velocity filter produces results of improved quality as shown in Figure 13. The 'halo' effect is largely alleviated. The flow field away form 'halo' region is not as high quality (noise-free) as the vertex-projection filter solutions. Nevertheless, the value of the discrete divergence is much lower than before, but the quality of the solution is still unacceptable. This is most clearly seen by looking at the pressure field (see Figure 13(c)) where there is a great deal of noise inside the drop.

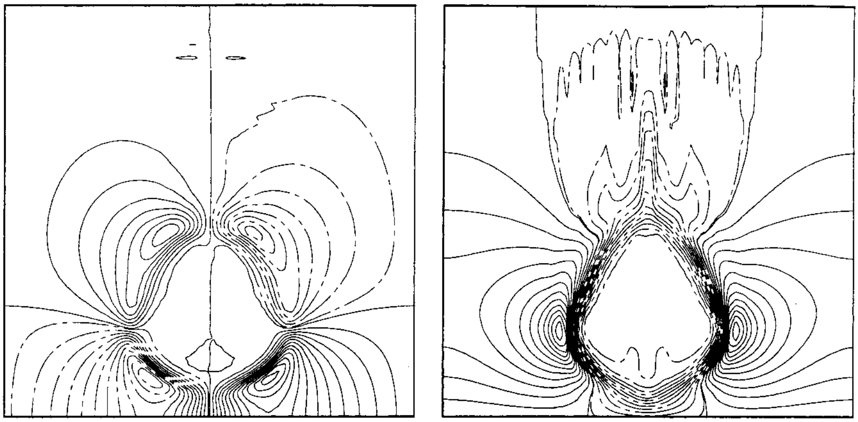
Next shown is the impact of using the combination of vertex- and edge-projection filters. These results are shown in Figure 14. This shows a large improvement in the solution quality.

The divergence error is concentrated around the current location of the drop. The 'halo' effect is largely suppressed, and the velocities are much smoother.

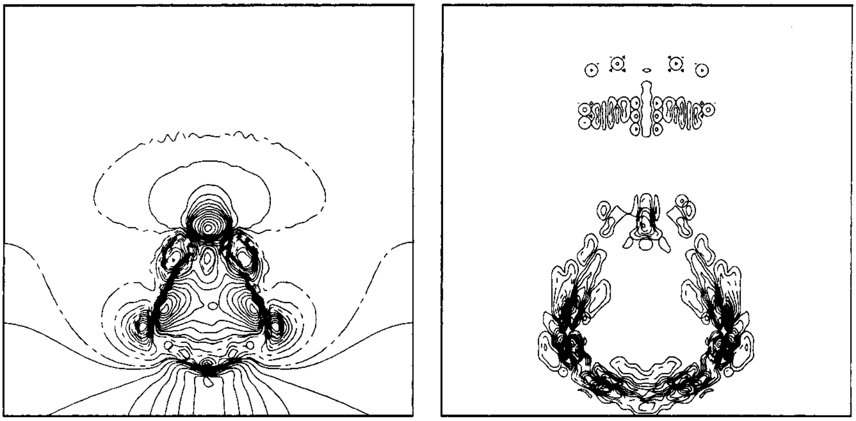
The vertex-projection filter with the hourglass velocity filter produce improved results, with a large smoothing of the flow field. The 'halo' is reduced, although it can be clearly seen in Figure 15(b) and (d). In general, the results are quite good with this combination of filters, but can be improved upon.

Figure 16 shows the combined effect of the vertex-projection and the cell-centered velocity filter. The 'halo' is largely removed from the solution, and each solution field is much smoother than earlier algorithmic combinations. The discrete divergence is concentrated around the current drop location, but some large error resides near the original location. In this case, its adverse impact on the solution appears minimal.

Figure 17 shows a comparison of all the methods given for the discontinuous drop problem. The maximum of the discrete divergence is used as a yardstick to measure the quality of each solution. The visual evidence is largely confirmed by these figures. What these figures make abundantly clear is the quantitative magnitude of the improvement achieved through the use of filtering.

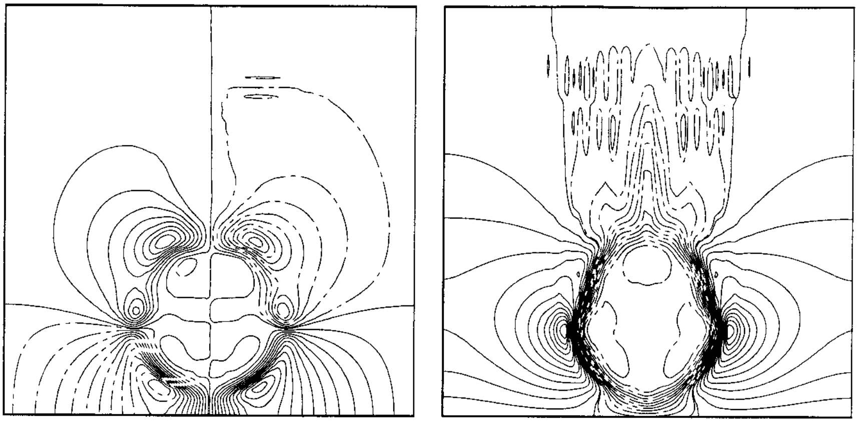


(a) x-velocity (b) y-velocity

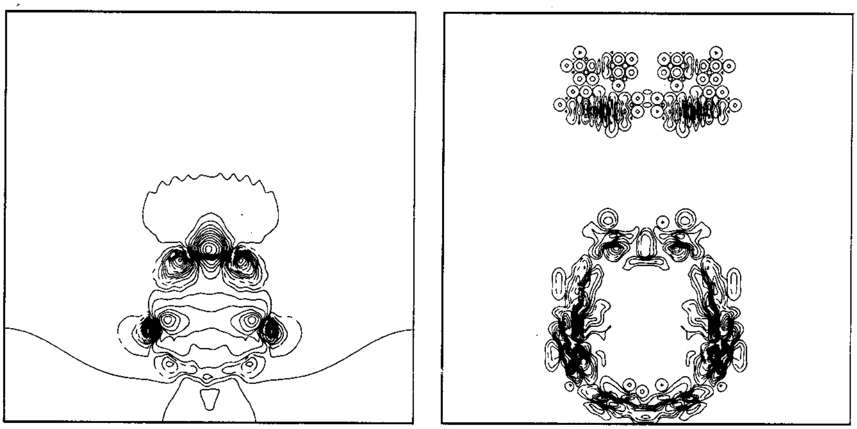


(c) pressure (d) $D\mathbf{u}^{n+1}$

Figure 14. Inviscid discontinuous drop results with a vertex- and edge-projection filters.



(a) x-velocity (b) y-velocity



(c) pressure (d) $D\mathbf{u}^{n+1}$

Figure 15. Inviscid discontinuous drop results with vertex-projection and hourglass velocity filters.

Another test problem will be the vortex-in-a-box problem used in References [2,3,6,10,11]. It uses a streamfunction

$$\Psi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y),$$

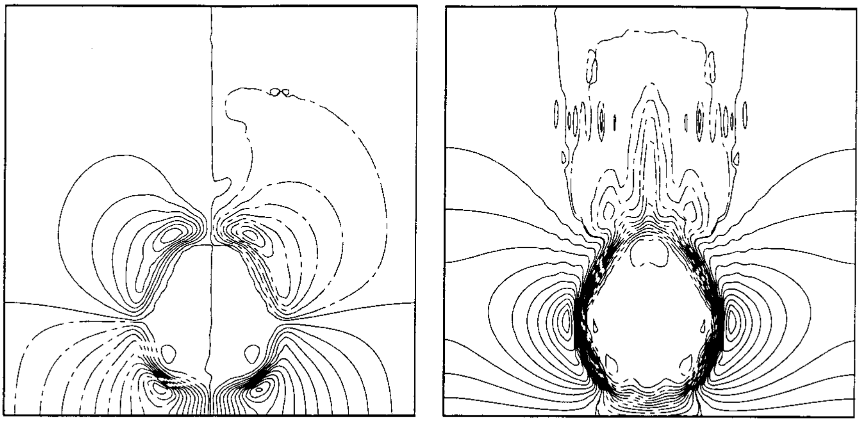
in a unit square with homogeneous velocity boundary conditions to define the initial conditions. Initial velocities are $u^0 = -\Psi_y$ and $v^0 = \Psi_x$. Similar to Reference [2], $\Delta x = \Delta y = 1/2^n$ for $n = 5$–8. The time step here is set to $\Delta t = \Delta x$ for an effective CFL number of approximately one. Convergence will be tested using Richardson extrapolation.

Also applied is the ubiquitous vortex-in-a-box problem to the analysis of the filter's performance. First, the convergence rates and error produced by the projections are tested in the same manner as before. This is followed by a more stringent use of this problem. Still using the Euler equations, the vortex-in-a-box will be run out to 60 time units on a 128^2 grid with a CFL number of 0.95. This will require nearly 8000 times steps and should provide a test of the long-term behavior of both the projections and the filter's impact.

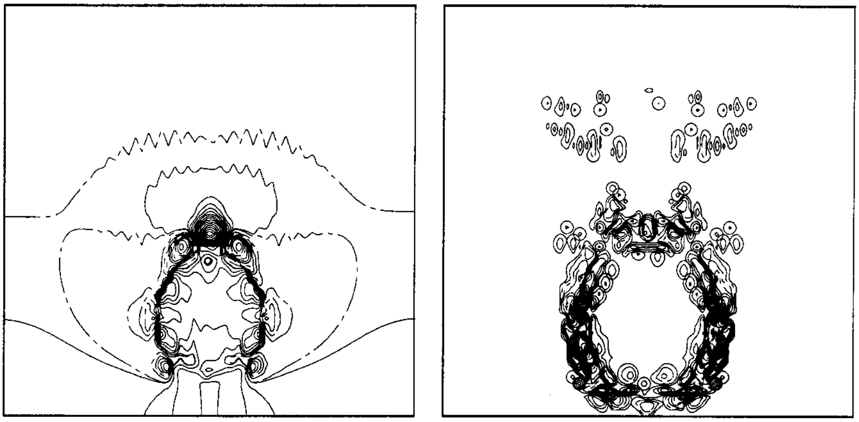
For the velocity field, the convergence rates and error resulting from using or not using the filters is shown in Table I. Without exception, the filters produce solutions with lower measured error than the unfiltered solution. The convergence rates vary slightly from method to method, but all are second-order-accurate. This table infers that everything seems to be working, but to get more critical results, one must turn to a longer time integration of the vortex-in-a-box problem.

First, results with no filters will be looked at. Figure 18 clearly shows that solution is not acceptable, with the error destroying the symmetry of the solution. The break in the symmetry is caused by the red–black Gauss–Seidel relaxation used in the multigrid solver. Figure 18(b) shows that the problem is with the now familiar line and diagonal modes of discrete divergence. There are other signs of problems such as erratic kinetic energy decay (growth), and jumps in maximum vorticity as well as a large growth in the discrete divergence.

Now that what is not wanted in a solution is known, an attempt to repair the algorithm for long-term integrations will be made. Applying the vertex-projection filter, one gets a distinct

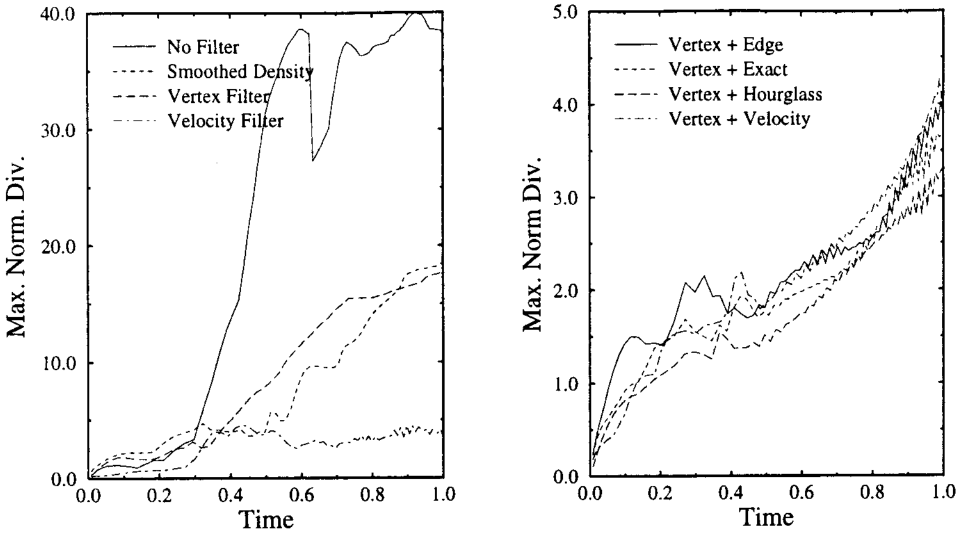


(a) x-velocity **(b) y-velocity**



(c) pressure **(d) Du^{n+1}**

Figure 16. Inviscid discontinuous drop results with vertex-projection and cell-centered velocity filters.



(a) Simple Options **(b) Dual Filters**

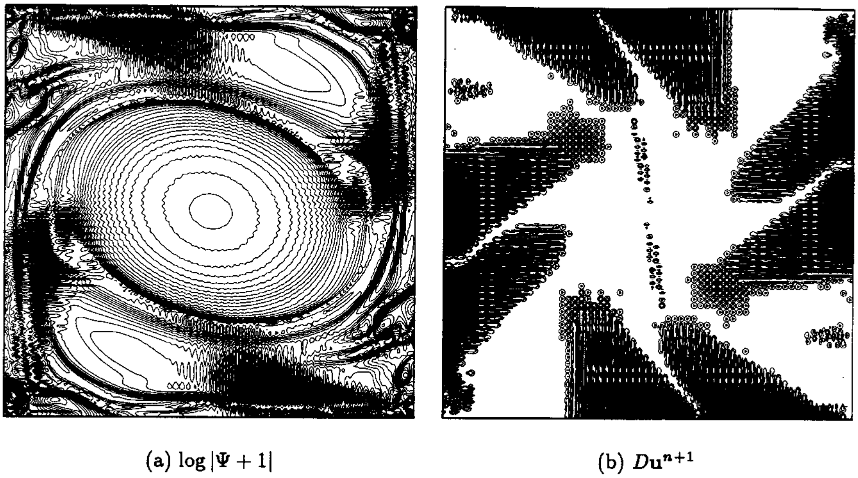
Figure 17. A comparison of method-performance for the discontinuous drop problem using the discrete divergence.

improvement in the algorithm's performance. This is made clear by Figure 19. The kinetic energy and vorticity behave in physical fashions with the discrete divergence under seeming control. Figure 19 also makes it clear that there is still more work to do. Small errors near the wall dominate the divergence error and visibly distort the vorticity profile. The discrete divergence is also rising constantly, thus signaling the eventual demise of the solution.

Adding the edge-projection filter to the vertex-projection filter gives much more satisfying results. Figure 20 shows that this combination of filters produce visually clean results and that the global characteristics of the flow are physical. Nearly identical results can be had with the combination of the vertex-projection and hourglass velocity filters, see Figure 21 (showing that the edge-projection and hourglass velocity filters are similar if not nearly identical in effect), as shown in Figures 14 and 15.

Table I. Convergence rates of the velocity field for the vortex-in-a-box problem for different combinations of filters

Case	L_2 norms				
	32^2–64^2	Rate	64^2–128^2	Rate	128^2–256^2
No filter	4.0×10^{-3}	2.18	8.8×10^{-4}	2.22	1.9×10^{-4}
Vertex-projection filter	2.8×10^{-3}	1.97	7.1×10^{-4}	2.14	1.6×10^{-4}
Edge-projection filter	3.0×10^{-3}	1.95	7.9×10^{-4}	2.30	1.6×10^{-4}
Vertex- and edge-projection filters	3.3×10^{-3}	2.01	8.3×10^{-4}	2.28	1.7×10^{-4}
Cell-centered velocity filter	3.7×10^{-3}	2.08	8.7×10^{-4}	2.22	1.9×10^{-4}
Vertex-projection and hourglass velocity filters	3.1×10^{-3}	1.96	7.9×10^{-4}	2.30	1.6×10^{-4}
Vertex- and exact-projection filters	3.8×10^{-3}	2.23	8.0×10^{-4}	2.19	1.8×10^{-4}
Cell-centered velocity filters	2.8×10^{-3}	1.98	7.1×10^{-4}	2.15	1.6×10^{-4}



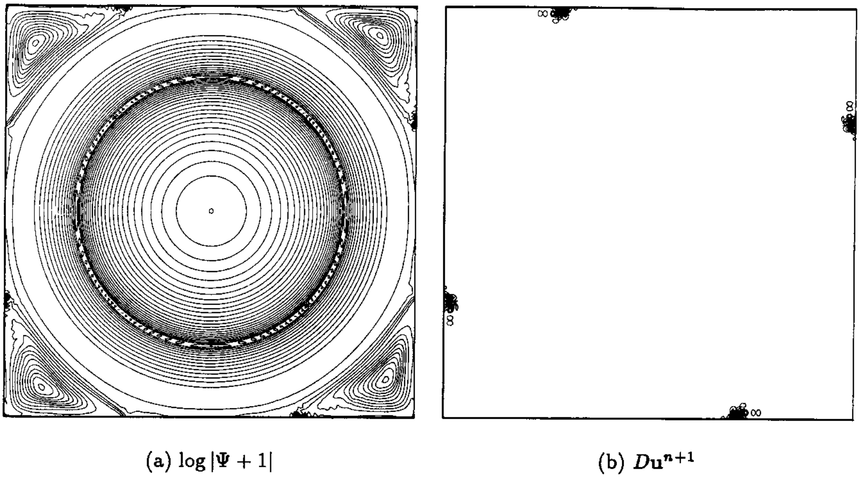
(a) $\log|\Psi + 1|$ (b) $D\mathbf{u}^{n+1}$

Figure 18. Long-time integration of vortex-in-a-box problem with no filters for a 128^2 grid.

Alone the cell-centered velocity filter should suffice to clean up the solution. This proposition is clearly shown to be correct with Figure 22. The solutions are visually pleasing and the quantities are physical. The solutions are not as high in quality as the combination of vertex- and edge-protection or hourglass velocity filters. As the reader may have already guessed, the combination of vertex-projection and cell-centered velocity filters gives outstanding results.

The reason why these two filters were chosen for tandem use can be gleaned from Figure 23. The vertex- and edge-projection filter tandem produces the lowest divergence errors, but at the cost of increased loss of kinetic energy. The vertex-projection plus cell-centered velocity filter duo produces nearly as low divergence error, but at no cost in dissipation of kinetic energy.

The results obtained through using the approximate projection method with the vertex-projection and cell-centered velocity filter are shown in Figure 24. Divergence errors are under seeming control, and the solution is clean. Kinetic energy is decaying in a physical manner and the vorticity norms are behaving physically.



(a) $\log|\Psi + 1|$ (b) $D\mathbf{u}^{n+1}$

Figure 19. Long time integration of vortex-in-a-box problem with the vertex-projection filter for a 128^2 grid.

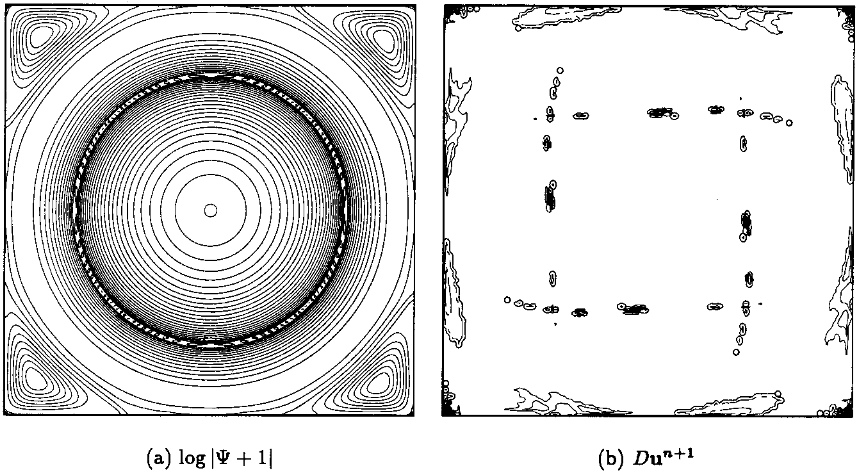

(a) $\log|\Psi + 1|$ (b) $D\mathbf{u}^{n+1}$

Figure 20. Long time integration of vortex-in-a-box problem with the edge- and vertex-projection filters for a 128^2 grid.

7. CONCLUSIONS

The presence of filters is shown to be useful in the case of variable density flows or long-time integrations where viscous forces are small. The combination of vertex-based projection filters with a velocity filter produced the best solutions in most applications.

If the algorithm was based on a vertex-staggered grid, one would likely find that a combination of edge-centered and/or the hourglass filter would be effective in removing unwanted oscillations. It should be noted that results from vertex-staggered approximate projection methods are less susceptible to oscillations because of the lower dimensions of the null space of the discrete divergence operator.

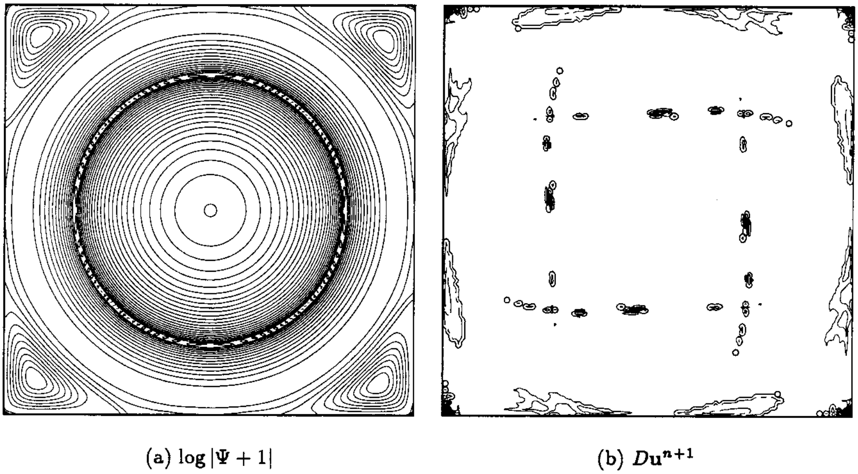
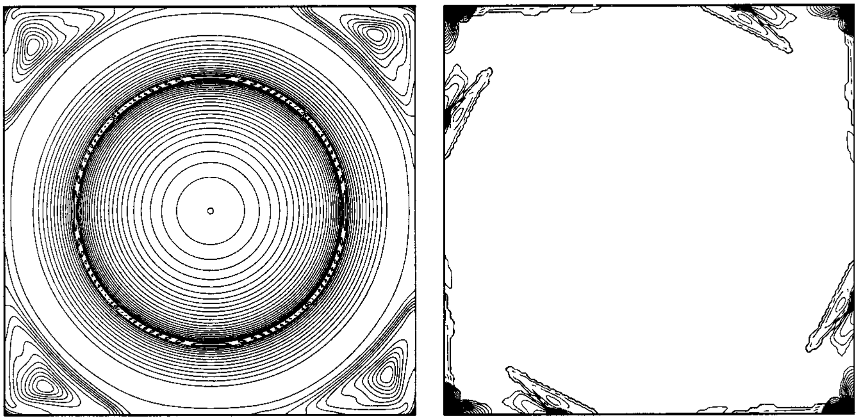

(a) $\log|\Psi + 1|$ (b) $D\mathbf{u}^{n+1}$

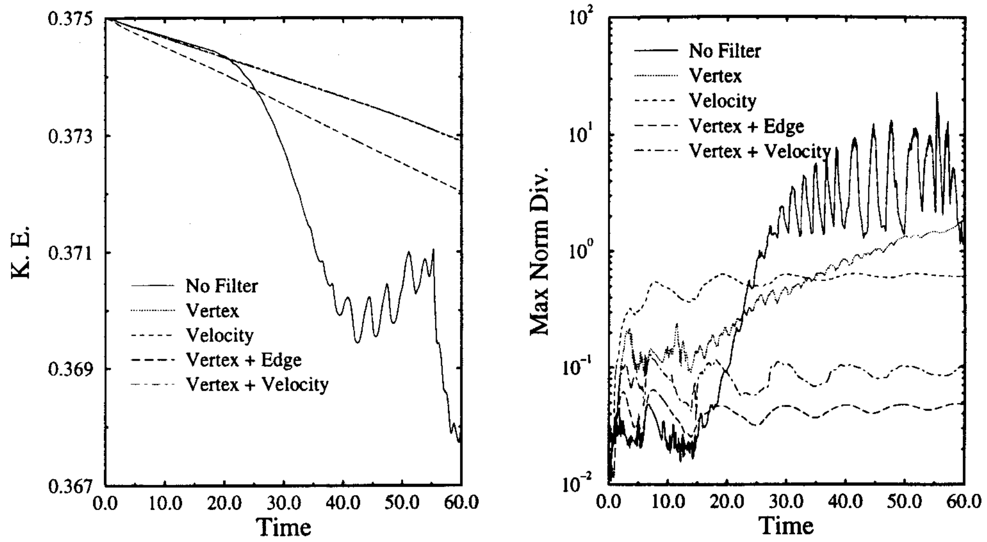
Figure 21. Long time integration of vortex-in-a-box problem with vertex-projection and hourglass velocity filters for a 128^2 grid.



(a) $\log|\Psi + 1|$ (b) $D\mathbf{u}^{n+1}$

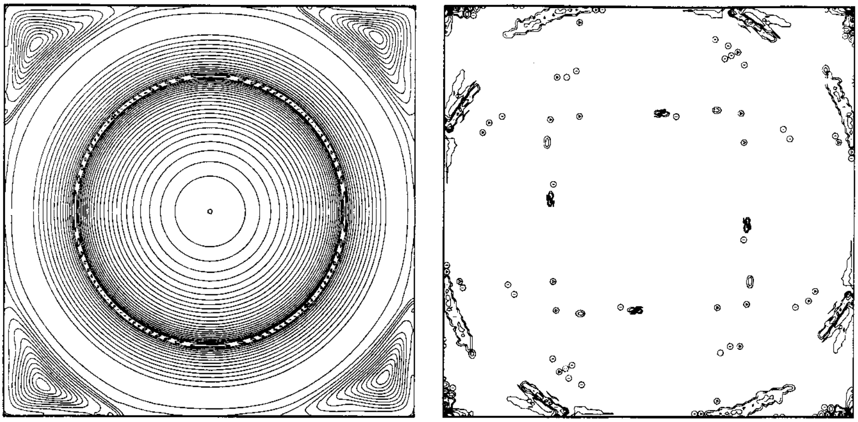
Figure 22. Long time integration of vortex-in-a-box problem with a cell-centered velocity filter for a 128^2 grid.

Although these methods were presented in the framework of an approximate projection method, they may be applicable to PPE-based methods. The important work involves identifying the null space of the divergence operator and then applying the proper filter to reduce the decoupling. This should provide PPE-based approaches (where $D\mathbf{u}^{n+1} \neq 0$) a means to improving their solution.



(a) $||\mathbf{u}||_2^2$ (b) Maximum Divergence Norms

Figure 23. Long time integration of vortex-in-a-box problem comparison of results for a 128^2 grid.



(a) $\log|\Psi + 1|$ (b) $D\mathbf{u}^{n+1}$

Figure 24. Long-time integration of vortex-in-a-box problem with the approximate projection and vertex-projection, and cell-centered velocity filters for a 128^2 grid.

ACKNOWLEDGMENTS

This work was performed under the auspices of the US Department of Energy by Los Alamos National Laboratory under Contract W-7405-ENG-36.

REFERENCES

1. A.J. Chorin, 'Numerical solution of the Navier–Stokes equations', *Math. Comput.*, **22**, 745–762 (1968).
2. J.B. Bell, P. Colella and H.M. Glaz, 'A second-order projection method of the incompressible Navier–Stokes equations', *J. Comput. Phys.*, **85**, 257–283 (1989).
3. J.B. Bell and D.L. Marcus, 'A second-order projection method variable-density flows', *J. Comput. Phys.*, **101**, 334–348 (1992).
4. P.M. Gresho, 'On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix, part 1: Theory', *Int. j. numer. methods fluids*, **11**, 587–620 (1990).
5. P. Colella, 'Multidimensional upwind methods for hyperbolic conservation laws', *J. Comput. Phys.*, **87**, 171–200 (1990).
6. J.B. Bell, P. Colella and L. Howell, 'An efficient second-order projection method for viscous incompressible flow', in D. Kwak (ed), *Proc. AIAA 10th Computational Fluid Dynamics Conference*, *AIAA Paper 911560*, 1991, pp. 360–367.
7. D.L. Brown and M.L. Minion, 'Perfromance of underresolved two-dimensional incompressible flow simulations', *J. Comput. Phys.*, **122**,165–183 (1995).
8. E.Y. Tau, 'A second-order projection method for the incompressible Navier–Stokes equations in arbitrary domains', *J. Comput. Phys.*, **115**,147–152 (1994).
9. J.C. Strikwerda, 'Finite difference methods for the Stokes and Navier–Stokes equations', *SIAM J. Sci. Stat. Comput.*, **5**, 56–68 (1984).
10. M.F. Lai, 'A projection method for reacting flow in the zero Mach number limit', *Ph.D. Thesis*, University of California, Berkeley, CA, 1993.
11. M. Lai, J.B. Bell and P. Colella, 'A projection method for combustion in the zero Mach number limit', in J.L. Thomas (ed), *Proc. AIAA 11th Computational Fluid Dynamics Conference*, *AIAA Paper 93–3369*, 1993, pp. 776–783.
12. L.H. Howell, 'A multilevel adaptive projection method for unsteady incompressible flow', in N.D. Melson, T.A. Manteuffel and S.F. McCormick (eds), *Proc. 6th Copper Mountain Conference on Multigrid Methods*, 1993, pp. 243–257.
13. A.S. Almgren, J.B. Bell and W.G. Szymczak, 'A numerical method for the incompressible Navier–Stokes equations based on an approximate projection', *SIAM J. Sci. Comput.*, **17**, 358–369 (1996).
14. A.J. Chorin and G. Marsden, *A Mathematical Introduction to Fluid Mechanics*, Springer, Berlin, 1993.

15. A.J. Chorin, 'On the convergence of discrete approximations to the Navier–Stokes equations', *Math. Comput.*, **23**, 341–353 (1969).
16. A.S. Almgren, J.B. Bell, P. Colella and L.H. Howell, 'An adaptive projection method for the incompressible Euler equations', in J.L. Thomas (ed), *Proc. AIAA 11th Computational Fluid Dynamics Conference*, *AIAA Paper 93–3345*, 1993, pp. 530–539.
17. W.J. Rider, 'The robust formulation of approximate projection methods for incompressible flows', *Tech. Rep. LA-UR-94-3015*, Los Alamos National Laboratory, 1994 (available on WWW at http://www-xdiv.lanl.gov/XHM/personnel-/wjr/Web_papers/pubs.html).
18. S.V. Patankar, *Numerical Neat Transfer and Fluid Flow*, Hemisphere, Washington, 1980.
19. W.L. Briggs, *A Multigrid Tutorial*, SIAM, Philadelphia, 1987.
20. L.G. Margolin and J.J. Pyun, 'A method for treating hourglass patterns', in C. Taylor, W.G. Habashi and M.M. Hafez (eds), *Numerical Methods in Laminar and Turbulent Flow*, Pineridge, Swansea, UK, 1987, pp. 149–160.
21. D.J. Benson, 'Computational methods in Lagrangian and Eulerian hydrocodes', *Comput. Methods Appl. Mech. Eng.*, **99**, 235–394 (1992).
22. W.J. Rider, 'Approximate projection methods for incompressible flow: implementation, variants and robustness', *Tech. Rep. LA-UR-94-2000*, Los Alamos National Laboratory, 1994 (available on WWW at http://www-xdiv.lanl.gov/XHM-/personnel/wjr/Web_papers/pubs.html).