Journal of Computational Physics **148,** 366–396 (1999)
Article ID jcph.1998.6115, available online at http://www.idealibrary.com on IDEAL®

A Numerical Method for Solving Incompressible Flow Problems with a Surface of Discontinuity

B. T. Helenbrook,* L. Martinelli,† and C. K. Law‡

Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, New Jersey 08544
E-mail: *helenbrk@stanford.edu, †gigi@phantom2.princeton.edu, and ‡cklaw@princeton.edu

Received June 20, 1997; revised October 1, 1998

A numerical method for solving problems in which a moving surface of discontinuity separates regions of incompressible flow is presented. The method developed is notable in that it does not introduce any artificial smoothing of the change in fluid properties across the surface of discontinuity. This results in an increase in accuracy relative to methods which introduce smoothing effects. The method was also shown to be fairly versatile; problems describing a free surface, an immiscible fluid interface, and a premixed flame discontinuity were solved. There is a limitation, however, in that the method appears to be most suitable for application to inviscid problems. The reason for this limitation and possible approaches toward resolving it are discussed.

INTRODUCTION

Many physical problems can be mathematically treated as two incompressible fluids separated by a moving surface of discontinuity. Typical examples include the flow of immiscible fluids, the flow of liquid–gas mixtures, premixed flame propagation, and vortex sheet dynamics, among others. Interest stemming from practical applications, such as ocean waves and sailing vessels, engine sprays and burning, and even plastic extrusion, has motivated a significant effort to numerically simulate these problems. Due to complexities resulting from the moving surface of discontinuity, however, numerical simulation techniques have been much slower to develop than those for single fluid incompressible flow.

Recently, however, a class of methods, which we shall refer to as "continuum" methods, has been demonstrated to be both robust and versatile in solving these problems [6–8, 12, 23, 25]. Simulations have been presented which include a broad range of physical effects such as both density and viscosity variation across the discontinuity, surface tension, highly convoluted surface geometries, and in some cases even evaporation. These simulations have provided some excellent qualitative insight into two-fluid problems. The quantitative

0021-9991/99 $30.00

accuracy of these methods, however, may be somewhat limited for reasons which are made clear in the following paragraph.

The commonality in the continuum methods discussed above is their treatment of the discontinuous change in properties between the two fluids. The treatment used is to smooth the discontinuous change over some artificial finite width, usually around 4–6 computational grid cells. This allows the simplification that conventional finite difference techniques designed for smoothly varying fluid properties can be used, which in turn leads to robust and versatile schemes. The difficulty is that the smoothing of the discontinuity causes error in the simulations with a magnitude proportional to the smoothing width relative to the scale of the flow being resolved; accurate results are obtained when the smoothing width is thin relative to the flow scale. For a typical resolution of a wavelength of the flow, say 24 grid cells/wavelength, the transition width (4–6 computational cells) is only moderately smaller than the flow scale, approximately 20%, and thus the simulations can be fairly inaccurate. Furthermore, since the transition width is proportional to the grid size, this error decreases only linearly as the grid size is decreased. Thus, these methods are only first-order accurate in space.

The goal of this work was to be able to incorporate a range of physical effects comparable to that of a "continuum" method but with increased accuracy relative to these methods. To achieve this, we have developed a method which maintains the discontinuous properties of the solution; no artificial smoothing at the discontinuity is introduced. Consequently, second-order spatial accuracy can be obtained, as we demonstrate in the paper. We hasten to clarify, however, that although inviscid problems can be solved readily, application to viscous problems requires further study. Thus, although the method is more accurate than the continuum method for inviscid problems, it does not achieve the same degree of versatility. In this paper, we describe this method and further detail its characteristics.

In the first section following the Introduction, the mathematical and physical problem is defined in the most general form which is currently solvable. This definition gives an idea of the effects which can currently be simulated with this method.

The next three sections describe the numerical method. We begin by summarizing the numerical algorithm used to solve the incompressible flow equations. This algorithm was developed by Belov *et al.* [4] and has been shown to be efficient and accurate for the computation of both viscous and inviscid, unsteady, incompressible flows.

The fifth section describes the modifications made to the algorithm to incorporate the surface of discontinuity. The treatment of the discontinuity is unique in that, as mentioned above, no artificial smoothing of the change in properties across the discontinuity is introduced.

The last section in the statement of the numerical method describes the algorithm used for tracking the position of the discontinuity in time. The algorithm used is a level-set approach similar to that originally proposed by Osher and Sethian [18]. It was chosen over a volume-of-fluid [14, 20] approach or a "front-tracking" [25, 26] approach, both of which can also be used to give accurate results, because the update of the discontinuity position could be performed in a manner similar to that used for the solution of the incompressible flow equations. As we show, this made the implementation of the level-set approach relatively easy. We did, however, have a couple of difficulties in achieving accurate results using the level-set approach. These difficulties and how they were overcome are also described.

With the description of the algorithm complete, the accuracy and efficiency of the resulting scheme are verified with test cases originally used by Baker *et al.* [2]. These are a free

surface wave and an interfacial wave which propagate with a stationary shape. The results show that for a given mesh size the magnitude of the error is significantly smaller than that expected from a continuum method for inviscid flows and that second-order spatial accuracy is achievable. An application to premixed flames is also presented with several simulations to further demonstrate the potential of this method.

PROBLEM DEFINITION

The class of problems that the numerical algorithm is intended for is the flow of two incompressible fluids of constant but different densities and viscosities which are separated by a moving discontinuity surface. A straightforward example of a physical problem which can be represented in this manner is the flow of two immiscible fluids such as oil and water. To describe these flows mathematically, equations governing the incompressible flow of both fluids must be given. Additionally, conditions which describe the relationship between the flow conditions on either side of the discontinuity and the motion of the discontinuity must be provided. In the following, we present the governing equations and surface conditions in the most general form which can be solved using our numerical algorithm.

The incompressible Navier–Stokes equations describe the motion of the flow. The formulation of these equations allows for two fluids of constant but different density and viscosity and also includes the effects of gravity. In vector form, these equations are

$$\text{diag}[0,\,1,\,1]\cdot\frac{\partial \mathbf{w}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}=\frac{\partial \mathbf{r}}{\partial x}+\frac{\partial \mathbf{s}}{\partial y}+\begin{bmatrix}0\\0\\-\rho_k/\text{Fr}\end{bmatrix}, \tag{1}$$

where x and y are the horizontal and vertical coordinates, respectively; t is the time variable; and $\mathbf{w}$ is vector notation for the flow variables

$$\mathbf{w}=\begin{Bmatrix}p\\ \rho_k u\\ \rho_k v\end{Bmatrix}, \tag{2}$$

where p, u, and v are the pressure, horizontal velocity, and vertical velocity, and ρ_k is the constant density of fluid 1 or 2. In Eq. (1), $\mathbf{f}$ and $\mathbf{g}$ are the Euler flux vectors

$$\mathbf{f}=\begin{Bmatrix}\rho_k u\\ \rho_k u^2+p\\ \rho_k uv\end{Bmatrix},\qquad \mathbf{g}=\begin{Bmatrix}\rho_k v\\ \rho_k uv\\ \rho_k v^2+p\end{Bmatrix}, \tag{3}$$

and $\mathbf{r}$ and $\mathbf{s}$ are the viscous flux vectors

$$\mathbf{r}=\begin{Bmatrix}0\\ \tau_{xx}\\ \tau_{xy}\end{Bmatrix},\qquad \mathbf{s}=\begin{Bmatrix}0\\ \tau_{xy}\\ \tau_{yy}\end{Bmatrix} \tag{4}$$

with

$$\tau_{xx}=2\frac{\mu_k}{\text{Re}}\frac{\partial u}{\partial x}\qquad \tau_{yy}=2\frac{\mu_k}{\text{Re}}\frac{\partial v}{\partial y}\qquad \tau_{xy}=\frac{\mu_k}{\text{Re}}\left[\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right], \tag{5}$$

where μ_k is the viscosity of fluid 1 or 2. In Eq. (1), diag[0, 1, 1] is the identity matrix with a zero first diagonal entry. This matrix removes the time derivative of the pressure term from the continuity equation in this vector notation. The above equations are made nondimensional by U and L, the characteristic velocity and length of the system, and by ρ_1 and μ_1, the density and viscosity of the more dense of the two fluids. $\mathrm{Re} = UL/\nu_1$ is the Reynolds number, where ν is the kinematic viscosity. $\mathrm{Fr} = U^2/(gL)$ is the square of the Froude number representing the force of gravity in the negative y direction, where g is the acceleration of gravity.

To complete the description of the problem, jump conditions across the discontinuity surface and the velocity at which the surface moves must be specified. Currently, the algorithm can solve problems in which the jump conditions are a function of the flow properties on both sides of the surface and the normal and curvature of the surface. Problems in which the jump conditions are also a function of the spatial derivatives of the flow variables at the surface as yet cannot be solved. This is important to recognize because in many viscous problems the jump conditions *are* a function of the spatial flow derivatives at the discontinuity. For example, the jump conditions for the viscous flow of two immiscible fluids with surface tension depend on the viscous stress tensor evaluated at the discontinuity (see [7]). The viscous stress tensor is a function of the spatial flow derivatives, and thus this problem cannot be solved. The inviscid jump conditions, however, are functions of only the flow variables and the curvature of the discontinuity through the surface tension, and thus immiscible fluid flow problems with surface tension and inviscid jump conditions can be solved. The reason for this limitation and possible methods around it are discussed later.

Similar to the jump conditions, the normal velocity of the surface can be specified as a function of the curvature of the surface as well as the flow properties at the surface. A detailed description of the jump conditions and surface velocity as well as a description of the physical domain and boundary conditions are given in the discussion of each physical problem simulated.

NUMERICAL ALGORITHM

The numerical algorithm used to solve the incompressible flow equations was originally presented in [4]. This algorithm is a cell-centered, finite-volume method which was designed for constant density and viscosity incompressible flows. In the following, a summary of the discretization of the spatial and temporal derivatives in Eq. (1) and the method used to solve the resulting discretized equations is presented. Readers interested in a more detailed description of the algorithm are referred to [5].

The discretization of the spatial derivatives in the algorithm are for a curvilinear, structured mesh. For simplicity, here we only discuss the effective discretization that this results in on an equally spaced Cartesian mesh. In this case, the discretization of the Euler terms reduces to a standard central difference formula. To avoid the odd–even decoupling which can occur with central differences, third-order artificial dissipation is added to the discretization. The dissipation added to avoid decoupling in the y direction is of the form $\lambda/32\Delta y^3[\partial^4\mathbf{w}/\partial y^4]$, where λ is an $O(1)$ scaling constant which is discussed in [5], Δy is the grid spacing in the y-direction, and the fourth derivative is evaluated using a standard five-point central difference. A corresponding term is added to avoid odd–even decoupling in the x-direction. The discretization of the viscous terms on a Cartesian mesh is equivalent to a central

difference formula for a second derivative. The above discretizations give second-order accuracy in space. It is important to note for discussion later that discretization of both the Euler and viscous terms requires only a three-point stencil in each coordinate direction to evaluate the derivatives.

To discretize the solution in time, a second-order accurate, implicit scheme is used. This results in discretized governing equations of the form

$$\text{diag}[0, 1, 1] \cdot \frac{\left[3\mathbf{w}_{i,j}^{n+1} - 4\mathbf{w}_{i,j}^{n} + \mathbf{w}_{i,j}^{n-1}\right]}{2\Delta t} + \mathbf{R}\left(\mathbf{w}_{i,j}^{n+1}\right) = 0, \tag{6}$$

where the first term in the equation is the time derivative and $\mathbf{R}$ is a shorthand notation denoting the evaluation as discussed above of the spatial derivatives of Eq. (1) at the cell i, j. Δt is the implicit time step, and the superscripts on $\mathbf{w}$ represent the time level of the solution. This scheme is implicit because the evaluation of the spatial derivatives in $\mathbf{R}$ is at time level $n+1$. Given the solution at two previous time levels, $\mathbf{w}^n$ and $\mathbf{w}^{n-1}$, the above equations must be solved for $\mathbf{w}^{n+1}$ to advance the solution in time. This approach has two advantages. The first is that the scheme is A-stable [10] and therefore the magnitude of Δt is not limited by a stability restriction but only by the required accuracy. The second is that the update of the solution in time satisfies the discretized form of the continuity equation without requiring any special treatment to enforce this constraint such as in a projection method (see [3]). On the other hand, Eq. (6) for $\mathbf{w}_{i,j}^{n+1}$ is spatially coupled through the operator $\mathbf{R}$ and is also nonlinear. To make this method practical, an efficient way of solving these equations is necessary.

To find the solution to Eq. (6), we transform to a problem of evolving an unsteady system of equations to steady state in a fictitious time or "pseudo-time." This transformation is accomplished using the following equations which define the pseudo-time evolution

$$\mathbf{E}(\mathbf{w}_{i,j}) = \text{diag}[0, 1, 1] \cdot \frac{\left[3\mathbf{w}_{i,j} - 4\mathbf{w}_{i,j}^{n} + \mathbf{w}_{i,j}^{n-1}\right]}{2\Delta t} + \mathbf{R}(\mathbf{w}_{i,j}) \tag{7}$$

$$\frac{\partial \mathbf{w}_{i,j}}{\partial t^*} + \text{diag}[\Gamma^2, 1, 1] \cdot \mathbf{E}(\mathbf{w}_{i,j}) = 0 \tag{8}$$

$$\mathbf{w}_{i,j}|_{t^*=0} = \mathbf{w}_{i,j}^{n}. \tag{9}$$

$\mathbf{E}(\mathbf{w}_{i,j})$ is the error in the solution to Eq. (6) when $\mathbf{w}_{i,j}$ is substituted for $\mathbf{w}_{i,j}^{n+1}$, Γ^2 is a constant which will be discussed shortly, and t^* is the pseudo-time variable. If this set of equations for the evolution of $\mathbf{w}_{i,j}$ can be evolved to a steady solution, $\mathbf{E}(\mathbf{w}_{i,j})$ will be zero by Eq. (8) and thus we will have found the solution $\mathbf{w}_{i,j}^{n+1}$.

In the simplified case in which in Eq. (7) the implicit time derivative terms in $\mathbf{E}$ are neglected, the pseudo-time evolution equations reduce to those introduced by Chorin [9] for finding steady solutions in viscous incompressible flow. Analysis of these equations for the inviscid case by Rizzi and Eriksson [21] has shown that the equations are hyperbolic in nature, and therefore an explicit time stepping scheme appropriate for hyperbolic systems can be used to evolve to steady state. In [21] for example, a three-stage Runge–Kutta scheme is used with a time step limit based on the wave speeds of the hyperbolic system. The analysis of Rizzi and Eriksson also suggested that $\Gamma^2 = \max(0.1, u^2 + v^2)$ is an optimum choice for achieving the maximum convergence rate of the explicit scheme to steady state in the pseudo-time evolution. For the viscous case, as originally demonstrated by Chorin,

an explicit scheme can also be used with the only modification being the adjustment of the allowable time step of the explicit pseudo-time scheme to account for the viscous terms. A discussion of the time-stepping limit in the viscous case can be found in [5, 9].

The implicit time derivative terms in **E** do not change the approach originally proposed by Chorin; an explicit time stepping scheme is used to find the steady-state solution in t^*, which in this case is the solution to Eq. (6). The second two terms in the implicit time derivative, $-4\mathbf{w}_{i,j}^n/\Delta t$ and $\mathbf{w}_{i,j}^{n-1}/\Delta t$, are constant during the pseudo-time evolution and thus have no effect on the stability of the explicit pseudo-time stepping. The first term of the implicit time derivative, $3\mathbf{w}_{i,j}/2\Delta t$, is unsteady in t^* and can affect the stability of the explicit scheme. To incorporate this term into the pseudo-time evolution with the minimum impact on the stability limit of the explicit scheme, a point-implicit formulation is used. The following equations demonstrate the point-implicit approach in the simplified case of a one-step explicit update of the solution. The extension to a multistage or Runge–Kutta type scheme is trivial. We begin by rewriting Eq. (8) as

$$\frac{\partial \mathbf{w}_{i,j}}{\partial t^*} + \text{diag}[0, 1, 1] \cdot \frac{3\mathbf{w}_{i,j}}{2\Delta t} + \mathbf{E}^*(\mathbf{w}_{i,j}) = 0, \tag{10}$$

where $\mathbf{E}^*(\mathbf{w}_{i,j})$ is the sum of the spatial derivatives and the constant terms in the implicit time derivative. In the simplified case of a one-stage explicit scheme, this is discretized in pseudo-time as

$$\frac{\mathbf{w}_{i,j}^{t^*+dt^*} - \mathbf{w}_{i,j}^{t^*}}{dt^*} + \text{diag}[0, 1, 1] \cdot \frac{3\mathbf{w}_{i,j}^{t^*+dt^*}}{2\Delta t} + \mathbf{E}^*\left(\mathbf{w}_{i,j}^{t^*}\right) = 0, \tag{11}$$

where dt^* is the explicit time step in pseudo-time. The update for **w** can then be solved iteratively from the above algebraic equation. Because the unsteady term in the implicit time derivative is evaluated at $t^* + dt^*$, the point-implicit formulation minimally affects the stability of the explicit scheme. This is shown by the analysis given in [5].

Finally, we note that the evolution to steady state in pseudo-time must be performed for each implicit time step update of the solution, and thus the time required for the entire calculation is strongly affected by the time required to reach convergence to steady state in pseudo-time. Since it is not necessary to be time accurate in pseudo-time, several optimization techniques designed for accelerating the convergence to steady state are used. These include an explicit time-stepping scheme designed for large pseudo-time steps described in [5], local time stepping in which each cell is advanced at its maximum stable pseudo-time step rather than a uniform time step, and multigrid which is described for a single fluid flow in [5].

DISCONTINUITY SURFACE

The algorithm described above is for incompressible flows with constant density and viscosity. To solve problems with two fluids, we must account for the surface of discontinuity between the two fluids. As discussed in the Introduction, we have avoided using a continuum formulation of the discontinuity because of the first-order spatial error which can arise with this approach. Our approach treats the surface of discontinuity as a moving boundary which subdivides the numerical domain into two separate computational regions, one for each fluid. The computational region that any given numerical cell belongs to is determined by which side of the boundary the cell center lies on. Figure 1 shows a simple schematic of

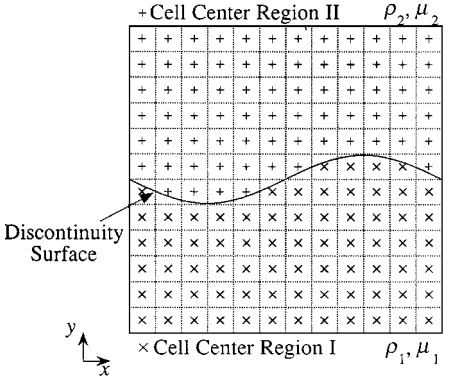

FIG. 1. Subdivision of the computational domain into two distinct regions.

such a division. In each computational region, the idea is to solve the incompressible flow equations with constant density and viscosity in a manner which is consistent with the jump conditions applied at the surface of discontinuity.

The implementation of this approach is described in the following. In this section, we describe the modifications which were made to the incompressible flow algorithm to allow for the moving internal boundary and jump conditions. This description is presented under the assumption that the position of the discontinuity surface in time, as well as the normal and curvature along the surface, is known. In the following section, the method for determining the position, normal, and curvature of the discontinuity surface are discussed.

To illustrate the modifications made to the incompressible flow algorithm, we begin by giving a step-by-step explanation of the spatial finite differencing used to form the y-derivatives for a column of cell centers which crosses the discontinuity. The explanation uses the schematic shown in Fig. 2, which is a typical one-dimensional pressure profile taken along a column of cell centers for some fixed i. We again assume that the grid is an equally spaced Cartesian mesh, although the procedure can easily be extended to curvilinear coordinates. In this example profile, the discontinuity is between cells $(i, 5)$ and $(i, 6)$. At the point at which the surface of discontinuity is crossed, χ, there is a discontinuity in pressure, the magnitude of which is governed by the jump conditions describing the physical problem. As discussed above, this location is treated as a boundary to two separate regions of incompressible flow. Accordingly, the finite differencing on both sides of the boundary must be modified.

Because only a three-point stencil is needed in each coordinate direction to evaluate the Euler and viscous derivatives, the finite differences for these terms only need to be

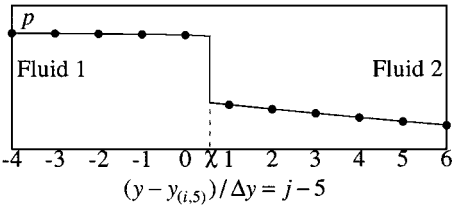

FIG. 2. One-dimensional profile of the pressure taken along a column of cell centers which crosses the discontinuity. The solid circles are the values at the solution at the cell centers.

modified at cells which are immediately adjacent to the surface of discontinuity. In Fig. 2, these are cells $(i, 5)$ and $(i, 6)$. For cells which are one or more cell centers away from the discontinuity, the standard finite difference stencil can be used, so no modifications are required to evaluate the Euler and viscous derivatives.

To evaluate the y-derivatives for the points adjacent to the discontinuity, the general approach is to define a pair of values at the discontinuity which satisfies the jump conditions and then evaluate the spatial derivatives based on these values. In choosing the values at the discontinuity, however, we must ensure that information can be transmitted through the discontinuity in both directions. For example, if at a given time step we change the pressure at the far-field boundary of fluid 1, this information must be transmitted by the pseudo-time evolution throughout fluid 1 and across the discontinuity to fluid 2 or vice versa in order to achieve a converged solution for the next time step. To ensure that this can occur, we use the following procedure.

First, we use third-order extrapolations from both sides of the discontinuity to determine values at the surface of discontinuity,

$$\mathbf{s}_1 = \frac{\chi(\chi+1)}{2}\mathbf{w}_{i,3} - \chi(\chi+2)\mathbf{w}_{i,4} + \frac{(\chi+1)(\chi+2)}{2}\mathbf{w}_{i,5} \tag{12}$$

$$\mathbf{s}_2 = \frac{(1-\chi)(2-\chi)}{2}\mathbf{w}_{i,8} - (1-\chi)(3-\chi)\mathbf{w}_{i,7} + \frac{(2-\chi)(3-\chi)}{2}\mathbf{w}_{i,6}, \tag{13}$$

where $\mathbf{s}_1$ and $\mathbf{s}_2$ are the values at the surface of discontinuity extrapolated from fluids 1 and 2, respectively. The location of these values is shown in Fig. 3. The notation $p_{\mathbf{s}_1}$ used in this figure is equivalent to $[1, 0, 0]\cdot\mathbf{s}_1$. $\mathbf{s}_1$ gives an estimate of the values at the discontinuity which is completely based on the solution in fluid 1, while $\mathbf{s}_2$ is solely based on the solution in fluid 2. Due to the error inherent in the extrapolation and the error in the estimate for $\mathbf{w}^{\mathbf{n+1}}$ present during the pseudo-time evolution, the values at the discontinuity usually do not satisfy the jump conditions. The next step in the procedure is to use this information to define a pair of values at the discontinuity which does satisfy the jump conditions.

To do this, we begin by using the jump conditions and $\mathbf{s}_2$ to estimate an additional value at the surface in fluid 1, $\mathbf{s}_{1,\text{jump}}$. The details of determining $\mathbf{s}_{1,\text{jump}}$ can vary somewhat depending on the exact form of the jump conditions. This procedure will be made more specific when we examine individual problems. $\mathbf{s}_1$ and $\mathbf{s}_{1,\text{jump}}$ are not equal because $\mathbf{s}_1$ and $\mathbf{s}_2$ do not satisfy the jump conditions. We then choose a single value from $\mathbf{s}_1$ and $\mathbf{s}_{1,\text{jump}}$ based on the direction which information is propagated by the pseudo-time evolution. Remember that the pseudo-time evolution equations are hyperbolic in nature; therefore in the direction normal to the discontinuity three wave speeds and three associated characteristic variables can be determined. A description of the linearization of the pseudo-time equations used to

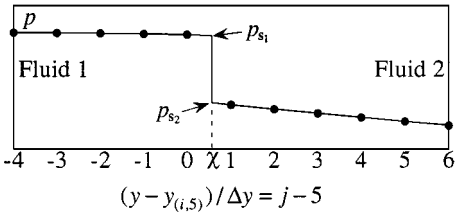

FIG. 3. One-dimensional pressure profile showing the location of the values $\mathbf{s}_1$ and $\mathbf{s}_2$.

determine these variables and wave speeds can be found in [5, 21] in their discussion of nonreflecting far-field boundary conditions. If the wave propagation direction for a characteristic variable is from fluid 1 toward fluid 2, that variable is estimated using $\mathbf{s}_1$. If the wave propagates in the opposite direction, the characteristic variable is estimated from $\mathbf{s}_{1,\text{jump}}$. This determines a single vector of flow values, $\mathbf{s}_1'$ in fluid 1 at the discontinuity. Values in fluid 2 at the discontinuity, $\mathbf{s}_2'$, are then estimated using $\mathbf{s}_1'$ and the jump conditions. This method defines a pair of values which satisfy the jump conditions, and it also transmits information through the discontinuity in a manner consistent with the wave propagation direction of the pseudo-time evolution.

The above linearization of the pseudo-time equations could just as easily have been performed in fluid 2. We found that both the convergence rate to steady state in pseudo-time and the final converged solution were insensitive to whether we performed the linearization in fluid 1 or fluid 2.

The final step is to evaluate the y-direction Euler and viscous derivatives at the points $(i, 5)$ and $(i, 6)$ based on the values $\mathbf{s}_1'$ and $\mathbf{s}_2'$. To evaluate these derivatives, we first extrapolate the solution to dummy values

$$\mathbf{d}_{i,6} = \frac{2(1-\chi)}{\chi+2}\mathbf{w}_{i,3} + \frac{3(\chi-1)}{\chi+1}\mathbf{w}_{i,4} + \frac{6}{(1+\chi)(2+\chi)}\mathbf{s}_1' \tag{14}$$

$$\mathbf{d}_{i,5} = \frac{2\chi}{3-\chi}\mathbf{w}_{i,8} - \frac{3\chi}{2-\chi}\mathbf{w}_{i,7} + \frac{6}{(2-\chi)(3-\chi)}\mathbf{s}_2', \tag{15}$$

where $\mathbf{d}$ is used to denote a dummy value. The locations of the dummy values are shown in Fig. 4. We then use a standard central difference formula with the dummy values to evaluate the derivatives at the adjacent points. This approach is well defined even when χ becomes very close to 0 or 1 and is second-order accurate for the Euler terms and first-order accurate for the viscous terms. A central difference at point $(i, 5)$ using the dummy value, $\mathbf{d}_{i,6}$, is equivalent to evaluating the derivative using the values at points $(i, 3)$ and $(i, 4)$ with $\mathbf{s}_1'$. Extrapolating to the dummy value and then using a central difference has some advantages which will become clear in the following.

We also need to discuss the evaluation of the artificial dissipation. The fourth-order derivative needed for the artificial dissipation is calculated by subtracting third-order derivatives found at a location midway between cell centers,

$$\Delta y \left.\frac{\partial^4 \mathbf{w}}{\partial y^4}\right|_{i,j} = \left.\frac{\partial^3 \mathbf{w}}{\partial y^3}\right|_{i,j+1/2} - \left.\frac{\partial^3 \mathbf{w}}{\partial y^3}\right|_{i,j-1/2}. \tag{16}$$

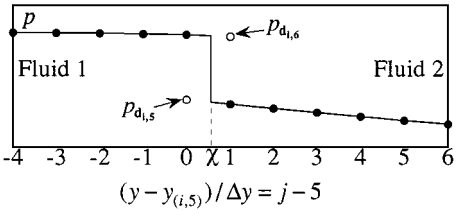

FIG. 4. One-dimensional pressure profile showing the location of the dummy values.

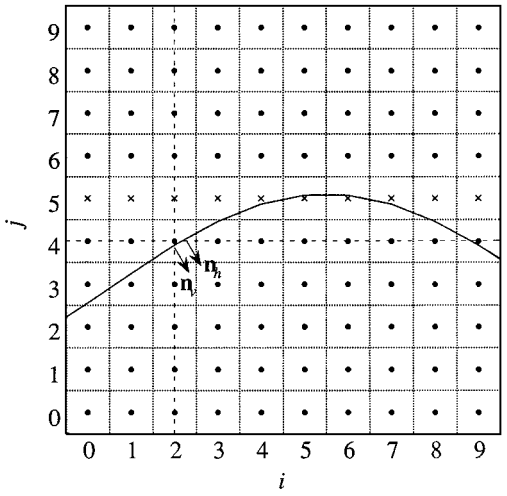

FIG. 5. One-dimensional discontinuity geometry which causes difficulty.

The third-order derivatives at the midpoints are calculated with a four-point stencil:

$$\Delta y^3 \frac{\partial^3 \mathbf{w}}{\partial y^3}\bigg|_{i,j+1/2} = \mathbf{w}_{i,j+2} - 3\mathbf{w}_{i,j+1} + 3\mathbf{w}_{i,j} - \mathbf{w}_{i,j-1}. \tag{17}$$

Near the discontinuity, calculation of the dissipation is modified by setting the third-order derivative at the midpoint of the two cell centers bordering the discontinuity to zero. One midpoint away in either direction, the dummy cell value is used in the formula for the third-order derivative. This method is stable and adds at most a second-order correction to the solution near the discontinuity.

The above procedure can be used in both coordinate directions to evaluate the x and y spatial derivatives near the discontinuity. However, for certain surface geometries, the above approach may not work. Figure 5 shows an example of a surface geometry which causes difficulty for evaluation of derivatives in the x-direction. Along the horizontal row of cell centers shown as crosses, $\times$, there are only two cell centers in fluid 1 between intersections with the discontinuity surface. For this row, the above procedure cannot be used to form the x-derivatives at the points adjacent to the discontinuity because a three-point stencil is needed on both sides of the discontinuity. To avoid this difficulty, we combine the dummy values found from both coordinate directions to define a unique dummy value for each point adjacent to the discontinuity. For example, the dummy values at points such as (4, 5), (5, 5), (6, 5), and (7, 5) which cannot be approximated in the x-direction are determined by using the above procedure in the y-direction. For dummy value locations that can be approximated from both coordinate directions such as at (2, 4) in Fig. 5, we average the values approximated from each coordinate direction with the weighting

$$\mathbf{d}_{2,4} = \frac{\mathbf{d}_{2,4,h}|\mathbf{n}_h \cdot (1,0)| + \mathbf{d}_{2,4,v}|\mathbf{n}_v \cdot (0,1)|}{|\mathbf{n}_h \cdot (1,0)| + |\mathbf{n}_v \cdot (0,1)|}, \tag{18}$$

where $\mathbf{d}_{2,4,h}$ and $\mathbf{d}_{2,4,v}$ are the values found using the above procedure in the horizontal and vertical directions, respectively. $\mathbf{n}_h$ and $\mathbf{n}_v$ are the normals to the discontinuity at the

point at which the discontinuity was crossed in the horizontal or vertical direction. This is also shown in Fig. 5. By combining information from both directions, we can determine a dummy value for each point adjacent to the discontinuity which then can be used with a central difference formula to evaluate the spatial derivatives at the adjacent points.

The above procedure for determining the dummy values was performed at the beginning of each multistage update of the solution in pseudo-time. The dummy values were then used during the multistage update to form the derivatives near the discontinuity. This completes the description of the modifications used to evaluate the spatial derivatives near the discontinuity.

In performing calculations with the above method, we found that due to the change in the finite difference stencil near the discontinuity, the estimate of the maximum stable pseudo-time step predicted in [5] for constant density and viscosity flows had to be reduced by approximately a factor of 2 at points adjacent to the discontinuity. Because a local time stepping approach was used in which each cell was advanced at its maximum stable pseudo-time step, this restriction only affected the pseudo-time step taken at the cells adjacent to the discontinuity.

By using the above procedure, the pseudo-time iteration converged to a solution whose evolution in real time was based on values at the discontinuity which satisfied the jump conditions. Furthermore, by choosing values at the discontinuity in a manner consistent with the pseudo-time iteration, changes in the fluid properties on either side of the discontinuity could be transmitted through the discontinuity. This allows the pseudo-time evolution to converge to a solution which satisfies the far-field boundary conditions and, as shown in the results, gives an accurate prediction of the unsteady evolution of the flow.

Some additional changes to the incompressible flow algorithm are needed when the discontinuity moves across a cell center point. To describe these changes, we assume for now that the discontinuity moves across a cell center between multistage updates of the flow in pseudo-time. The method we use to determine the discontinuity movement is described in the next section. When the discontinuity moves across a cell center, we make the following changes which are consistent with the philosophy of treating the discontinuity as a moving boundary separating two distinct computational regions. First, the density and viscosity values of the cell which was crossed are changed to values corresponding to its new computational region. Second, the estimate for the solution at time level $n+1$, $\mathbf{w}_{i,j}$, is changed to $\mathbf{d}_{i,j}$, the dummy value at the cell before it was crossed by the discontinuity. The reason for this will be made clear shortly. Third, the evaluation of the implicit time derivative is modified to provide an estimate of the time derivative in the cell's new computational region. The new estimate is based on the dummy values for that point at previous time steps, $\mathbf{d}_{i,j}^{n}$ and $\mathbf{d}_{i,j}^{n-1}$,

$$\frac{\partial \mathbf{w}_{i,j}}{\partial t} = \frac{\left[3\mathbf{w}_{i,j} - 4\mathbf{d}_{i,j}^{n} + \mathbf{d}_{i,j}^{n-1}\right]}{2\Delta t}. \tag{19}$$

Figure 6 shows typical ρv profiles at three implicit time levels with a moving discontinuity to clarify the logic for this approach. By using the dummy values, an accurate estimate of the solution and the time derivative in the cell's new computational region is obtained.

After the above changes are made, the pseudo-time evolution is continued until a converged solution for $\mathbf{w}^{n+1}$ is obtained. This approach can be used as long as the discontinuity does not cross more than one cell every two implicit time steps. If the interface moves faster

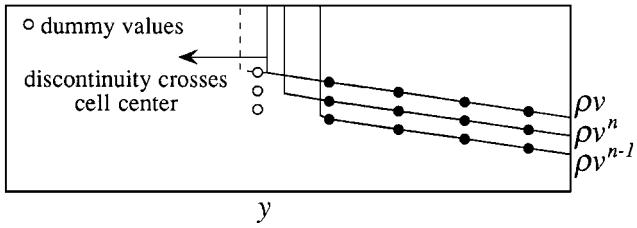

FIG. 6. ρv profiles at three consecutive time levels.

than this, the same general approach can be used, but we must extrapolate the solution at previous time steps more than one cell across the interface to form the time derivative.

Given the movement of the discontinuity, the above approach can be used to find accurate solutions for the evolution of the incompressible flow with fairly general jump conditions. However, there were some difficulties with this approach. One such difficulty is that for discontinuities which are convoluted on the scale of the grid resolution, there may not be a large enough stencil of points in *either* coordinate direction to extrapolate the dummy values. A circle of two cell diameter is a simple example. In general, when this problem develops it means that the grid resolution must be increased, although there are cases in which this occurs independent of the grid resolution. An example is the merging of two nearly planar surfaces. Independent of the grid resolution, the surfaces will eventually become close enough together such that there are only two points across the gap between them, and thus the dummmy values cannot be extrapolated. If this problem developed during a calculation, the calculation was aborted.

As discussed in the physical definition of the problem, another difficulty with the method is that the jump conditions cannot be a function of the spatial derivatives of the flow variables at the discontinuity. The reason for this can now be made clear. In the procedure for determining the dummy values, a one-dimensional stencil of points is used to extrapolate the values of the flow variables at the discontinuity (Eq. (13)). To extrapolate values of the spatial derivatives, $\partial \mathbf{w}/\partial x$ and $\partial \mathbf{w}/\partial y$, which are needed to evaluate viscous jump conditions, a two-dimensional stencil of points is needed on both sides of the discontinuity. For even moderately convoluted discontinuity geometries relative to the numerical grid, it becomes difficult to find a stencil of points for the extrapolation which results in both a stable and accurate scheme. A possible extension of this method which avoids this problem and the limitation discussed in the preceding paragraph is discussed under Conclusion.

DETERMINATION OF THE POSITION OF THE DISCONTINUITY

To determine the evolution of the position of the discontinuity in time, we use a level set formulation [18]. In this formulation, a scalar field, $\phi(x, y)$, is defined such that the initial position of the surface of discontinuity coincides with the zero level of the scalar field. The position of the discontinuity is then tracked in time by updating the scalar field using the equation [18]

$$\frac{\partial \phi}{\partial t} + (u, v) \cdot \nabla \phi = S|\nabla \phi|. \tag{20}$$

This equation propagates level surfaces of ϕ with a velocity of $(u, v) \cdot \mathbf{n} - S$ along the normal to the level surface, where $\mathbf{n}$ is the normal. By setting the parameter S to zero, we

can track a discontinuity which is passively convected by the flow. With a nonzero S, the surface actively propagates relative to the underlying flow. Most of the problems studied in the Results section involve only passive propagation of the discontinuity so that S is zero. However, for the premixed flame problem presented in the Results a nonzero S is required, and therefore we must account for this term in the formulation.

If we can solve the above equation, the information needed for the treatment of the discontinuity is easily obtained from ϕ. For example, along a column of cell centers, we can locate the discontinuity by checking for a change in sign in ϕ and then linearly interpolating between the two cell centers to determine the zero point. To determine the normal at this point, we first calculate the normal to ϕ at the cell center points adjacent to the discontinuity. These are calculated using a central difference of the formula

$$(n_x, n_y) = \nabla\phi/|\nabla\phi|. \tag{21}$$

We then linearly interpolate between these values to determine the normal at the zero point of ϕ. The curvature at the discontinuity point is obtained in a similar manner. This information is then used in the procedure of the preceding section to determine the dummy values.

To solve Eq. (20) numerically, we must discretize the spatial and temporal derivatives. Many successful spatial discretizations of this equation have been developed. We refer the reader to [18, 28] for well-described examples. In this discussion, we focus on the time discretization of this equation. In order to provide updates of the position of the discontinuity in time which are compatible with the time updates of the incompressible flow, we use the same implicit time discretization as the incompressible flow algorithm. Similar to the incompressible flow equations, this results in coupled nonlinear equations of the form

$$\frac{[3\phi^{n+1} - 4\phi^n + \phi^{n-1}]}{2\Delta t} + (u, v)^{n+1} \cdot \nabla\phi^{n+1} = S|\nabla\phi^{n+1}|. \tag{22}$$

Because the update of the position of the discontinuity depends on the update of the flow variables through $(u, v)^{n+1}$, and the update of the flow depends on the position of the discontinuity, the incompressible flow equations and Eq. (22) must be solved simultaneously. Fortunately, this can be easily accomplished in a pseudo-time iteration framework by adding the following equations to the pseudo-time evolution:

$$E_\phi(\phi, \mathbf{w}) = \frac{[3\phi - 4\phi^n + \phi^{n-1}]}{2\Delta t} + R_\phi(\mathbf{w}, \phi) \tag{23}$$

$$\frac{\partial\phi}{\partial t^*} + E_\phi(\phi, \mathbf{w}) = 0 \tag{24}$$

$$\phi|_{t^*=0} = \phi^n; \tag{25}$$

here R_ϕ is a shorthand notation for the spatial derivative terms in Eq. (22). The evolution of ϕ in t^* is performed with the same explicit five-stage scheme as is used for the incompressible flow equations and also uses a point-implicit treatment of the time-derivative term. Since the point-implicit treatment of these terms minimally affects the stability of an explicit scheme, these terms can be neglected in determining the allowable time step for the explicit update. In this case, the pseudo-time equation for ϕ in t^* is equivalent to the original level-set evolution equation in t. The analysis of the local pseudo-time step limit is then analogous to that found in [18, 28], to which, rather than restate the analysis here, we refer

the reader. To achieve the maximum convergence rate to steady state, we again used local time stepping in which the maximum stable pseudo-time step is taken at each computational cell.

The evolution of ϕ and the flow variables in t^* was performed by alternating between advancing the level-set function in t^* and advancing the flow variables in t^*. This is permissible because only the steady-state solution in pseudo-time is of interest; the details of the evolution are immaterial. The main reason that this was done was so that we could implement the changes required when the discontinuity crossed over cell centers. For example, we advance ϕ in t^* and then check for a change in sign in ϕ at any cell center. This indicates that the cell center is now on the opposite side of the discontinuity since the discontinuity is described by the zero level of ϕ. For the cells that were crossed we implement the required changes and then proceed to advance the flow variables in t^*. In this way, the flow variables in each computational region are consistent with the position of the discontinuity during the update of the flow.

Another reason for alternating between the equations for the flow variables and for ϕ is that the changes made when a cell center is crossed often lead to an increase in the error as defined by Eq. (6) at that cell and the neighboring cells. As such, it was desirable to have most of the crossing of the cell centers occur early in the pseudo-time evolution. Thus, for the first evolution of ϕ a large number (four to six) of pseudo-time steps were taken to ensure that the discontinuity was fairly close to its final position even before the evolution of the flow was begun.

For some implicit time steps, a situation arose in which the discontinuity would oscillate back and forth over a cell center, preventing the pseudo-time iteration from reaching a converged solution. By using an alternating approach to the advancement of ϕ and $\mathbf{w}$, we were able to cope with this problem. When any cell center changed signs more than twice, we simply stopped the update of ϕ in pseudo-time and continued the evolution for the flow variables. Since, as discussed in the preceding paragraph, the discontinuity is fairly close to its final position even after the very first update of ϕ, this should not lead to a very significant error for that time step. For the calculations we performed, this problem occurred at the most once every 100 implicit time steps.

The final reason for alternating between the evolution for the flow variables and the evolution for ϕ is that the amount of work spent on each could be controlled. In performing the calculations, we found that we could reduce the frequency of updates on ϕ without adversely affecting the rate of convergence to steady state. We also found that the multigrid acceleration technique used to accelerate the convergence of the flow equations was not as effective on the equation for ϕ. As such, for the calculations one explicit update of ϕ in t^* was performed every 5–10 multigrid updates of the flow equations. A discussion of the multigrid method used for the flow equations is given in the Appendix.

In implementing the above approach, we determined that independent of the chosen spatial discretization scheme, only a spatially first-order accurate prediction of the motion of the discontinuity was obtained. It was subsequently determined that the first-order error arises because of the nature of the solutions near the discontinuity. Consider curve 1 in Fig. 7, which is a typical profile of the normal velocity of the scalar field evaluated along a column of cell centers which crosses the discontinuity. For all of the problems studied, the normal velocity of the scalar field was C^0 continuous but not C^1 continuous across the discontinuity. This is shown by curve 1. An update of the scalar field based on a local evaluation of the scalar velocity at each cell center results in underprediction of the zero

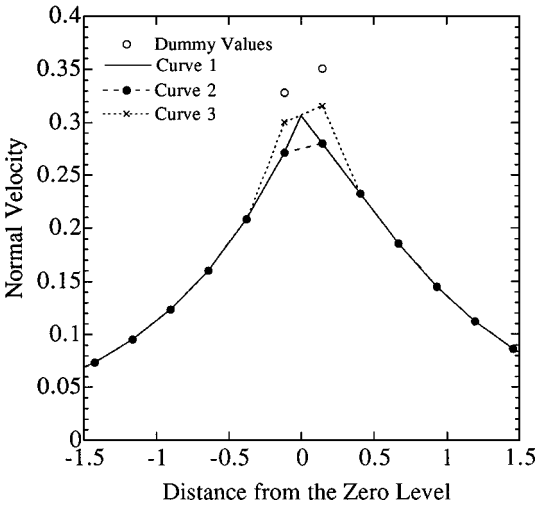

FIG. 7. Profile of the normal velocity of the scalar field evaluated along a column of cell centers which crosses the discontinuity.

level velocity by an amount proportional to the grid size and thus is first-order accurate. This is shown by curve 2. To gain higher-order accuracy, at the points next to the surface we average the local scalar velocity with the scalar velocity calculated with the dummy values. This results in curve 3 shown in Fig. 7, which approximates the discontinuity velocity with second-order accuracy. This problem was not observed in "continuum" approaches which used level sets [7, 23] because the discontinuity in the slope of the scalar velocity is not accurately resolved by these methods.

There was one more modification needed to obtain accurate results using the level-set equation. Because of the nonsmooth nature of the scalar normal velocity shown in Fig. 7, time integration of Eq. (20) caused the ϕ field in the local area around $\phi = 0$ to become convoluted on the scale of the grid resolution. An example of this is shown in Fig. 8, which

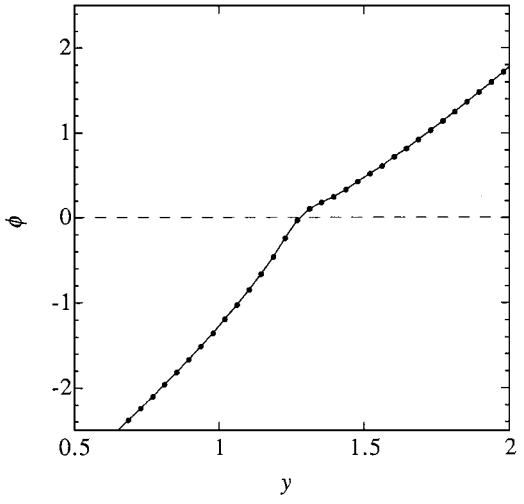

FIG. 8. One-dimensional profile of ϕ versus y showing the convolution of ϕ around $\phi = 0$.

is a one-dimensional profile of ϕ versus y. The rapid change in slope across $\phi = 0$ shown in the figure eventually destroyed the accuracy of the time integration for ϕ. Sussman *et al.* observed a similar effect [23]. To eliminate the problem, they proposed that ϕ be reinitialized after each time update of Eq. (20) with a method which maintains the position of the zero level. This is permissible since only the position of the zero level of ϕ is of importance. To test their method, we initialized ϕ on a 1×1 square as

$$\phi = \begin{cases} 1.1(y + 0.01\sin(2\pi x) - 0.5) & y > 0.5 - 0.01\sin(2\pi x) \\ 0.9(y + 0.01\sin(2\pi x) - 0.5) & y < 0.5 - 0.01\sin(2\pi x) \end{cases} \tag{26}$$

such that the zero level was a small amplitude sine wave and there was a discontinuity in slope of ϕ across the zero level. Applying the reinitialization procedure of Sussman *et al.* to this problem we found that the discontinuity in slope across $\phi = 0$ was eliminated. However, we also found that the reinitialization process caused the amplitude of the zero level sine wave to decay. At full convergence, the amplitude of the sine wave decayed to zero. Sussman *et al.* have recognized this problem and proposed an improved method which is described briefly in [24]. Since the improved method was not available at the time of this work, we developed an alternative approach which is described in the following.

We seek a reinitialization for ϕ which defines a smooth slope through $\phi = 0$ without changing the location of the zero level. To ensure that the position of the zero level is not changed, we fix the values of ϕ at the points adjacent to the zero level during the reinitialization procedure. Since these points determine a gradient for ϕ across the zero level, the reinitialization must extrapolate this gradient into the outer field to define a smooth ϕ. The following equations are used to accomplish this task:

$$\frac{\partial \phi_{i,j}}{\partial \tau} = \text{sign}(\phi_{0,i,j})(\psi_{i,j} - |\nabla \phi_{i,j}|) \tag{27}$$

$$\phi_{i,j}|_{\tau=0} = \phi_{0,i,j} \tag{28}$$

$$\phi_{i,j} = \phi_{0,i,j} \qquad \text{for } i, j = \text{point adjacent to zero level} \tag{29}$$

$$\frac{\partial \psi_{i,j}}{\partial \tau} = -\text{sign}(\phi_{0,i,j})\left(\frac{\nabla \phi_{i,j}}{\epsilon + |\nabla \phi_{i,j}|} \cdot \nabla \psi_{i,j}\right) \tag{30}$$

$$\psi_{i,j}|_{\tau=0} = \psi_{0,i,j} = |\nabla \phi_{0,i,j}| \tag{31}$$

$$\psi_{i,j} = \psi_{0,i,j} \qquad \text{for } i, j = \text{point adjacent to zero level.} \tag{32}$$

Equations (27)–(29) are designed to find a solution for ϕ which has a magnitude of gradient equal to ψ at every point and also is continuous with the values specified for ϕ at the points adjacent to the zero level. Equation (27) is a modification of the equation proposed in [23]. When this equation reaches steady state, the magnitude of the gradient in ϕ evaluated at any location is equal to ψ at that location. Equation (28) states that the initial conditions for ϕ in the reinitialization procedure are given by the solution for ϕ from the last implicit time step which is denoted as $\phi_{0,i,j}$. Equation (29) is a boundary condition specified at the points adjacent to the zero level. This condition is applied so that the reinitialized ϕ passes through the values adjacent to the discontinuity, but the position of $\phi = 0$ does not change during the reinitialization.

The equations for ψ are designed to extrapolate into the outer field the magnitude of the gradient in ϕ determined by the values of ϕ at the points adjacent to the discontinuity. Equation (30) is a convective equation for ψ which propagates the values specified for ψ

at the points adjacent to the zero level into the outer field. Equation (31) gives the initial conditions for ψ which are determined from $|\nabla\phi_{0,i,j}|$, and Eq. (32) fixes the boundary values for ψ at the points adjacent to the zero level. It is these values which are extrapolated into the outer field. In Eq. (30), ϵ is a constant used to avoid division by zero errors at local maximum or minimum of ϕ. This was taken as $0.01\Delta x$.

The spatial discretization of these equations is similar to the first order scheme given in [23]. The discretizations in the y-directions are defined by the following equations. The discretization in the x-direction is analogous:

$$\begin{aligned} s &= \text{sign}(\phi_{0,i,j}) \\ a &= s(\phi_{i,j} - \phi_{i,j-1})/\Delta y \\ b &= -s(\phi_{i,j+1} - \phi_{i,j})/\Delta y \\ c &= (\psi_{i,j} - \psi_{i,j-1})/\Delta y \\ d &= (\psi_{i,j+1} - \psi_{i,j})/\Delta y \end{aligned} \tag{33}$$

$$\begin{aligned} &\text{if } a > b \begin{cases} \frac{\partial \phi_{i,j}}{\partial y} = s\max(a,0) \\ \frac{\partial \psi_{i,j}}{\partial y} = c \end{cases} \\ &\text{if } a \le b \begin{cases} \frac{\partial \phi_{i,j}}{\partial y} = -s\max(b,0) \\ \frac{\partial \psi_{i,j}}{\partial y} = d \end{cases} . \end{aligned} \tag{34}$$

In regions without local extrema, this discretization is a first-order upwind scheme based on the direction of the signed normal, $\text{sign}(\phi_0)\nabla\phi/|\nabla\phi|$. The first-order scheme is chosen for two reasons. First, its diffusive properties help to smooth ϕ. The second reason is that with the above scheme, the values of $\psi_{0,i,j}$ calculated at the points adjacent to the discontinuity are completely determined by the values of ϕ at those points. This is important because it ensures that the reinitialized solution for ϕ will smoothly pass through the values at the adjacent points. This can be made clear by examining a one-dimensional example. Given the values ϕ_{-1}^-, ϕ_0^-, ϕ_1^+, and ϕ_2^+, where the superscripts indicate the sign of the values and the subscripts indicate the position on a regular spaced one-dimensional grid, we examine the initial evaluation of ψ at the cell centers adjacent to the discontinuity (points 0 and 1). Using the above finite differencing scheme, one can confirm that at point 0 this evaluation is $|(\phi_1 - \phi_0)/\Delta y|$. At point 1, the initial evaluation of ψ is also $|(\phi_1 - \phi_0)/\Delta y|$. Thus, the boundary values of $\psi_{0,i,j}$ specified at points on either side of the discontinuity define a single value for the slope for ϕ through the zero level which is consistent with the values of ϕ fixed at the adjacent points.

The time discretization of Eqs. (27) and (30) is accomplished by employing the same five-stage explicit scheme used for the flow solver [5] with a time stepping limit given by $2.5\Delta x\Delta y/(\Delta x + \Delta y)$. Figure 9 shows the one-dimensional ϕ profile shown in Fig. 8 after reinitialization with this technique. As can be seen, the reinitialization does not affect the values adjacent to the discontinuity, but defines a smooth surface for ϕ which passes through these points.

For most cases, the reinitialization procedure converged without difficulty. However, when either a local minimum for ϕ greater than zero or a local maximum for ϕ less than zero existed, a stable situation developed in which information for ϕ and ψ was convected away

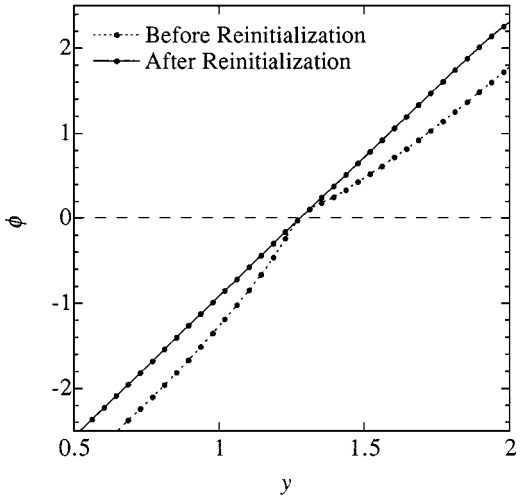

FIG. 9. Reinitialization of ϕ.

from these points even though no boundary conditions were specified there. To avoid this, we restricted the minimum value of ψ to be greater than one-half the minimum value of ψ on the boundary. One can confirm from Eq. (27) and the spatial finite difference scheme that this guarantees positive minimum and negative maximum decay. By using this reinitialization process after each implicit time step, we were able to keep the ϕ surface smooth and obtain accurate results for the update of discontinuity position. This is demonstrated by the results presented under Validation.

This defines the major changes in the incompressible flow algorithm needed to solve problems with a surface of discontinuity. At each implicit time step, the above techniques were used to converge to an accurate prediction of the update of the flow variables and discontinuity position. We note that in the following calculations, the pseudo-time iteration was not converged to machine order accurate steady-state solutions at each implicit time step. Instead, to save computational time, we advanced in pseudo-time until the maximum error in each of the pseudo-time equations (three equations for the flow, one for ϕ) decreased at least three orders of magnitude. Even two orders of magnitude convergence was enough to make the results of all of the following calculations insensitive to further iteration in pseudo-time.

VALIDATION

To validate the method, we have simulated the propagation of surface waves. These waves occur when a light fluid is layered over a more dense fluid. Under the influence of gravity, waves such as those seen on the surface of the ocean propagate along the surface of the two fluids. For the validation process, we examine periodic arrays of these waves occurring in inviscid fluids. In this case, steady wave profiles exist which propagate with a constant velocity. When the problem is made nondimensional using the wavenumber and the acceleration of gravity, the wave profile and velocity can be described uniquely as a function of a specified wave amplitude, h, and the Atwood number, $\mathrm{At} = (\rho_1 - \rho_2)/(\rho_1 + \rho_2)$. We have obtained a description of these profiles as well as the velocity at which the waves

propagate using the boundary integral method given in the Appendix of [2]. This information is used to evaluate the accuracy of the unsteady numerical simulation.

Two different test cases are studied. The first is for an Atwood number, At, of 1.0 and a wave amplitude of 0.6 measured from trough to peak. This case corresponds to a free surface wave. For the free surface problem, the pressure along the surface is fixed at 0, the velocity at the discontinuity is determined by solving the incompressible flow equations in the fluid, and the motion of the surface is determined by the velocity of the fluid at the surface. Since our method is designed for flows of two fluids, we define a complementary problem to be solved above the free surface. This problem is the incompressible flow of a fluid on a moving wall. The moving wall for this problem is the free surface whose motion is completely determined by fluid 1, the free surface fluid. The condition applied at the discontinuity for fluid 2 is then that the normal velocity of fluid 2 evaluated at the wall is equal to the normal velocity of the wall. A slip velocity is allowed between the wall and fluid 2 such that there are no conditions on the tangential velocity of fluid 2 at the wall.

These conditions are somewhat simpler than a two-fluid problem in which there is interaction across the discontinuity. As such, we can make some simplifications in determining the dummy values for both fluids. In the determination of the dummy values for fluid 1, we make the simplification

$$\mathbf{s}_1' = \text{diag}[0, 1, 1] \cdot \mathbf{s}_1 = \begin{Bmatrix} 0 \\ \rho_1 u_1 \\ \rho_1 v_1 \end{Bmatrix}; \tag{35}$$

these values are chosen since the pressure on the free surface is zero and the information for the velocity at the free surface must come from fluid 1, not fluid 2. For fluid 2, the following equations are used to determine $\mathbf{s}_2'$:

$$\begin{aligned} p_2' &= p_2 \\ (u_2', v_2') \cdot \mathbf{n} &= (u_1', v_1') \cdot \mathbf{n} \\ (u_2', v_2') \times \mathbf{n} &= (u_2, v_2) \times \mathbf{n}. \end{aligned} \tag{36}$$

These equations reflect the fact that the normal velocity of the wall is determined by the free surface fluid, while the tangential velocity and pressure at the wall in fluid 2 are determined by solving the incompressible flow equations in fluid 2. Given values $\mathbf{s}_1'$ and $\mathbf{s}_2'$, we can extrapolate dummy values and evaluate the derivatives near the discontinuity. The motion of the discontinuity is determined by the level-set equation with S equal to zero.

To complete the definition of the problem we must specify initial conditions and far-field boundary conditions for fluids 1 and 2. Initial conditions for the flow and the surface position are determined using the method of [2] with 180 points describing the surface position. For fluid 1, which is the lower of the two fluids, Euler (slip) boundary conditions are applied at the bottom boundary. For fluid 2, at the upper far-field boundary a nonreflecting boundary condition with $u, v, p = 0$ is used. This boundary condition is discussed in [5]. The domain height was taken as $4/3$ the wavelength of the wave with the wave at the midheight of the domain. At this domain height, the solution was minimally affected by increases in the height of the domain.

Figures 10a and 10b show the wave profiles calculated with the simulation at $t = 0$ and $t = 6.25$ which is approximately one period later. For comparison, the steady profile shifted by ct is also shown, where c is the velocity of the wave calculated to be 1.082 from the

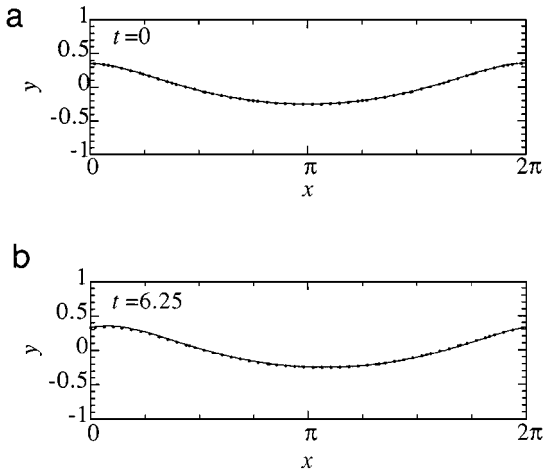

FIG. 10. Free surface wave profiles (a) at $t = 0$ and (b) at $t = 6.25$.

stationary solution. The results shown were computed with an equally spaced Cartesian mesh with 48 cells in the x-direction, 64 cells in the y-direction, and an implicit time step of 0.05. Qualitatively, the results agree well with the stationary solution. We make this comparison more quantitative after describing the second test problem.

The second test case is for an Atwood number of 0.8181 and an amplitude of 0.72. This case corresponds to an interfacial wave between two immiscible fluids. At the interface of the two fluids, the pressure and the normal velocity are required to be continuous, and there is a slip condition for the tangential velocity. These jump conditions are implemented as follows. To determine $\mathbf{s}_{1,\text{jump}}$ from $\mathbf{s}_2$ the following equations are used:

$$\begin{aligned} p_{1,\text{jump}} &= p_2 \\ (u_{1,\text{jump}}, v_{1,\text{jump}}) \cdot \mathbf{n} &= (u_2, v_2) \cdot \mathbf{n} \\ (u_{1,\text{jump}}, v_{1,\text{jump}}) \times \mathbf{n} &= (u_1, v_1) \times \mathbf{n}. \end{aligned} \tag{37}$$

These values are then used to determine $\mathbf{s}_1'$ as discussed in the section describing the discontinuity treatment. Given $\mathbf{s}_1'$, $\mathbf{s}_2'$ is determined by the equations

$$\begin{aligned} p_2' &= p_1' \\ (u_2', v_2') \cdot \mathbf{n} &= (u_1', v_1') \cdot \mathbf{n} \\ (u_2', v_2') \times \mathbf{n} &= (u_2, v_2) \times \mathbf{n}. \end{aligned} \tag{38}$$

This implementation of the jump conditions reflects the fact that there is no relation between the tangential velocity of the fluids on either side of the discontinuity. As such, the information for the tangential velocity on either side of the discontinuity must be extrapolated from the flow on that side of the discontinuity.

The initial conditions for the problem were again determined by using the method of [2]. The far-field boundary conditions for this problem are the same as those used for the previous problem. As in Fig. 10, Figs. 11a and 11b show the wave profiles at a time of 0 and 6.25 along with the shifted stationary profile. For this case, the wave speed, c, is equal to 0.963. These results were calculated on a 48×64 mesh with a time step of 0.05. Again, the results agree qualitatively well with the stationary solution.

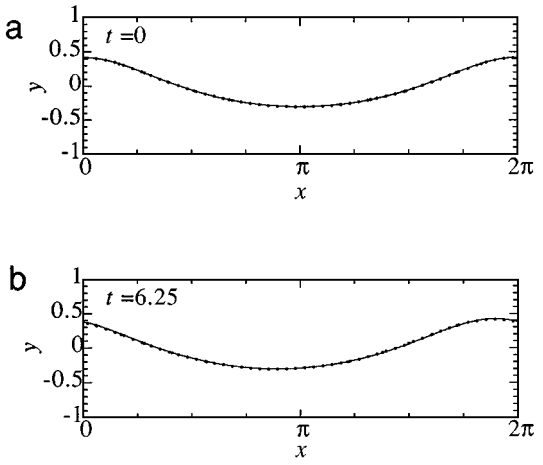

FIG. 11. Interfacial wave profiles (a) at $t = 0$ and (b) at $t = 6.25$.

ERROR ANALYSIS

To make a quantitative assessment of the accuracy of the method, we need to estimate the error in the solution as a function of time. To do this, we choose the error estimate

$$e(t) = \sqrt{1/2\pi h^2 \int_0^{2\pi} (y_0(x,t) - y_s(x-ct))^2 \, dx}, \tag{39}$$

where y_0 is the y coordinate of the surface and y_s is the y coordinate of the stationary solution. We chose this estimate because it is easy to calculate and because the surface position is usually the most examined result of the calculations.

Figure 12 shows the evolution of the error for the free surface problem calculated on a 24×32, 48×64, and 96×128 mesh all with the same implicit time step, 0.05. By

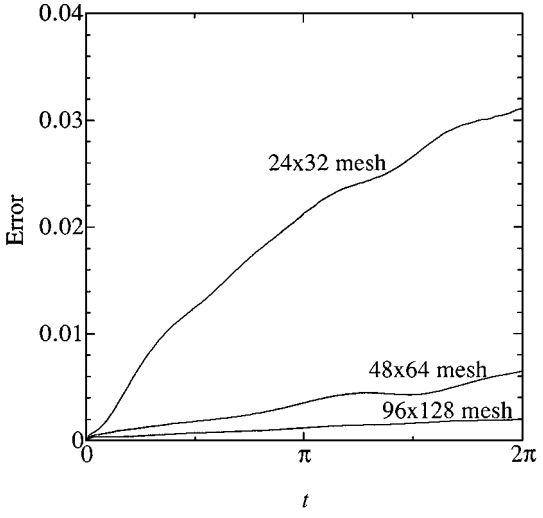

FIG. 12. Error growth for the free surface wave.

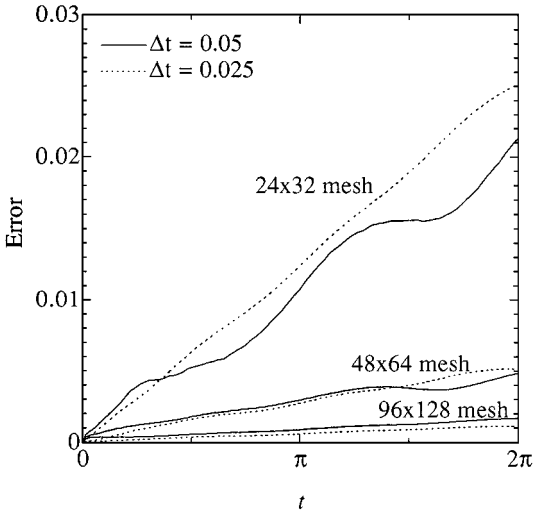

FIG. 13. Error growth for the interfacial wave.

comparing the slope of these curves, we can estimate the spatial order of accuracy of the scheme. Fitting lines to these curves, we find that the ratio of the slopes is approximately 1/4 when moving to a finer mesh. The actual values are 0.20 and 0.29. Therefore, the convergence is approximately second-order accurate in space. We have performed a similar analysis on the interfacial wave problem. The error growth for this problem is shown in Fig. 13. For this case we have presented results for both an implicit time step of 0.05 and 0.025 to ensure that time discretization was not significantly affecting the growth of the error. First, we see that the slopes of the curve for each grid size are approximately the same, which shows that the growth rate is predominately determined by the spatial discretization error for these time step sizes. Second, if we compare the slopes of the curves found with equal time step size, we find that the ratio of slopes as the grid size is doubled is again approximately 1/4. The values are 0.21 and 0.30 for $\Delta t = 0.05$ and 0.21, and 0.20 for $\Delta t = 0.025$. Thus, for inviscid problems, we can achieve second-order accuracy. We were not able to validate the method for any problems with viscosity, because all of the viscous solutions which were available for comparison had jump conditions which were a function of the spatial derivatives of the flow variables at the surface of discontinuity.

To justify our statement that the error with this approach is much less than that of a continuum method, we compare the order of convergence of this approach to that given for a continuum method. In [23], a convergence study was done for the simulation of a rising bubble using the continuum approach. The order of convergence given for the error in the length of the minor axis of the bubble, an error measure similar to that used in this paper, is approximately 1.6. If we consider that this value was obtained by decreasing both the grid size and the time step simultaneously, it is reasonable to estimate the spatial order of accuracy as only somewhat greater than unity. For our approach, if we use from the previous paragraph the value 0.21 as a typical value for the ratio of the errors on successive meshes, we estimate the spatial order of accuracy as $\ln(0.21)/\ln(1/2) = 2.25$. This comparison is not completely fair because the problem studied in [23] had the additional complications of surface tension and viscosity. In the next paragraph we will provide further arguments showing that this approach is more accurate than a continuum method for inviscid flows.

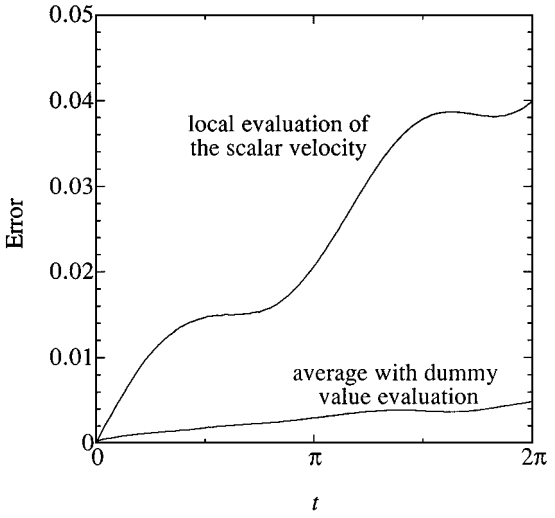

FIG. 14. Dependence of the error growth on the evaluation of the scalar velocity for the interfacial wave problem on a 48×64 mesh with $\Delta t = 0.05$.

To demonstrate the importance of accurately resolving the peak in the normal velocity of the scalar field shown in Fig. 7, we examine the error growth when the normal velocity is evaluated by averaging the local and dummy values at the points adjacent to the discontinuity versus the case where we simply use the local values. The results for $\mathrm{At} = 0.8181$ were recalculated on the 48×64 mesh using only the local scalar velocity at each point to determine the update for ϕ. In Fig. 14, the error in these results is shown along with the error obtained using the averaging technique. From this figure, we see that a local evaluation of the scalar velocity causes a much larger error. By using the averaging technique, we avoid this error and achieve second-order accuracy. From Figs. 7 and 14, we also argue that the error obtained using a continuum method for this problem would be of the same order as or larger than the error obtained using the local values only to evaluate the normal scalar velocity. This argument is based on the fact that if an artificial smoothing width is introduced into Fig. 7, we obtain a scalar normal velocity profile similar to that shown in Fig. 15. In the smoothed profile, there is an error in the prediction of the interface velocity which is proportional to the smoothing width. This error is similar to that which results by using the local velocity only to calculate the scalar normal velocity, and thus the error using a continuum method will be of the same order or larger depending on the number of grid cells over which the discontinuity is smoothed.

Finally, to investigate the importance of reinitializing the scalar surface, we have recalculated the results for $\mathrm{At} = 0.8181$ without reinitializing the surface after each implicit time step. The error on the 48×64 mesh without reinitialization along with the previous result is shown in Fig. 16. Initially the results agree, but as time progresses the scalar surface becomes convoluted, resulting in the rapid increase in the error which occurs at later times. Also in Fig. 16, results are shown using 10 reinitialization time steps as opposed to the 5 time steps used in the previous calculations. The results for the two cases are nearly indistinguishable, and thus 5 time steps is enough to ensure that the results are independent of the reinitialization process.

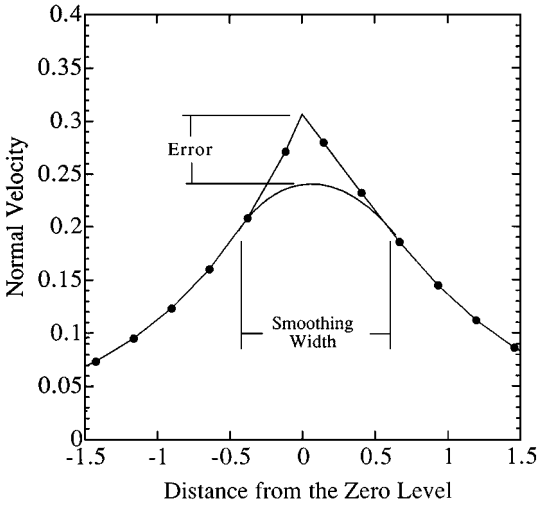

FIG. 15. Profile of the normal velocity of the scalar field showing the effect of smoothing in a continuum approach.

PREMIXED FLAME PROPAGATION

As a demonstration of the versatility of the computational method and to provide some groundwork for our future studies, we present the implementation of our method and the results of two calculations for the premixed flame propagation problem. The results verify the accuracy of the implementation and in addition demonstrate that our technique can be used for viscous flows and for more convoluted and dynamic surfaces than in the previous test cases.

In many combustion devices, the scale of the flow is much larger than the flame thickness. In this limit, a premixed flame can be approximated by a discontinuity in the flow field.

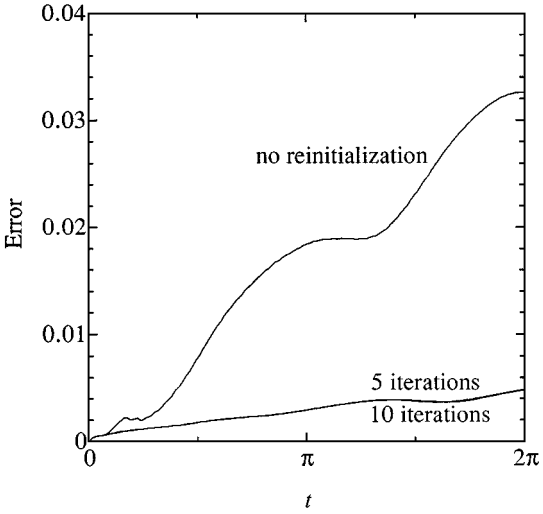

FIG. 16. Dependence of the error growth on the reinitialization of ϕ for the interfacial wave problem on a 48×64 mesh with $\Delta t = 0.05$.

Asymptotic analysis [17] has shown that to first order in the ratio of the flow scale to the flame thickness, the following jump conditions describe this discontinuity:

$$\begin{aligned} p_1 - p_2 &= m^2\left(\frac{1}{\rho_2} - \frac{1}{\rho_1}\right) \\ u_1 - u_2 &= -n_x m\left(\frac{1}{\rho_2} - \frac{1}{\rho_1}\right) \\ v_1 - v_2 &= -n_y m\left(\frac{1}{\rho_2} - \frac{1}{\rho_1}\right); \end{aligned} \tag{40}$$

here the subscripts 1 and 2 refer to the unburned and burned gas, respectively; m is the mass burning rate of the flame; and the scalar ϕ is initialized so that the normal as defined by Eq. (21) points toward the burned gas. These conditions reflect the requirements that mass and momentum are conserved across the flame and that the velocity tangent to the flame does not change across the discontinuity. The analysis also showed that the discontinuity propagates along its normal relative to the unburned flow with a speed of m/ρ_1, which reflects the fact that the flame is consuming unburned fluid. To implement this numerically, m/ρ was used for the speed, S, in Eq. (20). Although this speed is discontinuous across the surface, when combined with the discontinuity of the flow variables at the surface, it defines a C^0 continuous velocity for the scalar normal velocity, $(u, v) \cdot \mathbf{n} - m/\rho$, which propagates the flame with a speed of m/ρ_1 relative to the unburned gas.

To close the problem, the mass burning rate, m, and the density jump across the flame must be specified. These parameters are determined by the chemical and diffusive effects which occur within the flame. To leading order, m and the jump in density across the flame are independent of the flame shape and flow velocity [17]; therefore we can specify constant values for both to describe a specific fuel. Typical values for a hydrocarbon such as methane mixed in stochiometric proportions with air are approximately $0.035\ \mathrm{g/cm^2\ s}$ and $\rho_1/\rho_2 = 7$. In our calculations, we make the flow velocities nondimensional by m/ρ_1, which reduces the number of independent variables by one. With ρ_1/ρ_2 specified, the problem is closed and we can proceed with the calculations.

To verify the accuracy of the flame implementation, we compare the computational solution to the analytic solution for the Landau–Darrieus instability [15]. For the first-order flame approximations given above, this solution shows that perturbations to a planar flame grow exponentially with a growth rate, ω, of

$$\omega = \frac{km}{\rho_1 + \rho_2}\left[-1 + \left(1 + \frac{\rho_1}{\rho_2} - \frac{\rho_2}{\rho_1}\right)^{1/2}\right], \tag{41}$$

where k is the wavenumber of the disturbance. Figure 17 shows the amplitude growth of a sinusoidal perturbation to a planar flame for various values of ρ_1/ρ_2 calculated for both a 24×32 and a 48×64 grid. The analytic solution of Eq. (41) is also shown. The analytic and numerical solutions agree very well for both grid sizes, confirming that the simulation accurately captures flame behavior.

We have also simulated the interaction of a flame sheet with a vortex. This problem has received much attention since it can give insight into turbulent flame dynamics. Previous examinations of the problem have either treated the flame as a passive surface neglecting the gas expansion which occurs across the flame [1, 11, 13, 27, 28] or studied vortices which

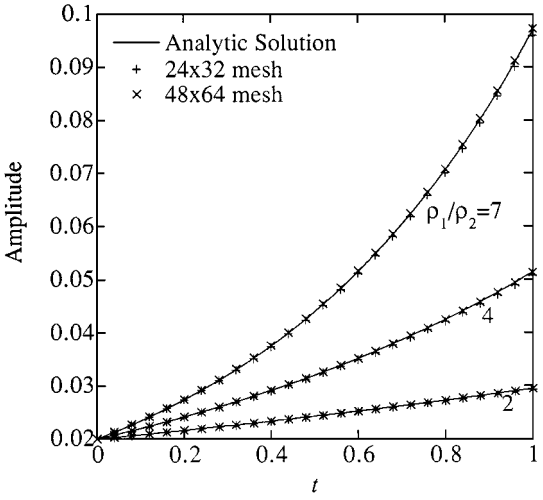

FIG. 17. Perturbation amplitude versus time for various density ratio premixed flames.

were on the same scale as the flame thickness, which reduces the resolution requirements of the problem [19, 22]. With our method, since we do not need any points to resolve the flame structure, we can more accurately and efficiently capture the dynamics of the interaction when the scale of the viscous vortex is much larger than the flame thickness.

To perform the simulation, rather than using a constant burning rate which is unstable to all wavelengths of disturbance, we use a burning rate which has a dependence on the curvature

$$m = 1 + \delta \nabla \cdot (n_x, n_y). \tag{42}$$

Equation (42) is a heuristic representation of the effect of the flame thickness on the mass burning rate; therefore δ should be of the order of the width of the flame relative to the flow scale. This effect damps the growth of small scale instabilities, allowing the problem to be resolved on the numerical grid. Physical scaling arguments reveal that δ and the Reynolds number are related through the Prandtl number, Re $\delta = 1/\mathrm{Pr}$. Thus for the vortex problem, since Pr is always near unity, the Reynolds number and δ are not independent. For the results we present, δ was taken to be 0.1, and the Reynolds number was calculated using a Prandtl number of 0.7. The jump in viscosity across the flame which occurs due to heat release in the flame is determined by the relationship between kinematic viscosity and density at constant pressure

$$\frac{\nu_1}{\nu_2} = \left(\frac{\rho_2}{\rho_1}\right)^{3/2}. \tag{43}$$

The pressure can be assumed constant to determine this relation because the problem we studied was unconfined and in the low Mach number approximation. Under these conditions, to first order the pressure is constant in an expansion in powers of the Mach number. This completes the definition of all the parameters in the problem.

To initiate the flame–vortex interaction, a potential solution for a vortex was added to the solution for a planar flame. For the test case we studied, the circulation of the vortex was 15. A schematic of the initial conditions and boundary conditions is shown in Fig. 18. A

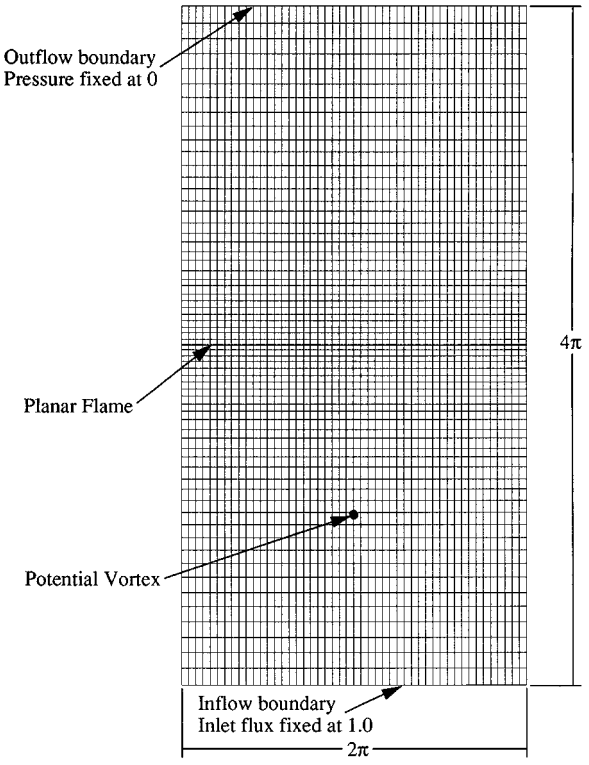

FIG. 18. Schematic of the grid and the initial and boundary conditions used for the flame–vortex problem.

constant inlet velocity of 1.0 was used so that the flame did not propagate out of the domain. At the exit, the pressure was fixed at zero, while the u and v velocities for the boundary were extrapolated from the interior of the domain. Calculations were performed on a 48×64 mesh with the cells compressed near the center of the domain for higher resolution of the flow features near the flame surface. All of the extrapolations for the dummy cells were done in i, j coordinates so that no modifications were needed to accommodate the variation of the mesh spacing.

In Fig. 19, flame profiles from this calculation are shown for every 10th implicit time step where the implicit time step was 0.075. The initial point vortex decays due to viscosity as

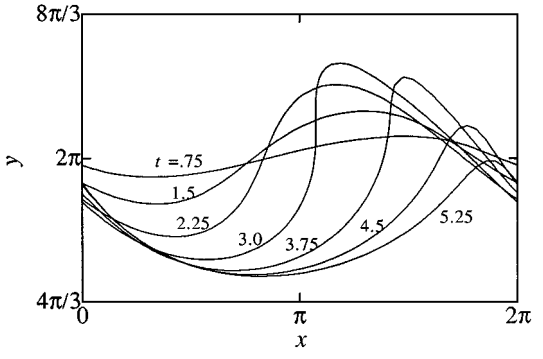

FIG. 19. Flame contours for the flame–vortex problem.

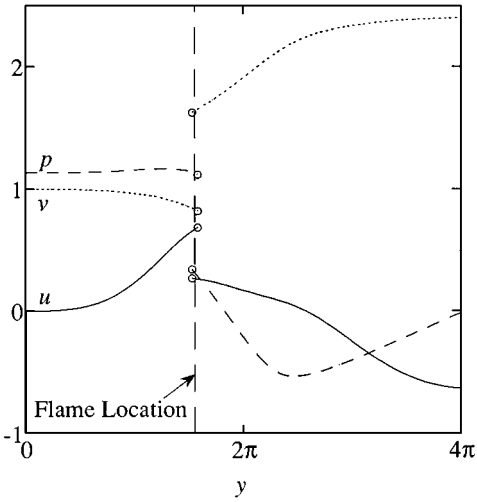

FIG. 20. Cross section of the flow variables at $x = \pi$ and $t = 3.75$.

it approaches the flame, but still has enough strength to cause a significant convolution of the flame surface. Our method for the treatment of the discontinuity was able to handle the flame geometry shown in Fig. 19 with no noticeable change in the quality of the solution. Figure 20 shows a vertical cross section of the flow variables with dummy values at $x = \pi/4$ and $t = 3.75$. The solution is captured without any smoothing or oscillation near the discontinuity. This gives not only greater accuracy, but also a clearer picture of the dynamics of the flow near the discontinuity.

CONCLUSIONS

We have developed an accurate method for solving 2-D incompressible flow problems with an internal discontinuity surface. This method addressed three issues associated with these problems: the solution of the incompressible flow equations, the treatment of the discontinuity in flow variables at the surface, and the tracking of the surface as it moves within the domain. Results and analysis have shown that for inviscid problems, the method is spatially second-order accurate. The calculations also showed that the method was fairly versatile in its ability to handle a variety of different jump conditions such as those describing a free surface, an immiscible fluid interface, and a premixed laminar flame.

The main limitation of the method is that the jump conditions cannot be a function of the spatial derivatives of the flow variables at the discontinuity. To handle these jump conditions estimates for the spatial derivatives must be extrapolated to the surface of discontinuity from the outer flow. On a fixed mesh, it is difficult to perform these extrapolations for moderately convoluted surface geometries. A natural extension of this method is to implement the technique on an unstructured mesh which moves with the discontinuity. In this way, the stencil of points near the discontinuity would remain fixed, allowing the derivative information to be extrapolated more easily. A moving unstructured mesh formulation has been recently implemented for inviscid free surface problems [16]. By combining the techniques given here and those of [16], it may be possible to solve two-fluid viscous problems very accurately.

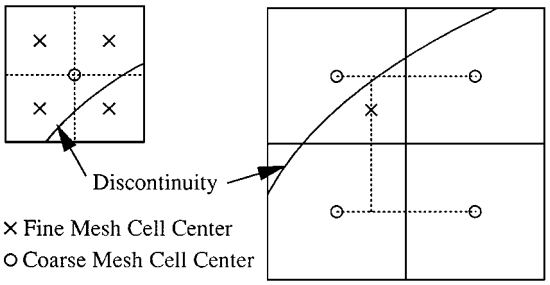

FIG. 21. Schematic describing the movement between the coarse and fine mesh.

APPENDIX: MULTIGRID WITH A DISCONTINUITY

The acceleration of convergence to steady state by the multigrid technique is crucial to the efficiency of the incompressible flow algorithm. However, it is beyond the scope of this paper to describe the full details of the technique. A detailed description can be found in [5]. Here, we focus on the changes needed to accommodate the surface of discontinuity.

There are basically three steps in the multigrid technique. In the first step, an estimate of the solution and the error in the solution as defined by Eq. (7) is transferred to a coarser mesh. This is normally done with a straightforward volume average. The first schematic in Fig. 21 shows a typical stencil of fine mesh cells used in the volume average for a coarse mesh cell. To transfer the discontinuous solution, we use a slightly altered volume average. This is given by the equations

$$\phi_c = \sum_{n=1}^{4} V_n \phi_n \Big/ \sum_{n=1}^{4} V_n \tag{44}$$

$$\mathbf{w}_c = \sum_{n=1}^{4} V_n[(\text{sign}(\phi_n) + \text{sign}(\phi_c))\mathbf{w}_n + (\text{sign}(\phi_n) - \text{sign}(\phi_c))\mathbf{d}_n]/2 \sum_{n=1}^{4} V_n, \tag{45}$$

where V_n is the volume of the fine mesh cells 1–4 shown in Fig. 21, and $\mathbf{w}_n$ and $\mathbf{d}_n$ are the flow values and dummy values for those cells. ϕ_c and $\mathbf{w}_c$ are then the volume-averaged values on the coarse mesh cell. By avoiding an average which uses $\mathbf{w}$ values from both sides of the discontinuity, a solution on the coarse mesh is produced which retains its discontinuous nature. In addition, because the average is based on the sign of ϕ on the fine and coarse mesh cells, the flow values on the coarse mesh are consistent with the side of the discontinuity that the coarse cell center is on as determined by the value of ϕ on the coarse mesh.

The volume average used to transfer an estimate of the error cannot be done in a similar manner because there are no dummy values of the error. Instead, we volume average over only those fine mesh cells which have the same sign of ϕ as the coarse mesh cell,

$$\mathbf{E}_c = \sum_{n=1}^{4} \mathbf{E}_\mathbf{n} V_n (\text{sign}(\phi_n) + \text{sign}(\phi_c)) \Big/ \sum_{n=1}^{4} V_n (\text{sign}(\phi_n) + \text{sign}(\phi_c)), \tag{46}$$

where $\mathbf{E}_c$ is the error at the coarse mesh cell and $\mathbf{E}_n$ is the fine mesh error. We avoid any coupling of the error across the surface in the average to be consistent with the philosophy of subdividing the numerical domain into two distinct computational regions.

The next step in the multigrid technique is to perform an update of the solution on the coarse mesh in a manner analogous to that used on the fine mesh. This can be accomplished by using all of the same techniques for handling the discontinuity as were used on the fine mesh since the solutions on the coarse and fine mesh have a similar discontinuous form. The only difference in the update is that the error used to drive the pseudo-time update, Eq. (7), is determined by using both the error transferred from the fine mesh and the error evaluated using the coarse mesh solution. A discussion of this can be found in [5]. We remark that although ϕ is transferred to the coarse mesh, this is only for the purpose of determining the position of the discontinuity for the update of the flow solution on the coarse mesh. No updates of ϕ are performed on the coarse mesh.

The final step is to use the change in the solution on the coarse mesh to define a change in the solution at the fine mesh points. This is accomplished by using a bilinear interpolation between the change in solution of the four cells on the coarse mesh which are nearest to the fine mesh cell for which we are finding a correction. A schematic of this is also shown in Fig. 21. This method is adapted to the region near the discontinuity in a manner similar to that used to move the solution to the coarse mesh. If the signs of ϕ on the coarse and fine meshes are the same, the change in the solution on the coarse mesh is used for the bilinear interpolation. If the signs are opposite, the change in the dummy value of the solution on the coarse mesh is used. This defines the update of the solution on the fine mesh, completing the description of the modifications to the multigrid method. The remainder of the method is completely analogous to that described in [5].

ACKNOWLEDGMENTS

This work was sponsored by the Air Force Office of Scientific Research under the technical monitoring of Dr. Mitat Birkan. B. T. H. was in addition supported by an Upton graduate fellowship, while L. M. by the Computational Mathematics Program of AFOSR.

REFERENCES

1. Wm. T. Ashurst, Flame propagation through swirling eddies, a recursive pattern, *Combust. Sci. Technol.* **92**, 87 (1993).
2. G. R. Baker, D. I. Meiron, and S. A. Orszag, Generalized vortex methods for free-surface flow problems, *J. Fluid Mech.* **123**, 477 (1982).
3. J. B. Bell, P. Colella, and H. Glaz, A second-order projection method for the incompressible Navier–Stokes equations, *J. Comput. Phys.* **85**, 237 (1989).
4. A. Belov, L. Martinelli, and A. Jameson, A new implicit algorithm with multigrid for unsteady incompressible flow calculations, AIAA 95-0049 (1995).
5. A. Belov, *A New Multigrid-Driven Algorithm for Unsteady Incompressible Flow Calculations on Parallel Computers*, Ph.D. thesis, Princeton University, June 1997.
6. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1192).
7. Y. C. Chang, T. Y. Hou, B. Merriman, and S. Osher, A level set formulation of Eulerian interface capturing methods for incompressible fluid flows, *J. Comput. Phys.* **124**, 449 (1996).
8. L. P. Chin, Numerical modeling of interactions between evaporating droplets, AIAA 97-0130 (1997).
9. A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2**, 12 (1967).

10. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice Hall, Englewood Cliffs, NJ, 1971).
11. B. T. Helenbrook, C. J. Sung, C. K. Law, and Wm. T. Ashurst, On stretch-affected flame propagation in vortical flows, *Combust. Flame* **104**, 460 (1996).
12. D. Juric, A numerical study of vapor explosions in microgravity, AIAA 97-0568 (1997).
13. A. R. Kerstein, Wm. T. Ashurst, and F. A. Williams, Field equation for interface propagation in an unsteady homogeneous flow field, *Phys. Rev. A* **37**, 2728 (1988).
14. D. B. Kothe and R. C. Mjolsness, RIPPLE: A new model for incompressible flows with free surfaces, *AIAA J.* **30**, 2694 (1992).
15. L. D. Landau and E. M. Lifshitz, *Fluid Mechanics* (Pergamon Press, London, 1959), p. 478.
16. B. H. Liou, L. Martinelli, T. J. Baker, and A. Jameson, submitted for publication.
17. M. Matalon and B. J. Matkowsky, Flames as gasdynamic discontinuities, *J. Fluid Mech.* **124**, 239 (1982).
18. S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
19. T. Poinsot, D. Veynante, and S. Candel, Diagrams of premixed turbulent combustion based on direct simulation, in *Twenty-Third Symposium (International) on Combustion* (The Combustion Institute, Pittsburgh, PA, 1990), p. 613.
20. E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* **130**, 269 (1997).
21. A. Rizzi and L. E. Eriksson, Computation of inviscid incompressible flow with rotation, *J. Fluid Mech.* **153**, 275 (1985).
22. C. J. Rutland, J. H. Ferziger, and S. H. El Tahry, Full numerical simulation and modeling of turbulent premixed flames, in *Twenty-Third Symposium (International) on Combustion* (The Combustion Institute, Pittsburgh, PA, 1990), p. 621.
23. M. Sussman, P. Smereka, and S. Osher, A level set approach for computing solutions to incompressible two-phase flow, *J. Comput. Phys.* **114**, 146 (1994).
24. M. Sussman and P. Smereka, Axisymmetric free boundary problems, *J. Fluid Mech.* **341**, 269 (1997).
25. S. O. Unverdi and G. Tryggvason, Computations of multi-fluid flows, *Physica D* **60**, 70 (1992).
26. S. O. Unverdi and G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* **100**, 25 (1992).
27. M.-S. Wu and J. F. Driscoll, a numerical simulation of a vortex convected through a laminar premixed flame, *Combust. Flame* **91**, 310 (1992).
28. J. Zhu and P. D. Ronney, Simulation of front propagation at large non-dimensional flow disturbance intensities, *Combust. Sci. Technol.* **100**, 183 (1994).