A level set based method for calculating flux densities in two-phase flows

By M. Raessi

1. Motivation and objectives

In many two-phase (liquid/gas) flows encountered in industrial applications, for example, liquid atomizers, boilers and spray systems, fluid properties such as density and viscosity vary by several orders of magnitude across fluid interfaces. This discontinuity in fluid properties is a great challenge in modeling such flows. In particular, at large density ratios (≥ 500), non-physical deformations can appear on the interface if the advection of momentum is not properly modeled.

To further explain this, we consider a finite-volume flow solver with staggered arrangement of variables: the scalars (e.g., pressure or volume fraction) are defined at the center of each numerical cell, and the velocities are on the faces. In this staggered arrangement, the mass and momentum control volumes are offset by half of the cell size. In a two-phase flow, this arrangement can lead to an inconsistent advection of mass and momentum, as illustrated by the following 1-D example.

In Fig. 1, mass control volumes are confined by solid lines (cells i and $i+1$); the momentum control volume is shown by dashed lines (cell $i+1/2$). Initially, cell i contains equal volumes of fluids A (gray) and B (white); cell $i+1$ contains only B. At a time step, as fluid A enters the momentum control volume, the same volume of fluid B must leave cell $i+1/2$; fluids are incompressible. The net momentum per volume, which will be simply referred to as momentum for ease of presentation, transferred into the momentum control volume at this time step is then $\rho_{\mathrm{A}} u_i - \rho_{\mathrm{B}} u_{i+1}$. During this transfer, the density of cell $i+1/2$ changes from ρ_{B} to $F\rho_{\mathrm{A}} + (1-F)\rho_{\mathrm{B}}$, where F is the volume fraction of cell $i+1/2$ occupied by fluid A following advection. Hence, $F = 1$ in a cell fully occupied by fluid A, $F = 0$ in a cell filled with fluid B, and $0 < F < 1$ in a cell containing a portion of the interface.

The traditional approach to approximating the density of momentum fluxes in staggered models is to use densities of mass control volumes. So, for the configuration of Fig. 1, the incoming momentum is calculated as $\rho_i u_i$ instead of $\rho_{\mathrm{A}} u_i$, where

$$\rho_i = \frac{\rho_{\mathrm{A}} + \rho_{\mathrm{B}}}{2},$$

which is not consistent with the density of the mass flux into cell $i+1/2$. For example, if $\rho_{\mathrm{A}} > \rho_{\mathrm{B}}$, the incoming momentum flux is incorrectly approximated as less than the actual value.

In such staggered models, the initial density of the momentum control volume is also calculated inconsistently, as the average density of the cells i and $i+1$; for the configuration of Fig. 1

$$\rho_{i+1/2} = \rho_{\mathrm{B}} + \frac{\rho_{\mathrm{A}} - \rho_{\mathrm{B}}}{4} \neq \rho_{\mathrm{B}},$$

where the error term $(\rho_{\mathrm{A}} - \rho_{\mathrm{B}})/4$ can be large for high-density ratios.

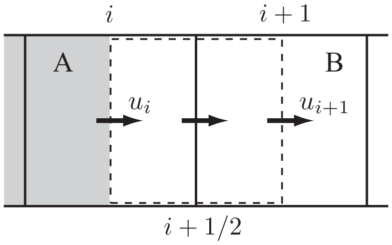

FIGURE 1. Momentum advection in a staggered arrangement of variables.

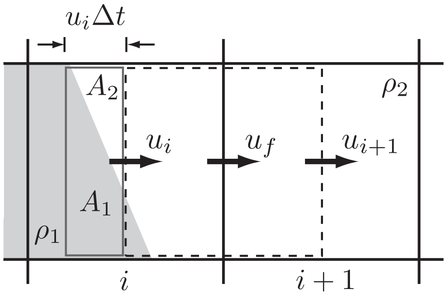

FIGURE 2. Advection of mass and momentum in two-fluid flows.

Although the above example was on a finite-volume flow solver, the same inconsistency in evaluating densities can be present in a finite-difference, staggered flow solver. For example, averaging cell center densities to obtain face densities can generally induce large errors, especially when the density ratio is high

The inconsistency between mass and momentum advection contributes to incorrect accelerations of fluids near interfaces, especially evident at high-density ratios. This has been discussed by Bussmann *et al.* (2002) and Rudman (1998) for incompressible flow solvers that use the VOF method: Bussmann *et al.* (2002) presents a technique for a finite-volume flow solver with collocated arrangement of variables in which mass and momentum control volumes coincide. And, Rudman (1998) gives a technique for finite-difference, staggered flow solvers, where mass is advected on a grid twice as fine as the momentum grid in order to calculate flux densities. It is important to note that in both models, Bussmann *et al.* (2002) and Rudman (1998), the flow equations are solved in the conservative form. And importantly, mass and momentum are advected consistently by correctly calculating the densities of mass fluxes entering and leaving momentum control volumes, as well as the densities of momentum control volumes. For example, consider the case illustrated in Fig. 2 in which the interface is inclined compared to the grids. To calculate the momentum flux across face i, area (in 2-D, or volume in 3-D) A_1 is assigned the density of fluid 1 and A_2 the density of fluid 2. The flux density at face i, denoted by $\tilde{\rho}_i$ is then

$$\tilde{\rho}_i = \frac{\rho_1 A_1 + \rho_2 A_2}{A_1 + A_2}, \tag{1.1}$$

which is the average density of fluids crossing face i. The flux density at face $i+1$ is ρ_2, and the density of the shown momentum control volume is calculated by considering the presence of a portion of the interface.

Areas A_1 and A_2 are readily available if the VOF method is used. In fact, the VOF method requires these areas to calculate mass fluxes and to advect fluid volumes (see

Bussmann *et al.* (2002) and Rudman (1998)). An equally common group of two-phase flow solvers uses the level set (LS) method (Sussman *et al.* 1994; Osher & Fedkiw 2003) for modeling interface dynamics. However, in the LS method, mass fluxes are not calculated explicitly, hence areas A_1 and A_2 are not readily available.

The objective of this work is then to develop a numerical method for accurate simulation of two-phase flows with large density ratio in the context of the LS method. This new method requires two steps: (1) devising an LS-based technique for calculating flux densities, and (2) accurate calculation of flux velocities near fluid interfaces. This article discusses the first step. Although our focus is on the flow solvers with staggered arrangement of variables, most of the techniques discussed here are also applicable to collocated arrangement of variables. We first review the governing equations of a two-phase flow and briefly present the numerical methods. The new technique is then introduced, followed by results.

2. Governing equations

Consider a two-phase flow of immiscible, incompressible and Newtonian fluids. The governing equations for the flow are conservation of mass and momentum

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = \frac{\partial \rho}{\partial t} + \boldsymbol{U} \cdot \nabla \rho = 0 \tag{2.1}$$

$$\frac{\partial (\rho \boldsymbol{U})}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}\boldsymbol{U}) = -\nabla p + \nabla \cdot \tau + \boldsymbol{F}_B, \tag{2.2}$$

where $\boldsymbol{U}$ denotes velocity, p pressure, $\boldsymbol{F}_B$ any body forces such as gravity, and τ the shear stress tensor $\tau = \mu(\nabla \boldsymbol{U} + \nabla \boldsymbol{U}^T)$.

Across the interface separating two fluids, the following jump conditions exist. The normal component of velocities in both fluids is equal (provided that there is no mass interchange between two different phases of a single substance (Bird *et al.* 2002)). That is

$$[\boldsymbol{U} \cdot \hat{n}] \equiv \boldsymbol{U}^{(1)} \cdot \hat{n} - \boldsymbol{U}^{(2)} \cdot \hat{n} = 0, \tag{2.3}$$

where superscripts (1) and (2) denote fluids 1 and 2, and $\hat{n}$ is the unit normal vector to the interface. If viscous effects are present, the tangential component of velocities is also equal, and therefore

$$[\boldsymbol{U}] = 0. \tag{2.4}$$

Finally, we have the following jump condition for stress (Landau & Lifshitz 1987):

$$[(-pI + \tau) \cdot \hat{n}] = \sigma \kappa \hat{n}, \tag{2.5}$$

where σ is the surface tension coefficient (assumed constant) and κ is interface curvature.

In addition to Eqs. (2.1) and (2.2) for flow, an evolution equation for fluid interfaces is also solved. We use the level set method and solve the following advection equation for a scalar indicator function $G = G(\boldsymbol{x}, t)$ to evolve the interface on a fixed numerical mesh:

$$\frac{\partial G}{\partial t} + \boldsymbol{U} \cdot \nabla G = 0. \tag{2.6}$$

The interface is represented by an isosurface G_0. Note that G can be any smooth function, but the most common one is a signed distance function, for which $G_0 = 0$.

A single set of Eqs. (2.1) and (2.2) is solved in both fluids, where the fluid properties at any point are calculated from G. Properties are constant in each fluid.

3. Numerical method

3.1. *Flow solver*

The numerical methods for solving the governing equations are described extensively in Desjardins *et al.* (2008*a*,*b*), and are only mentioned briefly here. The model is a 3-D structured, finite-difference flow solver with staggered arrangement of variables. It is a parallel model designed for direct numerical simulation and large-eddy simulation of turbulent reactive flows. Although the order of accuracy of spatial discretizations can be arbitrary for single-phase flows, the multi-phase flow model is second-order accurate in space and time.

The jump condition due to surface tension in Eq. (2.5) is incorporated by the Ghost-Fluid method (GFM) (Kang *et al.* 2000; Desjardins *et al.* 2008*b*). Interfacial quantities $\hat{n}$ and κ are computed from G by

$$\hat{n} = \frac{\nabla G}{|\nabla G|} \tag{3.1}$$

and

$$\kappa = -\nabla \cdot \hat{n}. \tag{3.2}$$

Unlike Desjardins *et al.* (2008*b*), we solve Eq. (2.2) in the conservative form when advecting momentum (i.e., we solve $\partial(\rho \boldsymbol{U})/\partial t = -\nabla \cdot (\rho \boldsymbol{U}\boldsymbol{U})$ in the momentum advection step). Once the momentum is advected, we return to the non-conservative form by dividing Eq. (2.2) by ρ and follow the methods described in Desjardins *et al.* (2008*b*).

To obtain $\boldsymbol{U}$ from $\rho\boldsymbol{U}$, however, we need density of momentum control volumes; we use the LS function at time step n to compute volume fraction of momentum control volumes, which is denoted by F^n. So, the density of a momentum control volume m at time step n is

$$\rho_m^n = F_m^n(\rho_1 - \rho_2) + \rho_2. \tag{3.3}$$

To obtain ρ_m^{n+1}, we do not use the interface location at $n+1$. Instead, we solve the following mass conservation equation for momentum control volumes with ρ_m^n as the initial condition:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{U}) = 0, \tag{3.4}$$

where we use the same flux densities as those used to advect momentum (explained in the following subsection). This gives a tight coupling between mass and momentum advections. The technique for calculating flux densities is presented below.

3.2. *Calculating flux densities*

We introduce a new technique for calculating flux densities in the level set context. Our technique, which is based on the signed distance function ϕ, is different than the approach introduced by Hu *et al.* (2006). Unlike Hu *et al.* (2006), we solve a single set of flow equations for both fluids, hence the new technique is tailored for such flow solvers.

The new technique is explained by the following 1-D example: consider an interface at time step n, denoted by Γ^n, as shown in Fig. 3. The distance function for node i at time n is shown as ϕ_i^n. As the interface moves to its new location Γ^{n+1}, the flux across the face at i (shown by dashed line) consists of some amount of fluid 2 and more of fluid 1. To obtain the flux density, we assume an imaginary cell (shown at the bottom of Fig. 3) whose boundaries are located at Γ^n and Γ^{n+1}, as well as an imaginary interface located at face i. We then swap the fluid locations in the imaginary cell (i.e., fluid 1 is now to

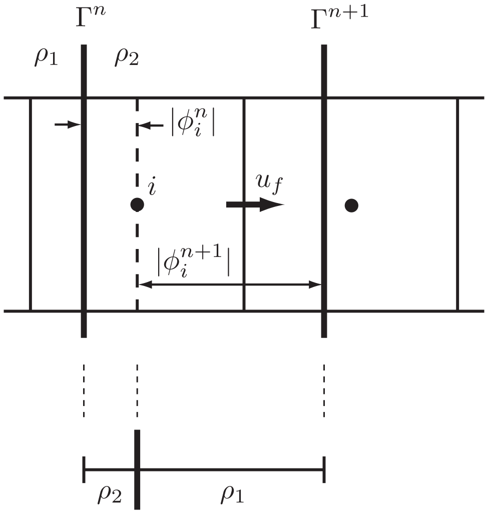

FIGURE 3. An interface Γ in 1-D at time steps n and $n+1$.

the right of the interface). The flux density across face i from time step n to $n+1$ is exactly equal to the density of the imaginary cell.

Assuming $\phi > 0$ in fluid 1, we can then establish the following formulations in 1-D for flux density across any face (not just faces near fluid interfaces):
If $\phi_i^n \phi_i^{n+1} > 0$ or $\phi_i^n = 0$ on the face, the flux density is

$$\tilde{\rho} = \begin{cases} \rho_1 \; ; & \phi_i^{n+1} \geq 0 \\ \\ \rho_2 \; ; & \text{otherwise} \end{cases} . \tag{3.5}$$

Else,

$$\tilde{\rho} = \begin{cases} \frac{|\phi_i^n|\rho_2 + |\phi_i^{n+1}|\rho_1}{|\phi_i^n| + |\phi_i^{n+1}|} \; ; & \phi_i^n < 0 \\ \\ \frac{|\phi_i^n|\rho_1 + |\phi_i^{n+1}|\rho_2}{|\phi_i^n| + |\phi_i^{n+1}|} \; ; & \text{otherwise} \end{cases} . \tag{3.6}$$

To extend these formulations to 2-D, we consider an interface at time steps n and $n+1$ in a numerical cell, as shown in Fig. 4. The total flux density across the face AD (see Fig. 4) is calculated by splitting AD into three segments: AB, BC and CD, which are found from the intersections between AD and Γ^n or Γ^{n+1}. Then, the above 1-D calculation is performed for the midpoint of each line segment, denoted by M_1, M_2 and M_3. The total flux density across face AD is then

$$\tilde{\rho}_{AD} = \frac{\tilde{\rho}_{M_1}\overline{AB} + \tilde{\rho}_{M_2}\overline{BC} + \tilde{\rho}_{M_3}\overline{CD}}{\overline{AD}}. \tag{3.7}$$

The intersection of an interface and cell boundaries are found by the marching cube algorithm. It is easy to note that if an interface moves normal to a face, the above method yields a time-averaged density along that face, which is, in fact, a very good estimation.

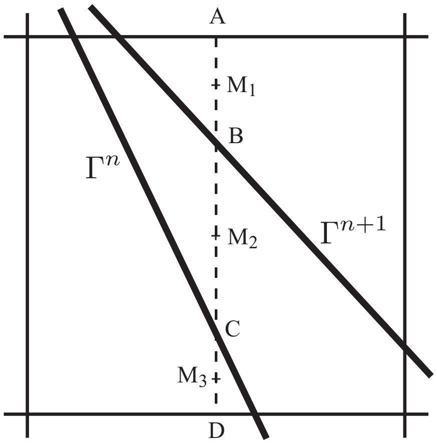

FIGURE 4. An interface Γ at time steps n and $n+1$ in a 2-D numerical cell.

4. Results

4.1. *Translation of a circle in a prescribed velocity field*

To assess the performance of the new technique for calculating flux densities, we first consider the translation of a circle of radius 0.2, positioned initially at the center of a 1×1 domain. Two uniform velocity fields are prescribed: $\boldsymbol{U} = (1, 0)$ and $(0, 1)$. The domain is discretized into N number of uniform grids in each direction, where $N \in \{10, 20, 40, 80, 160, 320\}$. For each grid size, we use three time steps, such that the CFL number is 0.25, 0.5 and 0.75. To calculate flux densities, we choose the density to be 1000 inside the circle, and 1 outside (note that the flow equations are not solved here). Computed flux densities are compared to those obtained from a VOF method (Youngs' method (Youngs 1982)). For the VOF results, we initialize the VOF field (volume fractions) using recursive local mesh refinement. We refine to 16 levels in interfacial cells, which yields the volume fractions to machine precision. Further subdivisions do not change the values. In the VOF results, we use the exact normal vectors for the piecewise-linear interface reconstruction. Since we run the test for only one time step, the VOF advection errors will have no effect on the results, and the calculated flux densities will converge to exact values with mesh refinement.

We calculate L_1, L_2, and L_∞ norms of the normalized error of $\tilde{\rho}$, defined as

$$Err. = \frac{|\tilde{\rho}^{\mathrm{VOF}} - \tilde{\rho}^{\mathrm{LS}}|}{\tilde{\rho}^{\mathrm{VOF}}}, \tag{4.1}$$

where the superscripts VOF and LS denote flux densities obtained from the VOF method, and the new LS-based technique, respectively. We present errors for flux densities of u and v momentum control volumes. As shown in Fig. 5, for a u momentum control volume, flux density across vertical and horizontal faces are denoted by $\tilde{\rho}_{f_{xx}}$ and $\tilde{\rho}_{f_{xy}}$, respectively; $\tilde{\rho}_{f_{yy}}$ and $\tilde{\rho}_{f_{yx}}$ are flux densities across horizontal and vertical faces of a v momentum control volume, respectively.

Figure 6 shows the errors of $\tilde{\rho}_{f_{xx}}$ for $\boldsymbol{U} = (1, 0)$ at various grid sizes and CFL numbers. L_1 (dashed line), L_2 (dotted line) norms exhibit second-order convergence, while the convergence of L_∞ (dashed-dotted line) norm is almost first-order. The same test was performed for $\tilde{\rho}_{f_{yy}}$ with $\boldsymbol{U} = (0, 1)$, and expectedly, the results (not shown) are exactly the same as those presented in Fig. 6. As can be seen, the time step size has almost no

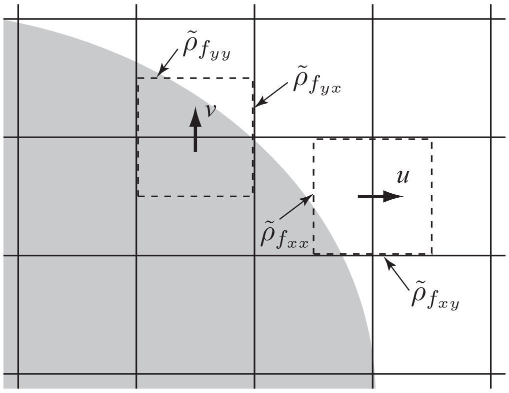

FIGURE 5. Flux densities across faces of momentum control volumes.

effect on the convergence rate of errors. Next, the errors of $\tilde{\rho}_{f_{xy}}$ are presented in Fig. 7 for $\boldsymbol{U} = (0, 1)$. They exhibit the same order of accuracy. The errors associated with $\tilde{\rho}_{f_{yx}}$ for $\boldsymbol{U} = (1, 0)$ are also exactly the same as $\tilde{\rho}_{f_{xy}}$ errors.

4.2. *Collapse of a water column*

Finally, we test the performance of the new algorithm in a flow solver. Consider a 2-D water (fluid 1) column in air (fluid 2), shown in Fig. 8, which corresponds to the experimental results of Martin & Moyce (1952). $\rho_1 = 1000$ (kg/m^3), $\rho_2 = 1.226$ (kg/m^3), $\mu_1 = 1.137 \times 10^{-3}$ (kg/ms), $\mu_2 = 1.78 \times 10^{-5}$ (kg/ms), $\sigma = 0.0728$ (N/m), $g = -9.81$ (m/s^2). The initial height and width of the column are both 5.715 cm. The domain size is 40×10 cm, and is discretized by 256×64 uniform grid points. To simulate the collapse of the water column, we use the new method, in which the flow equations are solved in the conservative form, as well as another model that solves the non-conservative form of the flow equations. Note that the only difference in these models lies in the treatment of the convective term in Eq. 2.2.

Figure 8 shows the results of the new (conservative) method. Qualitatively, they agree well with the experimental results of Martin & Moyce (1952) and with the numerical results of Bussmann *et al.* (2002) (a quantitative comparison needs to be done). The spreading of the water column is not affected by the air, as expected. In another simulation, we increased the water density ten-fold ($\rho_1/\rho_2 = 8150$) and ran a similar case. The results (not shown) are nearly identical to those shown in Fig. 8.

Next, the results of the non-conservative model are depicted in Fig. 9, where, compared to Fig. 8, a significant difference is seen: in Fig. 9, the collapsing water column spreads less, and compared to the experimental results, it assumes an unphysical topology. This is due to an incorrect momentum transfer near the interface. Results at higher grid resolutions show the same unphysical behavior. This test clearly illustrates the importance of solving the conservative form of the flow equations and the efficacy of the new method.

5. Future work

The new technique for calculating flux densities will be extended to 3-D. Better methods for computing flux velocities near fluid interfaces will also be sought (we currently use a first-order upwinding method near the interface). Similar to Herrmann *et al.* (2008), velocity jump conditions (Eqs. (2.3) and (2.4)) are proposed to be employed to calculate flux velocities near fluid interfaces.

Acknowledgments

The author is sincerely grateful to Profs. Heinz Pitsch and Olivier Desjardins for fruitful discussions, and Mr. Shashank and Dr. Guillaume Blanquart for their help on the NGA code. The insightful comments of Dr. Eric Johnsen are greatly appreciated.

REFERENCES

Bird, R. B., Stewart, W. E. & Lightfoot, E. N. 2002 *Transport Phenomena*, (2nd ed.). New York: John Wiley and Sons.

Bussmann, M., Kothe, D. B. & Sicilian, J. M. 2002 Modeling high density ratio incompressible interfacial flows. In *Proceedings of ASME 2002 Fluids Engineering Division Summer Meeting*. Montreal, Canada.

Desjardins, O., Blanquart, G., Balarac, G. & Pitsch, H. 2008*a* High order conservative finite difference scheme for variable density low Mach number turbulent flows. *J. Comput. Phys.* **227** (15), 7125–7159.

Desjardins, O., Moureau, V. & Pitsch, H. 2008*b* An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *J. Comput. Phys.* **227** (18), 8395–8416.

Herrmann, M., Moin, P. & Abarzhi, S. 2008 Nonlinear evolution of the Richtmyer–Meshkov instability. *J. Fluid Mech.* **612**, 311–338.

Hu, X. Y., Khoo, B. C., Adams, N. A. & Huang, F. L. 2006 A conservative interface method for compressible flows. *J. Comput. Phys.* **219** (2), 553–578.

Kang, M., Fedkiw, R. & Liu, X.-D. 2000 A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput.* **15** (3), 323–360.

Landau, L. D. & Lifshitz, E. M. 1987 *Fluid Mechanics*, (2nd ed.). Oxford: Pergamon Press.

Martin, J. & Moyce, W. 1952 An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos. Trans. R. Soc. London, Ser. A* **244**, 312–324.

Osher, S. & Fedkiw, R. 2003 *Level Set Methods and Dynamic Implicit Surfaces*. New York: Springer-Verlag.

Rudman, M. 1998 A volume-tracking method for incompressible multifluid flows with large density variations. *Int. J. Numer. Meth. Fluids* **28**, 357–378.

Sussman, M., Smereka, P. & Osher, S. 1994 A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114** (1), 146–159.

Youngs, D. L. 1982 Time-dependent multi-material flow with large fluid distortion. In *Numerical Methods for Fluid Dynamics* (ed. K. W. Morton & M. J. Baines), pp. 273–285. New York: Academic Press.

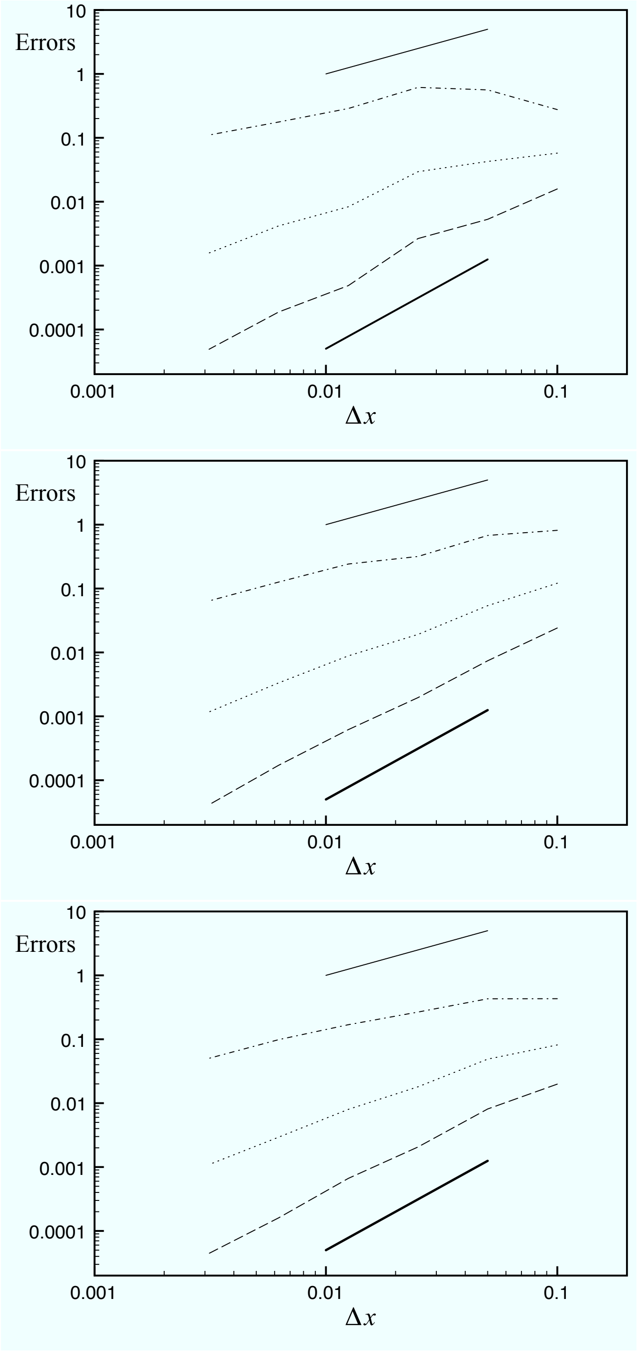

FIGURE 6. L_1 (dashed line), L_2 (dotted line) and L_∞ (dashed-dotted line) norms of normalized errors of $\tilde{\rho}_{f_{xx}}$ for CFL numbers 0.25, 0.5 and 0.75 (top to bottom figures). Thin and thick lines represent first- and second-order accuracy, respectively.

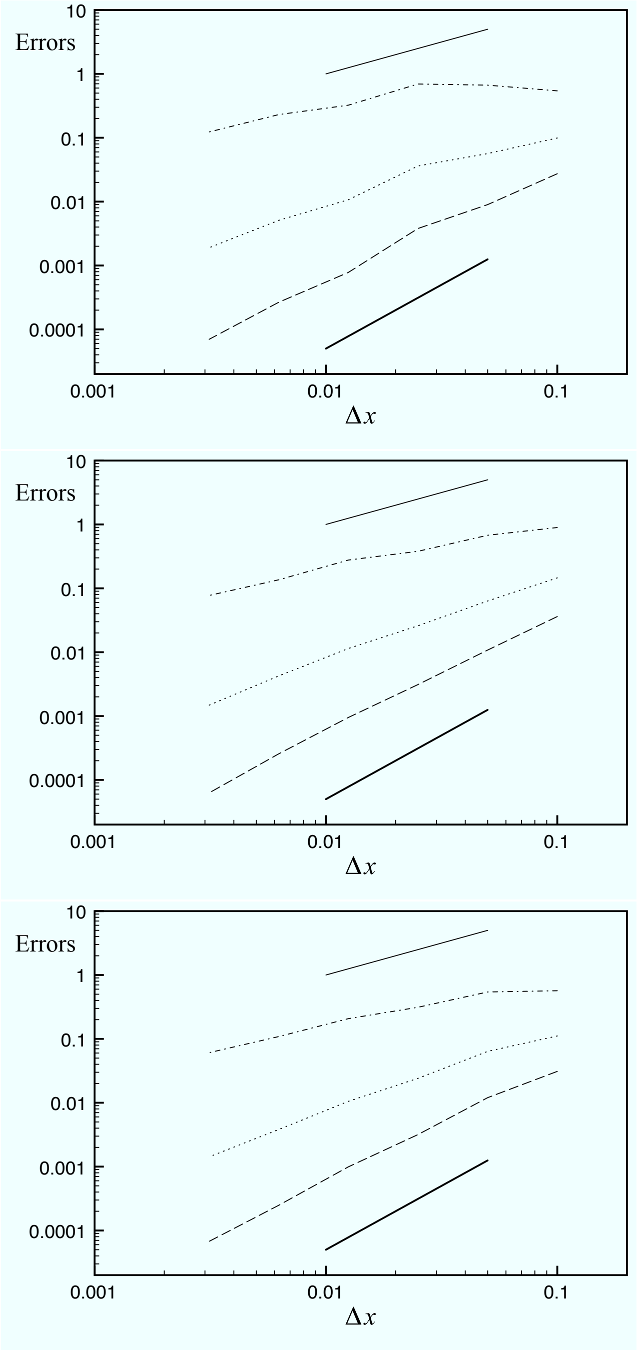

FIGURE 7. L_1 (dashed line), L_2 (dotted line) and L_∞ (dashed-dotted line) norms of normalized errors of $\tilde{\rho}_{f_{xy}}$ for CFL numbers 0.25, 0.5 and 0.75 (top to bottom figures). Thin and thick lines represent first- and second-order accuracy, respectively.

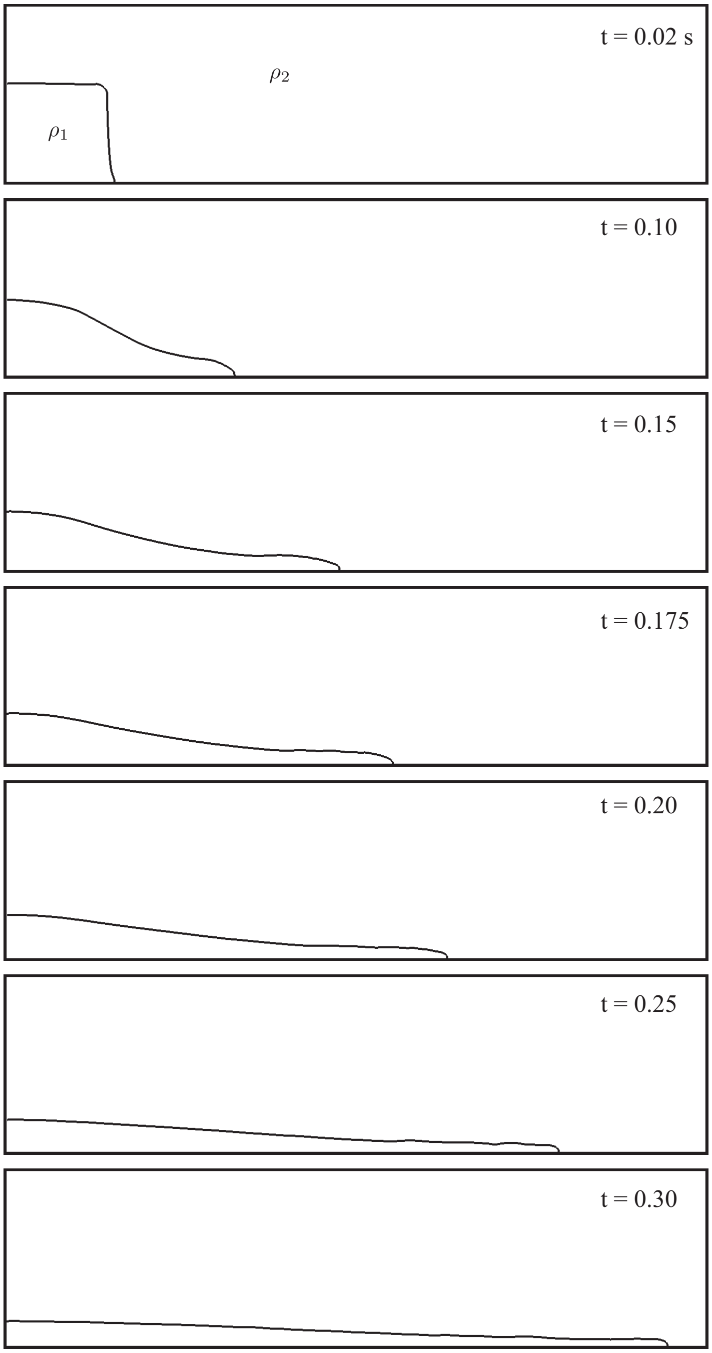

FIGURE 8. Numerical results for the collapse of a water column in air ($\rho_1/\rho_2 = 815$), obtained from the new conservative method. A similar simulation at $\rho_1/\rho_2 = 8150$ generated nearly identical results.

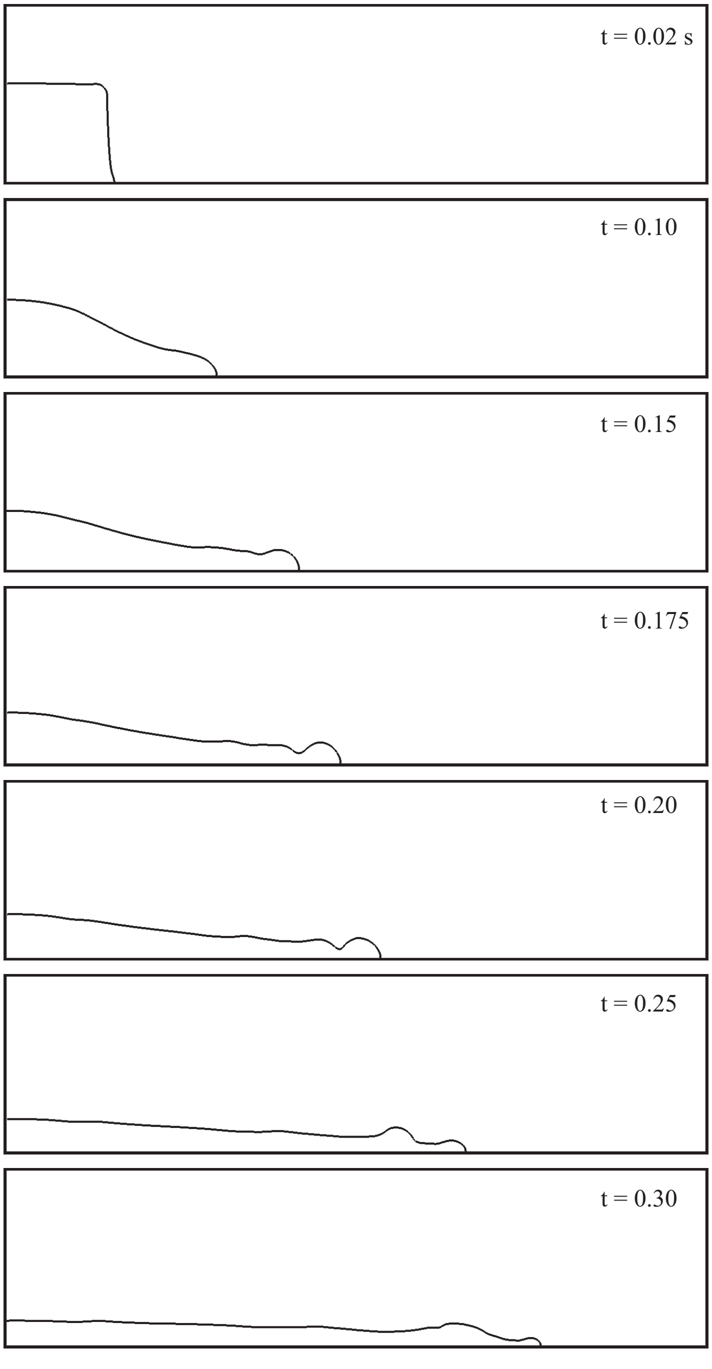

FIGURE 9. Numerical results for the collapse of a water column in air ($\rho_1/\rho_2 = 815$), obtained from a non-conservative model.