Journal of Computational Physics **169,** 652–677 (2001)
doi:10.1006/jcph.2000.6590, available online at http://www.idealibrary.com on IDEAL®

A Critical Analysis of Rayleigh–Taylor Growth Rates

J. Glimm,*,†,[1,2] J. W. Grove,‡,[3] X. L. Li,*,[1] W. Oh,*,[1] and D. H. Sharp§,[4]

*Department of Applied Mathematics and Statistics, University at Stony Brook, Stony Brook, New York 11794-3600; †Center for Data Intensive Computing, Brookhaven National Laboratory, Upton, New York 11793-6000; ‡Hydrodynamics Methods Group, Applied Physics Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545; and §Complex Systems Group, Theoretical Division, Los Alamos National Laboratory, Los Alamos, New Mexico 87545
E-mail: glimm@ams.sunysb.edu, jgrove@lanl.gov, linli@ams.sunysb.edu, woh@ams.sunysb.edu, dhs@t13.lanl.gov

Received December 16, 1999; revised July 5, 2000

Recent simulations of Rayleigh–Taylor instability growth rates display considerable spread. We provide evidence that differences in numerical dissipation effects (mass diffusion and viscosity) due to algorithmic differences and differences in simulation duration are the dominant factors that produce such different results. Within the simulation size and durations explored here, we provide evidence that the principal discrepancies are due to numerical dispersion through comparison of simulations using different algorithms. We present new 3D front tracking simulations that show tentative agreement with the range of reported experimental values. We begin an exploration of new physical length scales that may characterize a transition to a new Rayleigh–Taylor mixing regime. © 2001 Academic Press

1. INTRODUCTION

1.1. Purpose and Scope of the Paper

Accurate numerical simulation of multiphase fluid mixing rates is a long-standing challenge for computational fluid dynamics. Only recently has the available hardware allowed

[1] Supported in part by the Applied Mathematics Subprogram of the U.S. Department of Energy DE-FG02-90ER25084, DE-FG02-98ER2536, DE-FG03-98DP00206.

[2] Supported in part by the Department of Energy Office of Inertial Fusion, the Army Research Office, Grant DAAG559810313, the National Science Foundation, Grant DMS-9732876, and Los Alamos National Laboratory.

[3] Supported by the U.S. Department of Energy.

[4] Supported by the U.S. Department of Energy under Contract W-7405-ENG-36.

0021-9991/01 $35.00

significant three-dimensional studies. We consider here one of the most important of this class of problems, Rayleigh–Taylor instability. Rayleigh–Taylor instability results when a randomly perturbed density contrast interface is subject to continuous acceleration. A basic characteristic of Rayleigh–Taylor instability is the constant that describes the acceleration of the mixing zone edge.

A wide range of values for this acceleration constant have been reported on the basis of simulation studies, some of which fall outside the limits of experimental error. The purpose of this paper is to begin a systematic analysis of causes of these discrepancies. To do this, we summarize the results of previous Rayleigh–Taylor instability studies, identify potential sensitive factors in Rayleigh–Taylor simulations, and report new simulation results designed to quantify the effects of a number of these factors.

There are three main results reported in this paper. The first result shows that front tracking simulations using the *FronTier* code are in tentative agreement with experimental results. Documentation for this code is provided in

`http://www.ams.sunysb.edu/∼shock/FTdoc.FTmain.html.`

To achieve this agreement it is necessary to correct for finite compressibility effects and to compensate for different conventions in the definition of the growth rate.

The second main result is identifying a possible cause for the spread in simulation results. We compare distinct algorithms, paying special attention to dissipative effects. Over the simulation time and size considered here, we can duplicate the observed spread in simulation growth rates through comparison of capturing to tracking algorithms. After restriction to typical high and low values of the growth rate, there is approximately a factor of 2 to be explained. The low values reported for the growth rate are time dependent, and about half of the factor of 2 difference in simulations occurs during the simulation times reported in this paper. For simulations with identical gridding, simulation time, and other numerical parameters, we see a 40% decrease in the growth rate computed using capturing algorithms (which have artificial dissipation) compared to those obtained with *FronTier* (which completely eliminates dissipation for interfacial vorticity and for density discontinuities). Thus, essentially all of the discrepancy for the times studied here can be attributed to interfacial dissipative mechanisms in capturing algorithms. Moreover, we can tentatively identify viscosity rather than mass diffusion as the dominant cause through comparison of two capturing codes, one of which is designed to control mass diffusion while the other is not.

The third main result of this paper is an initial exploration of a possible new physics regime for Rayleigh–Taylor mixing through identification of a new length scale that is independent of the mixing zone width.

All studies in this paper need to be taken to later time. It is known that the discrepancies increase strongly with time. Moreover, the uncertainties associated with data analysis decrease with time, because this analysis is based on the assumption that the flow is in a self-similar (late time) regime.

1.2. Background Discussion of Rayleigh–Taylor Instability

An interface between fluids of different densities is unstable when subjected to an acceleration directed from the heavy fluid to the light fluid [35, 41, 44]. This instability, known as Rayleigh–Taylor instability, has been a challenge to computational fluid dynamics since the early days of computers [4]. The instability has a fingering nature, with bubbles of light fluid rising into the ambient heavy fluid and spikes of heavy fluid falling into the light fluid.

With $\rho_1 < \rho_2$ representing the light and heavy fluid densities, and the Atwood number $A = (\rho_2 - \rho_1)/(\rho_2 + \rho_1)$ a buoyancy renormalization to gravity g, the outer edges of the mixing zone $Z_k(t)$ are observed to obey the large time asymptotic scaling law

$$Z_k(t) = (-1)^k \alpha_k A g t^2, \tag{1}$$

where α_k is a constant. Here, to be consistent with the conventions of laboratory experiments, the acceleration (gravity) is directed along the negative z axis, so that bubbles "fall" downward, spikes rise, and $Z_1 < 0$.

Rayleigh–Taylor instability arises in a variety of applications, ranging from incompressible regimes such as wind shears in thunder shower systems to highly compressible flows as occur in inertial confinement fusion and in supernovas. For this reason, the use of two fluid Euler equations to model the fluid flow is appropriate. Laboratory experiments are nearly incompressible. There are four principal numerical difficulties:

1. The sharp interface between the distinct fluids is difficult to maintain for most Eulerian algorithms.
2. The geometric complexity of the late time unstable interface between the fluids is a source of difficulty for most Lagrangian algorithms.
3. The requirement for a fully compressible code which can be validated on nearly incompressible experiments imposes a strain on computational resources and algorithms.
4. The spatial complexity and late time simulations required to observe a well-developed self-similar flow regime pose a challenge in terms of simulation resources and algorithmic efficiency.

1.3. Summary of Rayleigh–Taylor Results

The bubble acceleration constant $\alpha_b \equiv \alpha_1$ provides the most basic characterization of the mixing zone. However, as Table I illustrates, simulations show considerable spread in

TABLE I

Determination of α_b by Experiment, Theory, and Simulation

Theory: Bubble Merger Models			
Sharp/Wheeler [42], Sharp [41]	1961	$\alpha_b \sim 0.01$–0.025	3D
Glimm/Sharp [21], Zhang [51]	1990	$\alpha_b \sim 0.06$	2D
Alon *et al.* [1]	1994	$\alpha_b \sim 0.05$	2D
Glimm/Sharp [22]	1997	$\alpha_b \sim 0.045$	2D
Cheng *et al.* [7]	1999	$\alpha_b \sim 0.06$	3D
Experiments			
Read/Youngs [36, 48]	1984	$\alpha_b \sim 0.58$–0.65	2D
		$\alpha_b \sim 0.063$–0.077	3D
Kucherenko *et al.* [29]	1991	$\alpha_b \sim 0.07$	3D
Snider/Andrews [43]	1994	$\alpha_b \sim 0.07 \pm 0.007$	3D
Schneider *et al.* [39]	1998	$\alpha_b \geq 0.054$	3D
Dimonte/Schneider [11]	1999	$\alpha_b \sim 0.05 \pm 0.01$	3D
Simulation			
Youngs [49]	1991	$\alpha_b \sim 0.04$–0.05	3D
Youngs [50]	1999	$\alpha_b \sim 0.03$	3D
S.-Y. Chen [5]	1999	$\alpha_b \sim 0.043$	3D
Cheng *et al.* [7]	1999	$\alpha_b \sim 0.08$	3D
Glimm *et al.* (this work)	1999	$\alpha_b \sim 0.07$	3D

reported values for α_b. Results from four independent series of experiments show a spread (including error bars) of nearly a factor of 2. Somewhat over half of this spread is due to systematic differences among the four series of experiments; the error bars reported for each single experiment account for the remainder. Theoretical results are generally consistent with the experiments. Most of the simulations give growth rates which lie within a factor of 2.5 of one another. The spread in simulation values has widened as increased computational resources have allowed exploration of larger spatial ensembles of random modes, carried to later times. Plainly, there is a need for an analysis of the simulation results and their algorithmic basis which can explain the spread in simulation values. There is also a need for simulation or other studies of the experiments to explain the spread in experimental values. The present paper is a beginning of such a study, with a focus on algorithmic issues.

1.4. Outline of Paper

In Section 2 we discuss the algorithmic and simulation factors that may influence the simulation values of α_b. The dissipative mechanisms of mass diffusion and viscosity (vorticity diffusion) are known to be important sources of numerical errors for flows with material interfaces. To assess and differentiate between the two dissipative mechanisms of mass and vorticity diffusion, we employ an artificial compression algorithm. This algorithm reduces the number of cells over which the mass diffuses, but it does not limit vorticity diffusion. In Section 3, we review the front tracking algorithm used in later sections of this paper. This algorithm is free from interfacial dispersion. Section 4 presents and analyzes new simulation studies. The purpose of the simulations and their analysis is to shed light on the role of potentially sensitive factors discussed in Section 2 and to show that *FronTier* simulations do in fact agree with the experiment. Conclusions are stated in Section 5.

2. PRINCIPAL FACTORS AFFECTING SIMULATION RESULTS

2.1. The Numerical Analysis of Mass and Vorticity Diffusion

This paper presents evidence suggesting that numerical dissipation, primarily mass diffusion and viscosity, is the dominant error contributing to the discrepancy between simulations and experiments. Density contrasts and vorticity are concentrated along the interface, and so this is where capturing schemes concentrate their errors. Numerical dissipation is exacerbated in the Rayleigh–Taylor instability problem by the long time of the simulation, by the dynamically growing interface length along which the numerical dissipation occurs, and by the dominant role that the density contrasts and vorticity concentrations along the interface play in the growth of the mixing zone.

2.1.1. *Physical Values of Mass Diffusivity*

The influence of diffusion on small amplitude Rayleigh–Taylor exponential growth rates was analyzed in [13] using an argon–helium mixture with a diffusion constant $D = 0.64$ cm^2/s. A reduction of the small amplitude exponential growth rate of about 20% was observed due to physical diffusivity. For comparison, fluid diffusivities are much smaller, on the order of 10^{-4} cm^2/s, and should not influence experimental Rayleigh–Taylor growth rate values.

2.1.2. *Estimates of Numerical Diffusivity*

Numerical diffusion and dispersion are known to be serious issues for Eulerian finite difference algorithms. In the case of Rayleigh–Taylor instability, where the instability is driven by density contrasts, these effects can be expected to be significant. Some insight into these effects can be gained by an investigation of the modified partial differential equations [23, 28, 31, 37, 47] for finite difference schemes for solving the linear advection equation

$$u_t + vu_x = 0. \tag{2}$$

This equation can be regarded as a model of the contact discontinuity mode for gas dynamics. For simplicity we assume $v > 0$ and consider schemes using flux limiting between the Lax–Wendroff method and upwind differencing. Following the notation of LeVeque [31], these schemes are of the form

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{\Delta x}\left(f_{j+1/2}^n - f_{j-1/2}^n\right), \tag{3}$$

where

$$f_{j+1/2}^n = vu_j + \frac{1}{2}v(1-\nu)\phi_j(u_{j+1} - u_j), \tag{4}$$

$\nu = v\Delta t/\Delta x$, and the limiter ϕ_j is given by

$$\phi_j = \phi(\theta_j) \quad \text{and} \quad \theta_j = \frac{u_j - u_{j-1}}{u_{j+1} - u_j}.$$

Such schemes are second-order accurate provided $\phi(1) = 1$. We also observe that the CFL condition requires that $\nu < 1$.

For the unlimited case $\phi(\theta) \equiv 1$, where this scheme reduces to the Lax–Wendroff method, Richtmyer and Morton [37] state that solutions to the modified partial differential equation

$$u_t + vu_x = -\frac{1}{6}v\Delta x^2(1-\nu^2)u_{xxx} - \frac{1}{8}v\Delta x^3\nu(1-\nu^2)u_{xxxx} \tag{5}$$

are fourth-order solutions to the finite difference scheme (3). More generally, for schemes with smooth $\phi(\theta)$, a straightforward but tedious computation yields a modified equation of the form

$$u_t + vu_x = \Delta x c_1(x,t)u_{xx} + \Delta x^2 c_2(x,t)u_{xxx} + \Delta x^3 c_3(x,t)u_{xxxx}, \tag{6}$$

where

$$\begin{aligned}
c_1(x,t) &= \frac{1}{2}v(1-\nu)(1-\phi(1))\\
c_2(x,t) &= -\frac{1}{6}v(1-\nu)(1+\nu(3\phi(1)-2)-3\phi'(1))\\
c_3(x,t) &= v(1-\nu)\left\{\frac{1}{4}\phi''(1)\frac{u_{xx}^2 - 2u_x u_{xxx}}{u_x^2}u_{xx}\right.\\
&\qquad \left. - \left(\frac{\nu+\nu^2+2(1-2\nu)\phi'(1)}{8} + \chi\frac{(1+6\nu^2)}{24} + \chi^2\frac{\nu(1-\nu)}{8}\right)u_{xxxx}\right\}
\end{aligned}$$

and $\chi = \phi(1) - 1$.

For the first-order upwind method, where $\phi(\theta) \equiv 0$, the modified equation has a leading order diffusion coefficient equal to

$$D = \frac{1}{2} v(1 - \nu)\Delta x. \tag{7}$$

As a model for gas dynamics the CFL condition requires that $(|v| + c)\Delta t/\Delta x < 1$. Here c is the sound speed. For the low compressibility flows of concern here, we can estimate a typical flow velocity as $|v| \approx 0.1c$ so that ν is on the order of a tenth or less. Translating grid units into physical units, with a 1-mm zoning and a time scale of seconds, we find a numerical diffusion on the order of 150 cm^2/s. One physical interpretation of this quantity is the viscous diffusion of velocity fluctuations (vorticity) in a fluid with mean velocity v, as measured by the kinematic viscosity. For materials such as air and water, the physical values of the viscosity are well known and are available in such handbooks as the *CRC Handbook of Chemistry and Physics*. Comparing the ratios of the numerical to physical viscosities, we see that the numerical viscosity of air is approximately three orders of magnitude greater than the physical viscosity while the numerical viscosity of water is approximately four orders of magnitude greater than its physical viscosity. Another interpretation is the physical diffusion of one material into another. Again referring to the CRC Handbook, we see that typical values for diffusivities of gases into liquids and various solutes into water are on the order of 10^{-5}–10^{-4} cm^2/s while the diffusivities of various gases into air are on the order of 10^{-1}–1 cm^2/s. Again comparing these coefficients to the numerical diffusivities above, we see that the numerical diffusion coefficients are approximately six orders of magnitude greater than the physical diffusivity for solutes in water and from two to four orders of magnitude greater for the diffusion of gases into air. Reduction of D to its physical value for any of the above models would require refining the zoning by up to a factor of 10^6 in the worst case with a corresponding increase in computational effort of $10^{(d+1)f}$ for a refinement factor of 10^f in $d = 1, 2$, or 3 space dimensions, a route plainly not feasible for any but the simplest of the above cases.

The estimates described in the previous paragraph are in fact overly pessimistic. Second-order methods, where $\phi(1) = 1$, eliminate the first-order diffusion terms in the finite difference equations, making the above order of magnitude estimates inappropriate. For such methods, in regions bounded away from spatial extrema in $u(x, t)$ and for Δx sufficiently small, the dominant term in Eq. (6) is the linear dispersion term

$$-\frac{1}{6} v\Delta x^2(1 - \nu)(1 + \nu - 3\phi'(1))u_{xxx}. \tag{8}$$

While these methods are all formally second-order accurate, for finite meshes the limiter will reduce to a locally first-order method in regions of strong flow gradients such as at jumps or at corners. However, near such regions, numerical diffusion will smooth out the steep gradients and hence decrease the influence of the limiter. It is important to note that the effect of the dispersion term on the discrete solution to the second-order method is qualitatively different from the effect of diffusion on the first-order method. The former leads to the dispersion of oscillations without damping their amplitudes, while the latter reduces the amplitude of the oscillations as they diffuse. The diffusion in the numerical solution arises from the fourth-order terms in (6). It is interesting to note that for a nontrivial limiter these diffusion terms are nonlinear.

For finite Δx, near jumps or corners, the finite difference solution behaves as a first-order equation with a $t^{1/2}$ rate of spreading. Once the discrete Laplacian $u_{j-1} - 2u_j + u_{j+1}$ is $\mathcal{O}(\Delta x^2)$, the higher order analysis of the limiters is applicable and the subsequent dispersion scales as $t^{1/3}$. This scaling is observed in numerical experiments [32], where the width is observed to be approximately $(4/3)t^{1/3}$. If this subdiffusive dissipation is modeled by a grid-dependent diffusivity as in Section 2.1.2, then the grid-dependent diffusivity D must also be time dependent and scale as $t^{-1/6}$.

2.1.3. *Numerical Diffusion Using Artificial Compression*

We refer to [32] for a discussion of the flux limited scheme with artificial compression. In this scheme, the numerical mass diffusion is limited to about 2.5 cells, according to numerical experiments on the linear advection equation. The nonlinear fluids simulations of Section 4.2 show a larger diffusion length of about 6–7 cells.

The scheme has the same conservation form as Eq. (3), with the flux $f_{j+1/2}$ defined as

$$f_{j+1/2} = \frac{1}{2}\left(vu_j^n + vu_{j+1}^n + g_j^n + g_{j+1}^n + L_j^n + L_{j+1}^n - \left|v + \gamma_{j+1/2}^n + \lambda_{j+1/2}^n\right|\Delta u_{j+1/2}^n\right) \quad (9)$$

and $\Delta u_{j+1/2}^n = u_{j+1}^n - u_j^n$. Here the vu terms in f generate a first-order central difference scheme, the g's define a total variation diminishing (TVD) [31] anti-diffusion term, and the L terms are artificial compression anti-diffusion. The terms proportional to $\Delta u_{j+1/2}^n$ are artificial viscosity terms. The first of these, proportional to v, converts central differencing to upwind differencing. The role of γ is to control the g terms and likewise the λ terms regulate the L terms. Thus, we define

$$\gamma_{j+1/2}^n = \left(g_{j+1}^n - g_j^n\right)/\Delta u_{j+1/2}^n \quad (10)$$

$$\lambda_{j+1/2}^n = \left(L_{j+1}^n - L_j^n\right)/\Delta u_{j+1/2}^n \quad (11)$$

if $\Delta u_{j+1/2}^n \neq 0$ and $\gamma_{j+1/2}^n = \lambda_{j+1/2}^n = 0$ otherwise.

The definitions

$$g_{j+1/2}^n = \frac{1}{2}\left(|v| - \frac{\Delta t}{\Delta x}v^2\right)\Delta u_{j+1/2}^n, \quad (12)$$

$$g_j^n = M\left(g_{j-1/2}^n, g_{j+1/2}^n\right), \quad (13)$$

define a TVD scheme if $L = 0$, where

$$M = \operatorname{sign} u_1 \min\{|u_1|, \ldots, |u_r|\} \quad (14)$$

if all u_i have the same sign and $M = 0$ otherwise.

L is the artificial compression anti-diffusion. Let

$$L_{j+1/2}^n = \frac{1}{2}\left(|v| - \frac{\Delta t}{\Delta x}v^2\right)\left[\Delta u_{j+1/2}^n - M\left(\Delta u_{j-1/2}^n, \Delta u_{j+1/2}^n, \Delta u_{j+1/2}^n\right)\right] \quad (15)$$

$$L_j^n = S \cdot \max\left(0, S \cdot M\left(\eta L_{j-1/2}^n, L_{j+1/2}^n\right), S \cdot M\left(L_{j-1/2}^n, \eta L_{j+1/2}^n\right)\right). \quad (16)$$

Here $S = \operatorname{sign} L_{j+1/2}^n$ and

$$\eta = 2\left|\frac{\left|\Delta u_{j-1/2}^n\right|^\beta - \left|\Delta u_{j+1/2}^n\right|^\beta}{\left|\Delta u_{j-1/2}^n\right|^\beta + \left|\Delta u_{j+1/2}^n\right|^\beta}\right|. \quad (17)$$

Note that both $\gamma^n_{j+1/2}$ and $\lambda^n_{j+1/2}$ are $\mathcal{O}(\Delta x)$ as $\Delta x \to 0$, so that the term in absolute values in Eq. (9) has the same sign as v for sufficiently small Δx. Thus, in the absence of the artificial compression terms, Eq. (9) is simply a generalization of Eq. (4) that encompasses both cases of $v > 0$ and $v < 0$ for the special choice of the flux limiter

$$\phi(\theta) = \phi_{\text{minmod}} = \begin{cases} 0 & \theta < 0, \\ \theta & 0 < \theta < 1, \\ 1 & 1 < \theta. \end{cases} \tag{18}$$

For $\eta = 0$, $L = 0$ and the scheme is TVD [31]. For $\eta = 2.5$ the diffusion width is shown [32] numerically to be about 2.5 cells wide, independent of the time t.

2.1.4. *Transitions to New Flow Regimes*

The sensitivity of multiphase flow to change of flow regime is well known [12]. The regime of a single length scale, for which the large-scale structures are on the order of the width of the mixing zone, is known as chunk mix. Transitions to other flow regimes are characterized by the introduction of one or more new length scales to describe the probability distribution function (pdf) for the distribution of droplet and bubble sizes or fluid volume or mass fraction fluctuation length scales.

Additional fluid waves, such as shock waves, can cause shattering of large-scale structures and a change in flow regimes. They are thus a mechanism for causing a change of flow structure. Dissipation, discussed in Section 2.1.2, may cause a change of flow regime.

Continued acceleration leads to velocities growing without limit. In the presence of viscosity or compressibility, vorticity will diffuse off the interface to the interior flow or will be generated there directly, giving rise to a transition to turbulent flow. Turbulent flows have an increased effective viscosity that decreases the observed values of α_b. Turbulent flow also drives turbulent effective diffusivity, leading to a further decrease in α_b. This range of issues has been considered by Youngs *et al.* in a series of papers; see, for example, [10, 33]. Numerical emulation of turbulent diffusion through numerical mass diffusion requires time-dependent gridding. Diffusivity based upon a time increasing Reynolds number cannot otherwise match the observed $t^{-1/6}$ diffusivity for the TVD algorithm, as discussed in Section 2.1.2.

Experiments show between three and five generations of bubble merger. The lower bound comes from counting the decrease in the number of ripples or bubbles in the experimental plates of Read and Youngs. The upper bound five comes from the theory of the most unstable wavelength for these experiments. Over the time period of the experiments, the observed growth rate is very nearly linear in t^2. Thus, any transition to a new physical regime has not had an opportunity to influence α_b in this time period. Simulations do not exceed the duration over which experiments are known to have occurred. Thus, α_b is constant over the time period of the experiments and any possible transition to a new flow regime that leads to a change in the value of α_b has not occurred. For this reason, any significant time dependence for α_b or any transition to a new flow regime that causes α_b to decrease in simulations is in disagreement with the experiment.

2.2. Definition of the Statistical Ensemble

Wavelengths present in initial perturbation. The self-similar t^2 growth rate for the mixing zone thickness at late time results from the progressive merger of bubbles [22]. The bubbles individually achieve a terminal velocity due to a balance between buoyancy and drag, but as bubbles are removed from the edge of the mixing zone and neighboring bubbles expand to take their place in a merger process, the size-dependent terminal velocity can increase. It is this continued increase in length scales that allows continued acceleration.

To observe a universal value for α_b, it is desirable not to insert long wavelengths into the initial data, and thus to avoid contamination of the bubble merger mechanism with the growth at long wavelengths initially seeded in the initial data. A recommended convention is to choose wavelengths (Fourier modes) with wavelength λ satisfying $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ with $\lambda_{\min} = \lambda_{\max}/2$. Some authors include a further modulation and decrease the amplitude of the random modes near the edge of the allowed interval.

The modeling of an actual experiment, which will necessarily contain some level of unavoidable low-frequency noise, motivates simulations which do not follow the above restrictions on wavelengths λ present in the initial random interface. In [33], a simple analysis is given for the influence of long wavelength "noise" as a small perturbation of a high wavelength random surface. This work concludes that the influence is potentially significant and could increase the experimentally observed value of α_b. This analysis is based on a simple single mode computation. It would be desirable to repeat this analysis using full scale simulation.

Initial amplitude of perturbation. To avoid introduction of a new length scale into the problem, we want to choose the initial amplitude to be small, within the limit of accuracy of the small amplitude Rayleigh–Taylor theory. This small amplitude theory is then used for initialization, giving in effect a zero or infinitesimal initial amplitude. Most Eulerian finite difference schemes have trouble with initialization of small amplitude perturbations. Unless several zones are included within the initial amplitude of the perturbation, an Eulerian simulation with an untracked interface will have difficulty in observing and responding to the perturbation at all. This requirement leads to very fine scale zoning per initial wavelength or to use of large amplitude initial conditions. Front tracking, with its subgrid resolution, does not suffer from this problem. See Section 3.

Size of statistical ensemble of initial perturbations. The statistical ensemble converges to an infinite volume limit with surprising speed in two-dimensional studies [6]. This issue has not been explored in three dimensions. The size, i.e., the number of initial bubbles, is more important as a restriction on the duration of the simulation, since two or three generations of bubble merger reduce the number of bubbles by factors of 16 or 64, and the number of bubbles at the end of the simulation must still be enough for statistical significance. The requirement for two or three generations of bubble merger is to ensure that the simulation has entered the self-similar regime and to explore the influence of numerical dissipation effects which could force a transition to a new flow regime, as discussed in Section 2.1.4.

2.3. Data Analysis and Asymptotics

Equation (1) assumes a self-similar flow, assumed to be valid in the large time asymptotic limit. There are two problems with using this picture and formula in the analysis of simulation. First, all simulations, including those reported here, are of limited time duration. Second, the experimental initial conditions are not observed. Neither the statistical ensemble nor even its variance at an initial time is known. We propose above to initialize, within the regime of linear analysis, amplitude $\ll$ wavelength. Any such statistical ensemble of initial conditions can be propagated backward in time by the linearized Rayleigh–Taylor equations to another such ensemble with a still smaller initial amplitude. Thus, neither the initial time nor the initial amplitude is determined by the condition of linearization within the linear regime. For this reason, we allow for a negative starting time, $t = -t_0 \leq 0$, and imagine that the ensemble has already evolved for a time $t_0 \geq 0$ (linearly) before the (nonlinear) simulation starts, with its time beginning at $t = 0$. This being the case, formula (1) should now read

$$Z_k(t) = (-1)^k \alpha_k Ag(t + t_0)^2. \tag{19}$$

The linear theory in an infinite domain gives

$$Z_k(t) = Z_k(0)\exp(\sqrt{\kappa Ag}t), \tag{20}$$

where $\kappa = 2\pi/\lambda$, for a wavelength λ. Thus, an initial amplitude decrease by a factor of $\exp(-t_0\sqrt{2\pi Ag/\lambda})$ is equivalent to a shift by t_0 in the starting time.

This same point has been emphasized in different terms [38], where the point is made that in a preasymptotic time regime, the scaling law (1) should be replaced by

$$Z_k(t) = (-1)^k \alpha_{2,k} Agt^2 + \alpha_{1,k}\sqrt{Ag}t, \tag{21}$$

with the two unknown coefficients $\alpha_{1,k}$ and $\alpha_{2,k}$ determined by fitting to the simulation data.

Either of these equivalent points of view introduces a source of ambiguity in the data analysis. The second parameter in (19) or (21) will not be well determined by the data given here, because the simulation times are not long enough. We measure the duration of the simulation by the dimensionless ratio of the penetration distance Z_k to a mean or representative initial wavelength λ. For simulations taken to a penetration of a wavelength ratio of about 33, which corresponds to late time simulation results of Youngs, we find a sensitivity of α to t_0 of about 10% for $0 \leq t_0 \leq 4$. For the simulations reported here, the sensitivity is higher. For this reason, the absolute values of α reported are regarded as tentative, while relative values (comparison of two *FronTier* simulations or comparison of *FronTier* to TVD simulations) have a greater assurance.

2.4. Other Factors

Mesh resolution per mode. Our simulations use about 10 zones per initial bubble. Many reported simulations are more coarsely zoned. At these resolutions, the simulations are underresolved. Additional studies on the influence of mesh refinement would be desirable. The effect should be to produce a moderate increase in observed α_b, on the basis of experience in two dimensions.

Length of domain. Waves reflecting from the ends of the computational domain can decrease the value of α_b according to two dimensional studies, especially if the domain boundary is too close to the edge of the mixing zone.

Compressibility. Compressibility has been observed to increase α_b moderately in two-dimensional simulations [6]. We introduce the dimensionless parameter $M^2 = \lambda g/c_2^2$, where c_2 is the sound speed in the heavy fluid and λ is a mean bubble width, measured at the initial time, to characterize the compressibility of the flow. We correct for the effect of finite $M^2 > 0$ in Section 4.1.

Definition of edge of mixing zone. The edge Z_k of the mixing zone is defined as the location of the furthest penetrating bubble ($k = 1$) or spike ($k = 2$), or equivalently the location of $\langle f_k \rangle = 0$. Here f_k is a local volume fraction and $\langle f_k \rangle$ is a transversely averaged volume fraction. This definition is inconvenient for untracked Eulerian simulation codes, as the mass diffusion extends the location of Z_k unrealistically under this definition. The definition has also been criticized as being statistically unstable in the limit of large ensemble size, i.e., of many bubbles. For this reason, many simulations and some experiments report a definition of α_b based on 5% values, $\langle f_k \rangle = 0.05$. The 5% definition leads to a small decrease in α_b. For convenience of comparison, we report 0, 1, and 5% definitions of α_b.

The simulations based on shock capturing algorithms presented here use a level set function to determine locally in x, y, z, t a 50% contour or zero-level surface to represent the interface. This contour can be used by any of the above methods to define global, i.e., transverse averaged, 0, 1, or 5% contours to determine the edge Z_b of the mixing zone and hence α_b. Only the 0% contours are reported for the capturing simulations.

Plainly, these various definitions of Z_b and α_b are not identical. (We observe about a 10% difference resulting from different definitions.) This fact must be kept in mind when comparing simulations to one another or to experiment. See Section 4.1.

3. THE FRONT TRACKING ALGORITHM

Front tracking is a numerical method in which selected waves are explicitly represented in the discrete form of the solution. Examples include shock waves, contact discontinuities, and material interfaces. Other waves, such as leading and trailing edges of rarefaction waves, have continuous states but jumps in their first derivatives. Tracked waves are propagated using the appropriate equations of motion for the given model. For example, if the system of equations consists of a set of hyperbolic conservation laws, $\mathbf{u}_t + \nabla \bullet \mathbf{f} = \mathbf{h}$, then the instantaneous velocity s of a discontinuity surface satisfies the Rankine–Hugoniot equations, $s[\mathbf{u}] = [\mathbf{f}] \bullet \mathbf{n}$. Here $\mathbf{n}$ is the unit normal to the discontinuity surface. During a time step propagation, the type of a wave and the flow field in a neighborhood of the wave determine a local time integrated velocity for each point on the wave in the direction normal to the wave front. Wave propagation consists of moving each point a distance $s\,\Delta t$ in the normal direction as well as computing the time updated states at the new position. Tracking preserves the mathematical structure of the discontinuous waves by maintaining the discrete jump at the wave front, thus eliminating numerical diffusion. It also allows the direct inclusion of the appropriate flow equations for the wave front in the numerical solution.

The front tracking algorithm is described, in its 3D version, in recent publications [15, 16, 18]. This algorithm has been developed into a computer code *FronTier*; see `http://www.ams.sunysb.edu/∼shock/FTdoc.FTmain.html`. There are two essential ideas to the front tracking method. The first is the description of a front or interface as a lower dimensional structure, with supporting data structures and its own dynamics derived from the differential equation being solved. The second essential idea is to use (nonlocal) Riemann solvers to define the dynamics of the front, and ghost cell extrapolation to define a finite difference algorithm to couple the interior cells to the front.

A front tracking code requires a variety of computational tools for its implementation. A geometry package [15, 16, 18, 20, 27] is needed for the description and manipulation of the tracked waves. This package includes data structures describing interface geometry (points, lines, curves, triangles, surfaces, etc.), constructors and destructors for these objects, and routines for the description of interface topology. The latter operations include the location of the nearest point on a tracked interface to a given location and the identification of the connected component containing a given location.

Another important package is a grid generator used to construct a global interpolant of the discrete solution. This interpolant uses the set of state values on a finite difference grid together with the bivalued states at the front to interpolate state values at arbitrary locations. An important feature of the interpolant is that it respect discontinuities at the front, i.e., no interpolations occur between states on opposite sides of a tracked front. An example of such an interpolant is an interface-constrained triangulation using the cell centers of the finite difference grid together with the surface elements from the fronts.

Additional packages include libraries for handling interactions between tracked waves, finite difference solvers, Riemann problem solution packages, equation of state packages, flow initialization, and printout. Also, a package for the redistribution of front points is needed to control numerical instabilities produced due to expanding and converging interface sections.

The construction and redistribution of a three-dimensional interface is described in [15] and is similar to Sethian’s Fast Marching Algorithm [40]. According to this algorithm, the interface is totally determined by its intersections with grid cell edges. Within each cell, the interface is reconstructed from these edge crossings. The reconstruction is unique except for instances of interface bifurcation or change of topology. There are only 16 nonisomorphic reconstructions possible within a local grid cell in 3D, and these are composed of 6 elementary connected interface pieces in each cell.

The flow field (Fig. 1) in a front tracking computation is represented on a set of finite difference cells together with a set of wave front hypersurfaces. In two space dimensions, fronts are sets of piecewise linear curves. Each linear segment is called a bond. States at the front are two valued corresponding to the limit of the flow as the front is approached from either side. Fronts are oriented hypersurfaces and we speak of the left- and right-hand sides of the front, respectively, and denote the corresponding states by the left or right state.

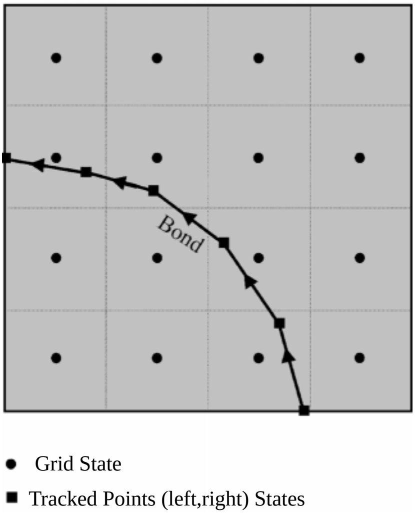


FIG. 1. A representation of the grid for a front tracking computation. The solution is represented on the union of a spatial finite difference grid and a dynamic grid that follows the fronts.

The applications discussed here are modeled by the Euler equations, which describe the conservation laws of mass, momentum, and total energy for a nonreacting compressible fluid. If we let ρ $\mathbf{u}$, P, e, and g, represent the mass density, fluid velocity vector, pressure, specific internal energy, and body force respectively, then these conservation laws can be written as the system of partial differential equations

$$\begin{aligned}\frac{\partial\rho}{\partial t}+\frac{\partial\rho\mathbf{u}_j}{\partial x_j}&=0\\ \frac{\partial\rho\mathbf{u}_i}{\partial t}+\frac{\partial\rho\mathbf{u}_i\mathbf{u}_j}{\partial x_j}+\frac{\partial P}{\partial x_i}&=\rho\mathbf{g}_i\\ \frac{\partial\rho\left(\frac{1}{2}\mathbf{u}^2+e\right)}{\partial t}+\frac{\partial\rho\mathbf{u}_j\left(\frac{1}{2}\mathbf{u}^2+h\right)}{\partial x_j}&=\rho\mathbf{g}_j\mathbf{u}_j.\end{aligned}\tag{22}$$

The system is closed via a thermodynamic equation of state that relates density, pressure, and energy, most commonly through a functional relation $P=P(\rho,e)$.

3.1. Propagation of Front Points

Point propagate [8] is a basic front tracking operation. This operator computes the time-advanced position and state of the front. The currently implemented algorithm uses local dimensional splitting to decompose the equations of motion into components normal and tangential to the interface. Figure 2a shows the basic stencil of states used to compute the contribution of the normal component of flow. The states sl_0 and sr_0 denote the left and right states at the point to be propagated. States sl_i and sr_i are interpolated at distances Δn in the direction normal to the front. The projection of these states onto the line normal to the front is used to compute the interface velocity at the point and a pair of time updated left and right states at front.

Point propagate uses three basic operations, slope reconstruction to compute approximations to the flow gradients along the normal line, Riemann problem solutions to compute interactions between states, and the method of characteristics to compute the contributions of incoming and outgoing waves to the front motion and states.

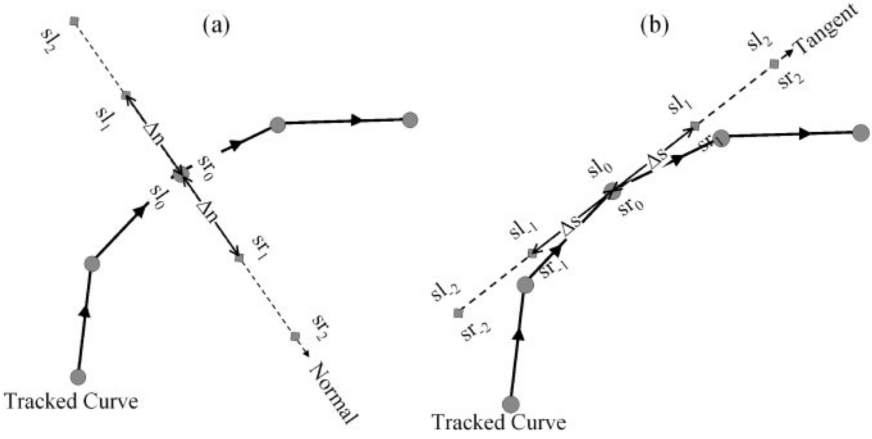


FIG. 2. A schematic showing the stencil of states used in propagating a front point. For simplicity the diagram is shown for two space dimensions. (a) The normal propagate stencil; (b) the stencil used in the tangential update.

The reconstruction step is similar to that used in many shock capturing methods [9, 23, 31, 45, 46] with one important exception. The existence and location of a discontinuity are explicitly known and represented in the reconstructed slopes so that no differencing is performed between states on opposite sides of the front. Otherwise standard limiters, such as the van Leer limiter, are used to compute the reconstructed slopes, and these slopes are used to define a one-dimensional interpolant for state values along the normal line.

Riemann problem solutions are used as predictors of the interface motion. For example, the solution of the Riemann problem with data sl_0, sr_0 defines a set of outgoing waves from the front. Each tracked wave carries a wave type that identifies it with one of the waves from the Riemann problem solution and the motion of that corresponding wave gives the predicted position and state at the next time step along the wave normal. This predictor is then corrected to account for flow gradients on either side of the wave using the method of characteristics. A specific example for the case of a material interface is presented below.

When projected onto the normal line at the point to be propagated, the data for point propagate can be viewed as a one-dimensional initial value problem with Cauchy data given by the reconstructed flow variables. We are interested in solving the Cauchy problem in a neighborhood of the front point for one time step over the time interval from times t_0 to $t_0 + \Delta t$. The tracked point position is denoted by $x_c(t)$ and is initially located as shown in Fig. 3a.

The first step in the propagation algorithm is to solve the Riemann problem with data defined by the projection of the states sl_0 and sr_0 onto the interface normal. That is, we use as the left state data ρ_{l0}, P_{l0}, $u_l = \mathbf{u}_{l0} \bullet \mathbf{n}$, and right state data ρ_{r0}, P_{r0}, $u_r = \mathbf{u}_{r0} \bullet \mathbf{n}$. The midstate velocity, u_m, from the Riemann problem solution is used as a predictor of the contact velocity so that the predicted motion of the point moves from $x_c(t_0)$ to $x_c(t_0) + u_m \Delta t \mathbf{n}$, as shown in Fig. 3a. We also obtain predicted values for the updated states on either side of the contact.

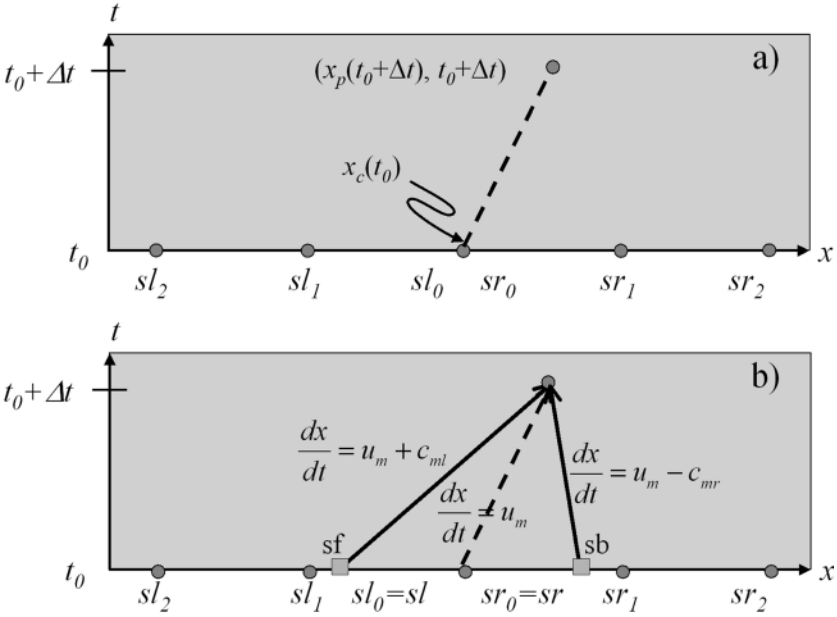


FIG. 3. In normal point propagate the flow state is projected onto a line normal to the interface point. A one-dimensional Cauchy problem is solved to compute the updated front position and state.

From the Riemann problem solution we obtain two sound speeds, cm_l and cm_r, and a midstate velocity u_m, for the states on either side of the contact in the Riemann problem solution. Using these values to approximate the wave speeds of the incoming characteristics on either side of the contact, we trace the incoming characteristics from time $t_0 + \Delta t$ back to time t_0 and interpolate states sf at $x_c(t_0) - cm_l \Delta t$, and sb at $x_c(t_0) + cm_r \Delta t$ at the feet of the incoming characteristics (Fig. 3b).

The contribution of flow gradients on either side of the front to the front motion and state is calculated using the method of characteristics. The Euler equations can be written in characteristic form:

$$\begin{aligned} D_f u + \frac{1}{\rho c} D_f P &= g, \\ D_0 e + P D_0 \frac{1}{\rho} &= 0, \\ D_b u - \frac{1}{\rho c} D_b P &= g. \end{aligned} \tag{23}$$

The differential operators D_f, D_b, and D_0 are defined by $D_f = \partial_t + (u + c)\partial_x$, $D_0 = \partial_t + u\partial_x$, and $D_b = \partial_t + (u - c)\partial_x$. If we integrate the characteristic equations on either side of the contact and approximate the resulting integrals using the two point trapezoidal rule, we obtain a set of implicit equations:

$$\begin{aligned} u^+ - u_f + \frac{1}{2}\left(\frac{1}{\rho_f c_{lf}} + \frac{1}{\rho_l^+ c_l^+}\right)(P^+ - P_f) &= g\Delta t \\ u^+ - u_b - \frac{1}{2}\left(\frac{1}{\rho_b c_b} + \frac{1}{\rho_r^+ c_r^+}\right)(P^+ - P_b) &= g\Delta t \\ e_l^+ - e_l + \frac{P^+ + P_l}{2}\left(\frac{1}{\rho_l^+} - \frac{1}{\rho_l}\right) &= 0 \\ e_r^+ - e_r + \frac{P^+ + P_r}{2}\left(\frac{1}{\rho_r^+} - \frac{1}{\rho_r}\right) &= 0 \\ P^+ = P_l(e_l^+, \rho_l^+), \quad c_l^+ &= c_l(e_l^+, \rho_l^+) \\ P^+ = P_r(e_r^+, \rho_r^+), \quad c_r^+ &= c_r(e_r^+, \rho_r^+). \end{aligned} \tag{24}$$

for the time updated states on the contact. Finally we use central difference in time to compute the net interface normal velocity,

$$\begin{aligned} v_{\text{interface}} &= \frac{1}{2}(u^+ + u_m), \\ x_c(t_0 + \Delta t) &= x_c(t_0) + v_{\text{interface}} \Delta t. \end{aligned}$$

3.2. Accuracy of Normal Point Propagate

As an illustration of the accuracy of the method we consider the specific example of the propagation of a shock front from time t_0 to time $t_0 + \Delta t$ for Burger's equation:

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0. \tag{25}$$

For simplicity we assume that $t_0 = 0$ and that the initial shock is located at $x = 0$ so that our initial data is given by

$$u(x, 0) = \begin{cases} u_l(x), & x < 0 \\ u_r(x), & 0 < x. \end{cases}$$

Let $s(t)$ be the shock location at time t, and define $u_-(t) = u(s(t)-, t)$, $u_+(t) = u(s(t)+, t)$. The Rankine–Hugoniot relations for Burger's equation imply that $\dot{s}(t) = \frac{1}{2}(u_-(t) + u_+(t))$, and since solutions to Burger's equation are constant along the characteristics $dx/dt = u$, we have $u_-(t) = u_l(s(t) - u_-(t)t)$ and $u_+(t) = u_r(s(t) - u_+(t)t)$. Differentiating these relations with respect to time we easily obtain

$$\begin{aligned} \dot{u}_\pm(t) &= u'_{rl}\{\dot{s} - \dot{u}_\pm t - u_\pm\}|_{(s(t),t)} \\ \ddot{u}_\pm(t) &= u''_{rl}\{\dot{s} - \dot{u}_\pm t - u_\pm\}^2 + u'_{rl}\{\ddot{s} - \ddot{u}_\pm t - 2\dot{u}_\pm\}|_{(s(t),t)}. \end{aligned}$$

Using the standard notation $[a] = a_- - a_+$ and $\bar{a} = \frac{1}{2}(a_- + a_+)$, we obtain the derivatives at time $t = 0$:

$$\begin{aligned} \dot{u}_\pm(0) &= \pm\frac{1}{2}u'_{rl}[u]|_{(0,0)} \\ \ddot{u}_\pm(0) &= \frac{1}{4}u''_{rl}[u]^2 \mp u'_{rl}\left(u'_{rl} \pm \frac{1}{4}[u']\right)[u]|_{(0,0)} \\ \dot{s}(0) &= \bar{u}|_{(0,0)} \\ \ddot{s}(0) &= -\frac{1}{4}[u'][u]|_{(0,0)} \\ \dddot{s}(0) &= \frac{1}{4}\{\bar{u}''[u] + 3\bar{u}'[u']\}[u]|_{(0,0)}. \end{aligned}$$

The numerical solution at time Δt, as computed by the normal propagate algorithm, is easily shown to be

$$\begin{aligned} u^n_\pm(\Delta t) &= u_{rl}\left(\pm\frac{1}{2}[u(0)]\Delta t\right) \\ s^n(\Delta t) &= \frac{1}{2}(\bar{u}(0) + \bar{u}^n(\Delta t))\Delta t. \end{aligned}$$

Expanding the numerical solution as a Taylor series in Δt and comparing this to the Taylor series for the exact solution we obtain

$$\begin{aligned} |u^n_\pm(\Delta t) - u_\pm(\Delta t)| &= \left|u'_{rl}\left(u'_{rl} \pm \frac{1}{4}[u']\right)[u]\right|\Big|_{(0,0)}\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3) \\ |s^n(\Delta t) - s(\Delta t)| &= \left|\frac{1}{4}\bar{u}''[u]^2 - \frac{3}{2}\bar{u}'[u'][u]\right|\frac{\Delta t^3}{3!} + \mathcal{O}(\Delta t^4), \end{aligned}$$

from which we conclude that the states at the front are correct to first order and the front position is correct to second order. It is interesting to note that the accuracy of the algorithm improves as the slopes on either side of the front approach zero with the states becoming second-order accurate in the limit where the first derivatives vanish at the front.

3.3. Tangential Sweep

Front points are propagated using dimensional splitting. The previous discussion described the operations used to compute the normal component of this dimensional splitting. This operation moves the tracked points to their time updated locations and updates the states on the front in the normal direction. Once points have been processed using the normal point propagate operator, a second sweep is performed to incorporate tangential flow information. This is accomplished by projecting the states on a tracked wave near a point onto the tangent plane at the point being updated. As shown in Fig. 2 for two space dimensions, states sl_i and sr_i are evaluated by interpolating points at distances Δs in arclength along the curve, each taken from its respective side of the curve. The tangential projections of these states are then used as data for a standard finite difference solver that provides the final time updated state at the front point.

Note on three dimensions. Normal point propagate in three space dimensions is essentially the same as described above. For the tangential step a two-dimensional array is projected onto the tangent plane of the front. The axes of this array are aligned with the axes of principal curvature of the front. These are then used as data for a two-dimensional finite difference solver.

3.4. Ghost Cell Extrapolation for Interior–Front Coupling

Since its inception (see [19]) *FronTier* has used the ghost cell extrapolation algorithm to provide the coupling between the front and the interior system of states and to update irregular cells, those whose regular finite difference stencil overlaps with the front. As mentioned previously, the flow field is represented by the union of a set of front states and finite difference cell states. We call the finite difference cells interior cells. Time stepping consists of two parts, update of the front states and position and update of the interior cell states. The latter process is called the interior sweep. In the current implementation the finite difference grid is a rectangular lattice. Interior cells are updated using standard finite differences (shock capturing) that treat the tracked fronts as internal time-dependent "boundaries." A main feature of the algorithm is that no differencing is done between cells located on opposite sides of the front. For efficiency the interior sweep is implemented as a two-pass process. The first pass ignores the tracked fronts and does a standard finite difference update on the cells. A second pass is then performed to correct those cell states whose domain of dependence overlaps the front. We use the terminology of regular and irregular cells to distinguish cells whose domain of dependence is disjoint from the front (regular cells) from those whose domain of dependence intersects the front (irregular cells). Figure 4 shows a schematic representation of a front section where the light cells indicate locations whose domain of dependence intersects the front. For simplicity a 3×3 stencil is assumed so that a cell is influenced by the front if it lies within one cell on any side to the front at times t or $t + \Delta t$.

Irregular cells are updated by constructing a pseudo-stencil of states taken from the appropriate side of the front. We use a simple algorithm of extrapolation by constant state to construct this stencil. Some representative examples for a one-dimensional case using a direct solver are shown in Fig. 5. On the upper left a front point is initially outside the domain of dependence of the point being updated but moves within that domain during the time step. A CFL condition is enforced that restricts the point from moving more than one

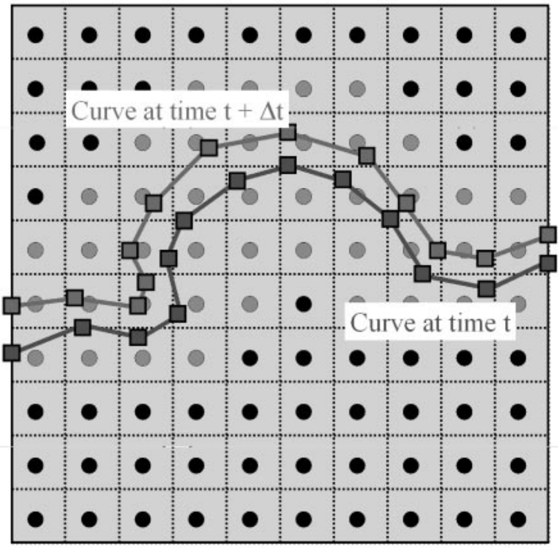


FIG. 4. Finite difference cells are divided into two classes: regular cells whose domain of dependence is disjoint from the front, and irregular cells whose domain of dependence intersects the front.

cell width during the time step so that this front point does not cross the cell center being updated. For a direct solver this means that this cell is essentially equivalent to a regular cell, although implicit solvers would need to treat this cell as irregular. The other three cases show situations where the front lies between the cell center being updated and other adjacent cells. In these cases we construct the pseudo-stencil by formally replacing the state occupying the "wrong" side of the front by the state on the tracked front. This replacement only applies to the stencil used to update the particular irregular cell. This stencil is then passed to a standard finite difference solver that internally treats the cell as a regular cell. Note that we do not try to account for the partial grid cells formed by the front. In effect

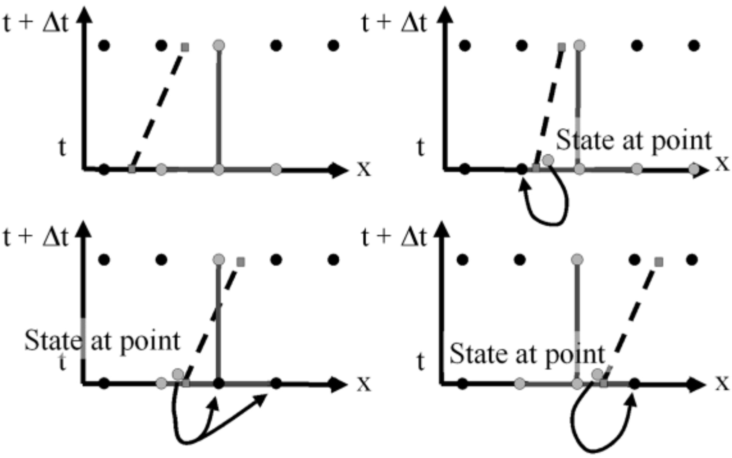


FIG. 5. Irregular cells are updated using pseudo-stencils generated by extrapolating front points to nearby cell centers.

we temporarily move the front to the appropriate cell center for the purposes of the update. This avoids CFL problems associated with small grid sizes.

The algorithms discussed here are implemented in *FronTier*. *FronTier* supports one-, two-, and three-dimensional flows in compressible gas dynamics, elasto-plastic solids, flow in porous media, and material deposition and etching. Finite difference methods implemented for gas dynamics include the Lax–Wendroff method [30, 37], the Colella piecewise linear method [9], and a version of the Bell–Colella–Trangenstein method [3]. The code is written in ANSI C and is portable to all standard Unix platforms, including Cray, IBM, HP, Linux (i686 and alpha), SGI IRIX, and Sun. In addition to the features described above, the code supports dynamic wave interactions such as shock refractions and bubble pinchoff [15–17, 24–27].

4. NEW SIMULATION RESULTS

All simulations reported here are performed on a $2 \times 2 \times 4$ computational domain with a $112 \times 112 \times 224$ grid. The initial interface is a perturbation of a planar $z = 0$ interface. The perturbation is defined by a sum of random Fourier modes, with between 10 and 15 Fourier modes per linear dimension in the initial perturbation for the *FronTier* simulations and between 5 and 15 for the TVD simulations. This definition yields an initial array of about 12×12 bubbles in the perturbation and corresponds to an average initial bubble width of 0.166. For *FronTier* simulations only, the initial velocities and other state variables are also perturbed, based on an analytic solution for the small amplitude (linearized) single mode Rayleigh–Taylor instability equations [14]. The amplitudes and phases are chosen by a random number generator with independent normal distributions. The variance determines the overall amplitude of the interface perturbation, which is approximately equal to a multiple 0.1 of the average initial bubble width. This amplitude is marginally within the domain of the linearized Rayleigh–Taylor theory. Larger initial amplitudes are commonly used, meaning that the simulations are started in a nonlinear regime with strong transients. Initial amplitudes are often reported in units of the full domain length, making the assessment of linear vs nonlinear flow initialization difficult. All simulations have an Atwood number $A = 0.5$ and, except where the compressibility is varied, an initial compressibility value $M^2 \equiv \lambda g / c_2^2 \approx 0.1$.

The three most important properties characterizing a mixing zone are its overall width, the distribution of mass or volume fraction across its width, and the degree of fine scale mixing. Here we analyze the simulations from these points of view.

4.1. *FronTier* with Small Compressibility

We consider $M^2 = 0.22$ and 0.11, both small, for the compressibility. The purpose of this choice is to allow extrapolation to the incompressible limit $M^2 = 0$ in the determination of α_b. The determination of α uses (19) with $t_0 = 4$ for the *FronTier* simulation. The equivalent decrease in initial amplitude is by a factor of 0.1. We set $t_0 = 2$ for the TVD simulations. Extrapolation to the incompressible limit gives a reduction of about 10% in α_b from the larger of the two (small) values of compressibility, and results in agreement with the experiment, see Tables I and II.

The late time interface separating heavy and light fluids is shown in Fig. 6. The increase in α_b with compressibility was reported earlier in 2D simulations [6], with an increase by up to a factor of 2 for $M^2 = 1.0$.

TABLE II
Values for α_b as Determined by *FronTier*: Comparison of the Effects of Three Definitions of α_b

	Definitions of α_b		
M^2	0%	1%	5%
0.22	0.083	0.078	0.070
0.11	0.076	0.074	0.068
0 (Extrap.)	0.069	0.070	0.066

Note. Values of α_b for compressible flow and extrapolation to $M^2 = 0$.

The 5% contour method for computing α_b gives similar values, and a further reduction of 5 to 10% for α_b. The resulting (lower) value of α_b for *FronTier* simulations probably improves the agreement with experiment. Results are presented in Table II. Data for the computation of α_b from the simulation of Fig. 6 is given in Fig. 7. Due to the uncertainty in the data analysis originating in the limited duration and penetration depth in these simulations, the quoted values of α_b are regarded as tentative. Other possible systematic errors in the simulation, such as grid resolution, were discussed in Section 2.

4.2. TVD with and without Artificial Compression

Here we show the influence of dissipative effects by comparing a *FronTier* (tracked) simulation with two TVD (capturing) simulations. We duplicate earlier reported capturing growth rates for comparable simulation times and ensemble sizes. Of the factor of 2 variation

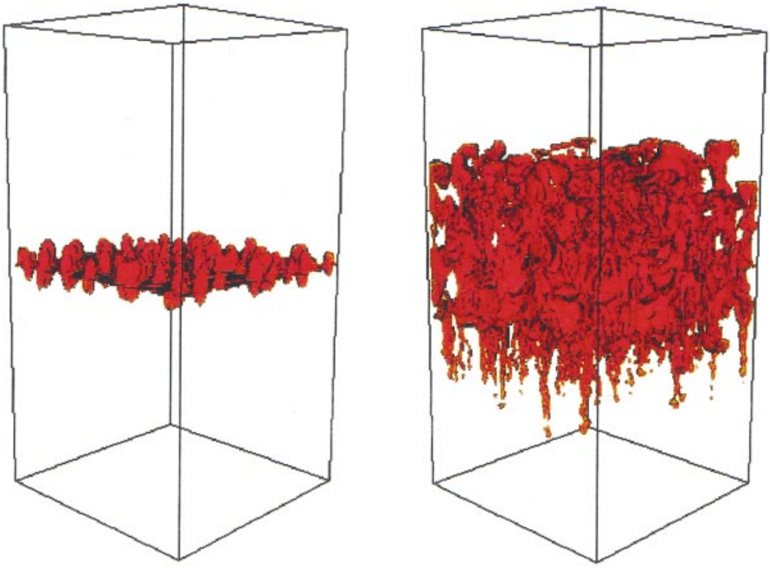


FIG. 6. Early and late time steps in a simulation of the Rayleigh–Taylor instability. The interface between the two fluids is shown. Here $M^2 = 0.11$.

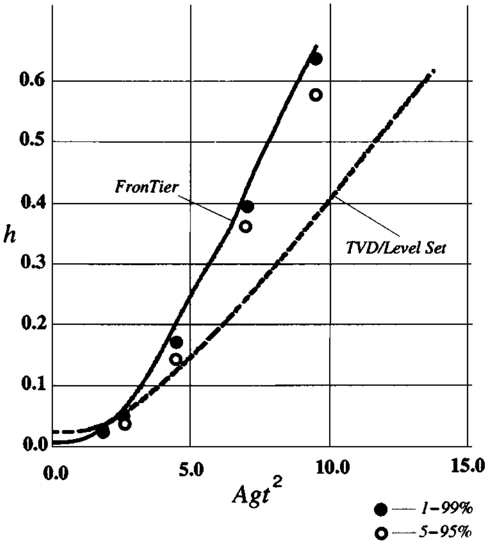


FIG. 7. Height vs Agt^2. Here the solid line is the 0% definition for the *FronTier* simulation and the dashed line comes from the TVD simulation. For the *FronTier* simulation, the black circles are the 1% definition and the open circles result from the 5% definition.

in principal simulation values for α_b, about half appears at the simulation times reported here. Our results explain this half quantitatively as due to the dissipative effects of capturing algorithms. By comparing TVD to *FronTier*, we infer that diffusion of interfacial vorticity and density jumps is significant, accounting for a 40% decrease in α_b, and about half of the total discrepancy with most capturing simulations. By comparing two different capturing simulations, one with artificial compression (AC) to limit mass diffusion, we infer that the major dissipative effect is viscosity. An alternate explanation is that the AC algorithm, being less effective in its nonlinear application, does not sufficiently control mass diffusion. For the purpose of this comparison, we keep the compressibility fixed at $M^2 \approx 0.1$. All inputs and sensitive factors (except as explicitly noted) are the same as those for the *FronTier* simulations, so the differences which result can be attributed to dissipative effects of the TVD capturing algorithm. The comparable incompressible values of α_b and the effect of different definitions of α_b are shown in Table II and summarized in Table III. The tabulated numerical diffusion length is the local width of the numerically defined interface, as determined at a middle value of t, $t = 6$, out of a total simulation time of $t = 12$. Observe that this length is comparable to the size of the bubbles and spikes in

TABLE III
Dependence of α_b on Numerical Diffusion and Viscosity

Method	Observed numerical diffusion length	α_b (0%; compressible)
FronTier	0 cells	0.083
AC	6–7 cells	0.053
TVD	11 cells	0.050

the flow for the two capturing algorithms. Artificial compression reduces this length by half.

4.3. Fine Scale vs Chunk Mix

4.3.1. *Numerical Mixing Fraction*

We introduce the local volume fraction $f_k(x, y, z, t)$ as the fraction of fluid k at the space–time location x, y, z, t. The numerical mixing fraction is defined [49] by the transverse (and ensemble) averaged correlation

$$\theta(z, t) = \frac{\langle f_1 f_2 \rangle}{\langle f_1 \rangle \langle f_2 \rangle}. \tag{26}$$

Youngs [49] reports mixing fractions $\theta \approx 0.8$ for a van Leer advection algorithm, indicating nearly perfect numerical mixing. *FronTier*, with zero mass diffusion across a tracked interface, has $\theta \equiv 0$. The Dimonte–Schneider experiments, performed with immiscible fluids, have a mixing fraction $\theta = 0$. Youngs' experiments used both miscible and immiscible fluids.

4.3.2. *The Coherence Correlation Length*

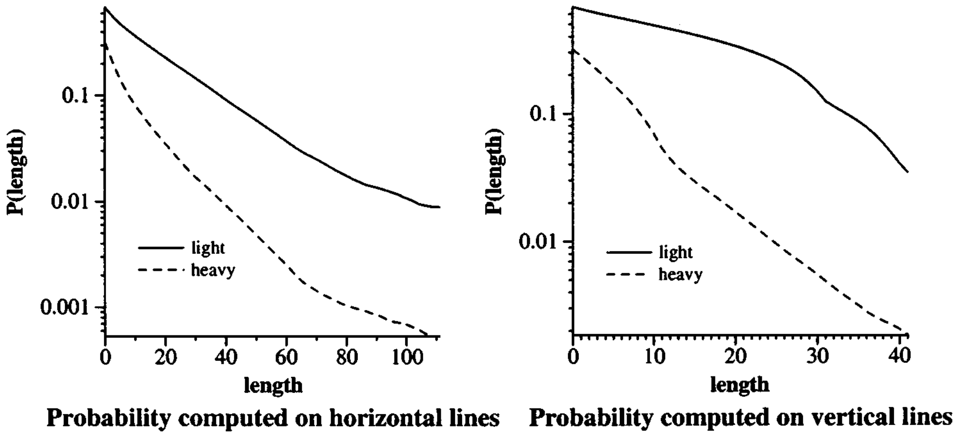
This section presents two main results. The first is the observation that turbulent mixing, as described both numerically and experimentally, generates a Markov random field, thereby making available a range of powerful analysis tools for its description. The second main result is the observation that the coherence length, defined by the Markov description, does not satisfy acceleration based t^2 scaling, and thus marks the introduction of a new length scale and possible beginning of a new flow regime. See also the turbulence-based analysis of [10]. We start with the observation that the coherence probability, i.e., the probability to remain within a single phase while moving on a straight line, satisfies an exponential fall off with distance. The characteristic length λ for this decay is called the coherence length.

The exponential law for the coherence pdf can be seen in simulations (Fig. 8) and experiments [11] (Fig. 9). The exponential structure for the pdf is equivalent to a Poisson process for phase boundaries encountered for motion along a straight line, and thus to a Markov property for the binary random field defined by the two phase flow. This fact leads to an elegant mathematical description of various transition probabilities [34].

Table IV lists coherence lengths as extracted from both simulation and experiment. The experimental data is generated from the central half of the mixing zone only. However, due to the narrowness of the simulation mixing zone, a larger central region was used to construct the simulation data. We note the important fact that the coherence length does not scale with t^2 or the width of the mixing zone, which changes by a factor of 3 between the two experimental times shown in Table IV and by a factor of 2 for the simulation times. Thus, it must describe a new length scale and the possible beginning of a new flow regime. The coherence pdf should not be confused with the bubble size pdf, which also satisfies an exponential law [2] but obeys a t^2 scaling. The experimental and simulation numbers show similar trends.

TABLE IV
Experimental and Simulation Values for the Coherence Length Scale in a Markov Random Field Description of the Interface Statistics

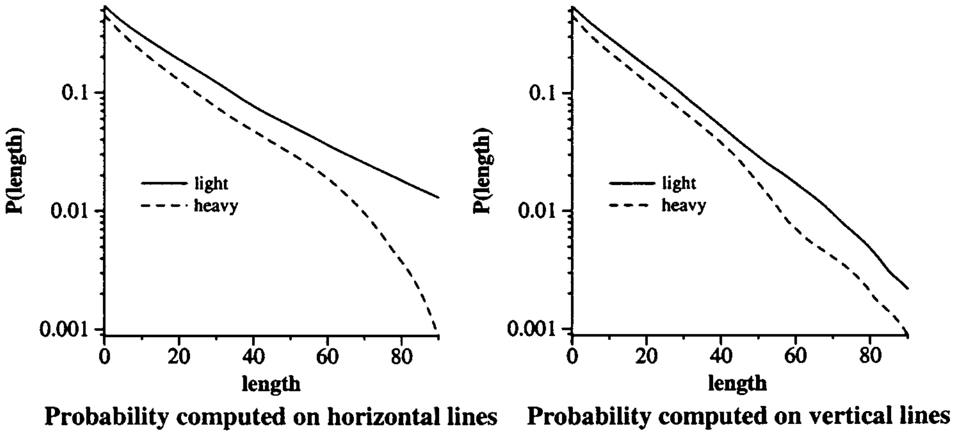
	Experiment late time	Experiment early time	Simulation late time	Simulation early time
λ_h				
Light fluid	24	12	23	22
Heavy fluid	19	19	15	24
λ_v				
Light fluid	17	10	29	5
Heavy fluid	15	13	6	7



Probability computed on horizontal lines

Probability computed on vertical lines

FIG. 8. Exponential rate laws for the probability of an interval of length l to lie totally within a specified phase, determined from *FronTier* simulations reported here. The data are well fit to an exponential law $\exp(-l/\lambda)$ over two orders of magnitude in probability.



Probability computed on horizontal lines

Probability computed on vertical lines

FIG. 9. Exponential rate laws for the probability of an interval of length l to lie totally within a specified phase as determined experimentally [11].

5. CONCLUSIONS

We have analyzed the mixing growth rate constant α_b in multimode (random) Rayleigh–Taylor instability in a 3D planar geometry. Despite the large effort made to determine this quantity, disagreements and inconsistencies have persisted.

We identified two significant factors which are sufficient to obtain *FronTier* simulations that agree with experimental data. The first factor is a correction for compressibility and the second is a correction for differing definitions of the constant α_b. We also identify a source of indeterminacy in the data analysis for the growth rate α.

The numerical dissipation of the capturing algorithms is a significant factor, sufficient to explain part of the spread in simulation values, and probably all of the principal discrepancies observed up to the simulation times studied here. See Table I. For the simulation times and ensemble sizes explored, numerical viscosity appears to play a larger role than numerical mass diffusion. See Table III.

We have begun an exploration of new physical length scales, which could signal a transition to a new flow regime for Rayleigh–Taylor mixing.

Further studies are needed to resolve remaining issues, including refinement of the mesh per mode, increase in the number of modes, and simulations carried to later time. The present paper provides a perspective on, but does not definitively resolve, the causes of the discrepancies concerning the growth rate α_b. For this reason we list some outstanding questions and propose possible research which could help to resolve remaining questions.

1. To what extent can long wavelength noise in the experiments contribute to the experimental value of α_b? How rigid a restriction on the noise spectrum is the observed growth of $Z_1(t)$ scaling linearly in t^2 over the experimental time periods? Experimental characterization of initial conditions would contribute to a resolution of this issue.
2. The decreasing, time-dependent simulation values of α_b for capturing algorithms signals a new length scale to break the t^2 scaling law. We propose here that this length scale originates in numerical dissipation.
3. The 40% difference reported here between the TVD algorithm and *FronTier*, when run to identical times and with identical resolution, indicates that dissipation is significant and sufficient to explain the principal discrepancies among simulations up to the simulation times reported here. We propose to run *FronTier* simulations to later time to determine the value of α_b which results.

REFERENCES

1. U. Alon, J. Hecht, D. Mukamel, and D. Shvarts, Scale invariant mixing rates of hydrodynamically unstable interfaces, *Phys. Rev. Lett.* **72**, 2867 (1994).
2. U. Alon, D. Shvarts, and D. Mukard, Scale-invariant regime in Rayleigh–Taylor bubble-front dynamics, *Phys. Rev. E* **48**, 1008 (1993).
3. J. Bell, P. Colella, and J. Trangenstein, Higher order Godunov methods for general systems of hyperbolic conservation laws, *J. Comput. Phys.* **82**, 362 (1989).
4. A. Blair, N. Metropolis, J. von Neuman, A. H. Taub, and M. Tsingou, A study of a numerical solution to a two-dimensional hydrodynamic problem, *Math. Tables Aids Comput.* **13**, 145 (1959).
5. Shiyi Chen, private communication (1999).
6. Y. Chen, Y. Deng, J. Glimm, G. Li, D. H. Sharp, and Q. Zhang, A renormalization group scaling analysis for compressible two-phase flow, *Phys. Fluids A* **5**(11), 2929 (1993).

7. B. Cheng, J. Glimm, X. L. Li, and D. H. Sharp, Subgrid models and DNS studies of fluid mixing, in *Proceedings of the 7th International Conference on the Physics of Compressible Turbulent Mixing, 1999* (2000), submitted.
8. I.-L. Chern, J. Glimm, O. McBryan, B. Plohr, and S. Yaniv, Front tracking for gas dynamics, *J. Comput. Phys.* **62**, 83 (1986).
9. P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, *SIAM J. Comput.* **6**(1), 104 (1985).
10. S. B. Dalziel, P. F. Linden, and D. L. Youngs, Self-similarity and internal structure of turbulence induced by Rayleigh–Taylor instability, *J. Fluid Mech.* (1999), **399**, 1 (1999).
11. G. Dimonte and M. Schneider, Density ratio dependence of Rayleigh–Taylor mixing for sustained and impulsive acceleration histories, *Phys. Fluids* (1999), **12**, 304 (2000).
12. D. Drew and S. Passman, *Theory of Multicomponent Fluids*, Applied Mathematical Sciences (Springer–Verlag, New York, 1999).
13. R. E. Duff, F. H. Harlow, and C. W. Hirt, Effects of diffusion on interface instability between gases, *Phys. Fluids* **5**, 417 (1962).
14. C. L. Gardner, J. Glimm, J. Grove, O. McBryan, R. Menikoff, D. H. Sharp, and Q. Zhang, A study of chaos and mixing in Rayleigh–Taylor and Richtmyer–Meshkov unstable interfaces, in *Proceedings of the International Conference on 'The Physics of Chaos and Systems Far from Equilibrium' (CHAOS' 87), Monterey, CA, USA, Jan. 11–14, 1987*, Special Issue of *Nuclear Physics B* (proceedings supplements section), edited by M. Duong-van and B. Nichols (1988).
15. J. Glimm, J. Grove, X.-L. Li, and D. C. Tan, Robust computational algorithms for dynamic interface tracking in three dimensions, *SIAM J. Sci. Comp.* (1999), in press.
16. J. Glimm, J. Grove, X.-L. Li, and N. Zhao, Simple front tracking, in *Contemporary Mathematics*, edited by G.-Q. Chen and E. D. Bebedetto (American Mathematical Society, Providence, 1999), Vol. 238, p. 133.
17. J. Glimm, J. Grove, W. B. Lindquist, O. McBryan, and G. Tryggvason, The bifurcation of tracked scalar waves, *SIAM J. Comput.* **9**, 61 (1988).
18. J. Glimm, J. W. Grove, X.-L. Li, K.-M. Shyue, Q. Zhang, and Y. Zeng, Three dimensional front tracking, *SIAM J. Sci. Comp.* **19**, 703 (1998).
19. J. Glimm, D. Marchesin, and O. McBryan, Subgrid resolution of fluid discontinuities II, *J. Comput. Phys.* **37**, 336 (1980).
20. J. Glimm and O. McBryan, A computational model for interfaces, *Adv. Appl. Math.* **6**, 422 (1985).
21. J. Glimm and D. H. Sharp, Chaotic mixing as a renormalization group fixed point, *Phys. Rev. Lett.* **64**, 2137 (1990).
22. J. Glimm and D. H. Sharp, Stochastic partial differential equations: Selected applications in continuum physics, in *Stochastic Partial Differential Equations: Six Perspectives, Mathematical Surveys and Monographs*, edited by R. A. Carmona and B. L. Rozovkii (American Mathematical Society, Providence, 1997).
23. E. Godlewski and P. A. Raviart, *Numerical Approximation of Hyperbolic Systems of Conservation Laws* (Springer–Verlag, New York, 1991).
24. J. Grove, The interaction of shock waves with fluid interfaces, *Adv. Appl. Math.* **10**, 201 (1989).
25. J. Grove, Irregular shock refractions at a material interface, in *Shock Compression of Condensed Matter 1991*, edited by S. Schmidt, R. Dick, J. Forbes, and D. Tasker (North–Holland, 1992), p. 241.
26. J. Grove and R. Menikoff, The anomalous reflection of a shock wave at a material interface, *J. Fluid Mech.* **219**, 313 (1990).
27. J. W. Grove, Applications of front tracking to the simulation of shock refractions and unstable mixing, *J. Appl. Num. Math.* **14**, 213 (1994).
28. G. Hedstrom, Models of difference schemes for $u_t + u_x = 0$ by partial differential equations, *Math. Comp.* **29**, 969 (1975).
29. Ya. Kucherenko, L. I. Shibarshev, V. I. Chitaikin, S. A. Balabin, and A. P. Palaev, Experimental investigation into the the self-similar mode of the gravitational turbulent mixing, in *Proceedings of the 3rd International Workshop on the Physics Compressible Turbulent Mixing* (1991).
30. P. Lax and B. Wendroff, Systems of conservation laws, *Commun. Pure Appl. Math.* **13**, 217 (1960).
31. R. LeVeque, *Numerical Methods for Conservation Laws* (Birkhäuser Verlag, Basel, 1992).

32. X.-L. Li, B. X. Jin, and J. Glimm, Numerical study for the three dimensional Rayleigh–Taylor instability using the TVD/AC scheme and parallel computation, *J. Comp. Phys.* **126**, 343 (1996).

33. P. F. Linden, J. M. Redondo, and D. L. Youngs, Molecular mixing in Rayleigh–Taylor instability, *J. Fluid Mech.* **265**, 97 (1994).

34. G. C. Pomraning, Transport theory in discrete stochastic mixtures, *Adv. Nucl. Sci. Technol.* **24**, 47 (1996).

35. Lord Rayleigh, Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density, in *Scientific Papers* (Cambridge Univ. Press, UK, 1900), Vol. II, p. 200.

36. K. I. Read, Experimental investigation of turbulent mixing by Rayleigh–Taylor instability, *Physica D* **12**, 45 (1984).

37. R. Richtmyer and K. Morton, *Difference Methods for Initial Value Problems*, 2nd ed. (Interscience, New York, 1967).

38. D. Saltz and D. Sendersky, Computation of two-phase mixing properties in Rayleigh–Taylor instability, Technical report, University at Stony Brook (1999).

39. M. B. Schneider, G. Dimonte, and B. Remington, Large and small scale structure in Rayleigh–Taylor mixing, *Phys. Rev. Lett.* **80**(16), 3507 (1998).

40. J. A. Sethian, *Level Set Methods* (Cambridge Univ. Press, UK, 1996).

41. D. H. Sharp, An overview of Rayleigh–Taylor instability, *Physica D* **12**, 3 (1984).

42. D. H. Sharp and J. A. Wheeler, Late stage of Rayleigh–Taylor instability, Technical report, Institute of Defense Analyses (1961), unpublished.

43. D. M. Snider and M. J. Andrews, Rayleigh–Taylor and shear driven mixing with an unstable thermal stratification, *Phys. Fluids* **6**(10), 3324 (1994).

44. G. I. Taylor, The instability of liquid surfaces when accelerated in a direction perpendicular to their planes I, *Proc. R. Soc. London A* **201**, 192 (1950).

45. E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer–Verlag, Berlin, 1997).

46. B. van Leer, Towards the ultimate conservative difference scheme: V. A second order sequel to Godunov's method, *J. Comput. Phys.* **32**, 101 (1979).

47. R. Warming and Hyett, The modified equation approach to the stability and accuracy analysis of finte-difference methods, *J. Comp. Phys.* **14**, 159 (1974).

48. D. L. Youngs, Numerical simulation of turbulent mixing by Rayleigh–Taylor instability, *Physica D* **12**, 32 (1984).

49. D. L. Youngs, Three-dimensional numerical simulation of turbulent mixing by Rayleigh–Taylor instability, *Phys. Fluids A* **3**, 1312 (1991).

50. D. L. Youngs, Private communication (1999).

51. Q. Zhang, Validation of the chaotic mixing renormalization group fixed point, *Phys. Lett. A* **151**, 18 (1990).