FronTier:
A Compressible Hydrodynamics Front Tracking Code

A Short Course in Front Tracking

Los Alamos National Laboratory Report LA-UR 99-3985

A series of lectures describing the theory and practice of the method of front tracking as implemented in the code FronTier

John W. Grove
mailto:jgrove@lanl.gov

Staff Scientist
Hydrodynamics Method Group
Applied Theoretical and Computational Physics Division
Los Alamos National Laboratory
Los Alamos, NM 87545

and

Professor
Department of Applied Mathematics and Statistics
University at Stony Brook
Stony Brook, New York 11794-3600

I. Systems of Conservation Laws

II. The Riemann Problem for Gas Dynamics

III. Finite Difference Methods for Gas Dynamics

IV. Front Tracking in One Space Dimension

V. Front Tracking in Two Space Dimensions

VI. Implementing Front Tracking

VII. Wave Interactions in Two Space Dimensions

VIII. Implementing Wave Interactions in Front Tracking

IX. Implementing Wave Interactions in Front Tracking II

X. Applications and Research

Systems of Conservation Laws

LA-UR 99-3985

8/2/99

Click here to start

Table of Contents

Principle of Conservation (fixed regions)

Principle of Conservation (moving regions)

Distributional Form of Conservation Laws

Generalizations and Specializations

Differential Form of the Conservation Laws

Continuum Flows

Flow Fields

Reynold's Transport Theorem

Conservation of Mass

Conservation of Momentum

Conservation of Energy

Creation of Entropy (Second Law of Thermodynamics)

Constitutive Equations (closure of the systems)

Constitutive Equations (equations of state)

Constitutive Equations (incomplete equations of state)

Author: John W. Grove
Hydrodynamics Methods Group
Los Alamos National Laboratory
Los Alamos, New Mexico 87545

Email: jgrove@lanl.gov

Exercises

Further Reading

Principle of Conservation

(fixed regions)

A flow in d space dimensions is described by three quantities, a state vector **u**, a flux tensor **f**, and a source vector **h**. In general -

$$\mathrm{u}(\mathbf{x},t)=\begin{pmatrix} u_1(\mathbf{x},t) \\ u_2(\mathbf{x},t) \\ \vdots \\ u_n(\mathbf{x},t) \end{pmatrix},\ \mathbf{f}(\mathbf{u},\mathbf{x},t)=\begin{pmatrix} f_{11}(\mathbf{u},\mathbf{x},t) & f_{12}(\mathbf{u},\mathbf{x},t) & \cdots & f_{1n}(\mathbf{u},\mathbf{x},t) \\ f_{21}(\mathbf{u},\mathbf{x},t) & f_{22}(\mathbf{u},\mathbf{x},t) & \cdots & f_{2n}(\mathbf{u},\mathbf{x},t) \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1}(\mathbf{u},\mathbf{x},t) & f_{n2}(\mathbf{u},\mathbf{x},t) & \cdots & f_{nn}(\mathbf{u},\mathbf{x},t) \end{pmatrix},\ \mathbf{h}(\mathbf{u},\mathbf{x},t)=\begin{pmatrix} h_1(\mathbf{u},\mathbf{x},t) \\ h_2(\mathbf{u},\mathbf{x},t) \\ \vdots \\ h_n(\mathbf{u},\mathbf{x},t) \end{pmatrix}.$$

$$\mathbf{x}=\begin{pmatrix} x_1 & x_2 & \cdots & x_d \end{pmatrix}$$

The principle of conservation states that the rate of change of the quantity of **u** in any region Ω within the domain of the flow is given by

$$(1)\qquad \frac{d}{dt}\int_{\Omega} \mathrm{d}\mathbf{x}\, u_i(\mathbf{x},t)+\oint_{\partial\Omega} \mathrm{dA}(\mathbf{x})\, f_{ij}(\mathbf{u}(\mathbf{x},t),\mathbf{x},t)n_j(\mathbf{x})=\int_{\Omega} \mathrm{d}\mathbf{x}\, h_i(\mathbf{u}(\mathbf{x},t),\mathbf{x},t)\ ,$$

where **n** is the outer pointing normal vector to the boundary of Ω. More compactly in vector notation this system can be written

$$(1_v)\qquad \frac{d}{dt}\int_{\Omega} \mathrm{d}\mathbf{x}\ \mathbf{u}\ +\oint_{\partial\Omega} \mathrm{dA}\ \mathbf{fn}=\int_{\Omega} \mathrm{d}\mathbf{x}\,\mathbf{h}\ .$$

Principle of Conservation

(moving regions)

If the region Ω is in motion with velocity vector $\mathbf{v}$ on the boundary of Ω then the conservation principle becomes:

$$(2) \qquad \frac{d}{dt}\int_{\Omega(t)} \mathrm{d}\mathbf{x}\, u_i(\mathbf{x},t) + \oint_{\partial\Omega(t)} \mathrm{dA}(\mathbf{x})\Big(f_{ij}(\mathbf{u}(\mathbf{x},t),\mathbf{x},t) - u_i(\mathrm{x},t)v_j \Big) n_j(\mathbf{x},t) = \int_{\Omega(t)} \mathrm{d}\mathbf{x}\, h_i(\mathbf{u}(\mathbf{x},t),\mathbf{x},t)\ ,$$

or in vector form

$$\left(2_v\right) \qquad \frac{d}{dt}\int_{\Omega(t)} \mathrm{d}\mathbf{x}\ \mathbf{u} + \oint_{\partial\Omega(t)} \mathrm{dA}(\mathbf{x})(\mathbf{f} - \mathbf{u}\otimes\mathbf{v})\mathbf{n} = \int_{\Omega(t)} \mathrm{d}\mathbf{x}\,\mathbf{h}\ .$$

Distributional Form of Conservation Laws

An alternate formulation of the conservation laws that is useful in analysis is the distributional form of the equations. This formulation requires that the equality

$$(3)\quad \int dt \int \mathrm{dx}\left(u_i(\mathrm{x},t)\frac{\partial \phi_i}{\partial t}(\mathrm{x},t) + f_{ij}(\mathbf{u}(\mathrm{x},t),\mathrm{x},t)\frac{\partial \phi_i}{\partial x_j}(\mathrm{x},t) - h_i(\mathbf{u}(\mathrm{x},t),\mathrm{x},t)\phi_i(\mathrm{x},t)\right) + \int \mathrm{dx}\; u_i(\mathrm{x},0)\phi_i(\mathrm{x},0) = 0$$

hold for arbitrary test functions $\phi_i(\mathbf{x},t)$ that are infinitely differential with compact support in the domain of the systems of equations.

We can also write this equation in a more compact vector form:

$$(3_v)\quad \int dt \int \mathrm{dx}\left(\mathbf{u}\bullet\frac{\partial \boldsymbol{\phi}}{\partial t} + \mathbf{f}:\nabla\boldsymbol{\phi} - \mathbf{h}\bullet\boldsymbol{\phi}\right) + \int \mathrm{dx}\; \mathbf{u}\bullet\boldsymbol{\phi}\Big|_{t=0} = 0$$

Generalizations and Specializations

Various generalizations and specializations of the conservation law equations are important in applications.

Homogeneous flows: The flux and source functions do not depend explicitly on **x** or *t*. (This is the typical case considered in this course.)

Anisotropic flows: Flows where the flux function **f** depends on the normal direction **n**. Note that in such flows the conservation law must be defined by formula 1. An example of an anisotropic flow is elastic flow in a crystal.

Viscous and heat conducting flows: The flux function also depends on the gradient of the solution. Note that in such cases it is assumed that solution function is sufficiently smooth.

The fluxes and sources may be functions of other variables that are solutions to separate dynamical equations.

Most of the examples considered in this course will be isotropic, homogeneous, inviscid (i.e. non-viscous and non-heat conducting) flows, so that the fluxes and sources only depend on the state vector **u**.

Differential Form of the Conservation Laws

It is traditional and convenient to rewrite the integral form of a conservation law in differential form. Suppose **u** is a smooth solution to the conservation law,

$$\frac{d}{dt}\int_{\Omega} d\mathbf{x}\,\mathbf{u} + \oint_{\partial\Omega} dA\,\mathbf{fn} = \int_{\Omega} d\mathbf{x}\,\mathbf{h} \quad \forall \quad \text{domains } \Omega.$$

Also assume that the flux and source vectors are smooth functions of their arguments. Then we can move the time derivative inside the integral and apply the divergence theorem to the surface integral to obtain.

$$\int_{\Omega} d\mathbf{x}\left(\mathbf{u}_t + \nabla \bullet \mathbf{f} - \mathbf{h}\right) \quad \forall \quad \text{domains } \Omega.$$

So that **u** satisfies the partial differential equation.

$$\mathbf{u}_t + \nabla \bullet \mathbf{f} = \mathbf{h}$$

For nonsmooth **u**, stating that **u** is a solution to this partial differential equation is interpreted to mean **u** is a solution of the associated integral equation.

Continuum Flows

All of the numerical discussions in this class will focus on the flow of a compressible fluid, which is a special case of a general continuum flow. The state of the flow at a given point ($\mathbf{x}$,t) is defined by a mass density ρ, a momentum vector $\mathbf{m} = \rho\mathbf{v}$ ($\mathbf{v}$ is the fluid velocity vector), and a total energy density E. Usually we write the total energy density in terms of the kinetic energy density and the specific internal energy e, $E = \rho(\mathrm{v}^2/2 + e)$. Here $\mathrm{v}^2 = \langle\mathbf{v},\mathbf{v}\rangle$ is the square of the fluid flow speed. We will show in the next few slides that the physical laws of conservation of mass, momentum, and energy imply that such flows satisfy a system of form (1), where:

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho\,\mathbf{v} \\ \rho(\frac{1}{2}v^2 + e) \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho\,\mathbf{v} \\ \rho\,\mathbf{v}\otimes\mathbf{v} - \boldsymbol{\sigma} \\ \rho(\frac{1}{2}v^2 + e)\mathbf{v} - \boldsymbol{\sigma}\mathbf{v} + \mathbf{q} \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} 0 \\ \rho\,\mathbf{g} \\ -\rho\,\langle \mathbf{g}, \boldsymbol{\sigma}\mathbf{v} \rangle \end{pmatrix}$$

Here $\boldsymbol{\sigma} = (\sigma^{ij})$ is the Cauchy stress tensor. The surface force per unit area on a given surface element is given by $\boldsymbol{\sigma}\mathbf{n}$. The vector $\mathbf{g}$ is the body force per unit mass.

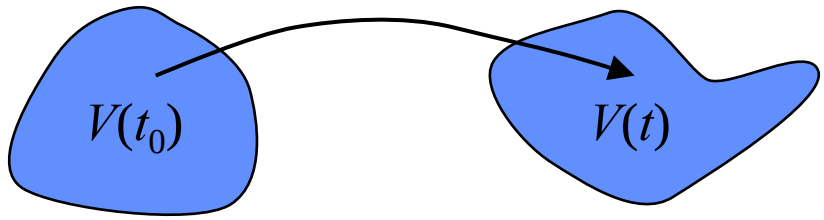
Flow Fields

A flow field is a time dependent vector field defined in some domain D in space so that at a given time t and position $\mathbf{x}$, a velocity vector $\mathbf{v}(\mathbf{x},t)$ is defined. This field need not be smooth (indeed it is in general not continuous), but we will assume sufficient regularity in the field so that all of the integrals considered in the subsequent discussion are well defined. The flow is defined by the solution to the set of ordinary differential equations:

$$\frac{d}{dt}\mathbf{x}(\mathbf{x}_0,t)=\mathbf{v}(\mathbf{x},t),\ \mathbf{x}(\mathrm{x}_0,t_0)=\mathbf{x}_0$$

If Ω is a regular region in D, we define the flow of Ω, $\Omega(t)$ defined for $t > t_{0,}$, as the set $\Omega(t) = \{\mathbf{x}(\mathbf{x}_0,t) \mid \mathbf{x}_0 \in \Omega\}$. Physically $\Omega(t)$ corresponds to the location of the fluid "particles" at time t that resided in the set Ω at time t_0. A volume that moves with the flow is called a material volume. More generally we define a flow as a function $\mathbf{x} = \mathbf{x}(\mathbf{X}, t)$, so that $\mathbf{x}(\mathbf{X}, t)$ gives the position of particle $\mathbf{X}$ at time t. The variable $\mathbf{X}$ is usually called the Lagrangian position of the fluid particle.

Reynold's Transport Theorem



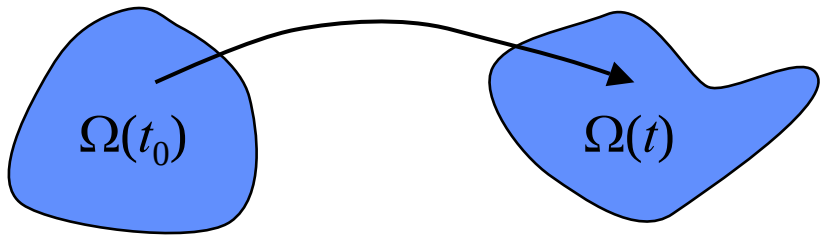
Let $V^*(t)$ be a moving volume with bounding surface $S^*(t)$ and outward unit normal vector **n**. Let **b** denote the velocity at a point on $S^*(t)$. If $\chi(\mathbf{x},t)$ is a integrable function on $V^*(t)$ the quantity of χ in $V^*(t)$ and the rate of change of the quantity of χ in $V^*(t)$ are given by the two integrals:

$$\int_{V^*(t)} d\mathbf{x}\,\chi(\mathbf{x},t) \text{ and } \frac{d}{dt}\int_{V^*(t)} d\mathbf{x}\,\chi(\mathbf{x},t) \text{ respectively.}$$

Reynold's transport theorem states that the rate of change of the quantity of χ in $V^*(t)$ can be computed by the formula:

$$\frac{d}{dt}\int_{V^*(t)} d\mathbf{x}\,\chi(\mathbf{x},t)\Bigg|_{t=t_0} = \frac{d}{dt}\int_{V^*(t_0)} d\mathbf{x}\,\chi(\mathbf{x},t) + \oint_{S^*(t_0)} dA(\mathbf{x})\,\chi(\mathbf{x},t_0)\,\mathbf{b}(\mathbf{x},t_0)\bullet\mathbf{n}(\mathbf{x},t_0).$$

Conservation of Mass



Conservation of mass states that the quantity of matter in a material volume does not change,

$$\int_{\Omega(t_0)} d\mathbf{x}\,\rho(\mathbf{x},t_0) = \int_{\Omega(t)} d\mathbf{x}\,\rho(\mathbf{x},t),\quad t \geq t_0,$$

or equivalently

$$\frac{d}{dt}\int_{\Omega(t)} d\mathbf{x}\,\rho(\mathbf{x},t) = 0.$$

Using Reynold's transport theorem and letting $\Omega = \Omega(t_0)$, we have at time $t = t_0$:

$$\frac{d}{dt}\int_{\Omega} d\mathbf{x}\,\rho(\mathbf{x},t) + \oint_{\partial\Omega} dA(x)\,\rho(\mathbf{x},t)\mathbf{v}(\mathbf{x},t)\bullet\mathbf{n}(\mathbf{x}) = 0$$

where **n** is the outer pointing unit normal to $\partial\Omega$ which is just the first equation of the conservation laws for continuum flows.

Conservation of Momentum

Conservation of Momentum (or Newton's second law) states that the time rate of change of the quantity of momentum in a material volume is equal to the net body forces plus the net surface forces that act on the volume. It can be shown that the force per unit area **T** acting on a surface element with unit normal **n** is **T** = **σn**, where s is Cauchy stress. If **g** is the body force per unit mass acting on the material volume. Then momentum conservation implies that:

$$\frac{d}{dt}\int_{\Omega(t)} d\mathbf{x}\,\rho\mathbf{v} \quad = \oint_{\partial\Omega(t)} dA\,\sigma\mathbf{n}(\mathbf{x},t) \;+\; \int_{\Omega(t)} d\mathbf{x}\,\rho\mathbf{g}\,.$$

Applying Reynold's transport theorem to the left hand side we obtain with $\Omega = \Omega(t_0)$:

$$\frac{d}{dt}\int_{\Omega} d\mathbf{x}\,\rho\mathbf{v} \;+\int_{\Omega} dA\,\rho\,\mathbf{v}\mathbf{v}\bullet\mathbf{n} \;-\; \oint_{\partial\Omega} dA\,\sigma\mathbf{n} \;=\int_{\Omega} d\mathbf{x}\,\rho\mathbf{g}\,.$$

We note that both **σ** and **g** may be functions of space, time, or other dynamical variables such as density and temperature.

Conservation of Energy

Conservation of energy is a formulation of first law of thermodynamics, dE = δW + δQ, i.e. the change in energy of a material volume is equal to the work done on the volume by body and surface forces plus the change in heat of the body. In integral terms this can be stated:

$$\frac{d}{dt}\int_{\Omega(t)} d\mathbf{x}\,\rho\left(\frac{1}{2}v^2+e\right) = \oint_{\partial\Omega(t)} dA\,\mathbf{v}\bullet\sigma\mathbf{n} + \int_{\Omega(t)} d\mathbf{x}\,\rho\mathbf{g}\bullet\mathbf{v} - \oint_{\partial\Omega(t)} dA\,\mathbf{q}\bullet\mathbf{n}\ .$$

As before using Reynold's transport theorem with $\Omega = \Omega(t_0)$ we can write this equation as:

$$\frac{d}{dt}\int_{\Omega} d\mathbf{x}\,\rho\left(\frac{1}{2}v^2+e\right) + \int_{\Omega} d\mathbf{x}\,\rho\left(\frac{1}{2}v^2+e\right)\mathbf{v}\bullet\mathbf{n} - \oint_{\partial\Omega} dA\,\mathbf{v}\bullet\sigma\mathbf{n} + \oint_{\partial\Omega} dA\,\mathbf{q}\bullet\mathbf{n} = \int_{\Omega} d\mathbf{x}\,\rho\mathbf{g}\bullet\mathbf{v}\ .$$

Once again we have introduced a new unknown variable **q**, the heat flux.

Creation of Entropy

(Second Law of Thermodynamics)

The second law of thermodynamics states that the entropy of a closed system can never decrease. In contrast to the previous conservation laws, which can be stated in terms of equalities, this law is expressed as an inequality:

$$\frac{d}{dt}\int_{\Omega(t)} d\mathbf{x}\,\rho S \quad + \quad \oint_{\partial\Omega(t)} dA\,\Xi \bullet \mathbf{n} \quad \geq 0$$

S is the specific entropy, and Ξ is the entropy flux through the boundary of the material region. For a Newtonian fluid, it can be shown that $\Xi = \mathbf{q}/T$, where T is the temperature. For the inviscid flows considered in much of this course, equality will hold in the above equation with $\Xi = 0$ (the flow is adiabatic). An important exception is for flows across shock waves, where there is net entropy production.

Constitutive Equations

(closure of the systems)

The equations for a continuum flow consist of up to five conservation laws in up to three space dimensions and time. The dependent variables are the mass density ρ, the fluid velocity $\mathbf{v}$, the Cauchy stress tensor $\boldsymbol{\sigma}$, the body force $\mathbf{g}$, the specific internal energy e, and the heat flux $\mathbf{q}$. In addition to apply the entropy inequality we have the specific entropy S, and the entropy flux Ξ. Since the Cauchy stress tensor is symmetric, we have respectively:

Number of Space Variables	Number of Conservation Laws	Number of Dependent Variables
1	3	8
2	4	14
3	5	21

We see that the system in not closed in the sense that we have more unknowns than equations. In many applications closure is obtained by imposing constitutive laws among the dependent variables and the space and time coordinates. For example in an inviscid gas flow, the heat and entropy fluxes are zero, the body force is independent of the state variables (perhaps constant), and the Cauchy stress is a scalar tensor $\sigma^{ij} = -P\delta^{ij}$, where $\delta^{ij} = 1$ for I = j, 0 otherwise. P is the thermodynamic pressure and is a given function $P = P(\rho,e)$. This special case of the continuum flow equations is called the Euler equations.

Constitutive Equations

(equations of state)

With this reduction we obtain the same number of equations as dependent variables and the system is closed. The relationship between pressure, density, and energy is called the equation of state (more properly the incomplete equation of state) of the material. It characterizes the material. By far the most common equation of state is the perfect gas (polytropic, gamma law) equation of state, where:

$$P = (\gamma - 1)\rho e$$

The constant γ is the ratio of the specific heats of the material $\gamma = c_P/c_V$, where by definition:

$$c_p = \left.\frac{\delta Q}{\delta T}\right|_P = T\left.\frac{\partial S}{\partial T}\right|_P$$ = change in heat per unit temperature at constant pressure

$$c_V = \left.\frac{\delta Q}{\delta T}\right|_V = T\left.\frac{\partial S}{\partial T}\right|_V$$ = change in heat per unit temperature at constant volume

Constitutive Equations

(incomplete equations of state)

The formulas for the specific heats use a version of the first law of thermodynamics that identifies the change in heat with the differential TdS, $\delta Q = TdS$, which in turn is related to the change in internal energy by the formula $de = TdS - PdV$, which is basically the statement that the change in energy of a system is equal to the work done on the system by the pressure plus the heat gained. The variable V is called the specific volume and equals $1/\rho$.

The relationship between density, pressure, and energy is called the incomplete equation of state since it does not include any information about the temperature of the system. In ordinary gas dynamics this is all that is needed to solve the systems of equations. In more complicated systems (such as molecular mixing), the temperature is an important variable. The corresponding complete equation of state for a perfect gas is:

$$e = c_V T, \quad PV = nRT$$

Where c_V is constant, n is the molecular weight of the material (moles/unit mass), and R is a universal constant. Both c_V and n are material properties.

Exercises

1. Consider the scalar advection equation in one space dimension, $u_t + (a\,u)_x = 0$, a constant. For smooth u derive both corresponding conservation laws (integral and distributional), and show that for general (i.e. measurable and locally integrable) u both the integral and distributional form of the conservation law are equivalent.

2. Derive the incomplete perfect gas equation of state from the corresponding complete formulation. In particular what is c_P and what is γ in terms of c_V, n, and R?

3. Write down the differential form of the Euler equations.

4. Prove Reynold's transport theorem for smooth χ:

$$\frac{d}{dt}\int_{\Omega(t)} d\mathbf{x}\chi(\mathbf{x},t)\bigg|_{t=t_0} = \frac{d}{dt}\int_{\Omega(t_0)} d\mathbf{x}\chi(\mathbf{x},t)\bigg|_{t=t_0} + \oint_{\partial\Omega(t_0)} dA(\mathbf{x})\chi(\mathbf{x},t_0)\mathbf{v}\bullet\mathbf{n}$$

where $\mathbf{v}$ is the velocity of $\partial\Omega$ and $\mathbf{n}$ is the outer normal to $\partial\Omega$.

Advanced Optional Exercise for the Mathematically Inclined:

Show that in general the integral and distributional form of the conservation laws are equivalent. Consider some of the following questions. Under what circumstances are the integrals that appear in the formulation defined, and what sort of assumptions need to be made regarding the dependence of the flux and sources on their arguments?

Further Reading

- P. Thompson, *Compressible-Fluid Dynamics*, McGraw-Hill 1972. (Main reference but out of print.)
- R. Menikoff and B. Plohr, *The Riemann problem for fluid flow of real materials*, Reviews of Modern Physics, Vol. 61, No. 1, January 1989.
- R. Courant and K. O. Friedrichs, *Supersonic Flow and Shock Waves*, Springer-Verlag 1948.
- L. D. Landau and E. M. Lifshitz, *Fluid Mechanics*, Pergamon Press,1959.

The Riemann Problem for Gas Dynamics

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

The Pure Initial Value Problem, and the Riemann Problem

The Riemann Problem

Hyperbolic Partial Differential Equations

Characteristics of Gas Dynamics (1)

Characteristics of Gas Dynamics (2)

Characteristics of Gas Dynamics (3)

Characteristics of Gas Dynamics (4)

Characteristics of Gas Dynamics (5)

Shock Waves

Rankine-Hugoniot Equations (1)

Rankine-Hugoniot Equations (2)

Rankine-Hugoniot Equations for Gas Dynamics (1)

Rankine-Hugoniot Equations for Gas Dynamics (2)

Self-Similarity

Rarefaction Waves in Gas Dynamics

The Riemann Wave Curve (1)

The Riemann Wave Curve (2)

The Riemann Wave Curve (perfect gas)

Solution of the Riemann Problem for gas dynamics (1)

Solution of the Riemann Problem for gas dynamics (2)

Graphical Solution of the Riemann Problem for Gas Dynamics

Exercises

Further Reading

The Pure Initial Value Problem, and the Riemann Problem

Consider the pure initial value problem for a homogeneous system of conservation laws with no source terms in one space dimension:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0, \quad \mathbf{u}(x,0) = u_0(x), -\infty < x < \infty$$

Where as discussed previously we interpret solutions to this partial differential equation in the weak sense:

$$\frac{d}{dt}\int_a^b \mathrm{d}x\ \mathbf{u}(x,t) + \mathbf{f}(\mathbf{u}(b,t)) - \mathbf{f}(\mathbf{u}(a,t)) = 0,\ \forall a,b$$

The Riemann problem is defined as the initial value problem for this system with two valued piecewise constant initial data. More precisely we have initial data:

$$u_0(x) = \begin{cases} u_l, & x < 0 \\ u_r, & x > 0 \end{cases}$$

The Riemann Problem

The Riemann problem is a fundamental tool for studying the interaction between waves. It has played a central role both in the theoretical analysis of systems of hyperbolic conservation laws and in the development and implementation of practical numerical solutions of such systems. Basically the Riemann problem gives the micro-wave structure of the flow. One can think of the propagation of the flow as a set of small scale Riemann problems between adjacent regions, followed by the interaction between the waves arising from these Riemann problems. This idea was formalized in the fundamental paper of Glimm, "*Solutions in the Large for Nonlinear Hyperbolic Systems of Equation*", that established the first existence theorem for solutions to the initial value problem for nonlinear hyperbolic systems of equations, as well as numerically by Godunov, "*A Finite Difference Method for the Numerical Computation and Discontinuous Solutions of the Equations of Fluid Dynamics*" , which forms the basis for many advanced numerical methods.

Before proceeding further we need to review briefly the notion of hyperbolicity.

Hyperbolic Partial Differential Equations

Smooth solutions of a system of conservation laws are also solutions of a corresponding quasi-linear partial differential equation. Let $\mathbf{A} = (a_{ij})$ be the Jacobi matrix corresponding to the flux function **f(u)**:

$$a_{ij} = \frac{\partial f_i(\mathbf{u})}{\partial u_j}$$

Then smooth solutions to $\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0$, also satisfy the equation $\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x} = 0$. This system is said to be hyperbolic if the matrix **A** has all real eigenvalues and a complete set of eigenvectors:

$$\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n, \quad \mathbf{A}\mathbf{r}_i = \lambda_i \mathbf{r}_i, \ \mathbf{l}_i\mathbf{A} = \lambda_i \mathbf{l}_i, \ \mathbf{l}_i\mathbf{r}_i = \delta_{ij}$$

Where n is the number of equations in the system. These eigenvectors are called the characteristic speeds of the hyperbolic system.

Characteristics of Gas Dynamics (1)

For gas dynamics we have the Euler equations:

$$\frac{\partial \rho}{\partial t} + \nabla \bullet \rho \mathbf{v} = 0$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \bullet \rho \mathbf{v} \otimes \mathbf{v} + \nabla P = 0$$

$$\frac{\partial \rho (\frac{1}{2} v^2 + e)}{\partial t} + \nabla \bullet \left(\rho \mathbf{v} (\frac{1}{2} v^2 + e) + P \mathbf{v} \right) = 0$$

For smooth flow we can write this equation in quasi-linear form:

$$\frac{D\rho}{Dt} + \rho \nabla \bullet \mathbf{v} = 0$$

$$\frac{D\mathbf{v}}{Dt} + \frac{\nabla P}{\rho} = 0$$

$$\frac{De}{Dt} + \frac{P}{\rho} \nabla \bullet \mathbf{v} = 0$$

Where, $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{v} \bullet \nabla$ is the material derivative of the flow.

Characteristics of Gas Dynamics (2)

The pressure P is related to the density and energy by an incomplete equation of state, $P = P(\rho,e)$. If we define the Grüneisen exponent Γ and the sound speed c by the formulas:

$$\Gamma = \rho \left.\frac{\partial P}{\partial e}\right|_{\rho}, \qquad c^2 = \left.\frac{\partial P}{\partial \rho}\right|_{e} + \frac{P}{\rho^2}\left.\frac{\partial P}{\partial e}\right|_{\rho},$$

$$\nabla P = \rho\Gamma\nabla e + \left(c^2 - \Gamma\frac{P}{\rho}\right)\nabla\rho$$

We can write the one dimensional Euler equations in quasi-linear form

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ u\\ e\end{pmatrix} + \begin{pmatrix} u & \rho & 0 \\ \frac{1}{\rho}\left(c^2 - \frac{P}{\rho}\Gamma\right) & u & \Gamma \\ 0 & \frac{P}{\rho} & u \end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}\rho\\ u\\ e\end{pmatrix} = 0$$

Where u is the x component of the velocity vector.

Characteristics of Gas Dynamics (3)

One can easily compute the eigenvalues and eigenvectors of this system. We have:

$$\lambda_1 = u - c,\ r_1 = \begin{pmatrix} \rho \\ -c \\ P/\rho \end{pmatrix}, \qquad l_1 = \left(\frac{c^2 - \Gamma P/\rho}{2\rho c^2},\quad -\frac{1}{2c},\quad \frac{\Gamma}{2c^2} \right)$$

$$\lambda_2 = u, \qquad r_2 = \begin{pmatrix} \Gamma \\ 0 \\ -\dfrac{c^2 - \Gamma P/\rho}{\rho} \end{pmatrix},\ l_2 = \left(\frac{P}{\rho c^2},\quad 0,\quad \frac{-\rho}{c^2} \right)$$

$$\lambda_3 = u + c,\ r_3 = \begin{pmatrix} \rho \\ c \\ P/\rho \end{pmatrix}, \qquad l_3 = \left(\frac{c^2 - \Gamma P/\rho}{2\rho c^2},\quad \frac{1}{2c},\quad \frac{\Gamma}{2c^2} \right)$$

Characteristics of Gas Dynamics (4)

The analysis of the characteristics for gas dynamics can be considerably simplified by introducing the entropy S (the evaluation of which requires the complete equation of state), and using pressure as a fundamental thermodynamic variable. By the first law of thermodynamics we have $TdS = dE+PdV$,

and if we write $\rho = \rho(P,S)$, (recall $V=1/\rho$). Then one can show that:

$$\Gamma = -\frac{c^2}{\rho T}\left.\frac{\partial \rho}{\partial S}\right|_P, \quad c^2 = \left.\frac{\partial P}{\partial \rho}\right|_S = \left(\left.\frac{\partial \rho}{\partial P}\right|_S\right)^{-1}$$

$$\frac{DP}{Dt} + \rho c^2 \nabla \bullet \mathbf{v} = 0$$

$$\frac{D\mathbf{v}}{Dt} + \frac{\nabla P}{\rho} = 0$$

$$\frac{DS}{Dt} = 0$$

$$\frac{\partial}{\partial t}\begin{pmatrix} P \\ u \\ S \end{pmatrix} + \begin{pmatrix} u & \rho c^2 & 0 \\ \frac{1}{\rho} & u & 0 \\ 0 & 0 & u \end{pmatrix} \frac{\partial}{\partial x}\begin{pmatrix} P \\ u \\ S \end{pmatrix} = 0$$

Characteristics of Gas Dynamics (5)

In terms of pressure, velocity, and entropy, the eigenvectors become:

$$\lambda_1 = u - c,\ r_1 = \begin{pmatrix} \rho c \\ -1 \\ 0 \end{pmatrix}, \qquad l_1 = \begin{pmatrix} \frac{1}{2\rho c}, & -\frac{1}{2}, & 0 \end{pmatrix}$$

$$\lambda_2 = u, \qquad r_2 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad l_2 = (0, \quad 0, \quad 1)$$

$$\lambda_3 = u + c,\ r_3 = \begin{pmatrix} \rho c \\ 1 \\ 0 \end{pmatrix}, \qquad l_3 = \begin{pmatrix} \frac{1}{2\rho c}, & \frac{1}{2}, & 0 \end{pmatrix}$$

Shock Waves

An important property of gas dynamics and other nonlinear hyperbolic systems is the existence of discontinuous solutions. Indeed such discontinuities are unavoidable and can arise spontaneously from smooth solution through the phenomenon of shock breaking. Such solutions must be interpreted in the weak sense as described in the previous lecture. A critical aspect of discontinuous solutions is that the states on either side of a discontinuity are not arbitrary but must be related by a system of algebraic equations known as the Rankine-Hugoniot equations. The next series of slides will discuss the derivation of these equations and their applications to gas dynamics.

Consider a system of conservation laws, $\mathbf{u}_t + \nabla \bullet \mathbf{f} = \mathbf{h}$ in a domain Ω. Suppose that the flow in Ω is smooth except across a smooth moving surface S(t). We assume that S(t) divides Ω into two regular regions Ω_1 and Ω_2 . Let $\phi(\mathbf{x},t)$ be a vector of C^∞ test functions with support in the interior of Ω.

Rankine-Hugoniot Equations (1)

Applying the distributional form of the conservation law, we have:

$$\int_t^{t+\Delta t} dt \int_\Omega d\mathbf{x}\ \left(\mathbf{u}\bullet\phi_t + \mathbf{f}\bullet\nabla\phi + \mathbf{h}\bullet\phi\right) = 0,$$

$$\int_t^{t+\Delta t} dt \int_{\Omega_1(t)} d\mathbf{x}\ \left(\mathbf{u}\bullet\phi_t + \mathbf{f}\bullet\nabla\phi + \mathbf{h}\bullet\phi\right) + \int_t^{t+\Delta t} dt \int_{\Omega_2(t)} d\mathbf{x}\ \left(\mathbf{u}\bullet\phi_t + \mathbf{f}\bullet\nabla\phi + \mathbf{h}\bullet\phi\right) = 0$$

Using the fact that u is smooth inside both $\Omega_1(t)$ and $\Omega_2(t)$, we can add the quantity $(\mathbf{u}_t + \nabla\bullet\mathbf{f} - \mathbf{h})\bullet\phi = 0$ inside the two integrals over these two regions separately to obtain:

$$\int_t^{t+\Delta t} dt \int_{\Omega_1(t)} d\mathbf{x}\ \left((\mathbf{u}\bullet\phi)_t + \nabla\bullet(\mathbf{f}\,\phi)\right) + \int_t^{t+\Delta t} dt \int_{\Omega_2(t)} d\mathbf{x}\left((\mathbf{u}\bullet\phi)_t + \nabla\bullet(\mathbf{f}\,\phi)\right) = 0$$

Rankine-Hugoniot Equations (2)

Applying the divergence theorem to the space time regions swept out by $\Omega_i(t)$, we obtain:

$$\int_t^{t+\Delta t} dt \int_{S(t)} dA(\mathbf{x}) \left([\mathbf{f}]\mathbf{n} - s[\mathbf{u}] \right) \bullet \phi = 0$$

Where $\mathbf{n}(\mathbf{x},t)$ is the spatial unit normal to $S(t)$ and s is the speed of the moving surface in the direction $\mathbf{n}$, and [] denotes the jump in a quantity across the surface $S(t)$ in the direction $\mathbf{n}$. Since the test function ϕ and Δt are arbitrary we have at each point on the discontinuity surface the equations:

$$s[\mathbf{u}] = [\mathbf{f}]\mathbf{n}$$

Rankine-Hugoniot Equations for Gas Dynamics (1)

Applying the Rankine Hugoniot equations to the gas dynamic equations, we derive the formulas:

$$\rho_0(s - \mathbf{v}_0 \bullet \mathbf{n}) = \rho_1(s - \mathbf{v}_1 \bullet \mathbf{n}) = m,$$

$$m\mathbf{v}_0 - P_0\mathbf{n} = m\mathbf{v}_1 - P_1\mathbf{n}, \quad \text{or}$$

$$\left(\rho_0(\mathbf{v}_0 \bullet \mathbf{n} - s)^2 + P_0\right)\mathbf{n} - m\left(\mathbf{v}_0 - (\mathbf{v}_0 \bullet \mathbf{n})\mathbf{n}\right) = \left(\rho_1(\mathbf{v}_1 \bullet \mathbf{n} - s)^2 + P_1\right)\mathbf{n} - m\left(\mathbf{v}_1 - (\mathbf{v}_1 \bullet \mathbf{n})\mathbf{n}\right),$$

$$m\left(e_1 - e_0 - \left(\frac{P_1 + P_0}{2}\right)(V_0 - V_1)\right) = 0.$$

There are two cases depending on whether the mass flux m is zero or nonzero:

$$m \neq 0,$$

$$m^2 = \frac{P_1 - P_0}{V_0 - V_1},$$

$$\mathbf{v}_1 = \mathbf{v}_0 + \frac{(P_1 - P_0)}{m}\mathbf{n},$$

$$s = \mathbf{v}_0 \bullet \mathbf{n} + V_0 m = \mathbf{v}_1 \bullet \mathbf{n} + V_1 m,$$

$$e_1 = e_0 + \left(\frac{P_1 + P_0}{2}\right)(V_0 - V_1),$$

$$m = 0,$$

$$P_0 = P_1,$$

$$s = \mathbf{v}_0 \bullet \mathbf{n} = \mathbf{v}_1 \bullet \mathbf{n}.$$

Waves with m ≠ 0 are shocks, waves with m = 0 contact discontinuities.

Rankine-Hugoniot Equations for Gas Dynamics (2)

For perfect gas the Hugoniot equations for shocks can be further simplified. Here we have $c^2 = \gamma P/\rho$, and:

$$m^2 = \rho_0^2 c_0^2 \frac{P_1/P_0 + \mu^2}{1+\mu^2}, \qquad s = \mathbf{v}_0 \bullet \mathbf{n} \pm c_0 \sqrt{\frac{P_1/P_0 + \mu^2}{1+\mu^2}},$$

$$\rho_1 = \rho_0 \frac{P_1/P_0 + \mu^2}{1+\mu^2\, P_1/P_0}, \qquad e_1 = e_0\, P_1/P_0 \frac{1+\mu^2\, P_1/P_0}{P_1/P_0 + \mu^2},$$

$$\mathbf{v}_1 = \mathbf{v}_0 \pm \frac{c_0}{\sqrt{\dfrac{P_1/P_0 + \mu^2}{1-\mu^2}}}\mathbf{n}, \qquad \mu^2 = \frac{\gamma - 1}{\gamma + 1}$$

We see that the flow state behind the shock is completely determined by the flow state ahead of the shock, the shock normal, and the pressure behind the shock. This statement is still true for general equations of state, but it may not be possible to explicitly write down the solution as for a perfect gas. More generally, given the state ahead of a shock and the shock normal, the flow behind the shock is determined by one additional condition. The variables commonly used include the shock speed, the shock Mach number (ratio of the speed of the shock with respect to a fluid at rest and the sound speed), the pressure behind the shock, or the density behind the shock.

Self-Similarity

Homogeneous, isotropic conservation laws with no body forces, and the Euler equations for gas dynamics (with no gravity) in particular, satisfy the important property of self-similarity, the equations are invariant under the transformation $t \to \alpha t$, $\mathbf{x} \to \alpha \mathbf{x}$, $\alpha > 0$. If $\mathbf{u}(\mathbf{x},t)$ is a solution, then so is $\mathbf{u}(\alpha\mathbf{x},\alpha t)$. Since the initial data for a Riemann problem is invariant under the transformation $x \to \alpha x$, we conclude that if the solution to the Riemann problem is unique, then it must satisfy $\mathbf{u}(x,t) = \mathbf{u}(\alpha x,\alpha t)$, i.e. $\mathbf{u}(x,t) = \mathbf{u}(x/t)$ for $t > 0$. If we let $\xi = x/t$, assume the solution is smooth, and insert this into the quasi-linear form of the partial differential equation, we obtain:

$$\xi \frac{d\mathbf{u}}{d\xi} = \mathbf{A}(\mathbf{u}) \frac{d\mathbf{u}}{d\xi}.$$

We see that ξ is an eigenvalue of $\mathbf{A}(\mathbf{u})=d\mathbf{f}/d\mathbf{u}$, and $d\mathbf{u}/d\xi$ is a corresponding eigenvector. Such solutions are called centered rarefaction waves or centered simple waves. Alternatively, if $\mathbf{u}$ is not smooth, then it can consist of a jump discontinuity connecting two constant regions, and the values of the solution are related by the Rankine-Hugoniot equations.

Rarefaction Waves in Gas Dynamics

Applying the simple wave equations to gas dynamics we obtain the solutions:

$$\xi = u \pm c, \qquad u_1 = u_0 \pm \int_{P_0}^{P_1} \left.\frac{dP}{\rho c}\right|_S, \qquad S_1 = S_0$$

$$\xi = u, \qquad u_1 = u_0, \qquad P_1 = P_0$$

For the left hand case, the density is found by solving the equation $\rho = \rho(P_1, S_0)$ given by the equation of state relation between density, pressure, and entropy. For a perfect gas this can be solved to obtain:

$$\rho_1 = \rho_0 \left(\frac{P_1}{P_0} \right)^{\frac{1}{\gamma}}$$

The Riemann Wave Curve (1)

The shock wave curve through a given state (indexed by 0) is defined as the locus of all states that can be connected to that state by a shock wave. This curve consists of two branches corresponding to a positive or negative mass flux across the wave. The side of the shock from which fluid particles travel into and through the shock is called the ahead side of the shock. If the given state corresponds to the state ahead of a shock, and m is positive, we say the wave is forward moving, for m negative we call the wave backward moving. Thermodynamics requires that the entropy produced across the shock be nonnegative. For most equations of state this corresponds to an increase in pressure from the ahead to behind side of the shock. Thus only the branch of the wave curve corresponding to increasing pressure is physical. A decrease in pressure across a wave corresponds to a rarefaction wave. The wave curve, defined for all P is the concatenation of the shock wave curves for $P > P_0$ and the rarefaction curve for $P < P_0$. It can be shown that this curve is twice continuously differentiable. Of particular interest is the component of the wave curve that relates the velocity behind the wave and the pressure behind the wave. The formula for this curve is given on the next slide.

The Riemann Wave Curve (2)

$$u = u_0 \pm \begin{cases} \sqrt{(P-P_0)(V_0 - V(P,P_0,V_0))}, & P > P_0 \\ \displaystyle\int_{P_0}^{P} \left.\frac{dP}{\rho c}\right|_S, & P < P_0 \end{cases}$$

The function $V(P,P_0,V_0)$ is found by solving the Hugoniot relation,

$$e = e_0 + \left(\frac{P+P_0}{2}\right)(V_0 - V),$$

$$P = P(e,V),$$

and $P = P(e,V)$ is the incomplete equation of state. If we define the mass flux across a general wave m, by the formula $[P] = m[u]$, we can write the wave curve as:

$$u = u_0 \pm \frac{(P-P_0)}{m(P,P_0,V_0)}, \quad m(P,P_0,V_0) = \begin{cases} \sqrt{\dfrac{(P-P_0)}{(V_0 - V(P,P_0,V_0))}}, & P > P_0 \\ \dfrac{(P-P_0)}{\displaystyle\int_{P_0}^{P} \left.\frac{dP}{\rho c}\right|_S}, & P < P_0 \end{cases}$$

The Riemann Wave Curve (perfect gas)

For a perfect gas Riemann wave curve can be computed explicitly.

$$u = u_0 \pm \frac{c_0}{\gamma}\begin{cases}\left(P/P_0 - 1\right)\sqrt{\dfrac{1+\mu^2}{P/P_0+\mu^2}}, P > P_0 \\ \dfrac{2\gamma}{\gamma-1}\left(\left(P/P_0\right)^{\frac{\gamma-1}{2\gamma}} - 1\right), \quad P < P_0\end{cases}$$

or

$$\mathrm{u} = u_0 \pm \frac{\left(P - P_0\right)}{m}, \quad m = \rho_0 c_0 \begin{cases}\sqrt{\dfrac{P/P_0+\mu^2}{1+\mu^2}}, & P > P_0 \\ \dfrac{\gamma-1}{2\gamma}\dfrac{P/P_0 - 1}{\left(P/P_0\right)^{\frac{\gamma-1}{2\gamma}} - 1}, & P < P_0\end{cases}$$

Solution of the Riemann Problem for gas dynamics (1)

The solution of the Riemann problem for gas dynamics consistsof constant regions separated by waves. Moving from left to right, these are a backward shock or rarefaction wave, a contact discontinuity, and a forward moving shock or rarefaction wave. The Riemann data on the left will be the ahead state for the left moving wave, while the Riemann data on the right will be the ahead state for the right moving wave. The Rankine-Hugoniot conditions (which agree with the simple wave conditions for a contact discontinuity) for the middle wave are the pressure and velocity are continuous across this wave. If we let u_m and P_m denote the common values of the pressure and velocity on either side of the contact, and let $m_l = m(P,P_l,V_l)$ and $m_r = m(P,P_r,V_r)$. We have:

$$u_m = u_l - \frac{P_m - P_l}{m_l(P_m, P_l, V_l)} = u_r + \frac{P_m - P_r}{m_r(P_m, P_r, V_r)}$$

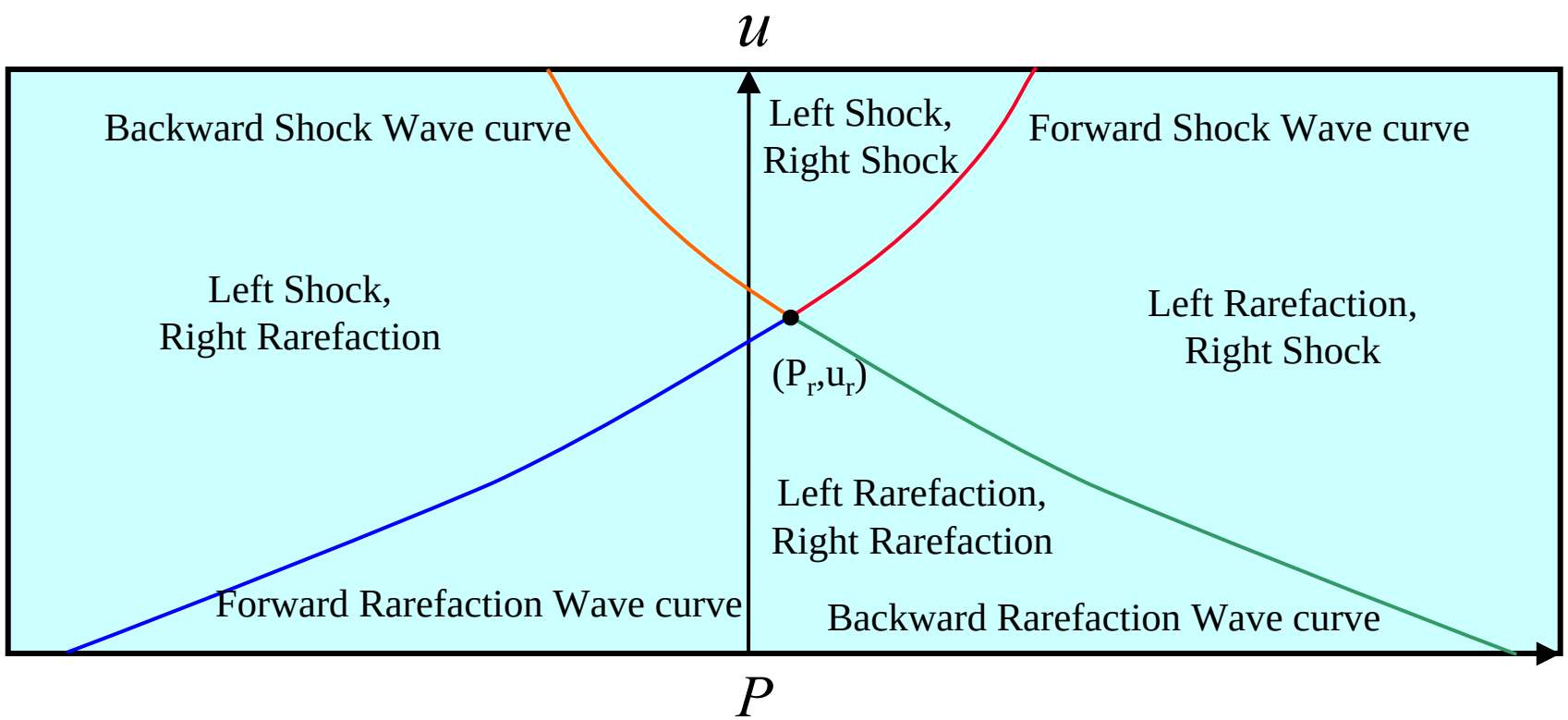
$$P_m = \frac{u_l - u_r + \dfrac{P_l}{m_l} + \dfrac{P_r}{m_r}}{\dfrac{1}{m_l} + \dfrac{1}{m_r}}, \quad u_m = \frac{m_l u_l + m_r u_r + P_l - P_r}{m_l + m_r}.$$

Solution of the Riemann Problem for gas dynamics (2)

The set of equations on the previous slide give the mid state pressure as the solution of a nonlinear algebraic equation. Once the mid state pressure is determined, the mid state velocity is computed from the corresponding formula. Finally using the mid state pressure, the left and right data states, and the Hugoniot or rarefaction equations, as appropriate, we evaluate the densities on either side of the contact discontinuity, and the corresponding waves. If a wave is a shock, its space-time position is a line from the origin with slope equal to the shock speed. If it is a rarefaction, it corresponds to a fan the slope of each ray of which is equal to $u \pm c$ (+ for a forward wave, – for a backward wave). Thus the entire structure of the solution to the Riemann problem is determined once the mid state pressure is found. One exceptional case occurs when the wave curves do not intersect for positive pressure. In this case the solution consists of two rarefaction fans, each of which expands into a vacuum. In this case the mid state velocity is undefined.

Graphical Solution of the Riemann Problem for Gas Dynamics

In conclusion we show that a graphical representation of the Riemann wave curves is useful in understanding the structure of the solution to the Riemann problem. Given the location of the right state, the wave structure is determined by the location of the left state with respect to the wave curve through the right state.



Exercises

1. Derive the quasi-linear form of Euler's equations from the conservation form.
2. Derive the Hugoniot relations for gas dynamics.
3. Verify by direct calculation that the Riemann wave curve for gas dynamics is C^2.

Further Reading

- R. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problem (2nd ed.)*, Interscience Publishers, 1967
- P. D. Lax, *Hyperbolic Systems of Conservation Laws and the Mathematical Theory of Shock Waves*, SIAM 1973.
- J. Smoller, *Shock Waves and Reaction-Diffusion Equations*, Springer-Verlag 1982.
- J. Glimm, *Solutions in the Large for Nonlinear Hyperbolic Systems of Equations*, Comm. Pure Appl. Math., 17, 509-526, 1964.

Finite Difference Methods for Gas Dynamics

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Conservative Finite Difference Methods (1)

Conservative Finite Difference Methods (2)

Conservative Differences in One Space Dimensional Flow

The Order of Accuracy of a Method (1)

The Order of Accuracy of a Method (2)

Stability of a Finite Difference Method

Lax's Equivalence Theorem

The Lax-Wendroff Method (1)

The Lax-Wendroff Method (2)

The Lax-Wendroff Method (3)

The Lax-Wendroff Method (4)

The Lax-Wendroff Method (5)

The Godunov Method

Higher Order Godunov Methods

Extensions to Multiple Space Dimensions

Operator Splitting

Exercises

Further Reading

Conservative Finite Difference Methods (1)

All numerical solutions to systems of partial differential equations are based on the notion of the discretization of the solution. For time dependent problems the basic concept is to use time-marching to advance the solution from one time level to the next, moving from a discrete representation of the solution at one time to another discrete representation of the solution at the next time. Conservation laws are uniquely suited for such a method since the divergence form of the equations gives a natural relation between spatial averages of the solution at two time levels and the space-time averages of the fluxes. Suppose $\Omega(t)$, $t_n \leq t \leq t_{n+1} = t_n + \Delta t_n$ forms a set of smoothly moving regions in space. We define the average of the state vector:

$$\overline{\mathbf{u}}(t) = \frac{\int_{\Omega(t)} d\mathbf{x}\, \mathbf{u}(\mathbf{x},t)}{\int_{\Omega(t)} d\mathbf{x}}$$

Conservative Finite Difference Methods (2)

If we define $\overline{\mathbf{u}}^n = \overline{\mathbf{u}}(t_n)$ then integrating the moving region form of the conservation law:

$$\frac{d}{dt}\int_{\Omega(t)} d\mathbf{x}\, \mathsf{u} + \oint_{\partial\Omega(t)} dA(\mathbf{x})(\mathsf{f}\bullet\mathsf{n} - \mathsf{u}\mathsf{v}\bullet\mathsf{n}) = \int_{\Omega(t)} d\mathbf{x}\, \mathsf{h}\ ,$$

from t_n to t_{n+1} we obtain:

$$\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^n - \left(\frac{\Delta t}{\int_{\Omega_{n+1}} d\mathbf{x}}\right)\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} dt \oint_{\partial\Omega(t)} dA(\mathbf{x})(\mathbf{f}\mathbf{n} - \mathbf{u}\mathbf{v}\bullet\mathbf{n}) + \left(\frac{\int_{t_n}^{t_{n+1}} dt \int_{\Omega(t)} d\mathbf{x}}{\int_{\Omega_{n+1}} d\mathbf{x}}\right)\frac{\int_{t_n}^{t_{n+1}} dt \int_{\Omega(t)} d\mathbf{x}\,\mathbf{h}}{\int_{t_n}^{t_{n+1}} dt \int_{\Omega(t)} d\mathbf{x}} - \left(\frac{\int_{\Omega_{n+1}} d\mathbf{x} - \int_{\Omega_n} d\mathbf{x}}{\int_{\Omega_{n+1}} d\mathbf{x}}\right)\overline{\mathbf{u}}^n .$$

This equation is considerable simpler for a fixed domain, $\Omega(t) = \Omega$:

$$\overline{\mathbf{u}}^{n+1} = \overline{\mathbf{u}}^n - \left(\frac{\Delta t}{vol(\Omega)}\right)\oint_{\partial\Omega} dA(\mathbf{x})\overline{\mathbf{f}}^{n+\frac{1}{2}}\mathbf{n} + \Delta t\, \overline{\mathbf{h}}^{n+\frac{1}{2}} .$$

Here $\overline{\mathbf{f}}^{n+\frac{1}{2}}$ is the average of flux over a space-time ray on $\partial\Omega$ and $\overline{\mathbf{h}}^{n+\frac{1}{2}}$ is the average of $\mathbf{h}$ over the space time region swept out by Ω from t_n to t_{n+1}.

Conservative Differences in One Space Dimensional Flow

The conservative difference equations of the previous slide form the basis of many advanced numerical methods. The basic operation is to construct approximations for the integrals of the flux functions over the space-time boundaries of the region over which the region is averaged, together with an approximation of the integral of the body sources over the space-time volume swept out by this region. For simplicity let us focus on the case of one space dimensional flows with no body forces. Then if we let Ω correspond to a mesh interval $x_{i-\frac{1}{2}} = x_i - \frac{\Delta x_i}{2} \le x \le x_i + \frac{\Delta x_i}{2} = x_{i+\frac{1}{2}}$ the conservative difference equations become:

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{\Delta x_i}\left(\mathbf{f}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{f}_{i-\frac{1}{2}}^{n+\frac{1}{2}}\right).$$

Here $\mathbf{u}_i^n$ is the average of u over the ith interval, and $\mathbf{f}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$ is the average of the flux over the interval $x = x_{i+\frac{1}{2}}$, $t_n \le t \le t_{n+1}$.

The Order of Accuracy of a Method (1)

In order for a numerical method to be effective there must be some notion of convergence of the discrete solution to the exact solution of the differential equation in the limit of vanishing grid size. Convergence is generally not obtained in the pointwise sense (except where the flow is smooth) but is instead obtained in some integral norm such as in L^1 or L^2 if the domain is compact (as is always the case in real computations) on locally in the sense that convergence occurs in the appropriate function space on every compact subset of the domain. Since the exact solution to the system of equations is generally unknown, the exact error in the numerical solution is also unknown. One measure of the appropriateness of a numerical method that can often be computed is the order of accuracy of the method. This is defined as the residual value of the difference equation if an exact solution of the corresponding partial differential equation is substituted into the difference equation. More precisely, we say that the order of accuracy of the numerical method is α if:

$$\frac{\mathbf{u}(x,t+\Delta t)-\ \mathbf{u}(x,t)}{\Delta t}+\frac{\mathbf{f}_{i+\frac{1}{2}}^{n+\frac{1}{2}}-\mathbf{f}_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x}=\mathrm{O}\left(\Delta t^{\alpha}\right).$$

Note here we assume that Δx is of the same order of magnitude as Δt.

The Order of Accuracy of a Method (2)

A positive order of accuracy ensures that the finite difference method is consistent with the corresponding partial differential equation in that, as the mesh is refined, solutions to the partial differential equation solve (to the order of accuracy) the difference equation. To be useful in practice a numerical method needs to be at least first order accurate. Most modern shock capturing numerical methods are second order accurate everywhere except near shocks or contact discontinuities. Also the order of accuracy is generally computed using smooth solutions. If discontinuous solutions are consider as well, the accuracy of the method must also include some notion of locating the discontinuities in the approximately correct location. For example we might require that the integral of the difference equation residual over any finite domain be proportional to the volume of the domain times the appropriate power of Δt.

$$\int_a^b dx \left(\frac{\mathbf{u}(x,t+\Delta t) - \mathbf{u}(x,t)}{\Delta t} + \frac{\mathbf{f}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{f}_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} \right) = (b-a)\,\mathrm{O}\left(\Delta t^{\alpha}\right)$$

This formula ensures that the jump discontinuities are located correctly to within the accuracy of the method.

Stability of a Finite Difference Method

In order for a difference method to be useful it is important that effects like truncation errors (closely related to the order of accuracy of the method) and finite arithmetic round off not dominate the numerical solution. Ideally these errors should stay bounded as the solution operator evolves. More formally, we interpret the finite difference method time stepping in terms of an evolution operator $C(\Delta t)$ so that the solution at each successive time is found by the operation $\mathbf{u}^n = C(\Delta t)\mathbf{u}^{n-1} = \cdots = C(\Delta t)^n\mathbf{u}^0$. For simplicity let us assume that Δt is fixed and $C(\Delta t)$ is linear. Richtmyer expresses the notion of stability as the requirement that as $\Delta t \to 0$ no component of $\mathbf{u}^0$ be amplified by the numerical procedure. Therefore, we say the numerical method is stable, if for a given T there is some $\tau > 0$, so that infinite set of operators:

$$C(\Delta t)^n, \quad \begin{array}{c} 0 < \Delta t < \tau \\ 0 \le n\Delta t \le T \end{array}$$

is uniformly bounded. Note that stability is a property of the numerical method, not the partial differential equation.

Lax's Equivalence Theorem

One of the few general theorems on the convergence of finite difference methods is due to Lax and states that for a consistent scheme (positive order of accuracy) stability is a necessary and sufficient condition for convergence. More precisely we have:

Suppose that the initial value problem for the linear differential equation,

$$\frac{d}{dt}u(t) = Au(t),\ 0 \le t \le T$$

$$u(0) = u^0,$$

where A is a linear operator on a Banach space $\mathcal{B}$ *and* u^0 *is a given element of* $\mathcal{B}$ *is well posed. Then for a finite difference approximation that satisfies the consistency condition, stability is the necessary and sufficient condition for convergence.*

For our purposes $\mathcal{B}$ is a function space such as L^2 and A is a differential operator on the function space. See Richtmyer and Morton, "Difference methods for initial value problem" for a proof of the Lax Equivalence Theorem.

The Lax-Wendroff Method (1)

The notion of order of accuracy of a numerical method can be illustrated by deriving the Lax-Wendroff method for the system of conservation laws,

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} = 0.$$

This method is also very useful in practice. The basic idea is to expand $\mathbf{u}(x,t)$ in a Taylor series to second order in time for fixed x, use the partial differential equation to replace the time derivatives with spatial derivatives, and the use central differences to approximate the resulting spatial derivative to second order. The resulting finite difference equation is then by construction second order accurate. Let $\mathbf{A} = d\mathbf{f}/d\mathbf{u}$ be the Jacobi matrix for the flux function. Then we can expand $\mathbf{u}(x,t+\Delta t)$ as follows:

The Lax-Wendroff Method (2)

$$\mathbf{u}(x,t+\Delta t)=\mathbf{u}(x,t)+\frac{\partial \mathbf{u}}{\partial t}(x,t)\Delta t+\frac{\partial^2 \mathbf{u}}{\partial t^2}(x,t)\frac{\Delta t^2}{2}+\mathrm{O}\left(\Delta t^3\right)$$

$$\mathbf{u}(x,t+\Delta t)=\mathbf{u}(x,t)-\frac{\partial \mathbf{f}(\mathbf{u})}{\partial x}(x,t)\Delta t-\frac{\partial}{\partial x}\left(\frac{\partial \mathbf{f}(\mathbf{u})}{\partial t}\right)(x,t)\frac{\Delta t^2}{2}+\mathrm{O}\left(\Delta t^3\right)$$

$$\mathbf{u}(x,t+\Delta t)=\mathbf{u}(x,t)-\frac{\partial \mathbf{f}(\mathbf{u})}{\partial x}(x,t)\Delta t-\frac{\partial}{\partial x}\left(\mathbf{A}\frac{\partial \mathbf{u}}{\partial t}\right)(x,t)\frac{\Delta t^2}{2}+\mathrm{O}\left(\Delta t^3\right)$$

$$\mathbf{u}(x,t+\Delta t)=\mathbf{u}(x,t)-\frac{\partial \mathbf{f}(\mathbf{u})}{\partial x}(x,t)\Delta t+\frac{\partial}{\partial x}\left(\mathbf{A}\frac{\partial \mathbf{f}(\mathbf{u})}{\partial x}\right)(x,t)\frac{\Delta t^2}{2}+\mathrm{O}\left(\Delta t^3\right)$$

$$\mathbf{u}(x,t+\Delta t)=\mathbf{u}(x,t)-\frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathbf{f}(\mathbf{u})(x+\Delta x,t)-\mathbf{f}(\mathbf{u})(x-\Delta x,t)\right)+$$

$$\frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2\left[\mathbf{A}(\mathbf{u})\left(x+\frac{\Delta x}{2},t\right)\left(\mathbf{f}(\mathbf{u})(x+\Delta x,t)-\mathbf{f}(\mathbf{u})(x,t)\right)-\mathbf{A}(\mathbf{u})\left(x-\frac{\Delta x}{2},t\right)\left(\mathbf{f}(\mathbf{u})(x,t)-\mathbf{f}(\mathbf{u})(x-\Delta x,t)\right)\right]+\mathrm{O}\left(\Delta t^3\right)$$

If we let $x = i\Delta x$, $t = n\Delta t$, we obtain the difference method:

$$\mathbf{u}_i^{n+1}=\mathbf{u}_i^n-\frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathbf{f}_{i+1}^n-\mathbf{f}_{i-1}^n\right)+\frac{1}{2}\left(\frac{\Delta t}{\Delta x}\right)^2\left[\mathbf{A}_{i+1/2}^n\left(\mathbf{f}_{i+1}^n-\mathbf{f}_i^n\right)-\mathbf{A}_{i-1/2}^n\left(\mathbf{f}_i^n-\mathbf{f}_{i-1}^n\right)\right]$$

The Lax-Wendroff Method (3)

$$\mathbf{f}_{i+1/2}^{n+1/2} = \frac{\mathbf{f}_{i+1}^{n} + \mathbf{f}_{i}^{n}}{2} - \frac{1}{2}\frac{\Delta t}{\Delta x}\mathbf{A}_{i+1/2}^{n}\left(\mathbf{f}_{i+1}^{n} - \mathbf{f}_{i}^{n}\right),$$

and we let $\mathbf{A}_{i+1/2}^{n} = \mathbf{A}\left(\left(u_{i+1}^{n} + u_{i}^{n}\right)/2\right)$. In practice the Lax-Wendroff method is implemented as a two step method that is identical with the above formula for linear fluxes (constant **A**), and is also second order in general.

$$\mathbf{u}_{i+1/2}^{n+1/2} = \frac{\mathbf{u}_{i+1}^{n} + \mathbf{u}_{i}^{n}}{2} - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathbf{f}_{i+1}^{n} - \mathbf{f}_{i}^{n}\right)$$

$$\mathbf{f}_{i+1/2}^{n+1/2} = \mathbf{f}\left(\mathbf{u}_{i+1/2}^{n+1/2}\right)$$

$$\mathbf{u}_{i}^{n+1} = \mathbf{u}_{i}^{n} - \frac{\Delta t}{\Delta x}\left(\mathbf{f}_{i+1/2}^{n+1/2} - \mathbf{f}_{i-1/2}^{n+1/2}\right)$$

It can be shown (see Richtmyer and Morton) that for constant **A** the Lax-Wendroff method is stable provided $|\lambda|\Delta t/\Delta x < 1$ for all eigenvalues of **A**. For gas dynamics this condition becomes $(|u|+c)\ \Delta t/\Delta x < 1$ for all velocities u and sound speed c.

The Lax-Wendroff Method (4)

One further modification to the Lax-Wendroff is needed before it can be used as an effective finite difference method for gas dynamics. In the presence of discontinuities this method generates severe oscillations that can destroy the integrity of the computation. These oscillations can be damped by the addition of artificial viscosity to the numerical method. It is beyond the scope of this course to discuss the analysis of artificial viscosity. Listed below are two modifications to the Lax-Wendroff fluxes that have proven to be useful in practice.

Linear artificial viscosity: We modify the Lax-Wendroff fluxes by adding a term that mimics the diffusive term $\varepsilon \mathbf{u}_{xx}$ added to the right hand side of the conservation law.

$$\mathbf{f}_{i+1/2}^{n+1/2} = \mathbf{f}_{i+1/2}^{n+1/2}\Big|_{\text{Lax-Wendroff}} + a\left(\mathbf{u}_{i+1}^{n} - \mathbf{u}_{i-1}^{n}\right)$$

The artificial viscosity coefficient a might be taken of the form $a = \chi s$ where χ is a numerical parameter and s is the maximum wave speed at the point being updated.

The Lax-Wendroff Method (5)

Lapidus artificial viscosity: For gas dynamics a more adaptive method of artificial viscosity that increases the artificial viscosity in regions of large gradient while reducing it in smooth regions was proposed by Lapidus (see Richtmyer and Morton). Let the coefficients $g^k_{i+1/2}$, k = 0, 1, 2. be chosen by the interpolation formulas:

$$g^0_{i+1/2} + g^1_{i+1/2}\left(u^n_{i+1/2} + c^n_{i+1/2}\right) + g^2_{i+1/2}\left(u^n_{i+1/2} + c^n_{i+1/2}\right)^2 = \chi\left|\left(u^n_{i+1} + c^n_{i+1}\right) - \left(u^n_i + c^n_i\right)\right|$$

$$g^0_{i+1/2} + g^1_{i+1/2}\left(u^n_{i+1/2}\right) + g^2_{i+1/2}\left(u^n_{i+1/2}\right)^2 = \chi\left|u^n_{i+1} - u^n_i\right|$$

$$g^0_{i+1/2} + g^1_{i+1/2}\left(u^n_{i+1/2} - c^n_{i+1/2}\right) + g^2_{i+1/2}\left(u^n_{i+1/2} - c^n_{i+1/2}\right)^2 = \chi\left|\left(u^n_{i+1} - c^n_{i+1}\right) - \left(u^n_i - c^n_i\right)\right|.$$

Then the modified Lapidus artificial viscosity Lax-Wendroff method is:

$$\mathbf{u}^{n+1/2}_{i+1/2} = \frac{\mathbf{u}^n_{i+1} + \mathbf{u}^n_i}{2} - \frac{1}{2}\left(\frac{\Delta t}{\Delta x} + g^2_{i+1/2}\right)\left(\mathbf{f}^n_{i+1} - \mathbf{f}^n_i\right) - \frac{1}{2} g^1_{i+1/2}\left(\mathbf{u}^n_{i+1} - \mathbf{u}^n_i\right)$$

$$\mathbf{f}^{n+1/2}_{i+1/2} = \mathbf{f}\left(\mathbf{u}^{n+1/2}_{i+1/2}\right) - \frac{1}{2} g^0_{i+1/2}\left(\mathbf{u}^n_{i+1} - \mathbf{u}^n_i\right)$$

$$\mathbf{u}^{n+1}_i = \mathbf{u}^n_i - \frac{\Delta t}{\Delta x}\left(\mathbf{f}^{n+1/2}_{i+1/2} - \mathbf{f}^{n+1/2}_{i-1/2}\right).$$

Remark: For this method the stability condition is modified:

$$\left(|u| + c\right)\frac{\Delta t}{\Delta x} < \left(1 + \frac{\chi^2}{4}\right)^{1/2} - \frac{\chi}{2}$$

The Godunov Method

Godunov's method computes the numerical fluxes using the solution to a Riemann problem. The value of the numerical flux is given by the formula:

$$\mathbf{f}_{i+1/2}^{n+1/2} = \mathbf{f}\left(\mathbf{u}_{i+1/2}^{n+1/2}\right),$$

Where $\mathbf{u}_{i+1/2}^{n+1/2}$ is the solution evaluated along the time axis of the Riemann problem with data $\mathbf{u}_l = \mathbf{u}_i$ and $\mathbf{u}_r = \mathbf{u}_{i+1}$. This method has the advantage of producing smooth and stable shock profiles. Its disadvantages are that it is only first order accurate and is quite diffusive. In practice the Godunov method is used as a component in a higher order method in which the numerical method adapts between a high order method for smooth flow regions and the Godunov method in regions with large gradients in the flow variables. Stability requires that the solutions to the Riemann problems at the cell edges not interact during the time step. Thus the stability condition becomes:

$$\max\left\{\left|shock\ speeds\right|, \left|u\right| + c : all\ Riemann\ problem\ solutions\right\}\frac{\Delta t}{\Delta x} < 1$$

Higher Order Godunov Methods

Most modern compressible hydrodynamics use some version of a higher order extension of the Godunov method. It is beyond the scope of this course to go into these methods in detail. These higher order Godunov methods are all outgrowths of the original method of van Leer. Some of the basic ideas in these higher order extensions of Godunov's method include:

- Reconstruction: Given a piecewise constant function construct a higher order interpolant whose cell averages equal the piecewise constant values.
- Limiting: Modify the high order interpolant to eliminate or moderate overshoot at cell edges.
- Riemann solvers: Construct approximate solutions to the Riemann problems between states at the cell edges as determined from the interpolant.
- Method of Characteristics: Use the smooth form of the partial differential equations to compute high order updates of a spatially variable flow. (The method of characteristics will be discussed in the next lecture.

Extensions to Multiple Space Dimensions

All of the difference methods discussed in this lecture have extensions (often quite complicated) to systems with more than one space dimension. Current computer technology makes high resolution computations possible for two space dimensional flows on moderately priced computers. Currently three dimensional flows require expensive supercomputers for even moderate resolution.

One standard technique for constructing multiple space dimensional solvers from one space dimensional solvers is operator splitting. This method has proven to be quite effective in gas dynamics. The basic idea is to solve one dimensional problems by freezing the flow in all directions except one, and then use the solution from each successive one dimensional problem as the data for the next sweep. We can illustrate this more easily by example in two space dimensions. Suppose we have the system:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y} = 0$$

Consider the two one dimensional systems, where we treat on space variable as a parameter:

$$\frac{\partial \mathbf{u}}{\partial t}(x,y,t) + \frac{\partial \mathbf{f}(\mathbf{u})}{\partial x}(x,y,t) = 0 \qquad \frac{\partial \mathbf{u}}{\partial t}(x,y,t) + \frac{\partial \mathbf{g}(\mathbf{u})}{\partial y}(x,y,t) = 0$$

Operator Splitting

If we treat y as a parameter in updating the x operator split equation and x as a parameter in the y equation, we obtain two one dimensional systems of equations. Suppose that $\mathbf{u}_{ij}^{n}$ is a discrete representation of the solution. Let $\mathbf{u}_{\cdot j}^{n}$ and $\mathbf{u}_{i\cdot}^{n}$ denote the row and column vectors of the solution respectively, and let L_x and L_y be discrete update operators for the two respective operator split equations. Then we update the solution a follows.

$$\hat{\mathbf{u}}_{\cdot j}^{2n+1} = L_x \mathbf{u}_{\cdot j}^{2n} \qquad \hat{\mathbf{u}}_{i\cdot}^{2n+2} = L_y \mathbf{u}_{i\cdot}^{2n+1}$$

$$\mathbf{u}_{i\cdot}^{2n+1} = L_y \hat{\mathbf{u}}_{i\cdot}^{2n+1} \qquad \mathbf{u}_{\cdot j}^{2n+2} = L_x \hat{\mathbf{u}}_{\cdot j}^{2n+1}$$

Omitting the spatial subscripts we see that an update from an even time step to the next odd time step consists of an x sweep following by a y sweep, while odd steps are updated by a y sweep followed by an x sweep.

$$\mathbf{u}^{2n+1} = L_y L_x \mathbf{u}^{2n}$$

$$\mathbf{u}^{2n+2} = L_x L_y \mathbf{u}^{2n+1}$$

The technique of alternating the order of the sweeps is called Strang splitting and increases the order of accuracy of the 2D update from first to second order assuming the one dimensional methods are at least second order accurate.

Exercises

1. Show that for the scalar advection equation $u_t + a\, u_x = 0$, $a > 0$, the Godunov method is first order accurate for smooth solutions. (Hint: the general solution of this equation is $u(x,t) = f(x-at)$ for an arbitrary function $f(x)$.)

2. Show that the Lax-Wendroff method for the scalar advection equation corresponds to a finite central difference expansion of the equation $u_t + a\, u_x = a^2/2\, \Delta t\, u_{xx}$. This illustrates the presence of numerical viscosity in the Lax-Wendroff method. One goal of artificial viscosity is to cancel this viscous term.

3. Suppose that $u^n(x)$ is periodic on the interval $[-\pi,\pi]$. Define $u^{n+1}(x)$ by the Lax-Wendroff update of $u^n(x)$ for the scalar advection equation:

$$u^{n+1}(x) = u^n(x) - \frac{1}{2}\left(a\frac{\Delta t}{\Delta x}\right)\left(u^n(x+\Delta x) - u^n(x-\Delta x)\right) + \frac{1}{2}\left(a\frac{\Delta t}{\Delta x}\right)^2\left(u^n(x+\Delta x) - 2u^n(x) + u^n(x-\Delta x)\right)$$

If $\hat{u}^n(k)$ are the Fourier coefficients of $u^n(x)$ show that:

$$\hat{u}^{n+1}(k) = g\left(a\frac{\Delta t}{\Delta x}, k\Delta x\right)\hat{u}^n(k)$$

$$g(\alpha,\beta) = 1 - i\alpha\sin\beta - \alpha^2(1-\cos\beta)$$

The quantity g is called the amplification matrix (here 1x1) of the difference method. Show that the stability condition $a\Delta t/\Delta x < 1$ is equivalent to the requirement that $g(a\Delta t/\Delta x,\beta)$ lies inside the unit circle except at $\beta = 0$.

Further Reading

- R. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problem (2nd ed.)*, Interscience Publishers, 1967
- R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser, 1990.
- E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag, 1997.
- E. Godlewski, P.-A. Raviart, Numerical Approximation of Systems of Hyperbolic Conservation Laws, Springer-Verlag, 1996.
- See the references in the books of Toro, or Godlewski and Raviart for references to the original papers of van Leer, Colella and Woodward, and Harten for more information on higher order Godunov methods

Front Tracking in One Space Dimension

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Multi-Material Flows

Numerical Solution of Multi-Material Flows (1)

Numerical Solution of Multi-Material Flows (2)

Numerical Solution of Multi-Material Flows (3)

Numerical Solution of Multi-Material Flows (4)

Front Tracking Representation of a Flow (1)

Front Tracking Representation of a Flow (2)

Propagating the Front Tracking Solution

The Method of Characteristics (1)

The Method of Characteristics (2)

Contact Discontinuity Propagation (1) (Method of Characteristics Version)

Contact Discontinuity Propagation (2) (Method of Characteristics Version)

Contact Discontinuity Propagation (3) (Method of Characteristics Version)

Contact Discontinuity Propagation (4) (Method of Characteristics Version)

Contact Discontinuity Propagation (5) (Method of Characteristics Version)

Contact Discontinuity Propagation (Riemann Problem Version)

Update of Interior States Near Fronts

Exercises

Further Reading

Multi-Material Flows

Flows with two or more materials are very important in applications and create a number of difficulties for numerical simulation. Examples of such flows include Rayleigh-Taylor and Richtmyer-Meshkov instability in which one fluid is accelerated into another by either a body force or shock wave respectively. These instabilities occur in such applications as inertial confinement fusion and supernova explosions. At an opposite end of the spectrum are diffusive type flows such as the motion of a hot fluid in a cold fluid, or the diffusion of a dye. These two extremes illustrate a general division of flow regimes, non-diffusion regimes where the interface between the two materials remains sharp, and diffusive regimes where the materials mix at the molecular level. Of course in reality all flows contain some mixture of the two regimes, but in practice the choice of the relevant length and time scales of interest dictate the appropriate flow regime. Compressible flows are usually associated with time scales that are short relative to the rate of molecular diffusion and so are often in the non-diffusive regime. Indeed the applications such as inertial confinement fusion and supernova explosions, the time scales are on the order of nanoseconds and are as much 6-9 orders of magnitude smaller than the time scale for significant molecular mixing. In this course we will focus on the sharp interface flow regime, and will discuss numerical methods that attempt to maintain relatively sharp material interfaces.

Numerical Solution of Multi-Material Flows (1)

Numerical methods for the solution of a multi-material flow are generally one of several types, although it is possible to develop algorithms that combine features of any or all of these methods.

- **Mixed cell methods** in which the fractional volume and fractional mass of each species is computed and the material is assumed to be well mixed at the cell level. Such methods are extremely diffusive, but when combined with adaptive mesh refinement that keeps the size of the mixed cells small this method can be quite effective. One important question that arises in mixed cell methods is how to specify the thermodynamic properties of the mixed material. If the mixing is molecular one generally has that the multiple materials are in thermal and mechanical equilibrium (equal temperature and pressure). This provides a method for computing an effective multi-material equation of state for the mixture. However for the applications considered in this course the mixing is very far from molecular and the question of an appropriate mixed material equation of state is quite open and indeed is problem dependent.

Numerical Solution of Multi-Material Flows (2)

- **Volume of fluid and interface reconstruction methods**. Like the mixed cell method, fractional volumes and masses of the individual species are computed. However this information is used to reconstruct interfaces between the materials so that they become effectively demixed at a subcell level. These interfaces are then used to compute the numerical fluxes at the cell boundaries to that mass for a given species can only flow into and out of the boundary through regions occupied by that material. This method eliminates the need for mixed material equations of state since the interfaces separate the materials into separate regions inside of which each material is pure. At the end of each time step the interfaces are discarded and recomputed during the next time step using only the fractional volume and mass information. Care is taken during the reconstruction so that the interfaces created in each cell are as consistent as possible with those in the next cell. This method is much less diffusive than the mixed cell method but is still much more diffusive than the real physical process.

Numerical Solution of Multi-Material Flows (3)

- **Level set methods**. In this method the interface between separate materials is modeled as the level set $t = \phi(\mathbf{x})$, and the numerical calculates seeks first to compute the level set as a solution to a Hamilton-Jacobi equation and then to update the physical flow variables using the computed interface. A nice feature of this method is that interface interactions are handled automatically. A major problem is that the level set function is not a true physical variable and as such the interface interactions may be handled in an unphysical manner. Depending on the implementation, the equation of state may be misapplied to materials due to the artificial location of the interface. Also this method has difficulty in treating triple points where three or more material meet at a point. Another problem is the topological restriction that the interface be the level set of a global function. This problem can be address by using a local level set approach in which the level set function only applies to a local region. Most of these difficulties are being addressed in current implementations of level set methods by applying front tracking type approaches to the algorithm.

Numerical Solution of Multi-Material Flows (4)

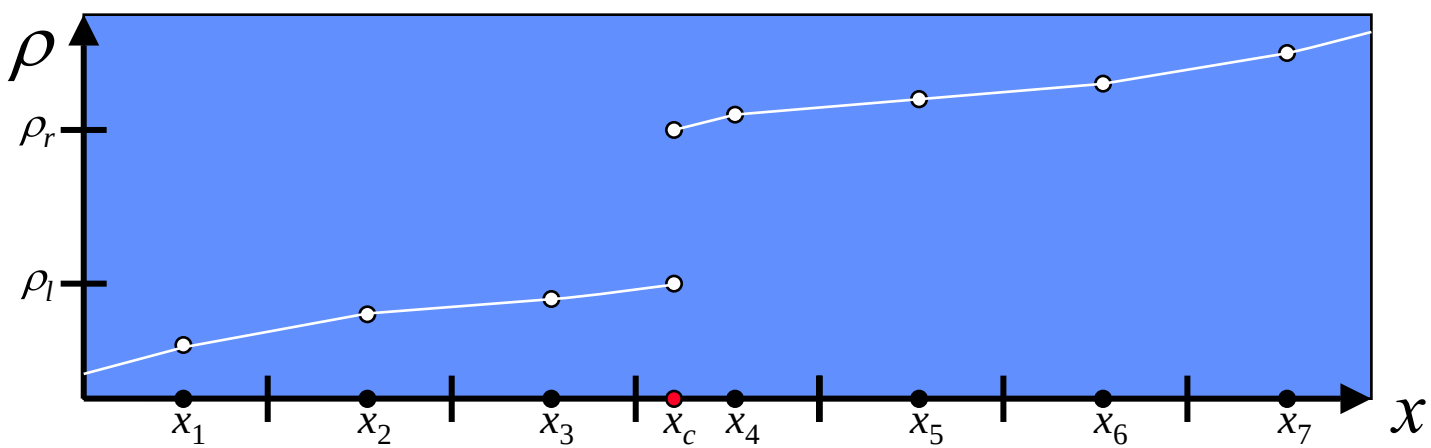
- **Front Tracking**. The numerical method that comes closest to modeling the physical behavior of a non-diffusive flow is front tracking. In this method the material interfaces become part of the numerical representation of the flow. These fronts are propagated using the mathematically correct Rankine-Hugoniot conditions at the front together with a method of characteristics (see the next slide) computation for coupling the front states to the flow adjacent to the front. For the update of the non-front (cell centered) states, the fronts provide moving space-time boundary conditions. The problem of mixed material equations of state does not occur since the fronts provide a sharp division between pure material regions. The physically correct balance conditions are maintained at material fronts so that the pressure is continuous (in the absence of surface tension) at the front, but temperature is maintained independently in the adjacent two materials. (Recall that the Euler equations models adiabatic flow.) Mathematically, front tracking eliminates numerical diffusion at the interfaces since the flow variables are allowed to jump discontinuously across tracked fronts. The remainder of this lecture will discuss front tracking algorithms for one dimensional flows. The following lecture will discuss extensions of this method to higher dimensional flows.

Front Tracking Representation of a Flow (1)

For simplicity we consider a one space dimensional gas dynamical flow with a single material discontinuity. Let the two materials be labeled by the indices l and r, where l indicated the material to the left of the material boundary and r indicates the material to the right. The flow is modeled by the Euler equations in which it is understood that at any position in space only one of the two materials is present. The two materials may have separate equations of state. Again for simplicity let us assume both materials are perfect gases with gammas and specific heats γ_i and c_v^i, i = l,r that are possibly different for the two materials. Suppose the position of the material interface is given by $x_c(t)$. The discrete representation of the flow consists of a set of cell centers and mesh sizes, x_i, Δx_i $i = 1,\ldots,N$, where $x_i + \Delta x_i/2 = x_{i+1} - \Delta x_{i+1}/2$, together with a density ρ_i, a momentum density $m_i = \rho_i u_i$, and a total energy density $E_i = \rho_i(\frac{1}{2}u_i^2 + e_i)$ associated with each cell. Except for the cell containing the tracked point $x_c(t)$ these values can be regarded as the cell averages of the corresponding flow variable over the indicated cell. Quantities derived from these primitive flow variables will also be indicated by the appropriate index. Associated with the tracked point $x_c(t)$ we have two sets of flow variables, ρ_l, m_l, and E_l corresponding to the flow on the left of $x_c(t)$, and ρ_r, m_r, and E_r corresponding to the flow on the right of this point.

Front Tracking Representation of a Flow (2)

The states at the front can be interpreted as the limit of the flow as the discontinuity front is approached from the appropriate side. The state associated with the cell center in the grid block containing the front is not the average of the flow across that cell, since such an interpretation would involve smearing the flow variables across the front, rather it should be regarded as a quantity associated with the side of the front on which the cell center lies. For example this value might be thought of as an approximation of the pointwise value of the flow variables at the given position. For simplicity we assume the mesh size $\Delta x_i = \Delta x$ is constant, so that $x_{i+1} = x_i + \Delta x$. The diagram below shows a schematic representation of the density.



Propagating the Front Tracking Solution

The time step propagation consists of advancing the flow states at each cell center, computing the new position of the tracked front and updating the left and right states at the tracked points. As implemented in the front tracking code *FronTier*, this consists of the following steps.

- Use Riemann problem solutions and method of characteristics computations to compute the time updated position and states at the propagated fronts.
- Use finite differences to time advance the interior states using the old and new front states and positions (i.e. the states and positions of the fronts at the beginning and end of the time step) as interior boundary conditions for the cell centered states. The critical operation here is to avoid taking finite differences across a tracked front, so that all difference quantities are taken from the same side of a discontinuity.

The Method of Characteristics (1)

The method of characteristics has now been mentioned at least twice without any explanation. Let is take a minute to review this important concept. Recall that for smooth flow the Euler equations can be written:

$$\frac{DP}{Dt} + \rho c^2 u_x = 0$$

$$\rho \frac{Du}{Dt} + P_x = 0$$

$$\frac{DS}{Dt} = 0$$

Adding and subtracting the first equation and c times the second we obtain:

$$\left(\frac{D}{Dt} + c\frac{\partial}{\partial x}\right)P + \rho c\left(\frac{D}{Dt} + c\frac{\partial}{\partial x}\right)u = 0$$

$$\left(\frac{D}{Dt} - c\frac{\partial}{\partial x}\right)P - \rho c\left(\frac{D}{Dt} - c\frac{\partial}{\partial x}\right)u = 0$$

$$\frac{DS}{Dt} = 0$$

or

$$\left(\frac{\partial}{\partial t} + (u+c)\frac{\partial}{\partial x}\right)P + \rho c\left(\frac{\partial}{\partial t} + (u+c)\frac{\partial}{\partial x}\right)u = 0$$

$$\left(\frac{\partial}{\partial t} + u\frac{\partial}{\partial x}\right)S = 0$$

$$\left(\frac{\partial}{\partial t} + (u-c)\frac{\partial}{\partial x}\right)P - \rho c\left(\frac{\partial}{\partial t} + (u-c)\frac{\partial}{\partial x}\right)u = 0$$

The Method of Characteristics (2)

The important feature of the characteristic equations is that all of the derivatives in each equation appear in the same directional form. More precisely we define the characteristic curves through each point in our domain by the formulas:

$$\frac{d}{dt}x_f(t,x_0,t_0)=u+c, \quad x_f(t_0,x_0,t_0)=x_0$$

$$\frac{d}{dt}x_p(t,x_0,t_0)=u, \quad x_f(t_0,x_0,t_0)=x_0$$

$$\frac{d}{dt}x_b(t,x_0,t_0)=u-c, \quad x_f(t_0,x_0,t_0)=x_0.$$

These three curves are referred to as the forward, particle, and backward characteristics through the given point. From the characteristic form of the Euler equations we see that the evolution of the state variables along the characteristics are related by the formulas:

$$\frac{dP}{dt}+\rho c\frac{du}{dt}=0, \quad \text{along} \quad \frac{dx}{dt}=u+c$$

$$\frac{dS}{dt}=0, \quad \text{along} \quad \frac{dx}{dt}=u$$

$$\frac{dP}{dt}-\rho c\frac{du}{dt}=0, \quad \text{along} \quad \frac{dx}{dt}=u-c$$

Contact Discontinuity Propagation (1)

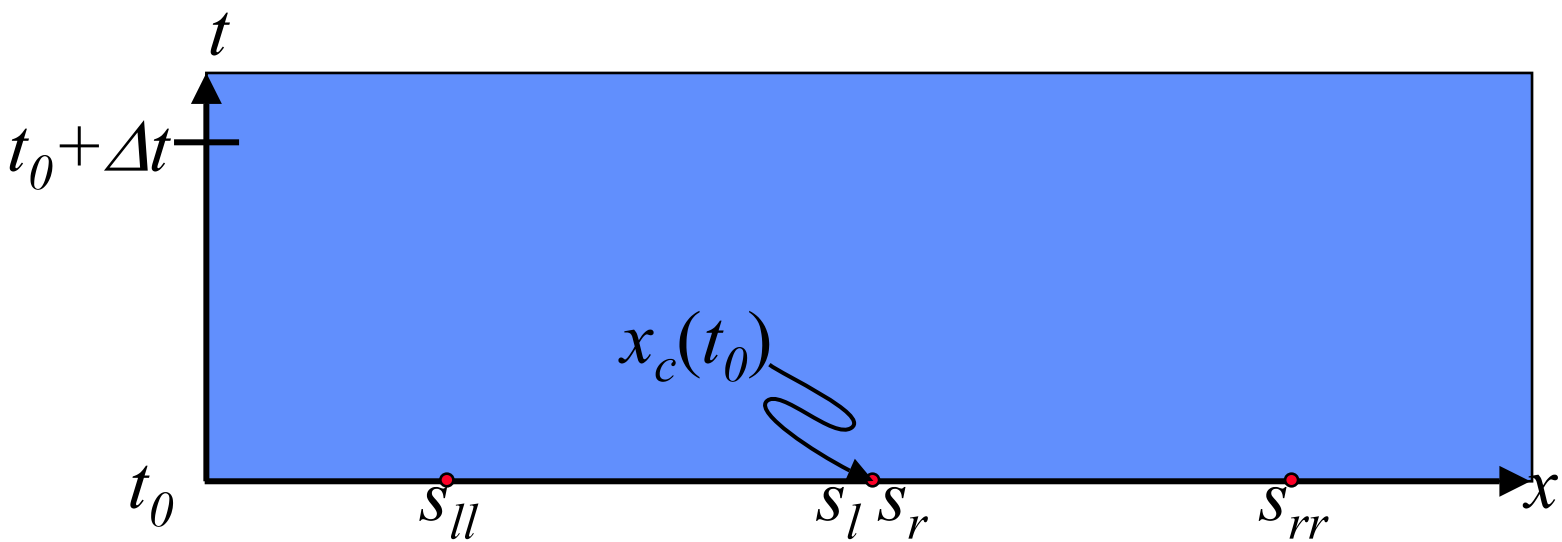
(Method of Characteristics Version)

The algorithm for propagating a contact discontinuity point (most commonly corresponding to a material interface) uses what is called a three point wave speed calculation. The data for the propagation consists of the position of the contact $x_c(t)$, the states on the left and right of wave, s_l and s_r, and two states s_{ll} and s_{rr} obtained by sampling the solution a distance Δx to the left and right of the contact point. Here by a "state" we mean the flow variables describing the flow at a given location, so for example s_l represents the density, velocity, and internal energy (and any other derived flow variable) at the left side of the contact. The off front states s_{ll} and s_{rr} are obtained by sampling a reconstructed interpolant of the solution. One possible choice for such an interpolant is the piecewise linear curve obtained by connecting the cell centered data and the front data with line segments, while allowing a jump in the variables across tracked fronts. The graph on slide 7 shows such an interpolant for density. Generally we only interpolate a minimal set of flow variables (such as density, momentum density, and total energy density or density, velocity, and specific internal energy) and compute the other flow variables using their formulas in terms of the minimal set.

Contact Discontinuity Propagation (2)

(Method of Characteristics Version)

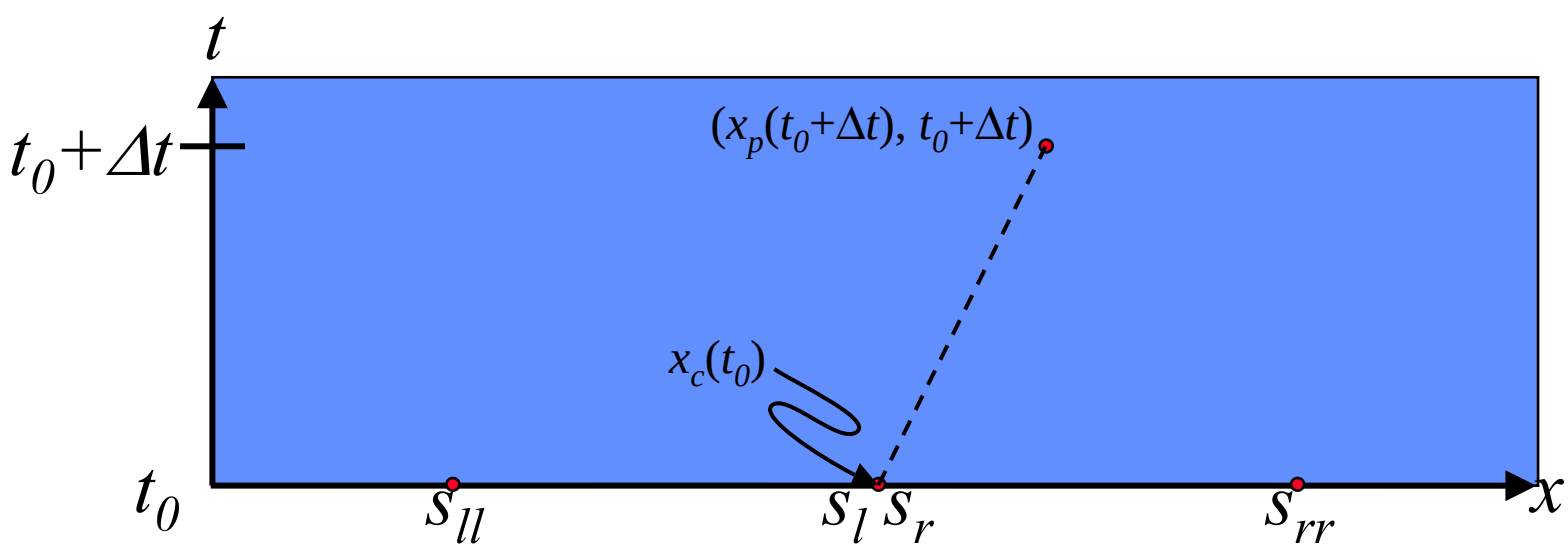
The chart below illustrates the choice of initial data.



Contact Discontinuity Propagation (3)

(Method of Characteristics Version)

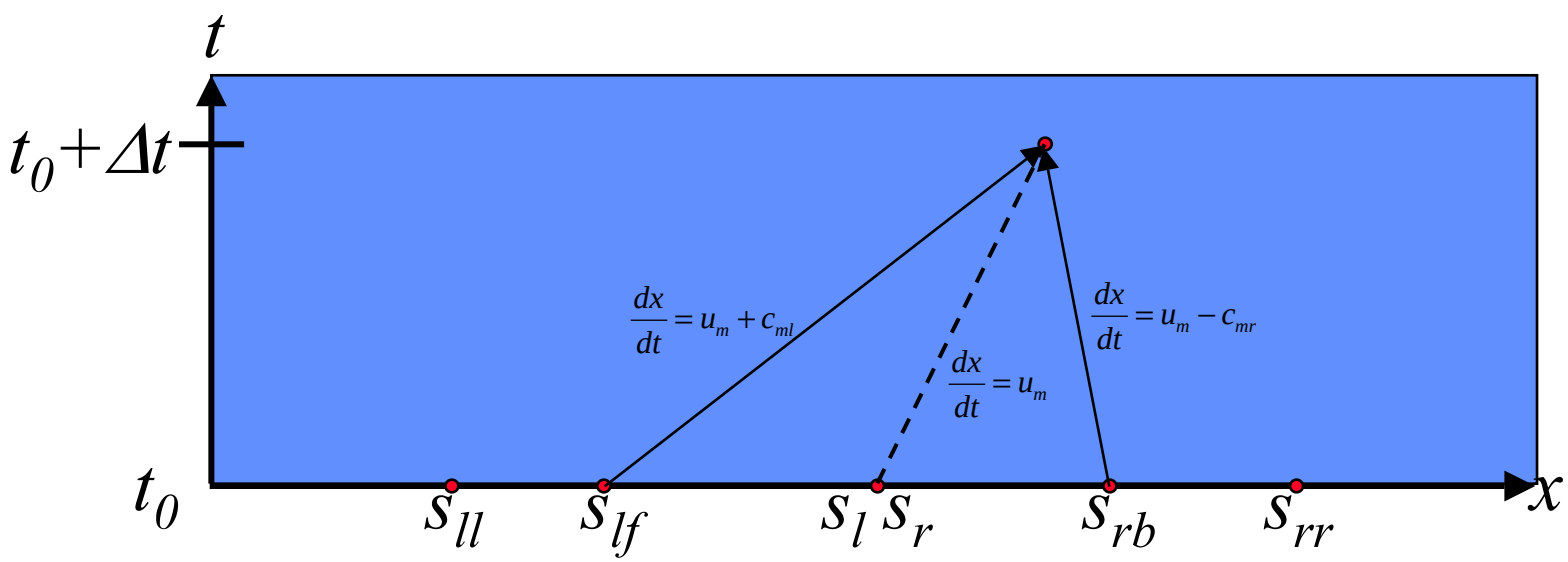
The first step in the propagation algorithm is to solve a Riemann problem with data s_l and s_r. The mid state speed u_m from the solution is used as a predictor for the updated contact position, $x_p(t_0+\Delta t) \approx x_c(t_0) + u_m \Delta t$.



Contact Discontinuity Propagation (4)

(Method of Characteristics Version)

Using the forward characteristic speed on the left side of the interface and the backward characteristic speed on the right, trace back a linear approximation to the incoming characteristics at the propagated point from time $t_0+\Delta t$ back to time t_0, obtaining positions $x_{lf}(t_0) = x_c(t_0) - c_{ml}\Delta t$, and $x_{rb}(t_0) = x_c(t_0) + c_{mr}\Delta t$ together with interpolated state values s_{lf} and s_{rb} at these locations.



Contact Discontinuity Propagation (5)

(Method of Characteristics Version)

Solve for the updated states on both sides of the contact using an implicit Euler integration:

$$u^+ - u_{lf} + \frac{1}{2}\left(\frac{1}{\rho_{lf} c_{lf}} + \frac{1}{\rho_l^+ c_l^+}\right)\left(P^+ - P_{lf}\right) = 0, \quad u_{rb} - u^+ + \frac{1}{2}\left(\frac{1}{\rho_{rb} c_{rb}} + \frac{1}{\rho_r^+ c_r^+}\right)\left(P^+ - P_{lf}\right) = 0$$

$$e_l^+ - e_l + \frac{1}{2}\left(P^+ + P_0\right)\left(\frac{1}{\rho_l^+} - \frac{1}{\rho_l}\right) = 0, \qquad e_r^+ - e_r + \frac{1}{2}\left(P^+ + P_0\right)\left(\frac{1}{\rho_r^+} - \frac{1}{\rho_r}\right) = 0$$

$$P^+ = P_l\left(e_l^+, \rho_l^+\right), \quad c_l^+ = c_l\left(e_l^+, \rho_l^+\right), \qquad P^+ = P_r\left(e_r^+, \rho_r^+\right), \quad c_r^+ = c_r\left(e_r^+, \rho_r^+\right).$$

Note that we use the separate equations of state on either side of the interface to relate the pressures, densities, specific internal energies, and sound speeds on either side of the interface. Finally we use central differences in time to determine the net interface speed and updated contact position.

$$v_{interface} = \frac{1}{2}\left(u^+ + u_m\right), \quad x_c\left(t_0 + \Delta t\right) = x_c\left(t_0\right) + v_{interface}\Delta t.$$

Contact Discontinuity Propagation

(Riemann Problem Version)

The algorithm just described makes an implicit assumption that the main discontinuity is at the tracked contact and that all other waves, either incoming or outgoing are relatively weak. This assumption is used to validate the use of the method of characteristics on either side of the interface, since this method is based on the smooth flow Euler equations. An alternative algorithm that replaces the method of characteristics calculations with the solution of Riemann problems has proven to be more useful in practice, especially for flows with strong captured waves. This variation proceeds identically to the previous version up until the point where the two states s_{lf} and s_{rb} have been found at the feet of the traced backed characteristics from the predicted new position of the contact. Then instead of applying the method of characteristics we solve a Riemann problem with data s_{lf} and s_{rb} and use the mid state solutions from this Riemann problem to replace the method of characteristic solutions found from the equations on the previous slide. The remainder of the algorithm proceeds as before. Variations of this algorithm may replace s_{lf} by s_{ll} and s_{rb} by s_{rr} to provide more upwinding for the incoming waves.

Update of Interior States Near Fronts

The final step in the time step update of the solution is to use finite differences to advance the cell centered states. For cells whose discrete domain of dependency stencil (i.e. the set of states used in the finite difference formula to update the solution at a given position) does not overlap the fronts, this is done using standard finite differences as described in the previous lecture. At points whose stencil overlaps the front we wish to use front information as local boundary conditions for the finite difference method. This can be done in a variety of different ways, but in practice a very simple method has proven adequate for many problems. We call this method extrapolation by constant state. We simply replace any state in the finite difference stencil that lies on a different side of the point being updated (with respect to the propagated interface) by the state on the interface nearest that point. This of course introduces some error in the computation near the front, but this is compensated by the removal of the differencing across the front and the more accurate computation of the front position and states. Several authors have proposed alternative methods that use more complicated algorithms, but the above method has proven to be surprisingly successful in practice.

Exercises

1.Show that for a perfect gas, with $e = c_V T$, $PV=RT$ (V=1/ρ):

$$\gamma = 1 + \frac{R}{c_V}, \quad e = \frac{\rho^{\gamma-1}}{\gamma-1}\exp\left(\frac{S-S_{ref}}{c_V}\right), \quad P = \rho^{\gamma}\exp\left(\frac{S-S_{ref}}{c_V}\right),$$

where S_{ref} is some reference entropy. (Note only changes in entropy have physical meaning for a perfect gas.)

2. Show that for a perfect gas, $\int_{P_0}^{P} \frac{dP}{\rho c}\bigg|_S = \frac{2}{\gamma-1}(c - c_0)$.

3. For fixed P_0 , define

$$r(P,S) = \int_{P_0}^{P} \frac{dp}{\rho c(p,S)}.$$

Show that the forward and backward characteristic equations can be written:

$$\frac{d}{dt}(u+r) = \left(\frac{\partial}{\partial S}\int_{P_0}^{P}\frac{dp}{\rho c(p,S)}\right)c\frac{\partial S}{\partial x}, \text{ along } \frac{dx}{dt} = u + c$$

$$\frac{d}{dt}(u-r) = \left(\frac{\partial}{\partial S}\int_{P_0}^{P}\frac{dp}{\rho c(p,S)}\right)c\frac{\partial S}{\partial x}, \text{ along } \frac{dx}{dt} = u - c$$

Further Reading

- R. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problem (2nd ed.)*, Interscience Publishers, 1967
- I-L. Chern, J. Glimm, O. McBryan, B. Plohr and S. Yaniv, *Front Tracking for Gas Dynamics*, J. Comp. Phys., **62**, pp. 83-110, 1986.
- J. W. Grove, *Applications of Front Tracking to the Simulation of Shock Refractions and Unstable Mixing*, J. Appl. Num. Math. **14**, pp. 213-237, 1994.

Front Tracking in Two Space Dimensions

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Two Dimensional vs. One Dimensional Flows

Characteristics in Multiple Space Dimensional Flows (1)

Characteristics in Multiple Space Dimensional Flows (2)

Characteristics in Multiple Space Dimensional Flows (3)

Bicharacteristics for the Euler Equations (1)

Bicharacteristics for the Euler Equations (2)

The Front Tracking Representation of a Flow in Two Space Dimensions

Front Tracking Terminology (1)

Front Tracking Terminology (2)

Front Tracking Terminology (3) Topological Structures

Front Tracking Terminology (4) Topological Structures

Front Tracking Terminology (5) Topological Structures

Types of Hypersurfaces (1)

Types of Hypersurfaces (2)

Types of Hypersurface Boundaries (1)

Types of Hypersurface Boundaries (2)

Front Tracking Operators

Front Tracking Operators

Point Propagation Operator

Tangential Update Operator

Exercises

Further Reading

Two Dimensional vs. One Dimensional Flows

Flows in two or more space dimensions are intrinsically more complicated than one dimensional flows. Primarily this is due to the greater degree of freedom of motion in higher dimensional flows, but there are important and related mathematical reasons for the greater complexity as well. As we saw in the previous lecture the method of characteristics played an important role in the design of the propagation algorithm for contact discontinuity motion. This is by no means the only application of this important theory. The method of characteristics also is central to the design of the higher order Godunov methods. It is basically through these ideas that the higher order corrections are added to the basic Godunov method. Perhaps the key feature that makes the method of characteristics useful in analysis and computation is that it reduces the relationships between flow variables from partial differential equation relations to ordinary differential equation relations along the (generally unknown) characteristic curves. Although the characteristics are unknown in the sense that they can not be determined without knowledge of the solution, they are generally easy to approximate for a single time step, and the characteristic equations for the flow variables can be easily approximated by discrete equations that yield second order accurate equations for the updated flow variables.

Characteristics in Multiple Space Dimensional Flows (1)

In contrast characteristics in higher space dimensions correspond to space-time hypersurfaces (e.g. moving curves in two space dimensions) and the flow variables along the hypersurfaces are related by partial differential equation relations. For concreteness let us give the formal definition of a characteristic surface for a multi-dimensional flow. Suppose we have a system of quasi-linear partial differential equations:

$$\frac{\partial u_i}{\partial t} + a_{ij}^k \frac{\partial u_j}{\partial x_k} = h_i .$$

Then a characteristic surface is a space-time hypersurface that is locally of the from $\phi(\mathbf{x},t) = 0$, where ϕ is a solution to the highly nonlinear partial differential equation:

$$\det\left(\frac{\partial \phi}{\partial t} \delta_{ij} + \frac{\partial \phi}{\partial x_k} a_{ij}^k \right) = 0.$$

Characteristics in Multiple Space Dimensional Flows (2)

If we let $\mathbf{n} = (n_k)$ denote the spatial normal to the characteristic surface for fixed t, and let λ denote the instantaneous speed of the moving surface in the direction $\mathbf{n}$ then:

$$n_k = \frac{\dfrac{\partial \phi}{\partial x_k}}{\|\nabla_{\mathbf{x}} \phi\|}, \qquad \lambda = -\frac{\dfrac{\partial \phi}{\partial t}}{\|\nabla_{\mathbf{x}} \phi\|},$$

and the characteristic equation becomes:

$$\det\left(\lambda \delta_{ij} - n_k a_{ij}^k\right) = 0,$$

That is λ is an eigenvector of the matrix:

$$\mathbf{A}(\mathbf{n}) = \left(n_k a_{ij}^k\right).$$

Characteristics in Multiple Space Dimensional Flows (3)

In contrast to one space dimensional flows where characteristics are spawned by a single point, characteristics in multi-space dimensional flows are generated by codimension two space-time surfaces, most commonly codimension 1 surfaces at a fixed time level (in 2D these are curves). Given such a initial object we form the characteristic surface through that object by solving for the bicharacteristic rays through each point on the generating object. We get one such ray for each characteristic family of the partial differential equation. Suppose $\lambda_k(\mathbf{u}(\mathbf{x},t),\boldsymbol{\xi})$ is the kth eigenvector of the matrix:

$$A = \left(\xi_k a_{ij}^k(\mathbf{u})\right).$$

Then the bicharacteristic ray through a point $\mathbf{x}_0$ on a surface with spatial normal $\mathbf{n}_0$ is given by the solution to the system of ordinary differential equations:

$$\frac{dx_i}{dt} = \frac{\partial \lambda_k}{\partial \xi_i}, \qquad \frac{d\xi_i}{dt} = -\frac{\partial \lambda_k}{\partial x_i}, \quad x_i(t_0) = x_i^0, \quad \xi_i(t_0) = n_i^0.$$

Bicharacteristics for the Euler Equations (1)

For the Euler equations the characteristic speeds are:

$$\lambda_1 = u_i \xi_i - c|\xi|, \;\; \lambda_2 = u_i \xi_i, \;\; \lambda_3 = u_i \xi_i + c|\xi|,$$

and the corresponding bicharacteristic rays:

$$\frac{dx_i}{dt} = u_i - cn_i, \qquad \frac{d\xi_i}{dt} = -\frac{\partial u_j}{\partial x_i}\xi_j + \frac{\partial c}{\partial x_i}|\xi|$$

$$\frac{dx_i}{dt} = u_i, \qquad \frac{d\xi_i}{dt} = -\frac{\partial u_j}{\partial x_i}\xi_j$$

$$\frac{dx_i}{dt} = u_i + cn_i, \qquad \frac{d\xi_i}{dt} = -\frac{\partial u_j}{\partial x_i}\xi_j - \frac{\partial c}{\partial x_i}|\xi|,$$

$$n_i = \frac{\xi_i}{|\xi|}.$$

Bicharacteristics for the Euler Equations (2)

We can write a bicharacteristic form for the Euler equations:

$$\left(\frac{DP}{Dt} \pm c\mathbf{n} \bullet \nabla P\right) \pm \rho c\mathbf{n} \bullet \left(\frac{D\mathbf{u}}{Dt} \pm c\mathbf{n} \bullet \nabla \mathbf{u}\right) = \rho c(\mathbf{n} \bullet \mathbf{n} \bullet \nabla \mathbf{u} - \nabla \bullet \mathbf{u})$$

$$\frac{DS}{Dt} = 0$$

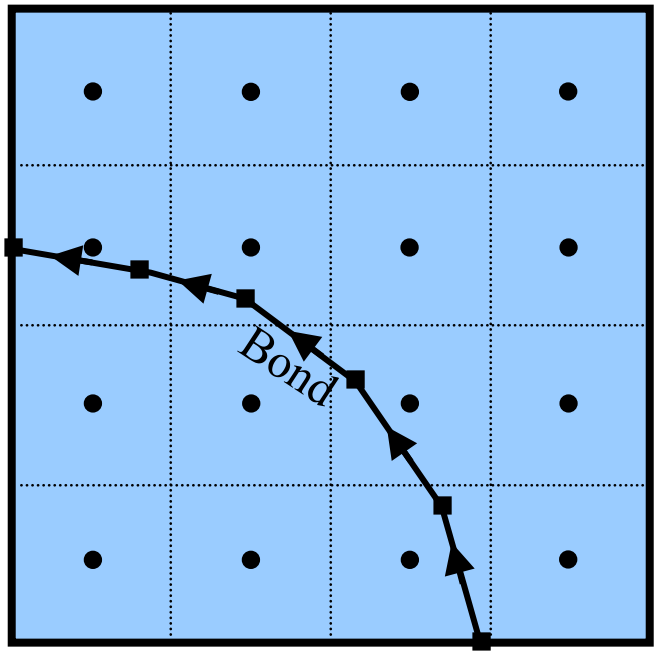
$$\mathbf{T} \bullet \frac{D\mathbf{u}}{Dt} + \frac{\mathbf{T} \bullet \nabla P}{\rho} = 0, \ \mathbf{T} \bullet \mathbf{u} = 0$$

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \bullet \nabla.$$

We see that these equations have a form similar to the one dimensional equations, but that in each case the flow along a characteristic is driven by terms involving derivatives orthogonal to the direction **n**. It is this additional coupling between directional derivatives that complicates the analysis of a multidimensional flow. For more information on bicharacteristics and their role in solving partial differential equations see Courant and Hilbert Volume II.

XHM The Front Tracking Representation of a Flow in Two Space Dimensions

The diagram on the right shows a schematic picture of the representation of the flow in the vicinity of a tracked interface. The representation consists of a set of lattice points that are associated with the flow states at the center of a spatial grid. For simplicity we assume this grid is rectangular. The tracked interface is shown as a piecewise linear curve. The linear segments are called bonds. The flow near the front is described by two states associated with the points on the curve. These states can be regarded as the limit of the flow variables as the discontinuity front is approached through a given side of the interface. Note that we assume the curve is oriented.



- ● Grid State
- ■ Tracked Points (left,right) States

Front Tracking Terminology (1)

An important aspect of the front tracking method in two or more space dimensions is the description of the geometry of the discontinuity fronts. The following set of definitions describe the most common set of objects used in describing the collection of tracked objects. Each definition corresponds to a data class in the front tracking implementation used in *FronTier*.

- **POINT**: Describes a geometric location in space. When associated with a tracked interface, **POINT**s are extended by inheritance to a data class that contains state data associated with the limiting values of the state data at either side of the interface. These states are referred to as the left and right states at the **POINT**.
- **BOND**: **BOND**s are directed connectors between pairs of **POINT**s. Each **BOND** structure contains pointers to a start and end **POINT** as well as pointers to a previous and next **BOND**. This linked list allows **BOND**s to be connected into piecewise linear curve segments.
- **TRI**: A **TRI** corresponds to a geometric triangle. Each **TRI** contains pointers to three **POINT**s that form the vertices of the triangle. A **TRI** contains pointers to three adjacent **TRI**s called its neighbors so that **TRI**s can be linked together to form piecewise linear surfaces. This structure is only used for three space dimensional flows, but is included here for completeness.

Front Tracking Terminology (2)

- **CURVE**: A **CURVE** is a piecewise linear curve. Since **BOND**s can be linked together to from lists, it suffices for a **CURVE** structure to contain pointers to the first and last **BOND** on the **CURVE**. The set of **POINT**s on the curve can be accessed by looping through the **BOND**s on the curve, starting at the first and ending at the last. The previous and last pointers of the first and last **BONDS** of a **CURVE** respectively are null.
- **NODE**: Interaction points between **CURVE**s are called **NODE**s. Each **CURVE** is associated with two (possible the same) **NODE**s that mark the start and end of that **CURVE**. Each **NODE** has an associated **POINT** that gives its position. **NODE**s are the only allowed locations where two or more **CURVE**s can meet. **NODE**s contain lists of pointers to the **CURVE**s that meet at that **NODE**.
- **SURFACE**: **SURFACE**s are three dimensional structures that correspond to a piecewise linear surface. They consist of a collection of connected **TRI**s and a set of **CURVE**s that form the topological boundary of the **SURFACE**.

Front Tracking Terminology (3)

Topological Structures

The previous set of data structures are used to describe the geometric and connecting properties of the tracked objects. In many cases the physical meaning of a geometric object is more naturally associated with the topological properties of the object, such as whether it locally separates space. The following set of data structures are designed to describe the embedding properties of the tracked objects in the relevant Euclidean space for the flow. Within the dimensional context of a flow each geometric object (point, curve, surface, etc.) has a pointer to a corresponding (possible empty) topological object and vice versa. So it is common when the flow dimension is understood to speak casually as if the geometric and topological object were the same. Thus one may speak of a surface in a three dimensional flow as being the same as a hypersurface. But one must be careful in the terminology since for a two dimensional flow a hypersurface corresponds to a curve.

Front Tracking Terminology (4)

Topological Structures

- **HYPERSURFACE**: A **HYPERSURFACE** is associated with every co-dimension one collective geometric object. For one dimensional flows these are **POINT**s, for two dimensional flows **CURVE**s, and for three dimensional flows **SURFACE**s. A **HYPERSURFACE** is generally regarded as a locally space separating oriented manifold with boundary. Each **HYPERSURFACE** contains two integer labels for the two spatial regions that bound it. It is required that these labels be consistent so that **HYPERSURFACE**s bounding the same region have the same label on their appropriate sides. These structures describe the main wave fronts in a flow. Lower dimensional objects are generally associated with interactions between waves.
- **HYPERSURFACE ELEMENT** : **HYPERSURFACE**s are composed of collections of **HYPERSURFACE ELEMENT**s. Depending on the flow dimension there are either empty, **BOND**s or **TRI**s for respectively one, two, or three space dimensional flows. Currently there are no topological properties associated with **HYPERSURFACE ELEMENT**s, this label just gives a convenient way of accessing the geometric information associated with surface element without having to reference the specific dimension of the flow.

Front Tracking Terminology (5)

Topological Structures

- **HYPERSURFACE BOUNDARY**: These are the boundary objects associated with a **HYPERSURFACE**. In two dimensional flows these are **NODE**s, while for three dimensional flows they are **CURVE**s. This object is undefined for a one dimensional flow.

There are several other miscellaneous structures that are used in various special cases. Furthermore each of the above objects constitutes an inherited class which means that the objects have associated operators for their allocation and manipulation. These objects may also be extended to incorporate new properties by higher level libraries.

One further data structure is an **INTERFACE** which is a collective object containing pointers to all geometric and topological objects that constitute the set of tracked features in the computation.

Types of Hypersurfaces (1)

Hypersurfaces correspond to wave fronts and are classified according to the type of wave they model. We generalize the notion of waves to include boundaries, both physical and computational.

- Boundary Hypersurfaces:
  - SUBDOMAIN - A boundary hypersurface between parallel subdomains, includes periodic boundaries.
  - REFLECTION - A artificial boundary indicating reflection boundary conditions. These are implemented like a subdomain boundary except that the velocity data is reflected.
  - DIRICHLET - A boundary where a specific flow state is imposed.
  - NEUMANN - A alternative form of a reflecting boundary that is suitable for hypersurfaces not aligned with the computational grid. Reflecting walls.
  - PASSIVE - An inactive boundary.

Types of Hypersurfaces (2)

- Wave Front Hypersurfaces:
  - CONTACT: A contact discontinuity, slip surface, or material interface.
  - Forward Wave Family
    - FORWARD SHOCK WAVE
    - FORWARD SOUND WAVE LEADING EDGE - The leading edge of a forward rarefaction wave.
    - FORWARD SOUND WAVE TRAILING EDGE - The trailing edge of a forward rarefaction wave.
  - Backward Wave Family
    - BACKWARD SHOCK WAVE
    - BACKWARD SOUND WAVE LEADING EDGE - The leading edge of a backward rarefaction wave. _
    - BACKWARD SOUND WAVE TRAILING EDGE - The trailing edge of a backward rarefaction wave.

When the dimension of a flow is understood we will often speak of the type of a point, curve, or surface as being the type of the corresponding hypersurface. Thus we may say a curve is a CONTACT in a two dimensional flow.

Types of Hypersurface Boundaries (1)

Hypersurface boundaries are associated with wave front interactions. They are also described by a type identifier. As with hypersurfaces, if the flow dimension is clear, we might speak of a node type in two dimensions as being the type of the corresponding hypersurface boundary.

- Boundary Hypersurface Boundaries:
  - PASSIVE - A hypersurface boundary for a pair of PASSIVE hypersurfaces
  - FIXED - A hypersurface boundary that is in a fixed position, such a corner of the computational domain.
  - CLOSED - An artificial hypersurface boundary formed by a closed loop.
  - NEUMANN - Intersection of a CONTACT with a NEUMANN boundary.
  - DIRICHLET - Intersection of a wave front with a DIRICHLET BOUNDARY
  - SUBDOMAIN - Intersection of a wave front with a SUBDOMAIN BOUNDARY

Types of Hypersurface Boundaries (2)

- Wave interactions:
  - REGULAR REFLECTION - Hypersurface boundary associated with the regular reflection of a shock with a NEUMANN boundary.
  - ATTACHED SHOCK - A shock wave attached to a wall corner.
  - MACH - A Mach triple point.
  - CROSS - A crossing of two shocks of different families.
  - OVERTAKE - An overtake of one shock by another of the same family.
  - REFRACTION - The refraction of a shock through a material interface.
  - TRANSMISSION - A refraction with a subsonic state behind the incident shock.
  - CONTACT-CONTACT - A triple point junction of three material interfaces.
  - WAVE END - A breaking point of a shock, the wave terminates at zero strength.
  - TOTAL INTERNAL REFLECTION - A refraction with no transmitted wave.

Front Tracking Operators

FronTier has been developed using an objected oriented approach so that manipulations and inspection of interface properties is done through a set of operators. It is beyond the scope of this course to go into detail, but some of the more important operations are listed below.

- Interface manipulation
  - create
  - copy
  - delete
  - add point
  - delete point
  - redistribute points
  - join/merge
  - split/divide

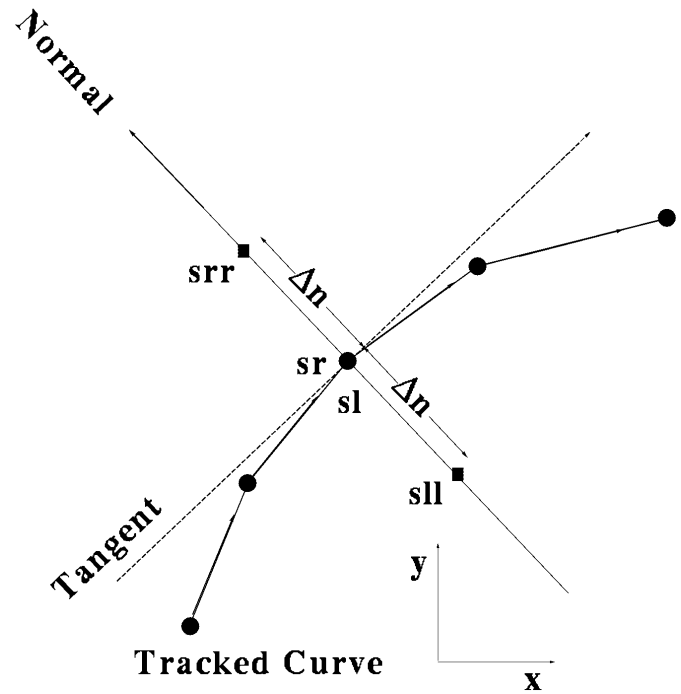
Front Tracking Operators

- Interface inspection
  - coordinates of a point
  - states at a point
  - compute normal at a point on a hypersurface
  - get neighboring object (such as from one bond to the next)
  - find the topological component of a location
- Interface propagation
  - Propagate (time advance) a point on a hypersurface
  - Propagate a wave interaction (for example a tracked shock wave refracting through a material interface
- Solution evaluation
  - evaluate the solution at an arbitrary location by use of an interpolant that preserves the discontinuous nature of the solution across a tracked interface.

Point Propagation Operator

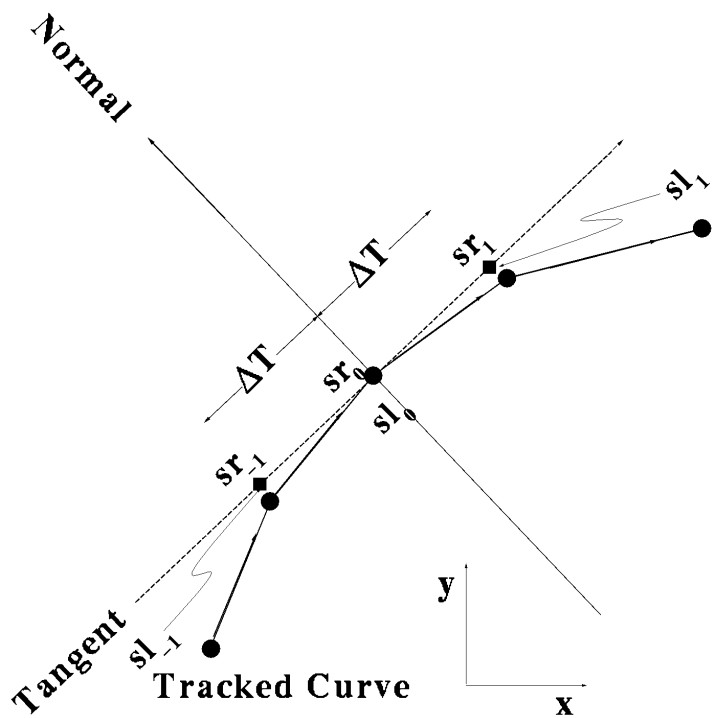
We will complete this lecture with a description of the two dimensional point propagate operator that advances the location and states of a point on a tracked front. This operation uses operator splitting by dividing the propagation step into two phases, a normal propagate and and tangential propagate.

The figure on the right shows the geometry for the normal point propagation operator. The normal at the give point is computed by forming the secant vector between the two adjacent points and rotating by 90 degrees. The solution function is evaluated to obtain two states located at a distance of Δn on either side of the front. Using these states together with the states on the front, the 1D propagate operator the normal direction is used to compute the new point position and a pair of normally propagated states.



Tangential Update Operator

Operator splitting requires that after the propagation in the direction normal to the interface is accomplished a second update in the tangential direction must be performed. The figure below shows the geometry for this update. The update is done in two steps, one on either side of the interface. Starting at the point to be updated we move along the tangent line in increments of ΔT, and obtain left and right states along this line by projection and interpolation along the interface. Note that we are assuming all points on the interface have already been propagated by the normal update operator. For simplicity the diagram shows a three point stencil. The resulting states are then passed to a one dimensional finite difference solver that returns a pair of updated states. The position of the interface points are not changed by the tangential update operator.



Exercises

1. Show that if a bicharacteristic ray intersects a characteristic surface of the same family, then that ray lies entirely inside the characteristic surface.

2. A shock wave is said to belong to the kth characteristic family and to be stable in the sense of Lax if the shock speed satisfies the inequality:

$$\lambda_k^r < s < \lambda_k^l$$

Show that for a perfect gas, forward shocks are stable in the sense of Lax if the pressure increases across the shock from right to left.

Further Reading

- Courant and Hilbert, Methods of Mathematical Physics, Volume II, Interscience Publishers, 1962. (see in particular chapter VI.)
- R. Richtmyer and K. W. Morton, *Difference Methods for Initial-Value Problem (2nd ed.)*, Interscience Publishers, 1967
- I-L. Chern, J. Glimm, O. McBryan, B. Plohr and S. Yaniv, *Front Tracking for Gas Dynamics*, J. Comp. Phys., **62**, pp. 83-110, 1986.
- J. W. Grove, *Applications of Front Tracking to the Simulation of Shock Refractions and Unstable Mixing*, J. Appl. Num. Math. **14**, pp. 213-237, 1994.

Implementing Front Tracking

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Front Tracking Utilities (1)

Front Tracking Utilities (2)

Front Tracking Utilities (3)

Front Tracking Utilities (4)

Components

The Topological Grid (1)

The Topological Grid (2)

Generation of the Interface Topology Lists

Intersections

Interface Crossings

Interface Redistribution

Redistribution Example

Examples of Triangle Redistributions

Remarks on Redistribution

The Hyperbolic Solution Function

Two Dimensional Triangulated Grid

The Hyperbolic Solution Function in 3D

Remarks on Implementation on Parallel Computers

Exercises

Further Reading

Front Tracking Utilities (1)

Before proceeding with a discussion of examples of interface interaction algorithms, we need to discuss some basic utilities a bit more in detail since these are basic tools used in implementing the interaction resolution methods. The basic operations are:

- **Component lookup**: given a position in space, identify the connected component formed by the tracked fronts that contains that position.
- **Nearest interface point**: given a position in space, find the nearest interface element to that point. This information includes the hypersurface element (point for 1D, bond for 2D, tri for 3D), the side on which the position lies with respect to the element, and the coordinates of the projection of that point onto the hypersurface element. If the orthogonal projection does not lie on the element, this projection is the nearest point on the element to the given position.
- **Intersections check**: tests all surface elements for non-trivial intersections. The intersections operation returns a possibly empty list of all intersections found. This test is performed every time step to check for wave interactions.
- **Interface redistribution**: as the interface expands and contracts in different locations, the distributions of the sizes (and shapes in 3D) of the hypersurface elements can become distorted. Redistribution regenerates the interface hypersurface elements into a more even distribution of sizes.

Front Tracking Utilities (2)

- **Solution function**: given a position and a component find an interpolated state value. If the given component does not match the component of the position, then that position is projected onto the nearest point with the correct component and interpolation is performed at the projected location. Optionally this function can specify a hypersurface, in which case the position is projected onto the hypersurface prior to interpolation. In this case the given component must belong to the hypersurface or else an error condition is flagged.
- **Interpolation functions**: Encapsulated functions to perform linear or bilinear interpolation on sets of state values. Input consists of the states to be interpolated together with the interpolation weights.
- **Equation of state encapsulation**: computes thermodynamic functions of a state. Such functions include pressure, specific internal energy, and temperature. This class hides specific information about the equation of state from most areas of the code.
- **Interface surgery utilities**: Functions to add/delete points from interface elements. Functions to split or join interface objects along specific objects (for example split a curve into two curves at an interior point thus creating a new node, or join two curves at a common node). Functions to trim sections from interface objects such as cutting off bond sections from the end of a curve.

Front Tracking Utilities (3)

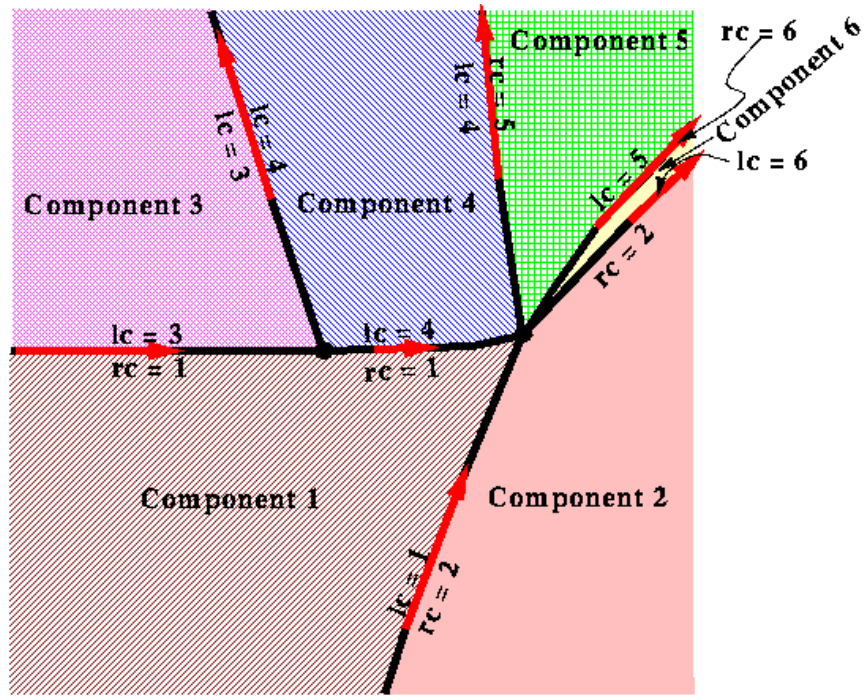
- **Point propagate**: Normal propagation and tangential propagation operators for points on hypersurfaces.
- **Node propagation encapsulation**: wrappers for physics dependent node propagation routines.
- **Encapsulated finite difference solvers**: this encapsulation allows the implementation of multiple solver options in the code.
- **I/O and diagnostic utilities**: print vector and scalar fields, restart dumps, and perform data analysis.
- **Initialization and restart**: Set up the Cauchy data for a simulation.
- **Parallel Communication utilities:**
  - Send and receive arrays of states data
  - Send and receive interface structures, include reconstruction of addresses
  - Clip interface to a rectangular region. This operation is used to send a copy of an interface in a buffer zone near a parallel boundary to a neighboring processor
  - Merge two interfaces along a common boundary. Used to merge the communicated interface into the main interface on a processor.
  - General parallel support provided by the Message Passing Interface Package.

Front Tracking Utilities (4)

This a partial list of some of the more important utilities used in *FronTier*. We will not go into a detailed discussion of most of these operations, but it is important to understand a few of the basic operations, in particular the computation of components, nearest interface points, and intersections, since these operations are central to methods used to propagate points and resolve wave interactions.

Components

A valid interface is by definition one that is non-self intersecting except at nodes (or in three space dimensions hypersurface boundaries, i.e. curves). This fact means that a valid interface divides the computational domain into a set of connected components and that the component value at a given location can be computed by projecting that point onto the nearest interface and using as the component value an integer label associated with the corresponding curves. More precisely each hypersurface has two associated component labels, one for each side. The consistency condition requires that hypersurfaces bounding a common region have the same component label on their sides that bound that region. Any inconsistency in the component labels is a sign that the interface is tangled. The figure shows a schematic of the mapping between interface component labels and the component assignment to a region.



The Topological Grid (1)

The operations of computing the component of a position, finding the nearest interface point, and determining intersections all have a common feature. They each involve looping over all interface hypersurface elements and performing an algebraic computation. Indeed the basic operation of finding a component essentially computes the nearest interface point and uses the component on the appropriate side from the hypersurface nearest the point to specify the component. Intersections on the other hand must loop through all pairs of hypersurface elements and compute the intersection, if it exists, of the two elements. In all three cases the algebraic operations are nontrivial and put a large load on the computation. Fortunately the basic operations in each case are also essentially local in the sense that if the two elements (point and hypersurface element or pair of hypersurface elements) are spatially distant from one another, then it is not necessary to actually compute the corresponding quantity (distance or intersection). This means we can considerable speed up the evaluation of a given operation by precomputing a hashed list of the interface elements associated with a geometric decomposition of the computational domain. In practice this decomposition is based on a rectangular lattice called the topological grid.

The Topological Grid (2)

The topological grid has a mesh size that is independent of the computational mesh used as the spatial discretization of the conservation system. Furthermore this grid is always required to be square. This grid is used as a hashing bin for the interface elements. In practice we usually use a grid size for the topological grid that is about three to six times that of the computational grid. Associated with the topological grid are lists of interface elements that lie close to each topological gird cell. For example in two space dimensions, associated with each $cell(i,j)$ of the topological grid are integers

$N(i,j)$ = number of bonds close to $cell(i,j)$
$M(i,j)$ = number of separate components close to $cell(i,j)$
$bond(i,j,k)$, $k = 1,..,N(i,j)$
$curve(i,j,k)$, $k = 1,..,N(i,j)$
$comp(i,j,k)$, $k = 1,..,M(i,j)$

that gives the addresses of the $N(i,j)$ bonds that lie close to that block, the address of the curve that contains each bond, and a list of the components associated with that cell. The notion of close can be taken to mean intersects for practical purposes. If no bonds lies in the vicinity of a given cell, then the corresponding list of bonds and curves for that cell is empty. In addition we have an array, compon(i,j), that contains either the unique component number of that cell or the identifier ONFRONT.

Generation of the Interface Topology Lists

The topology lists are formed by looping over all hypersurface elements on the tracked interface and for each element computing the blocks that intersect the element. This in turn is done by decomposing the element into pieces that are sufficiently small so that their endpoints lie in adjacent topological cells. The component numbers for the topological mesh cells are set by continuation. Once the component number of one cell is determined (say by projection onto a hypersurface element in an adjoining cell) this component number is assigned to all adjacent cells recursively in all directions until an ONFRONT cell is reached.

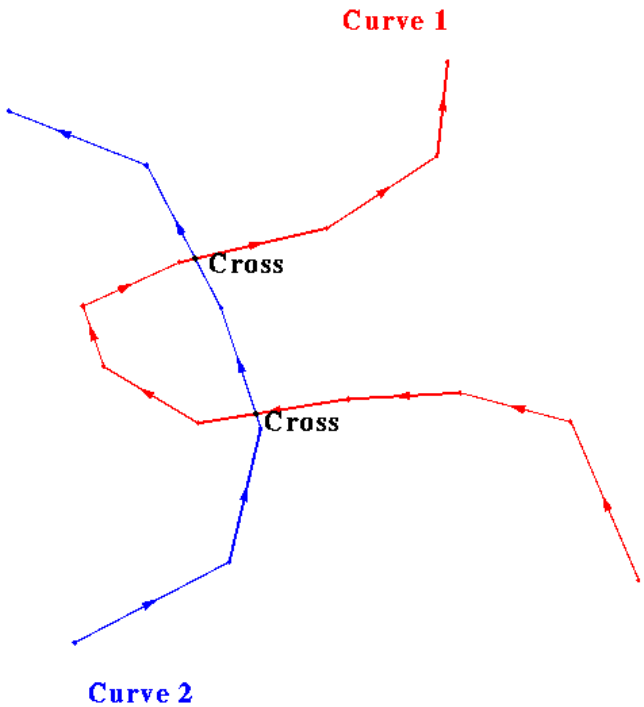
Utilization of Topology Lists

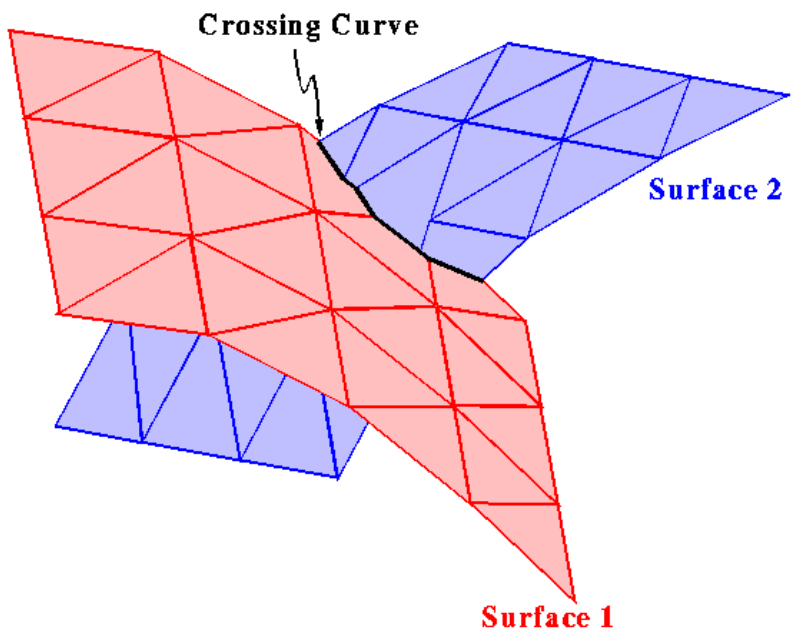
The topology lists are used in the obvious ways. To find the nearest interface point to a location, identify the grid element containing that location. If that cell is ONFRONT then we need only check elements in that and the immediately adjacent cells for the nearest element (it is here that the requirement that the topological grid be square is imposed). If the cell is not ONFRONT we resort to looping over all hypersurface elements on the interface. Fortunately this is a rare case since in practice we only require the nearest interface points for locations that are close to the interface. For intersections we only need to check for intersections between elements with a common cell.

Intersections

The intersections operator returns a list of the interface crossings. In two space dimensions each crossing structure contains the addresses of a pair of crossing bonds, the curves containing the crossing bonds, and a point marking the position of the intersection. Cross structures are maintained as a doubly linked list, which aids in the processing of the intersection list as described in the lecture on resolving tracked wave interactions. In three space dimensional flows the cross structure contains the assembled information describing the intersection of two surfaces. This consists of a curve like structure (called a c_curve) whose “bonds” correspond to the intersections of triangle elements. The data structure that describes a linear segment of a c_curve is called a c_bond and contains pointers to a pair of intersecting triangles, the surfaces containing the triangles, and the geometric information describing the intersection segment. This information will be used in the interface surgery functions that slice a pair of surfaces along their intersection forming a set of new surfaces that only intersect along the newly formed curve. This information is then passed along to other functions that determine which of the new surfaces are physical and should be retained and which should be removed from the computation.

Interface Crossings



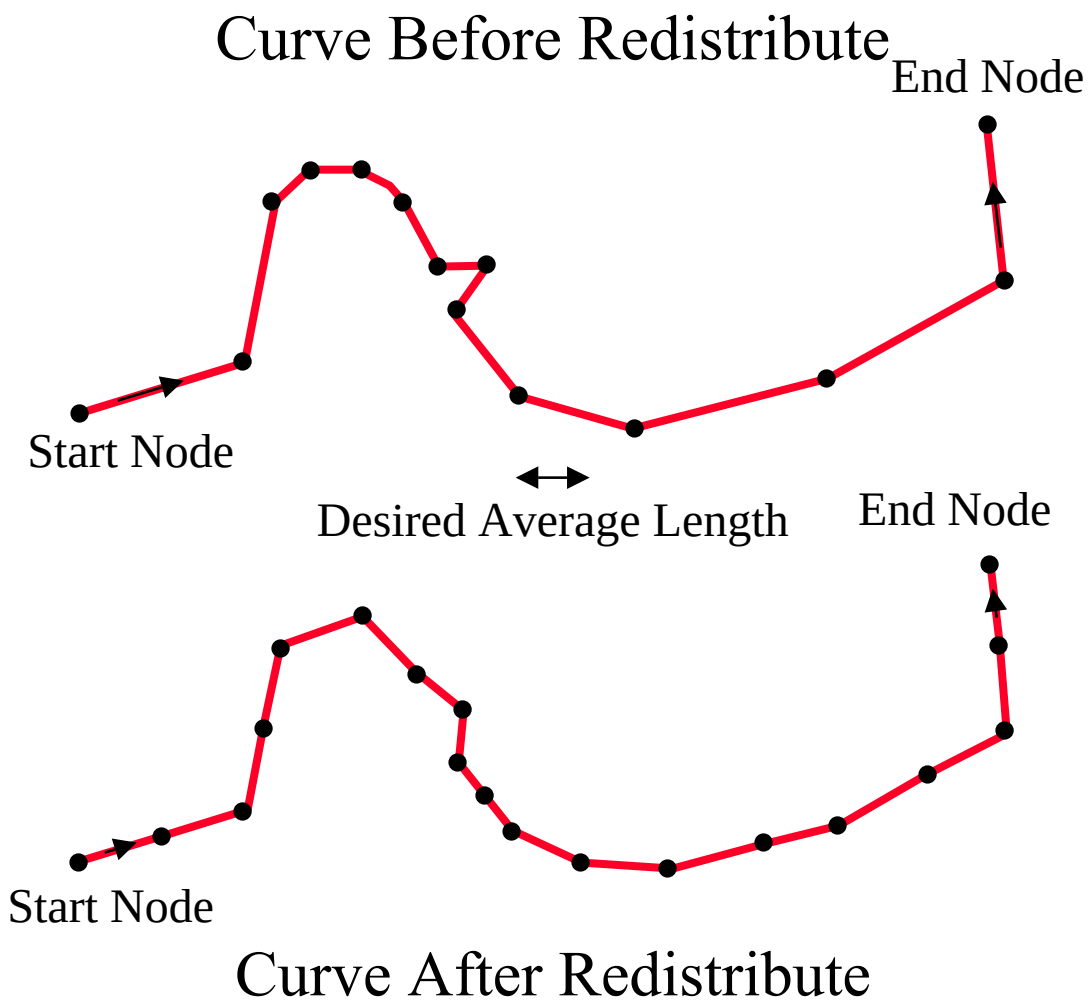


The two diagrams show examples of two and three dimensional crossings. Note that in two space dimensions crosses are discrete and thus it is necessary to process the cross list to identify corresponding crosses. Any formation of interface crossing indicates a wave interaction that must be resolved.

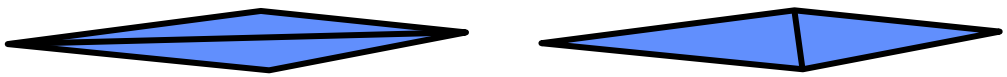
Interface Redistribution

Front tracking can be described as a semi-Lagrangian numerical method for the interface points. The points are propagated using the dynamics of the flow field in neighborhood of the points, and the hypersurface elements that join the points will stretch and contract as the points converge and diverge from each other. Eventually this can lead to a very high variance in the size and shape of the interface elements. Such a distribution will also lead to the formation of numerical instabilities in the interface. To avoid this problem the tracked interface is periodically re-interpolated to form new elements that have a more uniform distribution in the size and shape of the hypersurface elements. In two space dimensions this is done by moving from a physical node along a curve, inserting new points at a constant distance with respect to the arclength along the curve and removing the intervening points. The algorithm in three dimensions is more complicated. It requires examining the triangles on the interface according to their size and aspect ratio. Large triangles are subdivided into small ones, and small triangles are combined with adjacent triangles. Poor aspect ratio triangles can be treated in a variety of ways including deletion by merging two adjacent points and diagonal flipping between two adjacent triangles. A recent algorithm by X. L. Li uses a combination of this technique and an interface reconstruction technique based on level set ideas to provide a robust 3D algorithm that also automatically resolves 3D tangles.

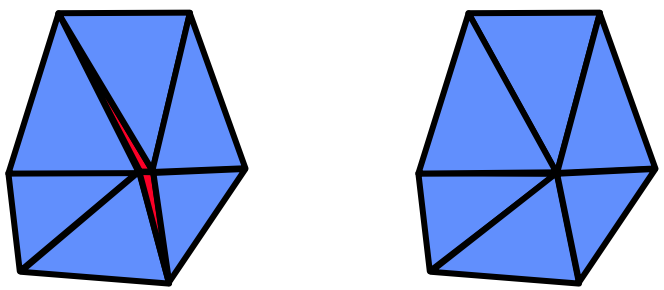
Redistribution Example



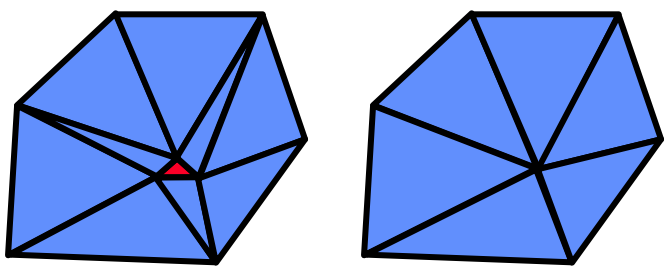
Examples of Triangle Redistributions



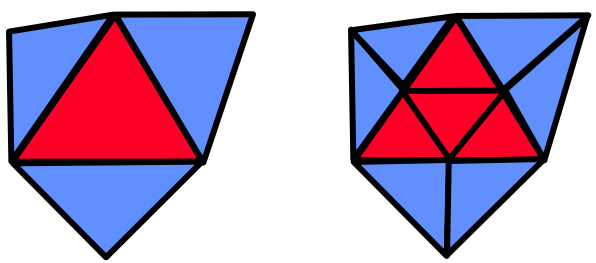
Flip Diagonals



Merge Edge



Delete Small Triangle



Split Triangle

Remarks on Redistribution

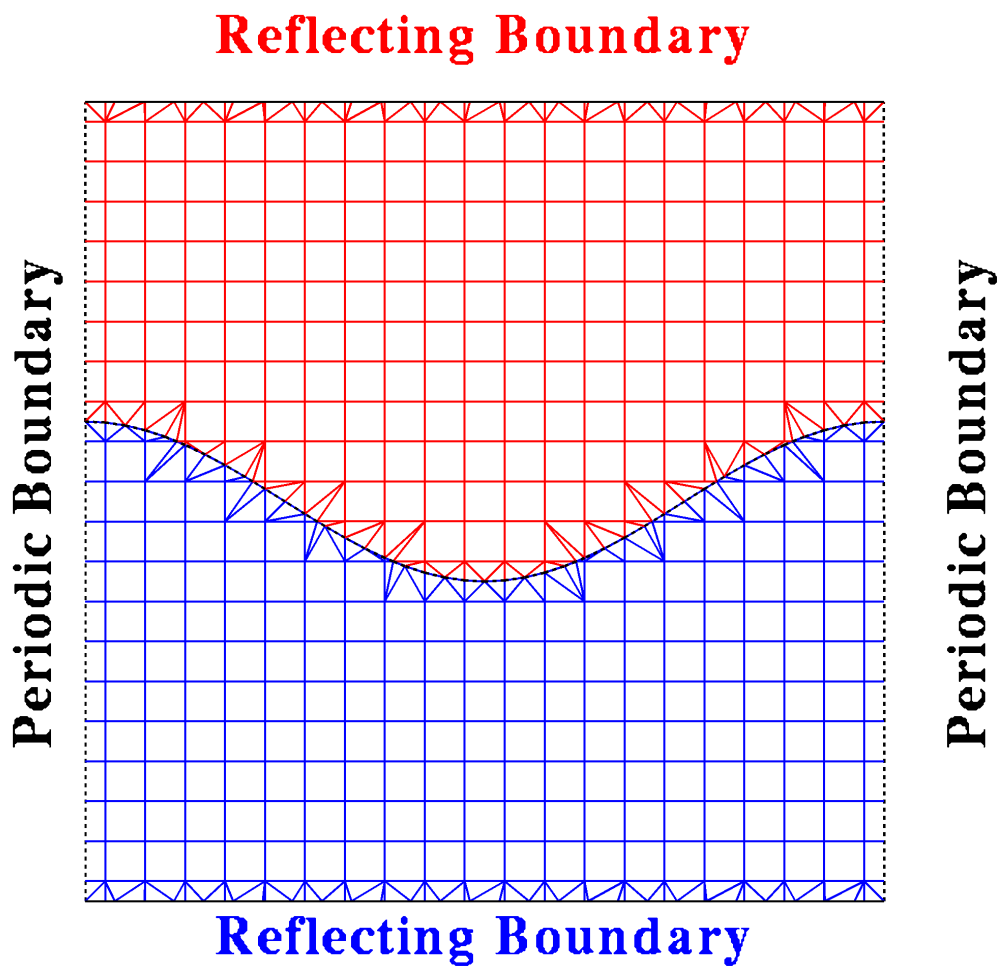
The need for interface redistribution is a necessary evil in the front tracking method and must be treated with care. Redistribution introduces an artificial surface force that has a stabilizing effect similar to surface tension. It is important that this operation be applied with the minimum frequency necessary to damp numerical oscillation but not to over suppress the physical growth of perturbations on the interface. It also has different affects for tracked shock wave verses contact discontinuities. In the case of shocks, which are generally fast moving waves that are asymptotically plane wave stable, a large amount of redistribution is generally desirable since it reinforces the natural tendency for shocks to approach a plane wave steady state. On the other hand contacts are unstable with respect to perturbations and a much lower frequency of redistribution is desired to avoid over damping the physical growth of perturbations. In practice we have found that for two dimensional flows a redistribution frequency of every four time steps for shocks and every twenty time steps for contacts are reasonable compromise values. However the true frequencies are problem dependent and are specified by the user.

The Hyperbolic Solution Function

As we have already seen, an important operation in the interface propagation algorithm is the evaluation of flow states at positions located at specific distances off the front in the direction normal to the front. This operation is supported by the hyperbolic solution function. The input for this operation is a spatial position and a component number for the region in which the evaluation should take place. If the component of the given position does not agree with the requested component, then that position will be projected onto the nearest position in the desired component.

In two space dimensions we construct a triangulation of the computational domain using the centers of the computational cells and the positions of the tracked points as vertices. This is a constrained triangulation with the requirement that no triangle cross the interface, or in other words, no triangles have vertices with different components. The solution function looks up the triangle that contains the given location and uses linear interpolation on the state data associated with the triangles vertices to evaluate the flow at the given location. In practice we only triangulate a region near the front. If the rectangle formed by four adjacent computational cell centers contains no interface point, we use bilinear interpolation on this region instead of linear interpolation.

Two Dimensional Triangulated Grid



The Hyperbolic Solution Function in 3D

In three space dimensions we do not construct a global interpolation grid due to cost and complexity of constructing a full 3d tetrahedrazation which would be the generalization of the 2D triangulation. Instead we first locate the rectangle formed by eight adjacent cell centers (a cell on what is referred to as the dual lattice to the computational grid) and using only those vertices inside this dual lattice cell whose component number agrees with the input component number we find the four closest vertices whose convex hull contains the given location, and then use linear interpolation on the states at these four vertices. If no such set of points is found then we obtain the state information by projection onto the nearest tracked hypersurface element with the correct component.

Remarks on Implementation on Parallel Computers

FronTier uses a distributed memory model for its parallel implementation and will work on any machine that supports the Message Passing Interface (MPI) standard. Parallelism is achieved through a fixed grid domain decomposition. This method, which is commonly used for hyperbolic problems, decomposes the computational domain into a set of disjoint rectangular subdomains that are assigned to separate processors. Adjoined to the boundary of each subdomain is a buffer zone of width four or more computational grid blocks. The buffer zone is sufficiently wide so that every cell in the subdomain has it domain of dependence within the union of the subdomain and its buffer zone. This allows an explicit numerical method to make a full time step without communicating with its neighboring processors. At the end of the time step the buffer zone information is discarded and replace by communicated data from the neighbor. The main extra step in front tracking is the need to clip the interface to the subdomain, communicate the interface information from the edges of the subdomain to its neighbor, and then then reassemble the interface. This method has the advantage that it minimizes parallel communication, which is a dominate cost in runs with large numbers of processors.

Exercises

Consider the inviscid Burger's equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{u^2}{2}\right) = 0$$

Write a one dimensional front tracking code to solve this problem. Use the Lax-Wendroff method for the interior solver. Develop a version of the method of characteristics suitable to use to track a discontinuity for this equation. Compare the results you obtain using front tracking with the results of a straightforward captured shock for the initial value problem:

$$u(x,0) = \begin{cases} 1 & x < 0 \\ 0 & x > 0 \end{cases}$$

Note that the exact solution to this problem consists of a shock moving with speed one half.

Further Reading

- J. W. Grove, *The Interaction of Shock Waves with Fluid Interfaces*, Adv. Appl. Math. **10**, pp. 201-227, 1989.
- J. Glimm and O. McBryan, *A Computational Model for Interfaces*, Adv. Appl. Math., **6**, pp. 422--435, 1985.
- *The Front Tracking Home Page*, http://www.ams.sunysb.edu/~shock/FTdoc/FTmain.html

Wave Interactions in Two Space Dimensions

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Wave Interactions in Higher Dimensional Flows

Multidimensional Riemann Problems (1)

Multidimensional Riemann Problems (2)

Elementary Wave Solutions

Steady State Self-Similar Flow

Steady State Self-Similar Flow Hugoniots (1)

Steady State Self-Similar Flow Hugoniots (2)

Shock Polars (1)

Shock Polars (2)

Prandtl-Meyer Waves (1)

Prandtl-Meyer Waves (2)

Prandtl-Meyer Waves (3)

Prandtl-Meyer Waves (4)

Prandtl-Meyer Waves (5)

The Steady State Wave Curve

Shock Refractions

Shock Refractions

Types of Elementary Waves

Exercises

Further Reading

XHM Wave Interactions in Higher Dimensional Flows

One major simplifying feature of one dimensional flows is that for many purposes wave interactions can be interpreted as local binary interactions between nearly constant flow regions. This is to say the local structure of a flow is characterized by Riemann problems and their solutions. This was one of the key properties exploited by Glimm in his proof of the convergence of the random choice method. The situation in higher space dimensions is considerably more complicated. Waves can exhibit complex behaviors in both space and time. A simple example is the steady state refraction of a shock wave through a material interface. In one dimension this is easily described by a Riemann problem and yields a solution that consists of a transmitted shock and a reflection shock or rarefaction wave. Momentum is transferred from the shock to the material interface, which causes the velocity of the interface to change after the refraction. In contrast the two dimensional refraction of a plane shock with a planar material interface exhibits a variety of behaviors depending upon the orientation of the two interfaces with respect to each other. This can range from simple one dimensional behavior if the fronts are parallel, through a series of steady state configurations if the angle of interaction is small, to extremely complex unsteady interactions for other configurations.

Multidimensional Riemann Problems (1)

The straightforward generalization of the Riemann problem to higher dimensional flows exploits the scale invariance of the equations in the absence of source terms. More precisely the Riemann problem is defined as an initial value problem with scale invariant initial data, that is data that is constant on rays centered at the origin. Then exactly as in the one dimensional case we can show that if the solution to:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \bullet \mathbf{f}(\mathbf{u}) = 0, \quad \mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}), \text{ where } \mathbf{u}_0(\alpha\mathbf{x}) = \mathbf{u}_0(\mathbf{x}), \ \forall \alpha > 0,$$

is unique, then the solution satisfies:

$$\mathbf{u}(\mathbf{x},t) = \mathbf{w}(\mathbf{x}/t) = \mathbf{w}(\boldsymbol{\xi}),$$

and that **w** is a solution to the conservation law:

$$\nabla_{\xi} \bullet \left(\mathbf{f}(\mathbf{w}) - \mathbf{w} \otimes \boldsymbol{\xi}\right) + d\mathbf{w} = 0,$$

where d is the spatial dimension of the flow. It is immediately obvious that in one space dimension, this reduction reduces to the solution of a Riemann problem, and that the above equation is just the formalized statement of this Riemann problem. In more that one space dimension the resulting equation is a conservation law itself and will exhibit a complex behavior.

Multidimensional Riemann Problems (2)

For the Euler equations the self-similar flow equations can be simplified by introducing the self-similar velocity $\mathbf{w} = \mathbf{u} - \mathbf{x}/t = \mathbf{u} - \xi$. It is left as an exercise to show that the conservative scale invariant form of the Euler equations is:

$$\nabla_\xi \bullet \rho\mathbf{w} + d\rho = 0$$

$$\nabla_\xi \bullet \rho\mathbf{w} \otimes \mathbf{w} + \nabla_\xi P + (d+1)\rho\mathbf{w} = 0$$

$$\nabla_\xi \bullet \rho\left(\tfrac{1}{2}\mathbf{w} \bullet \mathbf{w} + h\right)\mathbf{w} + d\rho\left(\tfrac{1}{2}\mathbf{w} \bullet \mathbf{w} + h\right) + \rho\mathbf{w} \bullet \mathbf{w} = 0, \quad h = e + P/\rho.$$

A major difficulty in analyzing 2 or 3 dimensional flows is the complex behavior of scale invariant solutions. This complex behavior makes it unlikely that a generalization of the random choice method to higher dimensional flows is possible. Furthermore in many situations self-similar flows are unstable with respect to non-scale invariant perturbations. A classic example is Kelvin-Helmholtz instability in which a planar shear wave is unstable with respect to perturbations in the wave amplitude. This fact has profound implications for numerical solutions since the discretization will generally impose such a perturbation with length scales given by the grid. Nevertheless, scale invariant solutions are extremely useful in understanding the structure of complex flows.

Elementary Wave Solutions

The complexity of solutions to the scale invariant equations in higher space dimensions greatly reduces their utility in the interpretation of flow behavior. What we tend to see in such flows are sets of coherent features that retain their shapes for significant time intervals and move with recognizable velocities. This suggests the notion of an elementary wave, which is a solution to the conservation system that is both scale invariant and steady state in the sense that there is some velocity $\mathbf{v}$ (unknown a priori) so that under the Galilean transformation $\mathbf{y} = \mathbf{x} - \mathbf{v}t$, the flow becomes steady. For a scale invariant function $\mathbf{u}(\mathbf{x},t) = \mathbf{w}(\mathbf{x}/t)$, this means that for some velocity $\mathbf{v}$, $\mathbf{w}(\xi) = \mathbf{g}(\xi - \mathbf{v})$ and $\mathbf{g}(\eta)$ is homogeneous of degree 0, $\mathbf{g}(\alpha\eta) = \mathbf{g}(\eta)$, $\forall\ \alpha > 0$. The effect is to further reduce the differential system for the solution to that of solving a conservation law on the unit sphere S^{d-1}. We will not pursue this further in generality, but will now specialize to the case of the Euler equations. A key property of the Euler equations is Galilean invariance. Thus we see that all elementary wave solutions of the Euler equation are Galilean transformations of scale invariant solutions to the steady state Euler equations. For simplicity we restrict the discussion to two dimensional flows.

Steady State Self-Similar Flow

In two space dimensions the steady state Euler equations are:

$$(\rho u)_x + (\rho v)_y = 0$$

$$\left(\rho u^2 + P\right)_x + (\rho u v)_y = 0$$

$$(\rho u v)_x + \left(\rho v^2 + P\right)_y = 0$$

$$\left[\rho\left\{\frac{1}{2}\left(u^2+v^2\right)+h\right\}u\right]_x + \left[\rho\left\{\frac{1}{2}\left(u^2+v^2\right)+h\right\}v\right]_y = 0.$$

If we write $u = q\cos(\theta)$, $v = q\sin(\theta)$, $x = r\cos(\phi)$, $y = r\sin(\phi)$, we can rewrite the system as:

$$\left(r\rho q\cos(\varphi)\right)_r + \left(\rho q\sin(\varphi)\right)_\phi = 0$$

$$\left(r\rho q^2\cos^2(\varphi) + rP\right)_r + \left(\rho q^2\cos(\varphi)\sin(\varphi)\right)_\phi = P + \rho q^2\sin^2(\varphi)$$

$$\left(r\rho q^2\cos(\varphi)\sin(\varphi)\right)_r + \left(\rho q^2\sin^2(\varphi) + P\right)_\phi = -\rho q^2\cos(\varphi)\sin(\varphi)$$

$$\left[r\rho\left\{\frac{1}{2}q^2+h\right\}q\cos(\varphi)\right]_r + \left[\rho\left\{\frac{1}{2}q^2+h\right\}q\sin(\varphi)\right]_\phi = 0$$

$$\varphi = \theta - \phi$$

Steady State Self-Similar Flow Hugoniots (1)

If a discontinuous wave front makes the angle β with respect to a flow, then we can show that the Hugoniot conditions for the wave are:

$$\rho_0 q_0 \sin\beta_0 = \rho_1 q_1 \sin\beta_1 = m$$
$$m q_0 \sin\beta_0 + P_0 = m q_1 \sin\beta_1 + P_1$$
$$m q_0 \cos\beta_0 = m q_1 \cos\beta_1$$
$$m\left(\tfrac{1}{2}q_0^2 + h_0\right) = m\left(\tfrac{1}{2}q_1^2 + h_1\right).$$

If $m = 0$ we have a contact discontinuity where the pressure and normal component of velocity are continuous across the wave. If $m \neq 0$ we introduce the turning angle θ through the wave so that $\beta_1 = \beta_0 - \theta$, and we get the relations:

$$\tan\theta = \frac{\Delta P}{\rho_0 q_0^2 - \Delta P}\cot\beta_0$$

$$\tfrac{1}{2}q_0^2 + h_0 = \tfrac{1}{2}q_1^2 + h_1, \qquad h_1 - h_0 = \frac{V_0 + V_1}{2}\left(P_1 - P_0\right).$$

Steady State Self-Similar Flow Hugoniots (2)

The variable m is the mass flux across the front as described previously. If the shock is stable in the sense of Lax, and 0 denotes thermodynamically ahead state of the shock, then $\rho_0 c_0 < m < \rho_1 c_1$ so that:

$$M_0 \sin\beta_0 = \frac{m}{\rho_0 c_0} > 1, \quad \text{and} \quad M_1 \sin\beta_1 = \frac{m}{\rho_1 c_1} < 1,$$

where $M_i = q_i/c_i$ is the shock Mach number. It follows that Lax stability requires that the flow ahead of the shock be supersonic, while the flow behind the shock may be either supersonic or subsonic. Note that as the shock approaches a normal shock, the Mach number behind the shock must eventually be less than one.

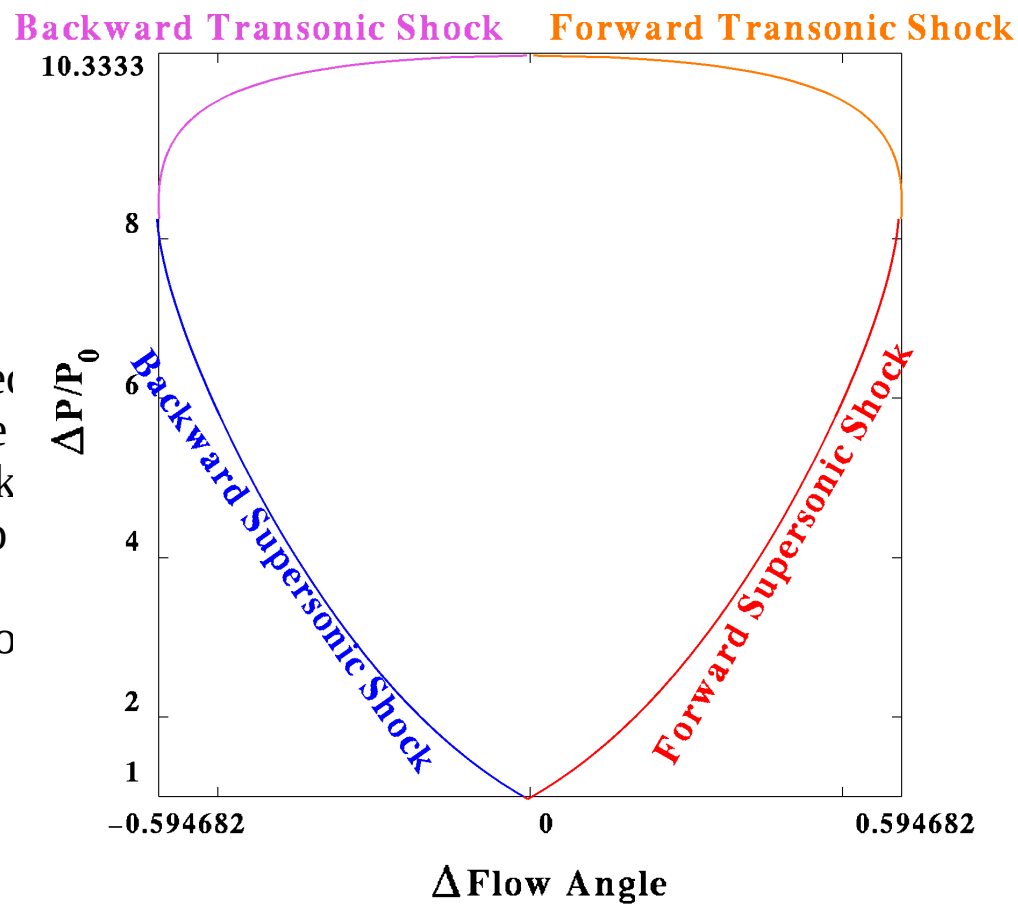
For a perfect gas we can write the turning angle explicitly in terms of the flow states:

$$\tan\theta = \pm \frac{\Delta P/P_0}{\gamma M_0^2 - \Delta P/P_0} \sqrt{\frac{\left(1+\mu^2\right)\left(M_0^2 - 1\right) - \Delta P/P_0}{\Delta P/P_0 + \left(1+\mu^2\right)}}$$

$$M_1^2 - 1 = \frac{c_0^2}{c_1^2}\left[\left(M_0^2 - 1\right) - \frac{\Delta P}{P_0}\left(\frac{P_1 + P_0}{P_1 + \mu^2 P_0}\right)\right]$$

Shock Polars (1)

For self-similar shocks the Hugoniot conditions derived on the previous slides suffice to describe the flow state about the shock. The relation between the pressure and turning angle is particularly useful. This function is called a shock polar. For a fixe ahead state it gives the flow angle of the streamline behind the shock as a function of the pressure jump across the shock. The plot to the right shows a representative plot o a shock polar for a perfect gas equation of state. The following observations are important properties of the shock polar.



Shock Polars (2)

- The shock polar forms a bounded loop with the maximum pressure behind the shock corresponding to a normal shock advancing into the ahead state region.
- The shock polar is divided into two symmetric branches depending on whether the flow through the shock is turned in the counterclockwise (forward or positive branch) or clockwise (backward or negative branch).
- Each branch is divided into two sub-branches according to whether the flow behind the shock is supersonic (supersonic shock) or subsonic (transsonic shock). Recall that the ahead state is always supersonic.
- The division points where the flow changes from supersonic to transonic are called the sonic points. A sonic transition occurs only once for a perfect gas equation of state.
- The shock polars possess local extrema in the turning angle. Again these occur at a single pressure for a perfect gas equation of state.
- For a perfect gas equation of state, the flow at the maximum turning angle is always transsonic, but this occurs at a pressure that is often close to the sonic point pressure.

Prandtl-Meyer Waves (1)

In order to complete the discussion of steady-state self-similar flows we need to examine possible smooth solutions to the self-similar flow equations. Returning to the polar version of the Euler equations in two space dimensions, we can write this system in smooth flow as:

$$\left\{\cos\varphi\frac{\partial}{\partial r}+\frac{\sin\varphi}{r}\frac{\partial}{\partial\phi}\right\}\theta \;+\; \frac{1}{\rho q^2}\left\{-\sin\varphi\frac{\partial}{\partial r}+\frac{\cos\varphi}{r}\frac{\partial}{\partial\phi}\right\}P \;=\; 0$$

$$\left\{\cos\varphi\frac{\partial}{\partial r}+\frac{\sin\varphi}{r}\frac{\partial}{\partial\phi}\right\}P \;+\; \frac{\rho q^2c^2}{q^2-c^2}\left\{-\sin\varphi\frac{\partial}{\partial r}+\frac{\cos\varphi}{r}\frac{\partial}{\partial\phi}\right\}\theta \;=\; 0$$

$$\left\{\cos\varphi\frac{\partial}{\partial r}+\frac{\sin\varphi}{r}\frac{\partial}{\partial\phi}\right\}\left[\tfrac{1}{2}q^2+h\right] \;=\; 0$$

$$\left\{\cos\varphi\frac{\partial}{\partial r}+\frac{\sin\varphi}{r}\frac{\partial}{\partial\phi}\right\}S \;=\; 0.$$

It is easy to check that the two differential operators in the above system of equations correspond to derivatives in the direction of the flow and the direction orthogonal to the flow:

$$\cos\varphi\frac{\partial}{\partial r}+\frac{\sin\varphi}{r}\frac{\partial}{\partial\phi}=\cos\theta\frac{\partial}{\partial x}+\sin\theta\frac{\partial}{\partial y},\quad -\sin\varphi\frac{\partial}{\partial r}+\frac{\cos\varphi}{r}\frac{\partial}{\partial\phi}=-\sin\theta\frac{\partial}{\partial x}+\cos\theta\frac{\partial}{\partial y}.$$

Prandtl-Meyer Waves (2)

You can show that this steady flow system written in the form on the previous slide is hyperbolic if and only if the flow is supersonic $q > c$. In this case we can write the equations in a simplified characteristic form by introducing the Mach angle A defined by $\sin(A) = c/q$. In this case the system can be written:

$$\left\{\cos(\varphi \pm A)\frac{\partial}{\partial r} + \frac{\sin(\varphi \pm A)}{r}\frac{\partial}{\partial \phi}\right\}P \;\pm\; \rho q^2 \tan A\left\{\cos(\varphi \pm A)\frac{\partial}{\partial r} + \frac{\sin(\varphi \pm A)}{r}\frac{\partial}{\partial \phi}\right\}\theta \;=\; 0$$

$$\left\{\cos\varphi\frac{\partial}{\partial r} + \frac{\sin\varphi}{r}\frac{\partial}{\partial \phi}\right\}\left[\tfrac{1}{2}q^2 + h\right] \;=\; 0$$

$$\left\{\cos\varphi\frac{\partial}{\partial r} + \frac{\sin\varphi}{r}\frac{\partial}{\partial \phi}\right\}S \;=\; 0.$$

The differential operators in the first two equations are just the derivatives in the directions that make an angle A with the flow direction.

$$\cos(\varphi \pm A)\frac{\partial}{\partial r} + \frac{\sin(\varphi \pm A)}{r}\frac{\partial}{\partial \phi} = \cos(\theta \pm A)\frac{\partial}{\partial x} + \sin(\theta \pm A)\frac{\partial}{\partial y}.$$

Prandtl-Meyer Waves (3)

Suppose now that we have a scale invariant flow, i.e. the flow variables are independent of r. Then we obtain the system:

$$\rho q^2 \frac{c^2/q^2}{1-c^2/q^2}\cos\varphi\frac{\partial\theta}{\partial\phi} + \sin\varphi\frac{\partial P}{\partial\phi} = 0$$

$$\rho q^2 \sin\varphi\frac{\partial\theta}{\partial\phi} + \cos\varphi\frac{\partial P}{\partial\phi} = 0$$

$$\frac{\partial}{\partial\phi}\left[\tfrac{1}{2}q^2 + h\right] = 0$$

$$\frac{\partial S}{\partial\phi} = 0.$$

We see immediately that the entropy and the quantity $q^2/2+h$ are constant in a self-similar flow region. If θ and P are not constant, them we can divide the first two equations to obtain the relation: $c^2/q^2 = \sin^2\varphi$. This shows both that the flow in self-similar region must be supersonic, and that in such a region, the flow angle θ, the direction angle ϕ, and the Mach angle A are related by the formula:

$$\varphi = \theta - \phi = \pm A$$

$$\phi = \theta \mp A.$$

Prandtl-Meyer Waves (4)

Since a non-constant self similar flow must be supersonic we can rewrite the equations in characteristic form:

$$\frac{\partial \theta}{\partial \phi} = \pm \frac{\cot A}{\rho q^2} \frac{\partial P}{\partial \phi}.$$

If we eliminate ϕ we get a formula analogous to that obtained for the relation between flow velocity and pressure in a one dimensional rarefaction wave:

$$\theta - \theta_0 = \pm \int_{P_0}^{P} \frac{\cot A}{\rho q^2} dP \Bigg|_{S, \frac{1}{2}q^2 + h}.$$

For a perfect gas equation of state we can compute the integral on the right to obtain:

$$\int_{P_0}^{P} \frac{\cot A}{\rho q^2} dP \Bigg|_{S, \frac{1}{2}q^2 + h} = \left(A_0 + \mu^{-1} \arctan\left(\mu \cot A_0\right)\right) - \left(A + \mu^{-1} \arctan\left(\mu \cot A\right)\right).$$

Prandtl-Meyer Waves (5)

Finally given a single point in a self-similar flow region we can solve for the flow in the entire region using the formulas:

$$\theta - \theta_0 = \pm \int_{P_0}^{P} \frac{\cot A}{\rho q^2} dP \Bigg|_{S, \frac{1}{2}q^2 + h}$$

$$\phi = \theta \mp A$$

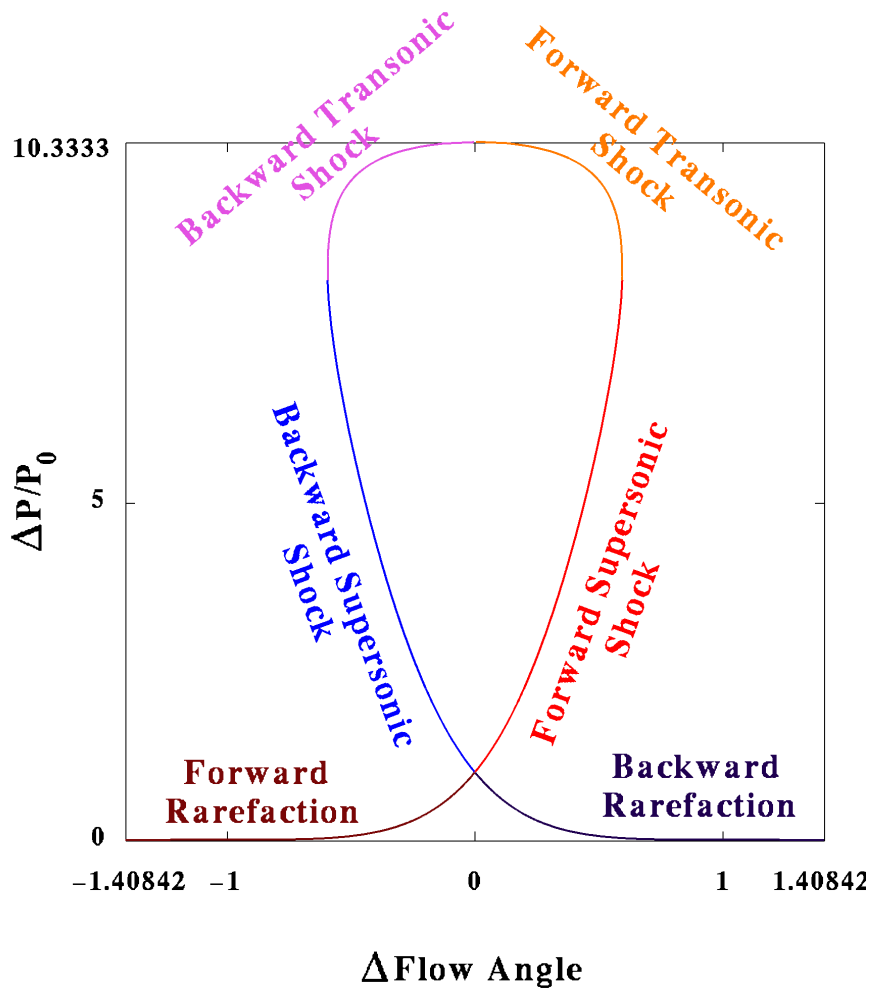
$$S = S_0$$

$$\frac{1}{2}q^2 + h = \frac{1}{2}q_0^2 + h_0 .$$

By solving this system in terms of the position angle ϕ we obtain the formula for the flow in a fan region emanating from the origin. Such a wave is called a Prandtl-Meyer wave. These waves play the same role in the solution of a supersonic steady state Riemann problem (to be discussed next) as one dimensional rarefaction waves. Just as in one dimensional flow they can be joined with the shock polars to produce a twice continuously differentiable wave curve that describes the full set of states that can be connected to a given state by either a steady state shock or a Prandtl-Meyer wave.

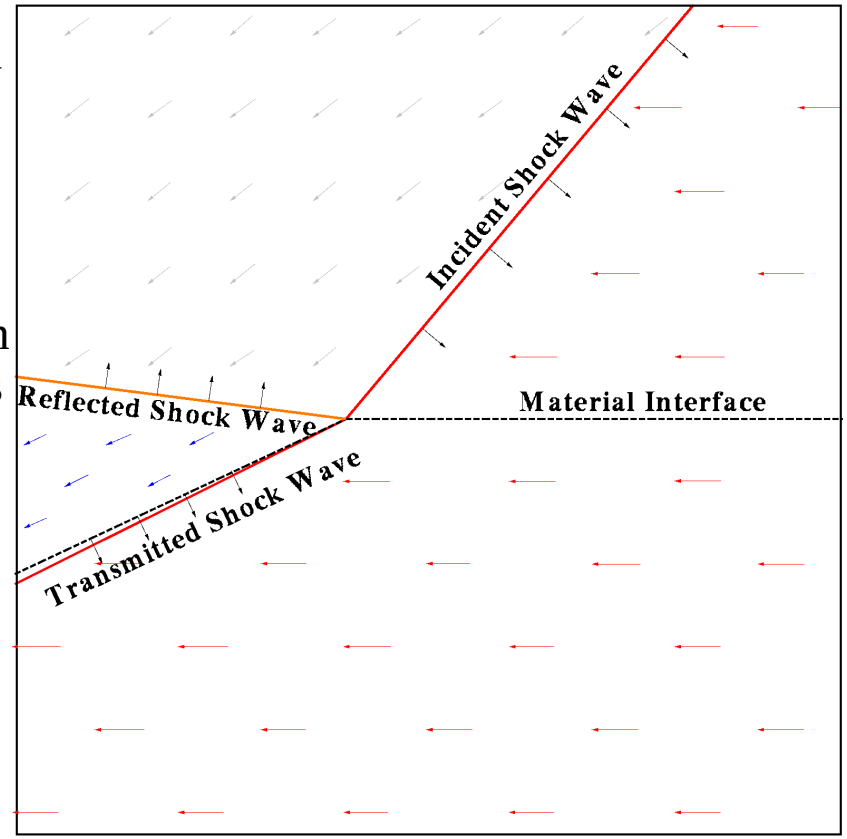
The Steady State Wave Curve

The plot on the right show the full wave curve combining both the shock and rarefaction portions. Intersections of these wave curves will be used to compute the solutions to wave interactions, shock refractions in particular.



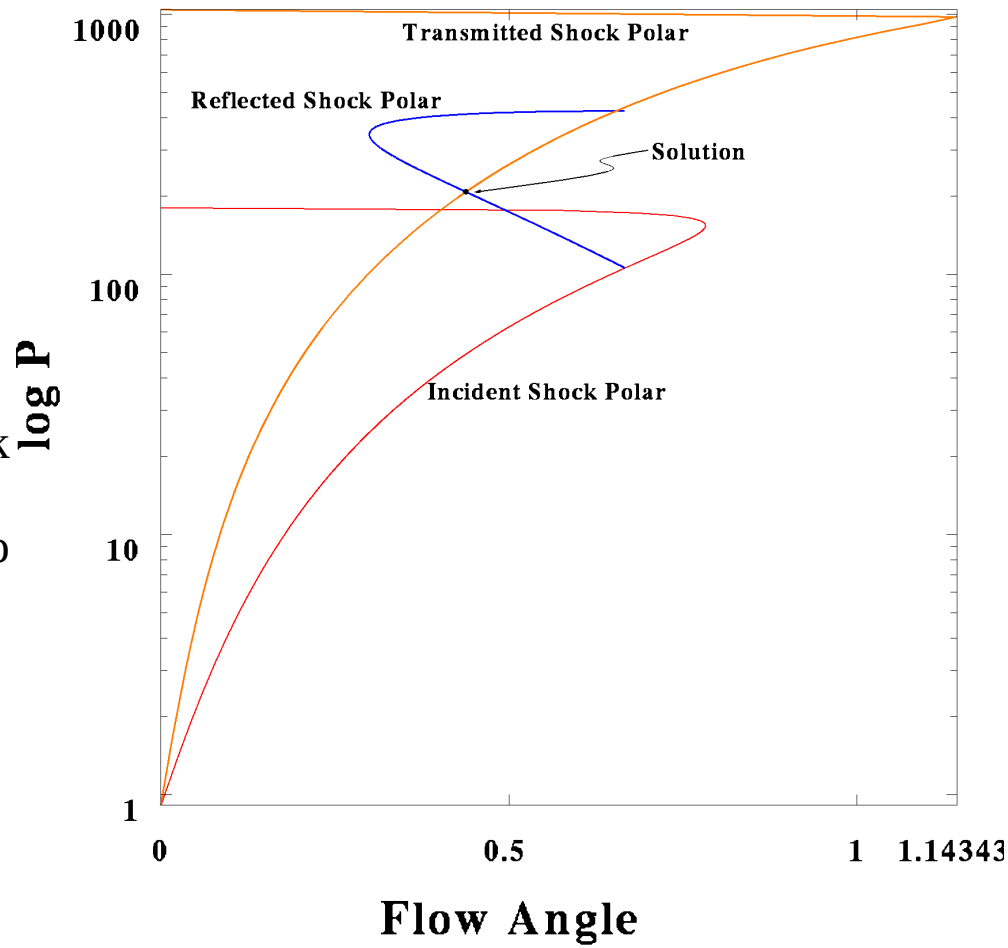
Shock Refractions

Consider a shock wave incident on an interface between two different materials as indicated in the figure below. This figure shows a Mach 10 shock in air incident on a material interface with sulfur-hexaflouride. The shock is refracted by the material interface into reflected and transmitted shocks. The material interface is also deflected by the shock wave. The black arrows at the shock fronts show the direction of propagation of the shocks, while the colored arrows show the flow velocity relative to the point of refraction. The flow state behind the incident shock together with the unshocked flow in the SF_6 serve as initial data for a supersonic steady state Riemann problem, the solution of which gives the reflected and transmitted shock data.



Shock Refractions

The shock polar diagram for the refraction shown on the previous slide is given below. Using the state on the opposite side of the material interface from the incident shock and the state behind the incident shock as Riemann problem data, the downstream flow states are completely determined by the intersection (provided it exists) of the two corresponding shock polars. Actually this diagram indicates that the solution lies above the mechanical equilibrium point (where all three shock polars would intersect) so that we would expect this wave configuration to be unstable with respect to a sufficiently large perturbation and the single point refraction would bifurcate into a more complex configuration.



Types of Elementary Waves

As we have just seen, one type of elementary wave is given by the refraction of a shock through a material interface. Basically all elementary waves in gas dynamics can be divided into two types, so called supersonic elementary waves that corresponding to a binary interaction between two wave fronts, either shock on shock or shock on material interface, or transonic elementary waves that correspond to a dynamic splitting of a wave. The best know example of the latter type is regular Mach reflection in which a shock incident on a wall must bifurcate into a pair of shocks and a contact discontinuity so that the flow near the shock can satisfy the wall boundary condition that the flow must be parallel to the wall.

It should be emphasized that elementary waves are only the building blocks out of which more complex configurations are composed. Furthermore exact elementary waves will generally only occur as asymptotic flow states near a point of interaction. Also the occurrence of subsonic flow regions near an interaction can lead to a loss of local self-similarity, and the flow becomes fully multidimensional. Nevertheless an understanding of such waves is extremely useful in interpreting flow phenomena and are also useful in numerical algorithms for tracking the interaction of wave fronts.

Exercises

1. Derive the conservation form of the scale invariant Euler equations.

2. Show that for a perfect gas, the steady state Riemann function for a Prandtl-Meyer wave is:

$$\int_{P_0}^{P} \frac{\cot A}{\rho q^2} dP \Bigg|_{S, \frac{1}{2}q^2 + h} = \left(A_0 + \mu^{-1} \arctan\left(\mu \cot A_0\right)\right) - \left(A + \mu^{-1} \arctan\left(\mu \cot A\right)\right).$$

3. Show that for a given ahead state and a perfect gas equation of state the sonic point pressure is unique.

Further Reading

- J. Glimm and D. H. Sharp, *A S-matrix Theory for Classical Nonlinear Physics*, Found. Phys. **16**, 126--141, 1986.
- R. Courant and K.O. Friedrichs, *Supersonic Flow and Shock Waves* , Springer-Verlag 1948.
- J. Glimm, J. Grove, W. B. Lindquist, O. McBryan, and G. Tryggvason, *The Bifurcation of Tracked Scalar Waves*, SIAM J. Sci. Stat. Comp., **9**, pp. 61-79, 1988.
- J. W. Grove, *The Interaction of Shock Waves with Fluid Interfaces*, Adv. Appl. Math. **10**, pp. 201-227, 1989.
- J. Grove and R. Menikoff, *The Anomalous Reflection of a Shock Wave at a Material Interface*, J. Fluid Mech. **219**, pp. 313-336 1990.
- J. Glimm, C. Klingenberg, O. McBryan, B. Plohr, D. Sharp, and S. Yaniv, *Front Tracking and Two-Dimensional Riemann Problems*, Adv. Appl. Math. **6**, pp. 259-290, 1985.

Implementing Wave Interactions in Front Tracking

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Numerical Implementation of Front Tracking Wave Interactions

Examples of Dynamic Wave Bifurcations

Examples of Wave Interactions

Classification of Numerical Wave Interactions

Propagation of a Regular Shock Refraction Node (1)

Propagation of a Regular Shock Refraction Node (2)

Propagation of a Regular Shock Refraction Node (3)

Propagation of a Regular Shock Refraction Node (4)

Remarks on the Propagation of Refraction Nodes

Topology Changing Interactions (1)

Topology Changing Interactions (2)

Node-Node Interactions (1)

Node-Node Interactions (2)

Node-Node Interactions (3)

Scalar-Vector Wave Collisions (1)

Scalar-Vector Wave Collisions (2)

Scalar-Vector Wave Collisions (3)

Scalar-Vector Wave Collisions (4)

Scalar-Vector Wave Collisions (5)

Exercises

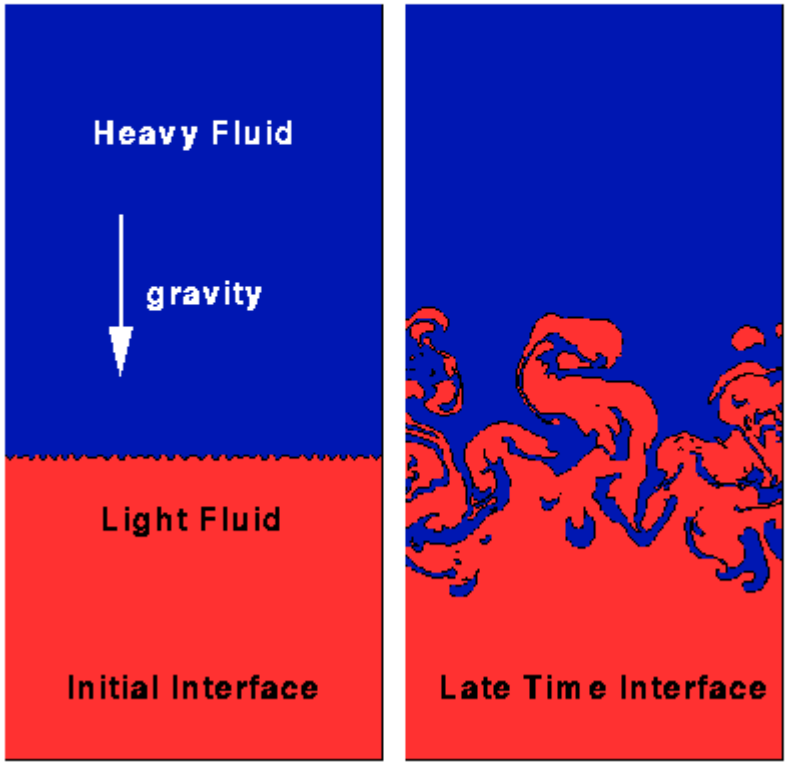
Further Reading

Numerical Implementation of Front Tracking Wave Interactions

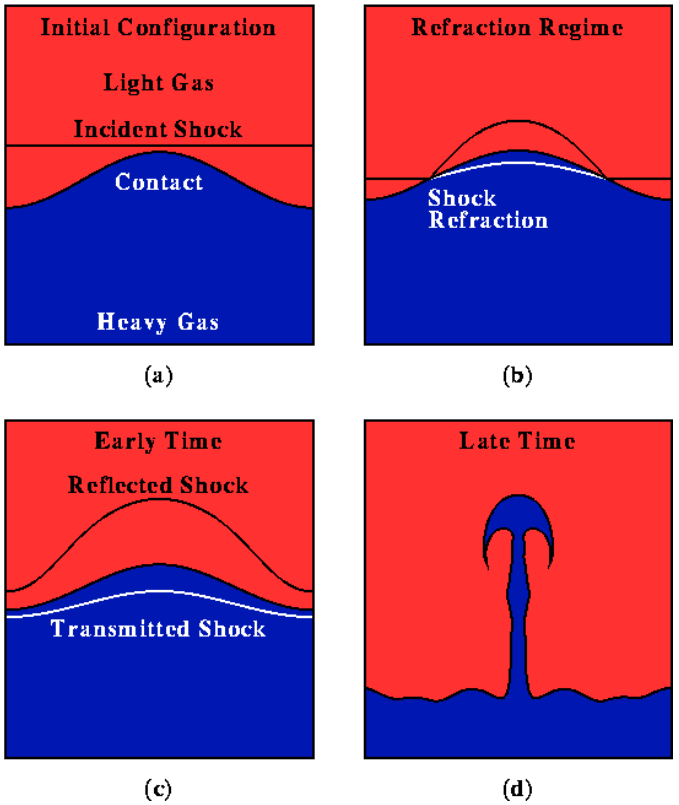
Front tracking provides a method of achieving enhanced resolution of selected strong waves. It eliminates numerical diffusion and provides a mechanism for applying wave propagation algorithms in coordinate systems that are naturally aligned with the fronts, and has the ability to apply special physics to the fronts. One of the prices for this ability is the need to detect, diagnose, and resolve interactions between tracked fronts. This lecture will discuss some of the structural aspects of the front tracking method as implemented in the *FronTier* for the resolution of wave interactions. The figures on the next few slides illustrates the type of complex wave bifurcations that are handled by *FronTier*.

Examples of Dynamic Wave Bifurcations

Rayleigh-Taylor Instability

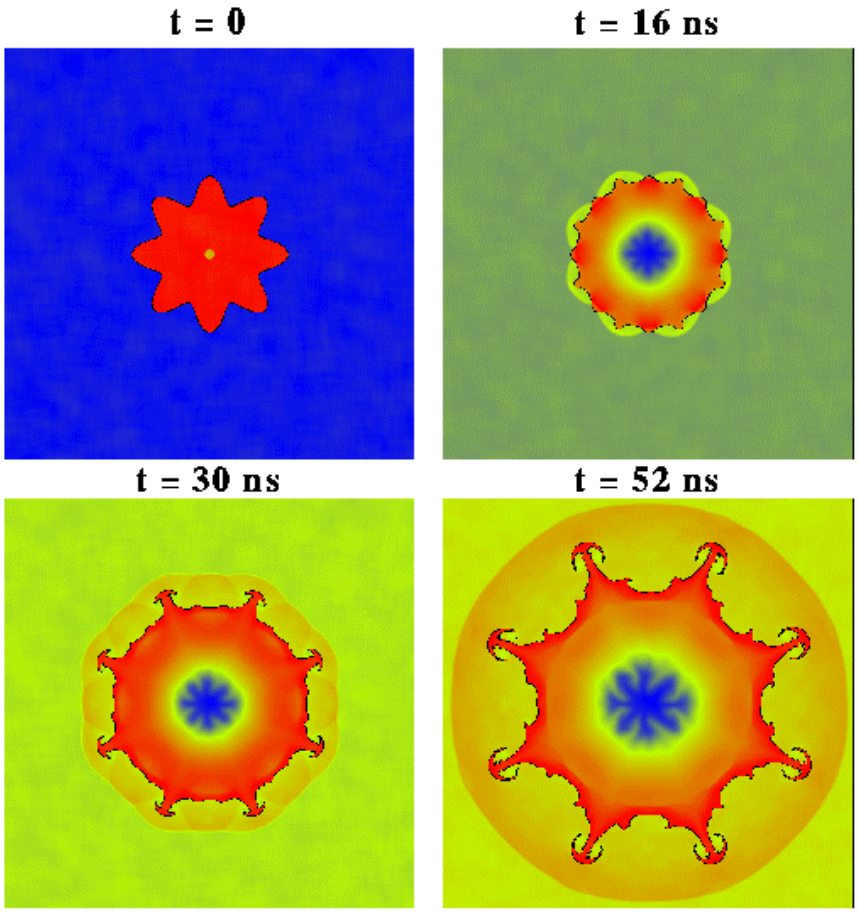


Richtmyer-Meshkov Instability
(reflected shock case)

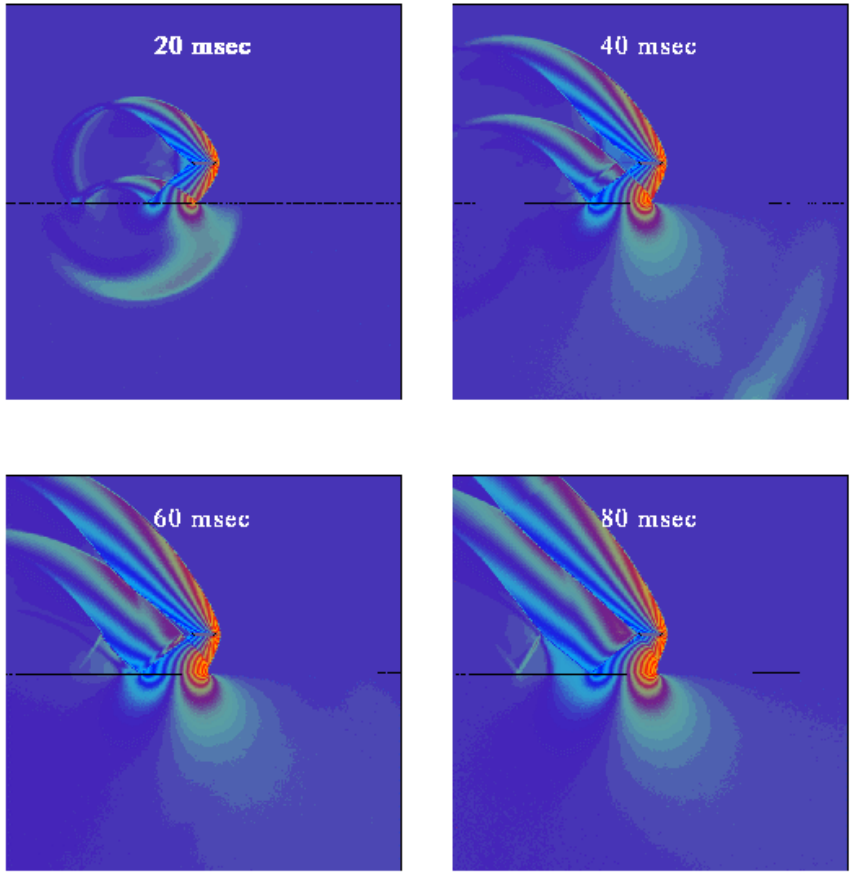


Examples of Wave Interactions

Supernova Explosion



Supersonic Missile Above Water

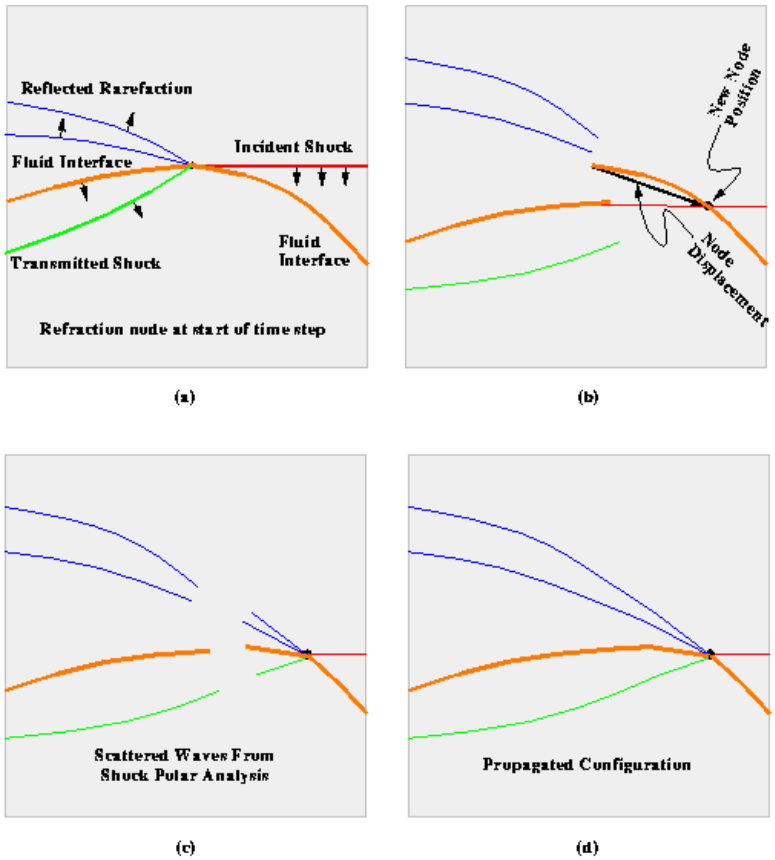


Classification of Numerical Wave Interactions

The way FronTier handles wave interactions depends on the nature of the waves involved in the interaction. We define as a vector wave, any wave belonging to a characteristic family with a nontrivial light cone. For gas dynamics such waves are shock fronts and acoustic waves. The latter most often are the tracked edges of rarefaction regions. A scalar wave is a wave belonging to a characteristic family with a degenerate light cone. Such waves are also said to be linearly degenerate, and in gas dynamics are slip lines, contact discontinuities, material interfaces, or in general a combination of all three. Each scalar wave moves with the local fluid velocity. In addition we distinguish interactions between those that produce a change in the tracked front topology and those that do not. The latter case corresponds to the pointwise interaction of wave fronts and is implemented in terms of the node propagation algorithm for the node that represents the point of interaction. This class is subdivided into individual cases depending on whether the wave interaction at the node is due to two shocks crossing, one shock overtaking another, a shock refracting through a material interface, a triple point junction of contact discontinuities, or a Mach triple point. We will discuss the shock refraction node propagation algorithm. The other nodes are similar except for the contact triple point, which has a complex structure. The waves that change the interface topology are due to collisions of separate wave fronts, and are handled by special code that identifies the wave structure and reconstructs a new interface topology to reflect the resolution of the wave interaction.

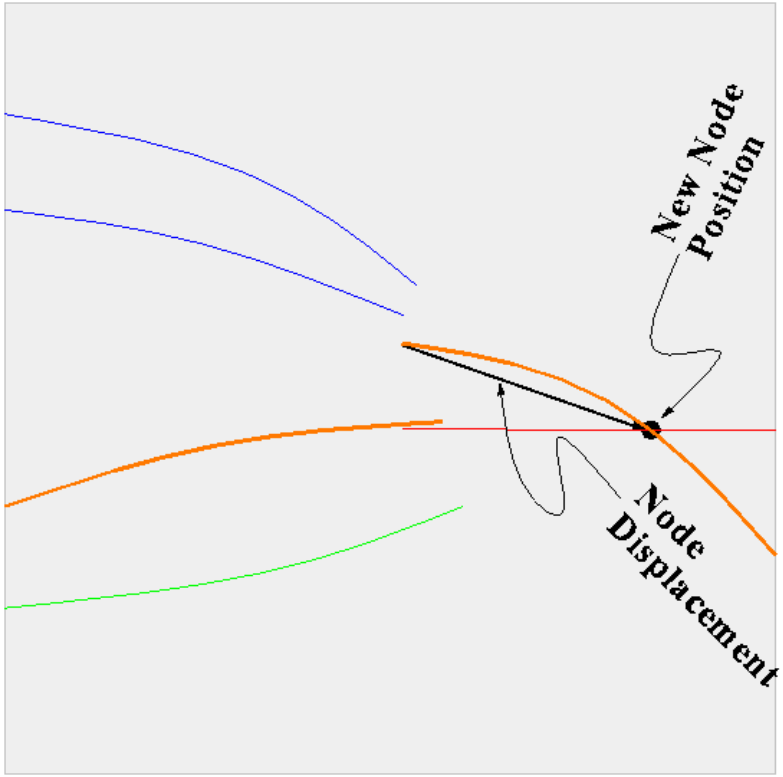
Propagation of a Regular Shock Refraction Node (1)

The figure on the right shows the separate steps in the propagation algorithm for a regular shock refraction node. Figure (a) shows the configuration at the start of the time step. The first step is to propagate each wave front separately, ignoring the interaction at the node. For this purpose the node is treated as six separate interior points, one for each curve entering the node. The point propagation algorithm, using one sided information at the node, determines separate states and positions for hypothetical points that lay in the interior of each curve and resided at the node position. This step leads to the configuration in (b) in which the wave fronts have become logically detached from each other.



Propagation of a Regular Shock Refraction Node (2)

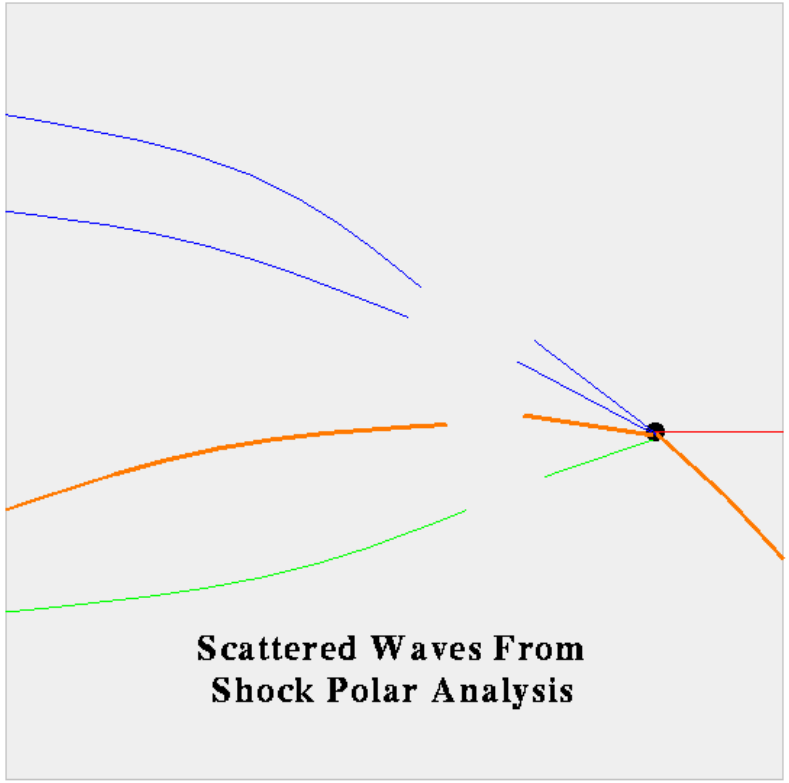
Once we have propagated the individual wave fronts at the node separately leading to the configuration in (b), we use the fact that a regular refraction node is locally a supersonic elementary wave where information flows from the upstream side into the downstream region. We identify the upstream incident shock and material interface through the use of integer labels attached to the curves at the node indicating whether that curve is an incident shock, a reflected shock or rarefaction edge, a transmitted shock, an upstream contact, or a downstream contact. Once the two upstream curves are identified we compute the intersection between their propagated sections and use the displacement between this point and the old node position to approximate the node velocity.



(b)

Propagation of a Regular Shock Refraction Node (3)

The node velocity gives the velocity transformation between the unsteady frame of the computation and the local steady frame of the node. We determine the upstream states at the new node by linear interpolation along the upstream propagated segments. This data when transformed into the steady from of the node is used as data for a steady state Riemann problem as described in the previous lecture. The solution from this Riemann problem gives the angles of the reflected and transmitted waves, the angle of the deflected contact, and the state values at the node on either side of these waves. We next trim off the sections of the upstream waves behind their point of intersection and install wave fragments corresponding to the scattered wave information as given by the Riemann problem solution.

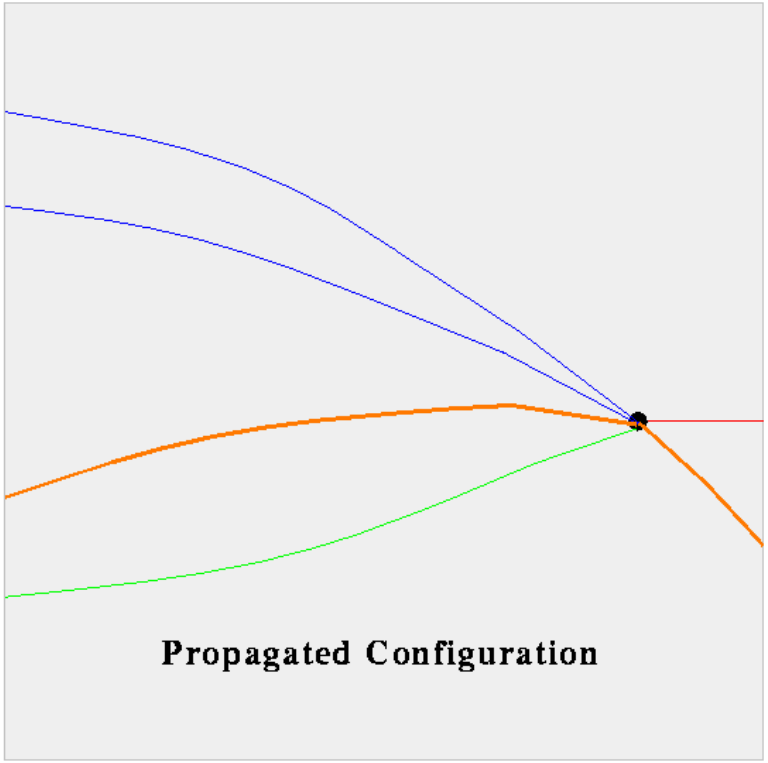


(c)

Propagation of a Regular Shock Refraction Node (4)

The lengths of the scattered wave segments is determined by the velocity of a sound wave in each section. This sets an upper bound on the distance that a wave can propagate out from the node in a single time step. The result is a configuration like that shown in figure (c). Finally the inserted sections are joined with their counterparts behind the node and the propagation is complete.

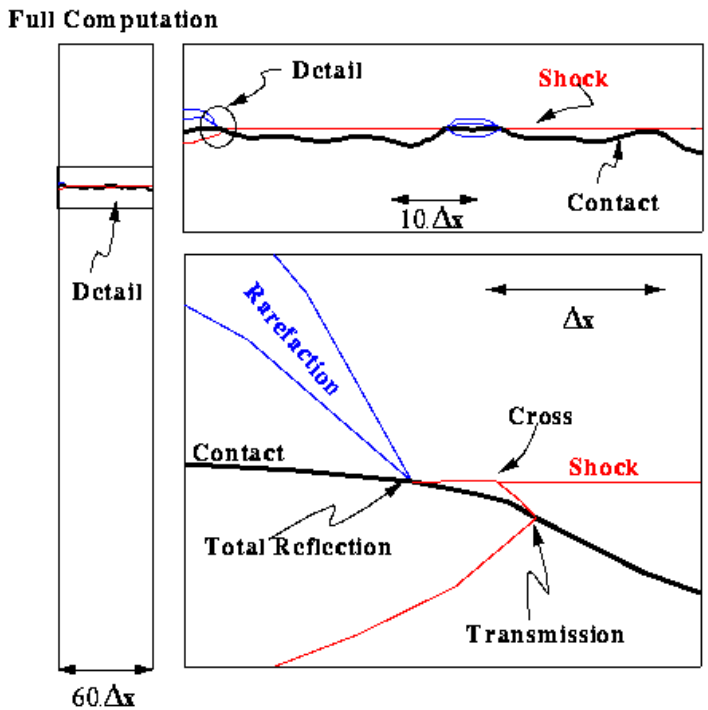
Note that this algorithm uses a local steady state approximation, but is repeated each time step with new data. This allows upstream effects such as changing geometry of the interacting waves to be propagated downstream. The effect is a fully dynamic algorithm for the propagation of the node.

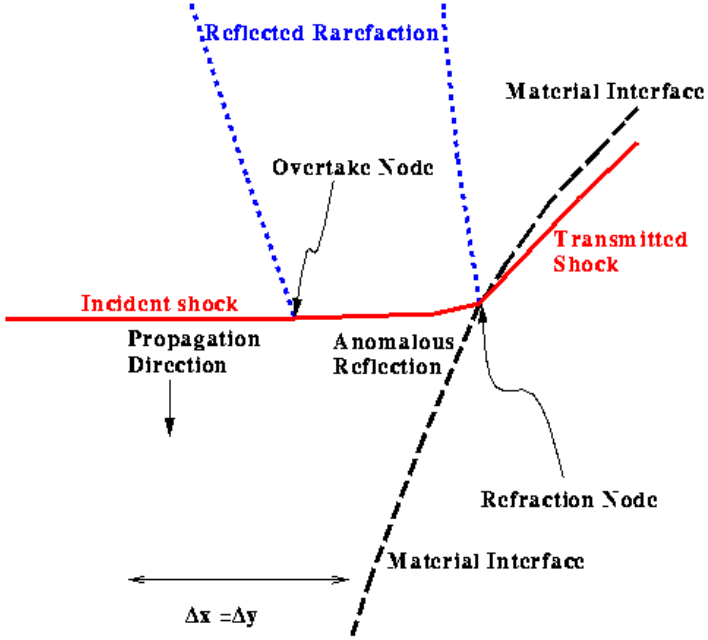


(d)

Remarks on the Propagation of Refraction Nodes

The existence of a regular refraction node depends upon the existence of a solution to the supersonic steady state Riemann problem corresponding to the local state of the flow around the node. In the event that this Riemann problem has no solution, the flow state at the node can not stay self-similar and the node breaks up into a complex configuration. The two figures below show just two of the possible irregular configurations that can be produced.





Topology Changing Interactions (1)

Wave interactions that change the topology of the interface are divided into two broad classes, interactions between nodes, i.e. interactions that occur when two or more nodes with a common curve collide, and tangles, where the discrete motion of two curves leads to the production of intersection points in the interior of the curves. Tangles are further classified by whether they are scalar-scalar, scalar-vector, or vector-vector interactions, according to whether two contacts become tangled, a contact tangles with a shock front, or two shocks become tangled. Both the node-node and curve-curve interactions use similar theory for their resolution, they primarily differ in how they are detected. In both cases the procedure follows the same basic pattern:

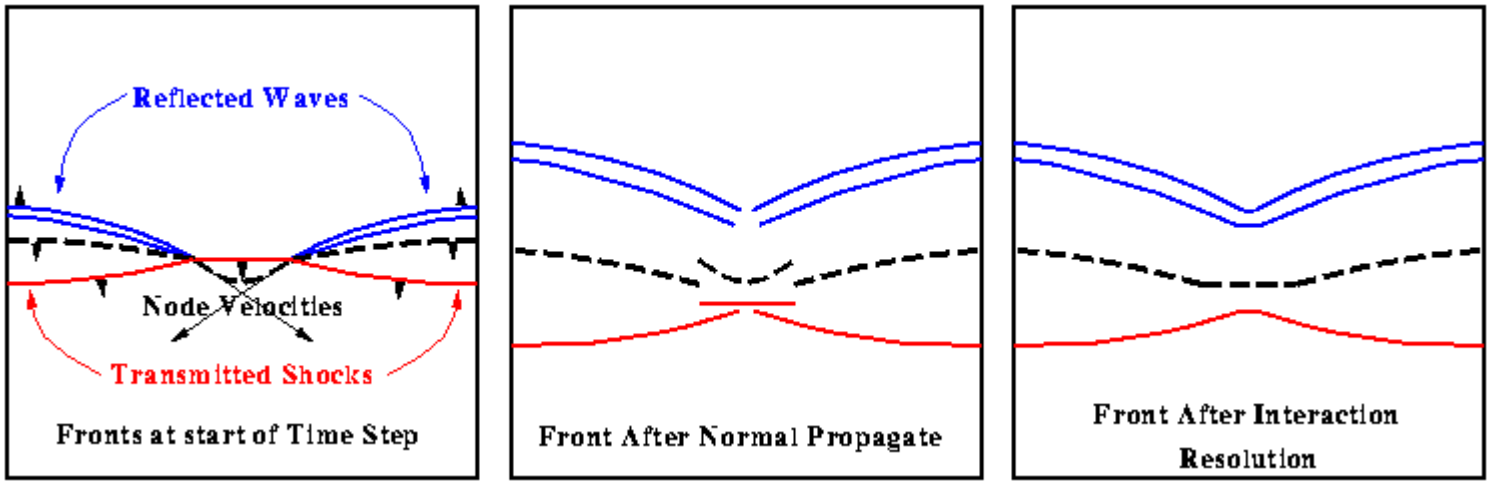
- Detect that an interaction has occurred
- Apply a sorting routine to identify the type of interaction based on the number and types of the waves involved.
- Use an interaction specific algorithm to compute the scattered waves from the interaction and install a representation of the solution into the tracked data structures.

Topology Changing Interactions (2)

One method that can be used to treat an interaction is to turn off tracking of the involved waves and use shock capturing to resolve the interaction. This necessarily results in the loss of resolution in the solution since the waves will spread out over regions that are on the order of three to five mesh blocks wide. However this provides a robust alternative for situations where the algorithm for a specific interaction is not implemented or the geometry of the waves is too complex for practical tracking. There is one situation where we do not allow tracking to be turned off. This is for interfaces across which there is a change in the material equation of state. In the current implementation *FronTier* assumes that no microscopic mixing occurs between separate fluid species, and each region of pure material is surrounded by a tracked material interface. This means that in any attempt to resolve an interaction by turning off tracking, we are free to remove any vector (shock) waves, but not to remove material interfaces. This has proven to be adequate for most purposes. We are currently developing a coupling of the tracking method with multi-material models of the flow field that will allow us to replace a tracked material interface by a diffusion layer, at the cost of spreading the mixing zone five to seven mesh blocks or more.

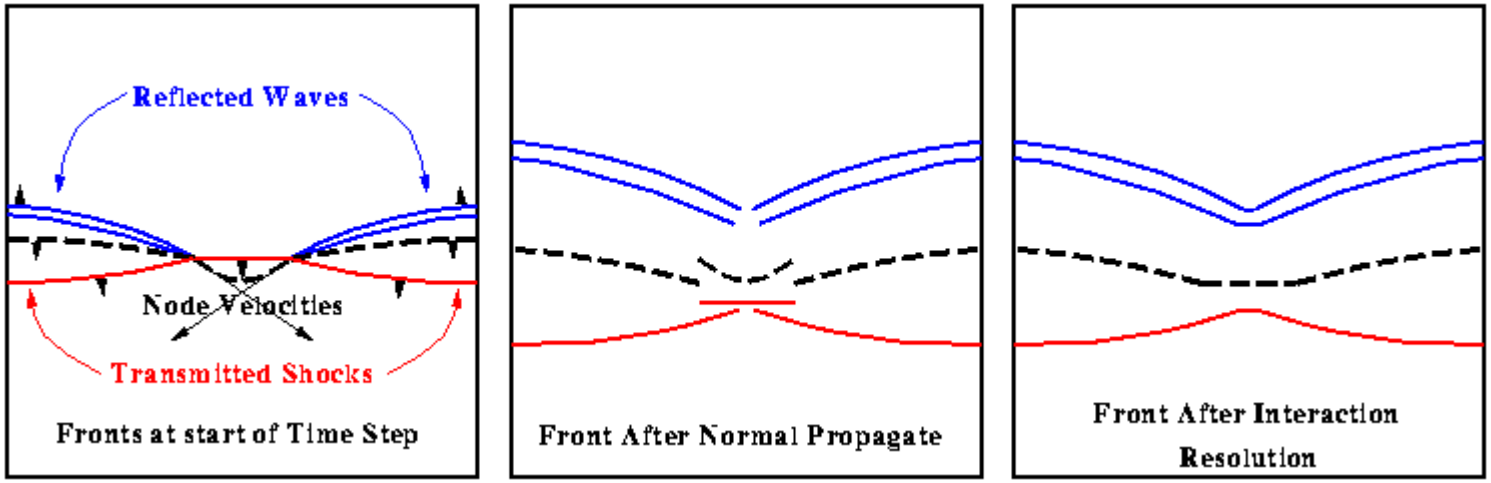
Node-Node Interactions (1)

The figure below shows an example of a node-node interaction that occurs in the simulation of Richtmyer-Meshkov instability. A shock is refracting through a material interface producing a pair of regular refraction nodes. In this case the scattered wave pattern at the nodes consist of a transmitted shock and a reflected Prandtl-Meyer wave. Near the time when the shock reaches the opposite end of the material interface, the velocities of the two nodes are directed towards each other, and are of sufficient magnitude to that in a single time step the nodes propagate past each other.



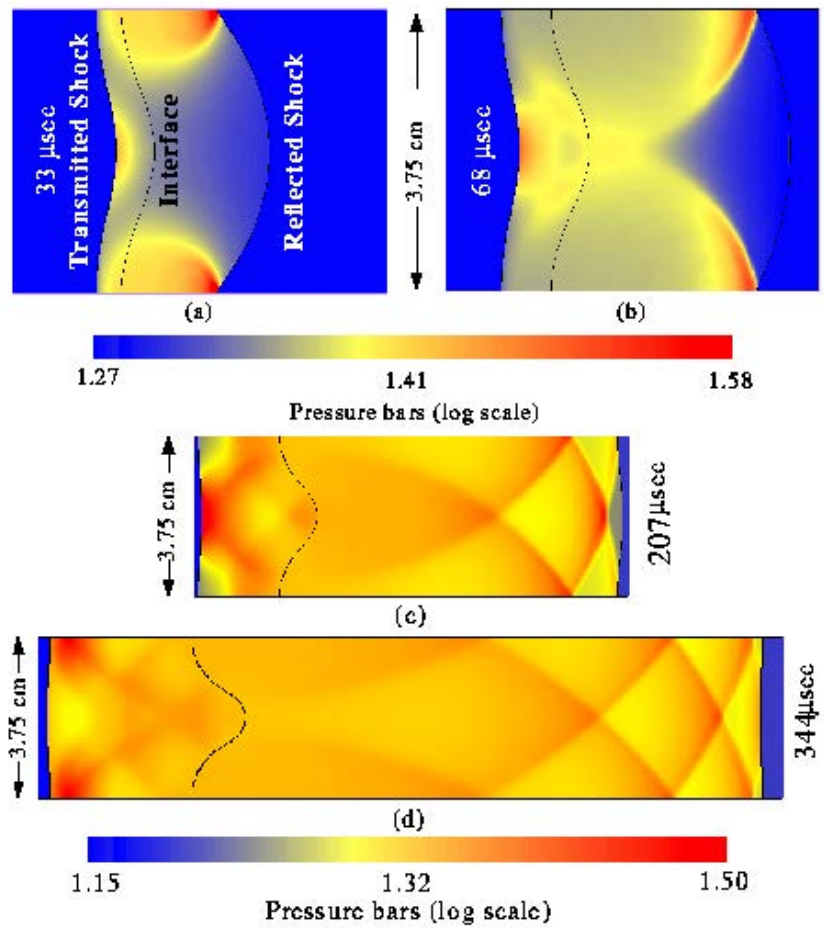
Node-Node Interactions (2)

This interaction is detected during the node propagation phase for the two nodes. The propagated segments of the upstream curves do not intersect as shown in the middle picture below. This failure signals a wave interaction. By examining the incoming waves we determine that they correspond to a pair of regular refraction nodes and that the interaction consists of a collision of the two nodes followed by a scattering of the interacting waves. The model for the resolution of this interaction is to connect the sets of reflected and transmitted waves, and to reconnect the material interface into a single curve. The actual physical interaction would produce additional waves directed towards the material interface. These waves are captured on the interior grid.



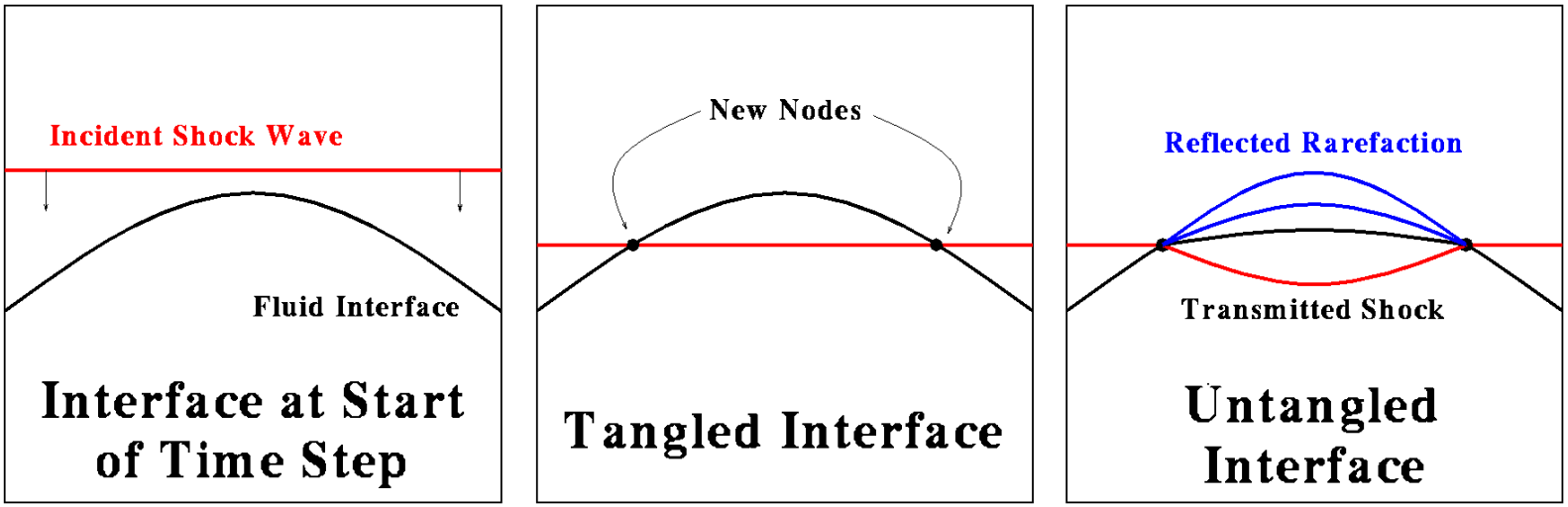
Node-Node Interactions (3)

The figure on the left shows the sort of captured waves that can be produced by the refraction node interaction. In this case the node-node interaction has occurred across the periodic boundaries at the top and bottom of the computational domain. This interaction produces a shock front that propagates towards the middle of the computational domain. Where this wave overtakes the reflected shock we would have an overtake node if the configuration were tracked. Since the only the leading edges of the reflected and transmitted shocks are tracked, this overtake node is not an explicit tracked feature, but is instead a captured wave. This mixing of tracked and captured features is an important aspect of the front tracking method.



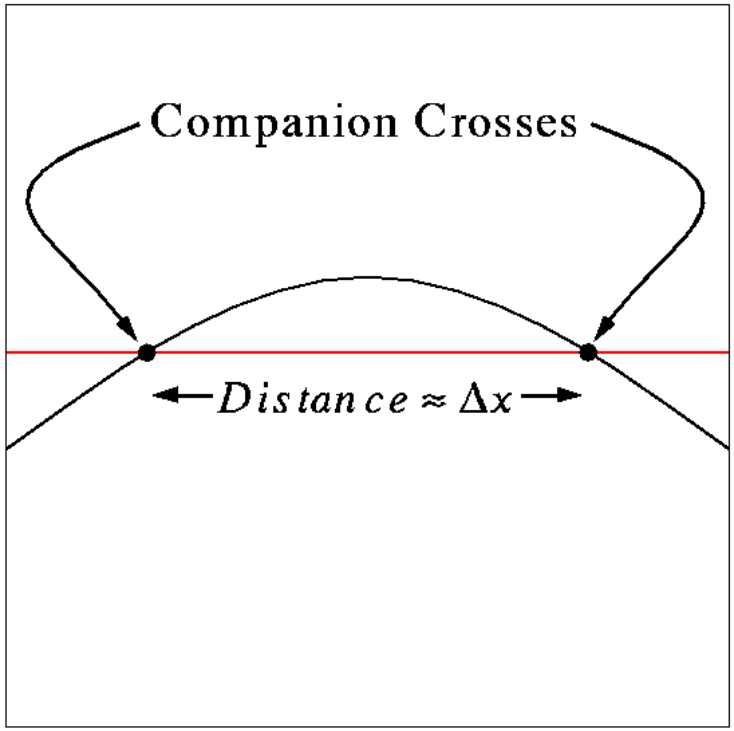
Scalar-Vector Wave Collisions (1)

The simulation shown on the previous slide came from a FronTier simulation of Richtmyer-Meshkov instability. This is a fluid interface instability caused by the acceleration of a material interface by a shock wave. The initialization of such a simulation consists of the installation of a shock at a positive distance from the material interface. During the first few time steps the two waves are separate and do not interact. At some point the shock reaches the material interface, becomes tangled with it, and leads to a scalar-vector untangle problem. The figure below shows a schematic representation of the resolution of the untangle.



Scalar-Vector Wave Collisions (2)

The first step in resolving the scalar-vector interaction is to identify the pair of interacting crosses. Recall that in two space dimensions the intersections function returns a list of discrete interface crossing. A pair of crosses that correspond to a wave interaction (called companions) must be formed by the same two waves. Once we have identified an intersection we sort through the set of crosses on the same two curves and take as the potential companion the cross that is closest. In practice this has proven to be sufficient to determine the companion. In cases of ambiguities the time step will be repeated with a smaller Δt so that pairs of separate interactions do not become confused. It is also required that the distance between the companions be on the order of spatial grid spacing. If the companions are too far apart the time step is repeated with a smaller Δt.

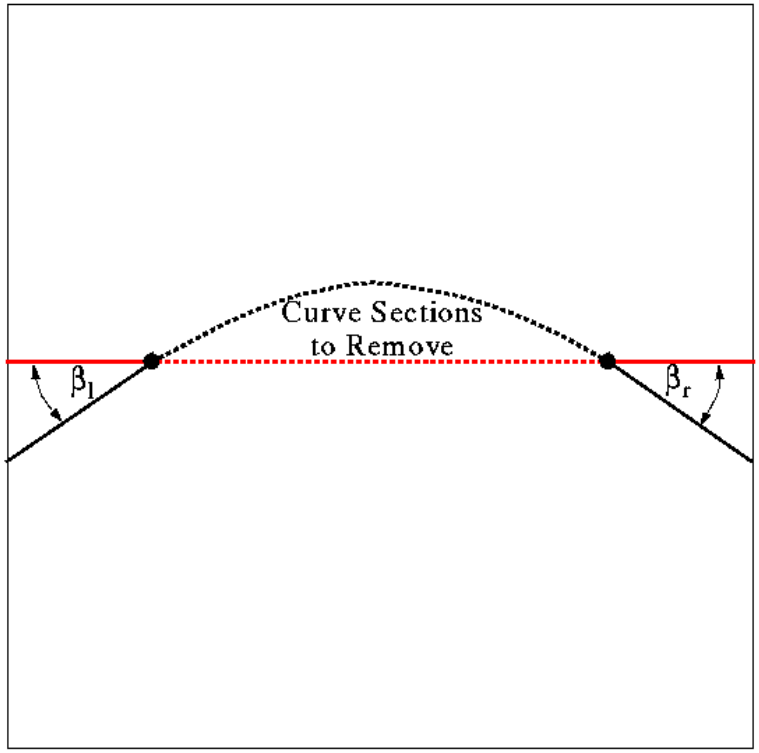


Scalar-Vector Wave Collisions (3)

The next set in the resolution of the scalar-vector tangle is to estimate the velocities of the new nodes, transform the upstream states to the steady frames of the nodes, and perform the shock polar analysis to determined the scattered wave configuration. Since these are new nodes we can not compute their velocities as in the node propagate algorithm. Instead we use the relation between the incoming flow states and the incident angle:

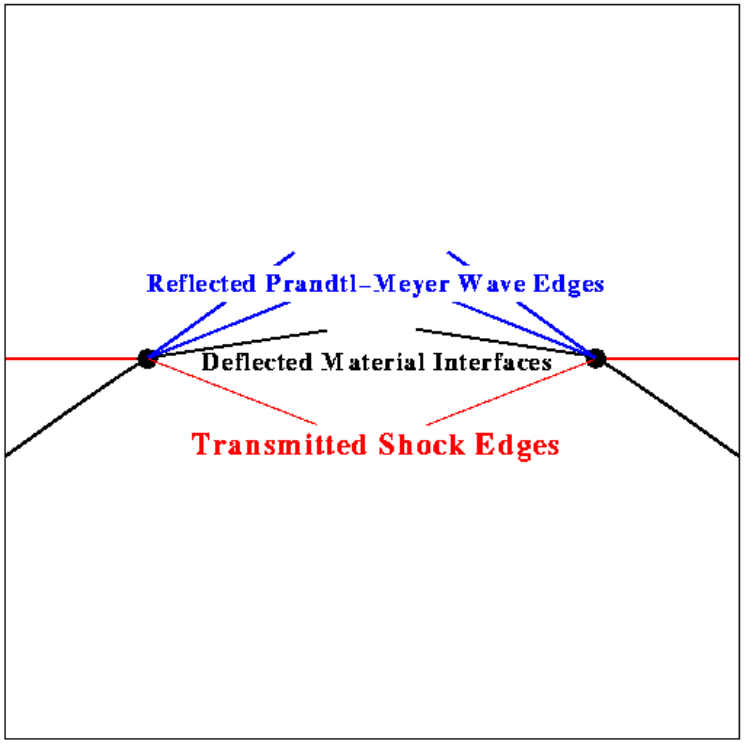
$$\rho_0 q_0 \sin\beta = m$$

where q_0 is the incoming flow speed in the frame of the node, m is the mass flux across the incident shock, and β is the incident angle. This data is all available from the upstream geometry and flow states. The direction of the node velocity is taken to be tangent to the contact.



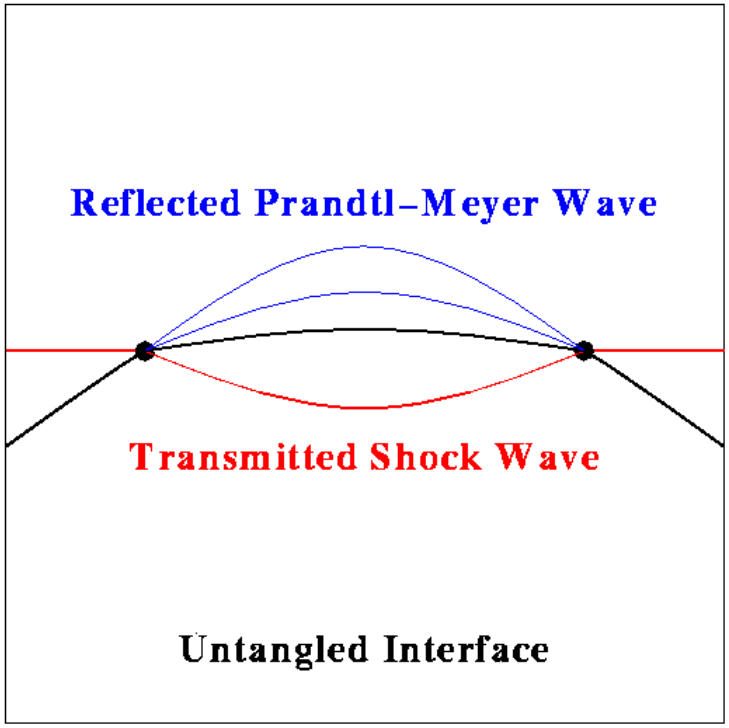
Scalar-Vector Wave Collisions (4)

The shock polar analysis provides the angles that the scattered waves make with respect to the incoming waves together with the states at the node. We install curve fragments corresponding to each of the scattered waves at each of two nodes. Note that unless the interaction is totally symmetric with respect to the two nodes the node velocities and incoming states at each node will be different and thus the scattered wave pattern will not be identical at the two nodes. However we assume that the time step is sufficiently small so that the scattered patterns are similar. In particular both waves must produce the same reflected wave type (either both shocks or both Prandtl-Meyer waves).



Scalar-Vector Wave Collisions (5)

The final step in the algorithm is to connect the new curve segments and interpolate intermediate states in the middle of the sections. There are two cases depending on whether the new segments intersect or not. In the former case a point is inserted at the intersection point and the curves are clipped to this point. In the latter case the two endpoints are joined. The states at the intermediate points are interpolated from the states at the nodes. The resulting configuration is initially rather coarse, but the scattering is done on a length scale of about one spatial mesh block. After a few time steps the configuration is smoothed out by the point and node propagation algorithm and yields a resolved wave pattern that is generally quite smooth and well behaved.



Exercises

Show that to first order the turning angle θ across a shock is related to the incident angle β by the formula:

$$\theta = \rho_0 \frac{\Delta u}{m} \beta + O\left(\beta^2\right)$$

where Δu is the change in the normal component of velocity, ρ_0 is the density ahead of the shock and m is the mass flux across the shock.

Further Reading

- J. Glimm and D. H. Sharp, *A S-matrix Theory for Classical Nonlinear Physics*, Found. Phys. **16**, 126--141, 1986.
- R. Courant and K.O. Friedrichs, *Supersonic Flow and Shock Waves* , Springer-Verlag 1948.
- J. W. Grove, *The Interaction of Shock Waves with Fluid Interfaces*, Adv. Appl. Math. **10**, pp. 201-227, 1989.
- J. Grove and R. Menikoff, *The Anomalous Reflection of a Shock Wave at a Material Interface*, J. Fluid Mech. **219**, pp. 313-336 1990.
- J. Glimm, C. Klingenberg, O. McBryan, B. Plohr, D. Sharp, and S. Yaniv, *Front Tracking and Two-Dimensional Riemann Problems*, Adv. Appl. Math. **6**, pp. 259-290, 1985.

Implementing Wave Interactions in Front Tracking II

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Scalar Wave Interactions (1)

Scalar Wave Interactions (2)

Examples of Scalar Tangles

The Scalar Untangle Algorithm (1)

The Scalar Untangle Algorithm (2)

Node List Structure

The Scalar Untangle Algorithm, Pass 1

The Scalar Untangle Algorithm, Pass 2

The Scalar Untangle Algorithm, Cleanup

The Scalar Untangle Algorithm, Remarks

The Scalar Untangle Algorithm, Three dimensional Flows

The Grid Reconstruction Scalar Untangle Algorithm (1)

The Grid Reconstruction Scalar Untangle Algorithm (2)

The Grid Reconstruction Scalar Untangle

Algorithm (3)

The Grid Reconstruction Scalar Untangle Algorithm (4)

The Grid Reconstruction Scalar Untangle Algorithm (5)

The Grid Reconstruction Scalar Untangle Algorithm (6)

The Grid Reconstruction Scalar Untangle Algorithm (7)

Surface Element Types for Three Dimensional Interface Rebuild

Surface Element Types for Three Dimensional Interface Rebuild

Exercises

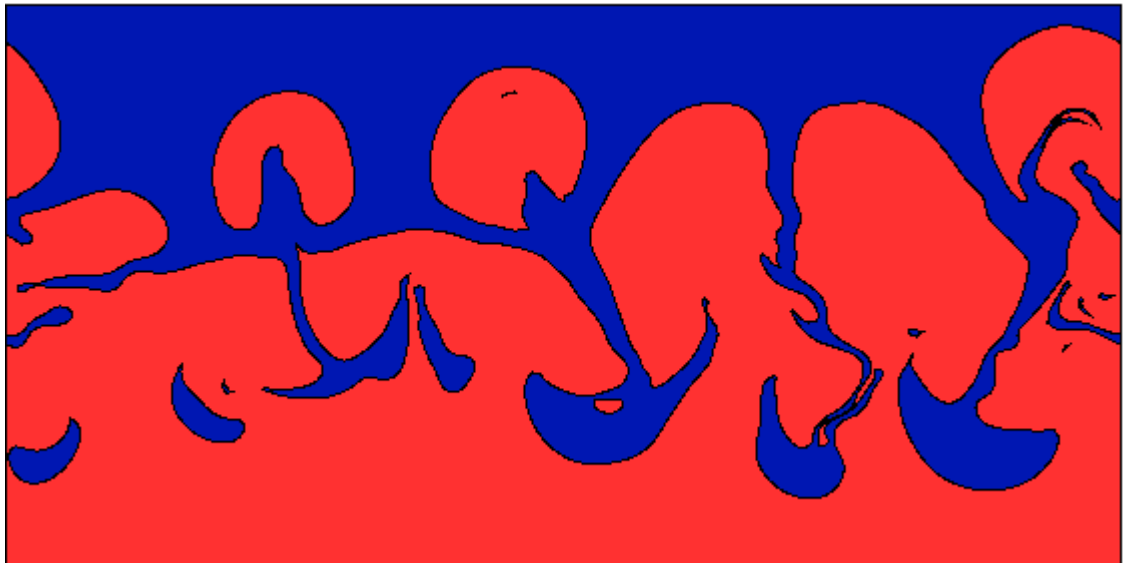
Further Reading

Scalar Wave Interactions (1)

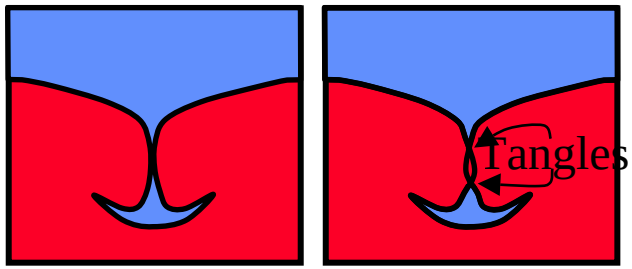
The scalar-vector wave interaction described in the previous lesson is a good example of how the front tracking method allows the inclusion of analytic theory in the computational method. However by far the most important class of waves that require tracking are contacts and material interfaces, and it is their interactions that must be dealt with if we are going to provide a viable computational tool for numerical simulations. In a strict mathematical sense one might say that contacts never truly interact in an inviscid flow. As long as the flow map remains continuous separate regions of the flow never break and remain connected. Separate contacts can come arbitrarily close and wind around each other is very complicated ways. Numerically however we have a finite length scale and at some point a topological change in the interface will occur. More importantly, all physical flows are viscous at some level and fluids will eventually mix at a molecular level once the interfaces become sufficiently entrained. From the modeling point of view we seek an algorithm that preserves the inviscid behavior of the flow while modeling the break off of droplets once a fluid filament becomes sufficiently thin. The finite resolution of the computation provides a natural means of implementing this feature through the resolution of interface tangles. <u>Simply remove the sections of interfaces that have crossed during the time step.</u>

Scalar Wave Interactions (2)

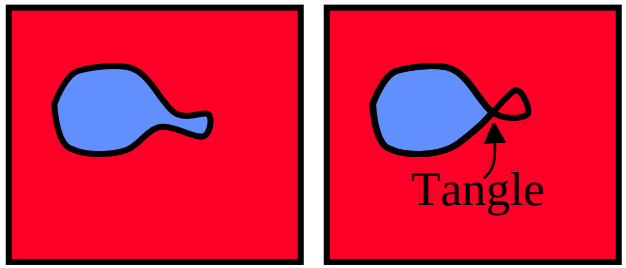
The figure below illustrates the type of interaction produced by scalar waves. This simulation shows the later time behavior of a Rayleigh-Taylor unstable interface. Initially the interface consisted of a randomly perturbed contact separating two connected regions. At later time we see the formation of islands of one material embedded in the other. This was the result of the pinch off of thin filaments that originally connected the islands to the main body of that fluid. Physically, forces due to surface tension and viscosity would produce such a behavior. Here we model the process by a simple topological cutoff.



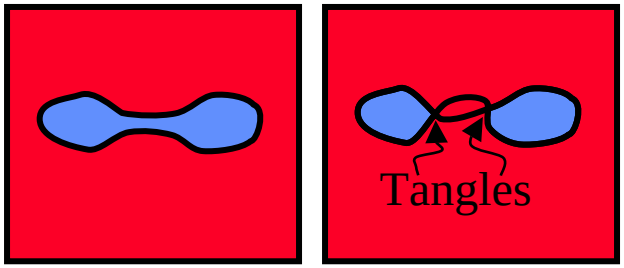
Examples of Scalar Tangles



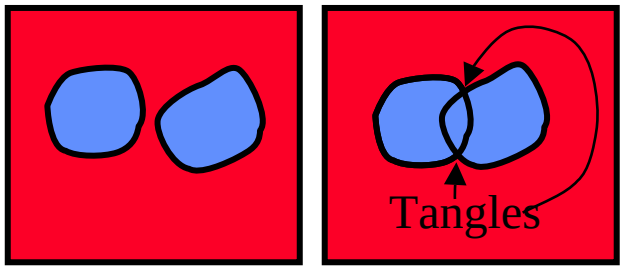
Pinch Off



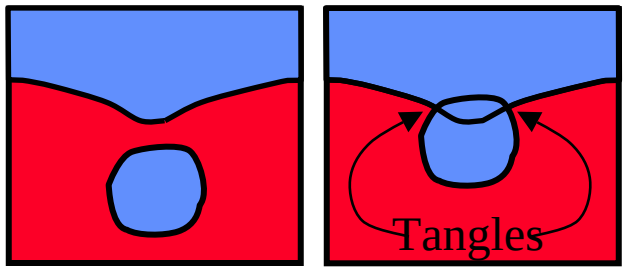
Twist Off



Split



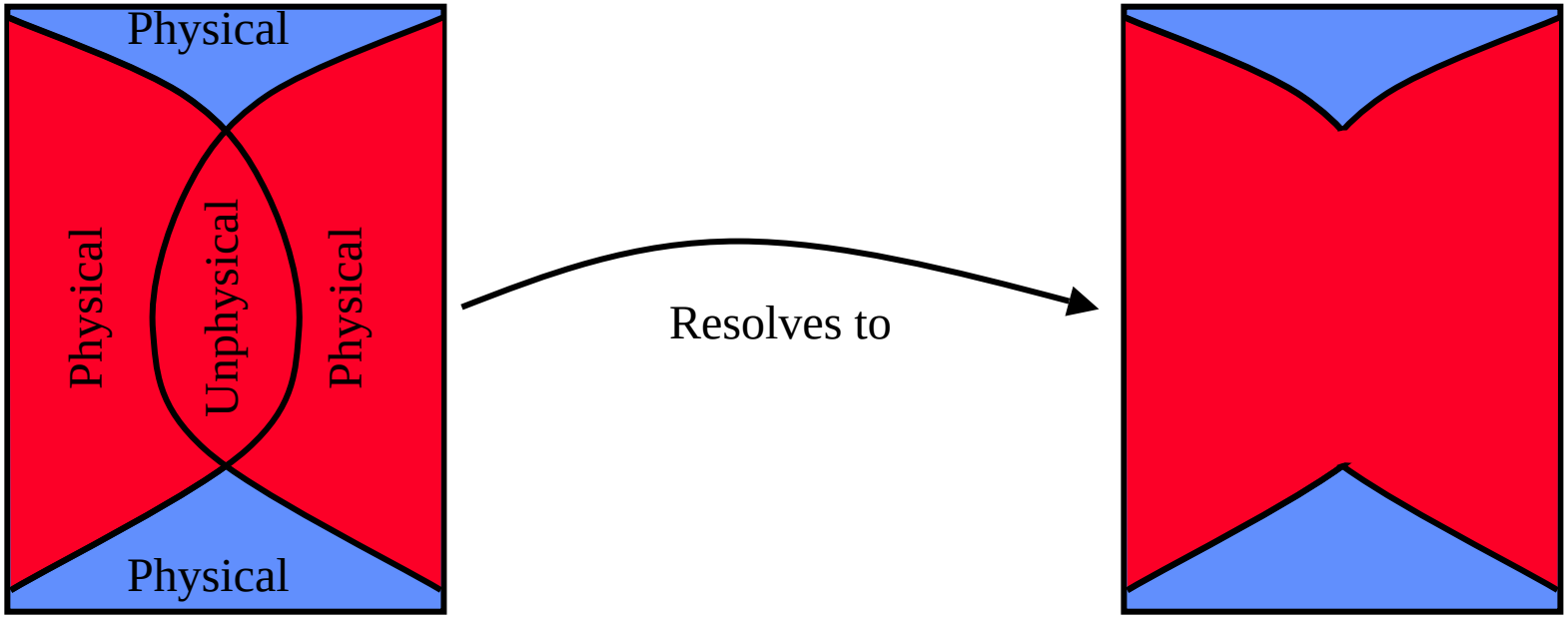
Blob Merge



Breakthrough

The Scalar Untangle Algorithm (1)

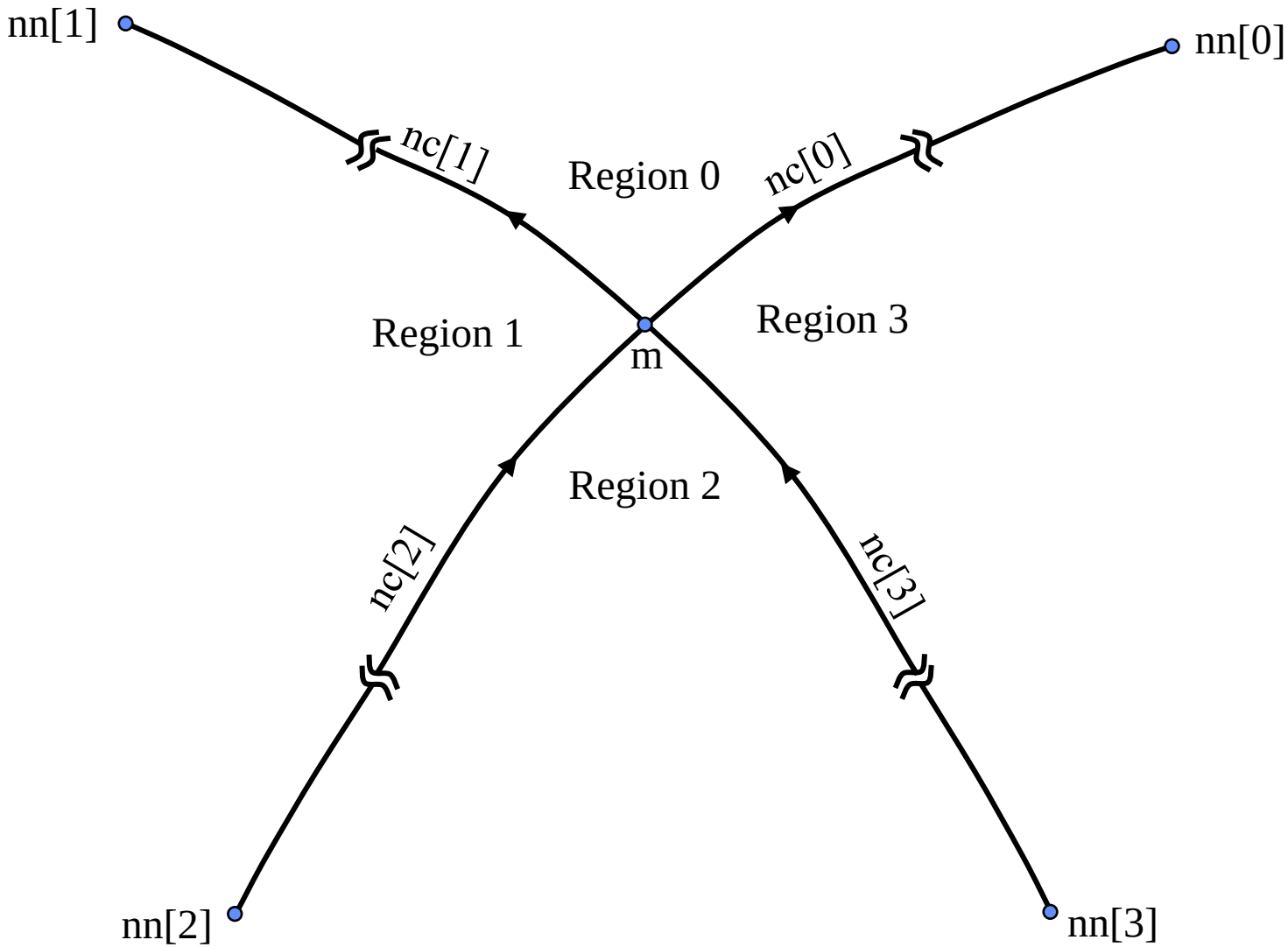
The algorithm for the resolution of scalar wave tangles is based on the identification of regions to be removed by deleting the curves that bound them. We define a local region around a cross point to be physical if one or both of the two bounding curves is to be retained, unphysical if both bounding curves are to be removed. For example, in the pinch off case the local configuration looks like:



The Scalar Untangle Algorithm (2)

- Detect and resolve all vector-vector and vector-scalar tangles.
- Eliminate degenerate cases by searching for extremely close crosses (within say 0.01 Δx of each other) and then deleting a point on one of the crossing bonds to remove the tangle. Recompute the cross list if necessary.
- Split the crossing curves at the cross points. This replaces the cross points with nodes connecting four curve sections. It is at this point that the states are determined at the new points created by the crossings. These states are obtained by linear interpolation along the crossing bonds.
- Remove any redundant closed nodes from the split curves. A closed node has no physical meaning other than as a point. It is simply an entry point into the linked list representation of a closed curve.
- Create a node list containing the following data:
  - The node m formed by the crossing point.
  - Curves nc[i], i=0,…3 for the four curves connecting the new node m.
  - Nodes nn[i], i=0,…3 that are the opposite node on the curve nc[i] from the node m. (Note: it is quite possible nc[i] = m for some i.)

Node List Structure



The Scalar Untangle Algorithm, Pass 1

- Perform a first pass loop over all of the new nodes in the node list. During this pass we identify physical regions by the criteria (all arithmetic is mod 4):
  - If nn[i] $\neq$ nn[i+1] then region i is physical.
  - If a region forms a closed loop with node m, then label the two neighboring regions as physical, i.e. if nn[i] = m and nc[i] = nc[i+1], then regions i–1 and i+1 are physical.
- Loop over node list and delete curves by the criteria:
  - If three out of the four regions about a node have been identified as physical, delete the curves bounding the single unphysical region.
  - If less than four curves remain at a node, then remove that node from the processing list (it was resolved based on the deletion of curves from a neighbor).
  - Continuing looping over the node list until no nodes fitting one of the two above conditions remain.

The Scalar Untangle Algorithm, Pass 2

The processing of the remaining tangles will require more topological and geometric information about the tangles. Most often these tangles are associated with closed curve tangles. Loop over the remaining tangles in the following way until all are resolved.

- Remove any nodes that contain only two curves.
- Compute the area of each of the remaining unidentified regions.
- For a cross point with exactly two regions identified as physical:
  - Delete the smaller area unidentified region if its area is very small with respect to the other unidentified region.
  - If one of the unidentified regions has the property that both bounding curves terminate at the same node (i.e. nn[i] = nn[i+1] = n) and n is not in the processing list, and the components are consistent at n, then label this region as physical and delete the curves bounding the other unphysical region at m.
  - If the component numbers in an unidentified loop are inconsistent then delete that loop.
  - Determine the component number exterior to the tangle. Delete the unidentified sector with this component number. This occurs in the split case.
  - Remove the loop with minimum area.
- If tangles remain repeat.

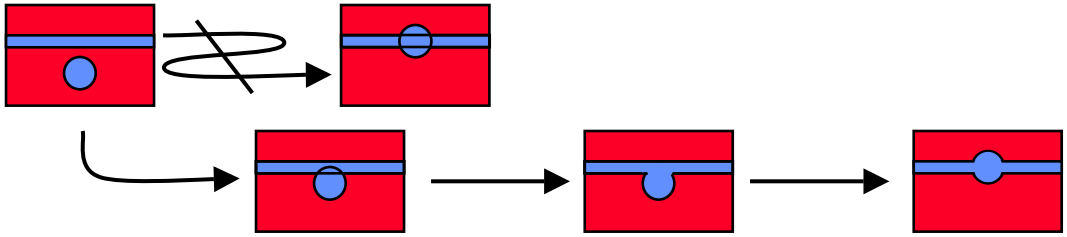
The Scalar Untangle Algorithm, Cleanup

The final step in the scalar untangle algorithm is to loop through the new nodes at the now untangled positions. Each such node will have have a single incoming and a single outgoing curve. Recursively we process this node list as follows.

- If one of the two curves at the node has a node other than the new node, join the two curves at the new node, replacing the node by a simple point.
- If the incoming and outgoing curves at the new node are the same curve, i.e. there is a closed loop at the node, then declared that node to be a closed node and remove it from the processing list
- Continue from the beginning of the list until all nodes are either eliminated by joining the curves or are declared closed.

The Scalar Untangle Algorithm, Remarks

- The scalar untangle algorithm proceeds recursively through the node list. Once a region is identified as unphysical and to be deleted, this deletes sections at other nodes, thus resolving them as well.
- In the event this algorithm is unable to resolve all tangles, the time step is repeated with a smaller Δt. Most often problems that occur are due to multiple wave crossing (for example one contact crosses through two others). Reducing the time step resolves this tertiary interaction into separate binary interactions spread over a couple of time steps.



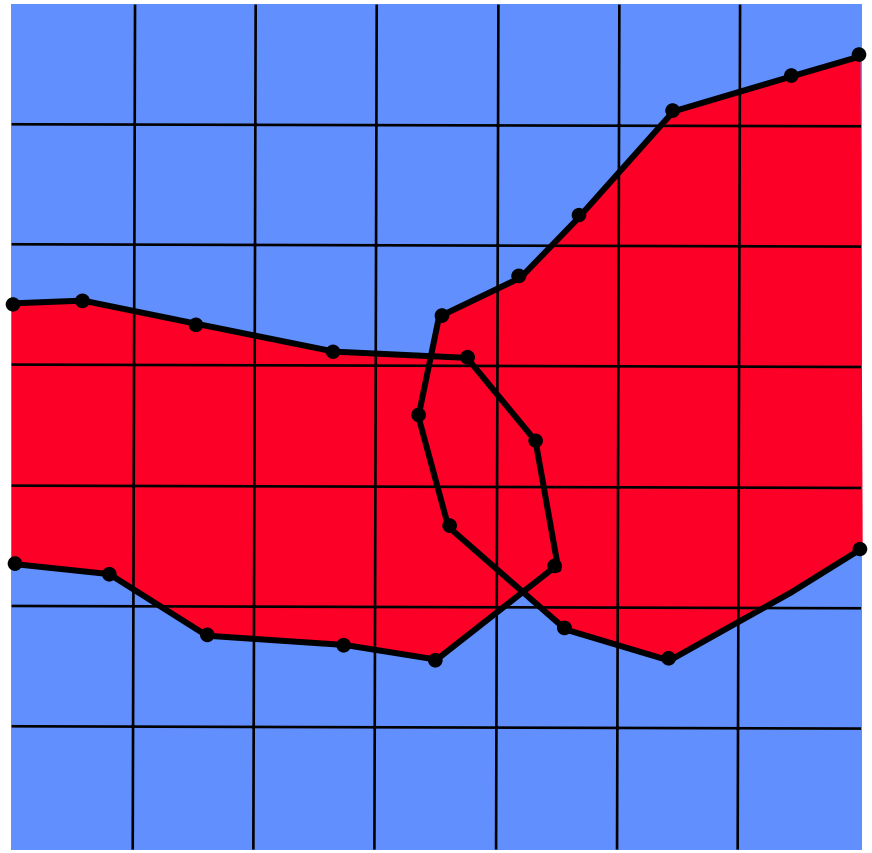
- The current algorithm does not try to conserve the mass of the deleted sections. This would clearly be a desirable upgrade of the algorithm. In the current algorithm time step control is used to ensure that the loops removed have area that are small relative to a grid block size. This helps keep the mass loss within the order of accuracy of the method, in this case first order at the front.
- In practice this algorithm is remarkably successful in two space dimensions. It has allowed us to compute flow patterns that become extremely broken up, and mass loss has been kept to a reasonable level.

The Scalar Untangle Algorithm, Three dimensional Flows

The generalization of the scalar untangle algorithm to three space dimensional flow proceeds along similar lines as the two dimensional scalar untangle. Intersecting surfaces are split along crossing curves, and the local topology is examined to select the surface sections to be removed. This method is used in conjunction with an alternative algorithm developed and implemented by X. L. Li that, at the expense of some loss of information on the surface, gives a robust method for both redistributing the interface and untangling the interface simultaneously. This new algorithm is based on ideas taken from the level set method of Sethian and Osher and is currently limited to flows with only two materials, although it should be possible to generalize it to more complicated regimes. The basic idea is to replace the surface information by a labeling of cell centers by their component numbers, and the interface grid crossings. This is accomplished as illustrated in the next series of slides. For simplicity the example is shown in two space dimensions. The procedure for three space dimensions is identical except for the last step where the grid crossings are rejoined into surface segments. We will discuss that procedure presently.

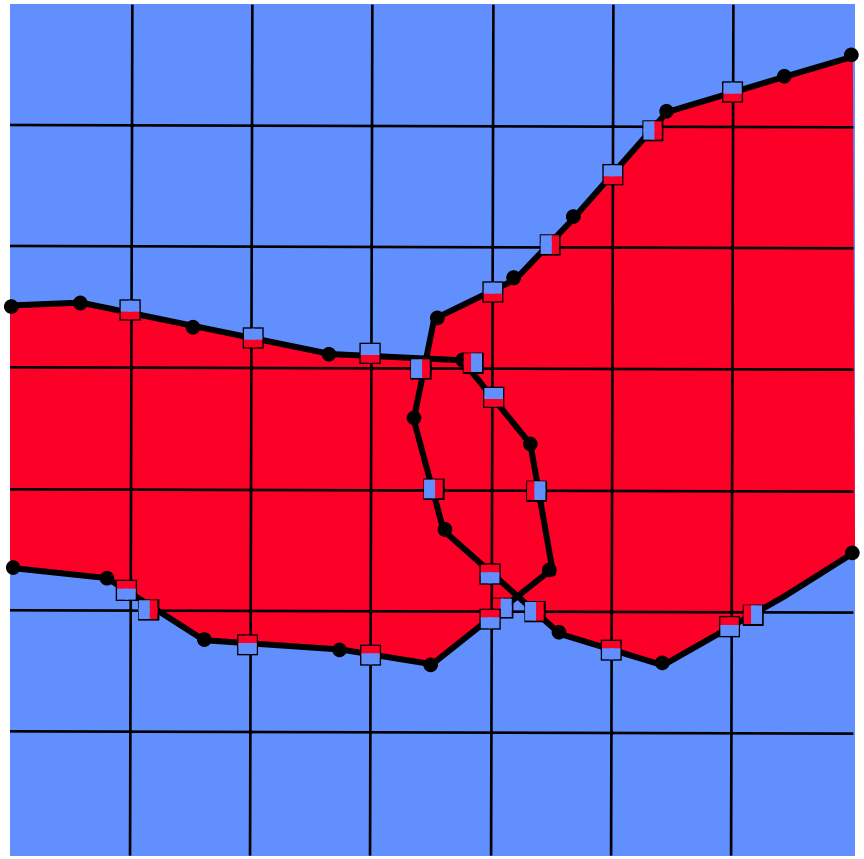
The Grid Reconstruction Scalar Untangle Algorithm (1)

This figure shows a pair of tangled interfaces. The black circles show the interface points, and the grid is the dual lattice obtained by joining the computational grid cell centers.



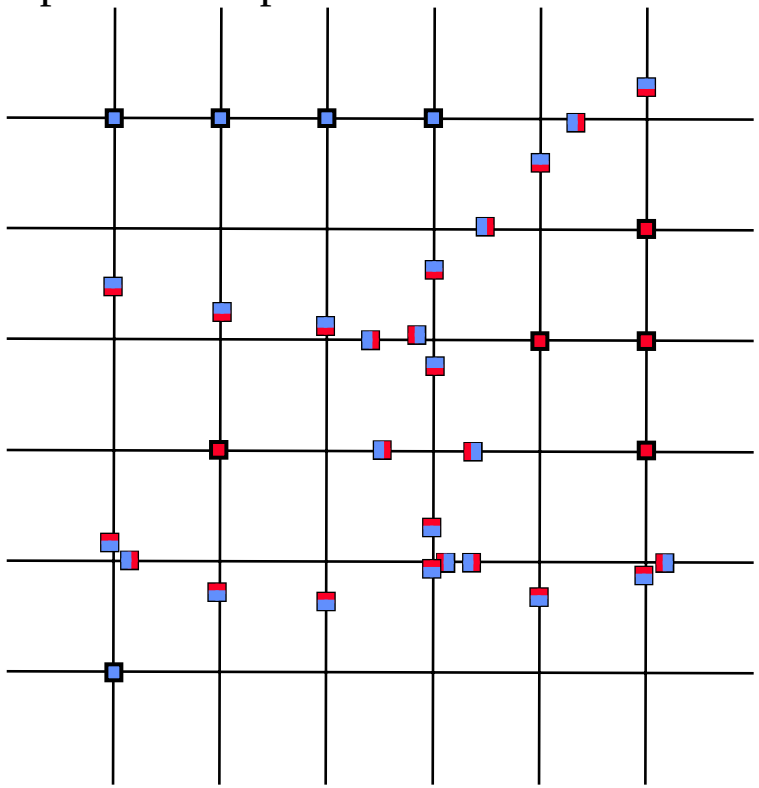
The Grid Reconstruction
Scalar Untangle Algorithm (2)

The first step in the algorithm is to compute the crossings of the interfaces with the dual lattice cell edges. We mark each crossing with the change of component direction.



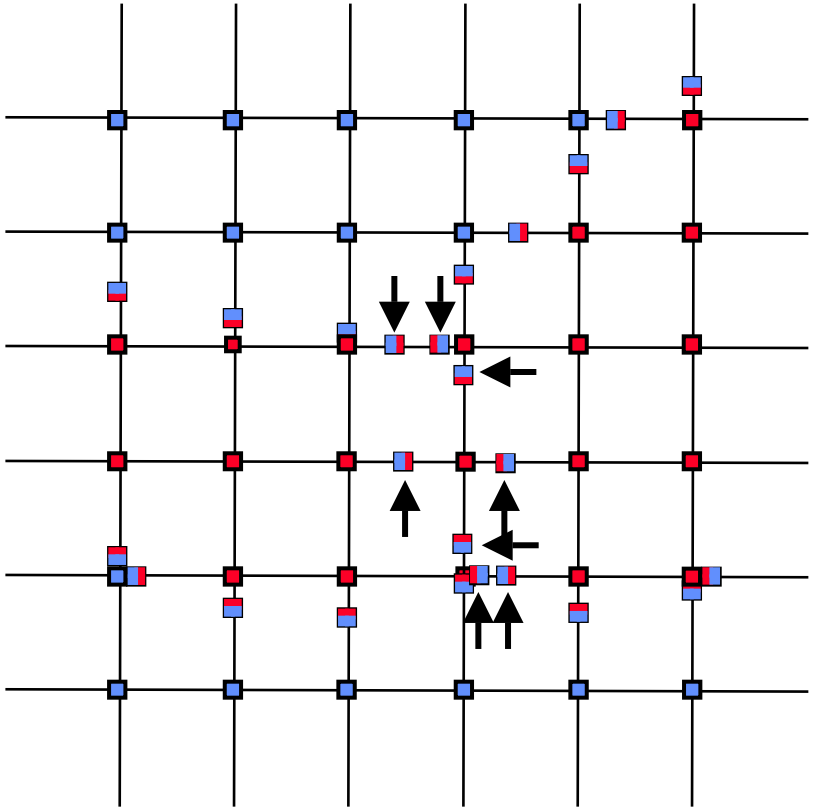
The Grid Reconstruction
Scalar Untangle Algorithm (3)

We now loop through the lattice points identifying places that have not been crossed by an interface during the time step. At such points we have reliable component information from the last time step. Marking the components at such points we then remove all interface elements leaving only the lattices points and the grid crossings.



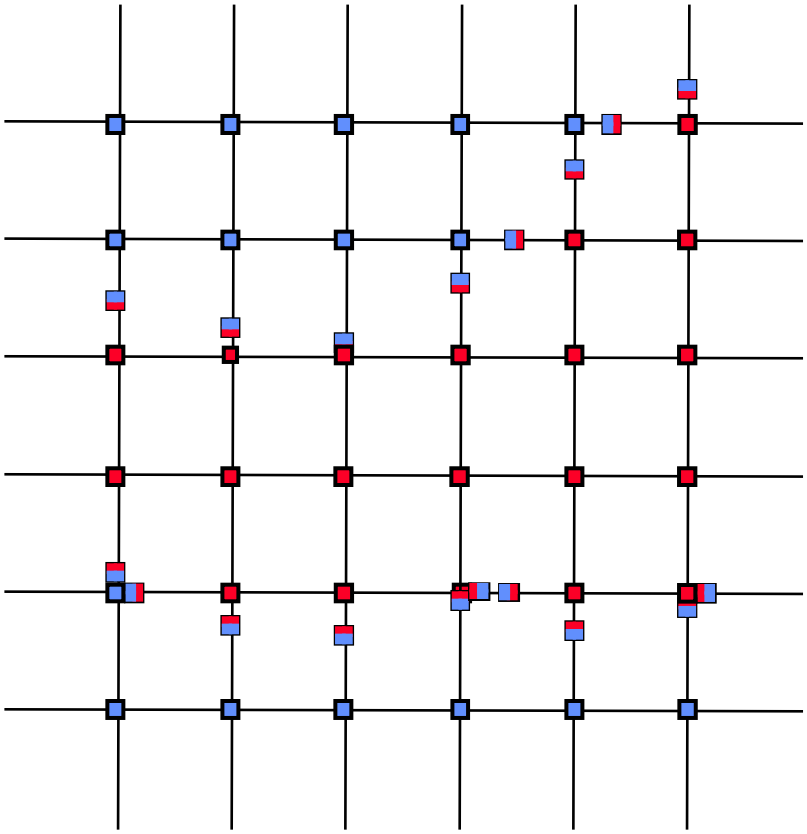
The Grid Reconstruction Scalar Untangle Algorithm (4)

Using a depth first transversal from each lattice point where the component is known, we identify the components of the remaining points on the lattice. Then moving from the lattice points to the grid crossings we identify those crossings that are inconsistent, i.e. the component changes in the reverse direction from adjoining lattice points. We also schedule for removal two consistent crosses that do not produce a net component change across the cell edge.



The Grid Reconstruction Scalar Untangle Algorithm (5)

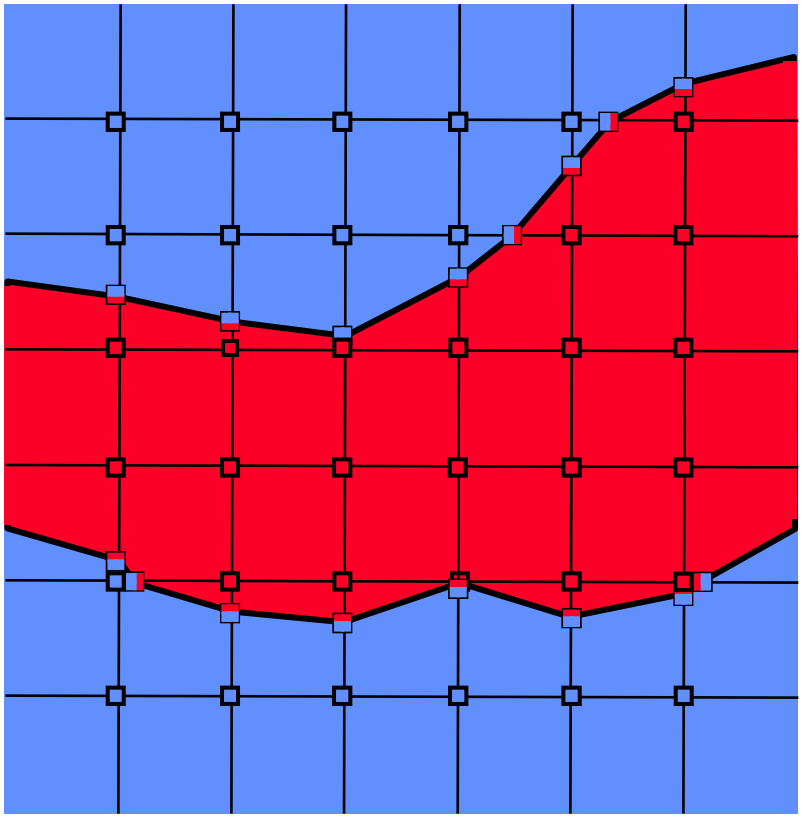
After removal of the inconsistent or redundant grid crossing we have a skeleton upon which we can rebuild the interface.



The Grid Reconstruction
Scalar Untangle Algorithm (6)

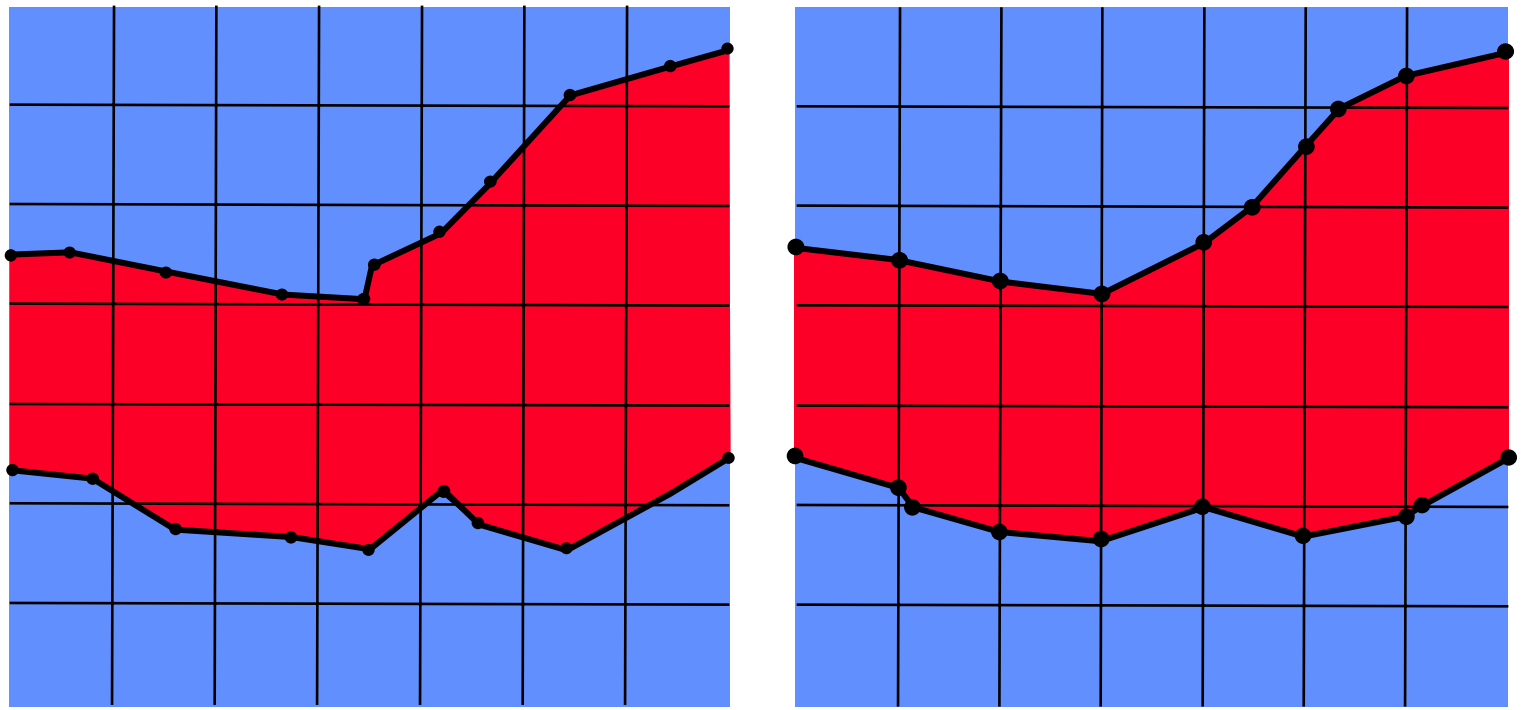
The Scalar Untangle Algorithm, Three dimensional Flows

The interface is rebuilt by adding surface elements in each dual lattice cell, and then connecting these elements into curves or surfaces. In two space dimensions This just means joining a pair of crossing points by a bond. The algorithm in three space dimensions requires identifying an appropriate surface fragment to join the crossing points. Fortunately it turns out that only thirteen basic cases need to be considered.



The Grid Reconstruction Scalar Untangle Algorithm (7)

The untangled interface formed by the grid based untangle is shown on the right. The left figure shows the result of interface based untangle. Both methods give similar results. The grid based method has considerably smoothed the interface and has produced a somewhat poorer point distribution in places. However the algorithm is very robust and does not require a detailed examination of the interface topology.



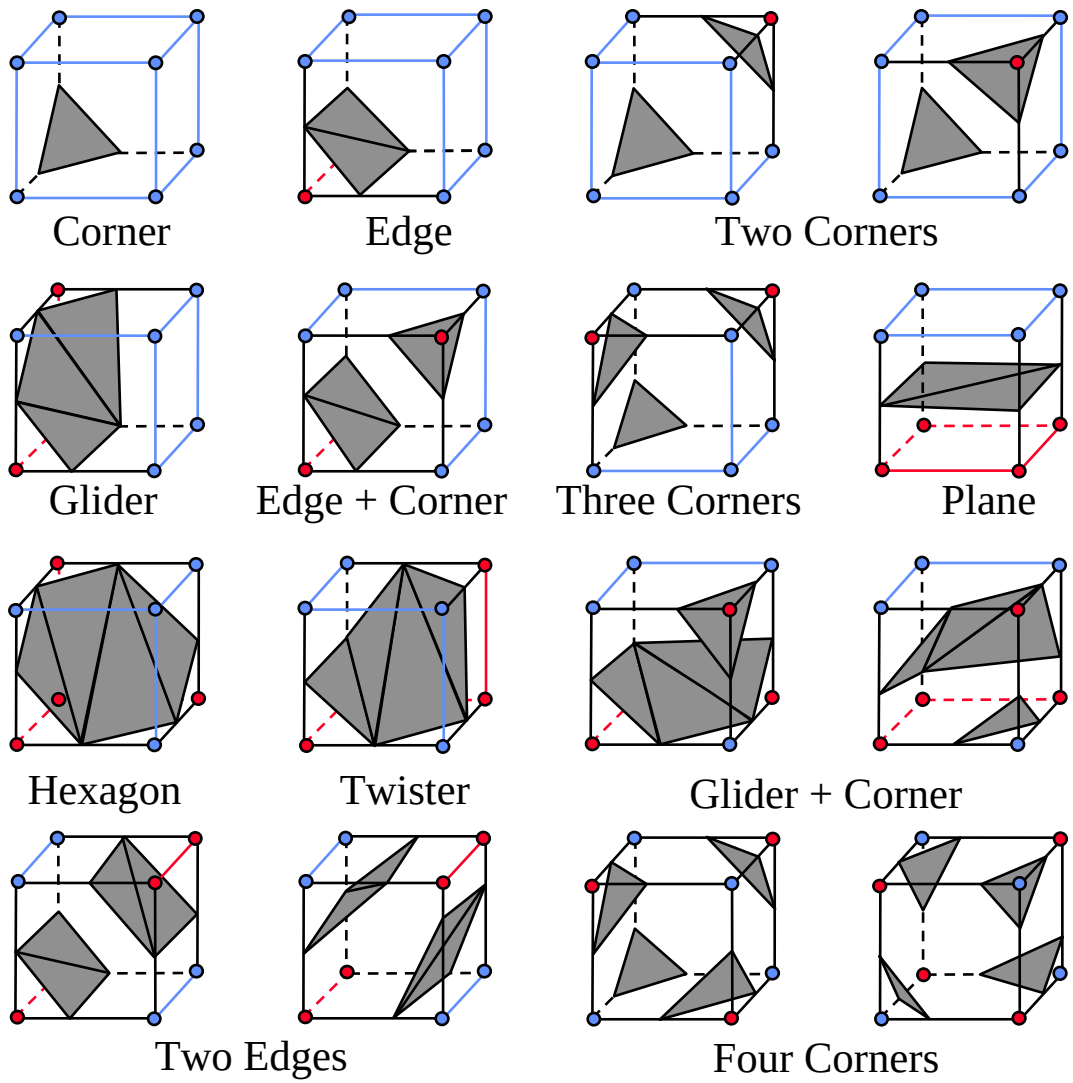
Surface Element Types for Three Dimensional Interface Rebuild

The surface rebuild algorithm is based on the following assumptions.

- At most two components intersect a cell.
- Each cell edge has at most one interface crossing.
- The cell corners and edges that lie on the same side of a rebuilt interface segment form a connected set.

Under these assumptions it can be shown that there are a total of thirteen geometrically distinct cases in the sense that under the operations of rotation, reflection, or interchange of component number any of the total of 256 possible colorings of the vertices of a cube can be brought to either the empty case of all vertices the same color, or into one of the thirteen cases shown on the next slide.

Surface Element Types for Three Dimensional Interface Rebuild



Exercises

Show that with respect to the equivalence relation of rotations, reflections, component inversion, and sliding of an interface edge crossing along an edge, the thirteen cases shown on the previous slides form a complete list of interface reconstructs subject to the requirements that:

- At most two components intersect a cell.
- Each cell edge has at most one interface crossing.
- The cell corners and edges that lie on the same side of a rebuilt interface segment form a connected set.

Further Reading

- J. Glimm, J. W. Grove, W. B. Lindquist, O. McBryan, and G. Tryggvason, *The Bifurcation of Tracked Scalar Waves*, SIAM J. Sci. Stat. Comp., **9**, pp. 61-79, 1988.
- J. Glimm, J. W. Grove, X. L. Li, and D. C. Tan, Robust Computational Algorithms for Dynamic Interface Tracking in Three Dimensions, submitted to SIAM J. Sci. Comp. 1998.

Applications and Research

LA-UR 99-3985

8/2/99

Click here to start

Author: John W. Grove

Email: jgrove@lanl.gov

Table of Contents

Rayleigh-Taylor Instability

Three Dimensional Rayleigh-Taylor Unstable Interface

Late Time Three Dimensional Rayleigh-Taylor Unstable Interface

A Topological Bifurcation in Three Dimensional Rayleigh-Taylor Instability

Richtmyer-Meshkov Instability

Data for Richtmyer-Meshkov Computations

Richtmyer-Meshkov Instability

Small Amplitude Convergence of nonlinear simulations to linearized solutions

Perturbation amplitude, and amplitude growth rate, of a shocked air-SF6 interface

Bubble and spike velocities at a shocked air-SF6 interface

Comparison of linear and nonlinear air-SF6 solutions

Single-Mode Richtmyer-Meshkov Instability: Experiment, Simulation and Theory

Description and goals of work

Richtmyer-Meshkov Experimental Configuration

Experiment and Simulation Densities at 2.8ns

Experiment and Simulation Densities at 4.2ns

Simulation Only Densities at 9.6 ns

Pressure and RAGE AMR Grid 9.6 ns

Amplitude/Growth Rate Comparison

Cylindrical Geometry Version of Dimonte Experiment (1)

Cylindrical Geometry Version of Dimonte Experiment (2)

Cylindrical Geometry Version of Dimonte Experiment (3)

Cylindrical Geometry Version of Dimonte Experiment (4)

Cylindrical Geometry Version of Dimonte Experiment (5)

Supersonic Random Velocity Inlet (1)

Supersonic Random Velocity Inlet (2)

Supersonic Random Velocity Inlet (3)

Supersonic Projectile over Water (1)

Supersonic Projectile over Water (20 msec)

Supersonic Projectile over Water (40 msec)

Supersonic Projectile over Water (60 msec)

Supersonic Projectile over Water (80 msec)

Open Problems and Current Research (1)

Open Problems and Current Research (2)

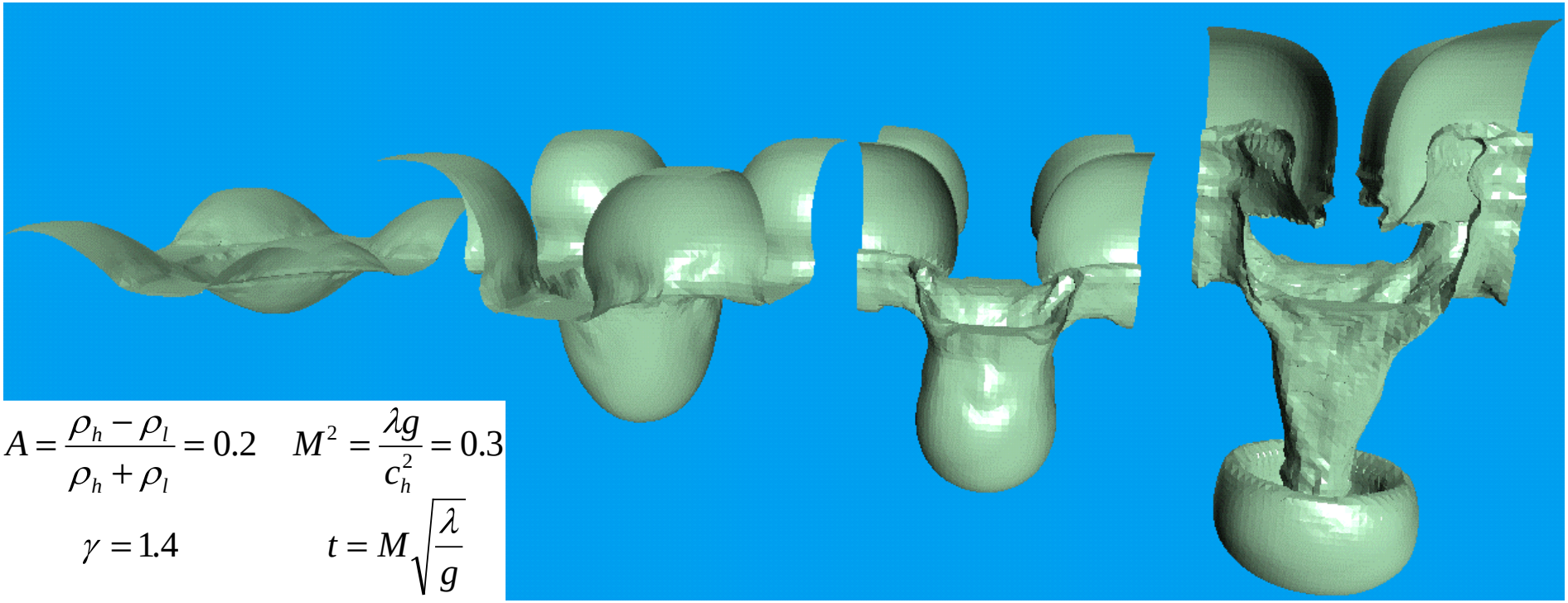
Open Problems and Current Research (3)

Further Reading

Rayleigh-Taylor Instability

Rayleigh-Taylor instability is generated when a heavy fluid sits above a lighter fluid in a gravitational field. The flow behavior is described in terms of bubbles of light fluid rising into the heavier medium and spikes of heavy fluid dropping into the lighter. More generally a material interface is said to be Rayleigh-Taylor unstable whenever the fluid acceleration has an opposite direction to the density gradient. Rayleigh-Taylor instability occurs in a variety of applications ranging from thermal down drafts caused by the suspension of a cold body of air above a warm region, to instability in inertial confinement laser fusion. The classical study of Rayleigh-Taylor instability was for incompressible flows, but much recent work has been devoted to understanding the more complex compressible regime. A major component of this research is devoted to understanding the behavior and growth of the mixing region between the two fluids. Various models have been proposed to predict the rate of growth of the mixing region, and direct numerical simulation plays an important role both in validating the models as well as providing a computational laboratory to gain insight and understanding of the flow.

Three Dimensional Rayleigh-Taylor Unstable Interface

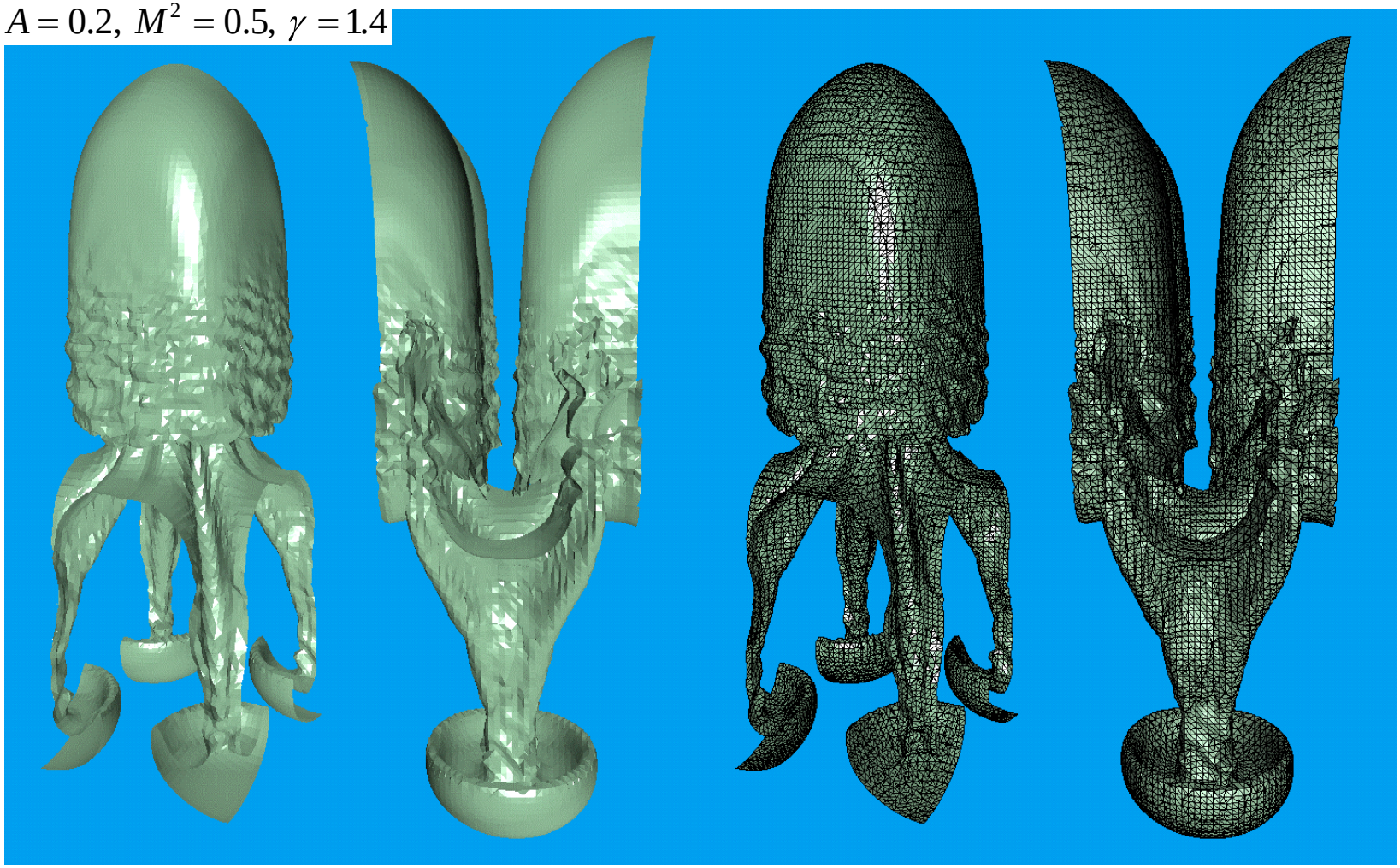


$$A = \frac{\rho_h - \rho_l}{\rho_h + \rho_l} = 0.2 \qquad M^2 = \frac{\lambda g}{c_h^2} = 0.3$$

$$\gamma = 1.4 \qquad t = M\sqrt{\frac{\lambda}{g}}$$

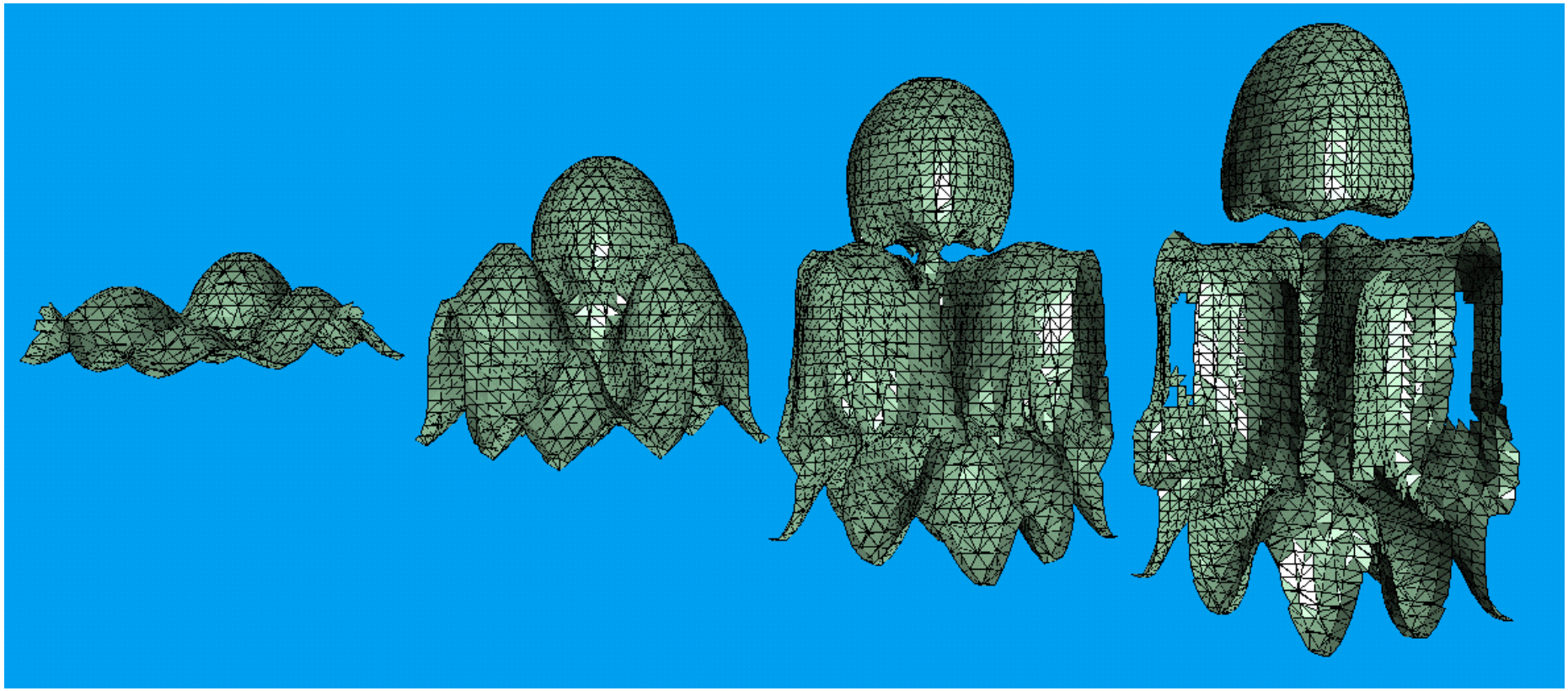
A Kelvin-Hemholtz like shear flow is developed in the wakes of the bubble and spike generating a vortex around the side of both the bubbles and the spikes. The computation is performed in a 40×40×240 regular grid and the grid-based tracking algorithm is used. From left to right $t = 4$, $t = 8$, $t = 12$ and $t = 16$.

Late Time Three Dimensional Rayleigh-Taylor Unstable Interface



The computation is performed in a 40×40×240 regular grid. The figures show views of the bubble and spike. The surface grid is show on the right.

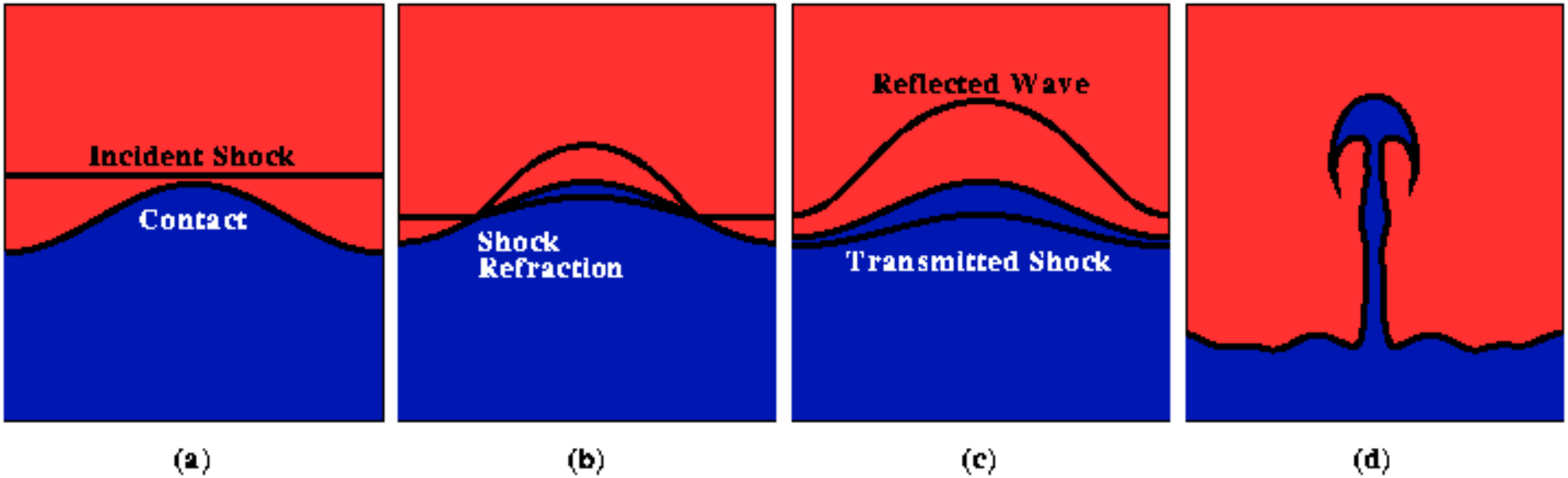
A Topological Bifurcation in Three Dimensional Rayleigh-Taylor Instability



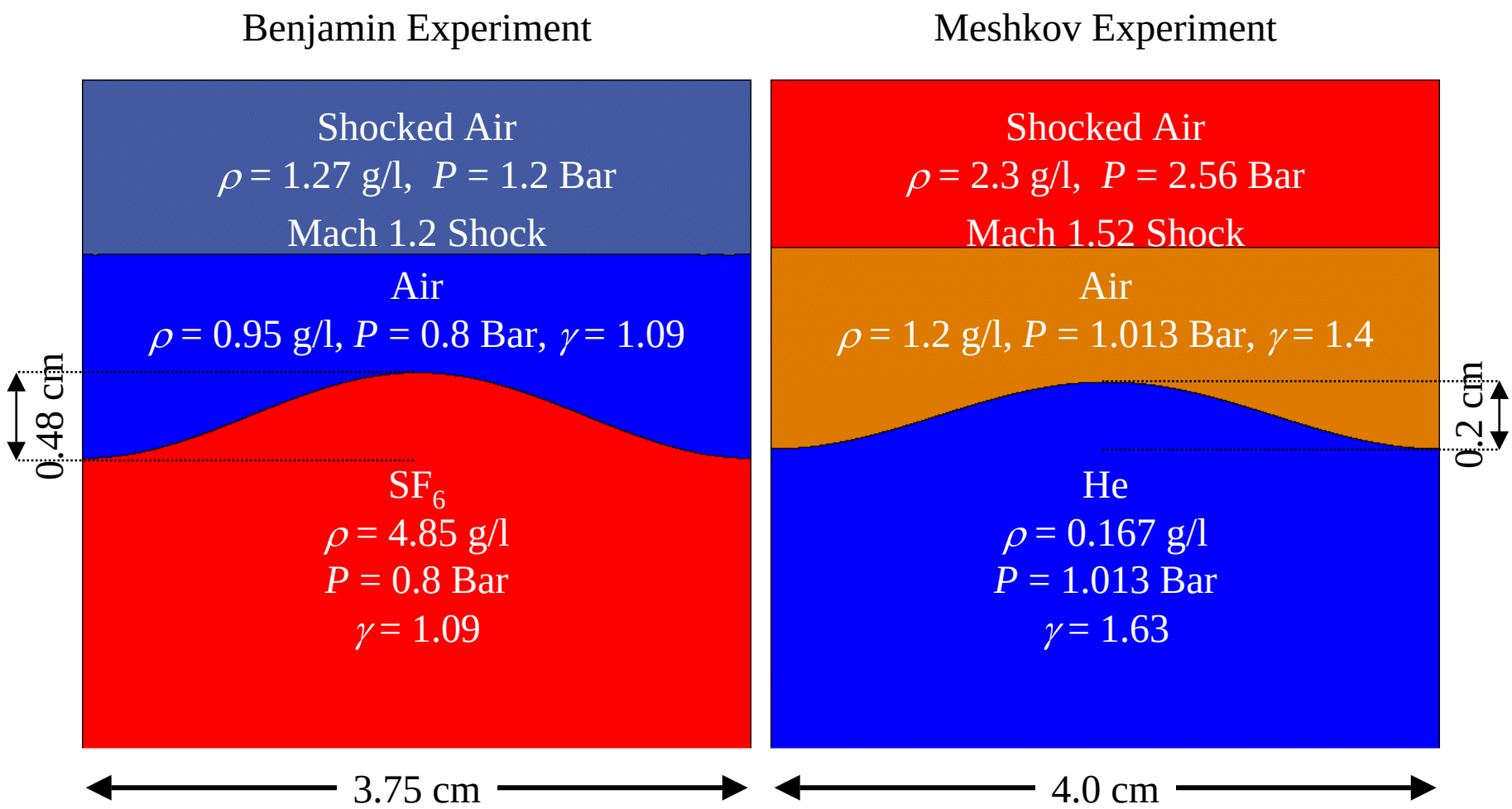
In this simulation there are four bubbles of different radii. The largest bubble moves ahead of the others and eventually pinches off from the main interface.

Richtmyer-Meshkov Instability

Richtmyer-Meshkov instability is a material surface instability generated by the refraction of shock wave through a material interface. It occurs in such applications as inertial confinement fusion and such natural phenomena as a supernova explosion. The schematic diagram below illustrates the process. A shock refracts through a material surface imparting differential vorticity along the interface. This velocity shear in turn drives the penetration of the heavy fluid into the light fluid.

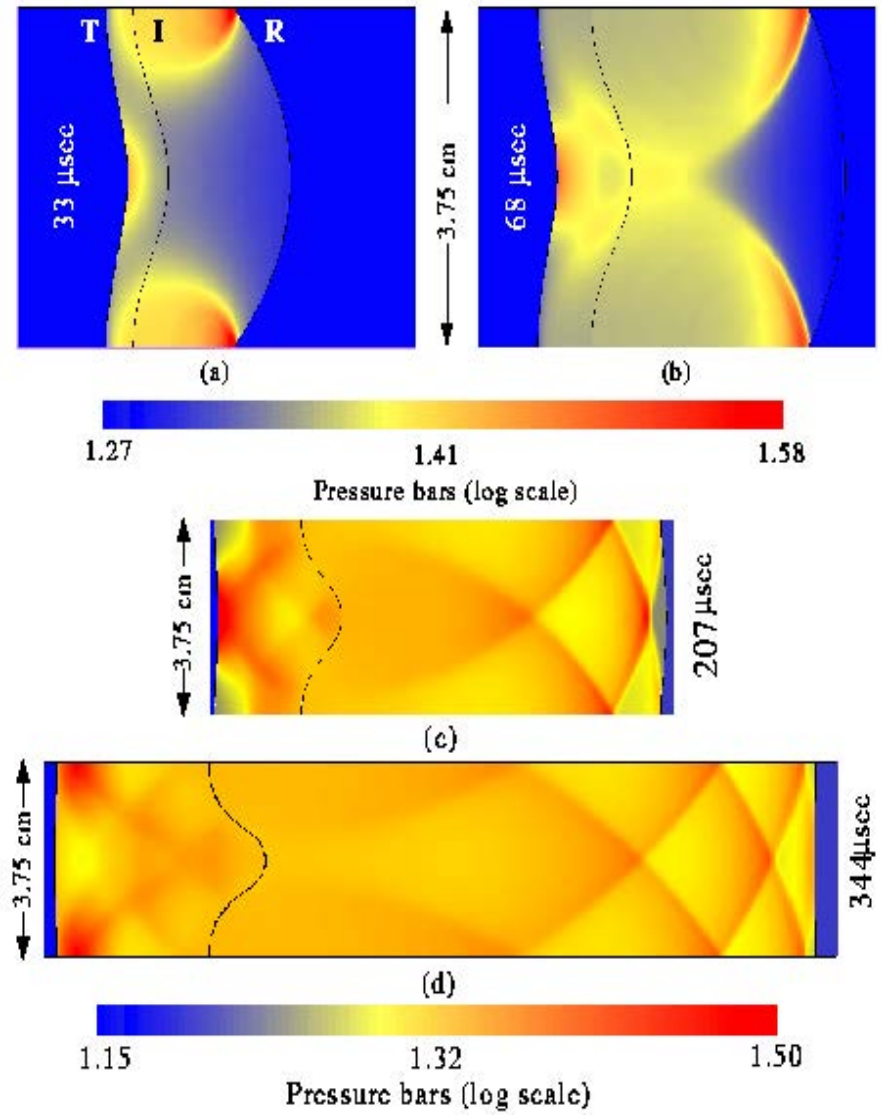


Data for Richtmyer-Meshkov Computations

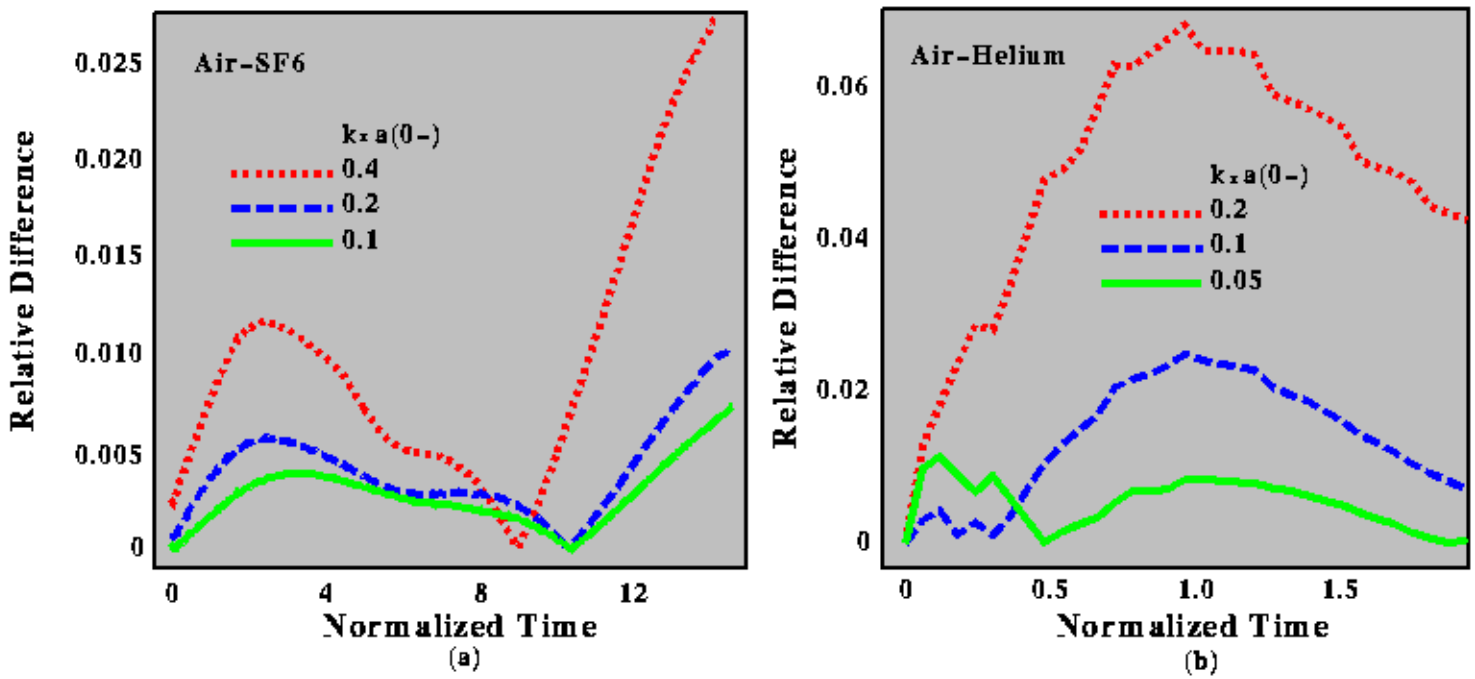


Richtmyer-Meshkov Instability

Pressure plots at a series of times for the nonlinear air-SF_6 solution. A cascade of shock waves generated by the self-interaction of the transmitted and reflected waves propagate back toward the interface and affect the perturbation growth rate at early and intermediate times. For reference we have labeled in the first frame the transmitted shock (T), the interface (I) and the reflected shock (R). Note that the color map has been adjusted so that the full range of color occurs within the region of interest (between the transmitted and reflected shocks). The blue regions at the left and right have pressures of 0.8 bars and 1.1 bars, respectively.

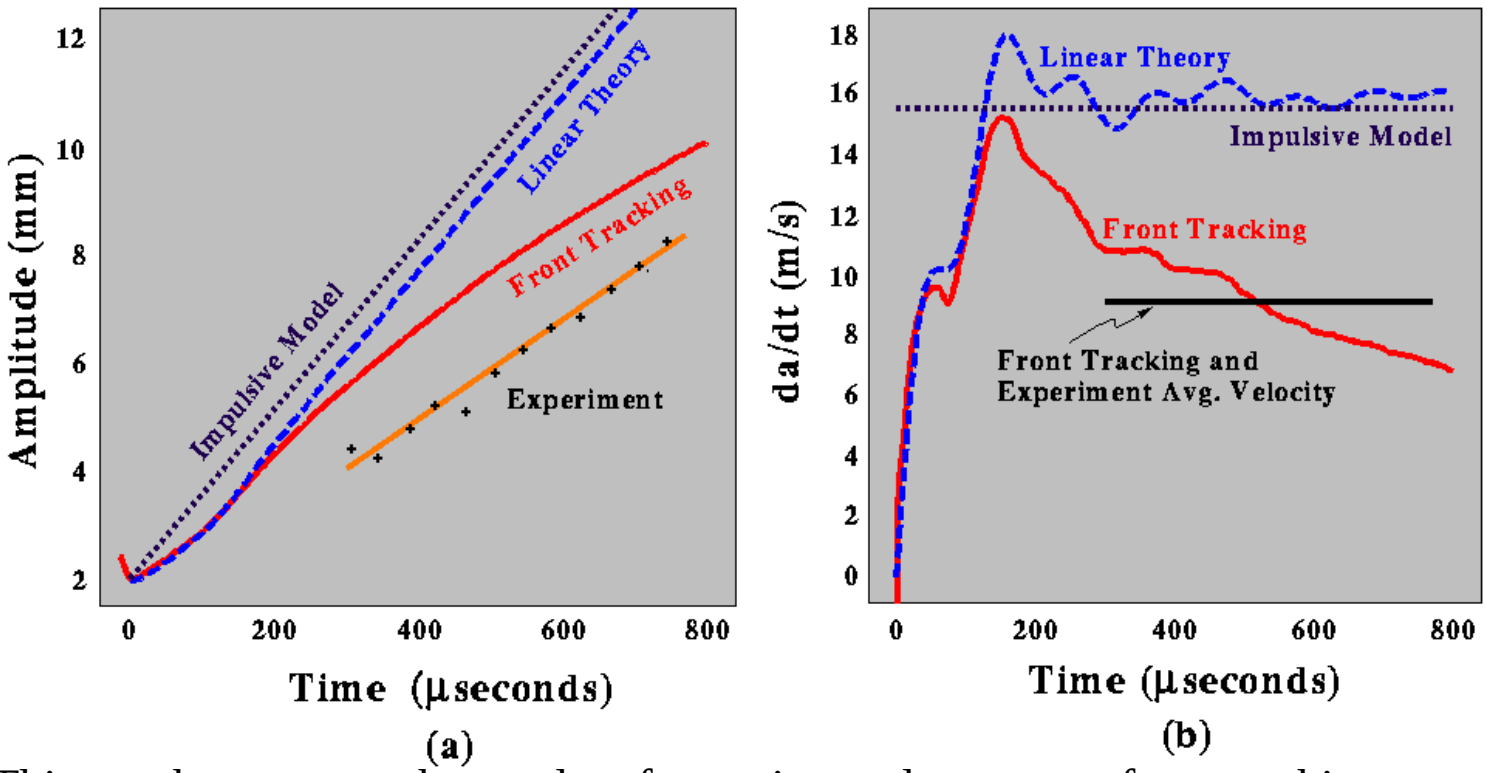


Small Amplitude Convergence of nonlinear simulations to linearized solutions



A series of calculations were carried out for a set of three initial amplitudes of decreasing size. The figures show results for shocked air-SF_6 and air-He interfaces, respectively. Amplitude convergence is measured in terms of the relative difference between the nonlinear and linear solutions, $|(a(t) - a_{lin}(t))|/|(a_{lin}(t)| + |a(0-))|$ where $a(0-)$ is the initial amplitude of the perturbation. The horizontal axis is in dimensionless time units kc_0M_0t, where k is the wave number of the perturbation, c_0 is the sound speed in the air ahead of the incident shock and M_0 is the incident shock Mach number. One dimensionless time unit is equivalent to approximately 14 μs for air-SF_6 case and 12.2 μs for air helium.

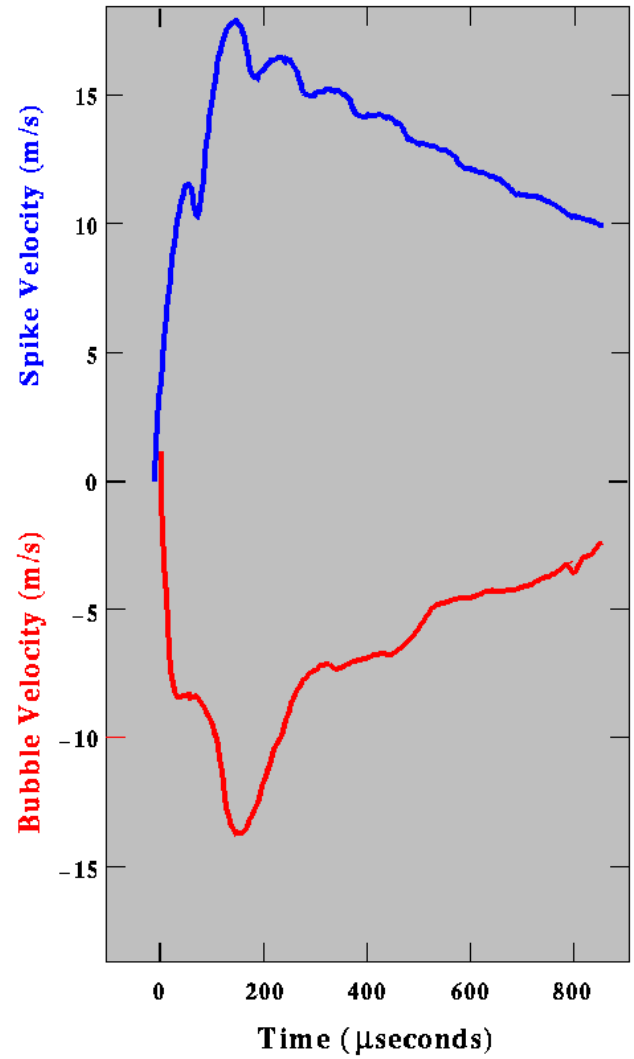
Perturbation amplitude, and amplitude growth rate, of a shocked air-SF_6 interface



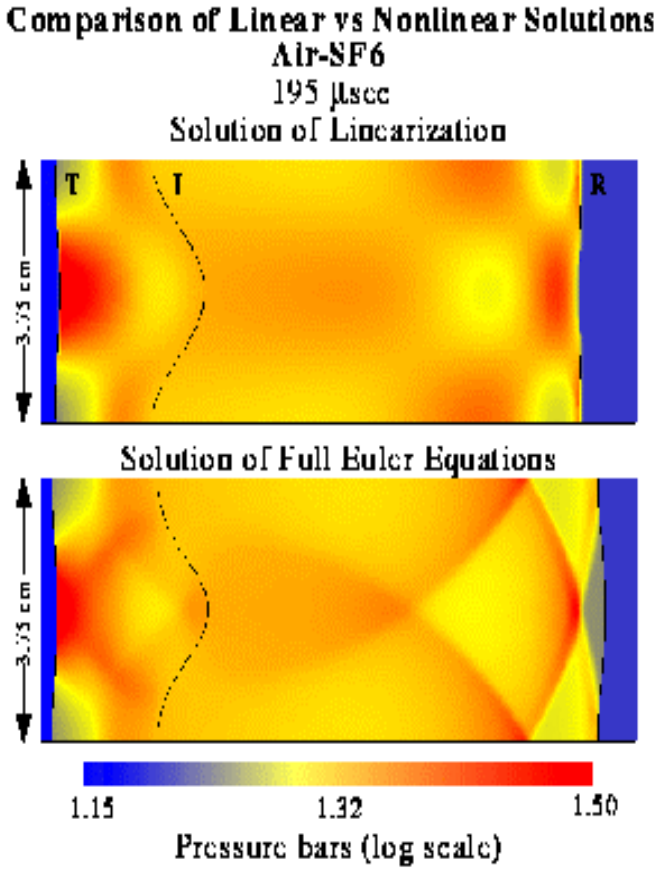
This graph compares the results of experimental averages, front tracking simulation, linear theory and Richtmyer's impulsive model. Also shown are results of a least squares fit to the front tracking amplitude data over the period of experimental observation. Note that the front tracking average growth rate is indistinguishable from the experimental value in figure (b). The plus marks (+) show the results of one particular experiment.

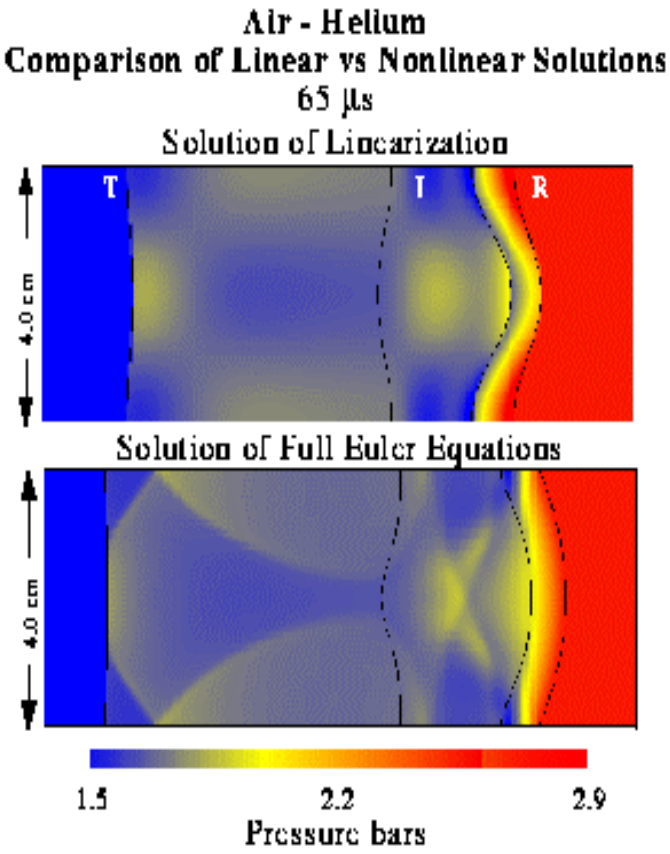
Bubble and spike velocities at a shocked air-SF_6 interface

The full amplitude growth rate is defined as $da/dt(t) = (v_{spike} - v_{bubble})/2$. Note how certain events in the amplitude growth rate occur at either the spike or the bubble, but not both.



Comparison of linear and nonlinear air-SF_6 solutions





The figures show comparisons of simulations of the Benjamin air-SF_6 experiment and the Meshkov air-He experiments with the solution to the linearized equations. The comparisons are at t = 195 µs on the left and 65 µs on the right. A key feature is that the linearized solutions fail to capture the sharp shock fronts seen in the full Euler solutions.

XHM Single-Mode Richtmyer-Meshkov Instability: Experiment, Simulation and Theory

Guy Dimonte, Lawrence Livermore National Laboratory
Bruce Fryxell, Goddard Space Flight Center
Michael L. Gittings, Science Applications International Corporation
John W. Grove, Los Alamos National Laboratory
Richard L. Holmes, Los Alamos National Laboratory
Marilyn Schneider Lawrence Livermore National Laboratory
David H. Sharp, Los Alamos National Laboratory
Robert P. Weaver, Los Alamos National Laboratory
Alexander L.Velikovich, Berkeley Research Associates
Qiang Zhang, University at Stony Brook

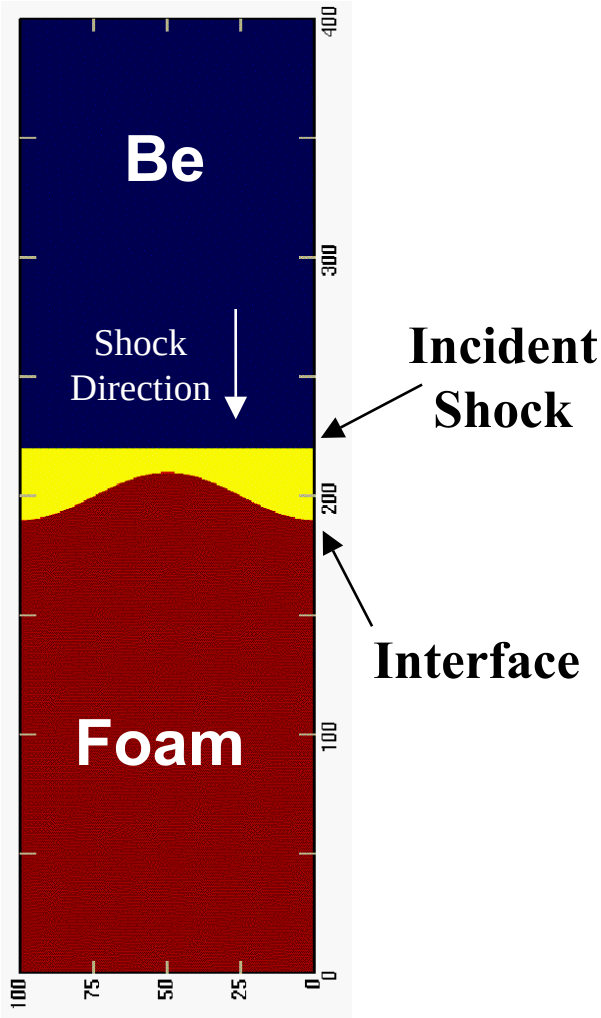
Description and goals of work

Study shock-induced interfacial instability between gases (Richtmyer-Meshkov Instability) through experiments, simulations and analytic models

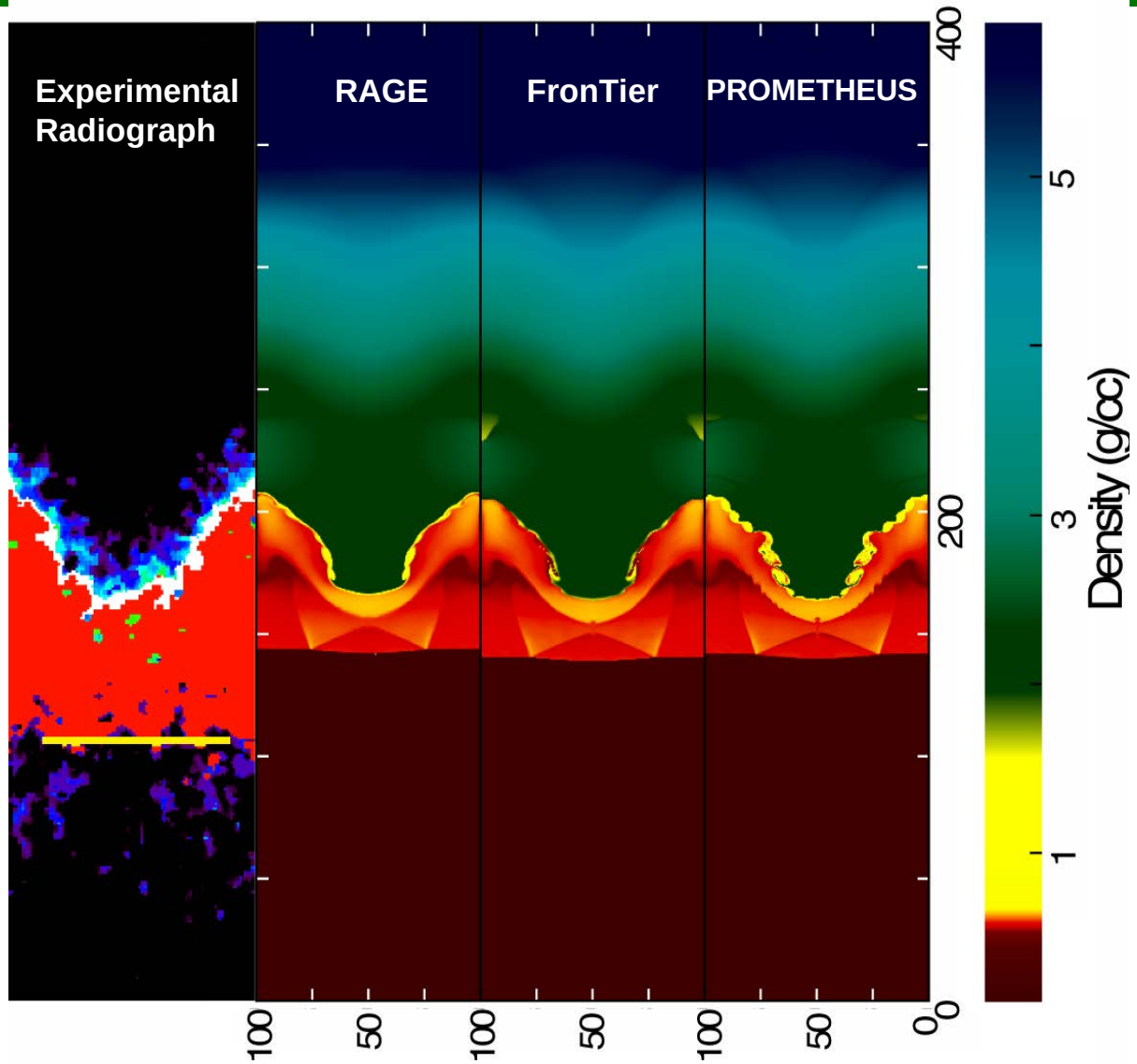
- Compare three independent codes
  - RAGE: AMR, 2nd order Godunov hydro
  - PROMETHEUS: Uniform grid, PPM hydro
  - FronTier: Front tracking coupled with 2nd order Godunov hydro
- Validate codes against experimental data in a particularly interesting regime
  - Improved confidence in predictive capabilities of simulations
- Study of validity and behavior of simulations and theories over a wide range of conditions
  - Seven combinations of initial shock strength and initial perturbation amplitude
- Improve understanding of relationship between experiment, simulation and theory
- Variety of researchers involved helped

Richtmyer-Meshkov Experimental Configuration

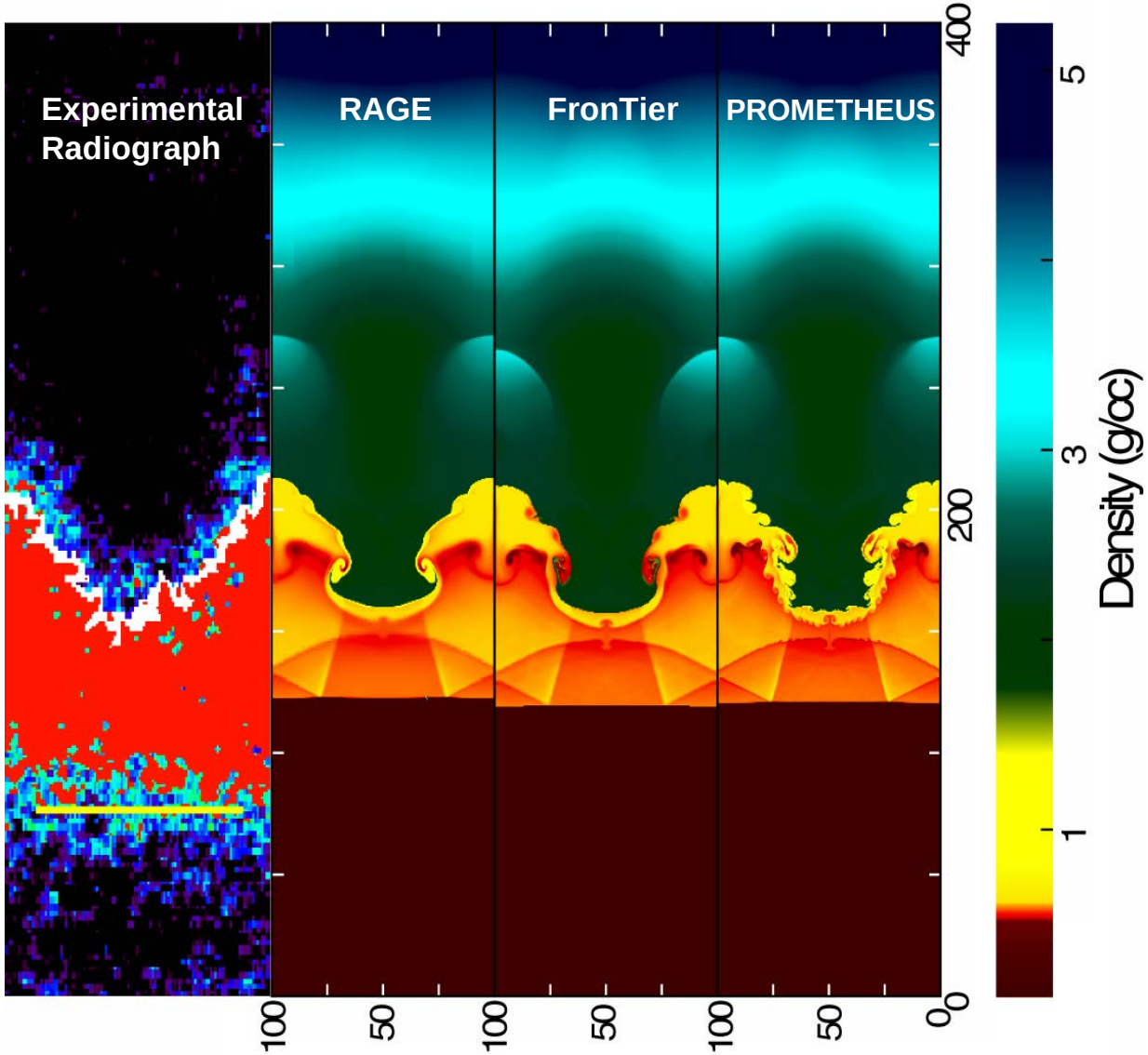
- Beryllium mated to foam with a perturbed interface between them
  - Plasmas at time of data acquisition
  - Materials modeled as perfect gases
- Incident shock generated by ablation at rear surface of beryllium
- Perturbations: single-mode sinusoidal
  - wavelength 100 µm, amplitude 10 µm
- Strong incident shock
  - Mach 15.3



Experiment and Simulation Densities at 2.8ns



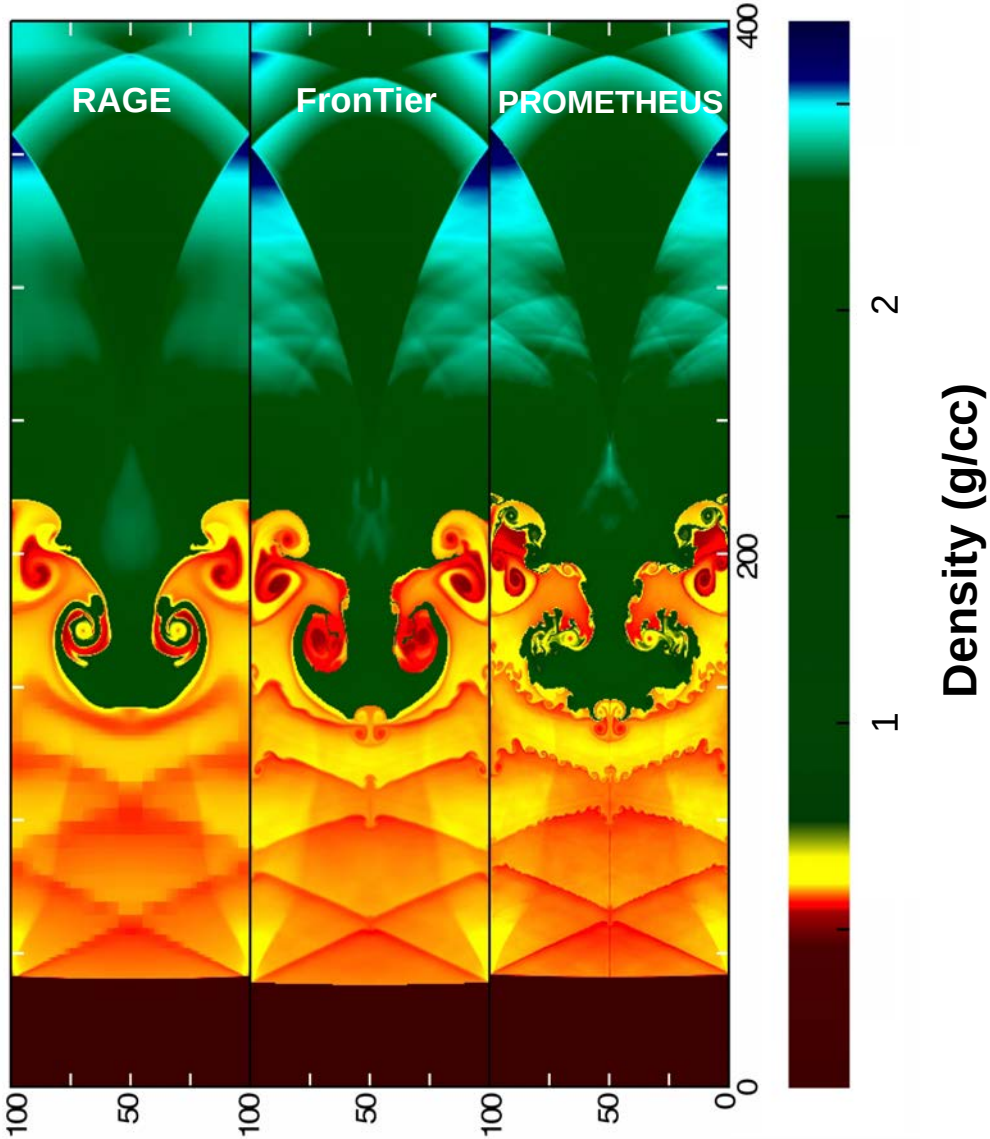
Experiment and Simulation Densities at 4.2ns



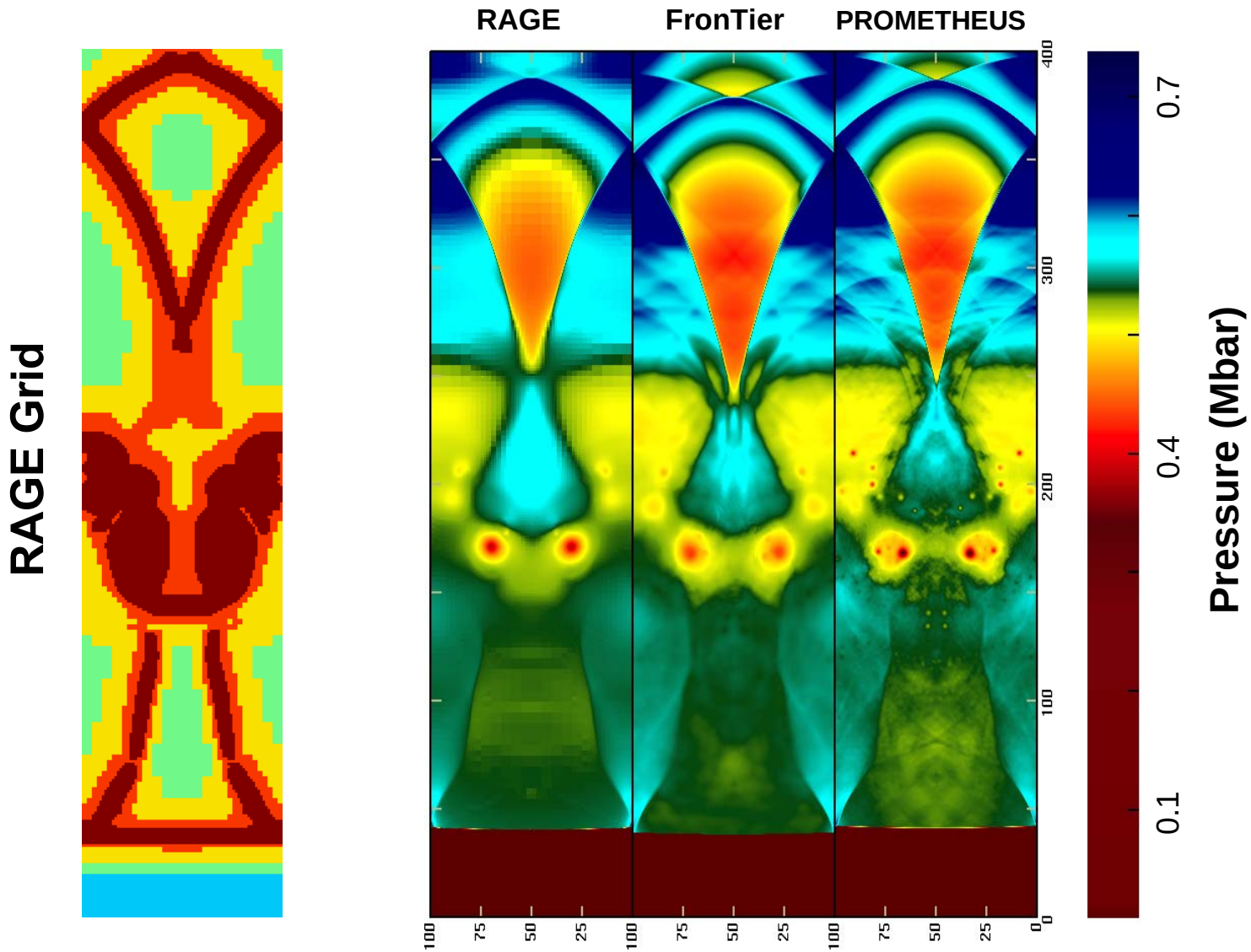
Simulation Only Densities at 9.6 ns

Do any of the codes have the secondary structure right?

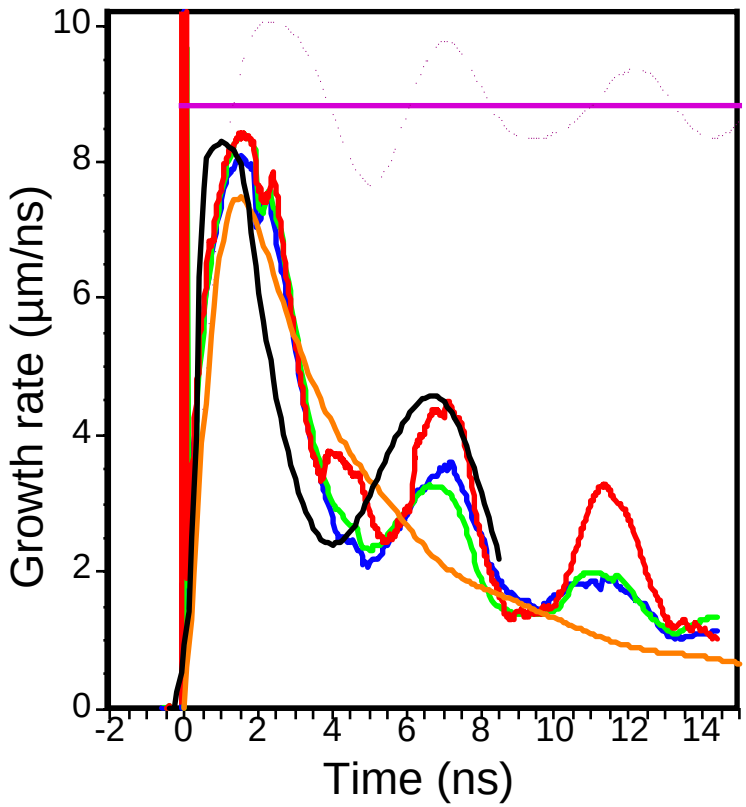
How would we know?

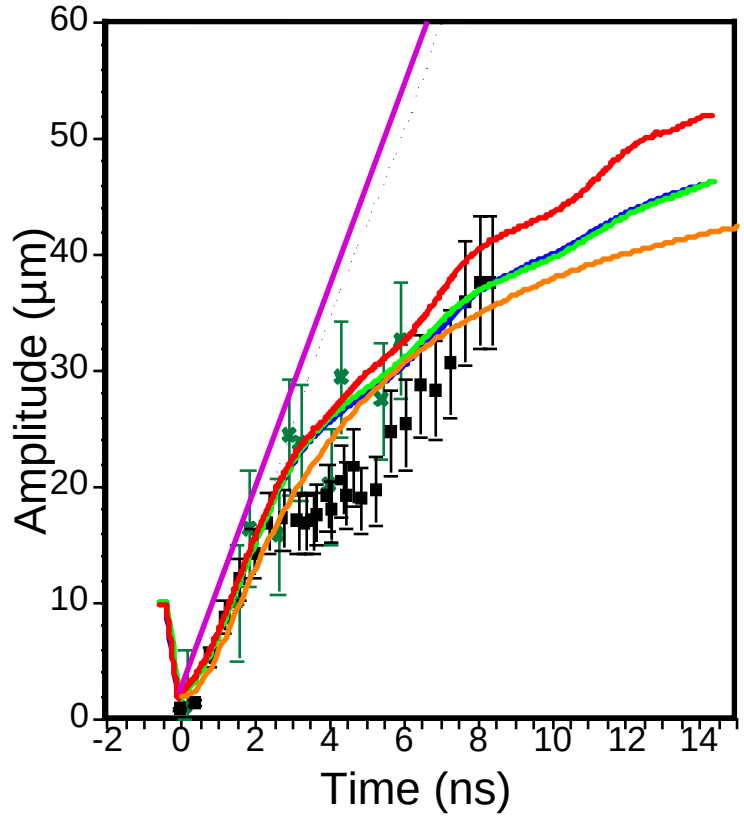


Pressure and RAGE AMR Grid 9.6 ns



Amplitude/Growth Rate Comparison





FronTier	Impulsive Model	Experimental Growth Rate
RAGE	Linear Theory	Experiment/Face-on Ampl.
PROMETHEUS	Zhang/Sohn	Experiment/Side-on Ampl.

Cylindrical Geometry Version of Dimonte Experiment (1)

Foam

$\overline{\rho} = 0.12\, g/cc,\ \sigma_\rho = 0.06\, g/cc,$ correlation length $= 20\, \mu m$

$P = 0.1\, Mbar$

$\mathbf{v} = 0$

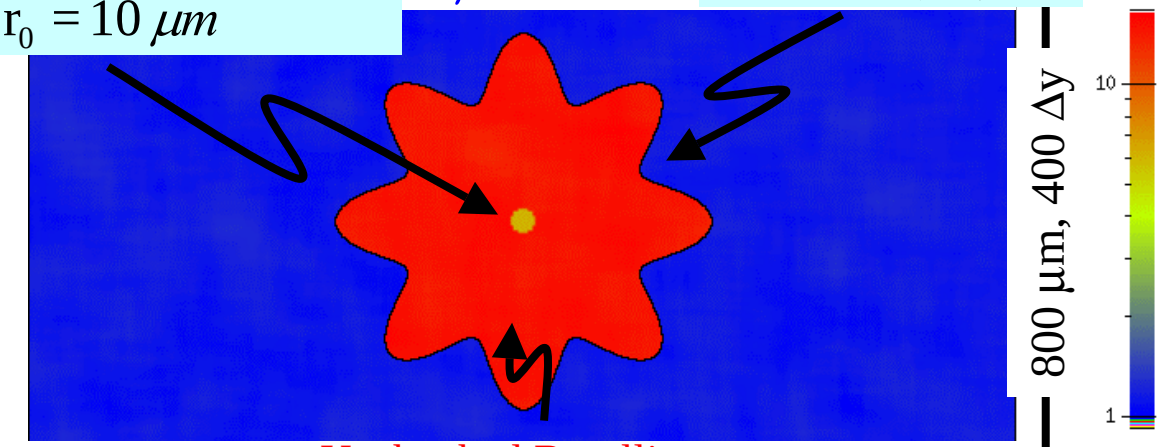
$\gamma = 1.45$

Mach 30 Expanding Shock

$r_0 = 10\ \mu m$

Material Interface

$r = 25\cos(8\theta)\ \mu m$



800 µm, 400 Δy

Unshocked Beryllium

$\overline{\rho} = 1.7\, g/cc,\ \sigma_\rho = 0.85\, g/cc,$ correlation length $= 20\, \mu m$

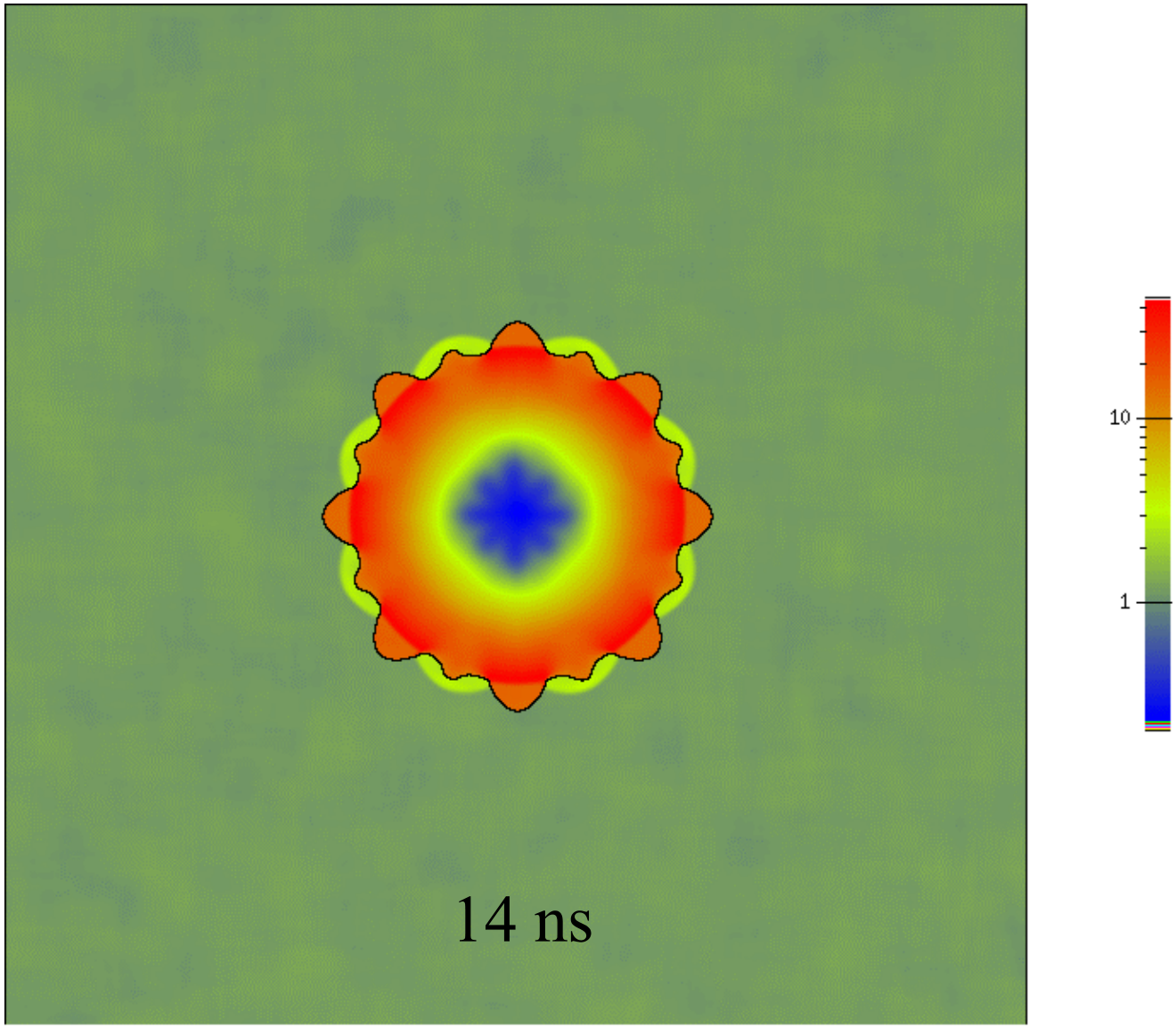
$P = 0.1\, Mbar$

$\mathbf{v} = 0$

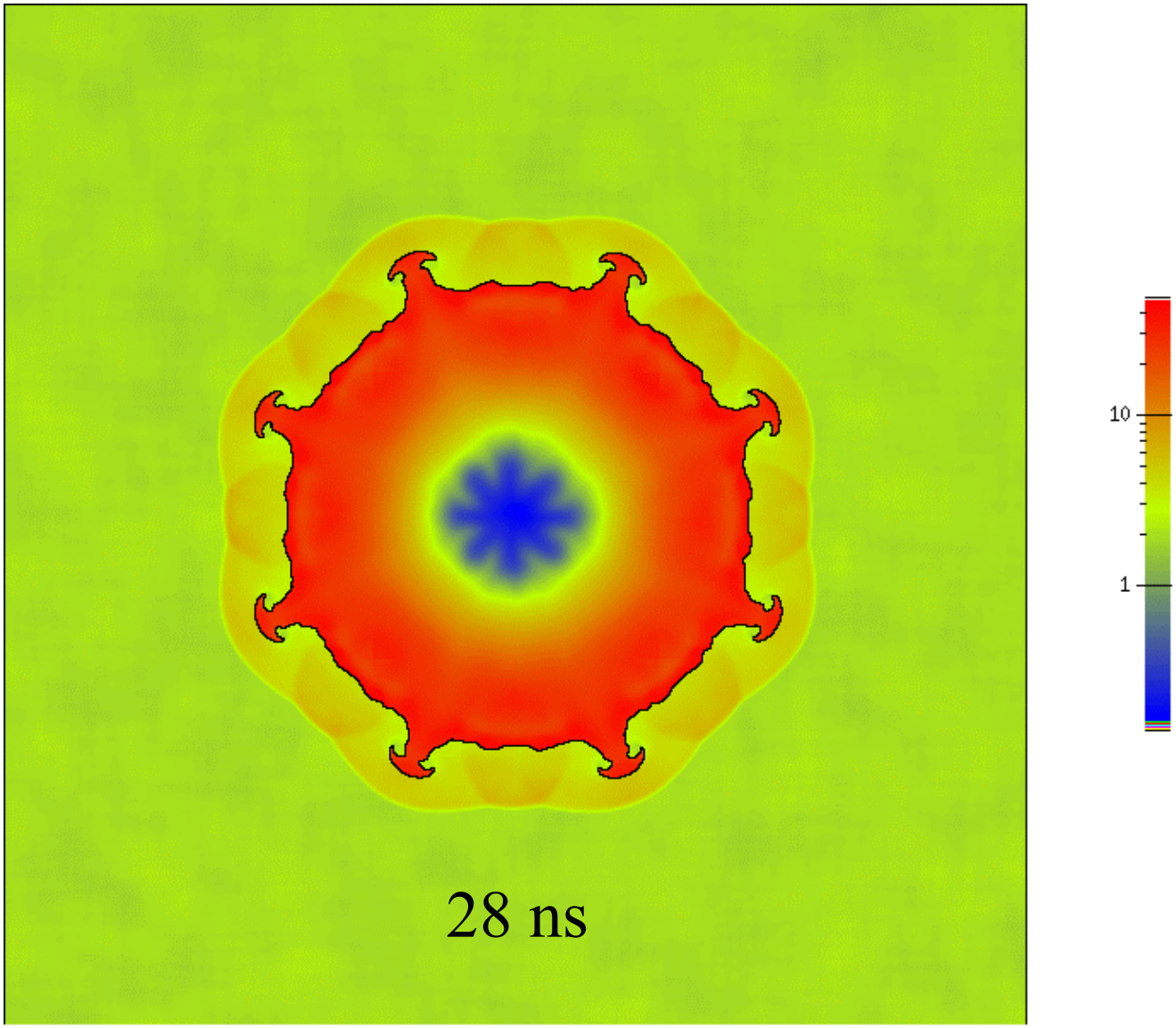
$\gamma = 1.8$

800 µm, 400 Δx

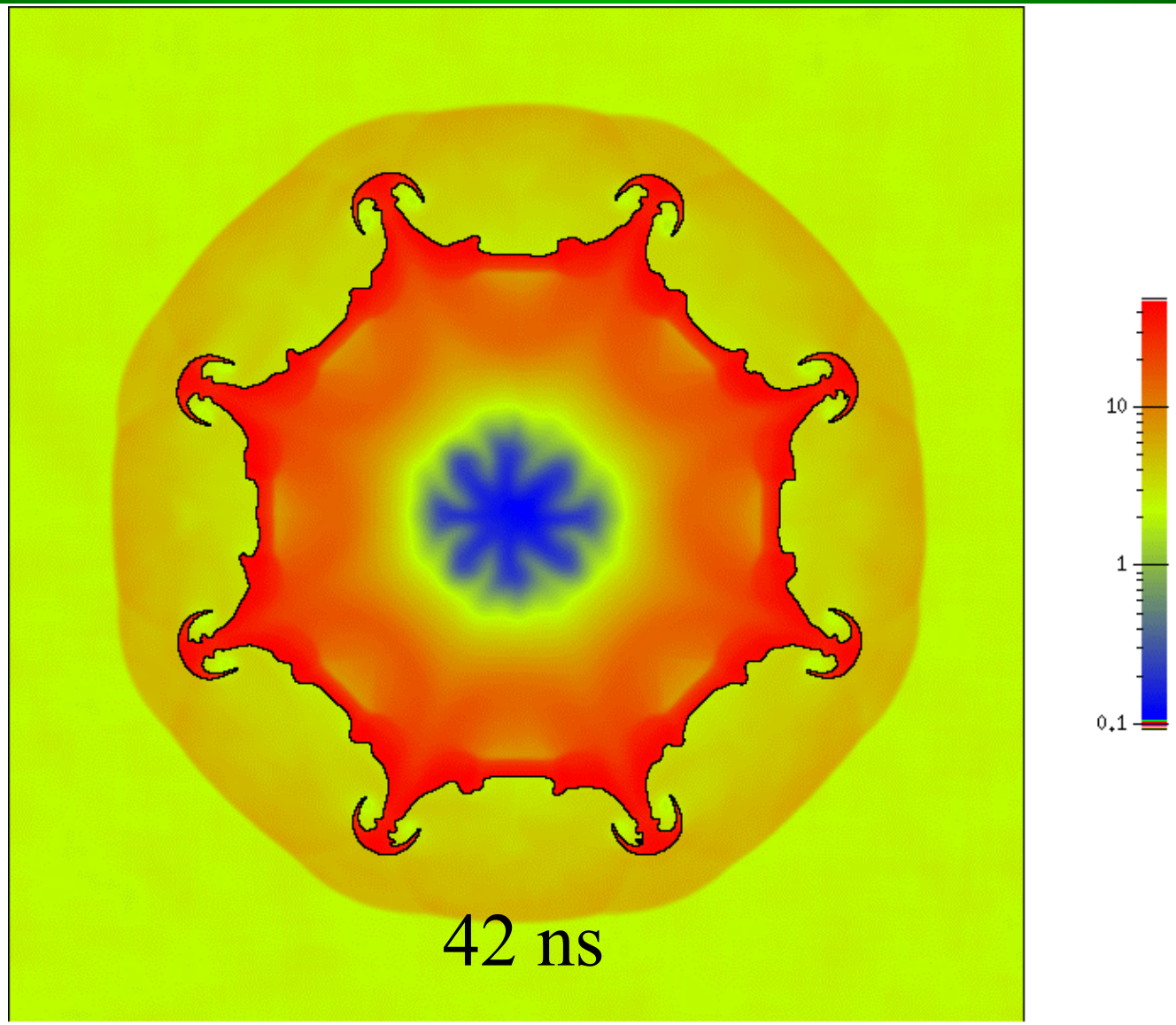
Cylindrical Geometry Version of Dimonte Experiment (2)



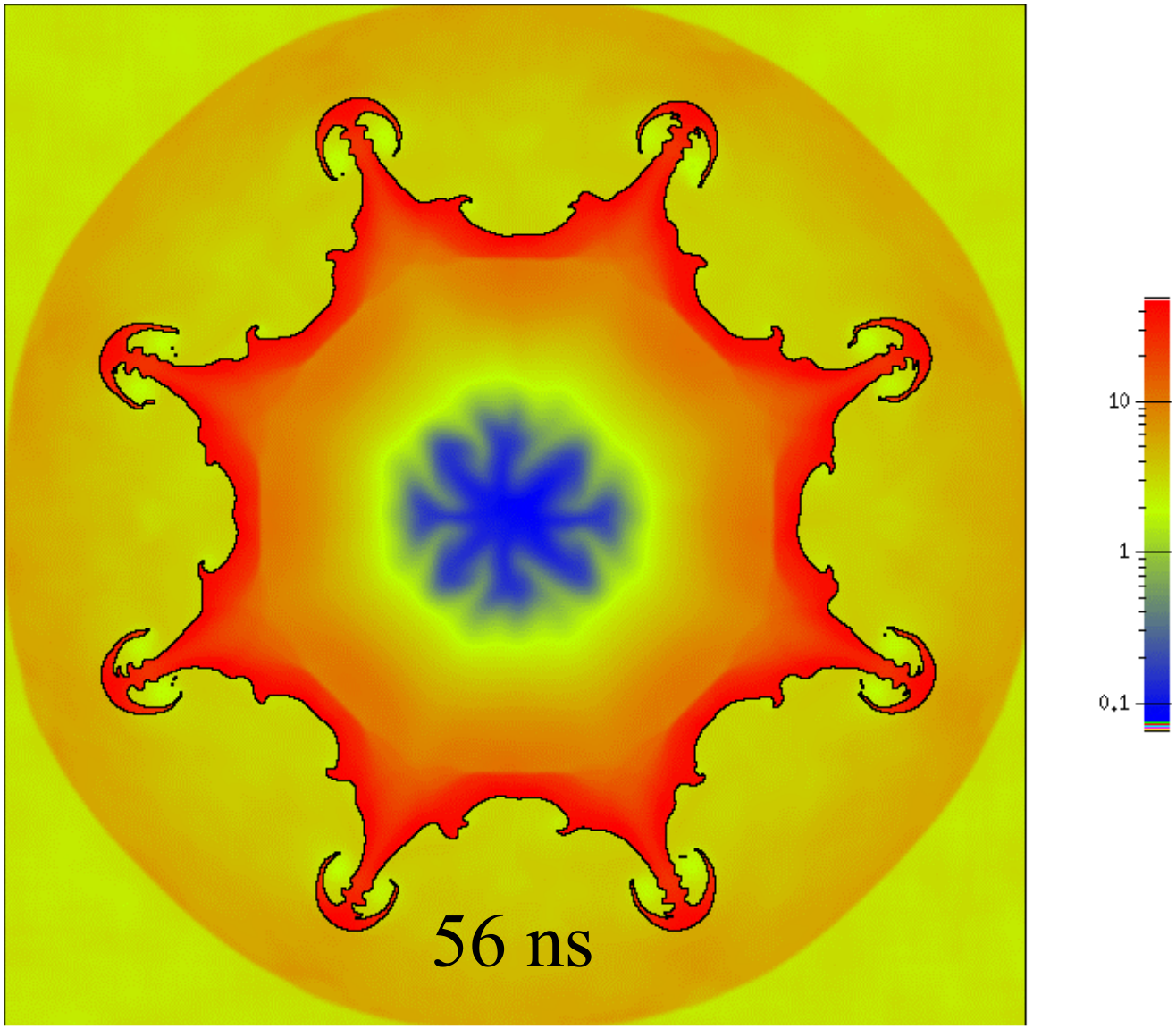
Cylindrical Geometry Version of Dimonte Experiment (3)



Cylindrical Geometry Version of Dimonte Experiment (4)



Cylindrical Geometry
Version of Dimonte Experiment (5)



Supersonic Random Velocity Inlet (1)

The next computation is a simulation of the supersonic injection of a gas into a closed box. The initial flow parameters are:

Region Outside of Jet

$$\text{Air}, g = 1.4$$
$$\rho = 1\,\text{g/l}$$
$$P = 1\,\text{bar}$$
$$v_x = v_y = 0$$

Inlet, width = 2.5 cm

$$\text{Air}, \gamma = 1.4$$
$$\rho = 10\,\text{g/l}$$
$$P = 10\,\text{bar}$$
$$\bar{v}_x = 0\,\text{cm/sec}$$
$$\sigma(\bar{v}_x) = 6.3\ \text{cm}/\mu\text{sec}$$
$$v_y = 31.6\ \text{cm}/\mu\text{sec}$$
$$\sigma(\bar{v}_y) = 6.3\ \text{cm}/\mu\text{sec}$$

correlation length = 0.25 cm

correlation time = 7.9 μsec

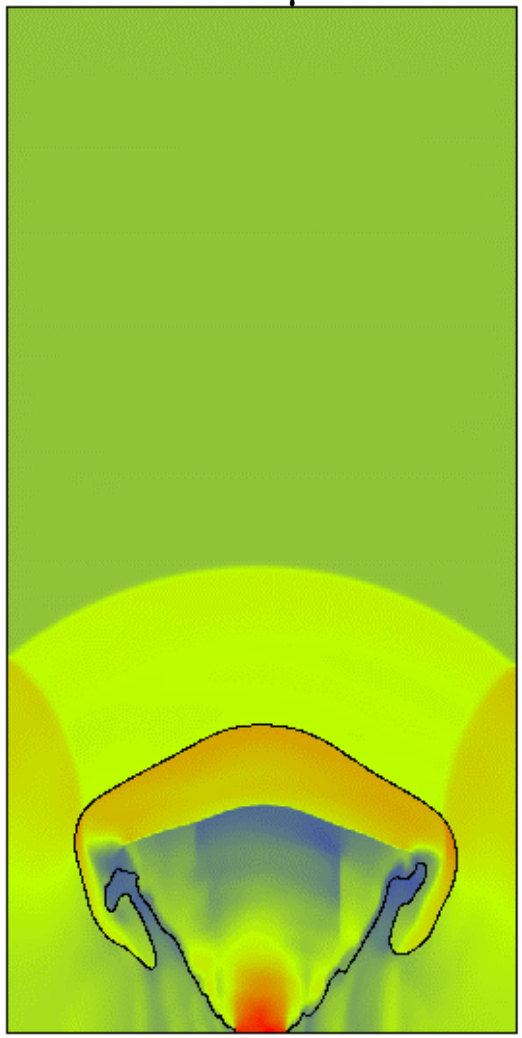
Computational Region:

width 25 cm, 400 Δx

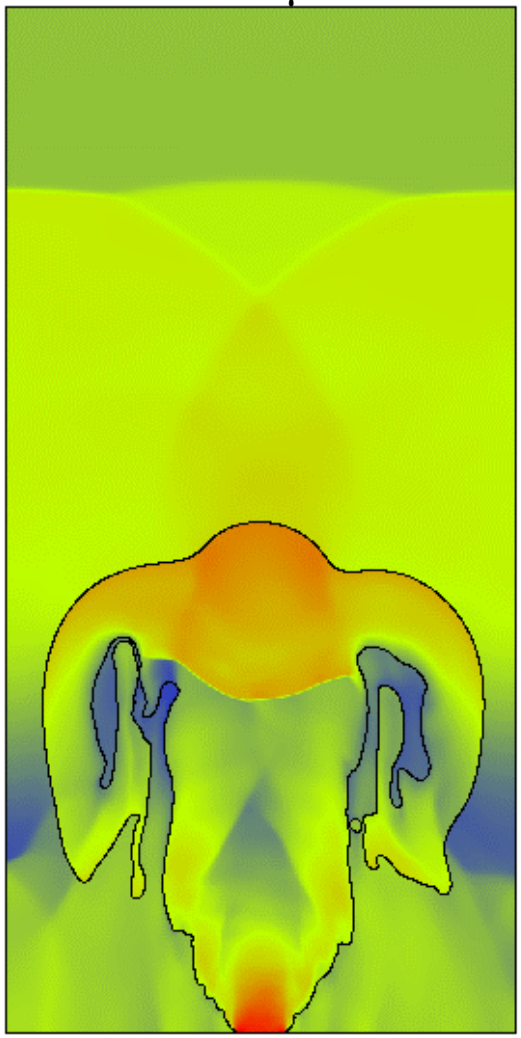
height 50 cm, 800 Δy

Supersonic Random Velocity Inlet (2)

t = 380 μsec

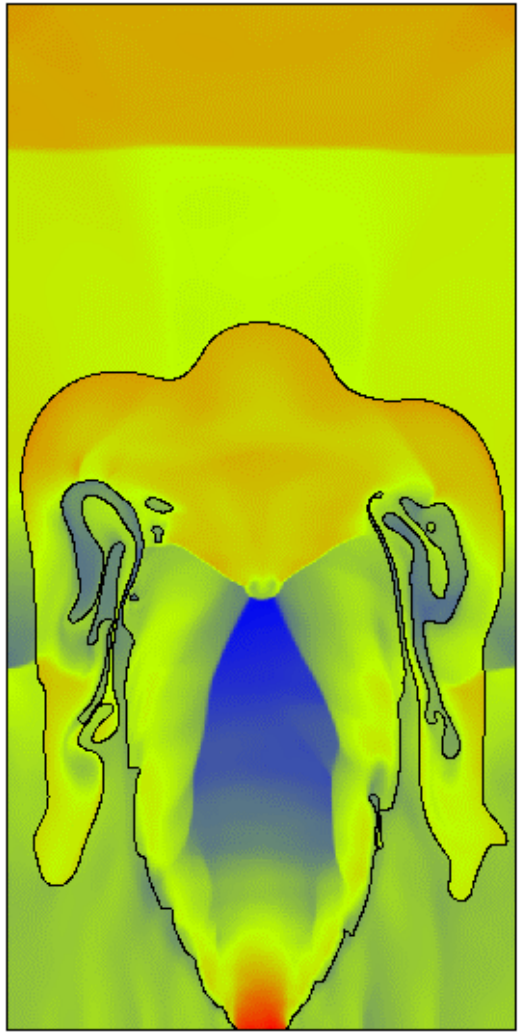


t = 760 μsec

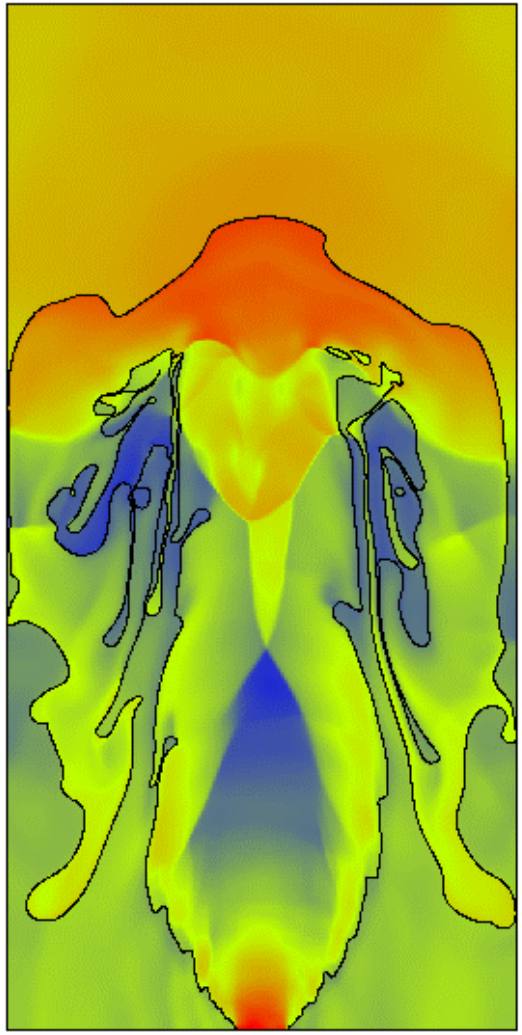


Supersonic Random Velocity Inlet (3)

t = 1.140 msec



t = 1.52 msec



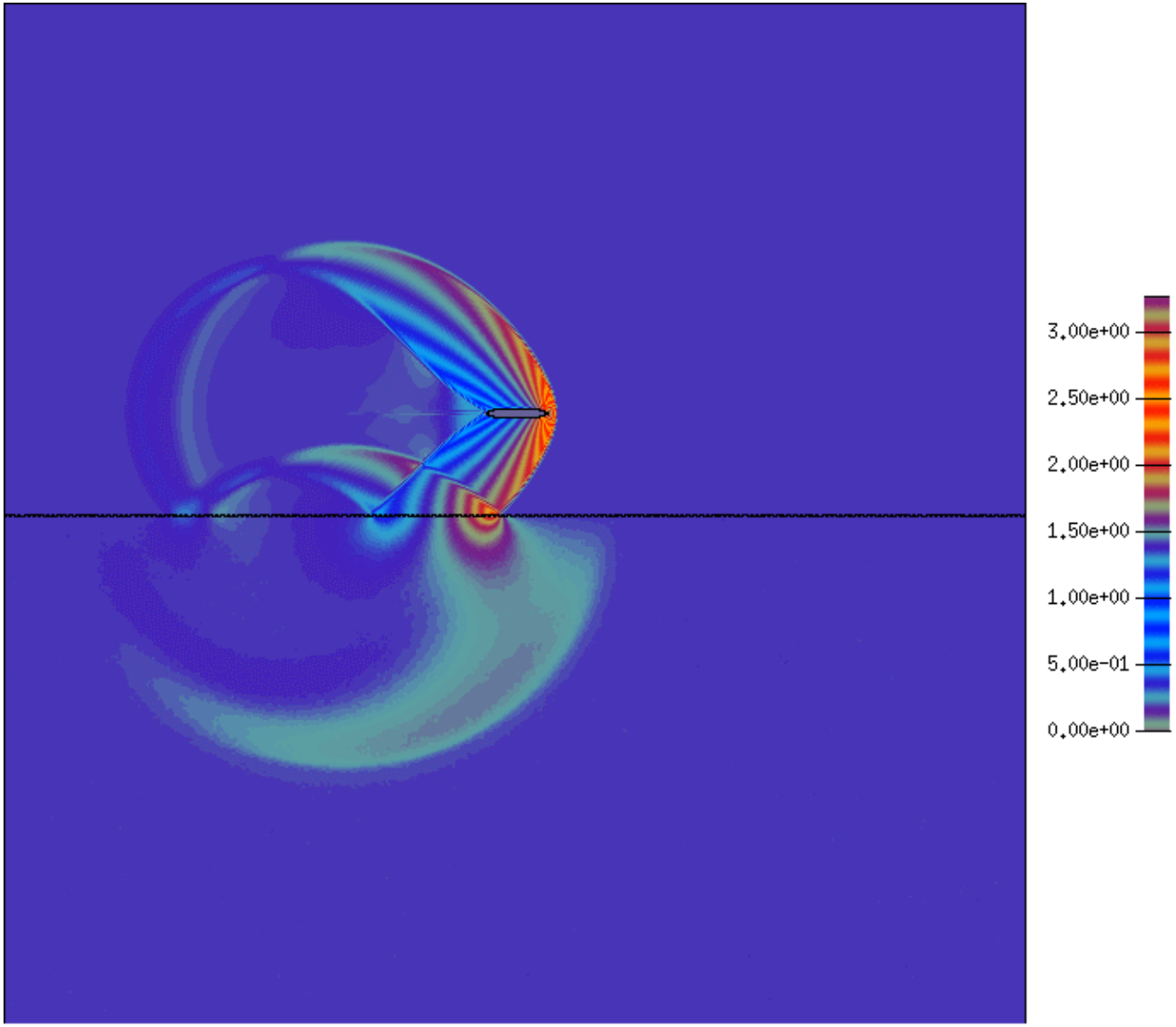
Supersonic Projectile over Water (1)

These calculations are intended to study the behavior of the refraction of a bow wave generated by a supersonic projectile through the surface of the ocean. The calculations show a projectile with the shape of an ellipse of diameters 3 by 0.5 meters, traveling at 500 meters per second at an altitude of 5 meters above a flat body of water. The projectile is traveling in air, which is modeled as a perfect gas with $\gamma = 1.4$ and density of 1 gram/liter and ambient pressure of 1 bar. The water is modeled as a stiffened polytropic gas, which has an equation of state of the form:

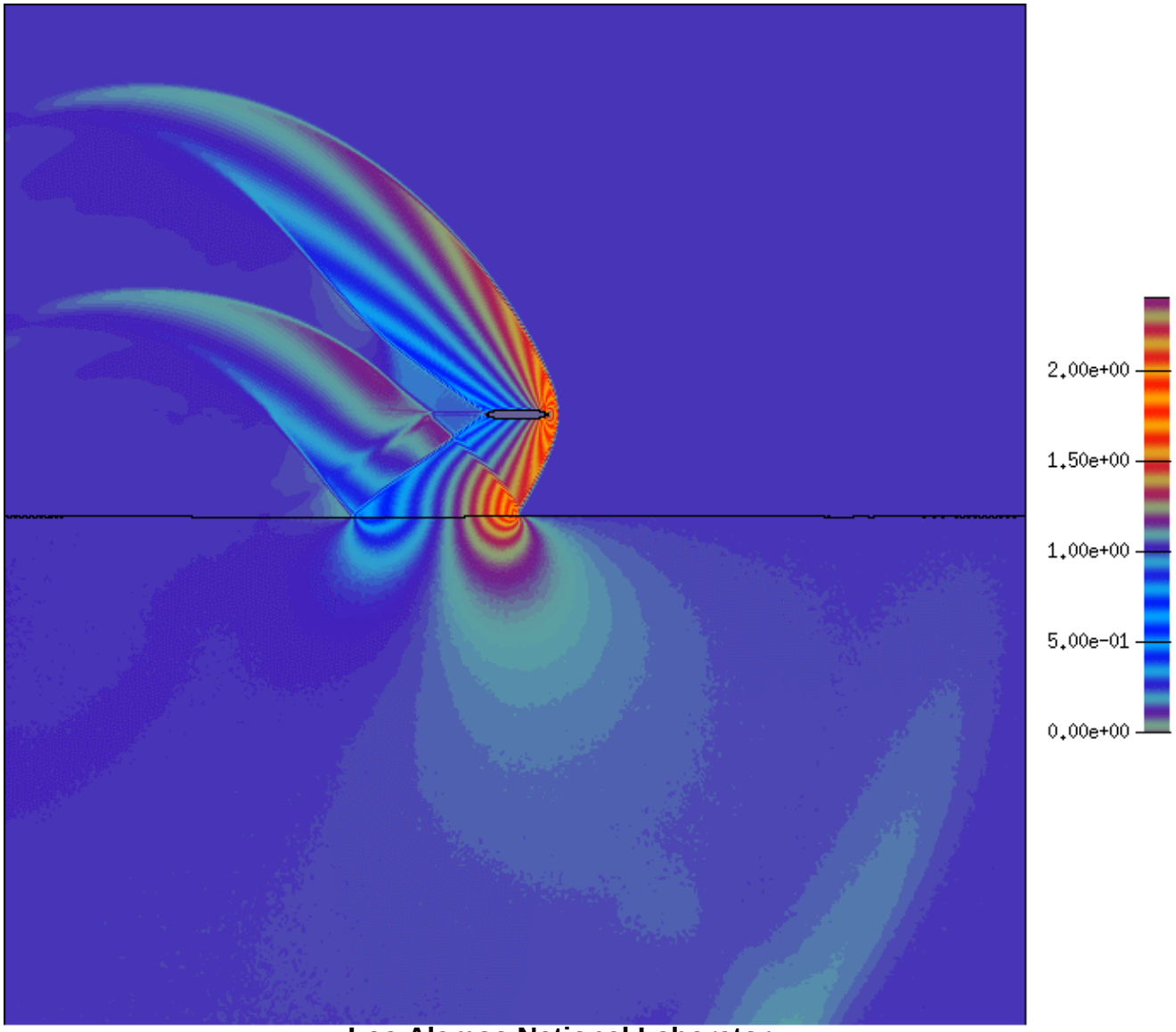
$$\rho e = \frac{P}{\Gamma} + E_\infty,\ E_\infty = \frac{\Gamma + 1}{\Gamma} P_\infty,\ \Gamma = 6,\ \text{and},\ P_\infty = 3040 \text{ bar}$$

and $\rho = 1$ gram/liter.

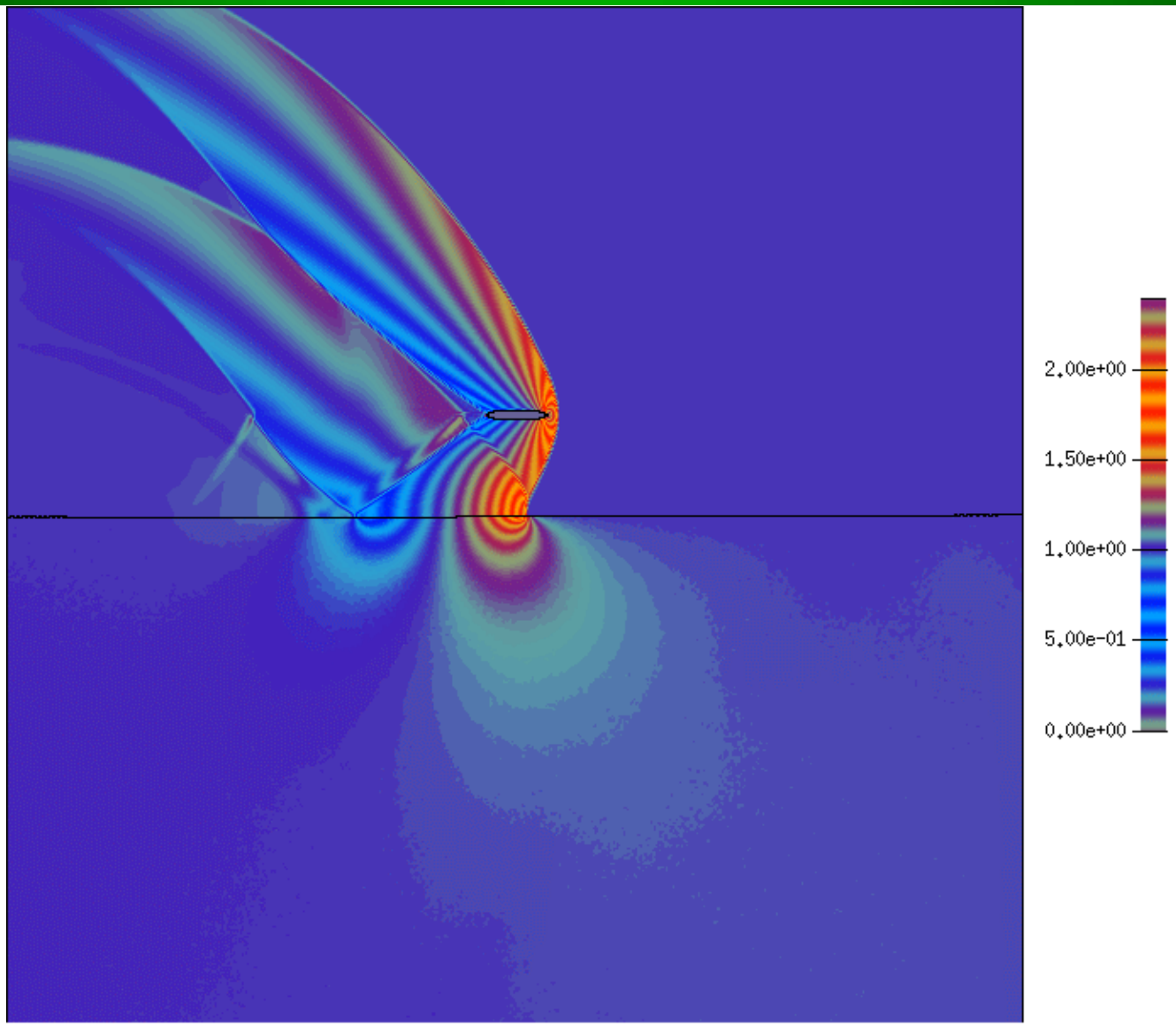
Supersonic Projectile over Water (20 msec)



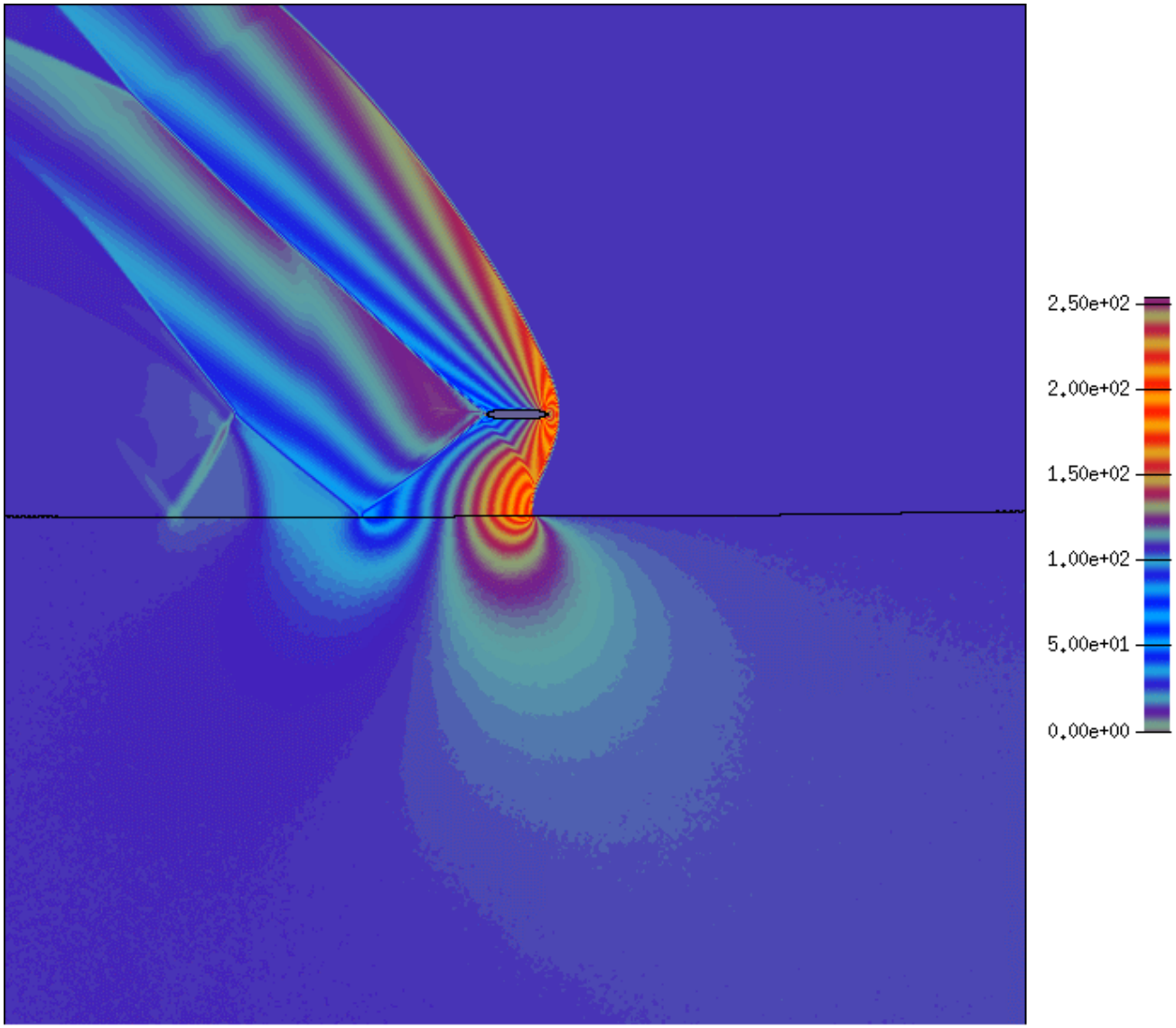
Supersonic Projectile over Water (40 msec)



Supersonic Projectile over Water (60 msec)



Supersonic Projectile over Water (80 msec)



Open Problems and Current Research (1)

- **Conservation and Front Tracking**
  - Algorithm makes conservation error on the order of the truncation error near the front. These are generally first order.
  - Exact conservation requires the construction of adaptive *space-time* grids near the front. Conservation means that only waves that satisfy the Rankine-Hugoniot conditions can support discontinuities. Sharp fronts require that volume cells must not cross fronts. Thus the finite volume cells must be aligned in space and time with the discrete fronts if exact conservation of all quantities is to be imposed.
- **Adaptive mesh algorithms and front tracking**. Much of the infrastructure of *FronTier* assumes a regular rectangular grid. Current development efforts are being directed towards replacing the rectangular grid with a binary by dimension tree structured grid.
- **Improved Three Dimensional Capabilities**: continue to improve the robustness and performance of the three dimensional front tracking code to provide a computational tool for studying complex flows.

Open Problems and Current Research (2)

- **More General Physics:**
  - **Elasto-plasticity:** *FronTier* now supports an elasto-plasticity library. Development is proceeding to implement improved solvers and interface propagation algorithms in this module.
  - **Multiphase flow models:** many new theoretical models are being invented to describe meso-scale mixing in which the fluids are separate at a molecular level, but are entrained at length scales that are very small with respect to the computational grid spacing. Designing and implementing these methods is a major thrust in our development effort.
- **Improved Performance on Parallel Computers:** These are the machines we have, we want to use them as effectively as possible. *FronTier* already supports domain decomposition parallelism, continuing work will be devoted to scaling the code from the current 10-100 processor mode to as many as 1000 processors.

Open Problems and Current Research (3)

- **Improved usability of the code:**
  - Add a graphical user interface for problem setup.
  - Enhance and add to the diagnostic capabilities of the code.
- **Incorporation of new advances in finite different and finite element technology.**
  - Modular structure of the code can be improved to allow easier insertion of new finite difference methods.
  - Implement unsplit solvers.
- **Theoretical and Numerical Investigations**
  - Study the behavior of fluid mixing zones generated by Rayleigh-Taylor and Richtmyer-Meshkov instability.
  - Study the role of vorticity in the growth of unstable features on an interface.

Further Reading

1. J. W. Grove, R. L. Holmes, D. H. Sharp, Y. Yang, and Q. Zhang, *Quantitative theory of Richtmyer-Meshkov instability*, Phys. Rev. Lett. **71** pp. 3473-3476 1993.

2. R. L. Holmes, J. W. Grove, and D. H. Sharp, *Numerical investigation of Richtmyer-Meshkov instability using front tracking*, J. Fluid Mech. **301** pp. 51-64, 1995.

3. Y. Chen, J. Glimm, D. H. Sharp, and Q. Zhang, *A two-phase flow model of the Rayleigh-Taylor mixing zone*, Phys. Fluids, **8**, pp. 816-825 1996.

4. J. Glimm, X.-L. Li, R. Menikoff, D. H. Sharp, and Q. Zhang, *A numerical study of bubble interactions in Rayleigh-Taylor instability for compressible fluids*, Phys. Fluids A, **2** pp. 2046-2054, 1990.

5. 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**, pp. 703-727 1998.

6. D. L. Youngs, *Numerical simulation of turbulent mixing by Rayleigh-Taylor instability*, Physica D, **12**, pp. 32-44 1984.

7. X.-L. Li, *Study of three dimensional Rayleigh-Taylor instability in compressible fluids through level set method and parallel computation*, Phys. Fluids A, **5**, pp. 1904-1913 1993.