



Front Tracking in Two Space Dimensions

LA-UR 99-3985

8/2/99

[Click here to start](#)

Table of Contents

Author: John W. Grove

Email: jgrove@lanl.gov

[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 form $\phi(\mathbf{x}, t) = 0$, 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:

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

and the characteristic equation becomes:

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

That is λ is an eigenvalue of the matrix:

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

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 $\lambda_k(\mathbf{u}(\mathbf{x},t),\xi)$ 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:

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

Bicharacteristics for the Euler Equations (1)

For the Euler equations the characteristic speeds are:

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

and the corresponding bicharacteristic rays:

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

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

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

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

Bicharacteristics for the Euler Equations (2)

We can write a bicharacteristic form for the Euler equations:

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

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

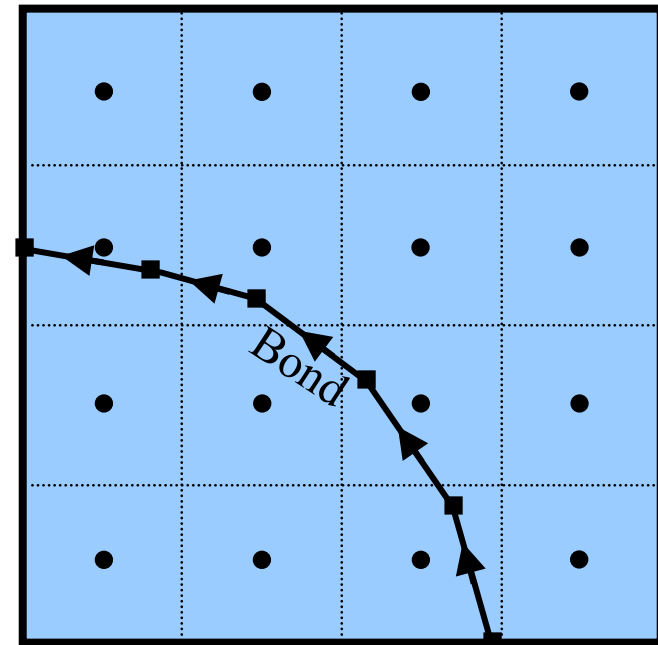
$$\mathbf{T} \cdot \frac{D\mathbf{u}}{Dt} + \frac{\mathbf{T} \cdot \nabla P}{\rho} = 0, \quad \mathbf{T} \cdot \mathbf{u} = 0$$

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

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 \mathbf{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, **POINTS** 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**: **BONDS** are directed connectors between pairs of **POINTS**. 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 **BONDS** to be connected into piecewise linear curve segments.
- **TRI**: A **TRI** corresponds to a geometric triangle. Each **TRI** contains pointers to three **POINTS** that form the vertices of the triangle. A **TRI** contains pointers to three adjacent **TRIs** called its neighbors so that **TRIs** 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 **BONDS** can be linked together to form lists, it suffices for a **CURVE** structure to contain pointers to the first and last **BOND** on the **CURVE**. The set of **POINTS** on the curve can be accessed by looping through the **BONDS** 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 **CURVES** are called **NODES**. Each **CURVE** is associated with two (possibly the same) **NODES** that mark the start and end of that **CURVE**. Each **NODE** has an associated **POINT** that gives its position. **NODES** are the only allowed locations where two or more **CURVES** can meet. **NODES** contain lists of pointers to the **CURVES** that meet at that **NODE**.
- **SURFACE**: **SURFACES** are three dimensional structures that correspond to a piecewise linear surface. They consist of a collection of connected **TRIS** and a set of **CURVES** 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 (possibly 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 **POINTS**, for two dimensional flows **CURVES**, and for three dimensional flows **SURFACES**. 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 **HYPERSURFACES** 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** : **HYPERSURFACES** are composed of collections of **HYPERSURFACE ELEMENTS**. Depending on the flow dimension there are either empty, **BONDS** or **TRIs** for respectively one, two, or three space dimensional flows. Currently there are no topological properties associated with **HYPERSURFACE ELEMENTS**, 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 **NODEs**, while for three dimensional flows they are **CURVEs**. 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

FrontTier 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

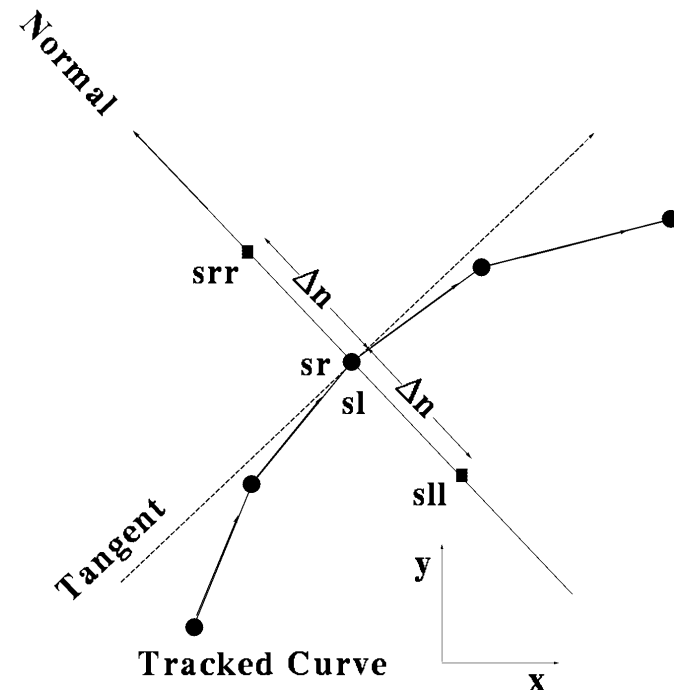
Front Tracking Operators

- 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

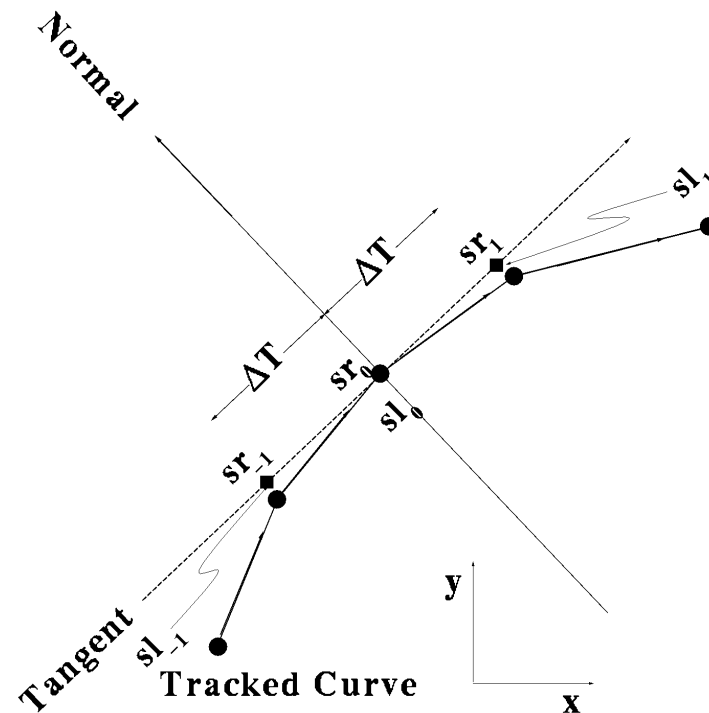
We will complete this lecture with a description of the two dimensional point propagate operator that advances the location and states of a point on a tracked front. This operation uses operator splitting by dividing the propagation step into two phases, a normal propagate and and tangential propagate.

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

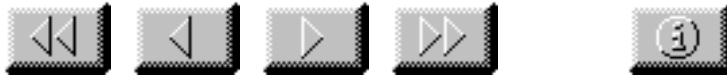
1. Show that if a bicharacteristic ray intersects a characteristic surface of the same family, then that ray lies entirely inside the characteristic surface.
2. A shock wave is said to belong to the k^{th} characteristic family and to be stable in the sense of Lax if the shock speed satisfies the inequality:

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

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](#)

Table of Contents

Author: John W. Grove

Email: jgrove@lanl.gov

[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.

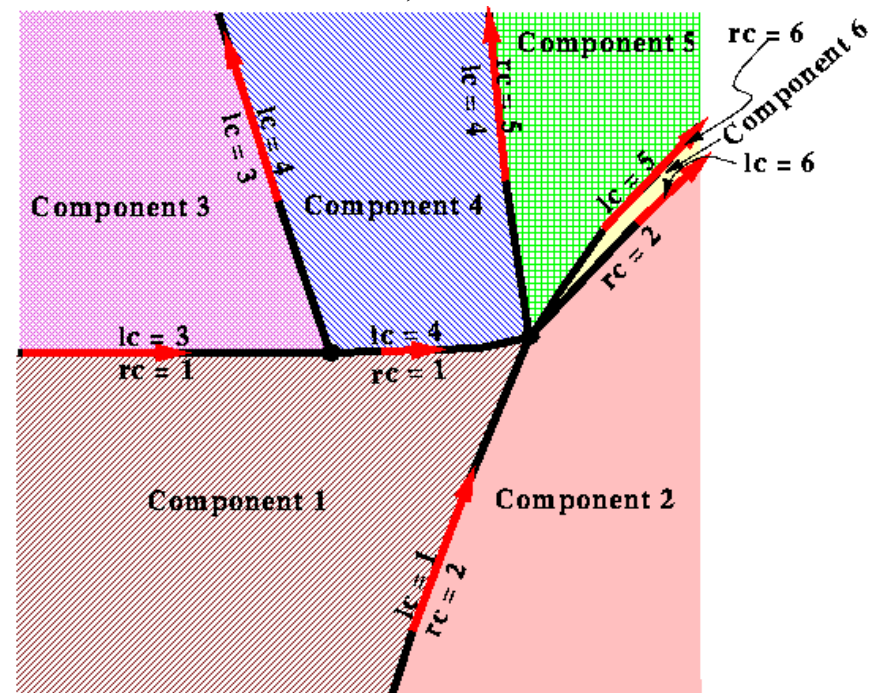
Front Tracking Utilities (4)

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 considerably 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 grid 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 $cell(i,j)$

$bond(i,j,k)$, $k = 1, \dots, N(i,j)$

$curve(i,j,k)$, $k = 1, \dots, N(i,j)$

$comp(i,j,k)$, $k = 1, \dots, M(i,j)$

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.

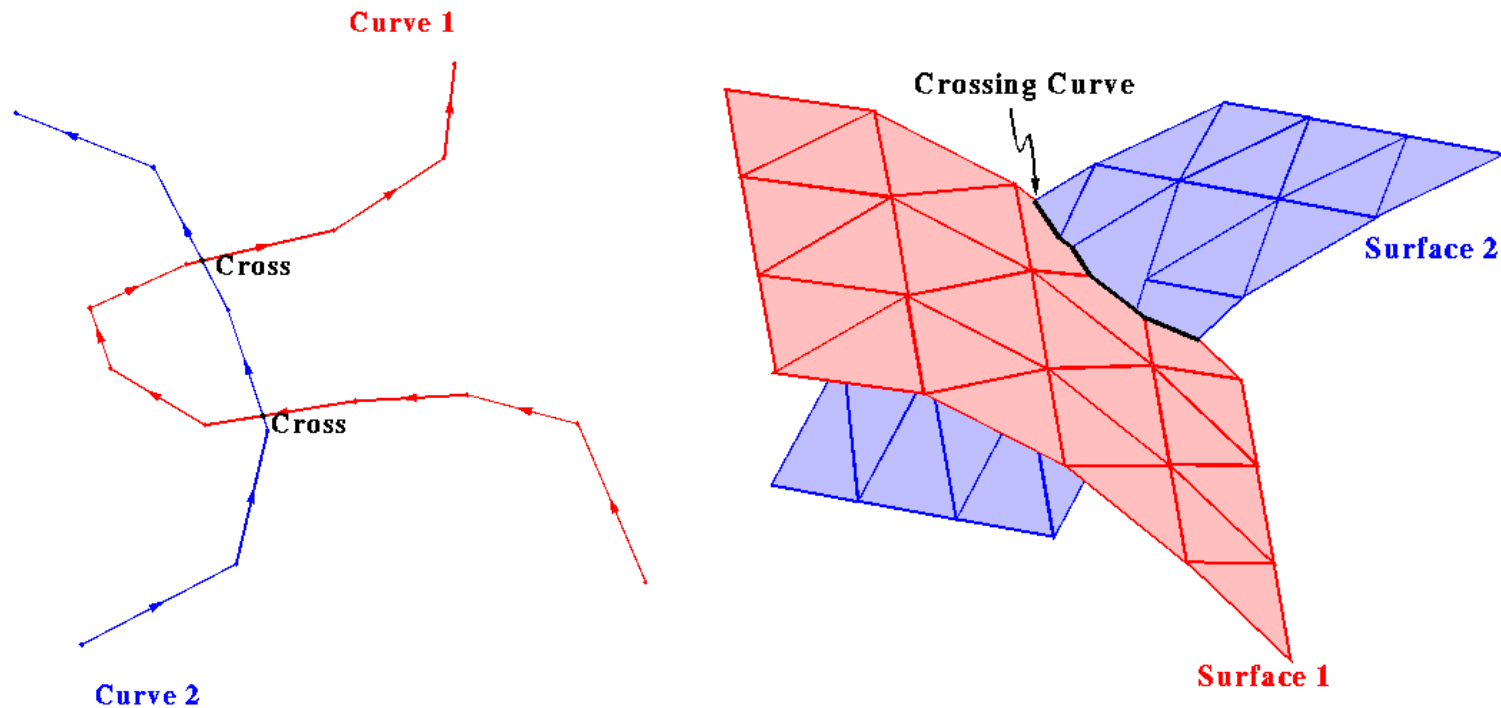
Utilization of Topology Lists

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

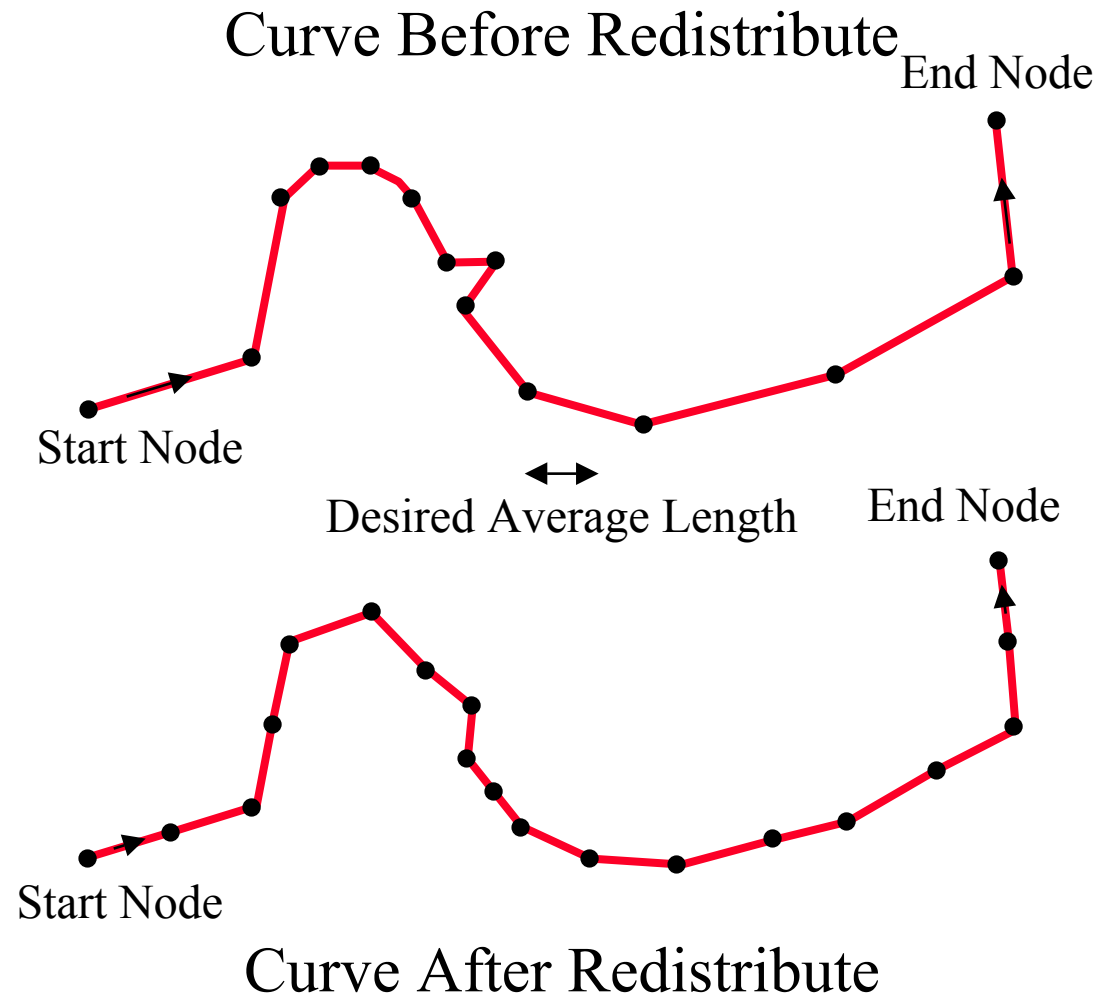


The two diagrams show examples of two and three dimensional crossings. Note that in two space dimensions crosses are discrete and thus it is necessary to process the cross list to identify corresponding crosses. Any formation of interface crossing indicates a wave interaction that must be resolved.

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.

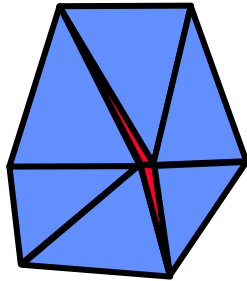
Redistribution Example



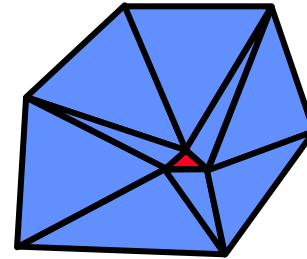
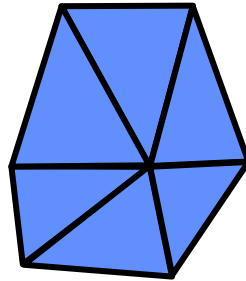
Examples of Triangle Redistributions



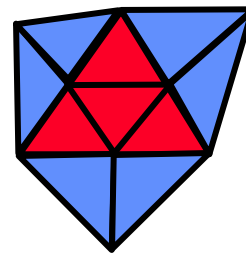
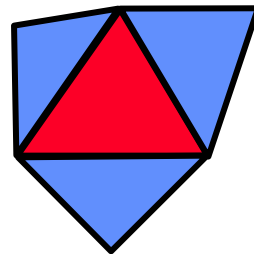
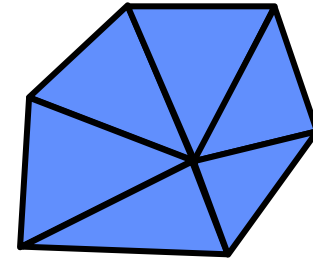
Flip Diagonals



Merge Edge



Delete Small Triangle



Split Triangle

Remarks on Redistribution

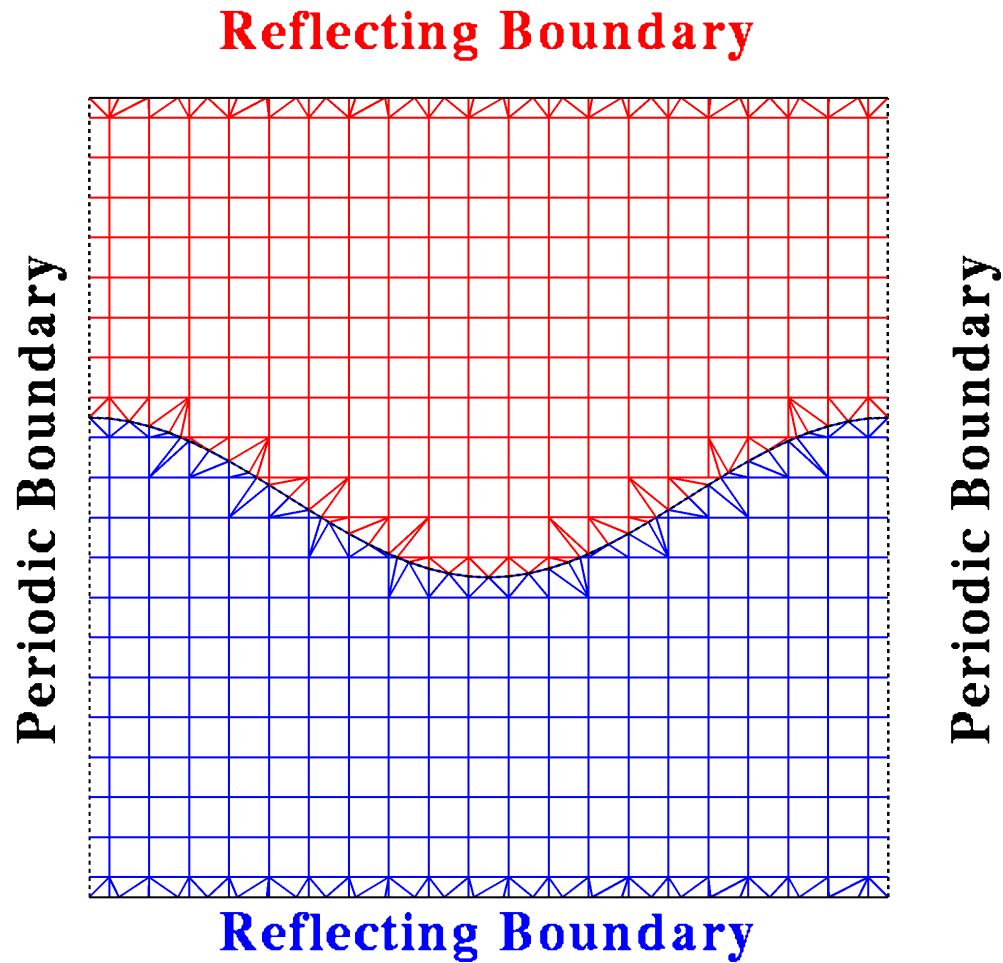
The need for interface redistribution is a necessary evil in the front tracking method and must be treated with care. Redistribution introduces an artificial surface force that has a stabilizing effect similar to surface tension. It is important that this operation be applied with the minimum frequency 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 tetrahedrization 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

FrontTier 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 its 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 replaced 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 reassemble the interface. This method has the advantage that it minimizes parallel communication, which is a dominant 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](#)

Table of Contents

Author: John W. Grove

Email: jgrove@lanl.gov

[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 Hugoniot \(1\)](#)

[Steady State Self-Similar Flow Hugoniot \(2\)](#)

[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:

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

is unique, then the solution satisfies:

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

and that \mathbf{w} is a solution to the conservation law:

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

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 than 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 $\mathbf{w} = \mathbf{u} - \mathbf{x}/t = \mathbf{u} - \boldsymbol{\xi}$. It is left as an exercise to show that the conservative scale invariant form of the Euler equations is:

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

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

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

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} , $\mathbf{w}(\boldsymbol{\xi}) = \mathbf{g}(\boldsymbol{\xi} - \mathbf{v})$ and $\mathbf{g}(\boldsymbol{\eta})$ is homogeneous of degree 0, $\mathbf{g}(\alpha\boldsymbol{\eta}) = \mathbf{g}(\boldsymbol{\eta})$, $\forall \alpha > 0$. 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:

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

$$(\rho u^2 + P)_x + (\rho uv)_y = 0$$

$$(\rho uv)_x + (\rho v^2 + P)_y = 0$$

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

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:

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

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

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

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

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

Steady State Self-Similar Flow Hugoniot (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:

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

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:

$$\begin{aligned}\tan \theta &= \frac{\Delta P}{\rho_0 q_0^2 - \Delta P} \cot \beta_0 \\ \frac{1}{2} q_0^2 + h_0 &= \frac{1}{2} q_1^2 + h_1, \quad h_1 - h_0 = \frac{V_0 + V_1}{2} (P_1 - P_0).\end{aligned}$$

Steady State Self-Similar Flow Hugoniot (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.

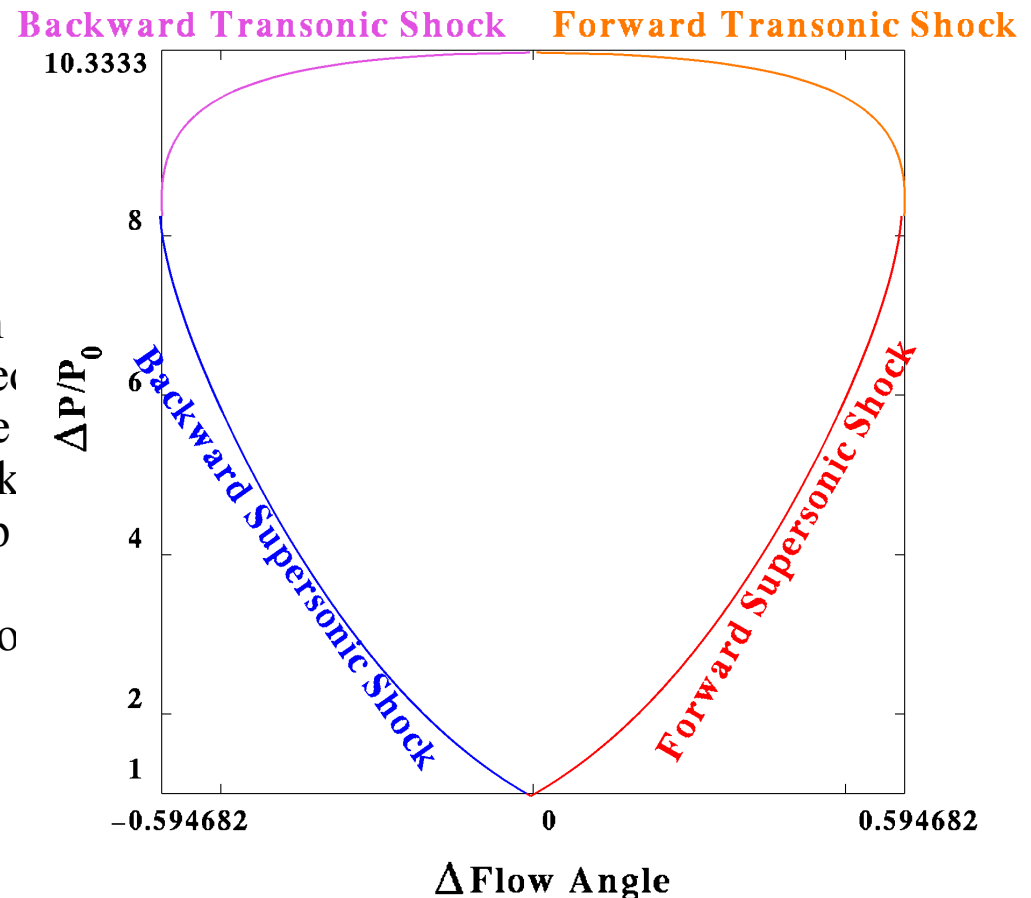
For a perfect gas we can write the turning angle explicitly in terms of the flow states:

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

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

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 fixed 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 of 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:

$$\begin{aligned} \left\{ \cos \varphi \frac{\partial}{\partial r} + \frac{\sin \varphi}{r} \frac{\partial}{\partial \phi} \right\} \theta + \frac{1}{\rho q^2} \left\{ -\sin \varphi \frac{\partial}{\partial r} + \frac{\cos \varphi}{r} \frac{\partial}{\partial \phi} \right\} P &= 0 \\ \left\{ \cos \varphi \frac{\partial}{\partial r} + \frac{\sin \varphi}{r} \frac{\partial}{\partial \phi} \right\} P + \frac{\rho q^2 c^2}{q^2 - c^2} \left\{ -\sin \varphi \frac{\partial}{\partial r} + \frac{\cos \varphi}{r} \frac{\partial}{\partial \phi} \right\} \theta &= 0 \\ \left\{ \cos \varphi \frac{\partial}{\partial r} + \frac{\sin \varphi}{r} \frac{\partial}{\partial \phi} \right\} \left[\frac{1}{2} q^2 + h \right] &= 0 \\ \left\{ \cos \varphi \frac{\partial}{\partial r} + \frac{\sin \varphi}{r} \frac{\partial}{\partial \phi} \right\} S &= 0. \end{aligned}$$

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:

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

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

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

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:

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

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

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

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

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, then 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:

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

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:

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

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

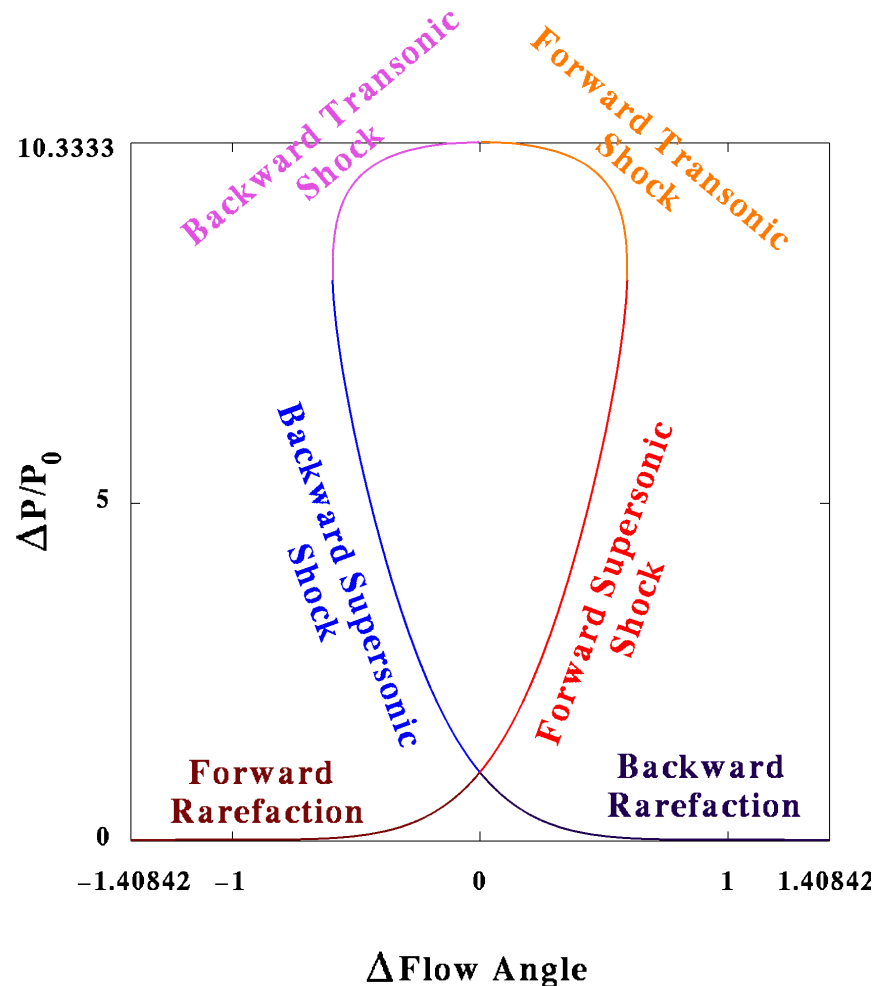
$$S = S_0$$

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

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.

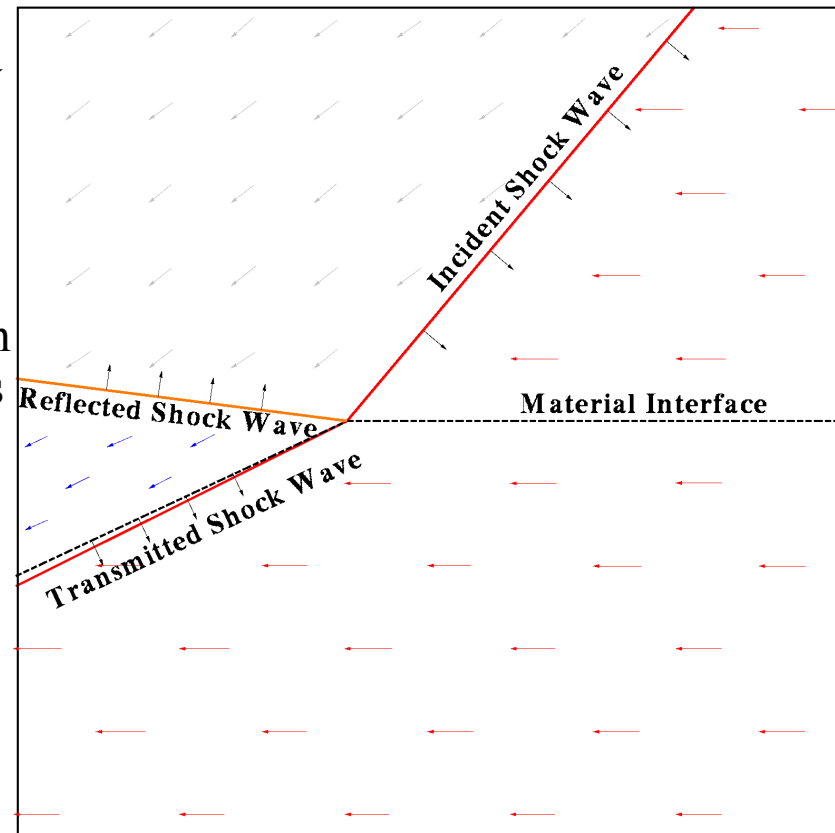
The Steady State Wave Curve

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.



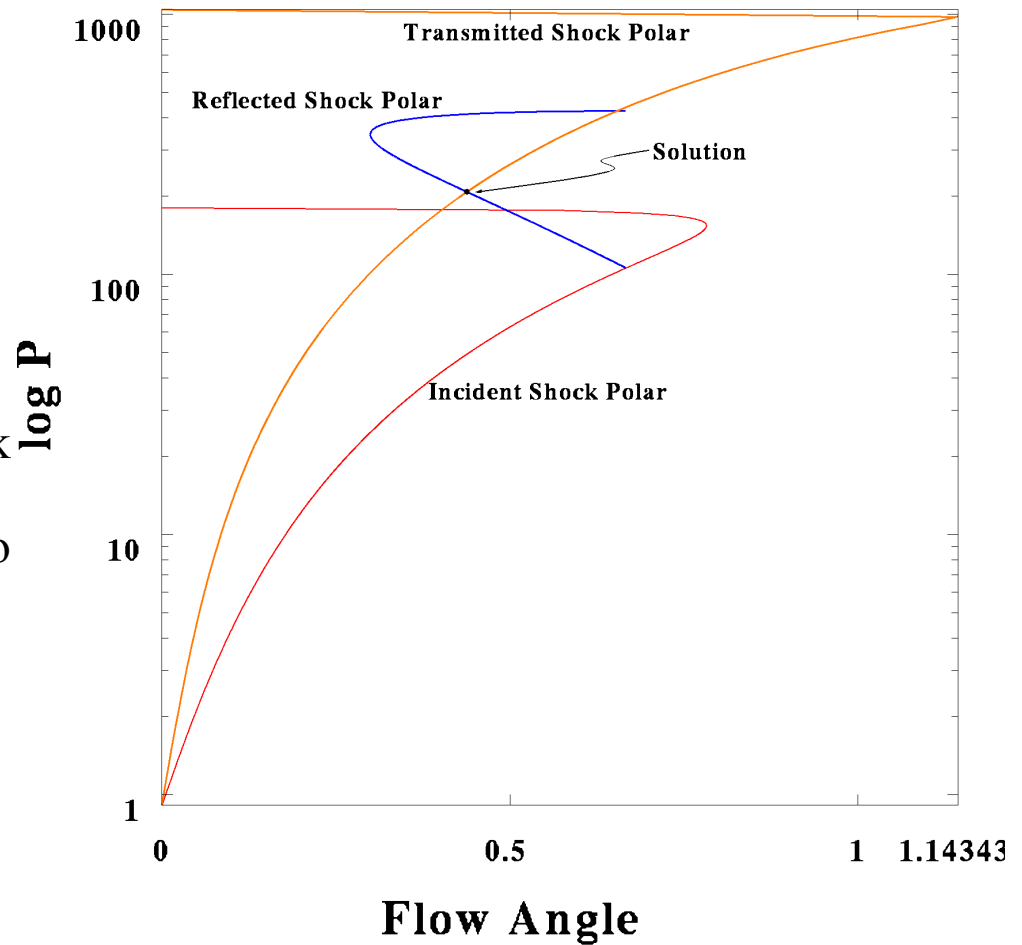
Shock Refractions

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-hexafluoride. 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:

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

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](#)

Table of Contents

Author: John W. Grove

Email: jgrove@lanl.gov

[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](#)

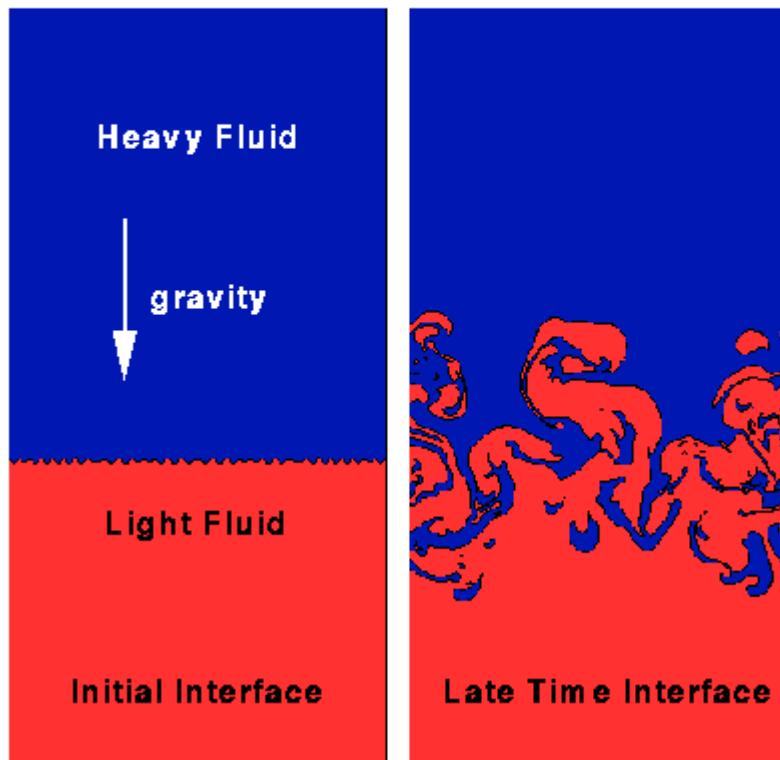
[Further Reading](#)

Numerical Implementation of Front Tracking Wave Interactions

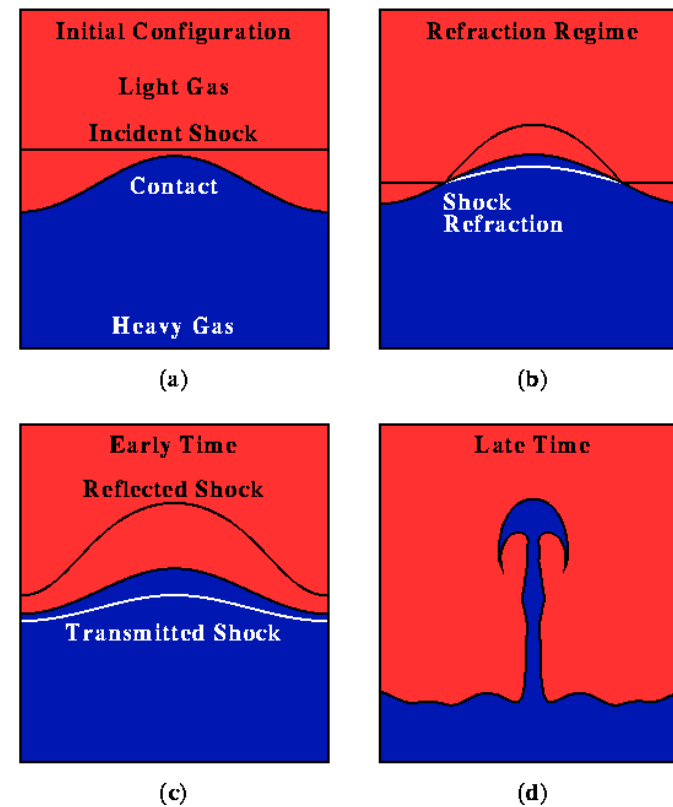
Front tracking provides a method of achieving enhanced resolution of selected strong waves. It eliminates numerical diffusion and provides a mechanism for applying wave propagation algorithms in coordinate systems that are naturally aligned with the fronts, and has the ability to apply special physics 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



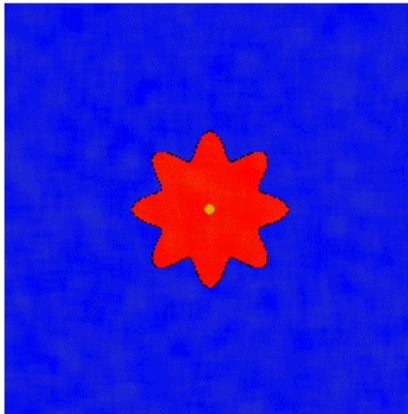
Richtmyer-Meshkov Instability (reflected shock case)



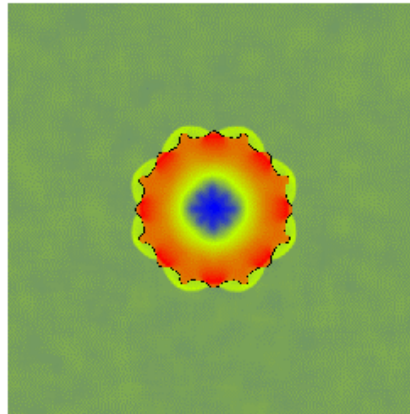
Examples of Wave Interactions

Supernova Explosion

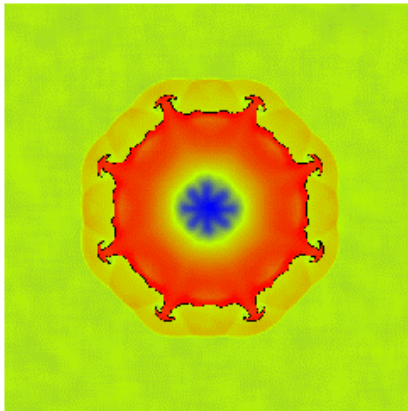
$t = 0$



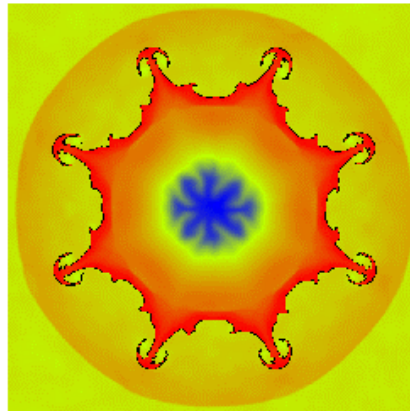
$t = 16$ ns



$t = 30$ ns

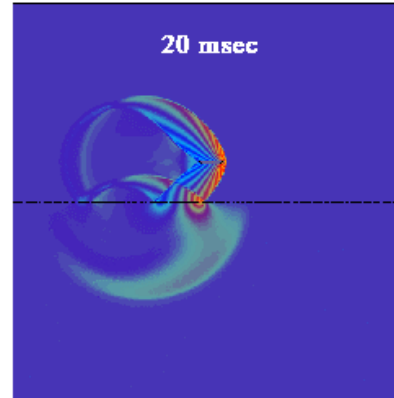


$t = 52$ ns

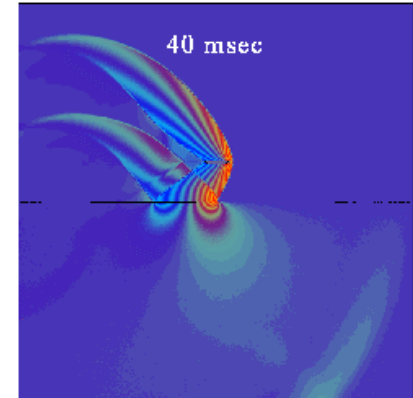


Supersonic Missile Above Water

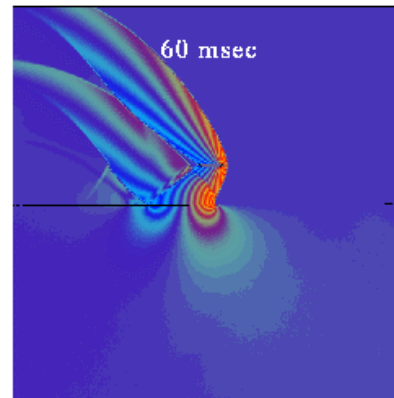
20 msec



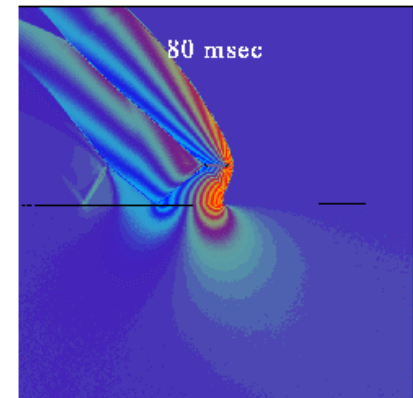
40 msec



60 msec



80 msec

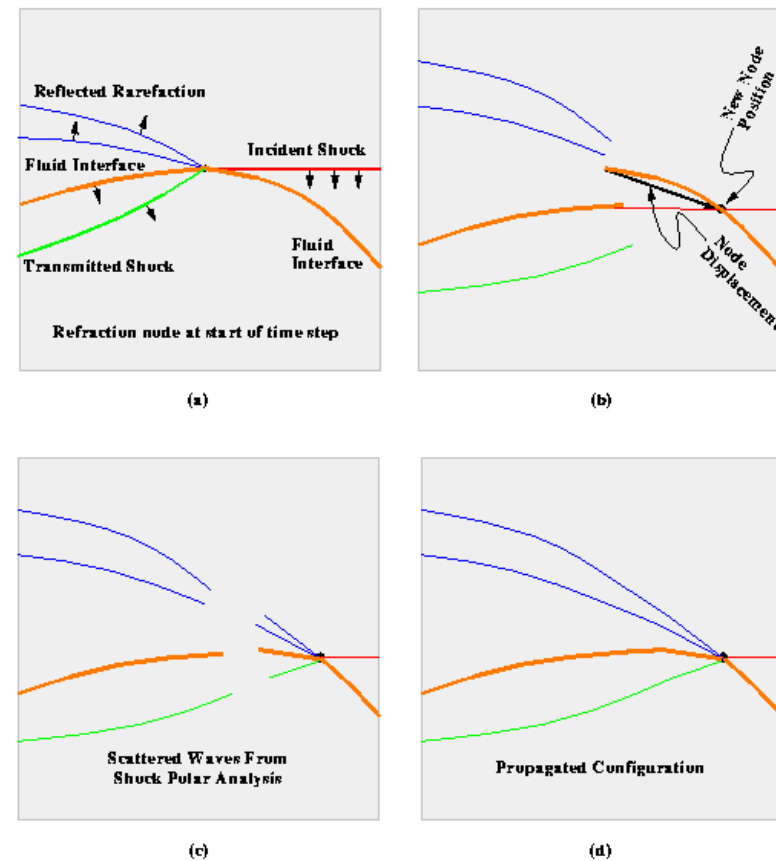


Classification of Numerical Wave Interactions

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.

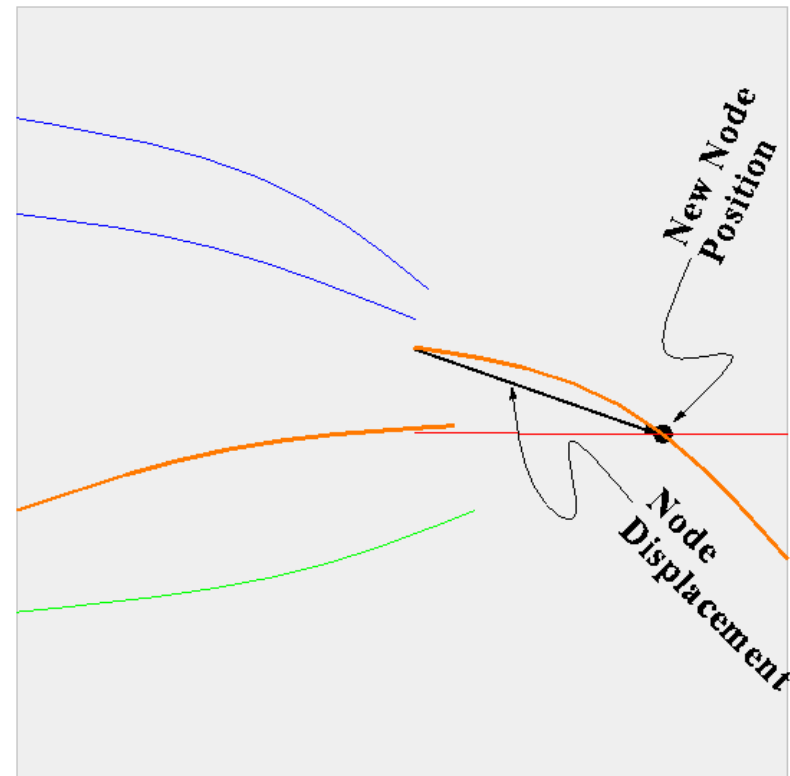
Propagation of a Regular Shock Refraction Node (1)

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)

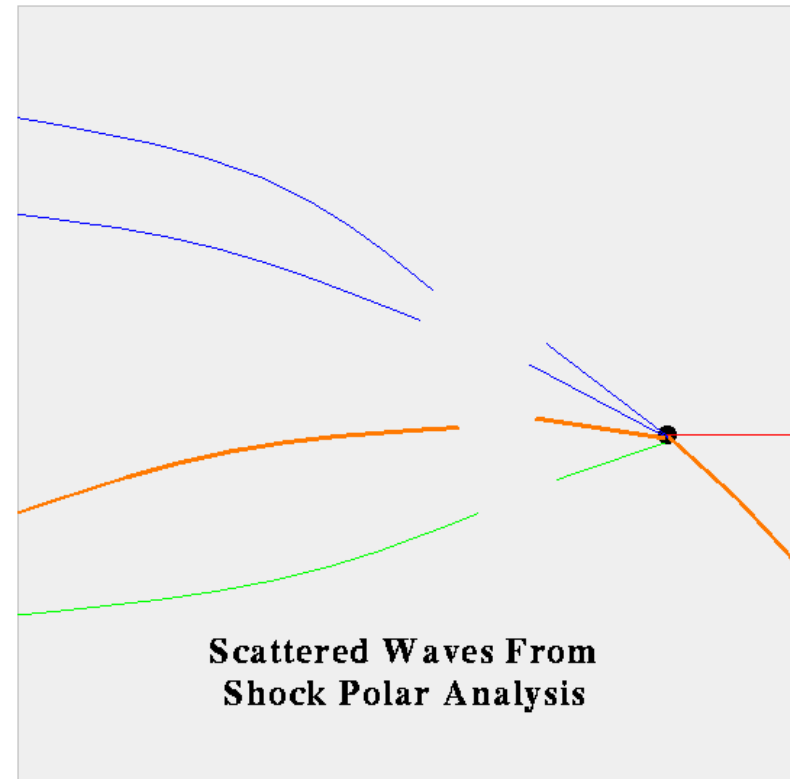
Once we have propagated the individual wave fronts at the node separately leading to the configuration in (b), we use the fact that a regular refraction node is locally a supersonic elementary wave where information flows from the upstream side into the downstream region. We identify the upstream incident 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 frame 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.

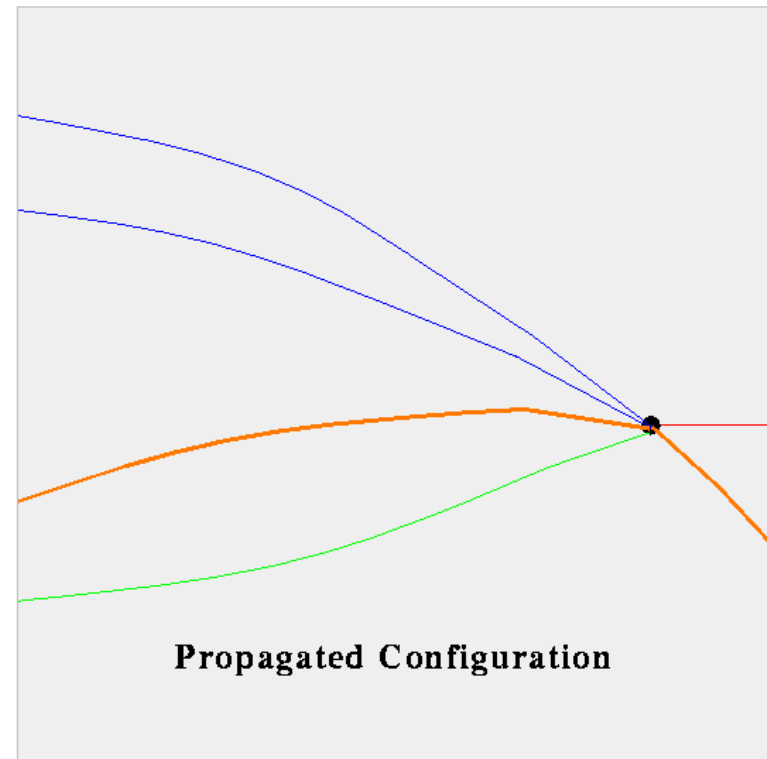


(c)

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.

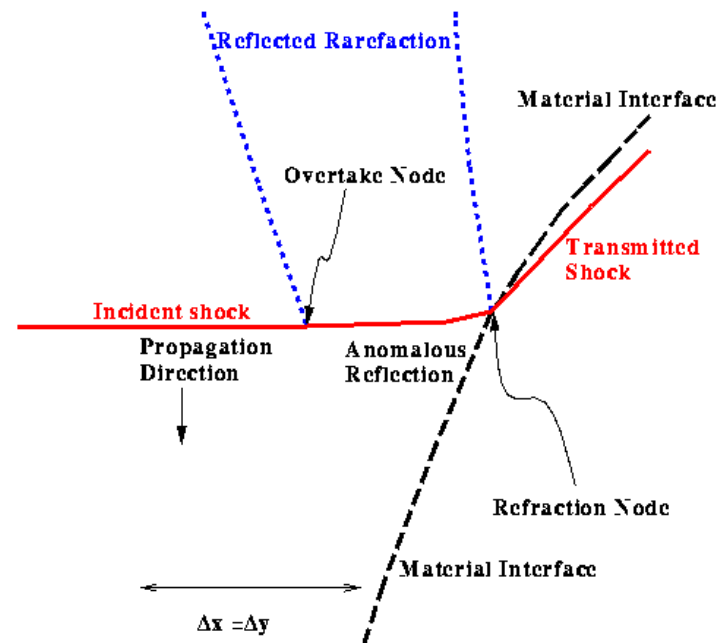
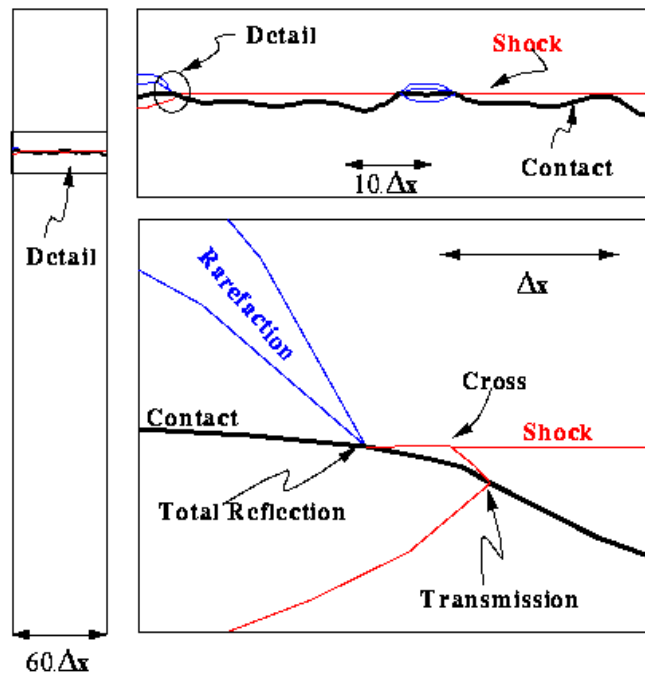


(d)

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.

Full Computation



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:

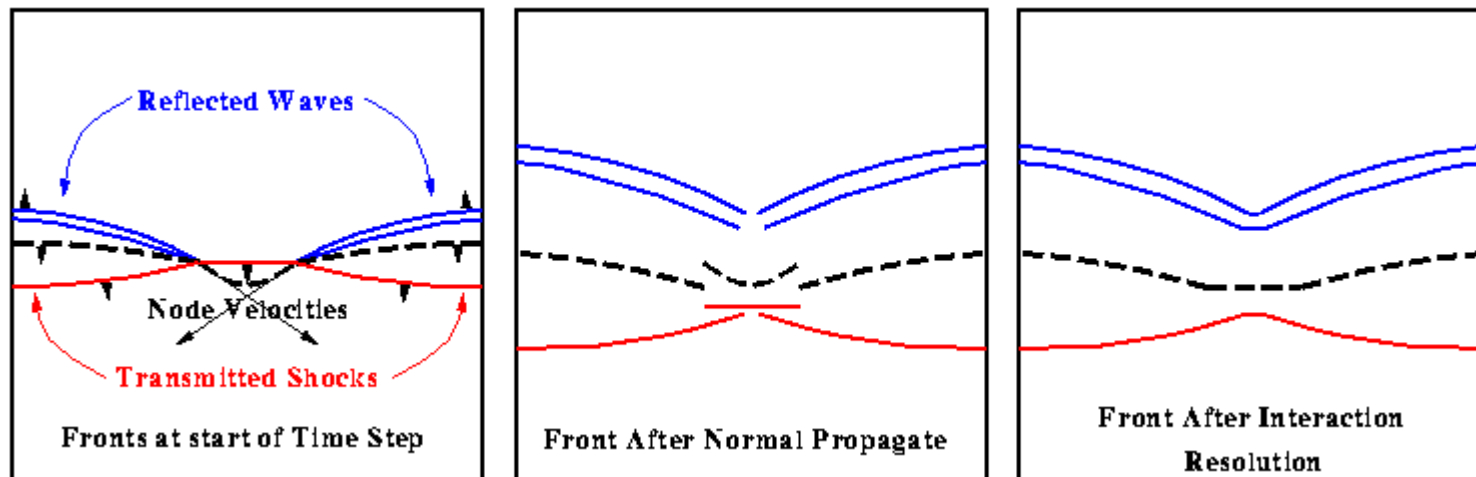
- Detect that an interaction has occurred
- Apply a sorting routine to identify the type of interaction based on the number and types of the waves involved.
- Use an interaction specific algorithm to compute the scattered waves from the interaction and install a representation of the solution into the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

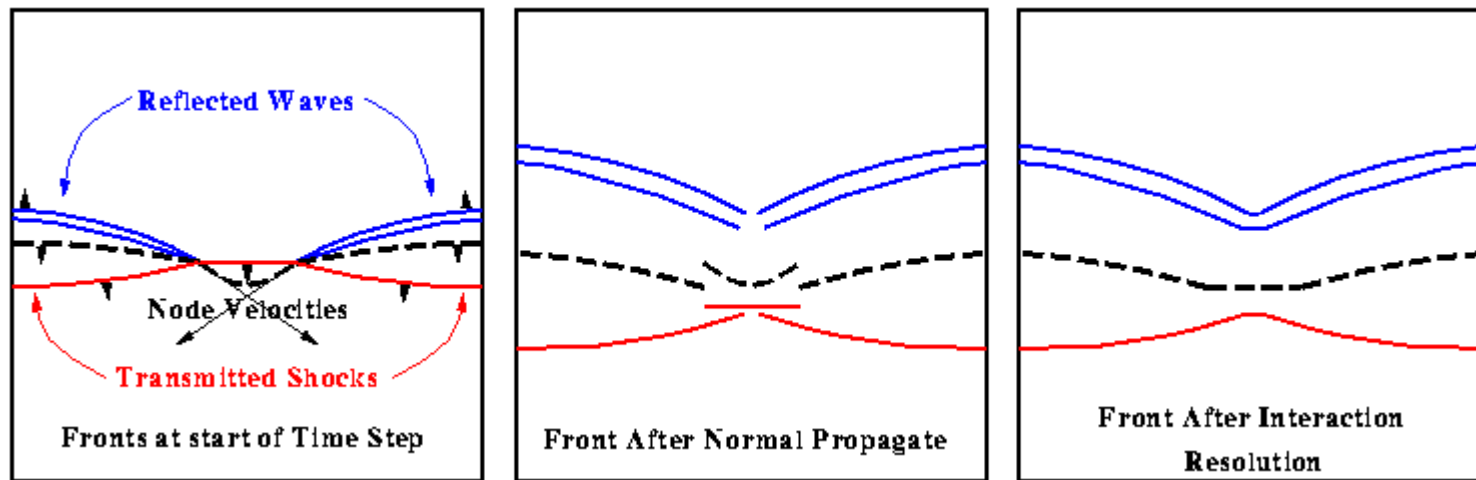
Node-Node Interactions (1)

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.



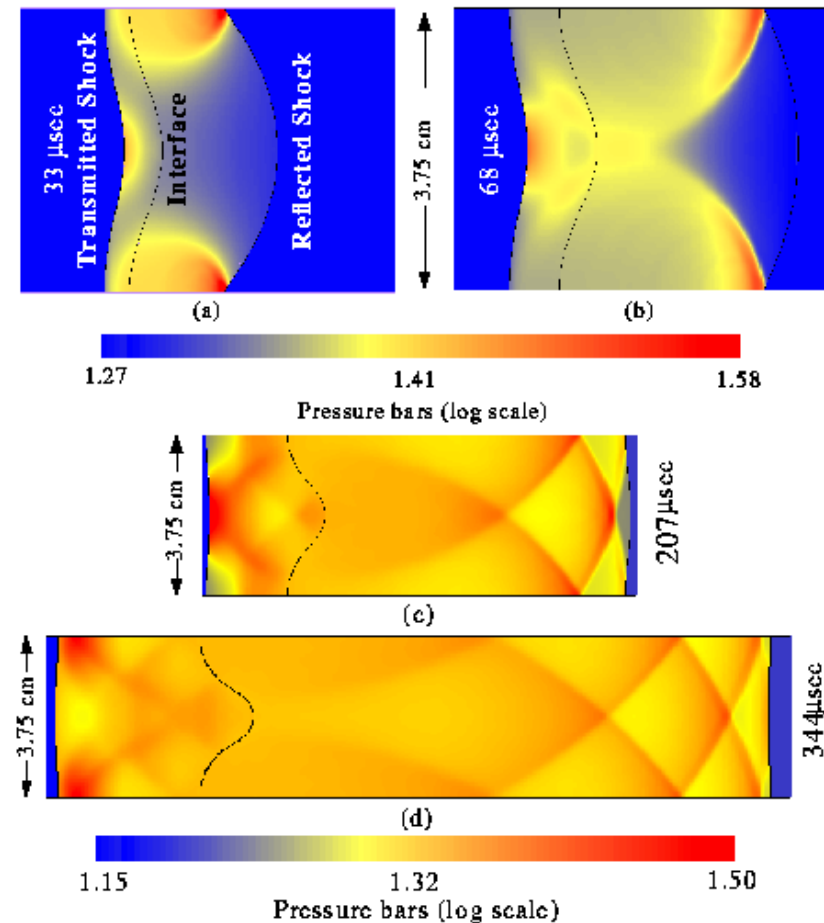
Node-Node Interactions (2)

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.



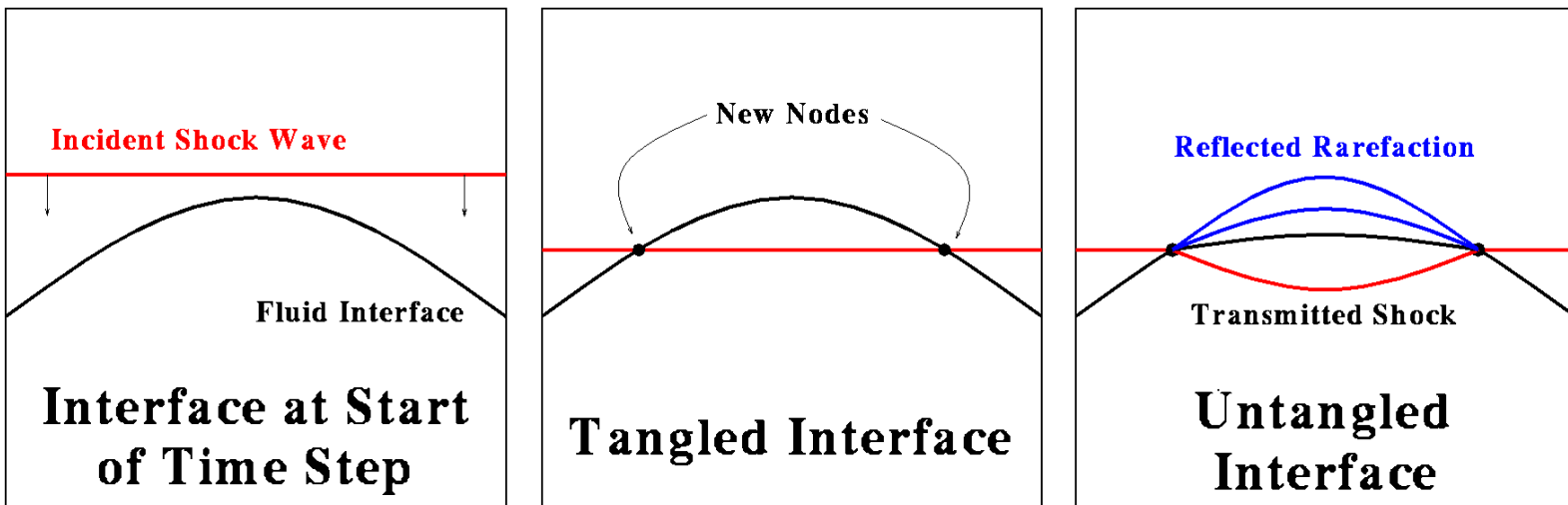
Node-Node Interactions (3)

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.



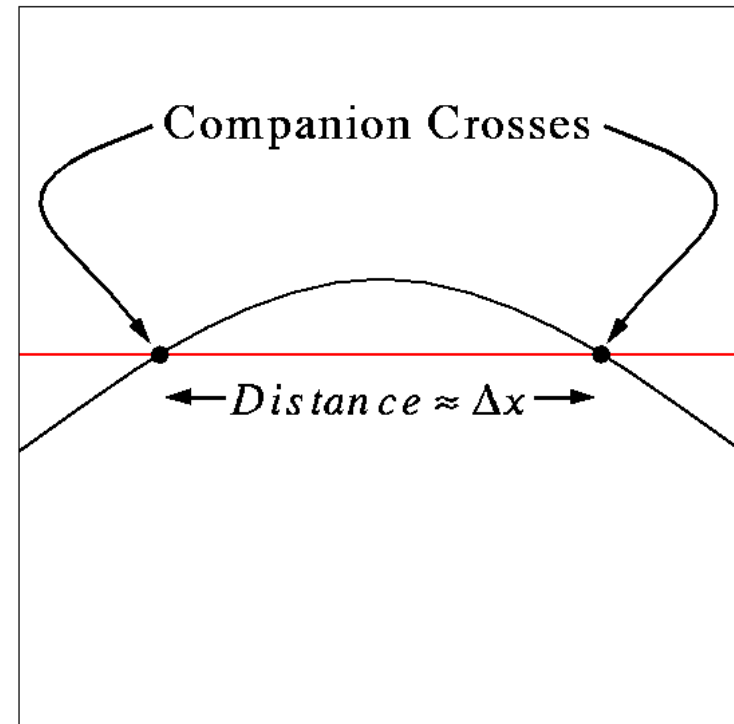
Scalar-Vector Wave Collisions (1)

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 .

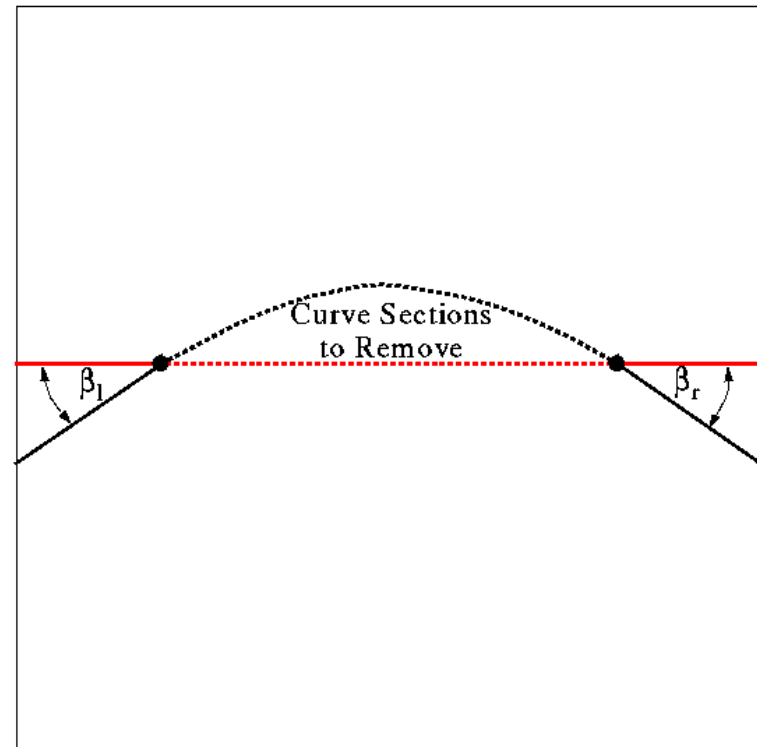


Scalar-Vector Wave Collisions (3)

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 determine 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:

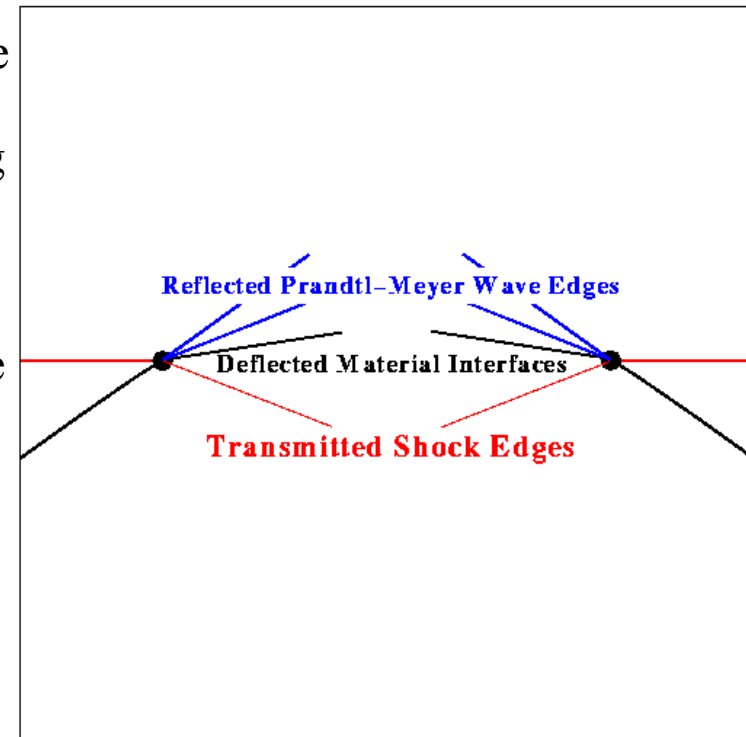
$$\rho_0 q_0 \sin \beta = m$$

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.



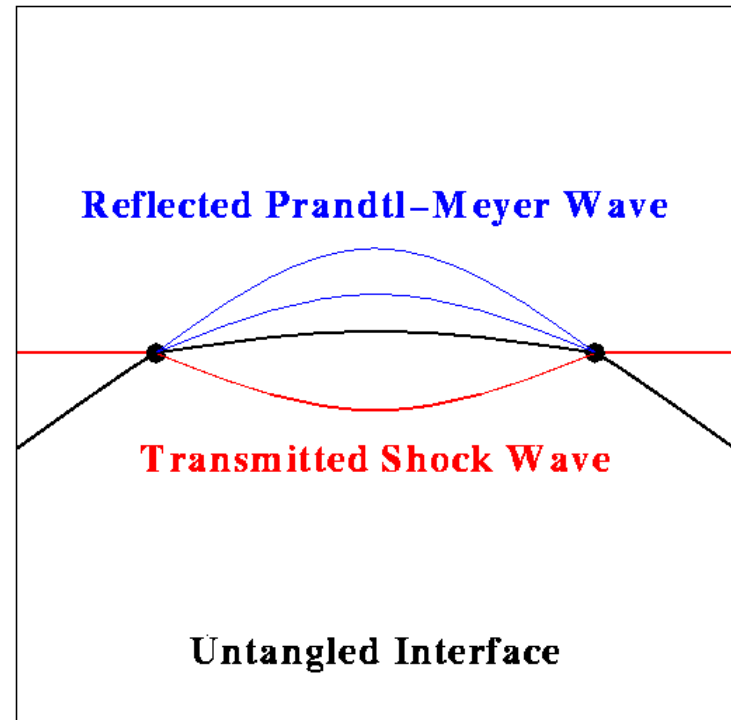
Scalar-Vector Wave Collisions (4)

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:

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

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.