# Wave-Propagation Methods and Software for Complex Applications[1]

**Randall J. LeVeque*** — **Sorin Mitran***

**Applied Mathematics Department,*
*University of Washington,*
*Box 352420,*
*Seattle, WA 98195-2420*

ABSTRACT. *Wave-propagation methods are high-resolution Godunov methods that are written in a general framework that can easily be applied to a wide variety of hyperbolic equations, whether or not they are in conservation form. These methods form the basis for the* CLAWPACK *software, which only requires the user to provide a Riemann solver for the equations to be solved. Recently a new version of this software has been developed with new capabilities. This* BEARCLAW *software is briefly reviewed along with some recent developments in wave-propagation methods. A sample calculation is presented showing an adaptively refined solution on a moving grid for a gas dynamics problem in a tube with a flexible elastic boundary.*

KEYWORDS: *finite volume methods, software, adaptive mesh refinement, high-resolution wave-propagation methods, moving grids, shock capturing.*

## 1. Introduction

Hyperbolic systems of PDEs arise in a broad range of applications areas when modeling wave motion or advective transport. Often, though not always, these equations are in conservation form

$$q_t + f(q)_x = 0 \tag{1}$$

(in one dimension) where $q(x,t) \in \mathbb{R}^m$ is the vector of conserved quantities and $f$ is the flux function. The problem is hyperbolic if the Jacobian matrix $A = f'(q)$ is diagonalizable with real eigenvalues and a complete set of eigenvectors. Finite volume methods based on the integral conservation law are popular since they properly handle shock waves in nonlinear problems. A

1. To appear in *Proc. Third International Symposium on Finite Volumes for Complex Applications*, Porquerolles, France, June, 2002.

robust and powerful class of high-resolution finite volume methods can be written in a general form that is easily applied to most hyperbolic equations, in either conservative or nonconservative formulations. These **wave-propagation algorithms** are second-order Godunov-type methods based on the following steps:

1) The Riemann problem is solved at each cell interface, using the cell average in the cells to either side as data. This results in a set of waves traveling at finite speeds.

2) These waves are used to update the cell averages to either side (Godunov's method).

3) Limiters are applied to the waves, and the limited waves are used to apply "second-order" correction terms to each cell average.

The resulting method is second-order accurate for smooth solutions and the limiters help reduce numerical dispersion and spurious oscillations. Multidimensional versions of this algorithm can also be developed in which the Riemann problems are solved normal to each cell interface and then "transverse Riemann solvers" are also defined to provide the appropriate cross-derivative terms needed to achieve second-order accuracy and good stability properties in multidimensions. These algorithms are described in detail in [LAN 00], [LEV 97], and the recent textbook [LEV 02a].

Applying these methods to a different hyperbolic system requires only a change in the Riemann solver, and so they are well suited to the development of general purpose software for solving hyperbolic problems. These methods form the basis for the CLAWPACK software (conservation laws package), which is freely available on the web at

`http://www.amath.washington.edu/~claw/`.

Numerous sample computations can be viewed from this webpage and the corresponding computer code downloaded. The basic CLAWPACK code includes software in 1, 2, and 3 space dimensions, along with the AMRCLAW software for adaptive mesh refinement in 2 and 3 dimensions, as developed by Berger and LeVeque [BER 98].

Recently a new version of this software has been developed as a supplement to the previous offerings. In this paper we briefly survey some of the capabilities of this software and some new variants of these finite volume methods that are useful for certain complex applications. In particular, an approach to handling spatially-varying flux functions and a moving grid version of the code are summarized. A sample calculation on a coupled fluid-elastic problem in a deforming region using adaptive refinement is shown as one example.

This new BEARCLAW software provides a more flexible environment for applying finite volume methods to large-scale practical problems. BEARCLAW stands for Boundary Embedded Adaptive Refinement for Conservation Laws, which reflects one capability that is still under development — the ability to embed complex geometries in Cartesian grids by the use of special formulas in the "cut cells" that are reduced in size by the boundary passing through. Finite volume methods of this type are becoming increasingly popular for problems where body-fitted grids are difficult to generate. We are currently investigating methods of the type described in the papers [BER 90a], [BER 90b], [LEV 01], [FOR 98], for example, and hope to eventually have a general implementation that is easy to apply to a broad class of hyperbolic problems in complex geometry. See the contribution of Berger and Helzel [BER 02] to these proceedings for some recent progress in this direction.

## 2. Wave-propagation algorithms

The CLAWPACK software requires the user to provide a Riemann solver subroutine, the means by which the particular problem being solved is specified. In one dimension this routine takes the set of cell averages $Q_i$ at the start of a time step and returns a set of waves $\mathcal{W}^p_{i-1/2}$ and speeds $s^p_{i-1/2}$ for $p = 1,\ 2,\ \dots,\ M_w$, where $M_w$ is the number of waves in the (approximate) Riemann solution. Often $M_w = m$ for a system of $m$ equations, but this is not required. Exact and approximate Riemann solvers for several applications are provided with the software. In addition to the waves and speeds, the Riemann solver must also return two $m$-vector quantities denoted $\mathcal{A}^-\Delta Q_{i-1/2}$ and $\mathcal{A}^+\Delta Q_{i-1/2}$ at each cell interface. These are the left-going and right-going "fluctuations", the Godunov update that should be made to the cell averages $Q_{i-1}$ and $Q_i$ respectively as a result of the waves emanating from the Riemann problem at $x_{i-1/2}$. Often these are related to $\mathcal{W}^p_{i-1/2}$ and $s^p_{i-1/2}$ by

$$\begin{aligned}
\mathcal{A}^-\Delta Q_{i-1/2} &= \sum_{p:s^p_{i-1/2}<0} s^p_{i-1/2}\mathcal{W}^p_{i-1/2} \equiv \sum_{p=1}^{M_w} (s^p_{i-1/2})^-\mathcal{W}^p_{i-1/2},\\
\mathcal{A}^+\Delta Q_{i-1/2} &= \sum_{p:s^p_{i-1/2}>0} s^p_{i-1/2}\mathcal{W}^p_{i-1/2} \equiv \sum_{p=1}^{M_w} (s^p_{i-1/2})^-\mathcal{W}^p_{i-1/2},
\end{aligned} \tag{2}$$

where $s^- = \min(s,0)$ and $s^+ = \max(s,0)$. The first-order Godunov method then takes the form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}[\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2}], \tag{3}$$

This method will be conservative provided that

$$\mathcal{A}^-\Delta Q_{i-1/2} + \mathcal{A}^+\Delta Q_{i-1/2} = f(Q_i) - f(Q_{i-1}) \tag{4}$$

for each $i$, in which case (3) can be rewritten in the more standard flux differencing form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}[F_{i+1/2} - F_{i-1/2}], \tag{5}$$

with

$$F_{i-1/2} = f(Q_{i-1}) - \mathcal{A}^-\Delta Q_{i-1/2} = f(Q_i) - \mathcal{A}^+\Delta Q_{i-1/2}. \qquad (6)$$

The advantage of using (3) over (5) is that it allows the application of these methods to hyperbolic problems that are not in conservation form, in which case (5) does not make sense. An example is linear acoustics in a heterogeneous medium, as discussed in [FOG 99], [LEV 97], [LEV 02a].

The quantities $\mathcal{W}^p_{i-1/2}$ and $s^p_{i-1/2}$ are used in order to apply high-resolution corrections to Godunov's method. The full method takes the form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}[\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2}] - \frac{\Delta t}{\Delta x}[\tilde{F}_{i+1/2} - \tilde{F}_{i-1/2}], \quad (7)$$

where

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{m} |s^p_{i-1/2}| \left(1 - \frac{\Delta t}{\Delta x}|s^p_{i-1/2}|\right) \widetilde{\mathcal{W}}^p_{i-1/2}. \qquad (8)$$

If $\widetilde{\mathcal{W}}^p_{i-1/2} \equiv \mathcal{W}^p_{i-1/2}$ then this gives a formally second-order accurate method for certain hyperbolic systems (e.g. for constant coefficient linear systems, in which case it reduces to the Lax-Wendroff method). In practice limiters are generally applied, and

$$\widetilde{\mathcal{W}}^p_{i-1/2} = \phi(\mathcal{W}^p_{I-1/2}, \mathcal{W}^p_{i-1/2})\mathcal{W}^p_{i-1/2}$$

where

$$I = \begin{cases} i-1 & \text{if } s^p_{i-1/2} > 0 \\ i+1 & \text{if } s^p_{i-1/2} < 0. \end{cases}$$

and $\phi : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a scalar-valued function. Standard limiters such as minmod, superbee or the MC limiter (see [LEV 97])can be applied.

The multidimensional software requires two Riemann solvers, one that solves the Riemann problem normal to each interface between grid cells (analogous to what was just described in one dimension), and "transverse Riemann solver" that takes the resulting fluctuations $\mathcal{A}^-\Delta Q_{i-1/2}$ and $\mathcal{A}^+\Delta Q_{i-1/2}$ and splits them into waves propagating in directions tangent to the interface. Details can be found in [LEV 97] in two dimensions and [LAN 00] in three dimensions. Alternatively, dimensional splitting is allowed as an option in the software, in which case only the normal Riemann solver is required.

An approximate Riemann solver is often based on choosing some set of $m$ linearly independent basis vectors $r^p_{i-1/2}$ $(p = 1,\ 2,\ ,\ldots,\ m)$ at each interface, decomposing the jumps in $Q$ across the interfaces as

$$Q_i - Q_{i-1} = \sum_{p=1}^{m} \alpha^p_{i-1/2} r^p_{i-1/2}$$

and using this to define the waves $\mathcal{W}^p_{i-1/2} \equiv \alpha^p_{i-1/2} r_{i-1/2}$. The vectors $r^p_{i-1/2}$ are typically the eigenvectors of some approximate Jacobian matrix, e.g., the Roe matrix $\hat{A}_{i-1/2}$ based on the data $Q_{i-1}$ and $Q_i$. The speeds $s^p_{i-1/2}$ are then given by the corresponding eigenvalues.

A variant of this algorithm has recently been proposed in [BAL 01] that has the advantage of extending naturally to problems of the form

$$q_t + f(q,x)_x = 0 \tag{9}$$

where the flux function $f(q,x)$ has an explicit dependence on $x$. Spatially-varying fluxes arise in many applications, e.g. nonlinear wave propagation in heterogeneous media. Often a set of basis vectors $r^p_{i-1/2}$ and speeds $s^p_{i-1/2}$ can be determined in some reasonable manner at the interface $x_{i-1/2}$ based on the medium to either side (see [LEV 02b] for an example in nonlinear elasticity). Then our variant consists of decomposing the flux difference into waves, rather than decomposing the $Q$ difference. If $f_i(q)$ denotes the flux function in the $i$th cell, then we can find scalars $\beta^p_{i-1/2}$ such that

$$f_i(Q_i) - f_{i-1}(Q_{i-1}) = \sum_{p=1}^{m} \beta^p_{i-1/2} r^p_{i-1/2} \equiv \sum_{p=1}^{m} \mathcal{Z}^p_{i-1/2}, \tag{10}$$

The f-waves $\mathcal{Z}^p_{i-1/2}$ can be used to define

$$\begin{aligned} \mathcal{A}^-\Delta Q_{i-1/2} &= \sum_{p:s^p_{i-1/2}<0} \mathcal{Z}^p_{i-1/2}, \\ \mathcal{A}^+\Delta Q_{i-1/2} &= \sum_{p:s^p_{i-1/2}>0} \mathcal{Z}^p_{i-1/2}, \end{aligned} \tag{11}$$

and corrections

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{M_w} \text{sgn}(s^p_{i-1/2}) \left(1 - \frac{\Delta t}{\Delta x}|s^p_{i-1/2}|\right) \widetilde{\mathcal{Z}}^p_{i-1/2}. \tag{12}$$

where the usual limiters are applied to $\mathcal{Z}^p$ to obtain the $\widetilde{\mathcal{Z}}^p$. For a constant coefficient linear problem this method is identical to the original method. For nonlinear problems it can be shown to be formally second-order accurate quite generally, at least when no limiters are used (see [BAL 01]).

This variant is also useful in many problems with source terms. The equation

$$q_t + f(q,x)_x = \psi(q,x) \tag{13}$$

can be discretized by decomposing

$$f_i(Q_i) - f_{i-1}(Q_{i-1}) - \Delta x \Psi_{i-1/2} = \sum_{p=1}^{m} \beta^p_{i-1/2} r^p_{i-1/2} \equiv \sum_{p=1}^{m} \mathcal{Z}^p_{i-1/2}. \tag{14}$$

where $\Psi_{i-1/2}$ is the source term at $x_{i-1/2}$. This has the advantage that steady state solutions with $f(q,x)_x = \psi(q,x)$ can be approximated very well numerically. If

$$\frac{f_i(Q_i) - f_{i-1}(Q_{i-1})}{\Delta x} = \psi_{i-1/2}$$

then the left hand side of (14) vanishes, all coefficients $\beta^p_{i-1/2}$ will be zero, and hence the solution remains unchanged. Even more importantly, for time dependent problems that are very close to steady state, it is only the deviation from steady state that is decomposed into waves. Since both the Godunov and high-resolution updates are based on these waves, very small perturbations from steady state can be accurately modeled. This is in sharp contrast to the use of fractional step methods, for example, in which a high-resolution method is used to advance $q_t + f_x = 0$ over a time step and then an ODE solver is used to advance $q_t = \psi$. The decoupled equations may each give rise to large changes in the solution that in principle should nearly cancel out but in practice will lead to a severe loss of accuracy in small amplitude waves. See [BAL 01] for more discussion and [LEV 98] for an earlier approach with similar properties.

Another useful variant of the wave-propagation algorithm is the "capacity form" differencing algorithm introduced in [LEV 97] for an equation of the form

$$\kappa(x) q_t + f(q)_x = 0 \tag{15}$$

(and generalizations to nonconservative form and multidimensions). A capacity function $\kappa(x)$ appears in many applications, e.g., porosity in porous media flow or cross-sectional area in quasi-one-dimensional flow. The capacity can also be used for the Jacobian of the grid mapping function when hyperbolic equations are solved on nonuniform grids, by reformulating the equations in the form (15) on a uniform computational grid. Capacity-form differencing in one dimension takes the form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\kappa_i \Delta x}[\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}] - \frac{\Delta t}{\Delta x}[\tilde{F}_{i+1/2} - \tilde{F}_{i-1/2}], \tag{16}$$

where we now define the correction fluxes $\tilde{F}_{i-1/2}$ by

$$\tilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{M_w}\left(1 - \frac{\Delta t}{\kappa_{i-1/2}\Delta x}|s^p_{i-1/2}|\right)|s^p_{i-1/2}|\,\widetilde{\mathcal{W}}^p_{i-1/2}. \tag{17}$$

The advantages of this formulation of the algorithm, as opposed to incorporating $\kappa(x)$ into the flux function, for example, are discussed in [LEV 97], [LEV 02a].

## 3. The BEARCLAW code

The basic finite volume algorithms described above are implemented in CLAWPACK and can be applied to general hyperbolic problems on a uniform grid using the basic CLAWPACK routines. In 2 and 3 dimensions the AMRCLAW software can be used to automatically apply adaptive mesh refinement with very little change to the user's code. An MPI version of the uniform grid code is also available in CLAWPACK, allowing the solution of larger problems on parallel machines, or clusters of workstations, by splitting the domain into an array of smaller subdomains. The BEARCLAW code offers some additional capabilities and in this section we summarize the main enhancements of this code.

The original CLAWPACK code is written in Fortran 77. AMRCLAW is also written in f77, and grew out of the adaptive refinement codes of Marsha Berger for the Euler equations, which have been evolving since the early 1980's; see [BER 84], [BER 89]. These were adapted to implement the more general CLAWPACK algorithms as described in [BER 98]. While the code is quite robust and well tested by now, it is a difficult code to understand or modify due to its hybrid history, the use of different data structures and notation in different parts of the code, and the lack of memory management features, pointers, and recursion in f77.

The new BEARCLAW code is written in Fortran 90 and was designed from scratch to incorporate the best features of AMRCLAW. The use of f90 and the fresh start have allowed the development of a cleaner code with more capabilities and the possibility of easier future extension.

The AMR aspect largely follows the design of AMRCLAW, with refinement of the domain on rectangular subregions that are determined by first performing error estimation to flag cells needing refinement, and then clustering the flagged cells into rectangular patches using the algorithm of Berger and Rigoutsos [BER 91]. One significant difference is that BEARCLAW uses a tree structure for refinement. Each patch at Level $L$ is a subregion of a single grid at Level $L-1$, its parent grid. By contrast, in AMRCLAW a grid at Level $L$ may overlap two or more grids at Level $L-1$. The tree-structured approach has the disadvantage of producing more grids and consequently more overhead in passing information between grids (which is accomplished using a layer of ghost cells around each grid, as in AMRCLAW; see [BER 98]).

On the other hand, there are several advantages to using a tree structure. General algorithms that must be applied over all grids are often easily defined recursively with this structure. In addition to the wave-propagation algorithms built into the codes, the user may want to apply other algorithms. This can often be done easily using the tree-traversal routines included in BEARCLAW, whereas modifying AMRCLAW to implement a new algorithm is potentially more difficult. In particular, implicit routines are often needed to implement parabolic terms (e.g., diffusion or viscosity) in many applications, and general approaches to efficiently doing so on adaptive grids are now being studied in BEARCLAW. Often these equations are coupled with elliptic equations that must be solved in each time step (e.g., in projection methods for incompressible flow or MHD equations), and this must also be done on the adaptive grids.

Another advantage to the tree structure used in BEARCLAW is that multi-physics problems can be solved, in which different equations must be used over different parts of the physical domain. In BEARCLAW, the top level can be a set of grids over these subdomains with a different set of equations posed on each, along with an appropriate specification of boundary condition routines that determine how the equations are coupled together at subdomain interfaces. These subdomains may even have different dimensionalities. An example is shown in Section 5 of a two-dimensional tube bounded by one-dimensional elastic membranes. The Euler equations are solved on a two-dimensional grid that is coupled to two one-dimensional grids on which the elastic membrane equations are solved. With adaptive refinement, these grids form the root nodes for a forest of trees, so that each can be refined independently. In this context it is imperative that refined patches not overlap multiple coarser grids.

Another feature of BEARCLAW is the ability to use MPI for parallel computing, which is not available in AMRCLAW. Each grid resides on a single processor, and a knapsack algorithm is used to distribute grids among processors in order to maintain good load balancing.

## 4. A moving grid algorithm

Recently we have developed a two-dimensional moving grid algorithm that generalizes the one-dimensional algorithm of Fazio and LeVeque [FAZ 02]. This algorithm, described more fully in [MIT 02], is intended primarily for problems in deforming geometry where the mesh must adjust to the physical domain. An example is shown in the next section. The motion of grid nodes (corners of the finite volume cells) is assumed to be constant over each time step. The boundary between two adjacent grid cells is taken to be linear between these moving points and so traces out a ruled surface in space-time. The solution to the Riemann problem between these cells at the initial time leads to waves propagating at constant velocity and hence are planar in physical space-time. In the computational domain, the grid is fixed and uniform while the waves become ruled surfaces. The effect of these waves on the adjacent cell averages is worked out in [MIT 02] and easily implemented using the wave-propagation algorithm with capacity-form differencing.

## 5. Gas dynamics in a flexible tube

As a sample application we consider inviscid compressible gas dynamics in a two-dimensional tube bounded by flexible membranes on either side. Initially the gas is stationary at constant pressure matching the pressure outside the tube (which is assumed to be constant for all time). At time $t = 0$ a jet of gas is turned on in one end of the tube and begins to inflate the tube. The Euler equations of gas dynamics and the elastic wave equation on the membranes are solved simultaneously. Elastic loads on the membranes are given by the local fluid pressure. The fluid motion takes place in the domain delimited by the elastic membranes.

The initial tube radius is $R = 1$. The jet diameter is $r = 0.1$ with velocity $u = 1$. The initial pressure and density in the jet and tube fluid are the same $p = 1$, $\rho = 1$. The elastic wave velocity in the membrane is $c = 0.5$ and the linear mass of the membrane is $\mu = 10$. The large linear mass leads to significant inertia within the membrane and subsequent effect upon the fluid.

The resulting motion of the gas and boundary is shown in Figures 1 and 2. Figures 1 and 2(a) show the initial stages of jet penetration into the tube. The computation is carried out with 3 refinement levels. Mach number contour lines are depicted along with local velocity vectors. Mach contour lines are drawn at 0.1 intervals on all interleaved grids. The extent to which they superimpose is an indication of grid convergence.

At later stages a complicated fluid-structure interaction is observed. The jet flow undergoes a Kelvin-Helmholtz instability which leads to the formation of vortical structures. These impinge upon the membrane from where upstream influences are propagated via elastic waves through the wall. A later stage contour plot of Mach number and velocity is shown in Figure 2(b).

## 6. Acknowledgments.

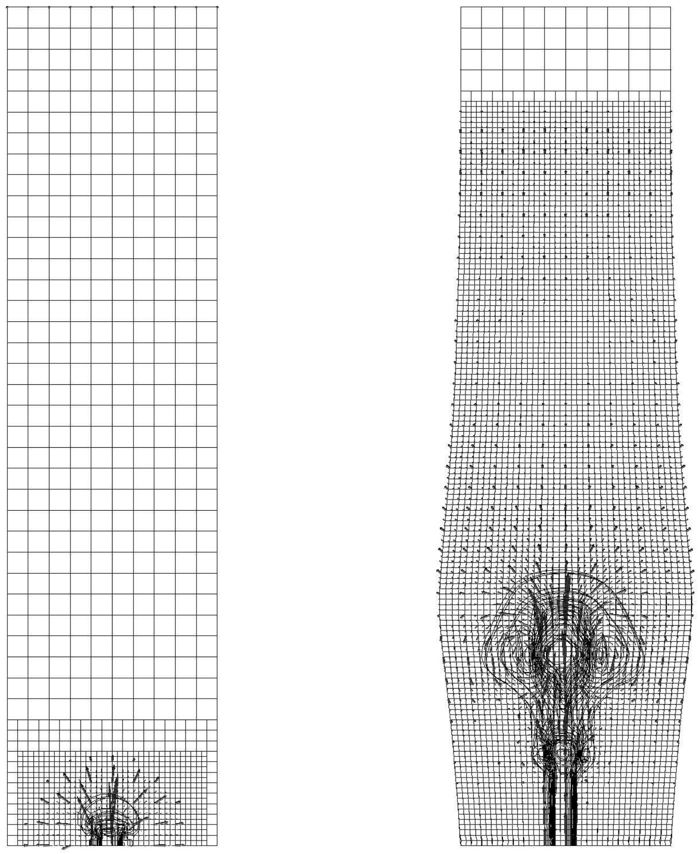
This work was supported in part by DOE grant DE-FG03-96ER25292 and NSF grants DMS-9803442 and DMS-0106511.

## References

[BER 89] BERGER, M.J. COLELLA, P., "Local adaptive mesh refinement for shock hydrodynamics", *J. Comput. Phys.*, 82:64–84, 1989.

[BER 02] BERGER, M.J. HELZEL, C., "Grid aligned h-box methods for conservation laws in complex geometries", In *this proceedings.*

[BER 90a] BERGER, M., LEVEQUE, R.J.. "Cartesian meshes and adaptive mesh refinement for hyperbolic partial differential equations", In B. Engquist and B. Gustafsson, editors, *Proc. Third Int'l Conf. Hyperbolic Problems*, pages 67–73, Lund, Sweden, 1990. Studentlitteratur.



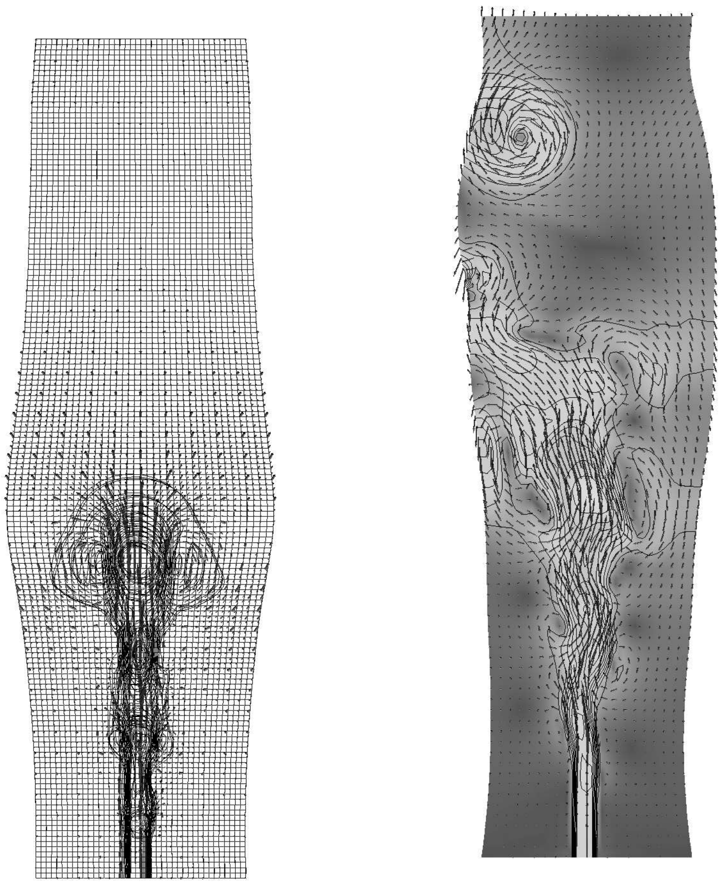
(a) (b)

**Figure 1.** *A jet of gas entering a flexible tube. Contours of Mach number are shown along with velocity vectors and the adaptive grids. (a) At time* $t = 0.5$*. (b) At time* $t = 7.5$*.*

[BER 90b] BERGER, M., LEVEQUE, R.J.. "Stable boundary conditions for Cartesian grid calculations", *Computing Systems in Engineering*, 1:305–311, 1990.

[BER 98] BERGER, M.J., LEVEQUE, R.J.. "Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems", *SIAM J. Numer. Anal.*, 35:2298–2316, 1998.

[BAL 01] BALE, D., LEVEQUE, R.J., MITRAN, S., ROSSMANITH, J., "A wave-propagation method for conservation laws and balance laws



(a) (b)

**Figure 2.** *A jet of gas entering a flexible tube. (c) At time $t = 8.5$. (b) At a later time, $t = 18.5$.*

with spatially varying flux functions", submitted, 2001. (`ftp://amath.washington.edu/pub/rjl/papers/fwave.ps.gz`).

[BER 84] BERGER, M., OLIGER, J., "Adaptive mesh refinement for hyperbolic partial differential equations", *J. Comput. Phys.*, 53:484–512, 1984.

[BER 91] BERGER, M.J., RIGOUTSOS, I., "An algorithm for point clustering and grid generation", *IEEE Trans. Sys. Man & Cyber.*, 21:1278–1286, 1991.

[FOR 98] FORRER, H., JELTSCH, R., "A higher-order boundary treatment for Cartesian-grid methods", *J. Comput. Phys.*, 140:259–277, 1998.

[FAZ 02] FAZIO, R., LEVEQUE, R.J., "Moving mesh methods for one-dimensional conservation laws using CLAWPACK", to appear in Comp. Math. Appl., 2002.
(`http://www.math.ntnu.no/conservation/1998/`)

[FOG 99] FOGARTY, T., LEVEQUE, R.J., "High-resolution finite volume methods for acoustics in periodic or random media", *J. Acoust. Soc. Am.*, 106:17–28, 1999.

[LEV 01] LEVEQUE, R.J., CALHOUN, D., "Cartesian grid methods for fluid flow in complex geometries", In L. J. Fauci and S. Gueron, editors, *Computational Modeling in Biological Fluid Dynamics*, volume 124 of *IMA Volumes in Mathematics and its Applications*, pages 117–143. Springer-Verlag, 2001.

[LEV 97] LEVEQUE, R.J., "Wave propagation algorithms for multi-dimensional hyperbolic systems", *J. Comput. Phys.*, 131:327–353, 1997.

[LEV 98] LEVEQUE, R.J., " Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm", *J. Comput. Phys.*, 146:346–365, 1998.

[LEV 02a] LEVEQUE, R.J., *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[LEV 02b] LEVEQUE, R.J., "Finite volume methods for nonlinear elasticity in heterogeneous media", Int. J. Numer. Meth. Fluids, in press, 2002.

[LAN 00] LANGSETH, J.O., LEVEQUE, R.J., "A wave-propagation method for three-dimensional hyperbolic conservation laws", *J. Comput. Phys.*, 165:126–166, 2000.

[MIT 02] MITRAN, S., LEVEQUE, R.J., "Multidimensional wave propagation methods on moving grids", preprint, 2002.