Cranfield

College of Aeronautics Report No. 9213
November 1992

A NUMERICAL INVESTIGATION OF THE ARTIFICIAL COMPRESSIBILITY METHOD FOR THE SOLUTION OF THE NAVIER-STOKES EQUATIONS

D.T.Elsworth and E.F.Toro

Department of Aerospace Science
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Bedford MK43 0AL. England

Cranfield

College of Aeronautics Report No. 9213
November 1992

A NUMERICAL INVESTIGATION OF THE ARTIFICIAL COMPRESSIBILITY METHOD FOR THE SOLUTION OF THE NAVIER-STOKES EQUATIONS

D.T.Elsworth and E.F.Toro

Department of Aerospace Science
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Bedford MK43 0AL. England

ISBN 1 871564 557

£8.00

CRANFIELD
CoA Report 9213
November 1992

A Numerical Investigation of the Artificial Compressibility Method For the Solution of the Navier-Stokes Equations.

D.T.Elsworth and E.F.Toro
Department of Aerospace Science
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Beds MK43 0AL.

A Numerical Investigation of the Atificial Compressibility Method for the Solution of the Navier-Stokes Equations.

D.T.Elsworth and E.F.Toro
Department of Aerospace Science
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Beds MK43 OAL

June 1992

1 Abstract

The Artificial Compressibility approach is an important numerical method for solving the incompressible Navier-Stokes Equations. The application of high resolution numerical methods to the equations of the artificial compressibility approach is a relatively new phenomenon and deserves further investigation. In this paper we examine the performance of five Riemann solvers: an exact Riemann solver, and four approximate solvers. The application of reflective boundaery conditions is investigated, as well as the way in which the artificial compressibility coefficient is chosen.

2 Introduction

The range of flows modeled by the incompressible Navier-Stokes equations encompasses a large number of industrially important applications for which the Mach number is less than 0.3. This includes use in the automobile industry, architectural flows, and sub-sea applications. All of these require accurate numerical solutions to the Navier-Stokes equations that can be obtained in a realistic time scale using modern computing facilities. However, the solution of the incompressible Navier-Stokes equations is still a significant numerical challenge. The reason for this is that there is a lack of a coupling between the velocity and the pressure fields. This means that the equations themselves provide no way of explicitly updating pressure as velocity is advanced. Several schemes have been developed to solve this problem, and they can be divided in to two catagories: primative variable and non-primative variable formulations.

The non-primative variable formulations are based on the introduction of dependent variables other than velocity and pressure. Examples of methods in this catagory are: the vector potential method, the two stream function method and the vorticity-velocity method. All of these present problems such as boundary conditions, amount of data that must be stored, and inefficiency.

Primative variable methods include the pressure correction method, and the artificial compressibility method.The artificial compressibility approach, devised by A.J.Chorin[1], is arrived at by altering the incompressible equations in such a way as to result in a system

of equations in which the left hand side is hyperbolic. We wish to take advantage of the hyperbolic nature of these equations and use Riemann-problem-based-numerical-methods (or RP methods).

Riemann-problem-based methods have long been used for the computation of inviscid, compressible, time dependent flows, and have more recently been extended for use with viscous, compressible flow regimes. For flows of these types, Riemann-problem-based methods offer high shock resolution, with a large reduction in post shock oscillation when compared with traditional finite difference techniques that use artificial viscosity. This is achieved by accurately representing the physical processes of shock propagation, adding the minimum amount of artificial diffusion needed to eliminate spurious oscillations, and applying this diffusion only where it is needed. Clearly those properties relating to numerical diffusion are advantageous for flow regimes where viscosity plays an important role in determining the flow solution. By adding only the minimum amount of artificial diffusion we can come closer to representing the true flow field.

In this work several aspects of the numerical solution of the incompressible Navier-Stokes equations are examined. We look at the use of five different Riemann solvers for the equations of the artificial compressibility method. Four of these are approximate Riemann solvers and one is an exact solver. We also examine three ways in which the boundary conditions can be treated. As well as this the choice of c, the artificial compressibility coefficient, is investigated.

3 Mathematical Model and Numerical Methods

The two-dimentional, time dependant, incompressible Navier-Stokes equations are

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1}$$

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2 + P)}{\partial x} + \frac{\partial vu}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{2}$$

$$\frac{\partial v}{\partial t} + \frac{\partial (v^2 + P)}{\partial y} + \frac{\partial uv}{\partial x} = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{3}$$

Note that the density has been subsumed into the pressure. These equations clearly show the lack of a coupling between presssure and velocity. To obtain the artificial compressibility method we add a time derivative of pressure to the continuity equation, together with the artificial compressibility coefficient, c. Here then are the equations that are to be solved.

$$\frac{\partial P}{\partial t} + \frac{\partial c^2 u}{\partial x} + \frac{\partial c^2 v}{\partial y} = 0 \tag{4}$$

$$\frac{\partial u}{\partial t} + \frac{\partial (u^2 + P)}{\partial x} + \frac{\partial vu}{\partial y} = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \tag{5}$$

$$\frac{\partial v}{\partial t} + \frac{\partial (v^2 + P)}{\partial y} + \frac{\partial uv}{\partial x} = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) \tag{6}$$

The artificial compressibility method constists of solving these altered Navier-Stokes equations in some time depenent manner, until a steady state is reached. When this happens the time derivative of pressure is zero and we are left with the original Navier-Stokes equations.

The first step in numerically solving these equations is to separate the convective and diffusive parts of the problem. This has several advantages, the most important of which is to allow us to choose the most appropriate methods for solving the different parts of the equations. Splitting gives us two sets of equations, one of which is a set of hyperbolic conservion laws, as is shown by Rizzi and Erikson[2], and the other is the diffusion equation. For the hyperbolic part of the problem we will adopt a modern high resolution method known as the Weighted Average Flux method, or WAF (see Toro[3]). This is a Total Variation Diminishing (TVD) scheme which ensures that no spurious oscillations are generated during the solution procedure. It can however introduce clipping of maxima. In regions where no limiting is required to eliminate oscillations, this method reduces to the Lax-Wendroff method, for which there are no diffusion terms present in the truncation error. This is a desirable property for any method that is to be used to solve viscous problems.

The diffusion terms are represented using second order finite difference approximations. This is perhaps the simplest way of representing these terms, but it has been found that the way in which the convective terms are represented is much more important than the diffusive terms. In fact van Leer et al.[4] state that the accurate representation of the contact discontinuity is of vital importance for viscous problems.

4 Numerical Investigations

4.1 Riemann Solvers.

The numerical methods used here rely heavily on the accurate and economical solution of a large number of Riemann problems. In the past this was a significant limitation, since the Riemann solvers in use relied on complicated multistep solutions to the exact Riemann problem. However, we now have a large range of approximate techniques available for solving Riemann problems. Several solvers have been constructed for the hyperbolic left hand side of the artificial compressibility equations. These are: an exact Riemann solver based on the method put forward by Toro[5], a Roe[6] type solver, an approximate Riemann solver based on a local linearization of the equations (see Toro[7]) here refered to as the Linearized solver, a two rarefaction solver, and a version of the Harten, Leer, and Lax[8] Riemann solver known as the HLLC solver, in which the contact wave has been restored (Toro et al.[9]). All of these Riemann solvers for the artificial compressibility problem are presented by Elsworth[10]. These solvers where all used to solve three test problems. Test (a) was intended to examine the correctness of the Riemann solvers for the simplest problem, and was a split two-dimensional, time depedent test with no diffusion. The different Riemann solvers were used to solve a global Riemann problem using the WAF method. The TVD amplifier used was MINACO presented in Toro[11]. This test was performed at a CFL coefficient of 0.9, with 100 computational cells, and the artificial compressibility coefficient , c, was set at 0.9. Since there is no diffusion present we are in effect we are solving the hyperbolic left hand

side of the artificial compressibility equations. The inital conditions of this test are given in table 1 and the results are shown in figures 1 to 5. The graphs on the left of the figures shows the exact (line) and the numerical results (symbol), and those on the right show the absolute error between the exact and numerical solutions. Although this test has no physical meaning it is a good way of investigating the performance of the Riemann solvers. As a whole the results point out one of the strenghts of Riemann-problem-based methods. Although some of the approximate Riemann solvers are quite inaccurate locally, with errors in fluxes of upto 10%, the quatilty of the global solutions is good. This is illustrated by table 2. This shows a sum of the difference between the results found using the exact Riemann solver and the approximate solvers, ie.

$$E = \frac{1}{N} \sum_{i=1}^{N} \sqrt{e_i^2}$$

where

$$e_i = \left| U_i^{approximate} - U_i^{exact} \right|$$

The *exact* and *approximate* superscripts refer to the Riemann solver used to generate the numerical results. This shows the that there is very little difference between the results generated using the exact solver and those generated using the approximate solvers. These results are very reasuring, and show that, on accuracy grounds, any of the Riemann solvers could safely be used.

Another aspect of the Riemann solvers that is of interest is their computational efficiency. Test (b) examined this property. In order to investigate this the same test problem as above was used, exept that this time there were 1000 computational cells. The CPU times taken to solve this problem were measured, and these can be seen in table 3. The most efficient is the linearized Riemann solver and the least efficient was the two rarefaction solver. At first glance this suprising; after all the two rarefaction solver is supposed to a simplification. However, the two rarefaction solver is iteratively solving the wrong exact relations for most intercell boundaries, and so it may take longer for the solution to converge.

In test (c) all of the Riemann solvers were used to find the solution to a viscous, two-dimentional, recirculating flow problem analogous to the driven cavity problem. The driven cavity problem consists of a rectangular box in which the lid is translated at a constant speed. Although it is one of the simplest examples of recirculating flow there is no known analytical solution. However, with the addition of a source term, and a lid that has a variable velocity along its lenght, we can arrive at a problem for which the solution is known. This is exactly what Shih et al.[12] have done. All of the Riemann solvers were used in a program written to solve the Shih et al. problem. This uses the WAF method, but with the amplifier set to give purely second order results (i.e. the Lax-Wendroff method). The numerical conditions for this test, can be found in table 4. From these results we can see that the Riemann solver has very little effect on the convergengence of the method (table 5). The maximum error in the v-velocity along a line parrallel to the moving lid of the cavity, and at half of the depth of the cavity is shown in table 6. The accuracy of the solution is not effected to any large extent by the choice Riemann solver. Bearing in mind the results of test (a) this is not surprising. On the basis of these results the most attractive Riemann

solver is the Linearized solver, which is the least computationally expensive. The HLLC solver is also a good choice, since it is known to be very robust.

4.2 Boundary Conditions.

The next investigation that was performed, test (d), was designed to analyse the effects that different boundary condition had on the results. Here we consider only boundary conditions for the hyperbolic part of the problem. The diffusive boundary conditions are based on the no-slip condition applied at the wall. Once again the Shih et al. test problem was solved, using the linearized Riemann solver, and the WAF amplifier set to provide the Lax-Wendroff method, as in test (b). See Table 7 for the numerical conditions of this test. Three types of boundary conditions were examined. These were: (i) a linear interpolation, as proposed by Marx[13], (ii) a first order boundary condition, and (iii) a second order condition. In the first of these, (i), the pressure on the wall of the cavity is given by $P_{wall} = (P_1 + P_2)$, where P_1 and P_2 are the pressures in the first and second cells away from the wall in. The Godunov flux is then used across the boundary. The second condition, (ii), is based on the exact solution to the boundary Riemann problem. Here the pressure on the wall is found by evaluating the expression for the incoming wave based on the state in the cell adjacent to the boundary, and the wall values of velocity. The Godunov flux is then used across the boundary. Finally, the third boundary condition, (iii), was based on the exact solution to the boundary Riemann problem. The state in a ghost cell outside the computational domain was found based on the state adjacent to the boundary, and the wall velocities. Here the WAF flux was applied across the boundary. The effect of the boundary conditions on convergence is shown in table 8.The differences between conditions was marked, with the Marx type boundary condition, (i), taking significantly more time steps to converge than the other two. The first order Riemann problem based condition, (i), took the least, with the second order Riemann problem based condition, condition (iii), in between. The second order condition performs better than we have a right to expect, since higher order boundary conditions often have a significant detrimental effect on convergence. Marx has already expressed his doubts about the convergence characteristics of condition (i). The quality of the solutions generated are illustrated in table 9. The Marx type condition gave the worst results, and the second order boundary condition the best. Conditions (i) and (ii) are are firmly based on the hyperbolic nature of the problem, and these results confim our suspisions that the characteristic nature of the equations should be considered when generating boundary conditions. In addition to this, our knowlege of the exact solution to the boundary conditions means that we can apply superiour boundary conditions.

4.3 Choice of c.

Test (e) was designed to investigate two strategies for choosing the value of the artificial compressibility coefficient . Both are based on the analysis performed by Turkel[14], who states that the value of c giving the best convergence properties is given by $c^2 = 3\sqrt{(u^2+v^2)_{max}}$, where $\sqrt{(u^2+v^2)_{max}}$ is the maximum velocity in the flow. This theoretical result has been verified numerically by Marx[13]. In the first calculation, (c)(i), c^2 was set at the constant optimal value throughout the calculation.This optimal value is based on our knowlege of the solution. In the second calculation, (c)(ii), c was choosen to be the optimal value for each time step. The numerical conditions for test (e) are given in table 10. Table 11 shows the convergence properties of the two methods. The use of constant c gave better con-

vergence properties by a substantial amount. Table 12 shows the maximum errors for the two methods of chosing c. Here the use of varing c gives better quality results than fixing c throughout the calculation. However this may have noting to do with the method for chosing c, but may be related to the fact that the varying c calculations were running for more iterations than the fixed c calculations.

With this test we have a difficult desision to make as to which is the best method to use. On the one hand the convergence results favour the fixed c method, whilst accuracy considerations point towards the use of varying c. Although we may be able to have a very good guess at what the maximum flow velocity is, for example from the free stream conditions, in general we do not know the optimal value of c before hand, since we do not know the final solution. Thus,if the code is to be as general as possible it is advisable to use the varying c method, as this does not require the user to provide an optimal value of c.

5 References

[1] **Chorin A.J. 1967.** A Numerical Method for Solving Incompressible Viscous Flow Problems. Journal of Computational Physics **2**, 12-26.

[2] **Rizzi A., Erikson L. 1985.** Computation of Inviscid Incompressible Flow with Rotation. Journal of Fluid Mechanics. vol 153, pp 275-312.

[3] **Toro E.F. 1991.** The Weighted Average Flux Method Applied to the Euler Equations. (In Press).

[4] **van Leer B., Thomas J.L., Roe P.L., Newsome R.W. 1987.** A Comparison of Numerical Flux Formulæ for the Euler and Navier-Stokes Equations. AIAA-87-1104.

[5] **Toro E.F. 1992.** Riemann Problems and the WAF Method for the Two-Dimensional Shallow Water Equations. Phil. Trans Roy. Soc. London. A (1992) 338, 43-68.

[6] **Roe P.L. 1981.** Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes. Journal of Computational Physics. 43: 357-365.

[7] **Toro E.F. 1991.** A Linearized Riemann Solver for the Time-Dependent Euler Equations of Gas Dynamics. Proc. R. Soc. Lond. A(1991) 683-693.

[8] **Harten A., Lax P., van Leer D. 1983.** On Upstream Differencing and Godunov-type Methods for Hyperbolic Conservation Laws. SIAM Review 25, 35-61.

[9] **Toro E.F., Spuce M., Spears W. 1992.** Restoration of the Contact Surface in the HLL Riemann Solver. (Submitted).

[10] **Elsworth D.T. 1992.** Riemann Solvers for Solving the Incompressible Navier-Stokes Equations using the Artificial Compressibility Method. Cranfield Report 9208.

[11] **Toro E.F. 1988.** A Weighted Average Flux Method for Hyperbolic Conservation

Laws. Proc. R. Soc. Lond. A **423**, 401-418.

[12] **Shih T.M., Tan C.H., Hwang B.C. 1989.** Effects of Grid Staggering on Numerical Schemes. International Journal for Numerical Methods In Fluids. Vol 9, pp193-212.

[13] **Marx Y.P. 1991.** Evaluation of the Artificial Compressibility Method for the Solution of the Incompressible Navier-Stokes Equations. 9th GAMM Conference of Numerical Methods in Fluid Mechanics. Lausanne, Sept. 1991.

[14] **Turkel E. 1986.** Preconditioned Methods for Solving Incompressible and Low Speed Flows. ICASE Report 86-14.

u_L	1.0
v_L	1.0
P_L	0.1
u_R	1.0
v_R	0.5
P_R	1.0

Table 1: Data for Test (a).

Solver	u	v	P
Exact	0.0	0.0	0.0
Two Rare'	1.1017×10^{-7}	1.6953×10^{-7}	3.5274×10^{-8}
Roe	6.4078×10^{-6}	4.0225×10^{-5}	1.4315×10^{-6}
Linear	6.4078×10^{-6}	4.0225×10^{-5}	1.4315×10^{-6}
HLLC	1.2012×10^{-5}	2.0604×10^{-5}	3.6120×10^{-5}

Table 2: Summation of errors for Test (a).

Solver	CPU Units
Exact	321.1
Two Rare'	327.0
Roe	116.1
Linear	104.8
HLLC	172.0

Table 3: CPU Units in seconds for Test (b).

test	Re	Cells	CFL	Convergence
(i)	100	32x32	0.9	1.0×10^{-8}
(ii)	1000	32x32	0.9	1.0×10^{-8}
(iii)	100	64x64	0.5	1.0×10^{-8}
(iv)	1000	64x64	0.5	1.0×10^{-8}

Table 4: Numerical Conditions for Test (c)

Solver	(i)	(ii)	(iii)	(iv)
Exact	2606	8134	6492	20216
Two Rare'	2606	8134	6492	20216
Roe	2606	8134	6492	20215
Linear	2606	8031	6492	20215
HLLC	2606	8138	6492	20217

Table 5: Riemann Solver Performance. Number of Iterations for Test (c).

Solver	(i)	(ii)	(iii)	(iv)
Exact	4.654×10^{-3}	2.986×10^{-2}	1.199×10^{-3}	8.346×10^{-3}
Two Rare'	4.654×10^{-3}	2.986×10^{-2}	1.199×10^{-3}	8.346×10^{-3}
Roe	4.470×10^{-3}	2.872×10^{-2}	1.176×10^{-3}	8.626×10^{-3}
Linear	4.470×10^{-3}	2.872×10^{-2}	1.176×10^{-3}	8.626×10^{-3}
HLLC	4.858×10^{-3}	3.002×10^{-2}	1.224×10^{-3}	8.128×10^{-3}

Table 6: Riemann Solver Performance. Maximum Error for Test (c).

Re	1000
Cells	32×32
CFL	0.9
Convergence	1.0×10^{-8}

Table 7: Boundary Conditions: Numerical Conditions for Test (d).

Type	Iterations
Marx	5289
1st Order	2564
2nd Order	2790

Table 8: Boundary Conditions. Number of Iterations for Test (d).

Type	Maximum Error
Marx	0.3522878
1st Order	0.2974891
2nd Order	0.2927429

Table 9: Boundary Conditions. Maximum Error for Test (d).

Re	1000
Cells	32×32
CFL	0.9
Convergence	1.0×10^{-6}

Table 10: Method for c. Numerical Conditions for Test (e)

Method	Linear Solver	Exact Solver
Fixed c	2550	2755
Varying c	3312	3314

Table 11: Method for c. Number of Iterations for Test (e).

Method	Linear Solver	Exact Solver
Fixed c	2.7871×10^{-2}	2.2192×10^{-2}
Varying c	1.5971×10^{-2}	1.6208×10^{-2}

Table 12: Method for c. Maximum Error for Test (e).

Exact Solver.

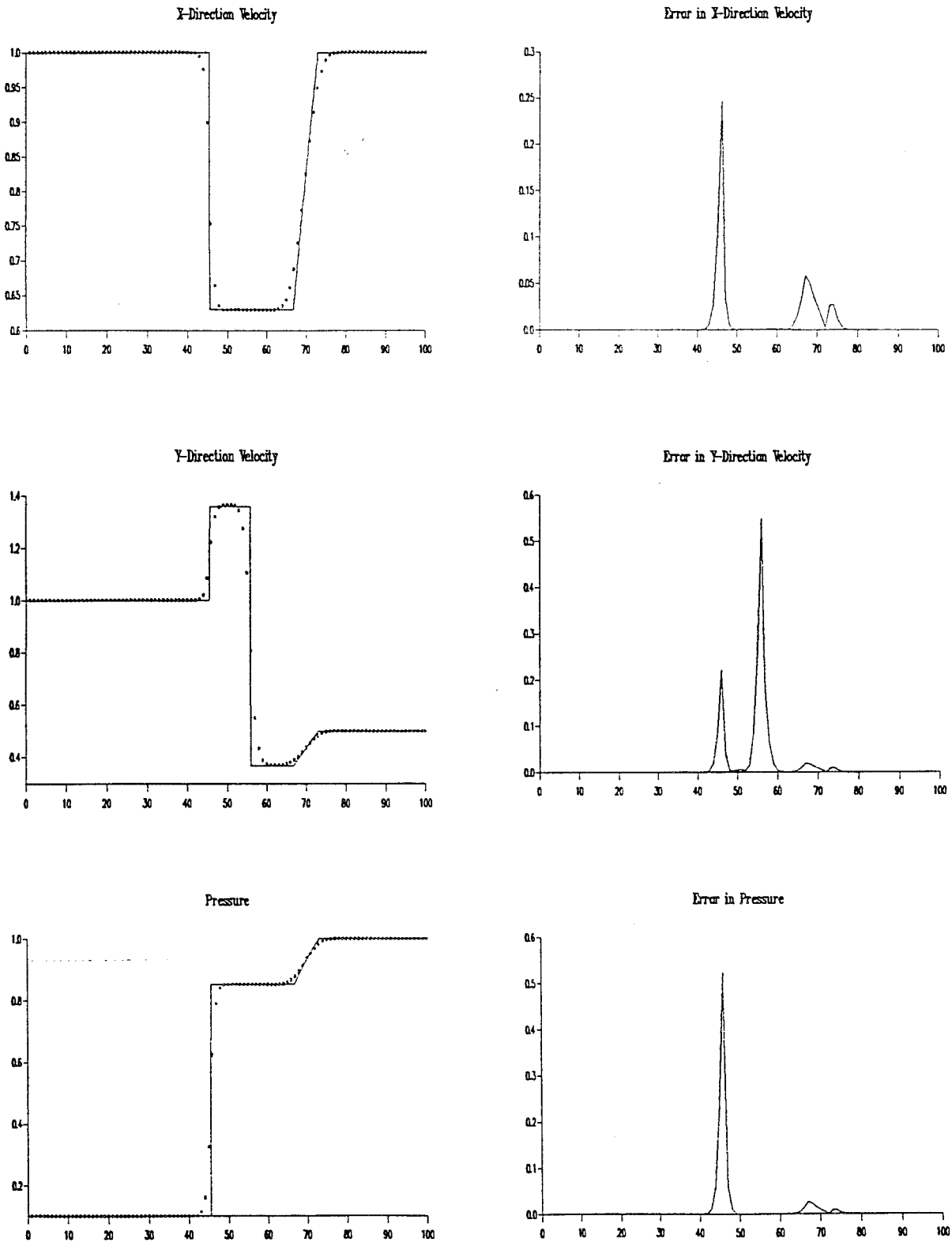

Figure 1: Exact Riemann Solver

Two Rarefaction Solver.

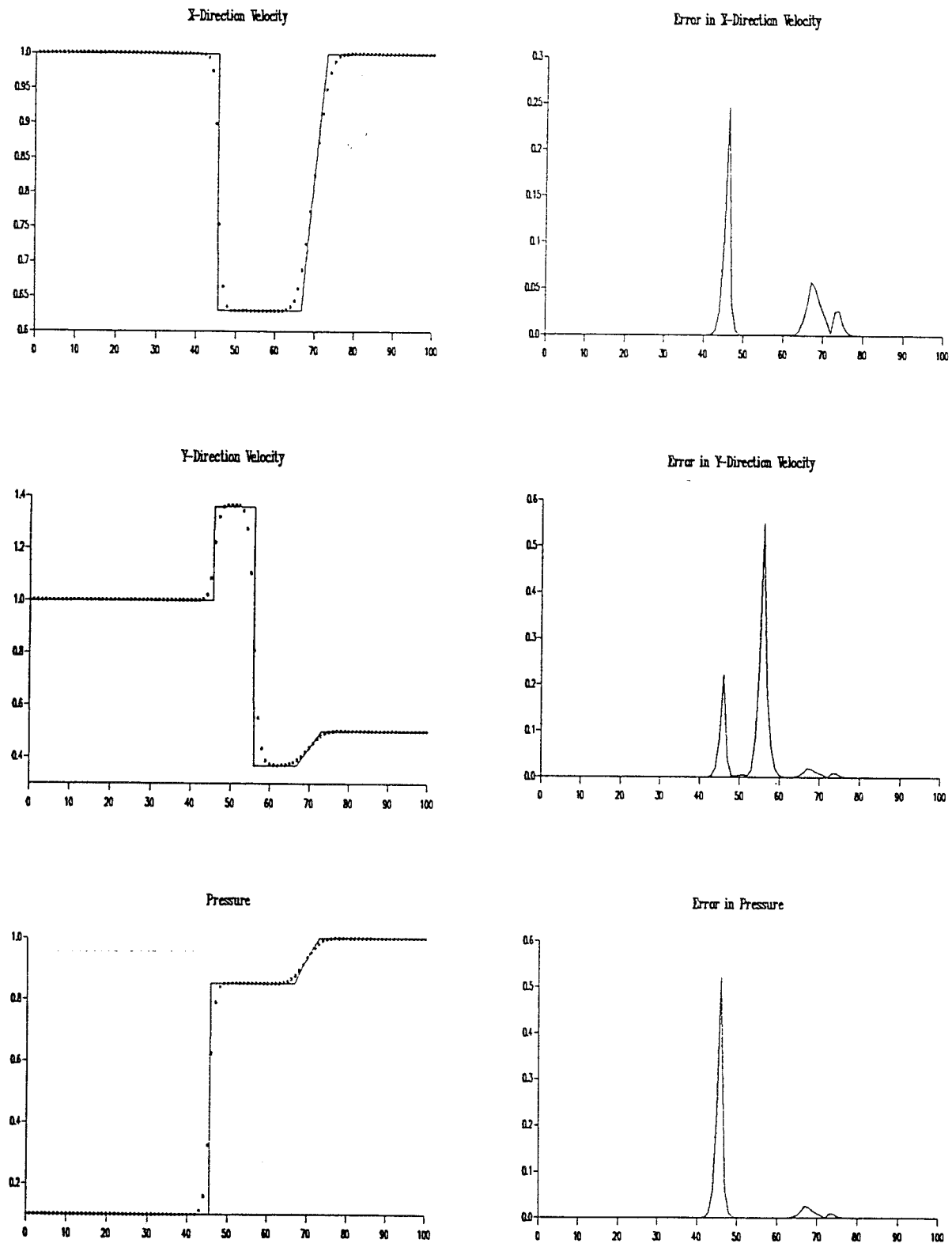

Figure 2: Two Rarefaction Riemann Solver

Roe Solver.

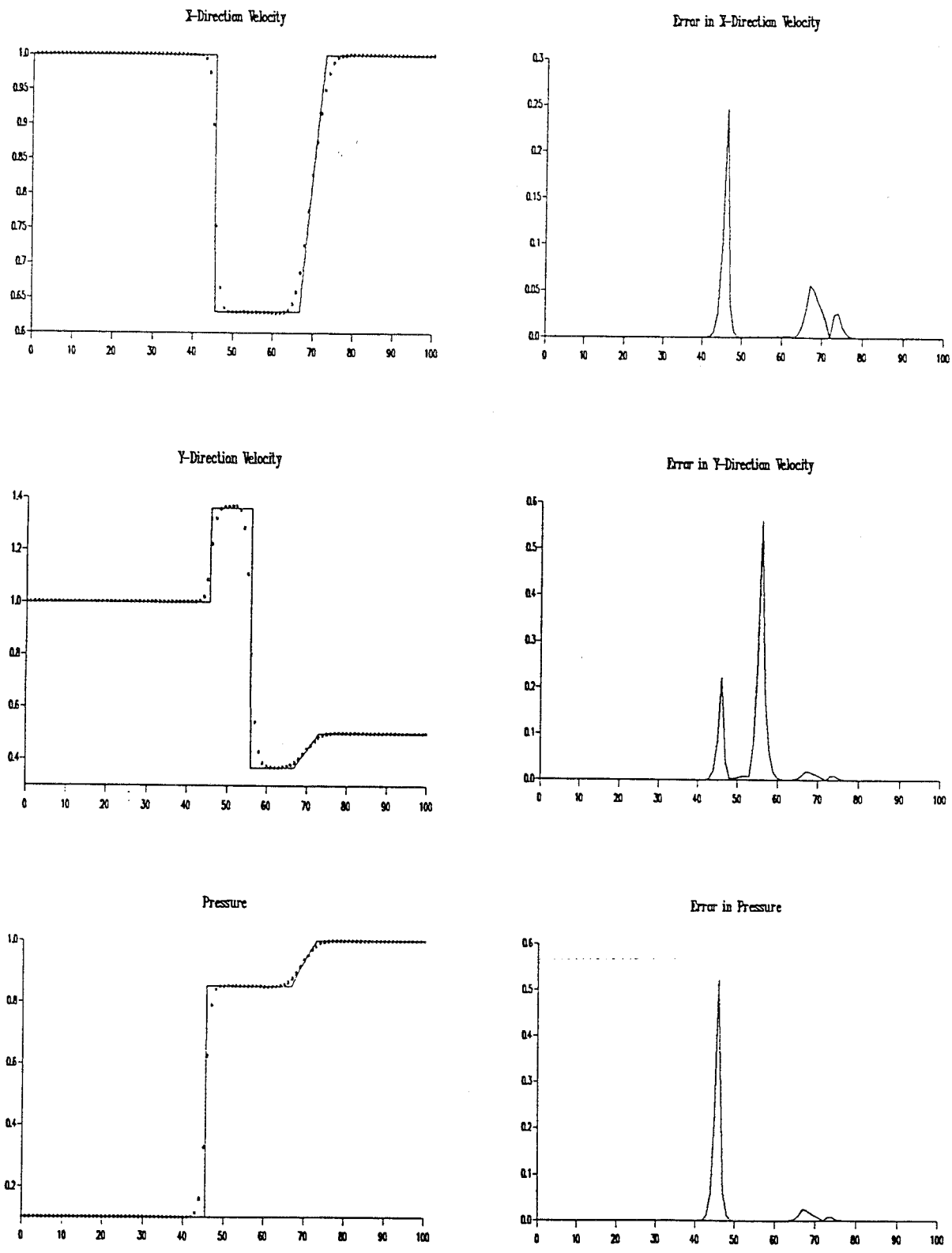

Figure 3: Roe Riemann Solver

Linearized Solver.

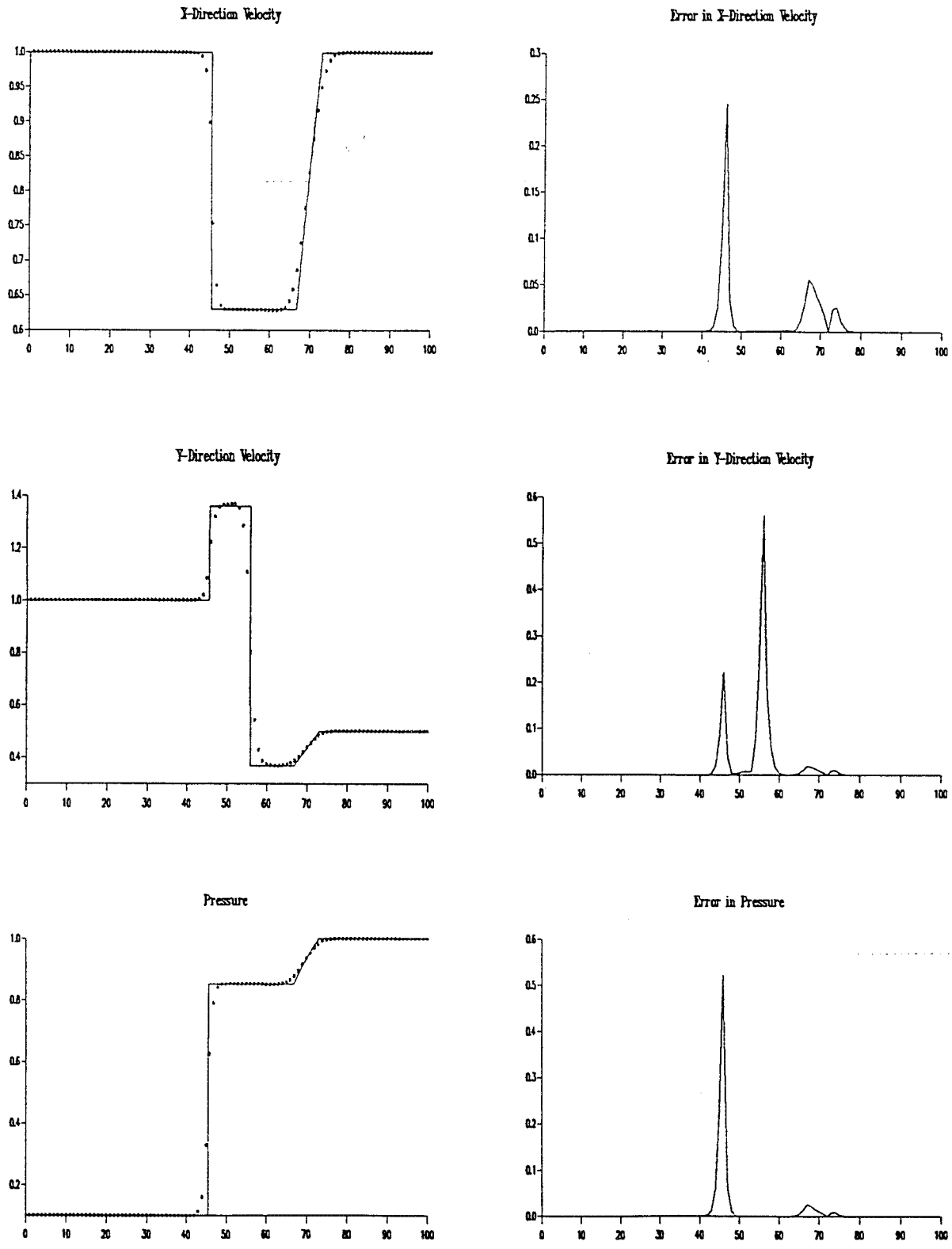

Figure 4: Linearized Riemann Solver

HLLC Solver.

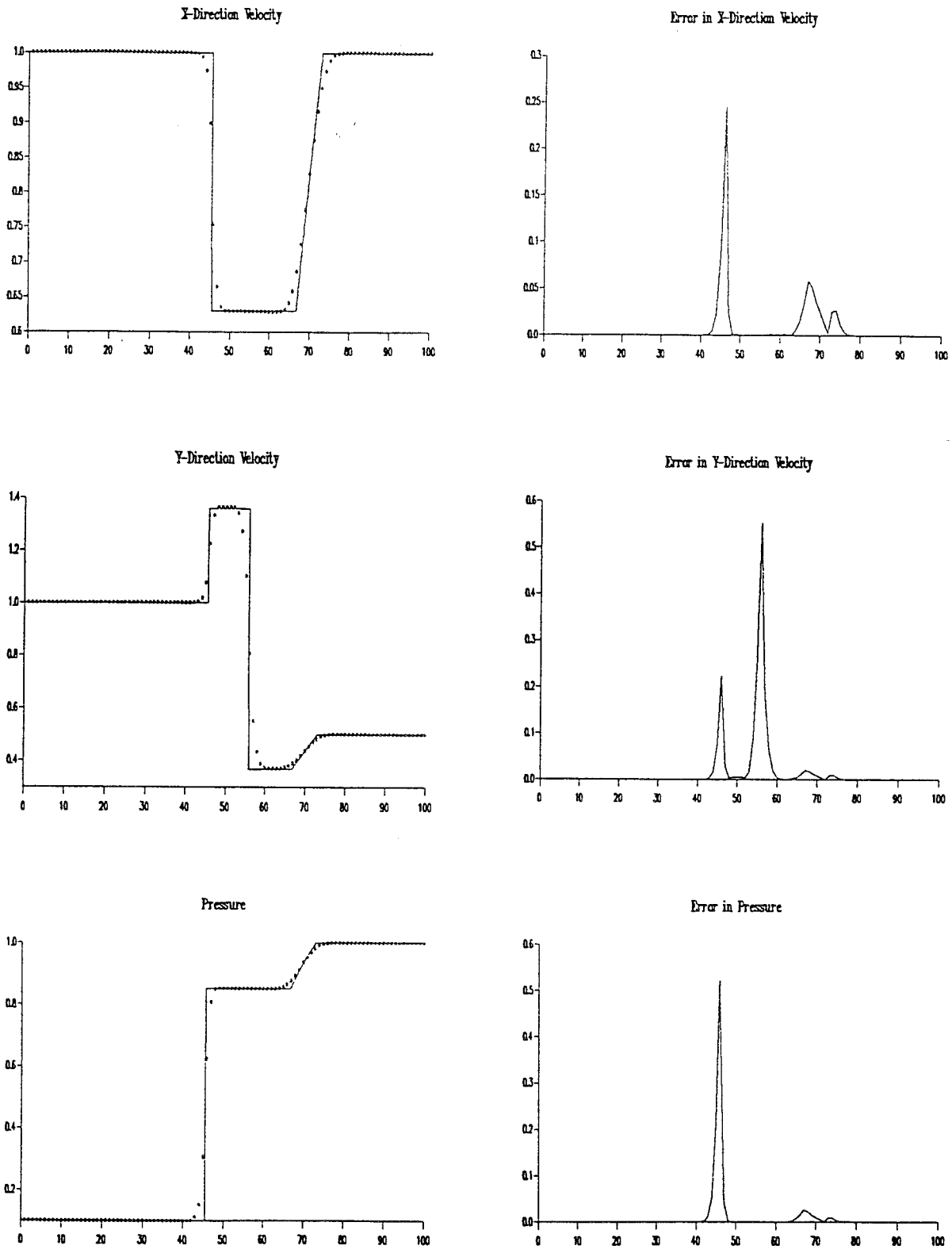

Figure 5: HLLC Riemann Solver