Cranfield
College of Aeronautics Report No. 8708
May, 1987

The Random Choice Method on a non-staggered grid utilising an efficient Riemann solver

by

E.F. Toro

Aerodynamics
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Bedford MK43 0AL, England

Cranfield
College of Aeronautics Report No. 8708
May, 1987

The Random Choice Method on a non-staggered grid utilising an efficient Riemann solver

by

E.F. Toro

Aerodynamics
College of Aeronautics
Cranfield Institute of Technology
Cranfield, Bedford MK43 0AL, England

ISBN 0 947767 60 6

£7.50

"The views expressed herein are those of the authors alone and do not necessarily represent those of the Institute."

Abstract

A practical description of the Random Choice Method (RCM) on non-staggered grids is presented. The original idea appears to be due to Colella (1982). The approach is much simpler than the staggered grid version of RCM, which is the one traditionally used. It also offers a number of advantages with regards to use of non-regular meshes in space and time, adaptive gridding and hybridisation of RCM with other methods.

Also an exact Riemann solver is presented that is shown to be very efficient and simple to use. Its performance is comparable to that of Gottlieb (1986) and significantly faster than the approximate Riemann solver of Dukowicz (1985), at least for the ideal equation of state considered here.

Table of Contents

1. Introduction

The Random Choice Method (RCM) is by now a well established numerical technique for one and quasi two-dimensional gas dynamical problems of technological interest. Some of its virtues are:

(a) ability to capture discontinuities (e.g. shocks and contacts) with infinite resolution, (b) ability to handle wave interactions efficiently and automatically, (c) absence of adhoc procedures and tuning of arbitrary parameters, (d) good physical and mathematical foundations.

The limitations of the technique as originally presented by Chorin (1976) have been gradually overcome. One of such limitations was the cost in terms of computing time, involved in solving the Riemann problem exactly. The original procedure using the Godunov's equations (Chorin 1976, Sod, 1978) consists of solving 3 non-linear algebraic equations. Solution of these equations by the iteration procedure proposed by Godunov is expensive; many iterations are needed for convergence. Solution of the same equations by the Newton-Raphson method requires less iterations but more function evaluations (Toro and Clarke 1985) so that the net gain is limited.

A recent exact Riemann solver due to Dutt (1986) does not have a very significant advantage over the above mentioned procedures.

Gottlieb (1986) has presented a new exact Riemann solver that is considerably superior to previous methods known to the author. The Riemann problem is reduced to a single non-linear algebraic equation (instead of three as in the Godunov's solution). Here we present another exact Riemann solver that is marginally better than that of Gottlieb's.

Indeed an alternative is to use approximate solutions to the Riemann problem. This, at least in principle, could result in further savings in computing time, but more importantly, these solvers have at present the ability to incorporate more general equations of state into the mathematical models. Two recent approximate Riemann solvers for general equations of state are due to Dukowicz (1985) and Glaister (1986).

Another limitation of RCM was the selection of sequences of random or quasi-random numbers with desirable statistical properties. A very satisfactory answer to this problem was provided by Collela (1982) who suggested the use of Van der Corput sequences (Hammersley and Handscombe, 1964).

Traditionally, the Random Choice Method has been applied as a two-step procedure with a secondary (staggered) grid. This approach gives rise to unnecessary programming complexities that contribute to make RCM unattractive to new users. Alternatively, RCM can be implemented as a one-step procedure on a single (non-staggered) grid. The latter approach has a number of advantages over the former one and it is the purpose of this report to present a practical description of the method and a complete FORTRAN 77 program for the Euler equations in one dimension.

Some of the advantages of programming RCM on a single grid are (a) programming is simpler (b) less storage is required (c) treatment of source terms in the governing equations to account, say, for area variations in space and time, chemical reactions, is much simpler; (d) physical meaning of sampling explicit wave systems is clearer. Also, if irregular meshes in space or time or both are to be used in the Random Choice Method (Toro 1987) then the single-grid approach becomes the only sensible way of programming RCM. Additional advantages are found when hybridisation of methods is considered (Toro, 1986, Toro & Roe 1987).

In section 2 we briefly review RCM and the main concepts involved. Section 3 is devoted to the solution of the Riemann problem and its numerical computation. Section 4 presents the way of generating the van der Corput sequences that are needed for sampling the solution of the Riemann problem. In section 5 we explain in detail ways of sampling the solution of the Riemann problem on a non-staggered grid. Section 6 deals with applications. The appendix contains a complete RCM code for the one-dimensional Euler equations.

2. Review of the Random Choice Method

RCM is applicable to quasi-linear hyperbolic systems of conservation laws

$$U_t + F(U)_x = 0 \qquad (1)$$

where, for the Euler equations

$$U = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix} \quad , \quad F(U) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{bmatrix} \qquad (2)$$

Here ρ is the density, u is the velocity and E is the total energy given by

$$E = \tfrac{1}{2}\rho u^2 + \rho e \tag{3}$$

In equation (3) e is the internal energy, which for ideal gases becomes

$$e = \frac{p}{\rho(\gamma - 1)} \tag{4}$$

γ is the ratio of specific heats. In (1) U is a function of time t and space x.

Solution of system (1) by analytical methods for general initial data is possible only for some special cases. However, for piece-wise constant data

$$U(o,x) = \begin{cases} U_\ell, & \text{if } x \leqslant x_o \\ U_r, & \text{if } x \geqslant x_o \end{cases} \tag{5}$$

the problem can be solved exactly, although not in closed form. The initial value problem (1), (5) is called the Riemann problem for (1).

The Random Choice Method uses the exact solution of the Riemann problem to provide numerical solutions to systems of the form (1) subject to general initial data. The first step in RCM is to assume that data at time t_n, say, can be approximated by piece-wise constant functions. Figure 1 illustrates the procedure for the case in which the spatial domain has been discretised into M computational cells of size Δx_i (= Δx = constant here). For smooth flows the approximation is poor but it is exact for discontinuous data, apart from errors in spatial discretisation (position of discontinuity). Data at time t_n is now a set M constant states. Fig. 1 depicts three states i - 1, i and i + 1. There are M - 1 pairs of states (i, i + 1), i = 1,2 .., M - 1. For each pair (i, i + 1) we have a Riemann problem RP (i, i + 1) where data can be described as in equation (5) with $x_o = x_i = i\Delta x$. The complete problem in complete discretised domain [0, L] is then a sequence of Riemann problems RP(i, i + 1), whose exact solutions are pieced together to form the solution to the full problem at the next time t_{n+1}.

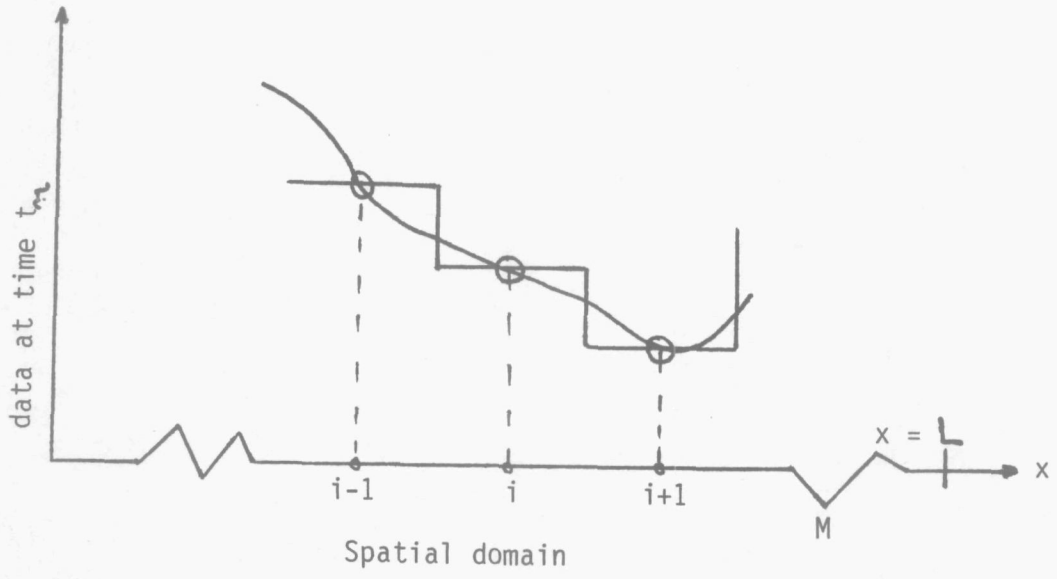


Figure 1 Approximation of data at time t_n in spatial domain [0, L] by piecewise constant functions.

The sequence of Riemann problems RP(i, i + 1) at time t_n can be defined as the set of initial value problems

$$U_t + F_x = 0, x \in [(i - \tfrac{1}{2})\Delta x, (i + \tfrac{1}{2})\Delta x], t \in [t_n, t_{n+1}]$$

$$U(t_n, x) = \begin{cases} U_i & \text{if } x < x_i = i\Delta x \\ U_{i+1} & \text{if } x > x_i \end{cases} \qquad (6)$$

As an example, consider the Euler equations (1) - (2) at time zero with $u_i = u_{i+1} = 0$, $\rho_i > \rho_{i+1}$, $p_i > p_{i+1}$. The initial profile for the density ρ (and pressure p) may then look as depicted in Fig. 2. This special Riemann problem in which the velocities in both left and right states i and i + 1 are identically zero is called a shock-tube problem.

The solution to this Riemann problem, when represented in an x-t picture, looks as depicted in Fig. 3. The solution consists of a right wave, a left wave and a middle wave. Both right and left waves can be either shocks or rarefactions and the middle wave is a contact discontinuity which separates gases originally in cell i from those originally in cell i + 1. In the present example (Fig. 3) the right wave is a shock and the left wave is a rarefaction. It is important to note that in practice one may typically have 100 Riemann problems at each time step. For most Riemann problems the structure of the wave system for each RP(i, i + 1) may be significantly simpler than that of Fig. 3.

Given states i and i + 1 by ρ_i, u_i, p_i, ρ_{i+1}, u_{i+1}, p_{i+1} the solution of the Riemann problem is completely determined if we know p* and u*, the pressure and velocity for the region between the left and right waves (see Fig. 3). Both p* and u* are constant, but the density ρ varies (discontinuously) across the contact discontinuity given by $\frac{dx}{dt} = u^*$. On the left $\rho = \rho^*_\ell$ and on the right $\rho = \rho^*_r$. Both ρ^*_ℓ and ρ^*_r are constant.

Having solved the Riemann problem with data at time t_n the exact solution at time $t = t_{n+1} > t_n$ (see Fig. 3) has a range of values that depend on the x-position. The Random Choice Method selects an x-position at random and the corresponding values of the Riemann problem solution are assigned to computing grid points.

The sampling of the explicit solution of the Riemann problem is carried out in terms of a sequence of random, or pseudorandom numbers $\{\theta^n\}$ in the interval [0,1].

The various aspects of RCM will be dealt with in more detail in the following sections.

3. Solution of the Riemann Problem

Here we solve the Riemann problem for the Euler equations (1) with the ideal equation of state (4). Re-stating the Riemann problem RP(i, i + 1) given by equation (6) we have

$$U_t + F_x = 0 \tag{7}$$

$$U(o,x) = \begin{cases} U_\ell & \text{if } x \leqslant 0 \\ U_r & \text{if } x \geqslant 0 \end{cases} \tag{8}$$

where, for convenience, the position of the initial discontinuity has been chosen to be $x_i = 0$. Subscripts ℓ and r denote left and right initial states respectively. They replace i and i + 1 in equation (6).

As pointed out in §2 the solution of the Riemann problem in the x-t plane looks as depicted in Fig. 4 and consists of three waves W_1, W_2, W_3. Wave W_2 is always a contact surface; W_1 is either a shock or a rarefaction; W_3 is either a shock or a rarefaction. Hence there are four possibilities namely,

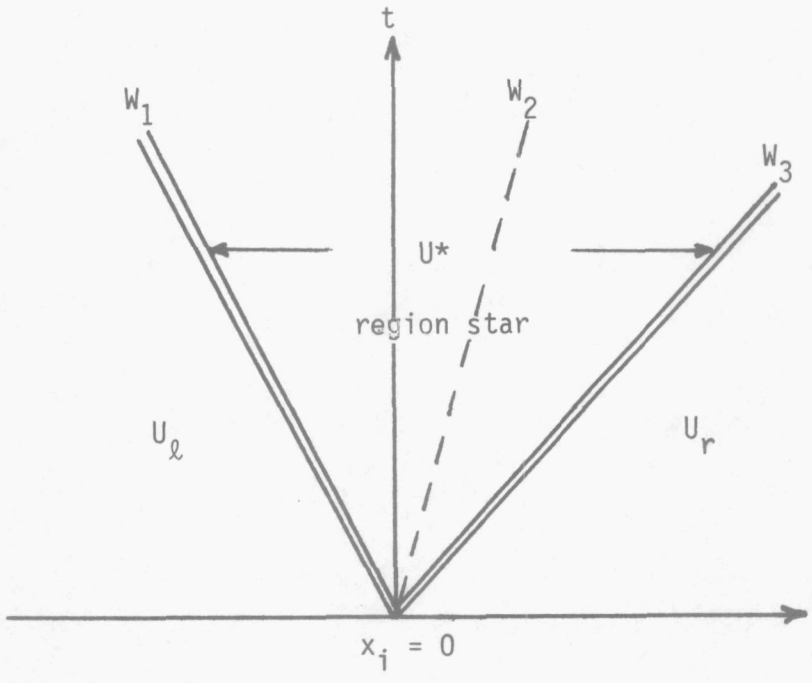


Figure 4 Solution of the Riemann problem for initial states U_ℓ (left) and U_r (right).

1. Left wave W_1 is a rarefaction and right wave W_3 is a shock. This corresponds to the shock-tube problem of Fig. 3.
2. Wave W_1 is a shock and wave W_3 is a rarefaction (mirror image of case 1).
3. Both W_1 and W_3 are rarefactions.
4. Both W_1 and W_3 are shocks.

The solution of the Riemann problem at a later time between waves W_1 and W_3 is denoted by U* and now it is convenient to take U* in terms of density, velocity and pressure, i.e. $U^* = (\rho^*, u^*, p^*)$. The density component ρ^* is piece-wise with two values $\rho^* = \rho^*_\ell$ on the left side of wave W_2 and $\rho^* = \rho^*_r$ on the right side of W_2. The velocity and pressure components u* and p* are constant throughout the region star (*) between W_1 and W_3. In finding the solution to the Riemann problem, it is these two quantities u* and p* that play the key role.

The simplest situation of all, from the point of view of the structure of the solution is case 4 above i.e. two shocks. Then the only unknowns are ρ_ℓ^*, ρ_r^*, u^* and p^*. If a rarefaction wave is present (wave W_1 or W_3 or both) then the distribution of ρ, u and p through the wave must be found (see Fig. 3). They are all continuous functions of x.

The solution strategy consists of expressing the unknown velocity u^* in terms of the pressure p^* (also unknown) and the prescribed data $U_\ell = (\rho_\ell, u_\ell, p_\ell)$ and $U_r = (\rho_r, u_r, p_r)$; i.e.

$$u^* = f(p^*, U_\ell), \quad u^* = g(p^*, U_r) \tag{9}$$

The exact form of the functions f and g in (9) has to be determined for each of the four cases referred to above. Here we follow closely the development of Glass et al (1953) in determining equation (9) for the various possible cases.

Each wave W_1 and W_3 is analysed separately.

Left wave W_1

Suppose first that the left wave W_1 is a rarefaction, then one can write

$$u^* = u_\ell + F_R(p^*, U_\ell) \tag{10}$$

with

$$F_R(p^*, U_\ell) = R_1 - R_2\, p^{*\frac{(\gamma-1)}{2\gamma}} \tag{11}$$

$$R_1(U_\ell) = \frac{2}{(\gamma-1)} C_\ell, \quad R_2(U_\ell) = \frac{2}{(\gamma-1)} C_\ell\, p_\ell^{-\frac{(\gamma-1)}{2\gamma}} \tag{12}$$

If W_1 is a shock wave then

$$u^* = U_\ell + F_S(p^*, U_\ell) \tag{13}$$

with

$$F_S(p^*, U_\ell) = S_1(p_\ell - p^*)(S_2 + p^*)^{-\frac{1}{2}} \tag{14}$$

$$S_1(U_\ell) = \left[\frac{2}{(\gamma+1)\rho_\ell}\right]^{\frac{1}{2}}, \quad S_2(U_\ell) = \frac{(\gamma-1)}{(\gamma+1)} p_\ell \tag{15}$$

Subindicies R and S are for rarefaction and shock cases respectively. C_ℓ is the sound speed on left state, i.e.

$$C = \sqrt{\frac{\gamma p_\ell}{\rho_\ell}} \tag{16}$$

Notice that both F_R and F_S are functions of one unknown only, namely p*, with data on left state as parameters.

Right Wave W_3

If W_3 is a rarefaction then

$$u^* = u_r - F_R\,(p^*, U_r) \tag{17}$$

For a right shock we have

$$u^* = u_r - F_S\,(p^*, U_r) \tag{18}$$

The functions F_R and F_S in (17) and (18) are the same as in equations (10) and (13) but evaluated in terms of the unknown p* and the right known state U_r.

These governing equations of the Riemann problem can be obtained by fairly simple algebraic manipulations of standard relations for shocks and rarefactions. For details see Glass et al 1953, Courant and Friedrichs (1948); see also Roe (1987).

In order to find the actual numerical values of u* and p* one has to consider the resulting equations for the two waves W_1 and W_3. This leads to the four cases mentioned earlier which are now summarised in Table 1. Each case gives two equations, one from the left wave W_1 and another from the right wave W_3.

Now the key step is the elimination of u* in each case. This leads to a single non-linear algebraic equation in the single unknown p*, i.e.

$$F(p^*, U_\ell, U_r) = 0 \tag{19}$$

F is shown on the last column of Table 1 for each case.

Case	Wave 1	Wave 3	relation between u* and p*	single equation F = 0 for p*
1	rarefaction	shock	$u^* = u_\ell + F_R(p^*, U_\ell)$ $u^* = u_r - F_S(p^*, U_r)$	$F_R(p^*, U_\ell) + F_S(p^*, U_r) + u_\ell - u_r = 0$
2	shock	rarefaction	$u^* = u_\ell + F_S(p^*, U_\ell)$ $u^* = u_r - F_R(p^*, U_r)$	$F_S(p^*, U_\ell) + F_R(p^*, U_r) + u_\ell - u_r = 0$
3	rarefaction	rarefaction	$u^* = u_\ell + F_R(p^*, U_\ell)$ $u^* = u_r - F_R(p^*, U_r)$	$F_R(p^*, U_\ell) + F_R(p^*, U_r) + u_\ell - u_r = 0$
4	shock	shock	$u^* = u_\ell + F_S(p^*, U_\ell)$ $u^* = u_r - F_S(p^*, U_r)$	$F_S(p^*, U_\ell) + F_S(p^*, U_r) + u_\ell - u_r = 0$

Table 1: Summary of all possible cases, relations between u* and p* and single equation for p*, where F_R and F_S are given by equations (11) and (14).

Equation (19) can be solved numerically. We do so here by a Newton-Raphson iterative procedure. Logical decisions are involved, because the form of F in (19) can be any of the four possibilities listed in Table 1, depending on the value of p* at any particular iteration, i.e. one does not know a priori which of the four possibilities will occur. If one assumes, however, that both waves W_1 and W_3 are rarefaction waves (case 3) then one can find an exact solution for p*. Interestingly enough it is also a good approximation to p* in other cases. From equation (11) and Table 1 (case 3) we have

$$u_\ell - u_r + R_1(U_\ell) + R_1(U_r) - [R_2(U_\ell) + R_2(U_r)]p^{*\frac{(\gamma-1)}{2\gamma}} = 0$$

which gives

$$p^* = \left\{ \frac{C_\ell + C_r + \frac{(\gamma-1)}{2}(u_\ell - u_r)}{C_\ell / p_\ell^{\frac{(\gamma-1)}{2\gamma}} + C_r / p_r^{\frac{(\gamma-1)}{2\gamma}}} \right\}^{\frac{2\gamma}{(\gamma-1)}} \tag{20}$$

This approximation for p* was first given by Gottlieb (1986) as an initial guess for the numerical solution of the Riemann problem. It is remarkably accurate even when shocks are involved.

Any of the other 3 cases will involve an algebraic equation that is difficult to handle. One possible approach is this: take p_0 to be the approximation given by equation (20), Taylor expand in a neighbourhood of p_0 and truncate after second order terms, say. This would give an approximate solution of the Riemann problem in terms of the roots of a quadratic equation. Provided p_0 is sufficiently reliable for determining the particular case of Table 1 to choose, then the quadratic solution would give a practical approximate solution. The algebra involved becomes too discouraging and the number of operations involved may well turn out to be larger than for the exact iterative solution. Here, we solve the algebraic equation for p* using an iterative numerical technique.

Numerical solution for p*

We want to solve equation (19) for p* numerically. Since it is a non-linear equation an iterative procedure must be employed. We use Newton-Raphson. It is fast but requires function evaluations and derivatives, which increases the expense. The secant method is an alternative; it only requires function values at each step but it is slower than Newton-Raphson, it requires about 1.5 times as many iterations as Newton-Raphson to achieve a given accuracy.

Newton-Raphson works like this: suppose we want to solve the non-linear equation

$$f(x) = 0 \tag{21}$$

for the unknown x. Taylor expanding about x_o, a guessed known value, we obtain

$$f(x) = f(x_o + h) = f(xo) + hf'(xo) + \frac{h^2}{2!} f''(x_o) + \ldots \tag{22}$$

Neglect terms of order higher than 1 and assume $x_o + h$ is solution of equation (21) for a value of the increment h still to be found i.e.

$$f(x_o + h) = 0$$

then

$$f(x_o) + hf'(x_o) = 0 \tag{23}$$

from which the unknown increment h is

$$h = -f(x_o)/f'(x_o) \tag{24}$$

This says that

$x_o + h = x_o - f(x_o)/f'(x_o)$ is a solution (approximate) of equation (21). One then uses this procedure in an iterative fashion and writes

$$x_{n+1} = x_n - f(x_n)/f'(x_n) \tag{25}$$

To stop the iteration procedure (25) a tolerance TOL is prescribed and if

$$| x_{n+1} - x_n | \leq TOL \tag{26}$$

the iteration is stopped and x_{n+1} is taken as the solution. For large values of x it is best to use the stopping criterion (relative error)

$$| (x_{n+1} - x_n)/x_{n+1} | \leq TOL \tag{27}$$

In applying procedure (25) to our equation (19) for p* we need to provide expressions for the derivatives. These are

$$\frac{dF_R}{dp^*} = \frac{C_k}{\gamma} p_k^{-\frac{(\gamma-1)}{2\gamma}} p^{*-\frac{(\gamma+1)}{2\gamma}} \tag{28}$$

and

$$\frac{dF_S}{dp^*} = -S_1(S_2 + p^*)^{-\frac{1}{2}}[1 + \tfrac{1}{2}(p_k - p^*)(S_2 + p^*)^{-1}] \tag{29}$$

for rarefactions and shocks respectively.

Here k denotes ℓ (left) or r (right) states. The program to compute p* using iteration (25) with p_o (guessed value) given by equation (20) is the subroutine RIEMANN. This is described in more detail in the appendix. Once p* is known (within a given accuracy), the velocity u* can be found from any of the relevant (correct case) equations of Table 1. We find that an economical way of doing this is by taking the mean value of the last function values used to find p* in the Newton-Raphson iteration, i.e.

$$u^*_{n+1} = \tfrac{1}{2}[u_\ell + u_r + F_{W_1}(p^*_n, U_\ell) + F_{W_3}(p^*_n, U_r)] \tag{30}$$

where $F_{W_1}(p^*_n, U_\ell)$ is the corresponding function for the left wave W_1 evaluated at the pennultimate iterate p^*_n and $F_{W_3}(p^*_n, U_r)$ is that for wave W_3.

Test problems

We consider four problems to test our Riemann solver. These are listed on Table 2 where comparison is also made with Gottlieb's Riemann solver. Test problem 1 consists of the initial data for Sod's shock-tube problem whose solution has a left travelling rarefaction, a right travelling shock followed by a contact discontinuity. The solution for p* and u* given by the two Riemann solvers are coincident as they should be. Notice how good the initial value p^*_o is, even for this case containing a fairly strong shock of pressure ratio greater than 3. Here we have taken TOL = 10^{-6} but in practice TOL = 10^{-4} gives sufficient accuracy. This reduces the number of iterations required for convergence from 3 to 2 in this case.

TEST	ρ_ℓ	u_ℓ	p_ℓ	ρ_r	u_r	p_r	guessed value p_o	Present Method			Gottlieb's Method
								p*	u*	Iterations	Iterations
1	1.0	0.0	1.0	0.125	0.0	0.1	0.3047667	0.303130	0.927453	3	3
2	1.0	-1.0	1.0	1.0	1.0	1.0	0.273586	0.273586	0.000000	1	1
3	1.0	1.0	1.0	1.0	-1.0	1.0	2.983884	2.926650	0.000000	3	3
4	0.353	-1.78	14.0	0.1	-11.6	0.5	16.96563	13.97732	-1.772092	4	5

Table 2. Test problems for Riemann solver. Here TOL = 10^{-6}, $\gamma = 1.4$ in tests 1 - 3 and $\gamma = 1.667$ in Test 4. Solution values for p* given by Gottlieb's solver are identical to those of present solver.

Test problem 2 is designed so as to have two rarefaction waves. This type of problem occurs in practice when using reflecting boundary conditions. Notice that the guessed value p_0^* is 'exact' here and so only one iteration is performed. For finding p* this iteration is not required in this case but it is not a total waste since the function values obtained in this iteration are to be used for evaluating u*, which is indeed required.

Test problem 3 is designed so as to produce 2 shocks. Again these problems occur when reflecting boundary conditions are being used. Here the guessed value p_0^* is still quite good; it only differs by about 2% from the correct value.

Finally, test problem 4 is designed so as to produce a single right facing (almost stagnant) shock wave of pressure ratio 27.95. Here p_0^* has an error of about 20%.

These test problems partially validate the Riemann solver presented here. It appears to be of comparable efficiency to Gottlieb's Riemann solver (Gottlieb, 1986). On operation count, per iteration, the present method has a slight advantage, although the comparison is based on the respective computer-program versions of the methods, both of these written by the present author and therefore accusations of bias are not discarded. Rarefactions require 7 operations for the present and 11 for that of Gottlieb. For shocks the number of operations is 11 for both methods. The kind of operations are roughly similar.

The reader should realise that when solving a practical problem one has to solve, say, 100 Riemann problems per time step. Probably 2 to 3 of these are severe and 2 to 3 iterations may be needed to find the solution in each case. The remaining Riemann problems are virtually trivial and only one iteration is required. Thus the expense of solving the Riemann problem exactly, is rather modest.

Dukowicz (1985) presented an approximate Riemann solver that requires no iteration. However, at least for the ideal equation of state, this method is extremely expensive in comparison with the present one (or that of Gottlieb's). By counting operations on Dukowicz's program one sees that for the cheapest (case A) of the four cases more than 60 operations are needed including the use of intrinsic functions. Case D (the most expensive requires about 100 operations. Roughly, the approximate Riemann solver is about four times more expensive than the present exact Riemann solver. It must be said however, that Dukowicz's approximate solver is valid for a general equation of state, unlike the present Riemann solver.

4. Generation of van der Corput pseudo-random numbers

The exact solution of the Riemann problem is available to us. Now we require a sequence of pseudo-random numbers (truly random numbers do not give the best results) $\{\Theta_n\}$, with Θ_n in [0,1] to sample the solution. Colella (1982) found that the van der Corput sequences give, overall, best results.

In this section we introduce the van der Corput sequences and discuss ways of generating them on a computer. A general van der Corput sequence $\{\Theta_n\}$ depends on two parameters k_1, k_2 with $k_1 > k_2 > 0$, both integer and relatively prime. Then the (k_1, k_2) van der Corput sequence $\{\Theta_n\}$ is formally defined as follows

$$\Theta_n = \sum_{i=0}^{m} A_i k_1^{-(i+1)} \tag{31}$$

where

$$A_i = k_2 a_i (\text{mod } k_1) \tag{32}$$

and

$$n = \sum_{i=0}^{m} a_i k_1^i \tag{33}$$

This says that the pseudo-random number $\Theta_n \varepsilon[0,1]$ is a summation of m terms, each one involving powers of the integer k_1. The coefficients A_i are found from equation (32) which in turn requires the determination of the coefficients a_i in equation (33).

Lets begin by considering equation (33). This says that the natural number n (non-negative integer) is expressed in scale of notation with radix k_1 (i.e. system of base k_1). For instance if $k_1 = 2$ then n is expressed in terms of the binary expansion. Equation (33) becomes

$$n = \sum_{i=0}^{m} a_i 2^i$$

If n = 3 then

$$3 = 1 + 1 \times 2^1 \quad , \quad m = 1, \; a_0 = 1, \; a_1 = 1.$$

Table 3 contains coefficients a_i of equation (33) for 10 natural numbers when $K_1 = 2$ and $k_1 = 3$.

	$k_1 = 2$					$k_1 = 3$				
n	a_o	a_1	a_2	a_3	m	a_o	a_1	a_2	a_3	m
1	1				0	1				0
2	0	1			2	2				1
3	1	1			2	0	1			2
4	0	0	1		3	1	1			2
5	1	0	1		3	2	1			2
6	0	1	1		3	0	2			2
7	1	1	1		3	1	2			2
8	0	0	0	1	4	2	2			2
9	1	0	0	1	4	0	0	1		3
10	0	1	0	1	4	1	0	1		3

Table 3 Coefficients a_i for $k_1 = 2$ and $k = 3$ for $n = 1, \ldots, 10$.

The next stage is to find the 'modified' coefficients A_i from equation (32) which means that A_i is the remainder when dividing $k_2 a_i$ by k_1 ($A_i < k_1$). The simplest example is $k_2 = 1$ then $A_i = a_i \ \forall_i$.

If we were to find the (3,2)($k_1 = 3$, $k_2 = 2$) van der Corput sequence then the a_i coefficients of Table 3 for $k_1 = 3$ must be modified. These (A_i) are given on Table 4.

Having found the coefficients A_i, the summation given by equation (31) is formed to obtain the random number Θ_n.

n	a_o	a_1	a_2	A_o	A_1	A_2
1	1			2		
2	2			1		
0	0	1		0	2	
4	1	1		2	2	
5	2	1		1	2	
6	0	2		0	1	
7	1	2		2	1	
8	2	2		1	1	
9	0	0	1	0	0	2
10	1	0	1	2	0	2

Table 4 Coefficients a_i and A_i for ten members $\Theta_n (n = 1, 10)$ of the van der Corput sequence (3,2).

Table 5 gives ten members of the van der Corput sequences (2,1) and (3,2) translated to the interval $[-\frac{1}{2}, \frac{1}{2}]$

n	Θ_n for (2,1)	Q_n for (3,2)
1	0.0	0.1667
2	-0.25	-0.1667
3	0.25	-0.2778
4	-0.375	0.3889
5	0.125	0.0556
6	-0.125	-0.3889
7	0.375	0.2778
8	-0.4375	-0.0556
9	0.0625	-0.4259
10	-0.1875	0.2407

Table 5

The subroutine VDCK12( ) given in the appendix performs the calculation of the quasi-random numbers Θ_n needed in the sampling procedure to be described in the next section.

5. Sampling the solution of the Riemann problem on a non-staggered grid

For a given time t_n data is available in the form of constant states ρ_i, u_i, p_i ($i = 1,2, .. M$). Each pair of neighbouring states i and i+1 form a Riemann problem RP(i, i+1). Consider a computational cell i of length Δx with nodal values $U_i^n = (\rho_i^n, u_i^n, p_i^n)$ at time level n (data). We want a procedure to update these values at a later time level n+1 <u>in one step</u>. This can be achieved by considering the two Riemann problems RP(i-1, i) and RP(i, i+1) whose solutions may be represented as in Fig. 4. For a sufficiently short time step ΔT, the solution U_i^{n+1} at $x = (i-\frac{1}{2})\Delta x$ (cell i) at time $t_{n+1} = t_n + \Delta T$ will only be affected by the right travelling waves of the left Riemann problem RP(i-1, i) and the left travelling waves of the right Riemann problem RP(i, i+1). These are the only waves transversing cell i.

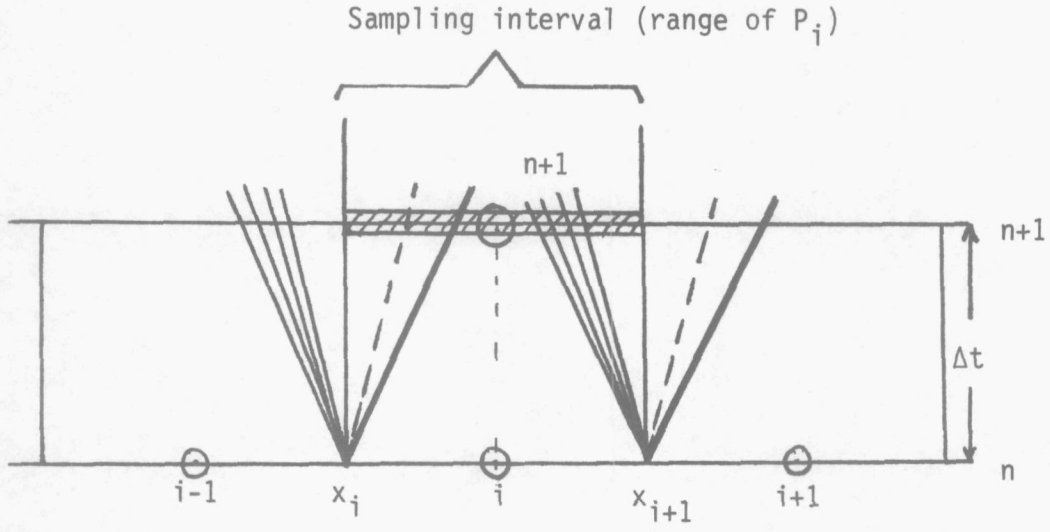


Figure 5 One-step RCM on single grid. Cell i is between $x_i = (i - 1)\Delta x$ and $x_{i+1} = i\Delta x$. Random Point P_i lies between x_i and x_{i+1}.

RCM will take the updated solution U_i^{n+1} at time level n+1 to be determined by the exact solution of the Riemann problems RP(i-1, i), RP(i, i+1) transversing cell i, evaluated at a random position $P_i = (x_i + \Theta_n \Delta x, t_{n+1})$ in the x-t plane of Fig. 4. Here Θ_n is a quasi-random number in the interval [0,1]. For instance if $\Theta_n = 0$ then P_i lies on the intercell boundary $x = x_i$ and the solution U_i^{n+1} is the exact solution of RP(i-1, i) at that position (between contact and rarefaction). If $\Theta_n = 1$ P_i is on right intercell boundary at $x = x_{i+1}$ and U_i^{n+1} takes on the value of the exact solution of RP(i, i+1) at $x = x_{i+1}$. If $\Theta_n = \frac{1}{2}$, then (depending on size of ΔT) $U_i^{n+1} = U_i^n$, i.e. old value U_i^n remains unaltered. In summary we may write

$$U_i^{n+1} = V_i^{n+1}(P_i) \tag{34}$$

where V_i^{n+1} is the exact solution of the Riemann problems RP(i-1, i) and RP(i, i+1) at time $t_{n+1} = t_n + \Delta T$ and

$$P_i = (x_i + \Theta_n \Delta x, t_n + \Delta T) \tag{35}$$

In choosing the time step size ΔT has to obey the Courant Friedrich - Lewy condition (CFL condition). In the context of the Random Choice Method the CFL condition simply says that the time step ΔT should be chosen small enough so as to avoid wave interaction before sampling the interval $[x_i, x_{i+1}]$ in search of position P_i. A popular CFL condition that satisfies this requirement (Ref. 8) is

$$\Delta T = C_S \, (\tfrac{1}{2}\Delta x)/S_{max} \tag{36}$$

where S_{max} is the maximum wave speed present and can be calculated from the data as

$$S_{max} = \max \{ |u_i^n| + a_i^n \} \tag{37}$$

where a_i^n is the sound speed. In equation (36) C_S is a safety coefficient in the interval $(0,1]$. The CFL condition (36) prevents waves from transversing more than half a cell size. This is a convenient CFL-condition that facilitates sampling procedures and storage of information as we shall see later, but it is inefficient. One can choose larger, and still regular, time steps that satisfy the CFL conditions of avoiding wave interaction before sampling (Ref.10). For simplicity we adopt CFL condition (36) here.

It should be made clear that for updating each cell value U_i^n we only require to solve one Riemann problem per cell i (except for cell i = 1). For cell i+1 the left Riemann problem solution is already available.

The complete sampling procedure is illustrated in the flow charts of Figs. 5 and 6. Subroutine SAMPL in the appendix carries out the procedure. The routine is written so as to sample an interval $[x_i - \frac{1}{2}\Delta x, x_i + \frac{1}{2}\Delta x]$ of the general Riemann problem PP(i-1, i). See Fig. 4. Suppose we want to update U_i^n in cell i to the new value U_i^{n+1}. Solve RP(i-1, i) and use a given random number Θ_n in [0, 1] (only one Θ_n per time step is used). If $0 < \Theta_n \leqslant \frac{1}{2}$ we call the routine SAMPL that will sample the right-hand part of the left Riemann problem RP(i-1, i) (see Fig. 4). This part will deal with the right moving waves of RP(i-1, i). If $\frac{1}{2} < \Theta_n \leqslant 1$ then we solve RP(i, i+1). We want to sample the left moving waves of the right-hand Riemann problem RP(i, i+1). Set $\bar{\Theta}_n = \Theta_n - 1$ and call SAMPL as before.

In the sampling procedure itself, irrespective of whether we are sampling the positive (right) or negative (left) part of a Riemann problem there are two main cases to consider. The sampled point P_i of equation (35) lies on the left of the contact discontinuity (that is the middle wave W_2) or on the right. The first case is illustrated by the flow chart of Fig. 6 while the second case is illustrated by Fig. 7.

Consider the case of Fig. 6, i.e. the sampled point P_i is on the left side of the contact discontinuity given by $\frac{dx}{dt} = u^*$. Then there are two possibilities. The left wave (to the left of the contact) is a shock (Picture 1 of Fig. 6) or a rarefaction (Picture 2 of Fig. 6). Consider Picture 1 first, i.e. the left wave is a shock wave, then P_i lies either

behind the shock (region 2) or P_i lies in front of the shock (region 1). In the first of these 2 cases the values of the quantities ρ, u, p are obtained by using standard shock relations. In fact u* and p* are already known from the solution of the Riemann problem. Calculating the density involves extra work. Using the shock relation

$$\frac{p^*}{p_\ell} = \frac{(\rho^*/\rho_\ell)(\gamma + 1) - (\gamma - 1)}{(\gamma + 1) - (\gamma - 1)(\rho^*/\rho_\ell)}$$

one gets the value ρ_ℓ^* for the density behind the shock as

$$\rho_\ell^* = \rho_\ell \left[\frac{(\gamma + 1)p^*/p_\ell + (\gamma - 1)}{(\gamma - 1)p^*/p_\ell + (\gamma + 1)}\right]$$

If the sampled point P_i lies in front of the shock (region 1) then the solution takes on the values of $\rho_\ell, u_\ell, p_\ell$ i.e. data on left state.

If the left wave is a rarefaction (Picture 2, Fig. 6) the sampling procedure is slightly more complicated than the shock case. There are now three possibilities, namely, region 3 (between the contact and the tail of the rarefaction); region 4 (within the rarefaction fan) and region 5 (the original left state). In region 3 we only need ρ_ℓ^*; values for the pressure and velocity are p* and u* which are known from the solution of the Riemann problem. Region 5 is trivial. Region 4 is the most difficult case. All flow quantities vary smoothly with x across the rarefaction fan. Now suppose the sampled point P_i lies inside the fan and has coordinates $(\hat{x}, \hat{t})$ (see equation 35). Consider a characteristic emanating from the origin and passing through P_i. This line has slope dx/dt = u - c where both u and C are unknown. This gives

$$C = u - \hat{x}/\hat{t}$$

Using constancy of the left Riemann invariant one can write

$$u_\ell + \frac{2C_\ell}{(\gamma - 1)} = u + \frac{2C}{(\gamma - 1)}$$

Substitution of C (sound speed) from previous relation one obtains the velocity u as

$$u = \frac{2}{(\gamma + 1)}\left[\frac{\hat{x}}{\hat{t}} + C_\ell + \frac{(\gamma - 1)}{2} u_\ell\right]$$

Values for pressure and density follow from the definition of sound speed (now known in terms of u) and the isentropic law i.e.

$$\rho = [C^2/(\gamma A)]^{\frac{1}{\gamma-1}} \quad \text{and}$$

$$p = A\rho^{\gamma}$$

So far we have dealt with half of the sampling procedure. If P_i lies to the right of the contact discontinuity an entirely similar process of sampling is carried out for waves on the right hand side of the contact. This is illustrated in detail in the flow chart of Fig. 7 We do not repeat the calculations.

The relations used are standard equations of Gas Dynamics and alternatives are possible. For instance for calculating the density behind a shock wave one could also use the shock Mach number which in turn can be used to find the shock speed. In fact this is what is done in the program SAMPL.

6. Applications

Here we apply RCM on a non-staggered grid to the shock-tube test problem devised by Sod (Sod 1978). A tube of length L = 1 is divided into two sections by a diaphragm at $x = \frac{1}{2}$. On the left half densities and pressures are higher than those on the right half. We have

$$\rho_\ell = 1.0, \quad u_\ell = 0, \quad p_\ell = 1.0$$
$$\rho_r = 0.125, \quad u_r = 0, \quad p_r = 0.1$$

The flow field created by the rupture of the diaphragm can be simulated by solving the 1-D unsteady Euler equations. Figs. 8 -10 show the computed flow field at three difficult times. In each Figure we plot density, velocity, pressure and specific internal energy against distance. Symbols denote the RCM solution and full lines denote the exact solution. Notice that the full structure of the wave system is already developed in Fig. 8. There is a shock wave travelling right, a constant discontinuity following the shock and a rarefaction wave travelling left. Notice how accurate the numerical solution is at early times. Randomness will be present at later times, see Fig. 10. Discontinuities are absolutely sharp and their positions are exact (at least for times shown).

7. Conclusions

A full description of the Random Choice Method on a non-staggered grid has been presented. Also, an exact Riemann solver is included, which is marginally faster than that of Gottlieb's (1986). A complete RCM program is given. Application to Sod's shock tube problem is made to validate the program. The reader can easily make relevant changes to apply the method to other problems.

The advantages of programming RCM on non-staggered grids are significant. Also, solving the Riemann problem exactly has become an efficient procedure that can be even faster than using approximate solvers.

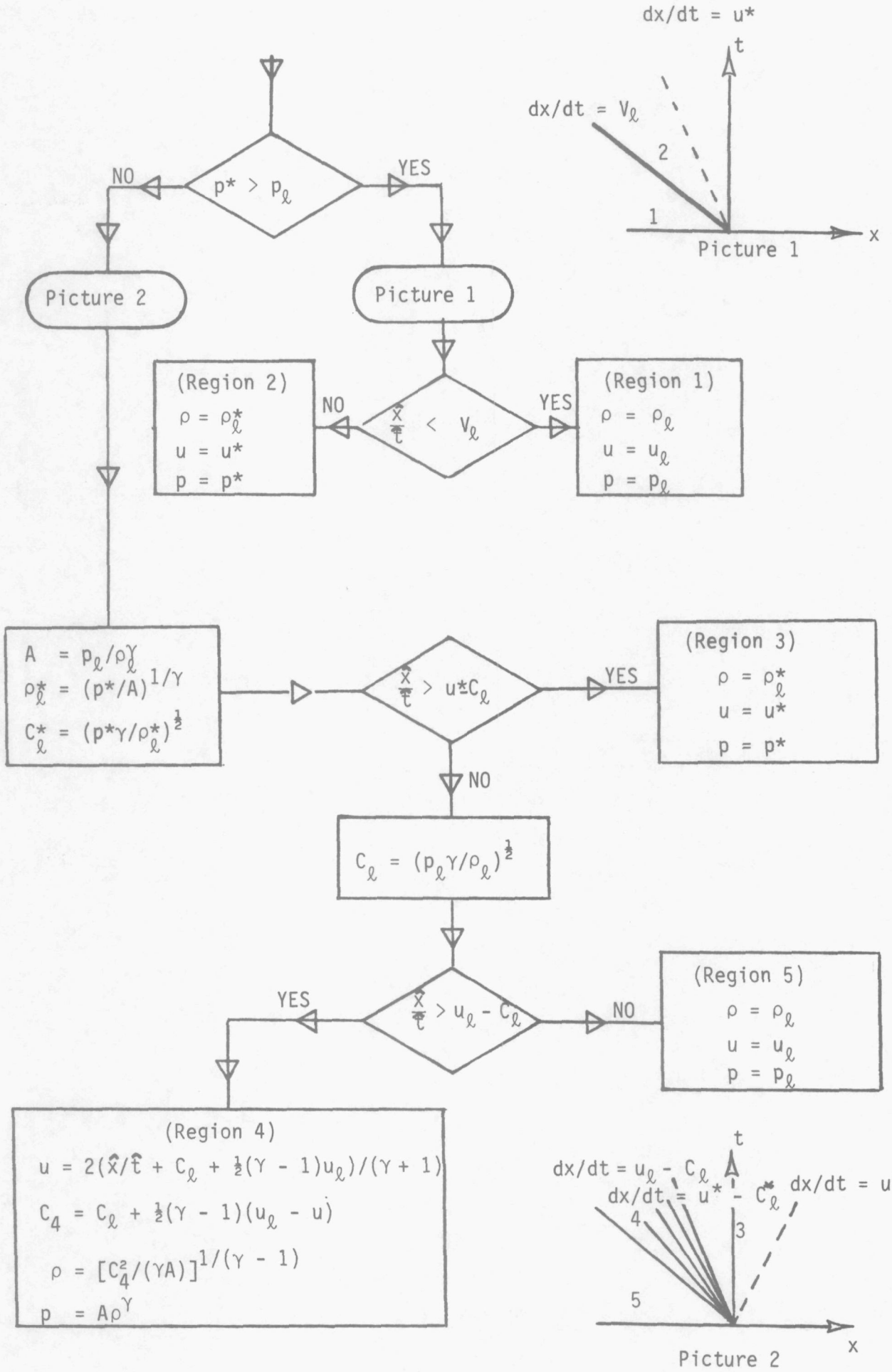


Figure 6 Sampling point $P_i(x,t)$ is on the left of contact surface dx/dt = u*.

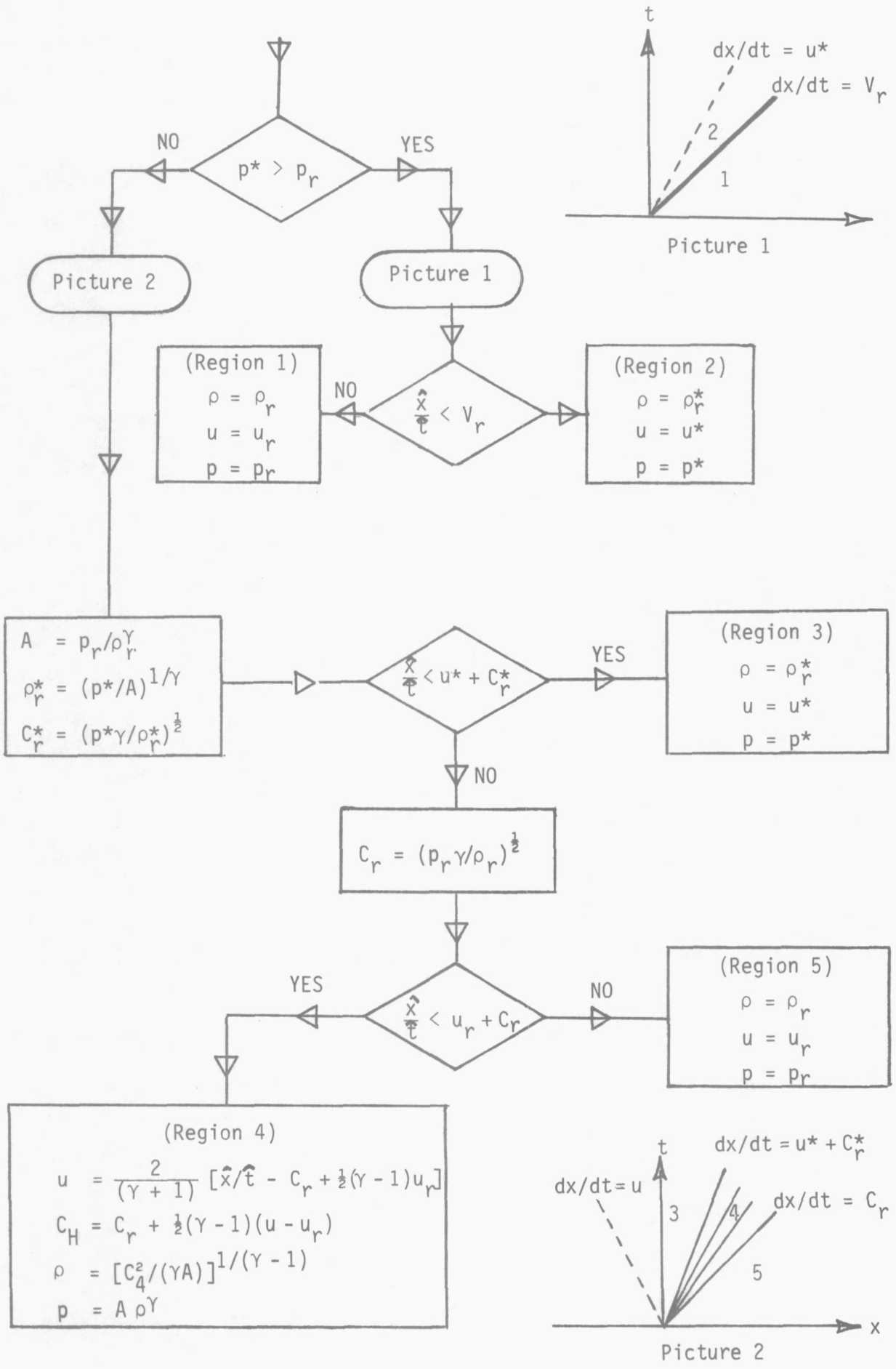
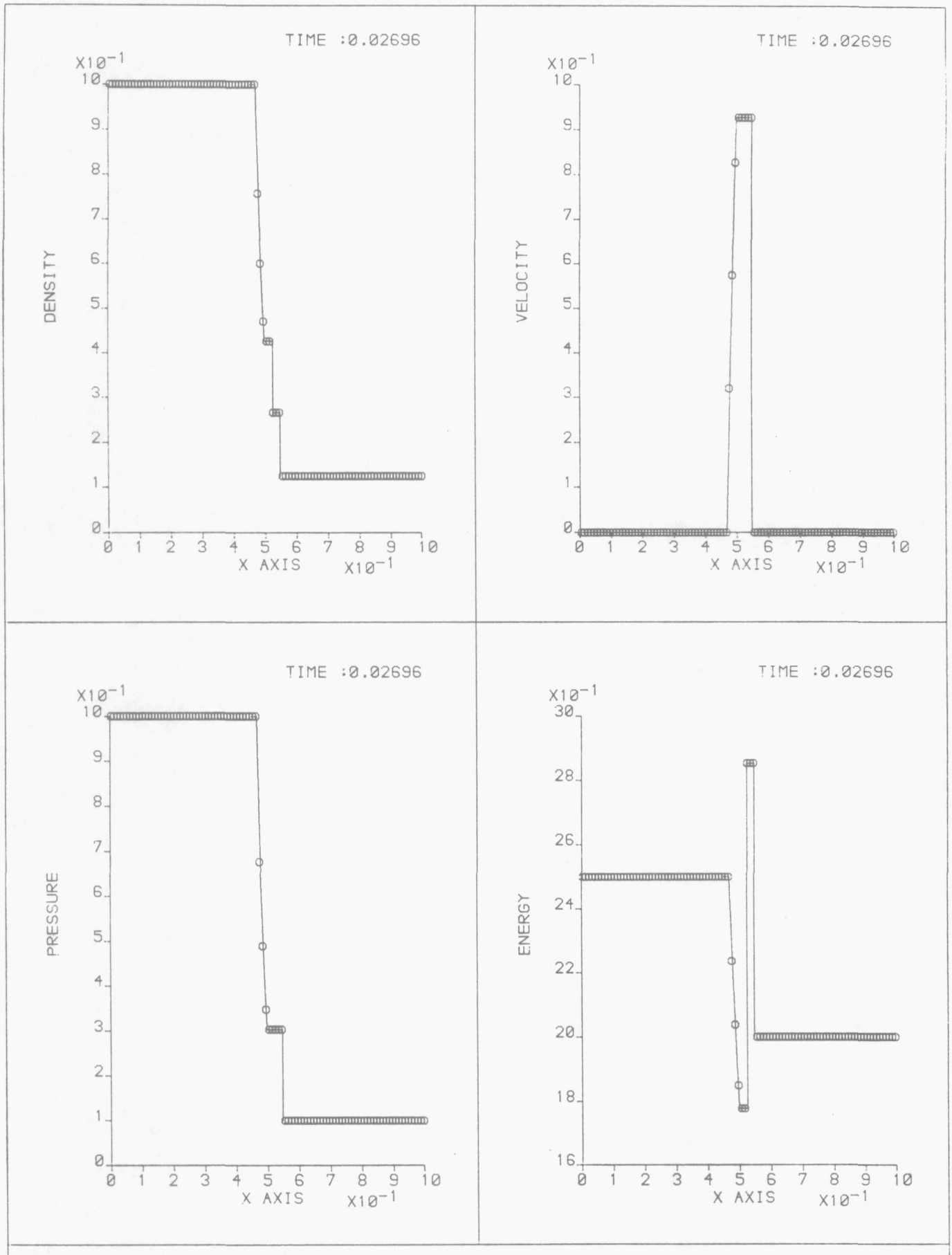
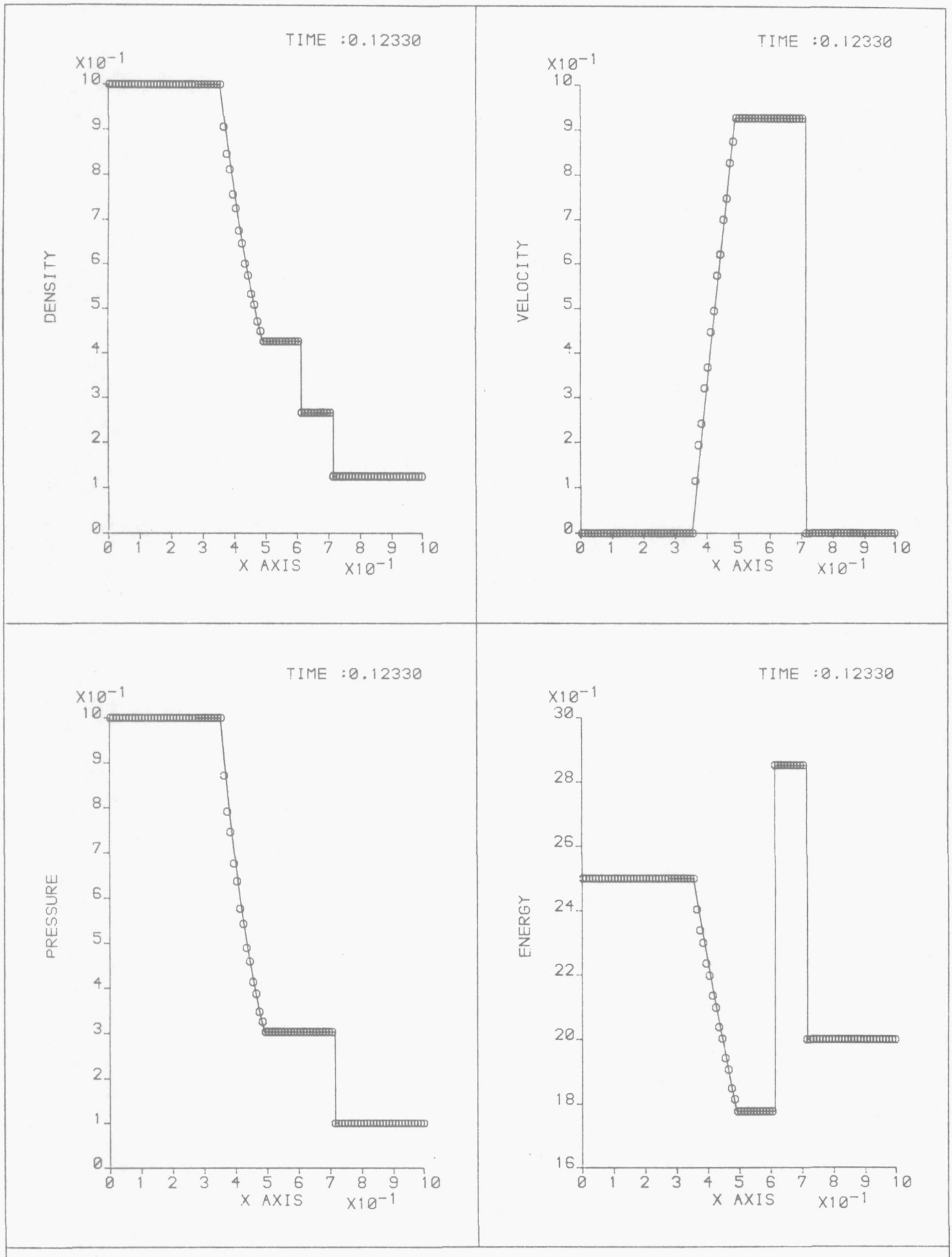


Figure 7 Sampling point $P_i(x,t)$ is on the right of the contact surface dx/dt = u*.



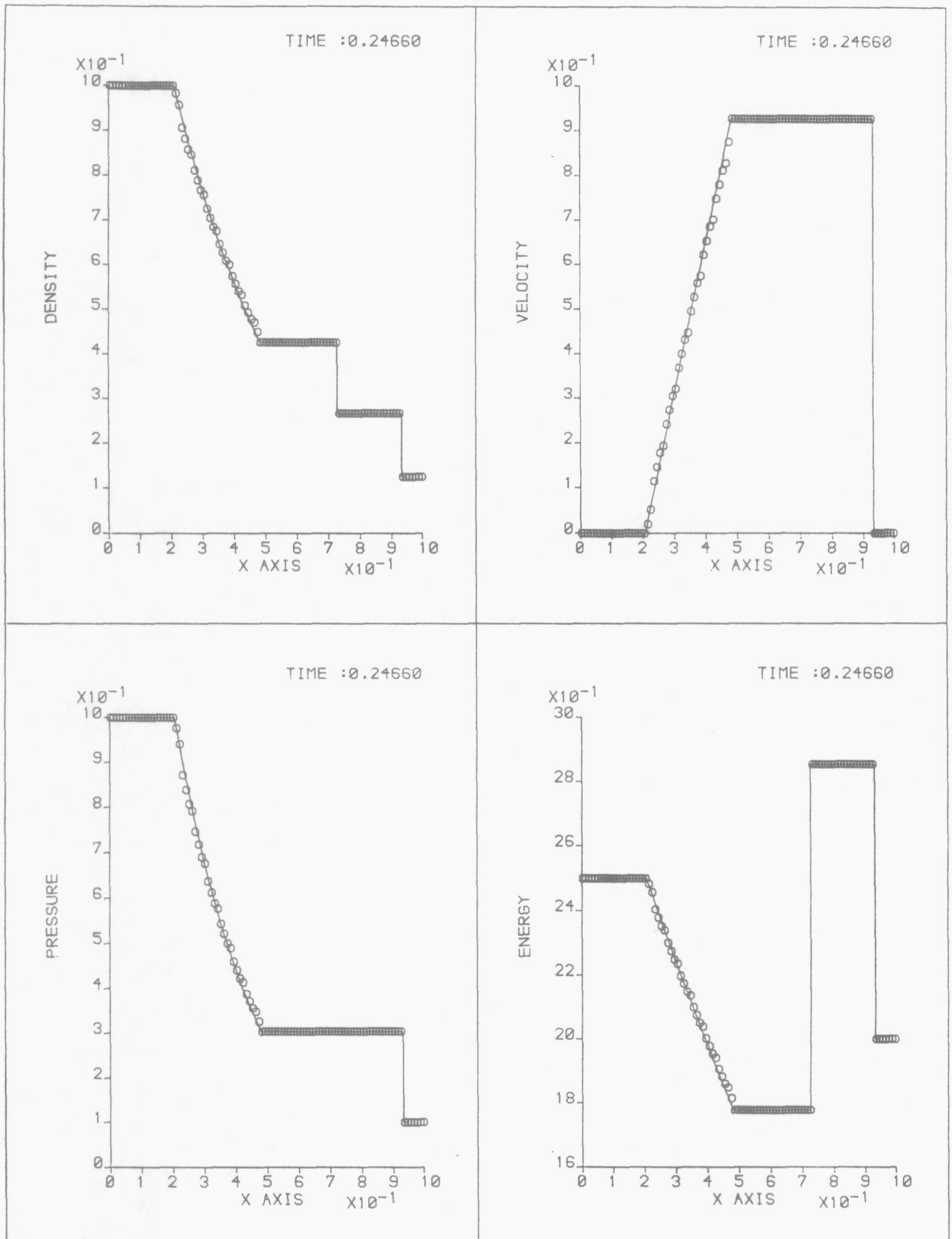
SOD'S SHOCK-TUBE TEST PROBLEM

FIGURE 8 : COMPUTED SOLUTION BY THE RANDOM CHOICE METHOD WITH VDC(2,1) SAMPLING. NUMERICAL (SYMBOL) AND EXACT (FULL LINE).



SOD'S SHOCK-TUBE TEST PROBLEM

FIGURE 9: COMPUTED SOLUTION BY THE RANDOM CHOICE METHOD WITH VDC(2,1) SAMPLING. NUMERICAL (SYMBOL) AND EXACT (FULL LINE).



SOD'S SHOCK-TUBE TEST PROBLEM

FIGURE 10: COMPUTED SOLUTION BY THE RANDOM CHOICE METHOD WITH VDC(2,1) SAMPLING. NUMERICAL (SYMBOL) AND EXACT (FULL LINE).

Appendix

Here we give a listing of a complete RCM program to solve the unsteady Euler equations in one space dimension. There are three main components of the method, namely, solution of Riemann problem, generation of pseudo-random numbers and sampling of the exact solution of the Riemann problem (on a non-staggered grid).

The program is divided into a DRIVER or main program and a set of SUBROUTINES.

DRIVER

There are three one-dimensional arrays of length MD+2 for density D velocity U and pressure P and a further one-dimensional array RN of length 1000 to store the required number of pseudo random numbers. These are generated before commencing time stepping. COMMON blocks CPGAM1 and CPGAM2 contain constants (various combinations of γ) which are also computed at the beginning. STATES contains the left and right constant states for each Riemann problem RP(i, i+1). STARSO contains solution of Riemann problem in region star (u*, p*) and sound speeds for left and right states. GAMTOL contains and TOL (tolerance for iterative solution of Riemann problem).

The following parameters are read in:

M: It defines the spatial discretisation (e.g. M = 100).

NOTIST: Number of time steps (e.g. 60).

NOPROF: Number of solution profiles to be printed out (e.g. 10).

NFREQ: Frequency of output.

TOL: Tolerance for iteration procedure of Riemann problem. e.g. 10^{-4}.

CFLCOE: Coefficient for calculating time step ΔT according to CFL condition ($0 < \text{CFLCOE} \leqslant 1$).

TUBLEN: Length of tube or domain.

GAMMA: γ, ratio of specific heats.

Main loop is 0001 fo time stepping. Loop 0003 solves M+1 Riemann problems and samples solution. There are the following subprograms.

SUBROUTINE SODDATA

It sets up initial data. In the present case data is that for Sod's shock-tube test problem. It also calculates a number of constants to be used during execution of program.

SUBROUTINE VDCK12

It generates NOTIST pseudo-random numbers to be stored in RN( ). The given version uses binary ($k_1 = 2$, $k_2 = 1$) van der Corput numbers. They can be changed (see section 4).

SUBRUTINE CFLOLD

It calculates a time step DTMIN according to the CFL condition that prevents waves from transversing more than half a cell size Δx.

SUBROUTINE CITRPN

It solves the Riemann problem, i.e. it gives u* (us) and p* (ps).

SUBROUTINE SAMPL

It samples solution of Riemann problem. Notice that for each I (loop 0003) we sample either the Riemann problem on the left of cell I (determined by DXDTL) on the Riemann problem on the right of cell I (DXDTR). This depends on the value of POINTER (set to ½ here). POINTER can only be changed if the CFL condition is changed to a condition that allows waves to go beyond one half a cell size.

References

1. Chorin, A. 1976
Random Choice Solution of Hyperbolic Systems.
J. Comp. Phys. 22, 517-536

2. Sod, G.A. 1978
A survey of several finite difference methods for systems of non-linear hyperbolic conservation laws.
J. Comp. Phys. 27, 1-31.

3. Toro E.F. & Clarke J.F. 1985
Application of the Random Choice Method to computing problems of solid propellant combustion in a closed vessel.
CoA Report NFP85/16, November 1985, Cranfield Institute of Technology, Cranfield, Beds, U.K.

4. Dutt, P. 1986
A Riemann solver based on a global existence proof for the Riemann problem.
ICASE Report No. 86-3.
NASA Langley Research Centre.

5. Gottlieb J.J. 1986
Lecture course notes on Random Choice Method for solving one-dimensional unsteady flows in ducts, shock tubes and blast wave simulators.
AC Laboratorium SPiez, Switzerland, May 1986.

6. Dukowicz J.K. 1985
A general non-iterative Riemann solver for Godunov's Method.
J. Comp. Phys. 61, 119-137

7. Glaister P. 1986
An approximate linearised Riemann solver for the Euler equations in one-dimension with a general equation of state.
Numerical Analysis Report 7/86 Mathematics Department, University of Reading,U.K.

8. Collela P. 1982
Glimm's Method for Gas Dynamics.
SIAM J. Sci Stat. Comput. Vol. 3, No. 1 March 1982.

9. Hammersley J.M. & Handscombe D.C., 1964
Monte Carlo Methods
Chapman and Hall.

10. Toro E.F. 1987
Irregular gridding in space and time for the Random Choice Method.
CoA Report, Cranfield Institute of Technology, Cranfield, Beds, U.K.
(in preparation).

11. Toro E.F. 1986
A new numerical technique for quasi-linear hyperbolic systems of conservation laws.
CoA Report 86/20, December 1986, Cranfield Institute of Technology, Cranfield, Beds, U.K.

12. Toro E.F. & Roe P.L.
A hybridised higher-order Random Choice Method for quasi-linear hyperbolic systems.
Proceedings 16th International Symposium on shock tubes and waves July 26-30, 1987, Aachen, W. Germany. (to appear).

13. Roe P.L. 1987
Lecture notes in Computational Fluid Dynamics. MSc Course, 1987, College of Aeronautics, Cranfield, U.K.

14. Glass I.I., Martin W. & Patterson G.N. 1953
A theoretical and experimental study of the shock tube.
Institute of Aerophysics, University of Toronto UTIA Report No. 2.

15. Courant R. & Friedricks K.O.
Supersonic flow and shock waves.
Springer-Verlag, 1985.

```
C       ******************************************************
C       *    MAIN PROGRAM FOR THE RANDOM CHOICE METHOD TO    *
C       *  SOLVE  THE UNSTEADY EULER EUQATIONS IN DIMENSION. *
C       *    THE RIEMANN PROBLEM IS SOLVED EXACTLY AND THE   *
C       *  SOLUTION IS SAMPLED ON A NON-STAGGERED GRID.      *
C       ******************************************************

        PROGRAM DRIVER
        PARAMETER (MD=200)
        DIMENSION D(0:MD+1),U(0:MD+1),P(0:MD+1),RN(1000)
        COMMON /CPGAM1/GP1,GM1,HGP1,HGM1,DGAM,DGP1,DGM1,SGAM,G1,G2
        COMMON /CPGAM2/G3,G4,G5,G6
        COMMON /STATES/DL,UL,PL,DR,UR,PR
        COMMON /STARSO/US,PS,CL,CR
        COMMON /GAMTOL/GAMMA,TOL
        DATA NC,TIME,POINTER/0,0.0,0.5/
        READ(90,*)M,NOTIST,NOPROF,NFREQ,TOL,CFLCOE,TUBLEN,GAMMA
        CALL SODDATA(M,TUBLEN,DX,GAMMA,D,U,P)
        CALL VDCK12(RN,NOTIST)
        DO 0001 N=1,NOTIST
        CALL CFLOLD(GAMMA,M,D,U,P,DX,DTMIN)
        DT=CFLCOE*DTMIN
C       REFLECTING BOUNDARY CONDITIONS APPLIED
        D(0)  =D(1)
        U(0)  =-U(1)
        P(0)  =P(1)
        D(M+1)=D(M)
        U(M+1)=-U(M)
        P(M+1)=P(M)
        TIME=TIME+DT
        RAND=RN(N)
        DTDX=DT/DX
        DXDTL=RAND/DTDX
        DXDTR=(RAND-1.0)/DTDX
           DO 0003 I=1,M
              IF(I.EQ.1)THEN
C                SOLVE RIEMANN PROBLEM AT THE LEFT BOUNDARY
                 DL=D(I-1)
                 UL=U(I-1)
                 PL=P(I-1)
                 DR=D(I)
                 UR=U(I)
                 PR=P(I)
                 CALL CITRPN
              ENDIF
              IF(RAND.LE.POINTER)THEN
                CALL SAMPL(GAMMA,D1,U1,P1,DXDTL)
              ENDIF
C       SOLVE RIEMANN PROBLEM RP(I,I+1)
              DL=D(I)
              UL=U(I)
              PL=P(I)
```

```
                DR=D(I+1)
                UR=U(I+1)
                PR=P(I+1)
                CALL CITRPN
                IF(RAND.GT.POINTER)THEN
                   CALL SAMPL(GAMMA,D1,U1,P1,DXDTR)
                ENDIF
                D(I)=D1
                U(I)=U1
                P(I)=P1
 0003        CONTINUE
           IF(MOD(N,NFREQ).EQ.0)THEN
              NC=NC+1
              CALL OUTPUT(TIME,M,NC,NOPROF,GM1,D,U,P)
           ENDIF
 0001      CONTINUE
           END

C          ************************************************
C          *     DATA FOR SOD'S PROBLEM IS SET UP AND VARIOUS   *
C          *                CONSTANTS ARE CALCULATED.           *
C          ************************************************

           SUBROUTINE SODDATA(M,TUBLEN,DX,GAMMA,D,U,P)
           PARAMETER (MD=200)
           DIMENSION D(0:MD+1),U(0:MD+1),P(0:MD+1)
           COMMON /CPGAM1/GP1,GM1,HGP1,HGM1,DGAM,DGP1,DGM1,SGAM,G1,G2
           COMMON /CPGAM2/G3,G4,G5,G6
           DATA D0,U0,P0/0.125,0.0,0.1/
           GP1=GAMMA+1.0
           GM1=GAMMA-1.0
           HGP1=0.5*GP1
           HGM1=0.5*GM1
           DGAM=1.0/GAMMA
           DGP1=1.0/GP1
           DGM1=1.0/GM1
           SGAM=SQRT(GAMMA)
           G0=SGAM
           G1=HGM1/GAMMA
           G2=HGP1/GAMMA
           G3=1.0/G1
           G4=1.0/HGM1
           G5=2.0/GP1
           G6=GM1/GP1
           DX=TUBLEN/REAL(M)
           IM=M/2
           DO 1000 I=1,M
               IF(I.LE.IM)THEN
                  D(I)=D0*8.0
                  U(I)=U0
                  P(I)=P0*10.0
               ELSE
                  D(I)=D0
                  U(I)=U0
                  P(I)=P0
```

```
          ENDIF
 1000   CONTINUE
        RETURN
        END

C       ********************************************************
C       *    CALCULATION OF MINIMUN TIME STEP SIZE DTMIN.      *
C       ********************************************************

        SUBROUTINE CFLOLD(GAMMA,M,D,U,P,DX,DTMIN)
        PARAMETER (MD=100)
        DIMENSION D(0:MD+1),U(0:MD+1),P(0:MD+1)
        SMAX=0.0
        DO 0001 I=1,M
           A=SQRT(GAMMA*P(I)/D(I))
           SMUA=ABS(U(I))+A
           IF(SMUA.GT.SMAX)SMAX=SMUA
 0001   CONTINUE
        DTMIN=0.5*DX/SMAX
        RETURN
        END

C       ********************************************************
C       * EXACT (ITERATIVE) RIEMANN SOLVER USING THE NEWTON-   *
C       *                   RAPHSON METHOD.                    *
C       ********************************************************

        SUBROUTINE CITRPN
        COMMON /STATES/DL,UL,PL,DR,UR,PR
        COMMON /STARSO/US,PS,CL,CR
        COMMON /GAMTOL/GAMMA,TOL
        COMMON /CPGAM1/GP1,GM1,HGP1,HGM1,DGAM,DGP1,DGM1,SGAM,G1,G2
        COMMON /CPGAM2/G3,G4,G5,G6
        CL    =SQRT(GAMMA*PL/DL)
        CR    =SQRT(GAMMA*PR/DR)
        DELU=UL-UR
C       GUESSED VALUE FOR PS IS PROVIDED
        CLPLG=CL/PL**G1
        CRPRG=CR/PR**G1
        PS    =((CL+CR+HGM1*DELU)/(CLPLG+CRPRG))**G3
C       WRITE(6,*)PS
        PS0   =PS
C       START ITERATION
        DO 0001 IT=1,50
C          LEFT WAVE
           IF(PL.LT.PS)THEN
              S1=SQRT(G5/DL)
              S2=G6*PL
              S2PS=S2+PS
              DELPLPS=PL-PS
              SQS2PS=1.0/SQRT(S2PS)
              FLEFVAL=S1*DELPLPS*SQS2PS
              FLEFDER=-S1*SQS2PS*(1.0+0.5*DELPLPS/S2PS)
           ELSE
              FLEFVAL=G4*(CL-CLPLG*PS**G1)
```

```
            FLEFDER=-DGAM*CLPLG*PS**(-G2)
         ENDIF
C        RIGHT WAVE
         IF(PR.LT.PS)THEN
            S1=SQRT(G5/DR)
            S2=G6*PR
            S2PS=S2+PS
            DELPRPS=PR-PS
            SQS2PS=1.0/SQRT(S2PS)
            FRIGVAL=S1*DELPRPS*SQS2PS
            FRIGDER=-S1*SQS2PS*(1.0+0.5*DELPRPS/S2PS)
         ELSE
            FRIGVAL=G4*(CR-CRPRG*PS**G1)
            FRIGDER=-DGAM*CRPRG*PS**(-G2)
         ENDIF
         FUNVAL=FLEFVAL+FRIGVAL+DELU
         FUNDER=FLEFDER+FRIGDER
         PS      =PS-FUNVAL/FUNDER
         TESTPS  =ABS(PS-PS0)
         IF(TESTPS.LE.TOL)GOTO 0002
         IF(PS.LT.TOL)PS=TOL
         PS0=PS
 0001    CONTINUE
 0002    CONTINUE
         US=0.5*(FLEFVAL-FRIGVAL+UL+UR)
         RETURN
         END

C        ********************************************************
C        *   RANDOM SAMPLING OF SOLUTION OF RIEMANN PROBLEM     *
C        ********************************************************

         SUBROUTINE SAMPL(GAMMA,D,U,P,DXDT)
         COMMON/STATES/DL,UL,PL,DR,UR,PR
         COMMON/STARSO/US,PS,CL,CR
         COMMON/CPGAM1/GP1,GM1,HGP1,HGM1,DGAM,DGP1,DGM1,SGAM,G1,G2
         SAVE /CPGAM1/
         IF(DXDT.GE.US)THEN
C        SAMPLING POINT LIES TO THE RIGHT OF SLIP LINE
            IF(PS.LE.PR)THEN
C           RIGHT WAVE IS A RAREFACTION WAVE
               IF(DXDT.LT.(UR+CR))THEN
                  AISEN=PR/DR**GAMMA
                  D3=(PS/AISEN)**DGAM
                  C3=SQRT(GAMMA*PS/D3)
                  IF(DXDT.LT.(US+C3))THEN
C                 LEFT OF RIGHT RAREFACTION
                     D=D3
                     U=US
                     P=PS
                  ELSE
C                 INSIDE RIGHT RAREFACTION
                     U=2.0*(DXDT-CR+HGM1*UR)/GP1
                     C4=CR+HGM1*(U-UR)
                     D=(C4*C4/(AISEN*GAMMA))**DGM1
```

```
                  P=AISEN*D**GAMMA
               ENDIF
            ELSE
C           RIGHT OF RIGHT RAREFACTION
               D=DR
               U=UR
               P=PR
            ENDIF
         ELSE
C        RIGHT WAVE IS A SHOCK WAVE
            SMR2=1.0+G2*(PS/PR-1.0)
            SMR=SQRT(SMR2)
            URS=UR+CR*SMR
            IF(DXDT.GE.URS)THEN
C           RIGHT OF RIGHT SHOCK
               D=DR
               U=UR
               P=PR
            ELSE
C           BEHIND RIGHT SHOCK
               D=GP1*DR*SMR2/(GM1*SMR2+2.0)
               U=US
               P=PS
            ENDIF
         ENDIF
      ELSE
C     SAMPLING POINT LIES TO THE LEFT OF SLIP LINE
         IF(PS.LE.PL)THEN
C           LEFT WAVE IS A RAREFACTION
            AISEN=PL/DL**GAMMA
            D3=(PS/AISEN)**DGAM
            C3=SQRT(GAMMA*PS/D3)
            IF(DXDT.LT.(US-C3))THEN
               IF(DXDT.LT.(UL-CL))THEN
C              LEFT OF LEFT RAREFACTION
                  D=DL
                  U=UL
                  P=PL
               ELSE
C              INSIDE LEFT RAREFACTION
                  U=2.0*(DXDT+CL+HGM1*UL)/GP1
                  C4=CL+HGM1*(UL-U)
                  D=(C4*C4/(GAMMA*AISEN))**DGM1
                  P=AISEN*D**GAMMA
               ENDIF
            ELSE
C           RIGHT OF LEFT RAREFACTION
                  D=D3
                  U=US
                  P=PS
            ENDIF
         ELSE
C           LEFT WAVE IS A SHOCK WAVE
            SML2=1.0+G2*(PS/PL-1.0)
            SML=-SQRT(SML2)
```

```
            ULS=UL+CL*SML
            IF(DXDT.GE.ULS)THEN
C           BEHIND LEFT SHOCK
               D=GP1*DL*SML2/(GM1*SML2+2.0)
               U=US
               P=PS
            ELSE
C           LEFT OF LEFT SHOCK
               D=DL
               U=UL
               P=PL
            ENDIF
         ENDIF
      ENDIF
      RETURN
      END

C     ********************************************************
C     *    GENERATE VAN DER CORPUT PSEUDO RANDOM NUMBERS     *
C     ********************************************************

      SUBROUTINE VDCK12(RN,NOTIST)
      PARAMETER (N1=100,N2=1000)
      DIMENSION NA(N1),JA(N1),RN(N2)
      DATA K1,K2,NRN0/2,1,100/
      DO 0001 NRN=NRN0,NOTIST+NRN0
         IS=0
         MM=NRN
         DO 0002 I=1,100
            IF(MM.EQ.0)GOTO 8888
            IS=IS+1
            NA(I)=MOD(MM,K1)
            MM=MM/K1
            KL=K2*NA(I)
            JA(I)=MOD(KL,K1)
0002     CONTINUE
8888     RANNUM=0.0
         DO 0004 K=1,IS
            RANNUM=RANNUM+REAL(JA(K))/(K1**K)
0004     CONTINUE
         NT=NRN-NRN0+1
         RN(NT)=RANNUM
0001  CONTINUE
      RETURN
      END

C     ********************************************************
C     *    PRINT RESULTS FOR DENSITY, PRESSURE, VELOCITY,    *
C     *     AND INTERNAL ENERGY TO FILES 1, 2, 3, AND 4.     *
C     ********************************************************

      SUBROUTINE OUTPUT(TIME,M,NC,NOPROF,GM1,D,U,P)
      PARAMETER (MD=200)
      DIMENSION D(0:MD+1),U(0:MD+1),P(0:MD+1)
      DIMENSION TM(20),R1(4,20,MD)
```

```
      TM(NC)=TIME
         DO 0001 I=1,M
             R1(1,NC,I)=D(I)
             R1(2,NC,I)=U(I)
             R1(3,NC,I)=P(I)
             R1(4,NC,I)=P(I)/D(I)/GM1
0001     CONTINUE
         IF(NC.EQ.NOPROF)THEN
            WRITE(1,0004)(TM(J),J=1,NOPROF)
            WRITE(2,0004)(TM(J),J=1,NOPROF)
            WRITE(3,0004)(TM(J),J=1,NOPROF)
            WRITE(4,0004)(TM(J),J=1,NOPROF)
            DO 0002 I=1,M
                WRITE(1,0003)I,(R1(1,J,I),J=1,NOPROF)
                WRITE(2,0003)I,(R1(2,J,I),J=1,NOPROF)
                WRITE(3,0003)I,(R1(3,J,I),J=1,NOPROF)
                WRITE(4,0003)I,(R1(4,J,I),J=1,NOPROF)
0002        CONTINUE
            NC=0
         ENDIF
0003   FORMAT(I4,1X,10(F6.4,2X))
0004   FORMAT(5X,10(F7.5,1X))
       RETURN
       END
```