

KfK 4154
Dezember 1986

CIP **A New Numerical Solver for General Nonlinear Hyperbolic Equations in Multi-Dimension**

T. Yabe, H. Takewaki
Institut für Neutronenphysik und Reaktortechnik

Kernforschungszentrum Karlsruhe

KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Neutronenphysik und Reaktortechnik

KfK 4154

CIP - A New Numerical Solver for General
Nonlinear Hyperbolic Equations in Multi-Dimension

Takashi YABE^{a)} and Hideaki TAKEWAKI^{b)}

a) On leave from
Institute of Laser Engineering,
Osaka University, Suita, Osaka 565, Japan

b) Permanent address:
FUJITSU LIMITED, Shinkamata, Ota-ku, Tokyo, Japan

Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript vervielfältigt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Abstract

A new method CIP (Cubic-Interpolated Pseudo-particle) to solve hyperbolic equations is proposed. The method gives a stable and less diffusive result for square wave propagation compared with FCT (Flux-Corrected Transport) and a better result for propagation of a sine wave with a discontinuity. The scheme is extended to nonlinear and multi-dimensional problems.

CIP - eine neue Lösungsmethode für allgemeine nichtlineare hyperbolische Gleichungen in mehreren Dimensionen

Zusammenfassung

Es wird eine neue Methode CIP (Cubic-Interpolated Pseudo-particle) zur Lösung hyperbolischer Gleichungen vorgeschlagen. Mit dieser Methode erhält man stabile und weniger verwaschene Lösungen für die Propagation einer Rechteck-Welle und einer Sinus-Welle mit Unstetigkeit als mit der FCT (Flux-Corrected Transport) Methode. Die Methode wird in dieser Arbeit auch auf nichtlineare und mehrdimensionale Probleme erweitert.

Table of Contents

I. INTRODUCTION -----	1
II. BASIC PRINCIPLE OF CIP-----	3
III. GENERALIZED ONE-DIMENSIONAL CIP -----	10
IV. SHOCK WAVE PROPAGATION IN PLANE GEOMETRY -----	13
V. MULTIDIMENSIONAL CIP -----	18
VI. CONCLUSION -----	21
REFERENCES -----	22
FIGURE CAPTIONS -----	23
APPENDIX -----	25
TABLES -----	28
FIGURES -----	29
ACKNOWLEDGEMENTS -----	41
COMPUTER PROGRAM FOR ONE-DIMENSIONAL HYDRODYNAMIC EQUATIONS. -----	42

I. INTRODUCTION

In a previous paper, we proposed a less-diffusive and stable algorithm, CIP (Cubic-Interpolated Pseudo-Particle) [1], for solving a linear hyperbolic equation. In the scheme, a quantity within a mesh is interpolated by a cubic polynomial. Since the gradient of the quantity is a free parameter, the scheme is completely different from the spline method [2,3] ; in the latter method, the gradient is calculated by assuming continuity of the quantity, the first derivative and the second derivative of the quantity at the mesh boundaries, and is independent of the equation. In contrast, the first derivative in the CIP is calculated by the model equation for the spatial derivative and is chosen so that it is consistent with the time evolution of the given equation.

Since the CIP does not use the 'limiting' procedure employed in the FCT[4], it does not suffer from a fictitious clipping in a triangular wave. This was demonstrated in propagation of a sine wave with a discontinuity in the previous paper[1].

The present paper gives generalization of the CIP scheme to nonlinear problems with non-advective terms and to multi-dimensional problems. In section II, the basic principle of the CIP is briefly explained from a different viewpoint from the previous paper[1]. In section III, the CIP is extended to nonlinear problems. In section IV, one-dimensional shock tube problems in plane geometry are

used for test runs. The CIP and various other schemes are compared with the analytic solution. In section V, the scheme is extended to a two-dimensional configuration, with a few numerical examples. The CIP scheme is straightforwardly extended to multi-dimensional problems by time splitting.

II. BASIC PRINCIPLE OF CIP

In this section, the basic principle of the CIP algorithm is briefly explained by comparing it with other schemes. For this purpose, let us use a simple hyperbolic equation ;

$$\partial f / \partial t + c \partial f / \partial x = 0, \quad (1)$$

where c is constant. Equation (1) is integrated over an interval $(x_{i-1/2}, x_{i+1/2}), (t_n, t_{n+1})$ to obtain

$$\int_{x_{i-1/2}}^{x_{i+1/2}} [f^{n+1}(x) - f^n(x)] dx = -\Delta F_{i+1/2} + \Delta F_{i-1/2}, \quad (2)$$

$$\Delta F_{i+1/2} = \int_{x_{i+1/2} - c\Delta t}^{x_{i+1/2}} f^n(x) dx, \quad (2')$$

where the superscript n and the subscript i on f mean the value of f at $t=t_n$ and $x=x_i$, respectively. In the right hand side of Eq.(2), the profile of f is assumed to remain rigid during the interval $\Delta t (= t_{n+1} - t_n)$ and to propagate by a distance $c\Delta t$ [1].

It is interesting to examine the assumptions made on the spatial profile of $f(x)$ for the integration of Eq.(2) in various schemes. In an example given in Fig.1, let the solid line be an actual profile of a wave, the dashed line be an employed profile in a scheme given below, and the shaded areas be ΔF . At first, let us begin with a

centered difference scheme. The assumptions made are as follows ;

1) left hand side : $f(x)$ is constant and $\int f dx = f_i \Delta x$

2) right hand side : although $f(x)$ is linearly interpolated as shown by the dashed line in Fig.1(a), only the values at $x = x_{i\pm 1/2}$ are used to estimate ΔF ; that is, the incoming and outgoing fluxes ΔF are given by the shaded rectangular areas and hence $\Delta F_{i\pm 1/2} = f_{i\pm 1/2} c \Delta t = (f_i + f_{i\pm 1}) c \Delta t / 2$.

This leads to

$$f_i^{n+1} - f_i^n = -(\kappa/2)(f_{i+1} - f_{i-1}), \quad (3)$$

κ being $c \Delta t / \Delta x$.

The second example is the Lax-Wendroff scheme[5], where the assumptions employed are as follows ;

1) left hand side : the same as the centered difference.

2) right hand side : $f(x)$ is linearly interpolated as shown by the dashed line and the fluxes ΔF are given by the shaded areas in Fig.1(b) and hence $\Delta F_{i\pm 1/2} = (f_i + f_{i\pm 1}) c \Delta t / 2 \pm (f_i - f_{i\pm 1}) c^2 \Delta t^2 / 2$.

This leads to

$$f_i^{n+1} - f_i^n = -\kappa/2 \cdot (f_{i+1} - f_{i-1}) + \kappa^2/2 \cdot (f_{i+1} - 2f_i + f_{i-1}), \quad (4)$$

which can be modified into a two-step form as in the two-step Lax-Wendroff scheme.

The above derivation clearly shows the assumptions made in the schemes ; it is interesting to see that the Lax-Wendroff scheme is only a slight modification of the centered difference as seen from the comparison between Figs.1(a) and 1(b).

Other than those schemes, there exist a number of schemes available at present and shown to be useful. In Fig.2, some of those schemes are introduced and tested with a typical problem, square wave propagation with $\kappa=0.2$; the profiles are shown after 1000 steps. In the leap-frog and the Lax-Wendroff schemes, overshooting and the so-called phase error are significant. Let us consider the cause of this error. In both schemes, although the spatial profile of f is linearly interpolated and ΔF is calculated based on the profile shown in Fig.1, the profile on the left hand side of Eq.(2) is assumed to be flat. If $\Delta F_{i+1/2}$ is less than $\Delta F_{i-1/2}$ for $c>0$, the value f_i in Fig.1 should always increase according to Eq.(2) and overshooting should occur ; the spatial profile of f on the left hand side of Eq.(2) is constant ($f=f_i$) and hence $\Delta F_{i-1/2} - \Delta F_{i+1/2}$ (>0) is used to increase f_i uniformly.

In contrast, if the spatial profile of f on the left hand side of Eq.(2) is correctly described, overhoot does not occur as illustrated in Fig.3 ; the solid and dashed lines show the profiles of a wave at $t=t_n$ and $t_{n+1}=t_n + c\Delta t$, respectively. Outgoing and incoming fluxes (Eq.(2')) are given by the shaded areas C

and A plus C' , respectively; C' is the same as C . If the real profiles are used in the integration on the left hand side of Eq.(2), Eq.(2) means that both the area A' and the difference between the outgoing and incoming fluxes, which is given by the area A , fill the area B without causing any overshoot. Consequently, it is essential to predict correctly the profile within a mesh at an advanced time $t=t_{n+1}$. In the PIC[10,11], a number of particles in a line represent a spatial profile of f . Then, time evolution of the spatial profile, which is only a spatial translation by $c\Delta t$ in the present case, is well described by the particles' movement; the spatial distribution of the particles after they move gives the spatial profile of f at the next time ($t=t_{n+1}$). If it is possible to imitate this movement of many particles by a simple function, it will provide a stable and less-diffusive scheme.

In the CIP method, the spatial profile of f within a mesh is approximated by a cubic polynomial. Then the integration on the right hand side of Eq.(2) is the same as that due to a particle's translation in space by $c\Delta t$. At the next time step (t_{n+1}), that is, after the profile moves, a new polynomial within a mesh is required in order to calculate the term on the left hand side of Eq.(2). Since the boundaries of an old polynomial shift by $c\Delta t$ from the cell boundaries, the polynomial at $t=t_n$ cannot be used as the profile of the quantity within a mesh at $t=t_{n+1}$. Hence, a new cubic polynomial must be generated to approximate a new profile as well as possible.

In ordinary cubic-spline interpolation, continuity of the value f , the first derivative f' , and the second derivative f'' is required to generate the piecewise cubic polynomials from the data given at some discrete points. This procedure is not suitable for the present problem, because the profile generated by the method is not consistent with the given equation. Even in recent work on spline interpolation, this physical requirement was neglected and the efforts were limited to constructing well-posed and monotonic splines [2,3]. The CIP method relaxes the requirements on continuity, that is, requires continuity only of f and f' at the mesh boundaries. Then f and f' at some discrete points are used to generate the piecewise cubic polynomials. The additional free parameter f' offers a tool to approximate the particle's movement as mentioned in the previous paragraph. This can be done by calculating the time evolution of f' by the equation

$$\partial f' / \partial t + c \partial f' / \partial x = 0, \quad (1')$$

as well as that of f by Eq.(1). In the previous paper[1], Eq.(1') was not solved by the difference method but it was proposed to use the first derivative of a cubic polynomial $f^n(x)$ at $x = x_i - c\Delta t$ for f'^{n+1} ; this corresponds to the translation of the profile ($f^n(x)$) by $c\Delta t$.

Here, we summarize the finite difference form of the CIP as follows :

$$H_i^{n+1} + G_i^{n+1} = H_i^n + G_i^n - (\Delta F_{i+1/2} - \Delta F_{i-1/2}) / \Delta x, \quad (5a)$$

$$H_i = 1/192(18f_{i+1} + 156f_i + 18f_{i-1}), \quad (5b)$$

$$G_i = 5/192(f'_{i-1} - f'_{i+1})\Delta x, \quad (5c)$$

$$\begin{aligned} \Delta F_{i+1/2} = & (-\kappa/8 + \kappa^2/8 + \kappa^3/6 - \kappa^4/4)f'_{i+1}{}^n \Delta x^2 \\ & + (\kappa/8 + \kappa^2/8 - \kappa^3/6 - \kappa^4/4)f'_i{}^n \Delta x^2 \\ & + (\kappa/2 - 3\kappa^2/4 + \kappa^4/2)f'_{i+1}{}^n \Delta x \\ & + (\kappa/2 + 3\kappa^2/4 - \kappa^4/2)f'_i{}^n \Delta x, \end{aligned} \quad (5d)$$

When all f_i^n and $f'_i{}^n$ are given, the value of the right hand side of Eq.(5a) is calculated. However, both f^{n+1} and f'^{n+1} cannot be calculated at once from this equation alone. Thus f'^{n+1} is determined by Eq.(1') as

$$f'_i{}^{n+1} = f'(x_i - c\Delta t, t_n). \quad (6)$$

Hence the equation which must be solved is the tridiagonal matrix for f_{i+1} , f_i , and f_{i-1} at $t = t_{n+1}$ in Eq.(5a). Since the solution Eq.(6) is only an approximate one, it will depart from the real solution after a long time and hence recorrection is required to

adjust it to the value f_i ; in the previous paper[1], this recorection was done once every 50 steps.

Before closing this brief review, we must make an important comment in solving Eq.(5a). Since tridiagonal inversion is slow when it is employed in vector machines, it should be avoided in the code. This can be done in the CIP. H_i^{n+1} in Eq.(5a) is re-written as

$$H_i^{n+1} = f_i^{n+1} + \tilde{H}_i(n), \quad (7a)$$

$$\tilde{H}_i(n) = (3/32)(\tilde{f}_{i+1} - 2\tilde{f}_i + \tilde{f}_{i-1}). \quad (7b)$$

Since \tilde{H} is smaller by a factor of 3/32 than f , a rough approximation can be used in estimating \tilde{H} . Thus, the quantities having \sim are calculated by shifting as in Eq.(6) because Eq.(1) is the same as Eq.(1') except for the difference of f and f' . According to this equation,

$$\tilde{f}_i = f(x_i - c\Delta t, t_n). \quad (8)$$

Consequently, the equation to be solved is

$$f_i^{n+1} = H_i^n - \tilde{H}_i(n) + G_i^n - G_i^{n+1} - (\Delta F_{i+1/2} - \Delta F_{i-1/2})/\Delta x, \quad (9)$$

instead of Eq.(5a). We found that Eq.(9) gives the same result as Eq.(5a) and it is used in Fig.2.

III. GENERALIZED ONE-DIMENSIONAL CIP

A large class of equations can be written in the form

$$\partial f / \partial t + \partial(fu) / \partial x = g. \quad (10)$$

Here f and g can be vector quantities. The inhomogeneous term g includes, for example, the pressure work in the hydrodynamic equation and the heat conduction in the energy equation and so forth. Furthermore, u is the advection velocity and is sometimes written in terms of f and hence Eq.(10) becomes a nonlinear equation.

In this section, we extend the CIP algorithm to the equation given in Eq.(10). The algorithm is divided into two phases, the Eulerian and the Lagrangian phases as in the PIC and SOAP codes.

1) Eulerian Phase

Equation (10) is solved without the advection term

$$f^*_i = f_i^n + g_i^n \Delta t. \quad (11)$$

The CIP algorithm needs information about the first spatial derivative of f . In the previous paper[1], we proposed to use the equation

$$\partial f' / \partial t = \partial g / \partial x, \quad (12)$$

for this purpose. We do not need to use an elaborate difference scheme in equation because of the factor $5/192$ in Eq.(5c) and hence Eq.(12) may be solved with a centered

finite difference. Although Eq.(12) works well, another simple procedure is preferable for general purposes because $\partial g/\partial x$ becomes a third order derivative if g is the heat conduction term and is written, for example, as $\partial^2 f/\partial x^2$. Here, we propose to use

$$f'_i{}^* = f'_i{}^n + (f_{i+1}{}^* - f_{i-1}{}^* - f_{i+1}{}^n + f_{i-1}{}^n)/2\Delta x, \quad (13)$$

Although this procedure is the same as Eq.(12), the procedure is greatly simplified ; only the values f^* given by Eq.(11) are used. Furthermore, the procedure offers another advantage. In the coupled hyperbolic-parabolic equation

$$\partial f/\partial t + \partial fu/\partial x = D\partial^2/\partial x^2, \quad (14)$$

the spatial derivative f' can be advanced in time by Eq.(13) after the non-advective term is solved implicitly as

$$f_i{}^* = f_i{}^n + D\Delta t/\Delta x^2 (f_{i+1}{}^* - 2f_i{}^* + f_{i-1}{}^*). \quad (15)$$

Thus the CIP can be used even if $D\Delta t/\Delta x^2 \gg 1$.

2) Lagrangian Phase

In this phase, the advection term is calculated by the CIP algorithm as given in Eq.(5c),(5d),(6),(7b),(8),and (9), but $\kappa = \kappa_i = u_{i+1/2}\Delta t/\Delta x$ is now space-dependent, and f^n , f'^n in Eq.(9) are

replaced by f^* , f'^* . The shifting process as in Eqs.(6) and (8) can be done by the spatial translation of the cubic-interpolated profile $f^*(x)$ by $u_i \Delta t$ as follows ;

$$\tilde{f}_i = a_i X^3 + b_i X^2 + f'_{i-1}{}^* X + f_{i-1}{}^*, \quad (16)$$

$$f'_{i}{}^{n+1} = 3a_i X^2 + 2b_i X + f'_{i-1}{}^*, \quad (17)$$

where

$$X = \Delta x - u_i \Delta t, \quad (18)$$

$$a_i = (f'_{i}{}^* + f'_{i-1}{}^*) / \Delta x^2 - 2(f_{i}{}^* - f_{i-1}{}^*) / \Delta x^3, \quad (19)$$

$$b_i = 3(f_{i}{}^* - f_{i-1}{}^*) / \Delta x^2 - (f'_{i}{}^* + 2f'_{i-1}{}^*) / \Delta x. \quad (20)$$

IV. SHOCK WAVE PROPAGATION IN PLANE GEOMETRY

In this section, the CIP algorithm is applied to the hydrodynamic equations ;

$$\partial \rho / \partial t + \partial \rho u / \partial r = - \delta \rho u / r, \quad (21a)$$

$$\partial m / \partial t + \partial m u / \partial r = - \partial p / \partial r - \delta m u / r, \quad (21b)$$

$$\partial E / \partial t + \partial E u / \partial r = - \partial p u / \partial r - \delta (E+p) u / r, \quad (21c)$$

$$p = (\gamma - 1) (E - m u / 2), \quad (21d)$$

where ρ , m , E , u , and p are the density, momentum, total energy, velocity, and pressure, respectively, and γ is the specific heat ratio. These equations can be used for plane, cylindrical, and spherical geometries by letting δ be 0, 1, and 2, respectively. In this section, the problem is limited to a plane geometry.

The scheme proposed in section III is checked by using the following test problem : the initial condition is $\rho=1$, $E=1$ for the leftmost 100 zones and $\rho=0.5$, $E=0$ for the other zones ; $u=0$ and $\gamma=5/3$ in all zones. The solution of this problem is given in Appendix. As is well known, overtaking occurs at a shock front if there is no dissipation term in Eqs.(21). In the discussion of such dissipation, we should distinguish the nonlinear problem from the linear one. In the linear problem the CIP

algorithm does not need any dissipation even at a steep gradient region. This is due to the fact that the CIP scheme tries to correctly solve the given equation. The situation becomes different in the nonlinear problem ; the dissipation term is physically required in order to obtain a single-valued function and all schemes use it whether it is explicit or implicit.

The three major forms of explicit numerical dissipation are the artificial viscosity, the "FCT"[4] and the "hybrid"[13] schemes. The limiter in the FCT sometimes leads to an incorrect result as pointed out in the review by McRae et al.[14]. The hybrid scheme uses a diffusive scheme at the shock front and a higher order scheme at other regions. In this paper, all three dissipation methods are examined.

(i) Artificial Viscosity

An artificial viscosity of the form[12]

$$Q = Q_1 + Q_2, \quad (22)$$

is used in the pressure term in Eqs.(21b) and (21c), where

$$Q_1 = \begin{cases} - a\rho C_s (\partial u / \partial x) \Delta x, & \text{if } \partial u / \partial x < 0, \\ 0, & \text{otherwise,} \end{cases} \quad (23a)$$

$$Q_2 = \begin{cases} - a\rho (\partial u / \partial x)^2 \Delta x^2, & \text{if } \partial u / \partial x < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (23b)$$

Here C_s means the sound velocity and a is an adjustable

parameter of order unity. The result with the CIP method after 800 time steps is shown in Fig.4(a). Here, a is set to 1.0 and Δt is fixed to 0.1 with $\Delta x=1.0$. Slight overshoots still remain behind the shock front and the contact surface. For comparison, an analytical solution is shown by the dashed line in the figure. It should be noted that the value $a=1.0$ is quite small compared with that used in other schemes ; in Fig.4(d) the result obtained with the MacCormack scheme[15] is shown, a being the same as in Fig.4(a).

(ii) Hybrid Scheme

For a flux ΔF in Eq.(9), the upstream flux

$$\Delta F_{i+1/2} = \begin{cases} \kappa f_i^* & , \text{ if } u_{i+1/2} > 0 \\ \kappa f_{i+1}^* & , \text{ otherwise} \end{cases} \quad (24)$$

is used instead of the flux (Eq.(5d)) in the region where

$$\partial u / \partial x < 0, \quad u \partial p / \partial x < 0, \quad (25)$$

is satisfied. Equation (25) only tracks the shock front well. It should be noticed that only ΔF is changed while the other CIP algorithm is not changed. The result with this method is shown in Fig.4(b). Compared with Fig.4(a), the contact discontinuity becomes slightly diffusive and the oscillation occurs in the area where the velocity is constant ahead of the expansion region. This oscillation

is due to incorrect switching of Eq.(24) and is improved if $(u_{i+1}-u_i)/|u_{i+1}+u_i| < 0.05$ is used instead of $(u_{i+1}-u_i) < 0$ ($\partial u/\partial x < 0$). The diffusion at the contact discontinuity is due to diffusion by Eq.(24) in the initial phase. However, the profile at the shock front is better than that in Fig.4(a). As shown in Fig.4(e), the upwind scheme alone cannot produce such a sharp discontinuity at the contact surface.

(iii) FCT

Since it is of no use to incorporate the limiting procedure into the CIP, which only makes the scheme complicated, the "phoenical-SHASTA"[9] scheme is used for comparison and its result is shown in Fig.4(c). The steepness of the shock front is almost the same as that by the hybrid CIP, while the contact discontinuity becomes more diffusive. Although no overshooting occurs at the shock front, the density behind the shock wave is unfortunately larger by about 5 percent than the analytical solution. It may be possible to use a different limiting procedure, for example, the limiting proposed by Zalesak [16]. However, it is true that we must be careful in using the limiting procedure.

The test problem given above is a special one because the Mach number of the shock wave is infinite. It is worthwhile to test the scheme in a lower Mach number regime. Figures 5 and 6 give other examples : the initial condition is $\rho=1$, $E=2.5$ ($p=1$) for the leftmost 100 zones and $\rho=0.125$, $E=0.25$ ($p=0.1$) for the other zones ; $u=0$ and

$\gamma=1.4$ in all zones and $\Delta x=0.01$. This example was used in Sod's review[17]. Here, open circles are the result with the CIP and the solid line shows the analytical result, which is given in Appendix. In Fig.5, the artificial viscosity is of the type given in Eq.(23). Although the contact discontinuity is successfully described, the shock front is quite diffusive. This behavior is attributed to the first order viscosity given by Eq.(23a). Since in Fig.4 the pressure in front of the shock wave is zero, no significant diffusion occurs there. In contrast, in Fig.5 the sound speed in front of the shock wave is as fast as that behind the shock wave and hence Q_1 causes diffusion there. This diffusion can be reduced as shown in Fig.6 if C_s in Eq.(23a) is replaced by $|u|$, since u is zero in front of the shock wave.

From these test problems, it is clear that the CIP can describe sharp discontinuities without any special techniques except for the shock front. Associated with the shock front, further improvement is presumably expected with a different artificial viscosity or other techniques that we have not tried.

V. MULTIDIMENSIONAL CIP

It is generally known that a one-dimensional scheme can be easily extended to multi-dimensions by using the time splitting technique. However, this technique should be applied to the CIP with care because the CIP method uses the gradient of the dependent variable as a free parameter. For instance, let us consider a two-dimensional convective equation such as

$$\partial f / \partial t + c_x \partial f / \partial x + c_y \partial f / \partial y = 0, \quad (26)$$

where c_x and c_y are the velocities in the x-direction and the y-direction, respectively ; now both are temporarily taken to be constant. The time splitting techniques means that Eq.(26) is split to be

$$\bar{f}_i = L(x) f_i^n, \quad (27a)$$

$$f_i^{n+1} = L(y) \bar{f}_i, \quad (27b)$$

where $L(\lambda)$ is an operator of any scheme which gives the solution of the equation

$$\partial f / \partial t + c_\lambda \partial f / \partial \lambda = 0. \quad (28)$$

In the CIP scheme, Eq.(27) should be modified, because the scheme needs the information on the spatial derivative of f . That is, in solving Eq.(27b), Eq.(9) should be

applied to the y-direction with the aid of Eqs.(5c), (5d), (6), (7b), and (8). However, $\bar{\partial f}/\partial y$ is not yet known because in Eq.(27a) only $\bar{\partial f}/\partial x$ is calculated according to Eq.(6). Consequently, some method is required to estimate $\bar{\partial f}/\partial y$. In order to clarify the computing process, let us introduce a "CIP operator" $C(\lambda)$;

$$\begin{pmatrix} f^{n+1} \\ \partial f^{n+1}/\partial \lambda \end{pmatrix} = C(\lambda) \begin{pmatrix} f^n \\ \partial f^n/\partial \lambda \end{pmatrix}, \quad (\lambda=x, y) \quad (29)$$

which represents the process given in Eqs.(5c), (5d), (6), (7b), (8), and (9). By using this expression, Eq.(26) can be solved as

$$\begin{pmatrix} \bar{f} \\ \bar{p} \end{pmatrix} = C(x) \begin{pmatrix} f^n \\ p^n \end{pmatrix}, \quad (30a)$$

$$\bar{q} = s(x, y)q^n, \quad (30b)$$

$$\begin{pmatrix} f^{n+1} \\ q^{n+1} \end{pmatrix} = C(y) \begin{pmatrix} \bar{f} \\ \bar{q} \end{pmatrix}, \quad (30c)$$

$$p^{n+1} = s(y, x)\bar{p}, \quad (30d)$$

where $p=\partial f/\partial x$ and $q=\partial f/\partial y$.

Here the operator $S(\lambda, \mu)$ gives the solution of the equation

$$\partial/\partial t[\partial f/\partial \mu] + c_\lambda \partial/\partial \lambda[\partial f/\partial \mu] = 0, \quad (31)$$

$$(\mu, \lambda = x \text{ or } y, \text{ but } \mu \neq \lambda).$$

That is, the spatial derivative of f in some direction is convected toward the perpendicular direction. In the calculation described below, we found that the result is insensitive to the method used to solve Eq.(31) and hence the donor-cell-type finite difference scheme is used.

Figure 7 shows the propagation of a square wave in the direction at an angle of 45 degrees to the axis. The parameters used are $\Delta x = \Delta y = 1.0$, $c_x = c_y = 1.0$, and $\Delta t = 0.2$, and the figure shows the profiles at the initial step and after 1000 steps. The gradient correction was inserted in the initial 10 time steps and once every 50 steps as in the previous paper[1]. Thus, the CIP algorithm has been extended successfully to two-dimensional problems by applying sequentially one-dimensional CIP algorithms. This procedure has a great advantage because the subroutine for the CIP solver can be used in any number of dimensions and hence the program is largely simplified. This scheme works well also for the solid body rotation test applied to Zalesak's fully multi-dimensional

FCT[16]. Figure 8 shows the profile at an initial time and after one revolution with the CIP ; the conditions for the calculation are all the same as those in Zalesak's test run. It should be noticed that the older form of FCT[4,9] cannot reproduce the result shown here and the CIP method can give a similar result to the Zalesak's method in less CPU time.

VI. CONCLUSION

In this paper, we have re-examined the basic principle of the CIP from a different viewpoint and found that the scheme can be modified into an explicit difference form ; hence the scheme can execute efficiently on vector machines. The scheme was extended to nonlinear and multi-dimensional problems. In the nonlinear problem, the flux switching at the shock front gave a satisfactory result for a one-dimensional shock tube problem, although a further improvement should be expected by choosing a better dissipation term. The CIP scheme could be straightforwardly extended to multi-dimensional problems by successive application of one-dimensional CIP schemes; hence the problem in any number of dimensions can be solved by the subroutine for the one-dimensional CIP.

References

1. H.Takewaki, A.Nishiguchi, and T.Yabe, J.Comput.Phys. 61,261(1985).
2. M.M.Shouri, J.Comput.Phys. 49,334(1983).
3. G.Knorr and M.Mond, J.Comput.Phys. 38,212(1980).
4. J.P.Boris and D.L.Book, J.Comput.Phys. 11,38(1973).
5. P.D.Lax and B.Wendroff, Comm. Pure Appl. Math. 13, 217(1960).
6. P.D.Lax, Comm. Pure Appl. Math. 11,175(1958).
7. J.E.Fromm, J.Comput.Phys. 3,176(1968).
8. R.S.Hirsh, J.Comput.Phys. 19,90(1975).
9. D.L.Book, J.P.Boris, and K.Hain, J.Comput.Phys. 18,248(1975).
10. A.Nishiguchi and T.Yabe, J.Comput.Phys. 52, 390(1983).
11. M.W.Evans and F.H.Harlow, "The Particle-in-Cell Method for Hydrodynamic Calculations", Los Alamos Scientific Laboratory Report LA-2139, Los Alamos, N.M., 1957 ; F.H.Harlow, in "Methods in Computational Physics", Vol.3, Academic Press, New York, 1964 ; A.A.Amsden, "The Particle-in-Cell Method for the Calculation of the Dynamic of Compressive Fluids", Los Alamos Scientific Laboratory Report LA-3466, Los Alamos, N.M., 1966.
12. M.L.Wilkins, J.Comput.Phys. 36,281(1980).
13. A.Harten and G.Zwas, J.Comput.Phys. 9,568(1972).
14. G.J.McRae, W.R.Goodin, and J.H.Seinfeld, J.Comput.

- Phys. 45, 1(1982).
15. R.W.MacCormack, "The Effect of Viscosity in Hyper-velocity Impact Cratering" American Institute for Aeronautics and Astronautics Paper, pp.69-354(1969).
 16. S.T.Zalesak, J.Comput.Phys. 31, 335(1979).
 17. G.A.Sod, J.Comput.Phys. 27, 1(1978).

Figure Captions

- Fig.1 : Schematics of flux calculation in (a) the centered finite difference and (b) Lax-Wendroff schemes.
- Fig.2 : Propagation of a square wave after 1000 time steps, κ being 0.2, with up-wind, Friedrichs-Lax[6], leap-frog, Lax-Wendroff, Fromm[7], compact-differencing[8], phoenical-SHASTA[9], and CIP.
- Fig.3 : The basic principle of CIP. The solid and dashed lines mean the profiles at $t=t_n$ and $t=t_{n+1}$, respectively. Both the area A' and the difference between the outgoing and incoming fluxes ($A=A+C'-C$) are used to fill the area B.
- Fig.4 : The shock tube problem in a high Mach number regime using (a) CIP (artificial viscosity), (b) CIP(hybrid), (c) phoenical-SHASTA, (d) MacCormack, and (e) up-wind schemes.
- Fig.5 : The shock tube problem in a low Mach number

regime with CIP ; the artificial viscosity in Eq.(23) is used. "Energy" in the right bottom figure means the specific internal energy[$= p/\rho(\gamma-1)$].

Fig.6 : The same problem as in Fig.5 with CIP but C_s in Eq.(23a) is replaced by $|u|$.

Fig.7 : Linear wave propagation in the direction 45 degrees from the axis; (a) zones used, (b) the initial profile, and (c) the profile after 1000 time steps.

Fig.8 : Solid body rotation as in Ref.[16]; (a) zones used, (b) the initial profile, (c) CIP without gradient correction[1] after one revolution, and (d) CIP with gradient correction.

Fig.9 : The Rieman problem : (a) the initial condition, (b) the profile at $t>0$.

Appendix

In order to give a concrete example, it should be better to summarize the Riemann problem here, although it is available in literature. Figure 9 represents the configuration used in this problem. Figure 9(a) shows the initial conditions; a diaphragm is placed at $x=0$ between the regions 1 and 5. At $t>0$, the profile will be modified into that in Fig.9(b).

If the Mach number M of the shock wave located at $x=x_4$ is defined by use of the isothermal sound speed $(p/\rho)^{1/2}$ at the region 5, the Rankine-Hugoniot relation leads to

$$\frac{p_4}{p_5} = \frac{2M^2 - (\gamma - 1)}{\gamma + 1} \quad (A-1)$$

$$\frac{\rho_4}{\rho_5} = \frac{(\gamma + 1)M^2}{(\gamma - 1)M^2 + 2\gamma} \quad (A-2)$$

$$u_4 = \frac{2(M^2 - \gamma)}{(\gamma + 1)M} \left(\frac{p_5}{\rho_5} \right)^{1/2} \quad (A-3)$$

At the contact surface ($x=x_3$), the pressure and the velocity should be continuous. Thus,

$$u_3 = u_4, \quad (A-4)$$

$$p_3 = p_4, \quad (A-5)$$

In the adiabatic expansion region (region 2), a simple relation is derived for a simple wave ; $u = u(\rho)$, that is, u is a single-valued function of ρ . After $\partial u / \partial r$ and

$\partial u/\partial t$ are transformed into $(du/d\rho)(\partial\rho/\partial r)$ and $(du/d\rho)(\partial\rho/\partial t)$, Eqs.(21a) and (21b) reduce to an equation:

$$\frac{\partial\rho}{\partial t} + \left[u - \left(\frac{\gamma p}{\rho} \right)^{1/2} \right] \frac{\partial\rho}{\partial r} = 0 \quad (A-6)$$

if the function $u(\rho)$ satisfies a relation

$$\frac{du}{d\rho} = - \frac{1}{\rho} \left(\frac{\gamma p}{\rho} \right)^{1/2} \quad (A-7)$$

This last expression gives

$$u_3 = \left(\frac{\gamma p_1}{\rho_1} \right)^{1/2} \frac{2}{\gamma-1} \left[1 - \left(\frac{p_3}{\rho_1} \right)^{(\gamma-1)/2} \right] \quad (A-8)$$

Equation (A-6) means that the rarefaction front ($x=x_1$) and the boundary $x=x_2$ move with the velocities $-(\gamma p_1/\rho_1)^{1/2}$ and $u_3 - (\gamma p_3/\rho_3)^{1/2}$, respectively.

Since the adiabatic relation

$$p_3/p_1 = (\rho_3/\rho_1)^\gamma \quad (A-9)$$

holds, u_3 is related to p_3 as

$$u_3 = \left(\frac{\gamma p_1}{\rho_1} \right)^{1/2} \frac{2}{\gamma-1} \left[1 - \left(\frac{p_3}{p_1} \right)^{(\gamma-1)/2\gamma} \right] \quad (A-10)$$

When Eqs.(A-1) and (A-3) with (A-4) and (A-5) are substituted into Eq.(A-10), we obtain the relation which determines the Mach number.

$$(\gamma-1) \frac{M^2 - \gamma}{(\gamma+1)M} \left(\frac{p_5 \rho_1}{\gamma p_1 \rho_5} \right)^{1/2} = 1 - \left(\frac{2M^2 - (\gamma-1)}{(\gamma+1)} \frac{p_5}{p_1} \right)^{(\gamma-1)/2\gamma} \quad (A-11)$$

In Table 1, the numerical values used in Figs.4 and 5 are summarized.

Table 1

	Fig.4	Fig.5
γ	1.4	5/3
P_1	1.0	2/3
P_1	1.0	1.0
P_5	0.1	0.0
P_5	0.125	0.5
M:Eq. (A-11)	1.958	∞
Shock Speed	1.751	0.7924
$u_3 = u_4$:Eq. (A-3)	0.9266	0.5943
$P_3 = P_4$:Eq. (A-1)	0.3028	0.2354
ρ_4 :Eq. (A-2)	0.2654	2.0
ρ_3 :Eq. (A-9)	0.4260	0.5355
Speed of $x=x_2$: $u_3 - C_3$	-0.071	-0.2616
Speed of $x=x_1$: $-C_1$	-1.183	-1.054

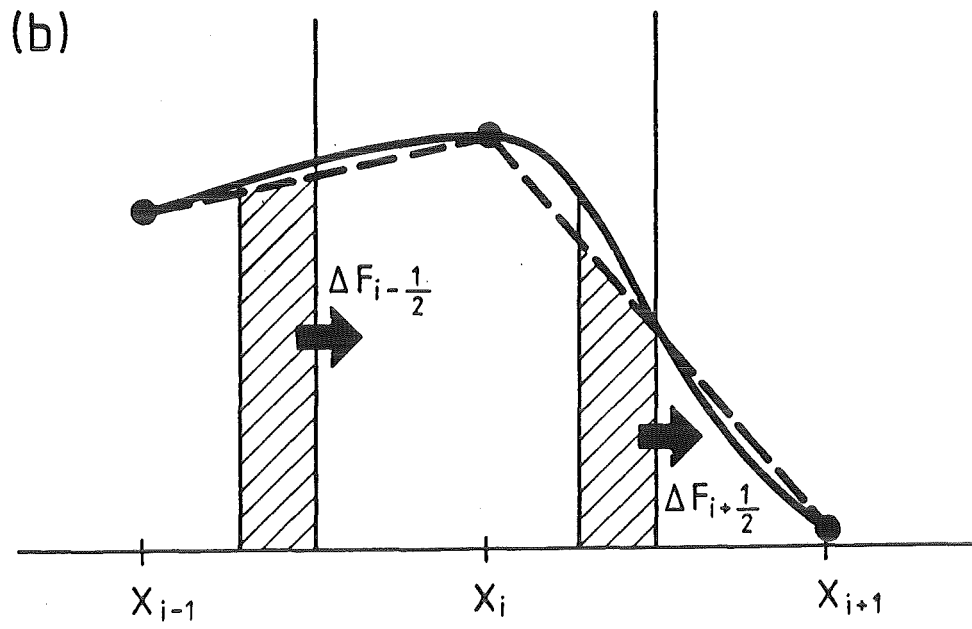
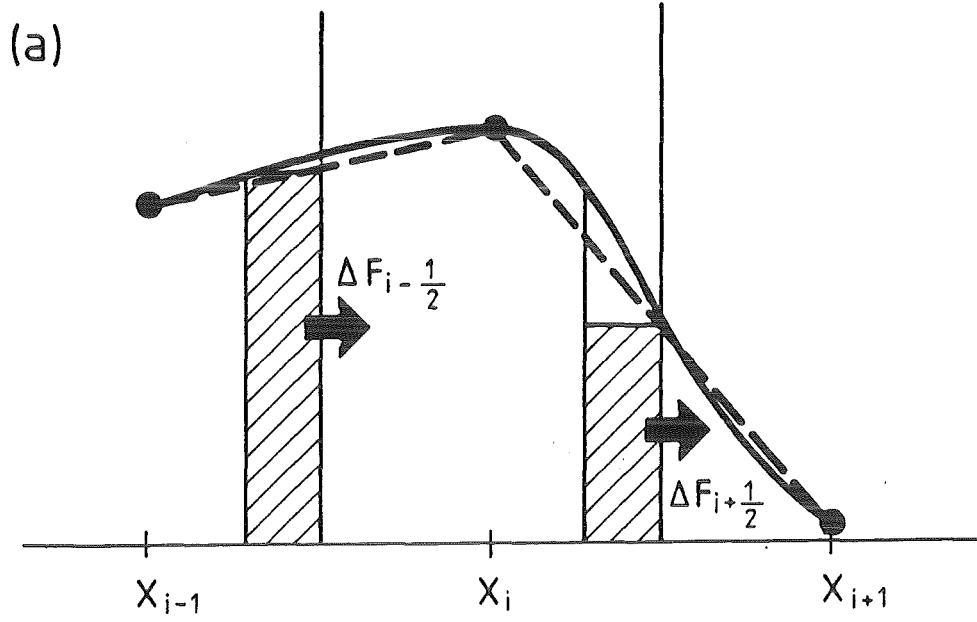
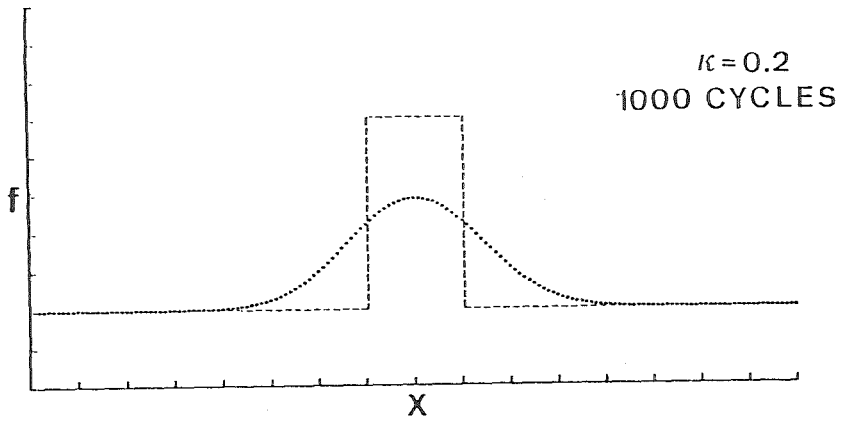
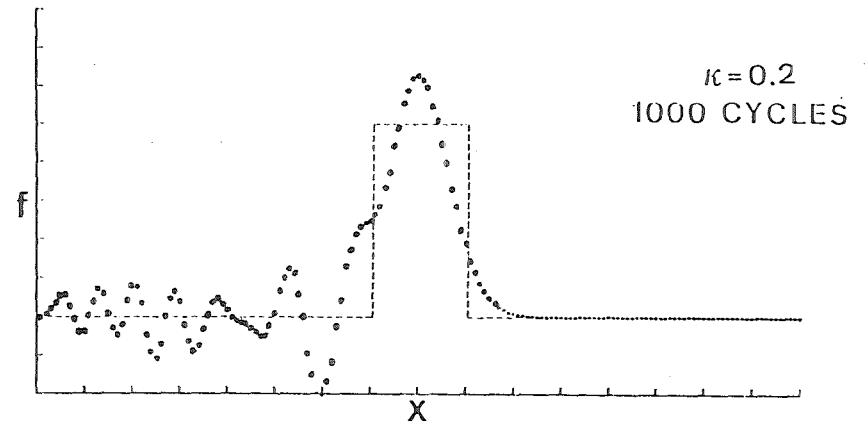


Fig. 1

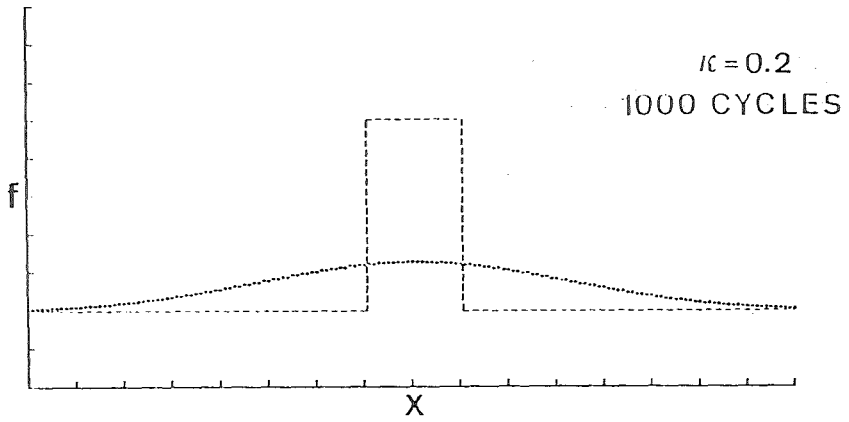
UP-WIND SCHEME



LEAP-FROG SCHEME



FRIEDRICHS-LAX SCHEME



LAX-WENDROFF SCHEME (2nd-order)

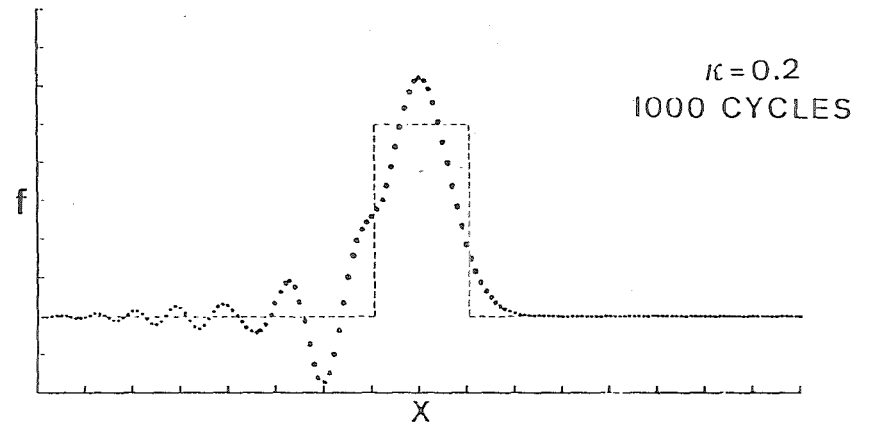
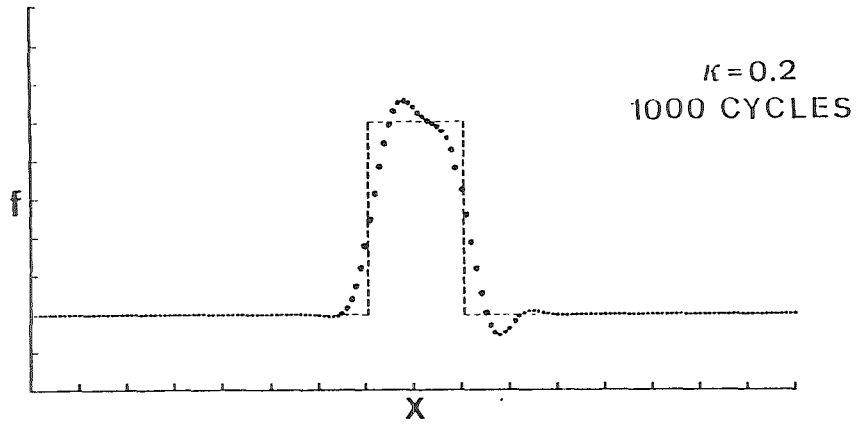


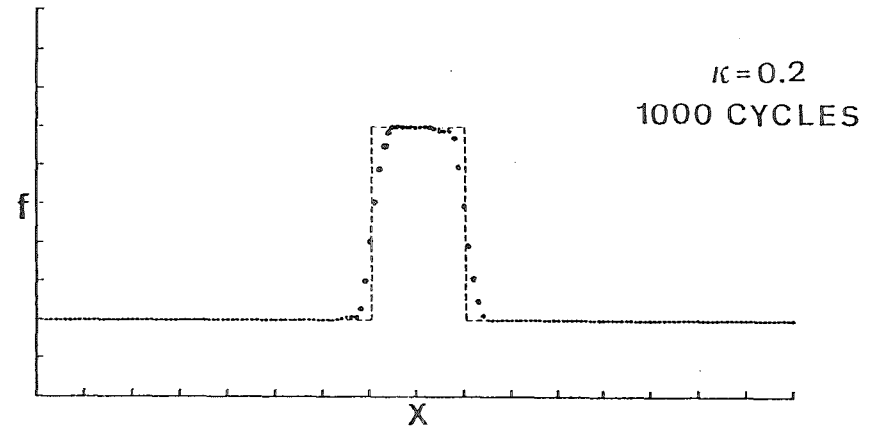
Fig.2

FROMM SCHEME

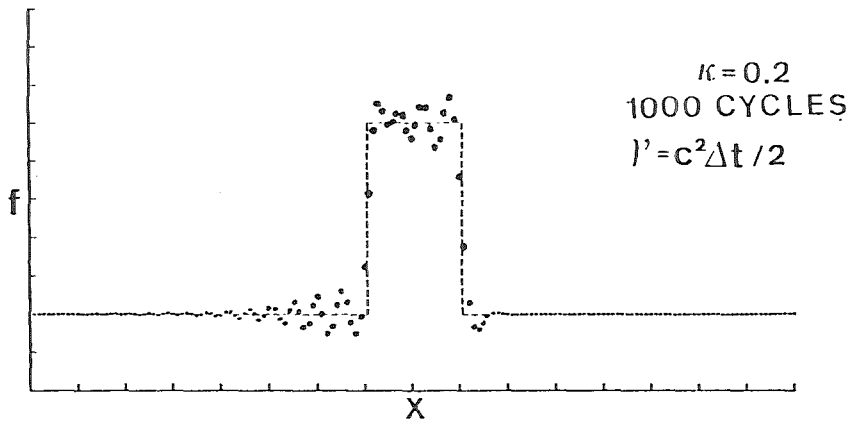


PHOENICAL SHASTA SCHEME

$$\eta = (1 - \kappa^2) / 6$$



COMPACT-DIFFERENCING SCHEME



CIP SCHEME

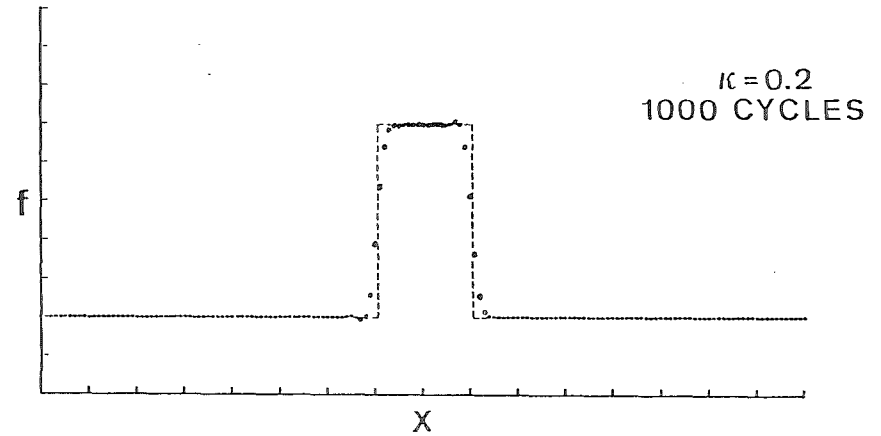


Fig. 2

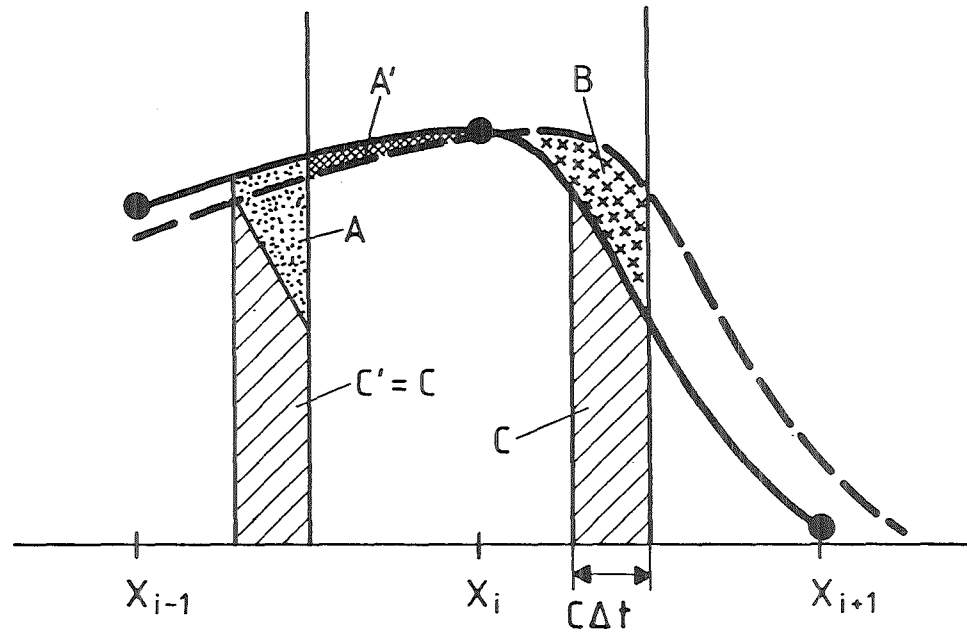


Fig. 3

C I P SCHEME

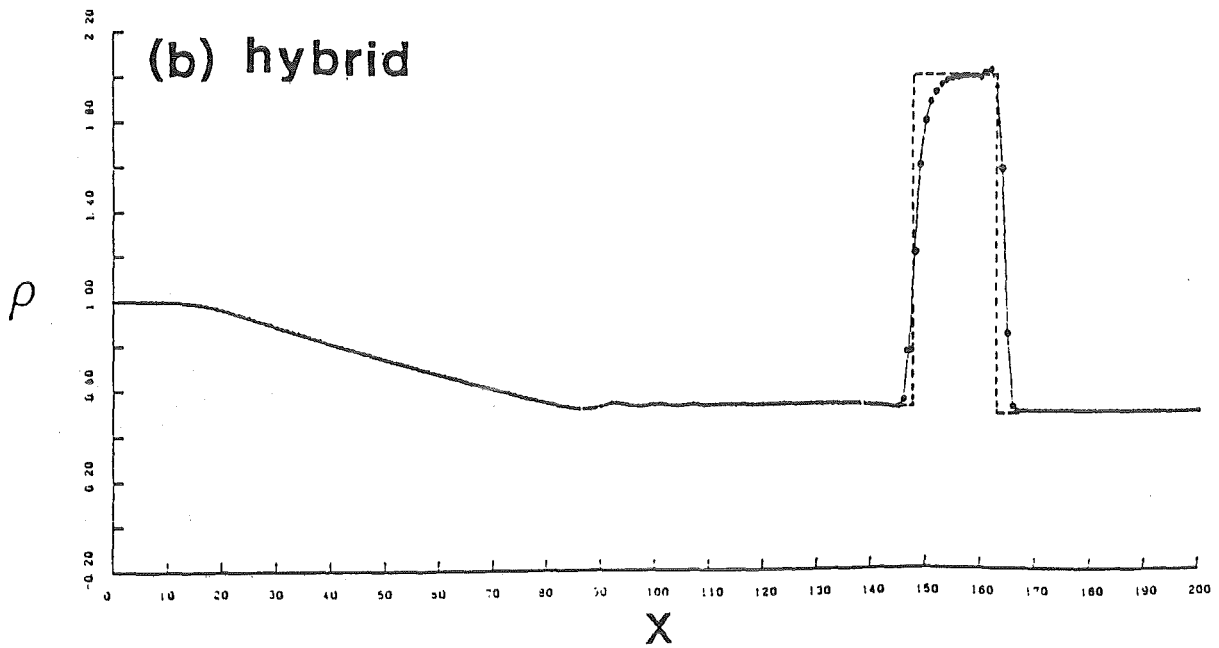
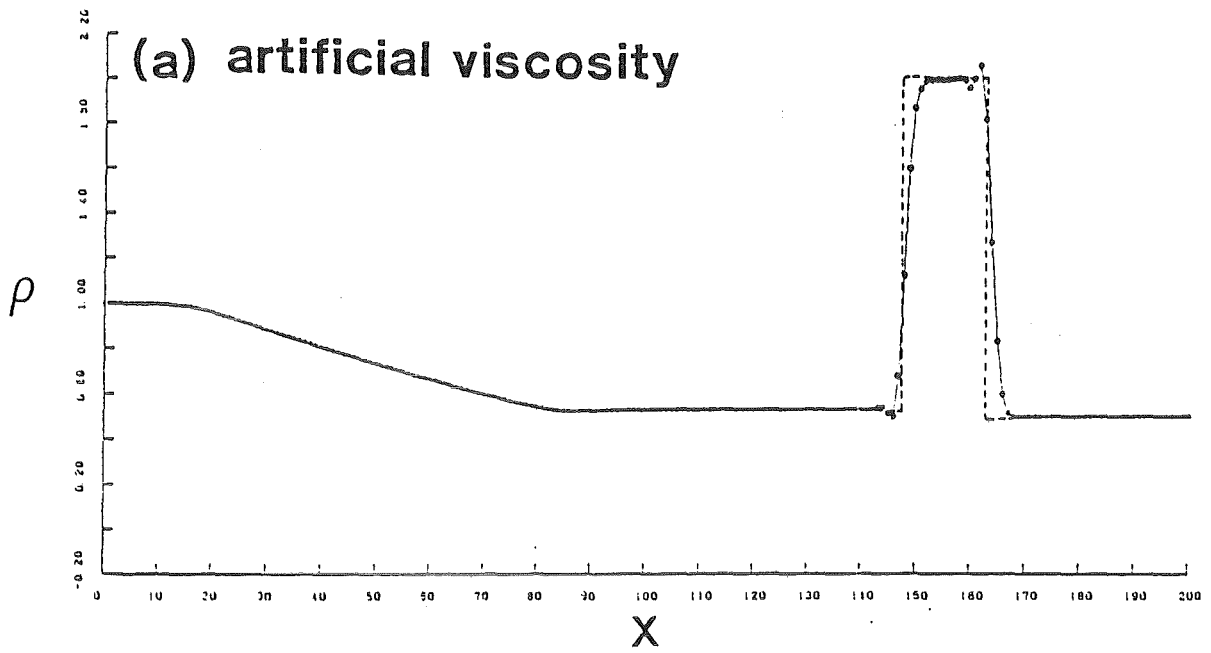
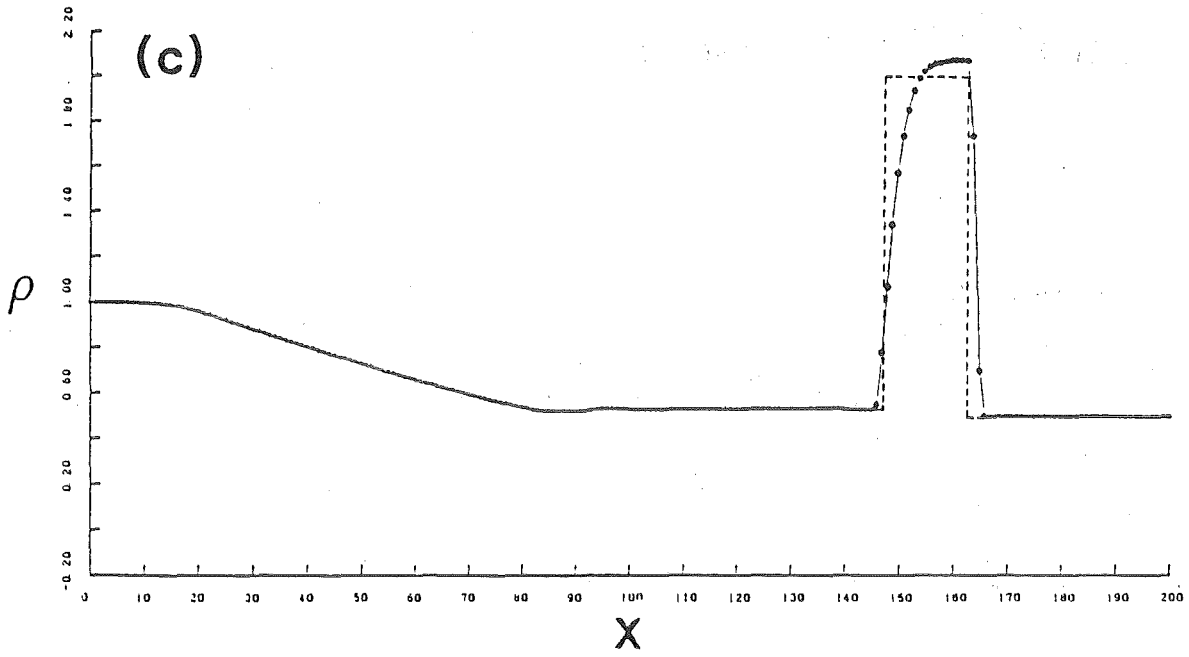


Fig.4

PHOENICAL-SHASTA SCHEME



MACCORMACK SCHEME

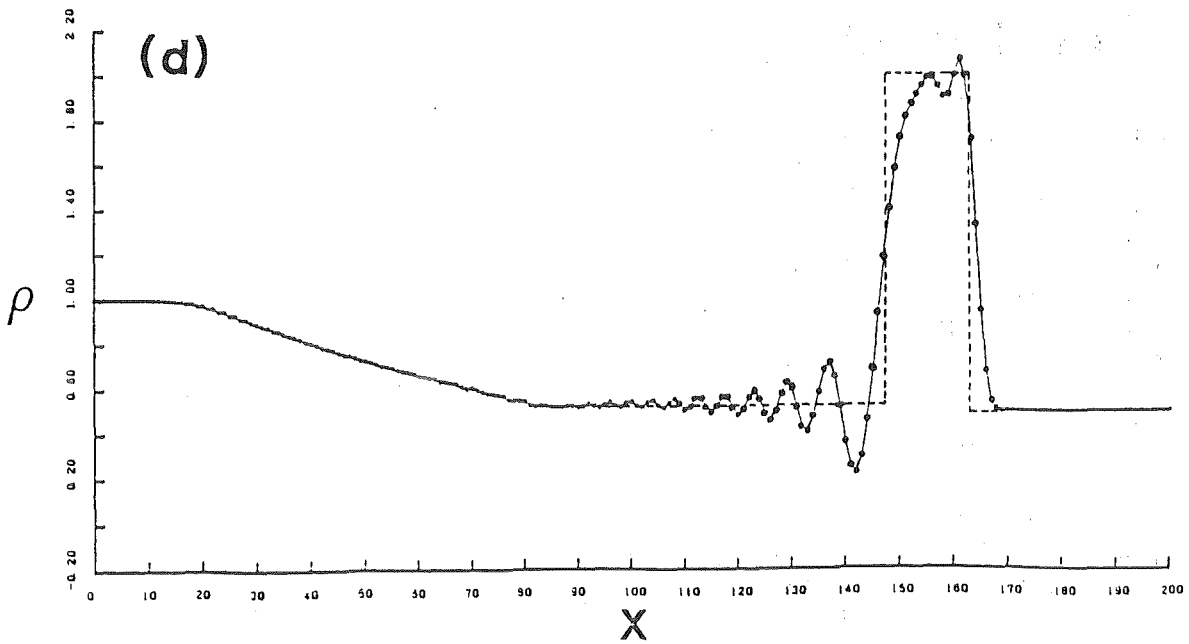


Fig. 4

UP-WIND SCHEME

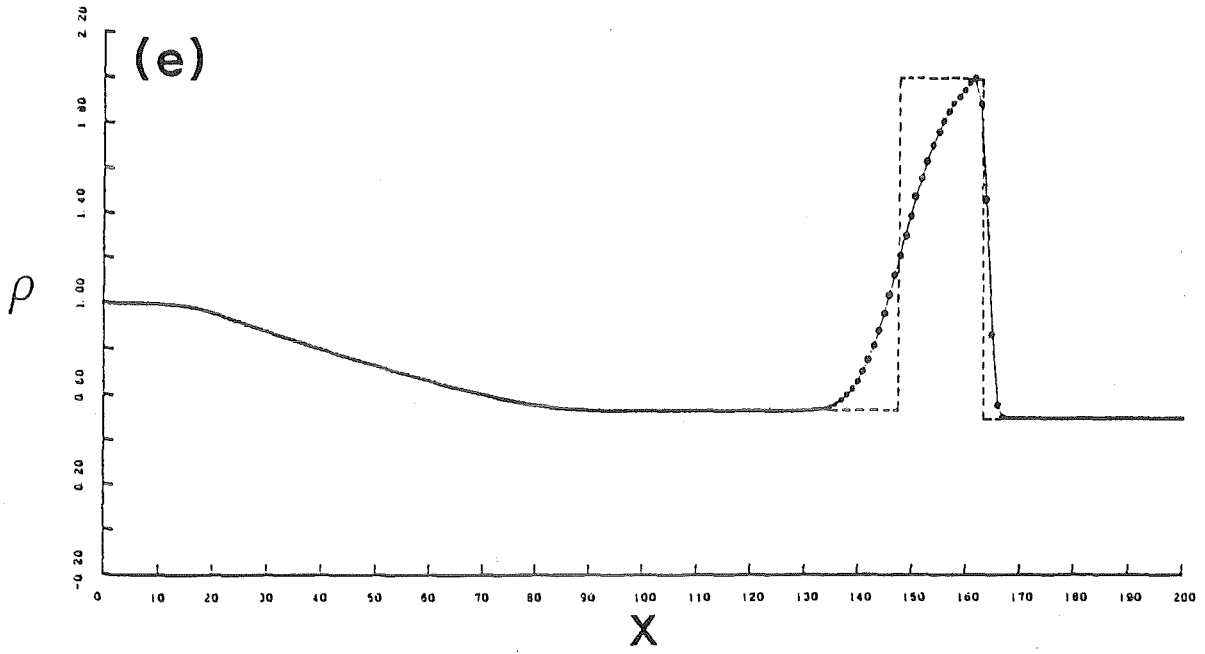


Fig. 4

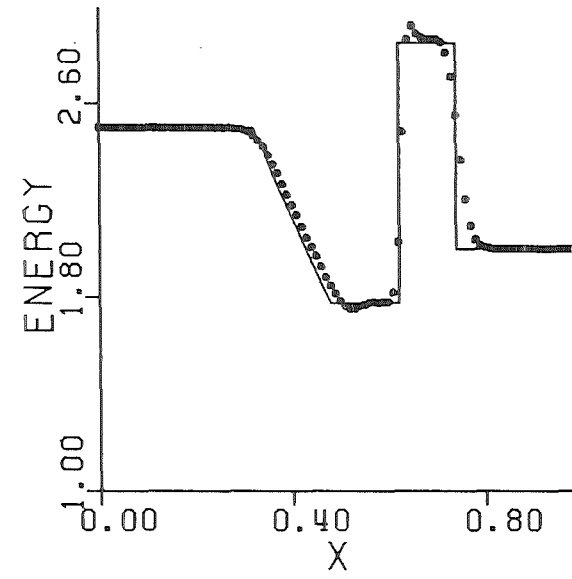
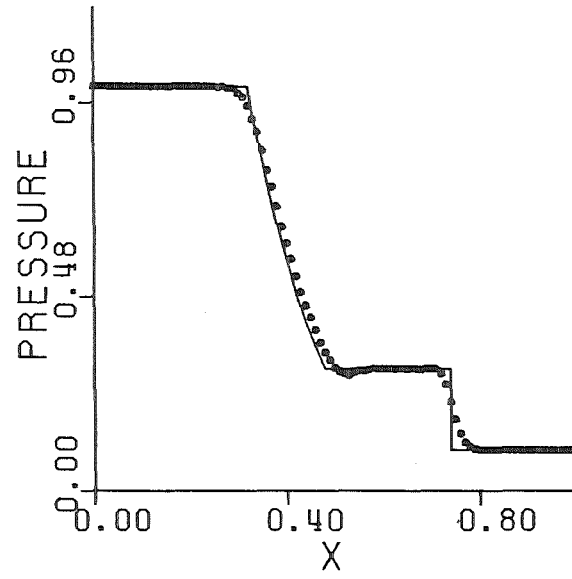
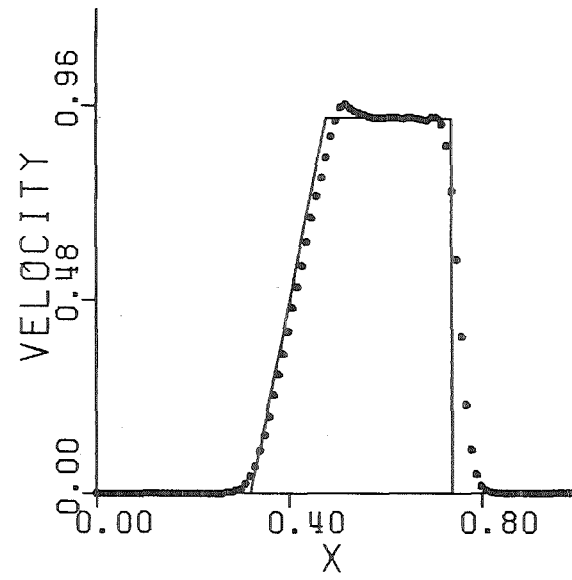
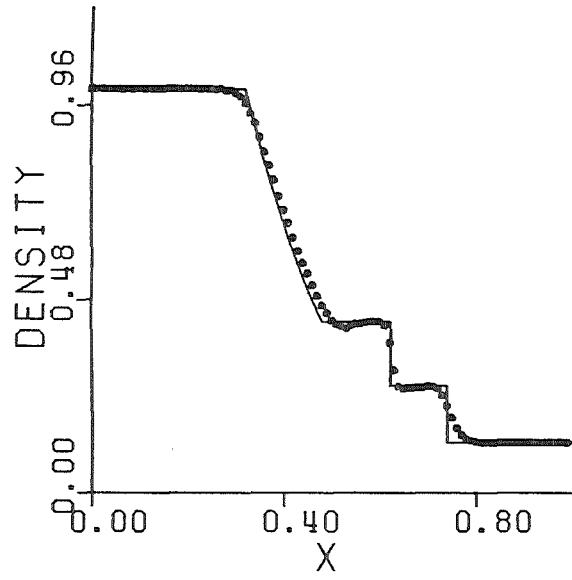


Fig. 5

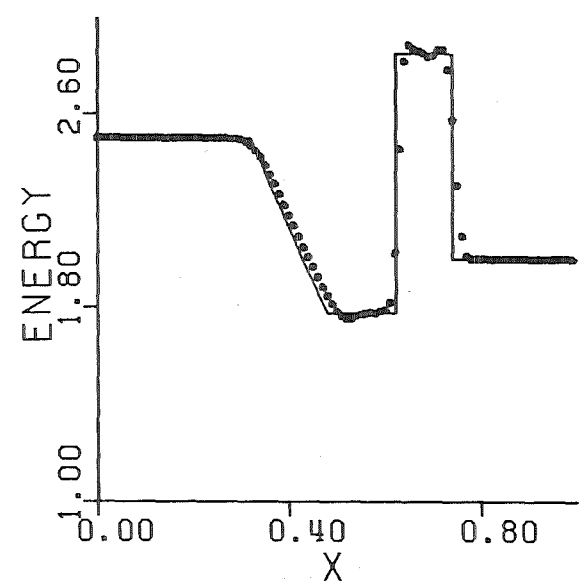
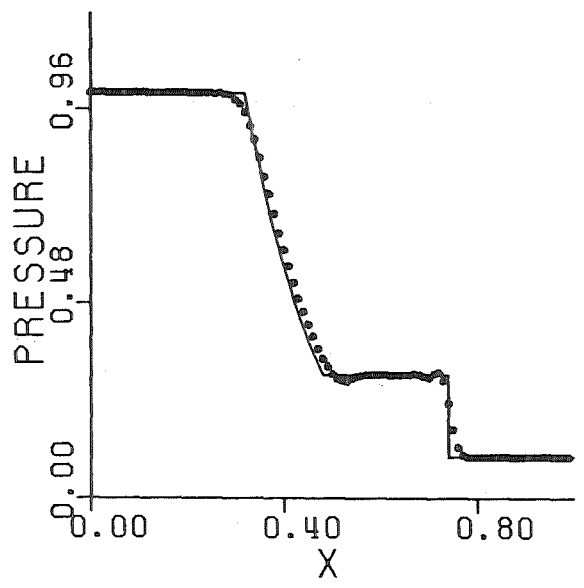
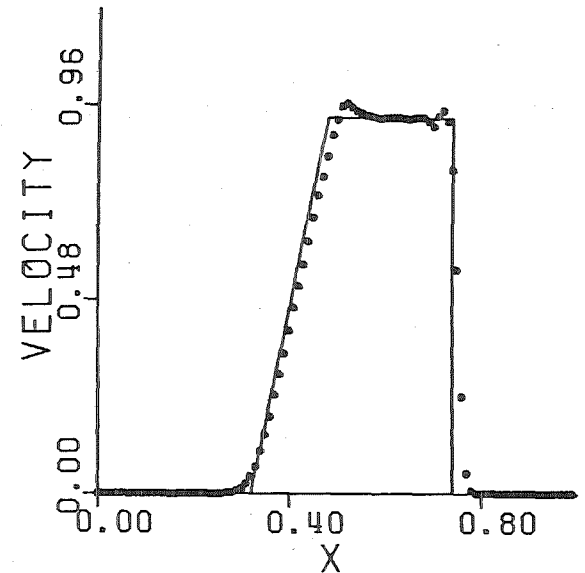
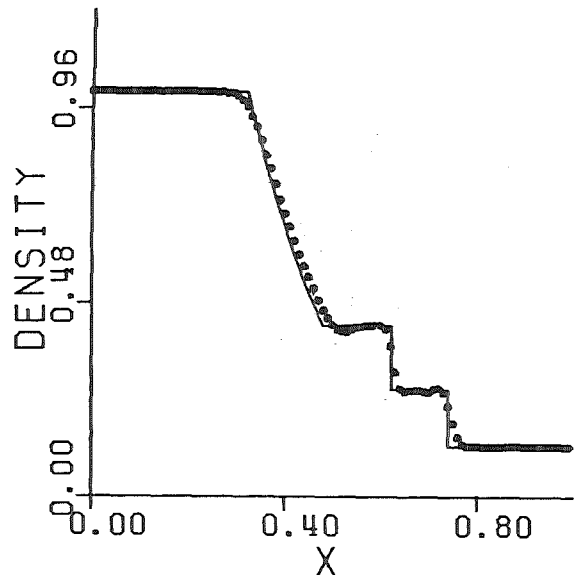


Fig. 6

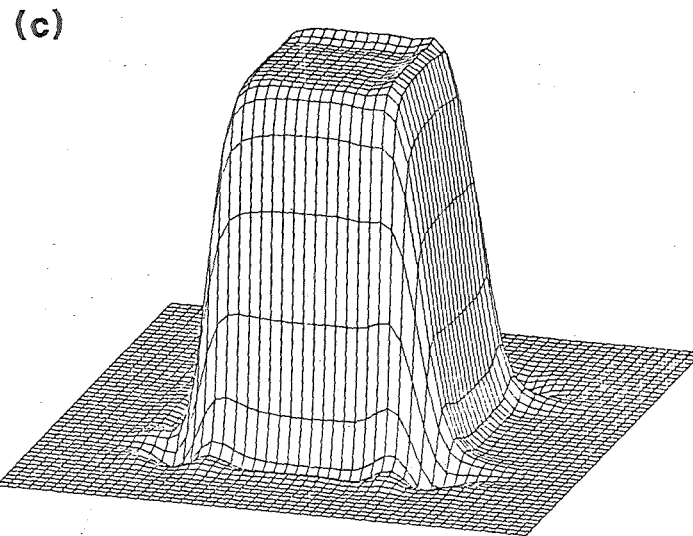
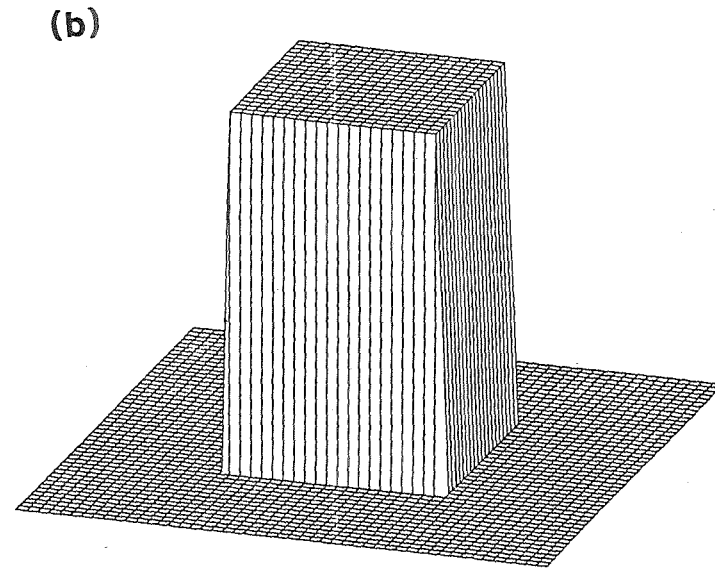
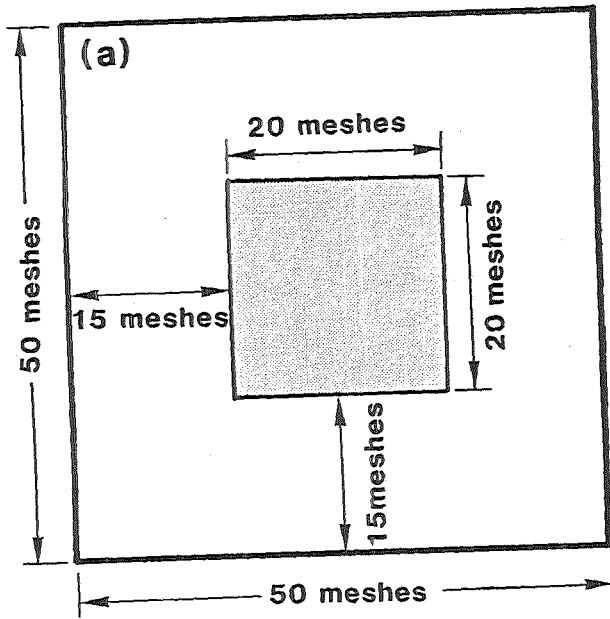


Fig. 7

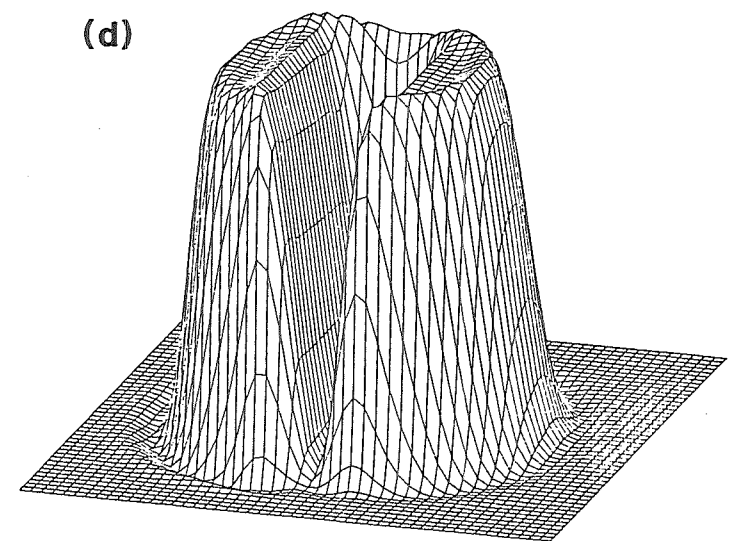
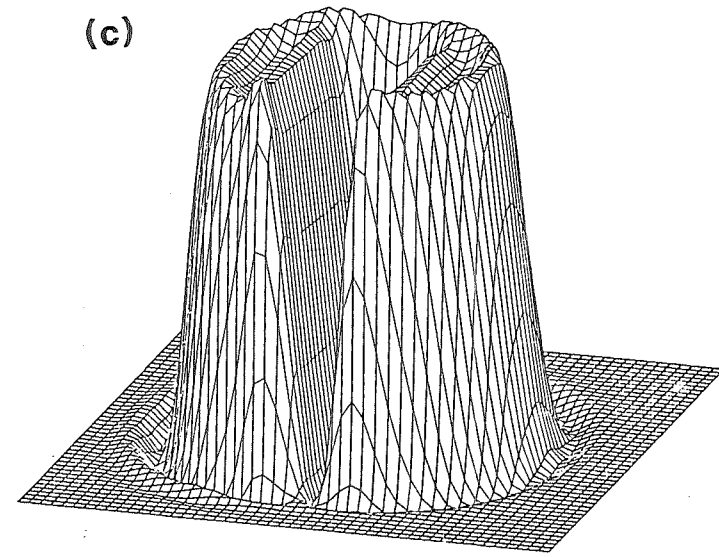
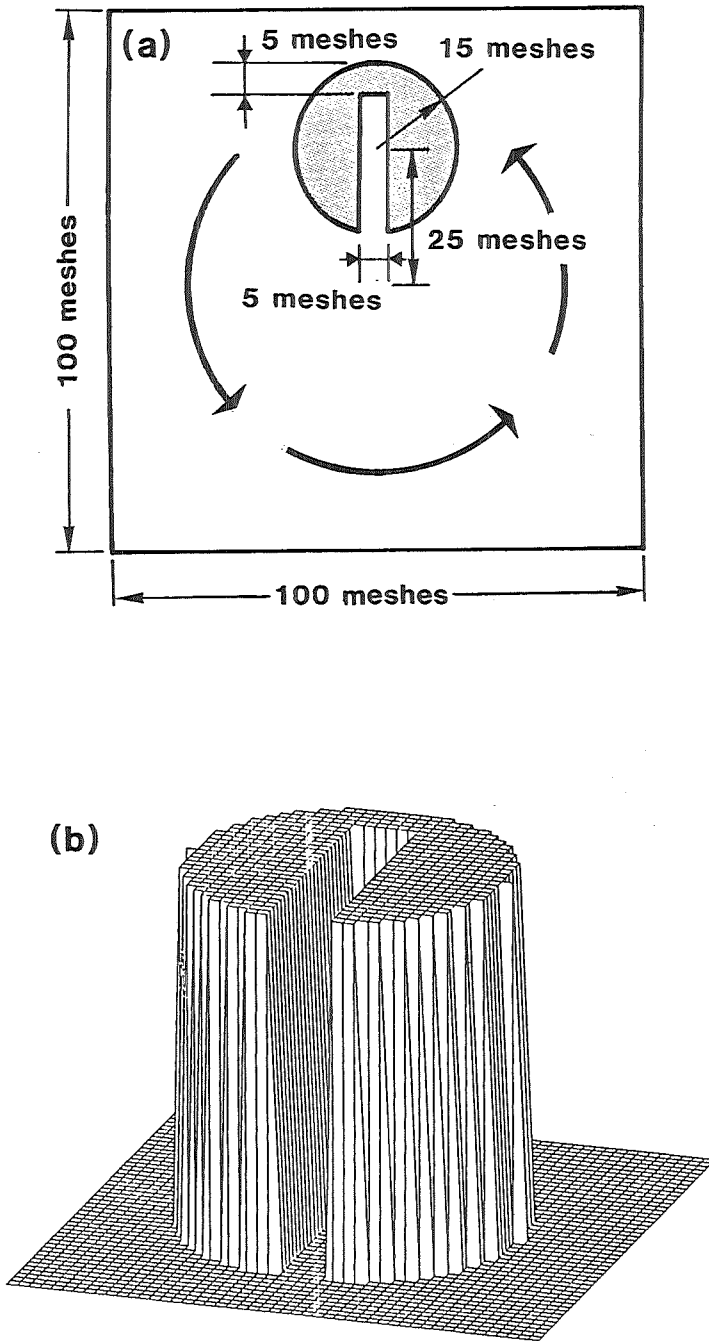


Fig. 8

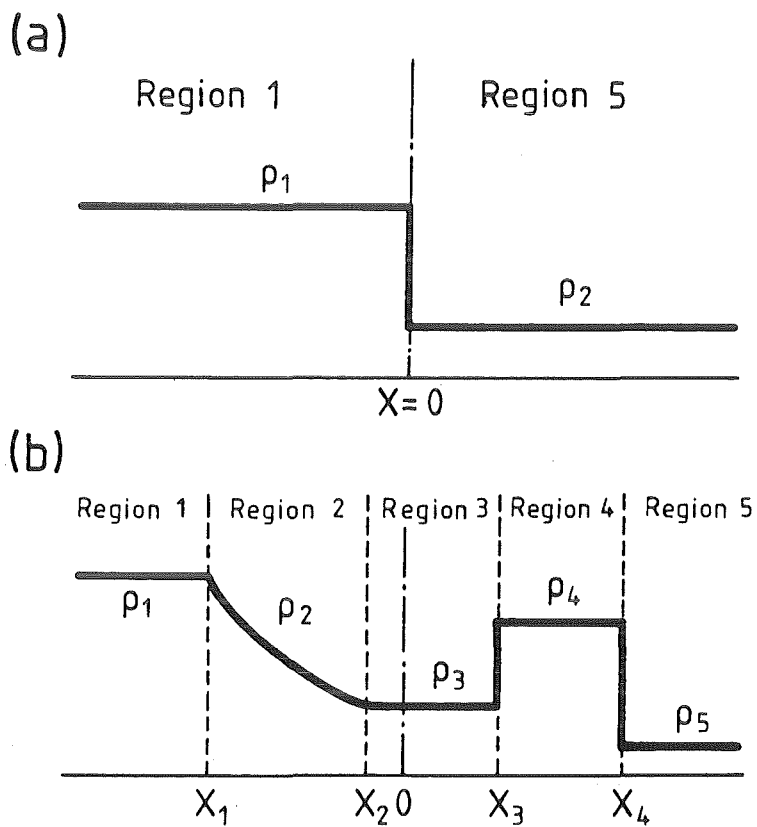


Fig. 9

Acknowledgements

One of the authors (T.Y.) would like to thank Professor Dr. G.Keßler and Dr. R.Fröhlich for giving him a chance to stay at KfK as a guest scientist and Dr. R.Fröhlich, Dr. B.Goel, and Dr. C.D.Munz for valuable discussion.

In the calculations included in this paper, we have used partly FACOM VP100 at Institute of Plasma Physics, Nagoya University, and partly IBM3090 and SIEMENS-7890 at KfK.

```
C |=====|
C |
C |          ### 1-D SHOCK TUBE PROBLEM ###          |
C |          |----- < BASIC EQUATION > -----|          |
C |          |-----|          |          |          |
C |          |-----|          |          |          |
C |          |-----|          |          |          |
C |          |-----|          |          |          |
C |=====|
```

<<<<<<<< TABLE OF CONSTANTS >>>>>>

- CK : DT/DR
- CGR : COEFFICIENT FOR CIP = (5/192)DR
- CVS : COEFFICIENT FOR ARTIFICIAL VISCOSITY
- DR : SPATIAL STEP
- DRI : 1/DR
- DT : TIME STEP
- GAMMA : SPECIFIC HEAT
- GM1 : GAMMA -1
- GP1H : (GAMMA+1)/2
- MESH : NUMBERS OF MESHES
- MM1 : MESH-1
- MM2 : MESH-2
- MP1 : MESH+1
- NGEOM : GEOMETRY(1,2,3) SEE INIT
- R1 : COEFFICIENT FOR CIP = -(5/192)DR
- R2 : COEFFICIENT FOR CIP = 0
- R3 : COEFFICIENT FOR CIP = (5/192)DR
- R4 : COEFFICIENT FOR CIP = (18/192)
- R5 : COEFFICIENT FOR CIP = (156/192)
- R6 : COEFFICIENT FOR CIP = (18/192)
- ZGEOM : NGEOM-1

<<<<<<<< TABLE OF VARIABLES >>>>>>

- FE : $H(N;I)+G(N,I)-(DF(I+1/2)-DF(I-1/2))/DR$ IN EQ.(9)-ENERGY
- FM : SAME - MOMENTUM
- FR : SAME - DENSITY
- FER : CONVECTIVE FLUX DF IN EQ.(9) - ENERGY
- FMR : SAME - MOMENTUM
- FRR : SAME - DENSITY
- GE : GRADIENT OF ENERGY
- GM : GRADIENT OF MOMENTUM
- GR : GRADIENT OF DENSITY
- YE : TOTAL ENERGY


```
C   YM   : MOMENTUM
C   YR   : DENSITY
C   YEGRD : USED FOR ADVANCING GRADIENT IN TIME EQ.(13) - ENERGY
C   YMGRD : SAME - MOMENTUM
C   YRGRD : SAME - DENSITY
C   YP   : PRESSURE
C   YU   : VELOCITY
C   YVIS : ARTIFICIAL VISCOSITY
```

```
C
C
C   <<<<<<<< TABLE END >>>>>>>>
```

```
C   IMPLICIT REAL*8(A-H,O-Z)
```

```
C   COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
&   YVIS(1000),FR(1000),FM(1000),FE(1000),
&   GR(1000),GM(1000),GE(1000)
```

```
C   COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
&   R1,R2,R3,R4,R5,R6,
&   ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
```

```
C   COMMON /CMGRAD/
YRGRD(1000),YMGRD(1000),YEGRD(1000),ZGEOM,NGEOM
```

```
C   DIMENSION FRR(1000),FMR(1000),FER(1000)
```

```
C===== INITIALIZE =====
```

```
C   CALL INIT
```

```
C   N = 1
```

```
C===== MAIN LOOP =====
```

```
C   DO 10 ISTEP=1,ILOOP
```

```
C----- CALCULATE VELOCITY -----
```

```
C   CALL NUCLC
```

```
C----- CALCULATE PRESSURE -----
```

```
C   CALL YPCLC
```

```
C----- CALCULATE VISCOSITY -----
```

```
C   CALL VSCLC
```

```
C
```

```
C===== EULERIAN PHASE =====
C
C
C----- DENSITY -----
C
C      CALL DENRHS
C
C----- MOMENTUM -----
C
C      CALL MOMRHS
C
C----- DENSITY GRADIENT -----
C
C      CALL NEWGRD(GR,YRGRD)
C
C----- MOMENTUM GRADIENT -----
C
C      CALL NEWGRD(GM,YMGRD)
C
C----- CALCULATE TEMPORAL VELOCITY FOR
ENERGY -----
C
C      CALL EUCLC
C
C----- TOTAL ENERGY -----
C
C      CALL ENERHS
C
C----- TOTAL ENERGY GRADIENT -----
C
C      CALL NEWGRD(GE,YEGRD)
C
C----- CALCULATE NEW VELOCITY -----
C
C      CALL NUCLC
C
C===== LAGRANGIAN PHASE =====
C
C      SOLVED BY CIP ALGORITHM
C
C=====
C
C      <<<<<<<<<< DENSITY >>>>>>>>>>
C
C..... CALCULATE FLUX - EQ.(5D) .....
C
C      CALL FLCLC(FRR,YR,GR,YU)
C
```

```
C..... CIP SOLVER .....
C
C    --- H+G-DF/DR IN EQ.(9) ---
C
C    CALL CIPRHS(FR,FRR,GR,YR)
C
C    --- PREDICT H AND G AT ADVANCED TIME AND COMPLETE EQ.(9) --
C
C    CALL CIPCLC(YR,GR,FR,YU)
C
C
C          <<<<<<<<<< MOMENTUM >>>>>>>>>>
C
C..... CALCULATE FLUX .....
C
C    CALL FLCLC(FMR,YM,GM,YU)
C
C..... CIP SOLVER .....
C
C    CALL CIPRHS(FM,FMR,GM,YM)
C    CALL CIPCLC(YM,GM,FM,YU)
C
C
C          <<<<<<<<<< ENERGY >>>>>>>>>>
C
C..... CALCULATE FLUX .....
C
C    CALL FLCLC(FER,YE,GE,YU)
C
C..... CIP SOLVER .....
C
C    CALL CIPRHS(FE,FER,GE,YE)
C    CALL CIPCLC(YE,GE,FE,YU)
C
C
C..... GRADIENT CORRECTION .....
C
C    IF ((ISTEP.LE.10).OR.(MOD(ISTEP,50).EQ.0)) THEN
C      CALL CORRCT(YR,GR,DR,MESH)
C      CALL CORRCT(YM,GM,DR,MESH)
C      CALL CORRCT(YE,GE,DR,MESH)
C    ENDIF
C
```

```
C----- TIME COUNT -----  
C  
      TIME = TIME + DT  
C  
C===== OUTPUT =====  
C  
C  
      IF(MOD(ISTEP,10).EQ.0) THEN  
        PRINT *, ' ISTEP= ',ISTEP,'   TIME= ',TIME  
        CALL YOUT(YR,1,100,'DENSITY  '  
        CALL YOUT(YM,1,100,'MOMENTUM'  
        CALL YOUT(YE,1,100,'ENERGY  '  
      ENDIF  
C  
10  C O N T I N U E  
C  
C-----  
      S T O P   0 0 0  
C-----  
C  
      E N D  
C
```

```
C |-----|
C |
C |
C | SET INITIAL CONDITIONS AND CONSTANTS
I
C |
C |
C |
C |-----|
C
C SUBROUTINE INIT
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
& YVIS(1000),FR(1000),FM(1000),FE(1000),
& GR(1000),GM(1000),GE(1000)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C COMMON /CMGRAD/
YRGRD(1000),YMGRD(1000),YEGRD(1000),ZGEOM,NGEOM
C
C-----
C
C.....TOTAL MESH USED
C
C MESH = 100
C
C..... 1(PLANE), 2(CYLINDER), 3(SPHERE)
C
C NGEOM = 1
C.....
C ZGEOM = DFLOAT(NGEOM-1)
C PRINT *, ' NGEOM = ',NGEOM
C
C..... MAXIMUM LOOP
C
C ILOOP = 100
C
C..... SPACE INCREMENT
C
C DR = 0.01D0
C
C..... TIME STEP
C
C DT = 0.001D0
C
```

```
C..... SPECIFIC HEAT RATIO
C
      GAMMA = 1.4D0
C
C..... COEFFICIENT FOR VISCOSITY
C
      CVS = 1.0D0
C
      WRITE(6,1000) DR,DT,GAMMA,CVS,ILOOP
1000  FORMAT(1H ,5X,'DR =',F7.2,1X,'DT =',F7.3,1X,'GAMMA =',F7.2,
&        1X,'CVS =',F7.3,1X,'ILOOP =',I5,/)
C
C===== CALCULATE THE COEFFICIENTS USED IN THE CODE
=====
C
      ISTEP = 0
      MP1 = MESH + 1
      MM1 = MESH - 1
      MM2 = MESH - 2
C
      TIME = 0.00D0
      CK = DT / DR
      DRI = 1.00D0 / DR
      CGR = 5.00D0 / 192.0D0 * DR
C
      GM1 = GAMMA - 1.00D0
      GP1H = 0.50D0 * ( GAMMA + 1.00D0 )
C
      R1 = - 5.0D0 / 192.0D0 * DR
      R2 = 0.0D0
      R3 = 5.0D0 / 192.0D0 * DR
      R4 = 18.0D0 / 192.0D0
      R5 = 156.0D0 / 192.0D0
      R6 = 18.0D0 / 192.0D0
C
C===== CLEAR ARRAYS =====
C
      DO 10 I=1,MESH
          YR(I) = 0.00D0
          YM(I) = 0.00D0
          YE(I) = 0.00D0
          YP(I) = 0.00D0
          YU(I) = 0.00D0
          GR(I) = 0.00D0
          GM(I) = 0.00D0
          GE(I) = 0.00D0
          FR(I) = 0.00D0
          FM(I) = 0.00D0
          FE(I) = 0.00D0
          YVIS(I) = 0.00D0
10  CONTINUE
C
```

```
C===== SET INITIAL PROFILE =====
C
C
C          <<<<<<<<<<  MOMENTUM  >>>>>>>>>
C
C      DO 30 I=1,MESH
C          YM(I) = 0.00D0
C 30  CONTINUE
C
C          <<<<<<<<<<  DENSITY  >>>>>>>>>
C          <<<<<<<<<<  ENERGY  >>>>>>>>>
C
C      DO 40 I=1,50
C          YE(I) = 2.50D0
C          YR(I) = 1.00D0
C 40  CONTINUE
C
C      DO 41 I=51,MESH
C          YE(I) = 0.25D0
C          YR(I) = 0.125D0
C 41  CONTINUE
C
C  -----
C  R E T U R N
C  -----
C
C  END
C
```

```
C |-----|
C |
C |   CALCULATE PRESSURE FROM
C |   ( (TOTAL ENERGY)-(KINETIC ENERGY) ) / ( GAMMA - 1 )
|
C |
C |
C |
C |-----|
C
C   SUBROUTINE YPCLC
C
C   IMPLICIT REAL*8(A-H,O-Z)
C
C   COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
&     YVIS(1000),FR(1000),FM(1000),FE(1000),
&     GR(1000),GM(1000),GE(1000)
C
C   COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
&     R1,R2,R3,R4,R5,R6,
&     ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C-----
C
C   DO 10 I=1,MESH
C     YP(I) = GM1 * ( YE(I) - 0.50D0*YR(I)*YU(I)*YU(I) )
C     IF (YP(I).LE.0.00D0)      YP(I) = 0.00D0
10  C O N T I N U E
C
C-----
C   R E T U R N
C-----
C
C   END
C
```



```
C |-----|
C |
C |
C | ARTIFICIAL VISCOSITY Q1 + Q2
C |
C | Q1 : FIRST ORDER           Q2 : SECOND ORDER
C |-----|
C
C SUBROUTINE VSCLC
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
& YVIS(1000),FR(1000),FM(1000),FE(1000),
& GR(1000),GM(1000),GE(1000)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C-----C
C
C DO 10 I=2,MESH
C   IM1 = I - 1
C   DLTU = YU(I) - YU(IM1)
C   RMAN = YR(I) + YR(IM1)
C   FSQR = RMAN * ( YP(I) + YP(IM1) )
C   IF ((DLTU.LT.0.0D0).AND.(FSQR.GE.0.0D0)) THEN
C     QA = - 0.50D0 * DLTU * SQRT( FSQR * GAMMA )
C     QB =  0.50D0 * RMAN * DLTU * DLTU
C     YVIS(I) = CVS * ( QA + GP1H*QB )
C   ELSE
C     YVIS(I) = 0.00D0
C   ENDIF
10 C O N T I N U E
C
C YVIS(1) = YVIS(2)
C
C-----C
C R E T U R N
C-----C
C
C END
```

```
C |-----|
C |
C |   CALCULATE NON-ADVECTIVE TERM
C |
C |           IN THE EULERIAN PHASE
C |
C |           <<<<<<< DENSITY >>>>>>>
C |-----|
C
C   SUBROUTINE DENRHS
C
C   IMPLICIT REAL*8(A-H,O-Z)
C
C   COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
&     YVIS(1000),FR(1000),FM(1000),FE(1000),
&     GR(1000),GM(1000),GE(1000)
C
C   COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
&     R1,R2,R3,R4,R5,R6,
&     ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C   COMMON /CMGRAD/
YRGRD(1000),YMGRD(1000),YEGRD(1000),ZGEOM,NGEOM
C
C-----
C
C   DO 10 I=1,MESH
C     POSI = 1.0D0 / ( DFLOAT(I)-0.50D0 ) * DRI
C     YRGRD(I) = - ZGEOM*YR(I)*YU(I)*POSI*DT
C     YR(I) = YR(I) + YRGRD(I)
10  C O N T I N U E
C
C     POSI = 1.0D0 / ( DFLOAT(MESH)-0.50D0 ) * DRI
C     YRGRD(MESH) = - ZGEOM*YR(MESH)*YU(MESH)*POSI*DT
C     YR(MESH) = YR(MESH) + YRGRD(MESH)
C
C
C-----
C   R E T U R N
C-----
C
C   END
C
```

```
C |-----|
C |
C |  CALCULATE NON-ADVECTIVE TERM
C |
C |          IN EULERIAN PHASE
C |
C |          <<<<<<< MOMENTUM >>>>>>>
C |-----|
C
C  SUBROUTINE MOMRHS
C
C  IMPLICIT REAL*8(A-H,O-Z)
C
C  COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
&    YVIS(1000),FR(1000),FM(1000),FE(1000),
&    GR(1000),GM(1000),GE(1000)
C
C  COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
&    R1,R2,R3,R4,R5,R6,
&    ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C  COMMON /CMGRAD/
YRGRD(1000),YMGRD(1000),YEGRD(1000),ZGEOM,NGEOM
C
C-----C
C
C  FPL = 0.00D0
C
C  DO 10 I=1,MM1
    IP1 = I + 1
    FPR = ( 0.50D0*(YP(IP1)+YP(I)) + YVIS(IP1) ) * CK
    POSI = 1.0D0 / ( DFLOAT(I)-0.50D0 ) * DRI
    YMGRD(I) = - FPR + FPL
&    - ZGEOM*YM(I)*YU(I)*POSI*DT
    YM(I) = YM(I) + YMGRD(I)
    FPL = FPR
10 C O N T I N U E
C
C  FPR = ( YP(MESH) + YVIS(MESH) ) * CK
    POSI = 1.0D0 / ( DFLOAT(MESH)-0.50D0 ) * DRI
    YMGRD(MESH) = - FPR + FPL
&    - ZGEOM*YM(MESH)*YU(MESH)*POSI*DT
    YM(MESH) = YM(MESH) + YMGRD(MESH)
C
C  YM(1) = 0.00D0
C
C
C-----C
C  R E T U R N
C-----C
C
C  END
C
```

```

C |-----|
C |
C | CALCULATE NON-ADVECTIVE TERM
|
C |           IN EULERIAN PHASE
C |
C |           <<<<<<<< TOTAL ENERGY >>>>>>>>
C |-----|
C
C SUBROUTINE ENERHS
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
& YVIS(1000),FR(1000),FM(1000),FE(1000),
& GR(1000),GM(1000),GE(1000)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C COMMON /CMGRAD/
YRGRD(1000),YMGRD(1000),YEGRD(1000),ZGEOM,NGEOM
C
C-----
C
C CKQ = 0.250D0 * CK
C
C FPL = 0.00D0
DO 10 I=1,MM1
  IP1 = I + 1
  DIC = ( DFLOAT(I) - 0.50D0 ) * DR
  POSI = 1.0D0 / DIC
  FPR = ( YP(IP1)+YP(I)+2.0D0*YVIS(IP1) )
& * ( YU(IP1)+YU(I) ) * CKQ
  Z10 = YP(I) + 0.5D0*(YVIS(IP1)+YVIS(I))
  YEGRD(I) = (FPL-FPR)
& - ZGEOM*(YE(I)+Z10)*YU(I)*POSI*DT
  YE(I) = YE(I) + YEGRD(I)
  FPL = FPR
10 C O N T I N U E
C
C DIC = ( DFLOAT(MESH) - 0.50D0 ) * DR
C POSI = 1.0D0 / DIC
C FPR = 2.0D0*( YP(MESH)+YVIS(MESH) )
& * ( YU(MESH)+YU(MM1) ) * CKQ
C Z10 = YP(MESH) + YVIS(MESH)
C YEGRD(I) = (FPL-FPR)
& - ZGEOM*(YE(MESH)+Z10)*YU(MESH)*POSI*DT
C YE(MESH) = YE(MESH) + YEGRD(MESH)
C
C-----
C R E T U R N
C-----
C END

```

```
C
C |-----|
C |
C | CALCULATE FIRST ORDER DERIVATIVE OF PROFILE
C |
C |
C | AFTER EULERIAN PHASE
C |
C |-----|
C
C SUBROUTINE NEWGRD(YY,YG)
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C DIMENSION YY(1),YG(1)
C
C-----
C
C DO 10 I=2,MM2
C   IP1 = I + 1
C   IM1 = I
C   YY(I) = YY(I) + ( YG(IP1)-YG(IM1) ) * 1.0D0 * DRI
10 C O N T I N U E
C
C   YY(1) = 0.00D0
C   YY(MM1) = 0.00D0
C   YY(MESH) = 0.00D0
C
C-----
C   R E T U R N
C-----
C
C END
C
```

```
C |-----|
C |
C |
C | CALCULATE FLUX
C |
C |
C |-----|
C
C SUBROUTINE FLCLC(FRR,YR,GR,YU)
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C DIMENSION FRR(1),YR(1),GR(1),YU(1)
C
C-----C
C
C DO 10 I=1,MM1
C
C IP1 = I + 1
C UF = 0.50D0 * ( YU(IP1) + YU(I) ) * CK
C
C
C UF2 = UF*UF
C UF3 = UF*UF2
C UF4 = UF*UF3
C
C QR1 = ( 0.125D0*UF-0.375D0*UF2+0.5D0*UF3-0.5D0*UF4 )
C QR2 = ( 0.250D0*UF-0.500D0*UF2+UF3/3.0D0 )
C QR3 = ( 0.500D0*UF-0.500D0*UF2 )
C QR4 = ( UF )
C
C QF1 = ( QR1 - QR2 ) * DR
C QF2 = ( QR1 - 2.0D0*QR2 + QR3 ) * DR
C QF3 = ( - 2.0D0*QR1 + 3.0D0*QR2 )
C QF4 = ( 2.0D0*QR1 - 3.0D0*QR2 + QR4 )
C
C FRR(I) = QF1 * GR(IP1)
& + QF2 * GR(I)
& + QF3 * YR(IP1)
& + QF4 * YR(I)
C
C 10 CONTINUE
C
```

```
C.....BOUNDARY FLUX ( MESH+1/2 )
C
C   UF = 0.50D0 * ( YU(MESH) + YU(MM1) ) * CK
C
C   UF2 = UF*UF
C   UF3 = UF*UF2
C   UF4 = UF*UF3
C
C   QR1 = ( 0.125D0*UF-0.375D0*UF2+0.5D0*UF3-0.5D0*UF4 )
C   QR2 = ( 0.250D0*UF-0.500D0*UF2+UF3/3.0D0 )
C   QR3 = ( 0.500D0*UF-0.500D0*UF2 )
C   QR4 = (   UF   )
C
C   QF1 = (   QR1 - QR2 ) * DR
C   QF2 = (   QR1 - 2.0D0*QR2 + QR3 ) * DR
C   QF3 = ( - 2.0D0*QR1 + 3.0D0*QR2 )
C   QF4 = (   2.0D0*QR1 - 3.0D0*QR2 + QR4 )
C
C   FRR(MESH) = QF3 * YR(MESH)
C   &          + QF4 * YR(MESH)
C
C
C   -----
C   R E T U R N
C   -----
C
C   END
C
```

```
C |-----|
C |
C |
C | CALCULATE ALL THE RIGHT HAND SIDE TERMS IN CIP
C |
C |
C |-----|
C
C   SUBROUTINE CIPRHS(FR,FRR,GR,YR)
C
C   IMPLICIT REAL*8(A-H,O-Z)
C
C   COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
&      R1,R2,R3,R4,R5,R6,
&      ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C   DIMENSION FR(1),FRR(1),GR(1),YR(1)
C
C
C   DO 20 I=2,MM1
C
C     IP1 = I + 1
C     IM1 = I - 1
C
C     FR(I) = R1 * GR(IP1)
&      + R3 * GR(IM1)
&      + R4 * YR(IP1)
&      + R5 * YR(I)
&      + R6 * YR(IM1)
&      + FRR(IM1) - FRR(I)
20 C O N T I N U E
C
C----- BOUNDARY CONDITION (LEFT) -----
C
C   FR(1) = R1 * GR(2)
&      + R4 * YR(2)
&      + R5 * YR(1)
&      + R6 * YR(1)
&      - FRR(1)
C
C
C----- BOUNDARY CONDITION (RIGHT) -----
C
C   FR(MESH) = R4 * YR(MESH)
&      + R5 * YR(MESH)
&      + R6 * YR(MM1)
&      - FRR(MESH) + FRR(MM1)
C
C-----
C   R E T U R N
C-----
C   END
```



```

C |-----|
C |
C | SOLVE CIP EQUATION |
C | (TRIDIAGONAL FORM IS TRANSFORMED INTO EXPLICIT DIFFERENCE) |
C |
C | ZBB : BOUNDARY CONDITION |
C |
C | GRN : PREDICTED GRADIENT |
C | YRN : PREDICTED QUANTITY |
C |-----|
C
C SUBROUTINE CIPCLC(YR,GR,FR,YU)
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C DIMENSION GRN(1000),YRN(1000),YR(1),GR(1),FR(1),YU(1)
C-----
C
C DO 10 I=2,MM1
C
C IP1 = I + 1
C IM1 = I - 1
C
C UK = DABS( YU(I) ) * CK
C UK2 = UK * UK
C UK3 = UK2* UK
C
C CX = 1.0D0 - UK
C CX2 = CX * CX
C CX3 = CX2* CX
C
C S1 = ( 1.0D0 - 4.0D0*UK + 3.0D0*UK2 )
C S2 = ( - 2.0D0*UK + 3.0D0*UK2 )
C S3 = ( 6.0D0*UK - 6.0D0*UK2 ) * DRI
C S4 = ( - 6.0D0*UK + 6.0D0*UK2 ) * DRI
C
C T1 = ( CX3 - CX2 ) * DR
C T2 = ( CX3 -2.0D0*CX2+CX ) * DR
C T3 = (-2.0D0*CX3+3.0D0*CX2 )
C T4 = ( 2.0D0*CX3-3.0D0*CX2+1.0D0)
C T5 = ( UK3 - UK2 ) * DR
C T6 = ( UK3 -2.0D0*UK2+UK ) * DR
C T7 = (-2.0D0*UK3+3.0D0*UK2 )
C T8 = ( 2.0D0*UK3-3.0D0*UK2+1.0D0)
C

```

```

C      IF(YU(I).GE.0.0D0) THEN
C          GRN(I) = S1 * GR(I)
&          + S2 * GR(IM1)
&          + S3 * YR(I)
&          + S4 * YR(IM1)
C
C          YRN(I)=T1*GR(I)+T2*GR(IM1)
&          +T3*YR(I)+T4*YR(IM1)
C
C      ELSE
C          GRN(I) = S2 * GR(IP1)
&          + S1 * GR(I)
&          + S3 * YR(IP1)
&          + S4 * YR(I)
C
C          YRN(I)=T5*GR(IP1)+T6*GR(I)
&          +T7*YR(IP1)+T8*YR(I)
C
C      ENDIF
C 10  C O N T I N U E
C      GRN(1) = 0.00D0
C      GRN(MESH) = 0.00D0
C      YRN(1) = YR(1)
C      YRN(MESH) = YR(MESH)
C..... DATA EXCHANGE
C      DO 30 I=1,MESH
C          GR(I) = GRN(I)
C          YR(I) = YRN(I)
C 30  CONTINUE
C
```

```
C..... USE THE PREDICTED VALUES TO SOLVE CIP
C
  CGRR = 18.0D0/192.0D0
  DO 40 I=2,MM1
    IP1 = I + 1
    IM1 = I - 1
    FR(I) = FR(I) + CGR * ( GR(IP1) - GR(IM1) )
    &      -CGRR*( YR(IP1)-2.0D0*YR(I)+YR(IM1) )
40  CONTINUE
C
  FR(1) = FR(1)-CGRR*(YR(2)-YR(1))
  FR(MESH) = FR(MESH)-CGRR*(YR(MM1)-YR(MESH))
C
  DO 50 I=1,MESH
    YR(I)=FR(I)
50  CONTINUE
C
C
C  -----
  R E T U R N
  -----
C
C
C  END
C
```

```
C |-----|
C |
C |
C | GRAIENT CORRECTION
C |
C |
C |-----|
C
C SUBROUTINE CORRCT(YZ,GZ,DRZ,MESHZ)
C
C REAL*8 YZ(1000),GZ(1000),DRZ
C INTEGER MESHZ
C-----C
C
C MM1 = MESHZ - 1
C
C DO 10 I=2,MM1
C   IP1 = I + 1
C   IM1 = I - 1
C   GL = ( YZ(I) - YZ(IM1) ) / DRZ
C   GR = ( YZ(IP1) - YZ(I) ) / DRZ
C   AGL = ABS(GL)
C   AGR = ABS(GR)
C   IF (AGR.GT.AGL) THEN
C     GZ(I) = GL
C   ELSE
C     GZ(I) = GR
C   ENDIF
C 10 CONTINUE
C
C   GZ(1) = 0.00D0
C   GZ(MESHZ) = 0.00D0
C
C-----C
C   R E T U R N
C-----C
C
C END
C
```

```
C |-----|
C |
C | CALCULATE TEMPORARY VELOCITY ( U-OLD + U-NEW ) / 2
I |
C |
C | IN ORDER TO CONSERVE THE ENERGY
I |
C |
C |-----|
C
C SUBROUTINE EUCLC
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
& YVIS(1000),FR(1000),FM(1000),FE(1000),
& GR(1000),GM(1000),GE(1000)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C-----C
C
C DO 10 I=1,MM1
C IF (YR(I).GT.0.00D0) THEN
C YU(I) = 0.50D0 * ( YU(I) + YM(I)/YR(I) )
C ELSE
C YU(I) = 0.00D0
C ENDIF
10 C O N T I N U E
C
C YU(MESH) = YU(MM1)
C
C-----C
C R E T U R N
C-----C
C
C END
C
```

```
C |-----|
C |
C |
C | CALCULATE VELOCITY FROM MOMENTUM AND DENSITY
I
C |
C |
C |
C |-----|
C
C SUBROUTINE NUCLC
C
C IMPLICIT REAL*8(A-H,O-Z)
C
C COMMON /CMPHYS/ YR(1000),YM(1000),YE(1000),YP(1000),YU(1000),
& YVIS(1000),FR(1000),FM(1000),FE(1000),
& GR(1000),GM(1000),GE(1000)
C
C COMMON /CMCNST/ DR,DT,CK,GAMMA,CVS,GM1,GP1H,DRI,CGR,TIME,
& R1,R2,R3,R4,R5,R6,
& ILOOP,ISTEP,MESH,MM1,MM2,MP1,N
C
C-----C
C
C DO 10 I=1,MM1
C IF (YR(I).GT.0.00D0) THEN
C YU(I) = YM(I) / YR(I)
C ELSE
C YU(I) = 0.00D0
C ENDIF
10 C O N T I N U E
C
C YU(MESH) = YU(MM1)
C
C-----C
C R E T U R N
C-----C
C
C END
C
```

```
C |-----|
C |
C |
C | OUTPUT
C |
C |
C |-----|
C
C SUBROUTINE YOUT(YZ,M,N,ICH)
C
C REAL*8 YZ(1000)
C INTEGER M,N,ICH(2)
C
C-----C
C
C WRITE(6,1000) ICH
1000 FORMAT(1H ,//,30X,'*** ',2A4,' ***')
C
C WRITE(6,1010) (YZ(I),I=M,N)
1010 FORMAT(1H ,1P,10E12.5)
C
C-----C
C RETURN
C-----C
C
C END
```