

A numerical algorithm for the solution of
systems of conservation laws
in two dimensions.

M. J. Baines

Numerical Analysis Report 2/80

Summary

A numerical algorithm for the solution of the two-dimensional incompressible fluid flow equations in conservation law form is described, based on the successful one-dimensional algorithm of P. L. Roe at RAE Bedford. The main features are the weighted allocation of flux quantities to nodes and a switching device to reduce oscillations.

Contents

1. Introduction
2. The one-dimensional method
3. The two-dimensional method
4. Stability of the two-dimensional method
5. Switching in two dimensions
6. A test problem
7. The choice of λ
8. The method in three dimensions
9. Boundary conditions
10. Conclusions and proposed extensions

References

Programs

§1 Introduction

This report describes work carried out at RAE Farnborough during a sabbatical term (Autumn 1979) from the University of Reading. The work arose from developments in solving the Euler equations for time-dependent compressible fluid flow being carried out at RAE Bedford. P. Roe at Bedford had investigated the one-dimensional Euler equations from a fundamental point of view and had proposed a novel approach to the construction of numerical algorithms. He had also devised a switching mechanism which had been very successful in suppressing unwanted oscillations near discontinuities. The resulting method had been applied successfully to test problems of various kinds, including a problem involving two intersecting shockwaves and a shock tube problem recently surveyed by G. Sod (1977).

To apply the method in more than one space-dimension a form of operator splitting in time had been used. A program to calculate the pressure distribution over an aerofoil had been written and found to be satisfactory although time-consuming. Another program, to calculate the flow past a flat faced step had run into difficulties.

It seemed worthwhile to try and apply the fundamental ideas of the successful one-dimensional method directly in two or three dimensions. It was hoped that the results of such an investigation would be useful in assessing the validity of operator splitting in several dimensions using Roe's one-dimensional method.

It is convenient to describe here the elements of the one-dimensional method prior to generalising to two or more dimensions.

§2 The one-dimensional method

The main features of the one-dimensional algorithm may be set down as (1) conservation on a regular grid, (2) propagation of waves in appropriate characteristic directions and (3) second order accuracy with freedom from spurious oscillations near discontinuities. We shall describe here the algorithm in relation to the simple scalar, but possibly non-linear equation

$$u_t + f_x = 0, \quad (1)$$

mentioning at this point that the extension to a system has been successfully achieved taking into account the different directions in which different types of waves travel. In particular, in the Euler equations, for which

$$u = \begin{pmatrix} \rho \\ \rho u \\ e + p \end{pmatrix} \quad f = \begin{pmatrix} \rho u \\ p + \rho u^2 \\ u(e + p) \end{pmatrix},$$

in the usual notation, differenced quantities are resolved into components

along the eigenvectors

$$\begin{pmatrix} \rho \\ \rho(u + c) \\ e + p + \rho uc \end{pmatrix} \quad \begin{pmatrix} \rho \\ \rho(u - c) \\ e + p - \rho uc \end{pmatrix} \quad \begin{pmatrix} \rho \\ \rho u \\ \frac{1}{2}\rho u^2 \end{pmatrix}$$

two wave directions and the stream direction.

An alternative statement of equation (1) is

$$\frac{\partial}{\partial t} \left(\int_{x_j}^{x_{j+1}} u dx \right) = -(f_{j+1} - f_j) \quad (2)$$

and a time discretization of this equation gives

$$\Delta \left(\frac{1}{\Delta x} \int_{x_j}^{x_{j+1}} u dx \right) = - \frac{\Delta t}{\Delta x} (f_{j+1} - f_j) = I_j \quad (3)$$

say. The quantity I_j on the r.h.s. of (3) is the familiar one which is used to increment values of u in many well-known algorithms. For example, in the predictor stage of MacCormack's algorithm, it is used to increment u_{j+1} if $f'(u) > 0$ and to decrement u if $f'(u) < 0$ (1st order upwind differencing). The r.h.s. of (3) can also be written

$$= \frac{\Delta t}{\Delta x} \left(\frac{f_{j+1} - f_j}{u_{j+1} - u_j} \right) (u_{j+1} - u_j) \quad (4)$$

whence it can be related to the 'characteristic' speed $f'(u)$. (5)

The reason for assigning I to the different end-points of the interval is then clear. At the corrector stage of the MacCormack algorithm and also in other 2nd order algorithms a further distribution of the r.h.s. of (3) (evaluated from predicted or "first stage" values) is made. Second order accuracy results from the subtraction of quantities whose difference is already first order. I_j

A way of achieving a 2nd order algorithm in one step while retaining conservation in a natural way is to assign the increment I in (3) to the ends of the interval (x_j, x_{j+1}) with different weights α and β . The weights are to be determined by imposing conditions on the method to ensure second order accuracy. This is illustrated first on the linear scalar equation

$$u_t + au_x = 0 \quad (6)$$

or

$$\Delta \left(\frac{1}{\Delta x} \int_{x_j}^{x_{j+1}} u dx \right) = -a \frac{\Delta t}{\Delta x} (u_{j+1} - u_j) = -v \Delta u \quad (7)$$

where $v = a \frac{\Delta t}{\Delta x}$ is the CFL number (cf. (3)).

To achieve 2nd order accuracy it is sufficient to ensure that the algorithm provides exact solutions for the basic polynomials $u = 1$, $u = x$, $u = x^2$. The case $u = 1$ is satisfied automatically, since no changes occur. In the cases $u = x$ and $u = x^2$, consider the point 0 in fig 1. and the adjacent intervals ZO, OA. The increment to point 0 is



fig 1.

α times $(-v\Delta u)_{OA}$ plus β times $(-v\Delta u)_{ZO}$.

The increment is therefore

$$-v\alpha(u_A - u_0) - v\beta(u_0 - u_Z). \quad (8)$$

Without loss of generality we can take 0 to be the origin.

In the case $u = x$, (8) becomes

$$-a \frac{\Delta t}{\Delta x} (\alpha + \beta) \Delta x \quad (9)$$

and the exact increment is $-a\Delta t$ (because the data is merely convected with speed a), giving

$$\alpha + \beta = 1 \quad (10)$$

Similarly, in the case $u = x^2$, (8) becomes

$$-a \frac{\Delta t}{\Delta x} [\alpha(\Delta x)^2 - \beta(\Delta x)^2] \quad (11)$$

and the exact increment is $a^2(\Delta t)^2$, so that

$$\alpha - \beta = -v \quad (12)$$

From (10) and (12),

$$\alpha = \frac{1}{2}(1 - v) \quad \beta = \frac{1}{2}(1 + v). \quad (13)$$

u_0 is therefore incremented by the quantity

$$\frac{1}{2}(1 - v)[-v(u_A - u_0)] + \frac{1}{2}(1 + v)[-v(u_0 - u_Z)] \quad (14)$$

giving the rule

$$u_0^{n+1} = u_0 - \frac{1}{2}v(u_A - u_Z) + \frac{1}{2}v^2(u_A - 2u_0 + u_Z), \quad (15)$$

the one-step Lax-Wendroff algorithm.

We can allow the r.h.s. of (7) to increment a different pair of points. Suppose that the increment $(-v\Delta u)$ is assigned to u_{j+1} and u_{j+2} with weights γ and δ . A similar calculation based on the requirements of 2nd order accuracy leads to

$$\gamma = \frac{1}{2}(3 - v) \qquad \delta = \frac{1}{2}(-1 + v), \qquad (16)$$

corresponding to the upwind algorithm of Warming and Beam. Conservation is assured because all quantities $-v\Delta u$ (or, more generally, $-\frac{\Delta t}{\Delta x}(f_{j+1} - f_j)$) are allocated at the next time level, so that

$$\sum u^{n+1} = \sum u^n - \frac{\Delta t}{\Delta x} \left[f_{j_{\max}}^n - f_{j_{\min}}^n \right]$$

A difficulty with second order algorithms is the appearance of unwanted oscillations near to discontinuities. In first order algorithms the phenomena can be avoided by building monotonicity into the scheme; monotonicity in this sense means that monotonic data at any time level imply monotonic values at the next time level. However, it has been shown by Godunov that there are no second order algorithms which preserve monotonicity.

One way of dealing with oscillations is to introduce a switching device which operates only in non-smooth parts of the flow and leaves the algorithm elsewhere unaffected. The reduction in accuracy is of small significance near discontinuities. P. Roe at Bedford has devised such a switching device which is easily implemented in terms of increments and has a sound mathematical basis. There is a formal generalisation of this device to several dimensions, although the mathematical basis needs further work (see §5).

The ideas in the above work have straightforward generalisations to non-linear equations and to systems of hyperbolic conservation laws.

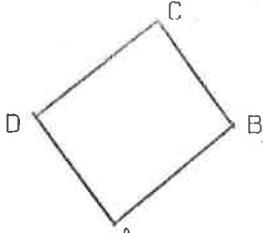
§3. The two-dimensional method

We attempt a 2-D generalisation of the ideas in §1. Equation (6) now becomes

$$u_t + f_x + g_y = 0 \quad (20)$$

or, in integral form,

$$\frac{\partial}{\partial t} \left[\int_{\Omega} u d\Omega \right] = - \int_{\Omega} (f_x + g_y) d\Omega = \int (f dy - g dx) \quad (21)$$



Evaluating the r.h. integral on the quadrilateral in fig 2., using the trapezium rule, we obtain

$$\begin{aligned} & \frac{1}{2} [(f_A + f_B)(y_B - y_A) + (f_B + f_C)(y_C - y_B) + (f_C + f_D)(y_D - y_C) + (f_D + f_A)(y_A - y_D)] \\ & - \frac{1}{2} [(g_A + g_B)(x_B - x_A) + (g_D + g_C)(x_C - x_B) + (g_C + g_D)(x_D - x_C) + (g_D + g_A)(x_A - x_D)] \\ = & \frac{1}{2} [(f_A - f_C)(y_B - y_D) - (f_B - f_D)(y_A - y_C)] \\ & - \frac{1}{2} [(g_A - g_C)(x_B - x_D) - (g_B - g_D)(x_A - x_C)] \quad (22) \end{aligned}$$

The area of the quadrilateral ABCD is $\Delta\Omega = \frac{1}{2} [(x_A - x_C)(y_B - y_D) - (x_B - x_D)(y_A - y_C)]$ and (21) can therefore be written in the form

$$\Delta \left[\frac{1}{\Delta\Omega} \int u d\Omega \right] = \Delta t \frac{(f_A - f_C)(y_B - y_D) - (f_B - f_D)(y_A - y_C) - (g_A - g_C)(x_B - x_D) + (g_B - g_D)(x_A - x_C)}{(x_A - x_C)(y_B - y_D) - (x_B - x_D)(y_A - y_C)} \quad (23)$$

and we have carried out the time discretisation. Note that only diagonal differences are involved.

If ABCD is a rectangle of sides $\Delta x, \Delta y$, this reduces to

$$\Delta \left[\frac{1}{\Delta x \Delta y} \int u d\Omega \right] = \frac{\Delta t}{2\Delta x} (f_A - f_C - f_B + f_D) + \frac{\Delta t}{2\Delta y} (g_A - g_C + g_B - g_D) = I, \quad (24)$$

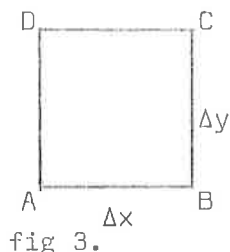


fig 3.

say. The terms in brackets on the r.h. side of (24) approximate $f'(u)$ and $g'(u)$ at the mid-point of the cell.

We seek now an algorithm which, for the model equation

$$u_t + au_x + bu_y = 0 \quad (25)$$

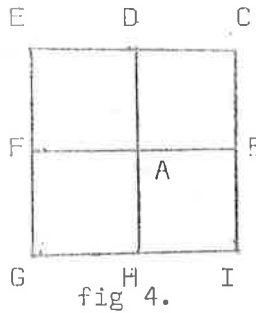
in the discretised form

$$\begin{aligned} \Delta t \left[\frac{1}{\Delta x \Delta y} \int_{\square} u d\Omega \right] &= \frac{a\Delta t}{2\Delta x} (u_A - u_C - u_B + u_D) + \frac{b\Delta t}{2\Delta y} (u_A - u_C + u_B - u_D) \\ &= \frac{1}{2}v_1 (u_A - u_C - u_B + u_D) + \frac{1}{2}v_2 (u_A - u_C + u_B - u_D) \end{aligned}$$

$$\text{where } v_1 = a\Delta t/\Delta x, \quad v_2 = b\Delta t/\Delta y, \quad (26)$$

is 2nd order accurate, i.e. it is exact for the specimen polynomials $u = 1, u = x, u = y, u = x^2, u = y^2, u = xy$. We suppose that the quantity on the r.h.s. of (26) is to be assigned to the four corners A, B, C, D of the rectangle in fig 3., with weights $\alpha, \beta, \gamma, \delta$.

At first sight there appear to be 6 conditions to be satisfied by the four weights $\alpha, \beta, \gamma, \delta$. This is not the case, for two reasons. First, the case $u = 1$ is satisfied automatically as in one-dimension. Secondly, the function $u = ay - bx$ is constant in the direction of the stream and is therefore an exact solution: it follows that if $\alpha, \beta, \gamma, \delta$ are such that the algorithm is exact for $u = x$, then it is also exact for $u = y$. Moreover, $u = (ay - bx)^2$ is also an exact solution, so that exactness for $u = x^2, u = y^2$ implies exactness for $u = xy$. There are therefore only 3 conditions to be met by $\alpha, \beta, \gamma, \delta$ and there will be one free parameter.



Consider the point A and its four adjacent rectangles (see fig 4.). The increment to u_A is α times the r.h.s. of (26) plus three corresponding terms from the three other rectangles ADEF, AFGH and AHIB. We may take the origin of coordinates at A. For $u = x$, we have from the rectangle ABCD a contribution $-\alpha a \Delta t$. The contribution from all four rectangles is

$$-(\alpha + \beta + \gamma + \delta)a\Delta t \quad (27)$$

and the exact solution is $-a\Delta t$. Hence we require that

$$\alpha + \beta + \gamma + \delta = 1, \quad (28)$$

and this condition covers the case $u = y$. First order accuracy is therefore obtained by any set of weights $\alpha, \beta, \gamma, \delta$, satisfying (28), in particular $\alpha = 1, \beta = \gamma = \delta = 0$ or a cyclic permutation of these. We obtain first order accuracy by assigning the contribution to any corner. Note also that the same result (28) is obtained for an arbitrary quadrilateral (see (23)), so the grid does not have to be regular at this stage.

For $u = x^2$, the r.h.s. of (26) for the rectangle ABCD reduces to $-a\Delta t(\Delta x)^2$ while for the rectangle ADEF it reduces to $a\Delta t(\Delta x)^2$. The total contribution from the four rectangles in this case is

$$(-\alpha + \beta + \gamma - \delta)a\Delta t(\Delta x)^2 \quad (29)$$

and the exact solution is $a^2\Delta t^2$. Hence we require also that

$$-\alpha + \beta + \gamma - \delta = v_1 \quad (30)$$

Similarly, for $u = y^2$, the condition to be satisfied is

$$-\alpha - \beta + \gamma + \delta = v_2 \quad (31)$$

Solving (28), (30), (31) leads to a one parameter family of solutions for $\alpha, \beta, \gamma, \delta$. Let the free parameter be λ and add a fourth equation

$$\alpha - \beta + \gamma - \delta = \lambda \quad (32)$$

Solving the four equations (28), (30), (31), (32) for $\alpha, \beta, \gamma, \delta$ gives

$$\left. \begin{aligned} \alpha &= \frac{1}{4}(1 + \lambda) - \frac{1}{4}(v_1 + v_2) \\ \beta &= \frac{1}{4}(1 - \lambda) - \frac{1}{4}(v_2 - v_1) \\ \gamma &= \frac{1}{4}(1 + \lambda) + \frac{1}{4}(v_1 + v_2) \\ \delta &= \frac{1}{4}(1 - \lambda) + \frac{1}{4}(v_2 - v_1) \end{aligned} \right\} \quad (33)$$

Second order accuracy is obtained by using these weights with any choice of λ . There is no reason why λ should not vary from cell to cell since the conditions (28), (30), (31) for second order accuracy are satisfied whatever value is taken by λ . It can also be shown that the same weights are obtained if the mesh is taken to consist of parallelograms rather than rectangles.

A diagonal bias is again apparent in (33). Indeed, if we take a square mesh with $\Delta x = \Delta y = h$ and rotate the grid through 45° to form a mesh of diamonds by means of the transformation

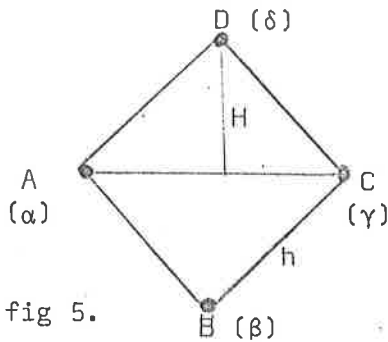


fig 5.

$$\begin{aligned} X &= \frac{1}{\sqrt{2}}(x + y) & Y &= \frac{1}{\sqrt{2}}(y - x) \\ A &= \frac{1}{\sqrt{2}}(a + b) & B &= \frac{1}{\sqrt{2}}(b - a) \\ F &= \frac{1}{\sqrt{2}}(f + g) & G &= \frac{1}{\sqrt{2}}(g - f) \end{aligned} \quad (34)$$

(see fig 5.) the r.h.s. of (24) becomes

$$\frac{\Delta t}{H} (F_A - F_C + G_B - G_D) \quad (35)$$

and the weights become

$$\begin{aligned} \alpha &= \frac{1}{4}(1 + \lambda) - \frac{1}{2}A\Delta t/H \\ \beta &= \frac{1}{4}(1 - \lambda) - \frac{1}{2}B\Delta t/H \\ \gamma &= \frac{1}{4}(1 + \lambda) + \frac{1}{2}A\Delta t/H \\ \delta &= \frac{1}{4}(1 - \lambda) + \frac{1}{2}B\Delta t/H \end{aligned} \quad (36)$$

where $H = h/\sqrt{2}$.

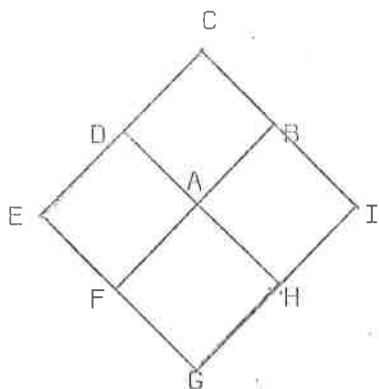


fig 6.

The quantity u_A is incremented by a quantity which corresponds to the 2-D Lax Wendroff scheme (ref. 9, p. 350) together with a term depending on λ , namely

$$\frac{\lambda \Delta t}{4H} \left[A(u_I + u_D + u_F - u_E - u_H - u_B) + B(u_G + u_B + u_D - u_C - u_F - u_H) \right] \quad (38)$$

(see fig 6.) whose significance is not yet clear. The algorithm reduces to Lax-Wendroff when $\lambda = 0$.

For a linear system of equations, the results go through with the constants a, b, c, d, A, B, C, D becoming matrices. Moreover the $\alpha, \beta, \gamma, \delta$ are also matrices and so is the free parameter λ . A natural extension to non-linear f and g , or F and G , is obtained by replacing a, A etc. by the appropriate Jacobians, or their approximations $(f_A - f_C)/(u_A - u_C)$ etc.

The stability analysis which follows is, however, only valid for scalar problems.

§4. Stability of the 2-D Method

The scheme given the weights (36) in the diagonally orientated square mesh is precisely the 2-D Lax Wendroff scheme as given in e.g., ref.9, p.362, but with the additional term (38). The amplification factor is therefore as in ref.9 p.362 equⁿ. 13.13 together with the additional term

$$-i\lambda(\cos \alpha - \cos \beta) \frac{\Delta t}{H} (A \sin \alpha + B \sin \beta), \quad (39)$$

where $\alpha = k_x H$, $\beta = k_y H$, k_x and k_y being the Fourier frequencies.

We therefore have the amplification factor

$$g = 1 - i \frac{\Delta t}{H} \{(1 + \lambda) \cos \alpha + (1 - \lambda) \cos \beta\} (A \sin \alpha + B \sin \beta) - 2 \left(\frac{\Delta t}{H} \right)^2 (A \sin \alpha + B \sin \beta)^2. \quad (40)$$

The modulus squared of the amplification factor is

$$\bar{g}g = \{1 - 2(\mu_1 \sin \alpha + \mu_2 \sin \beta)^2\}^2 + \{(1 + \lambda) \cos \alpha + (1 - \lambda) \cos \beta\}^2 (\mu_1 \sin \alpha + \mu_2 \sin \beta)^2 \quad (41)$$

$$\text{where } \mu_1 = \frac{A \Delta t}{H}, \quad \mu_2 = \frac{B \Delta t}{H}, \quad \text{i.e.} \quad (42)$$

$$\bar{g}g = 1 + \tau^2 [4(\mu_1 \sin \alpha + \mu_2 \sin \beta)^2 + \{(1 + \lambda) \cos \alpha + (1 - \lambda) \cos \beta\}^2 - 4] \quad (43)$$

where $\tau^2 = (\mu_1 \sin \alpha + \mu_2 \sin \beta)^2$. The condition $\bar{g}g < 1 \quad \forall$ real α, β

therefore reduces to the condition

$$G(\alpha, \beta) = 4(\mu_1 \sin \alpha + \mu_2 \sin \beta)^2 + \{(1 + \lambda) \cos \alpha + (1 - \lambda) \cos \beta\}^2 - 4 \leq 0 \quad (44)$$

for all $\cos \alpha, \cos \beta$ between -1 and $+1$.

Rewrite the middle term as

$$\begin{aligned} & (1 + \lambda^2)(\cos^2 \alpha + \cos^2 \beta) + 2\lambda(\cos^2 \alpha - \cos^2 \beta) + 2(1 - \lambda^2) \cos \alpha \cos \beta \\ &= (\lambda^2 - 1)(\cos^2 \alpha + \cos^2 \beta) + 2(\cos^2 \alpha + \cos^2 \beta) + 2\lambda(\cos^2 \alpha - \cos^2 \beta) + 2(1 - \lambda^2) \cos \alpha \cos \beta \\ &= 4 - 2\sin^2 \alpha - 2\sin^2 \beta - 2\lambda(\sin^2 \alpha - \sin^2 \beta) - (1 - \lambda^2)(\cos \alpha - \cos \beta)^2 \quad (45) \end{aligned}$$

whence

$$\begin{aligned} G(\alpha, \beta) &= \sin^2 \alpha [4\mu_1^2 - 2(1 + \lambda)] + \sin^2 \beta [4\mu_2^2 - 2(1 - \lambda)] \\ &\quad + 8\mu_1 \mu_2 \sin \alpha \sin \beta - (1 - \lambda^2)(\cos \alpha - \cos \beta)^2. \quad (46) \end{aligned}$$

Consider $\alpha = 0$, $\beta = \pi$. Then for $G(\alpha, \beta)$ to be non-positive it is necessary that

$$|\lambda| \leq 1. \quad (47)$$

In that case the final term in (46) is always negative. The remaining terms constitute a quadratic form in $\sin \alpha$, $\sin \beta$, which is non-positive if

$$(8\mu_1\mu_2)^2 \leq 4[4\mu_1^2 - 2(1 + \lambda)][4\mu_2^2 - 2(1 - \lambda)]. \quad (48)$$

This simplifies to

$$\frac{\mu_1^2}{1 + \lambda} + \frac{\mu_2^2}{1 - \lambda} \leq \frac{1}{2} \quad (49)$$

and this condition, together with (47), is sufficient for $G(\alpha, \beta)$ to be non-positive. The two conditions together are also necessary, since otherwise $G(\alpha, \beta)$ would be positive for some small (α, β) such that the cosine term is negligible.**

Reverting to the A, B notation, we have shown that necessary and sufficient conditions for the stability of the method are

$$\frac{(A\Delta t/H)^2}{1 + \lambda} + \frac{(B\Delta t/H)^2}{1 - \lambda} \leq \frac{1}{2} \quad (50)$$

$$|\lambda| \leq 1 \quad (51)$$

The conditions to be satisfied for $A\Delta t/H$ and $B\Delta t/H$ are shown in fig 7. Condition (50) requires the point $(A\Delta t/H, B\Delta t/H)$ to lie within an ellipse (for fixed λ), which can be shown to touch the four straight lines

$$\pm \frac{A\Delta t}{H} \pm \frac{B\Delta t}{H} = 1 \quad (52)$$

However, there is no reason why λ should not vary with A and B , as long as condition (51) is met.

Rewriting the inequality (49) as

$$(1 - \lambda)\mu_1^2 + (1 + \lambda)\mu_2^2 \leq \frac{1}{2}(1 - \lambda^2), \quad (53)$$

** I am indebted to P. L. Roe for the conditions found here.

using (47), and rearranging, we obtain

$$\lambda^2 - 2\lambda(\mu_1^2 - \mu_2^2) + 2\mu_1^2 + 2\mu_2^2 - 1 \leq 0.$$

For this condition to be satisfied, λ must lie between the real roots

$$(\mu_1^2 - \mu_2^2) \pm \sqrt{(\mu_1^2 - \mu_2^2)^2 + 1 - 2\mu_1^2 - 2\mu_2^2} \quad (54)$$

For the roots to be real, the expression inside the square root has to be positive. This is equivalent to the point (μ_1, μ_2) lying inside a diamond, (fig 7.), since

$$\begin{aligned} & (\mu_1^2 - \mu_2^2)^2 + 1 - 2\mu_1^2 - 2\mu_2^2 \\ &= (\mu_1^2 + \mu_2^2)^2 - 4\mu_1^2\mu_2^2 + 1 - 2\mu_1^2 - 2\mu_2^2 \\ &= (\mu_1^2 + \mu_2^2 - 1)^2 - 4\mu_1^2\mu_2^2 \\ &= (\mu_1^2 + \mu_2^2 - 1 + 2\mu_1\mu_2)(\mu_1^2 + \mu_2^2 - 1 - 2\mu_1\mu_2) \\ &= [1 - (\mu_1 + \mu_2)^2][1 - (\mu_1 - \mu_2)^2] \end{aligned}$$

Assuming that this condition is met, a variable λ which satisfies condition (54), and therefore (50) is

$$\begin{aligned} \lambda &= \mu_1^2 - \mu_2^2 \\ &= \left(\frac{A\Delta t}{H}\right)^2 - \left(\frac{B\Delta t}{H}\right)^2 \end{aligned} \quad (55)$$

and this leads to weights

$$\begin{aligned} \alpha &= \frac{1}{2} \left| \left(1 - \frac{A\Delta t}{H}\right)^2 - \left(\frac{B\Delta t}{H}\right)^2 \right| \\ \beta &= \frac{1}{2} \left| \left(1 - \frac{B\Delta t}{H}\right)^2 - \left(\frac{A\Delta t}{H}\right)^2 \right| \\ \gamma &= \frac{1}{2} \left| \left(1 + \frac{A\Delta t}{H}\right)^2 - \left(\frac{B\Delta t}{H}\right)^2 \right| \\ \delta &= \frac{1}{2} \left| \left(1 + \frac{B\Delta t}{H}\right)^2 - \left(\frac{A\Delta t}{H}\right)^2 \right| \end{aligned} \quad (56)$$

Nevertheless, a choice of possible λ still exists and λ may still be regarded as a free (though shackled) parameter. It is necessary to look further for guidance as to the best choice of λ .

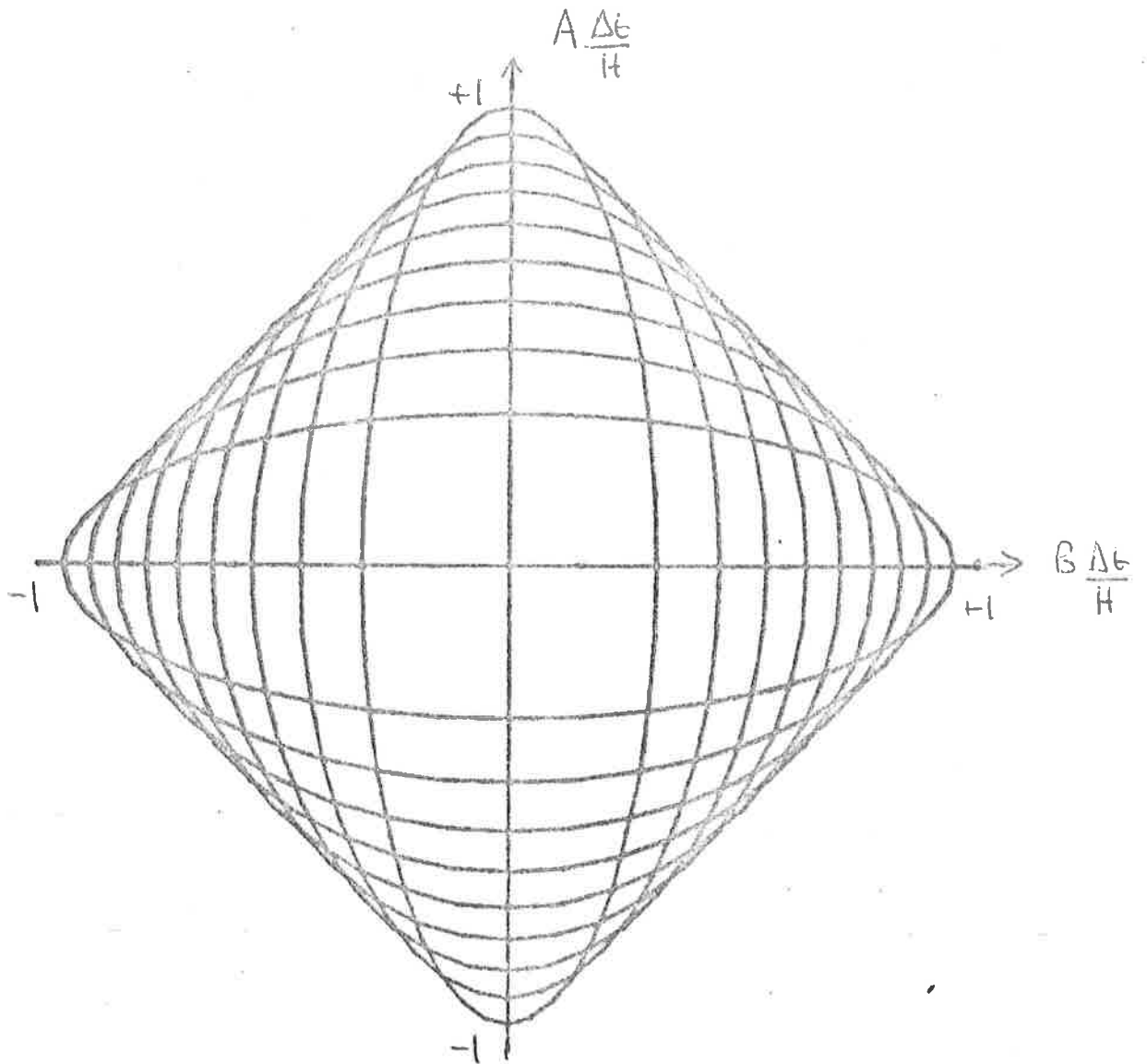


fig 7.

§5. Switching in two dimensions

The switching device introduced by P. L. Roe in his one-dimensional algorithm has at its base a mathematical concept called compatibility (ref.2). This concept, which is allied to monotonicity, can be readily incorporated into the one-dimensional algorithm with impressive results as regards elimination of oscillations.

In two dimensions it is not obvious how this concept is to be generalised and further work is required on the various possibilities. It is possible, however, to try various empirical devices which reduce to Roe's switching device in one dimensional flow. Thus, by applying Roe's criteria to the flux in cells adjacent in the x-direction, a switching rule for nodes on x-coordinate lines is immediately indicated, and similarly for the y-direction. What is new in two dimensions is the possibility of the criteria in both directions being satisfied simultaneously for a given cell. In that case the quantity to be switched is sent diagonally across the cell.

This purely empirical rule is effective in reducing (but not eliminating) oscillations in the test problem described in the next section. It is very close in spirit to Roe's switching device but lacks a mathematical basis as yet. However, it is a promising start in a situation which absolutely requires switching of some sort.

§6. A Test Problem

As a simple test problem, the problem considered recently by Dukiewicz and Ramshaw (1979) of the translation of an L-shaped discontinuity across a square region, was chosen.

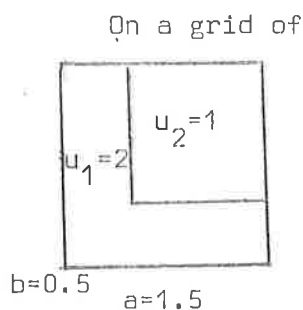


fig 8.

On a grid of 19×19 the initial position of the discontinuity was chosen such that the square region was 13×13 (see fig 8.). The time step was taken to be $\Delta t = 0.2$ with $\Delta x = \Delta y = 0.1$ and $a = 1.5$, $b = 0.5$, so that $v_1 = 0.3$, $v_2 = 0.1$. The algorithm ran for 20 time steps, giving the position of the disturbance after 4 seconds.

Programs were written for both the time splitting scheme and the two-dimensional scheme described here, in each case with and without switching. The programs which include switching are added in the Appendix. In the time splitting scheme, Roe's switching device is included, while in the two-dimensional scheme the empirical switching device discussed in §5 is incorporated. The programs were run on the DEC-10 at RAE Farnborough and effective use was made of a contour plotting routine made available by Keith Wilson of the Aero division.

On this simple test problem the best results, free of oscillations, were obtained from the time-splitting scheme using P. L. Roe's algorithm with switching. These results were, however, rivalled by the two-dimensional scheme when the switching described in §5 was used and when λ was manipulated to suppress alternating instabilities (see §7). With a more refined understanding of these two effects it might be possible to obtain results as good as those from time-splitting. In that case, the two-dimensional scheme, being closer to the physical problem, would certainly be more attractive.

A distinction between the boundary conditions needed for the two schemes became apparent during the programming. In the time-splitting scheme using a one-dimensional algorithm the required transparent condition at the boundary was never called upon whereas in the two-dimensional method the boundary conditions were much more crucial. The simple device of forcing the values of u to be equal at a few points in from the boundary, for each point of the boundary, was used to simulate the transparent boundary condition in the programs written so far. A more consistent approach is discussed in §9 below, however, and it is hoped that this will lead to a clearer understanding of how the results are being affected by boundary conditions.

The test problem here is of course just about the simplest two-dimensional problem that can be constructed, and it is, in fact, particularly well suited to the time splitting algorithm. Many further problems will be needed before any firm conclusions can be drawn.

§7. The choice of λ

By definition λ is the sum of the increments to the diagonally opposite points B, D in a cell less the sum of those to A, C. (See equⁿ. (32)). It is therefore closely connected with the phenomenon of alternating instability which occurs widely in 2-D flow field calculations. Such instabilities are uncontrolled by second order accurate schemes, which are insensitive to perturbations of this kind. Indeed, in a recent paper by Chan (1978) a parameter is defined to cope with this difficulty. This so-called "discrepancy parameter" corresponds to the parameter λ in the above sections and is used to correct new values in the same way as in the weights calculated above. An empirically estimated value of λ is employed.

In a follow-up letter, Strauss (1978) observes that the inclusion of the λ term causes less damping of long wavelengths than if it is omitted. If we allow a variable λ , the parameter may possibly be used to control alternating instabilities as they arise by varying λ to control the discrepancies from cell to cell. Some success has already been obtained on these lines with the test problem. However, as P. Roe has pointed out, the stability analysis is only valid for fixed λ .

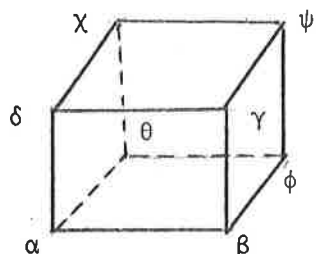
An alternative fate for λ is suggested by the breakdown of the constancy of $u = (bx - ay)^k$ ($k = 2$) in the direction of the stream when a, b vary or when the equation is non-linear. If it is no longer possible to assert that $(bx - ay)$ is an exact solution, a degree of freedom is lost and λ is no longer available as a free parameter.

§8. The method in three dimensions

In three dimensions, a similar approach to that in §3 yields ten conditions for second order accuracy, namely, the conditions for the exactness of the specimen solutions

$$u = 1, x, y, z, x^2, y^2, z^2, xy, yz, zx$$

and also yields eight unknowns, the weights $\alpha, \beta, \gamma, \delta, \theta, \phi, \psi, \chi$, distributing the flux to the eight corners of an element (see fig.). As before, $u = 1$ is satisfied exactly and so are solutions of the form $(px + qy + rz)^n$ with (p, q, r) perpendicular to the velocity of translation (a, b, c) . There are two independent such (p, q, r) 's for both $n = 1$ and $n = 2$ so that five conditions in all are automatically satisfied. Thus



the eight unknowns must satisfy five equations, and there will be solutions with three free parameters, $\lambda_1, \lambda_2, \lambda_3$, say.

The choice of the λ_i ($i = 1, 2, 3$) is even less clear than the choice of λ in two dimensions, although it is reasonable to expect that in three dimensions there will be three kinds of alternating instability, associated with the three coordinate directions, which might be controlled by the λ_i (see §7).

§9. Boundary conditions

Further consideration has been given to boundary conditions on rectangular walls.

Let us consider the two boundary conditions

$$(a) \quad u = \text{const.}$$

$$(b) \quad \frac{\partial u}{\partial n} = 0 \text{ (zero flux condition) .}$$

In the one-dimensional case suppose that these are applied, separately, at a right hand boundary B (see fig a.)

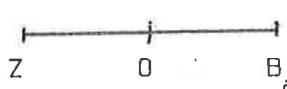


fig a.

In case (a), in order to maintain a zero increment to u_B , the flux from the cell OB has to be directed entirely to the point 0. This reduces the method to first order accuracy at the node 0, but the boundary condition at B is in any case incompatible with the accuracy conditions.

In the case (b), a natural procedure would be to locate the nodes in such a way that the boundary B lies halfway between the last two nodes (see fig .(b)).

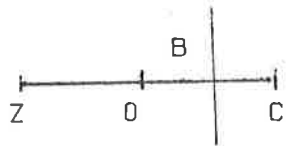


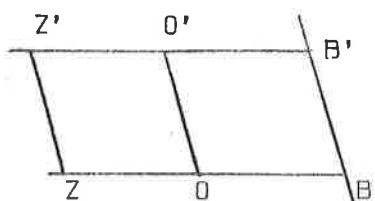
fig b.

Condition (b) can then be replaced by the condition that there is no flux in the last cell, so that the left hand end 0 of that cell is incremented only from the cell Z0. The point C is never used.

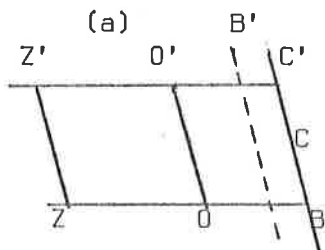
One can summarise by saying that condition (a) corresponds to the flux in the last cell being wholly given to the left end of the cell; and that condition (b) corresponds to the "last" node (0 in fig .) receiving no contribution from the cells to its right.

Periodic boundary conditions are readily simulated by imagining the axis "wrapped around" a cylinder whose circumference is equal to the length of the domain.

Moving into two dimensions, we consider now the two boundary conditions (a) and (b) as applied to a right hand boundary.



(a)



(b)

In case (a) the appropriate generalisation is that the flux from the cell $OBB'O'$ should be given entirely to the points O and O' with weights α, δ satisfying the accuracy conditions in the coordinate parallel to the boundary (cf (13)) (see fig .(a)). In case (b) it is simply that O and O' receive no contributions from the cells like $OCC'O'$ lying "across" the boundary (see fig .(b)).

Appropriate generalisations to 3-D grids are evident, and need not be explicitly described. It is hoped to incorporate these ideas into the programs of §6 as a next step.

§10. Conclusions and Proposed Extensions

The approach introduced by P. L. Roe and its extensions considered above constitute a novel and potentially powerful way of modelling fluid flow problems. Although the two-dimensional extension of the one-dimensional algorithm is at an early stage of development, it is sufficiently elegant and sufficiently promising to merit a detailed investigation, both on the mathematical and practical sides.

Further investigations required are (i) a mathematical analysis of the existing one-dimensional algorithm and its various implementations, (ii) the development and testing of the two-dimensional (and three-dimensional) algorithms, both for model problems and for the full Euler equations, (iii) the extension to higher dimensions of the successful technique for the suppression of oscillations in one-dimension, (iv) the incorporation of boundary conditions in a satisfactory way, and (v) the implementation of the algorithms on irregular grids and for non-uniform time steps.

Acknowledgement

It is a pleasure to acknowledge the stimulus and help given by P. L. Roe of RAE Bedford, and the generous hospitality of RAE Farnborough personified by J. H. B. Smith.

References

1. Roe, P. L., An Improved Version of MacCormack's Shock Capturing Algorithm. RAE Technical Report 79041, April 1979.
2. Roe, P. L., Numerical Algorithms for the Linear Wave equation. } private
3. Roe, P. L., Approximate Solutions to the Riemann Problem. } communications
4. Sod, G., (1978) A Survey of Several Finite Difference Methods for Systems of Nonlinear Hyperbolic Conservation Laws, J. Comp. Phys., 27, 1-31.
5. MacCormack, R. W., (1971), Numerical Solution of the Interaction of a Shock Wave with a laminar Boundary Layer, Lecture Notes in Physics, vol. 8.
6. Lax, P. D. and Wendroff, B., (1960), Systems of Conservation Laws., Comm. Pure & App. Math. 13, 217-237.
7. Warming, R. F. and Beam, R. M., (1976), Upwind second order difference schemes and application in aerodynamic flows, AIAA Journal 14, 1241-1249.
8. Godunov, S. K., see Richtmyer and Morton (loc. cit.), p.338 ff.
9. Richtmyer R.D. and Morton, K. W., (1967), Difference Methods for Initial Value Problems, 2nd ed., Interscience, New York.
10. Dukiewicz, J. K. and Ramshaw, J. D., (1979), Tensor Viscosity Method in Numerical Fluid Dynamics, J. Comp. Phys. 32, 71-79.
11. Chan, R. K-C., (1975), A Lagrangian-Eulerian Method, J. Comp. Phys. 17, 311-330.
12. Strauss, H., (1978), An Artificial Viscosity for Two-dimensional Hydrodynamics, J. Comp. Phys. 28, 437-438.

APPENDIX

Programs for solving the test problem in §6.

TIME SPLITTING SCHEME FOR TEST PROBLEM IN §6

```

00100 COMMON/A/ALPHA,BETA,GAMMA,DELTA,CFLX,CFLY
00200 COMMON/B/UNOW(19,19),UTHEN(19,19)
00300 DIMENSION UINIT(19,19),AMBDA(3)
00400 C
00500 C PARAMETERS
00600 C
00700 T=0.0
00800 DT=0.2
00900 DX=1.0
01000 DY=1.0
01100 A=1.5
01200 B=0.5
01300 CFLX=A*DT/DX
01400 CFLY=B*DT/DY
01500 SUMCFL=CFLX+CFLY
01600 DIFCFL=CFLY-CFLX
01700 C
01800 C WEIGHTS
01900 C
02000 ALPHA=0.25*(1-SUMCFL)
02100 BETA=0.25*(1-DIFCFL)
02200 GAMMA=0.25*(1+SUMCFL)
02300 DELTA=0.25*(1+DIFCFL)
02400 C
02500 C INITIALISE U
02600 C
02700 DO 1 I=1,19
02800 DO 1 J=1,19
02900 1 UINIT(I,J)=2.0
03000 DO 2 I=6,19
03100 DO 2 J=6,19
03200 2 UINIT(I,J)=1.0
03300
03400 C
03500 C SPLIT SCHEME
03600 C
03700 DO 14 I=1,19
03800 DO 11 J=1,19
03900 UNOW(I,J)=UINIT(I,J)
04000 11 UTHEN(I,J)=UNOW(I,J)
04100
04200 14 CONTINUE
04300 T=T+0.5*DT
04400 CALL XSTEP(0.5)
04500 DO 12 K=1,19
04600 T=T+DT.
04700 CALL YSTEP(1.0)
04800 T=T+DT
04900 12 CALL XSTEP(1.0)
05000 T=T+DT
05100 CALL YSTEP(1.0)
05200 T=T+0.5*DT
05300 CALL XSTEP(0.5)
05400 WRITE(5,90)
05500 WRITE(5,91)
05600 90 FORMAT(13H SPLIT SCHEME)
05700 91 FORMAT(7H SPLITS)
05800 CALL WRITE(18,18)
05900 END
06000 C
06100 C
06200 C

```

continued.

```

06000 C
06100 C
06200 C
06300 SUBROUTINE XSTEP(DTFRAC)
06400 COMMON/A/ALPHA,BETA,GAMMA,DELTA,CFLX,CFLY
06500 COMMON/B/UNOW(19,19),UTHEN(19,19)
06600 DIMENSION YIELD(19),XDIF(19),NUDGE(19)
06700 DO 104 J=1,19
06800 DO 101 I=1,18
06900 NUDGE(I)=0
07000 XDIF(I)=UNOW(I+1,J)-UNOW(I,J)
07100 101 YIELD(I)=-DTFRAC*CFLX*XDIF(I)
07200 DO 103 I=1,18
07300 UTHEN(I,J)=UTHEN(I,J)+(ALPHA+DELTA)*YIELD(I)
07400 103 UTHEN(I+1,J)=UTHEN(I+1,J)+(BETA+GAMMA)*YIELD(I)
07500 C
07600 C SWITCH
07700 C
07800 DO 106 I=1,18
07900 IF(ABS(XDIF(I+1)).GT.ABS(XDIF(I)).AND.CFLX.GT.0.0)NUDGE(I)=1
08000 IF(ABS(XDIF(I+1)).LT.ABS(XDIF(I)).AND.CFLX.LT.0.0)NUDGE(I+1)
08100 106 CONTINUE
08200 DO 108 I=1,17
08300 N=NUDGE(I)
08400 YY=YIELD(I+1)-YIELD(I)
08500 IF(N.EQ.1)UTHEN(I+1,J)=UTHEN(I+1,J)-
08600 8 (ALPHA+DELTA)*YY
08700 IF(N.EQ.1)UTHEN(I+2,J)=UTHEN(I+2,J)+
08800 9 (ALPHA+DELTA)*YY
08900 108 CONTINUE
09000 DO 107 I=1,17
09100 N=NUDGE(I+1)
09200 YY=YIELD(I+1)-YIELD(I+2)
09300 IF(N.EQ.-1)UTHEN(I,J)=UTHEN(I,J)-
09400 8 (BETA+GAMMA)*YY
09500 IF(N.EQ.-1)UTHEN(I+1,J)=UTHEN(I+1,J)+
09600 9 (BETA+GAMMA)*YY
09700 107 CONTINUE
09800 104 CONTINUE
09900 CALL UPDATE
10000 END
10100 C
10200 C
10300 C

```

Continued

```

10100 C
10200 C
10300 C
10400 SUBROUTINE YSTEP(DTFRAC)
10500 COMMON/A/ALPHA,BETA,GAMMA,DELTA,CFLX,CFLY
10600 COMMON/B/UNOW(19,19),UTHEN(19,19)
10700 DIMENSION YIELD(19),YDIF(19),NUDGE(19)
10800 DO 204 I=1,19
10900 DO 201 J=1,18
11000 NUDGE(J)=0
11100 YDIF(J)=UNOW(I,J+1)-UNOW(I,J)
11200 201 YIELD(J)=-DTFRAC*CFLY*YDIF(J)
11300 DO 203 J=1,18
11400 UTHEN(I,J)=UTHEN(I,J)+(ALPHA+BETA)*YIELD(J)
11500 203 UTHEN(I,J+1)=UTHEN(I,J+1)+(GAMMA+DELTA)*YIELD(J)
11600 C
11700 C SWITCH
11800 DO 206J=1,18
11900 IF(ABS(YDIF(J+1)).GT.ABS(YDIF(J)).AND.CFLY.GT.0.0)NUDGE(J)=1
12000 IF(ABS(YDIF(J+1)).LT.ABS(YDIF(J)).AND.CFLY.LT.0.0)NUDGE(J+1)=1
12100 206 CONTINUE
12200 DO 208 J=1,17
12300 N=NUDGE(J)
12400 YY=YIELD(J+1)-YIELD(J)
12500 IF(N.EQ.1)UTHEN(I,J+1)=UTHEN(I,J+1)-
12600 8 (ALPHA+BETA)*YY
12700 IF(N.EQ.1)UTHEN(I,J+2)=UTHEN(I,J+2)+
12800 9 (ALPHA+BETA)*YY
12900 208 CONTINUE
13000 DO 207 J=1,17
13100 N=NUDGE(J+1)
13200 YY=YIELD(J+1)-YIELD(J+2)
13300 IF(N.EQ.-1)UTHEN(I,J)=UTHEN(I,J)-
13400 8 (GAMMA+DELTA)*YY
13500 IF(N.EQ.-1)UTHEN(I,J+1)=UTHEN(I,J+1)+
13600 9 (GAMMA+DELTA)*YY
13700 207 CONTINUE
13800 204 CONTINUE
13900 CALL UPDATE
14000 END
14100 C
14200 C
14300 C
14400 SUBROUTINE UPDATE
14500 COMMON/B/UNOW(19,19),UTHEN(19,19)
14600 DO 401 I=2,18
14700 DO 402 J=2,18
14800 402 UNOW(I,J)=UTHEN(I,J)
14900 401 CONTINUE
15000 DO 403 K=1,19
15100 UNOW(1,K)=UTHEN(2,K)
15200 UNOW(K,1)=UTHEN(K,2)
15300 UNOW(19,K)=UTHEN(18,K)
15400 403 UNOW(K,19)=UTHEN(K,18)
15500 END
15600 C
15700 C
15800 C
15900 SUBROUTINE WRITE(LIMITI,LIMITJ)
16000 COMMON/B/UNOW(19,19),UTHEN(19,19)
16100 DO 500 J=1,LIMITJ
16200 500 WRITE(5,505)(UNOW(I,J),I=1,LIMITI)
16300 505 FORMAT(1H ,18F7.4)
16400 END

```

End of program

TWO-DIMENSIONAL SCHEME FOR TEST PROBLEM IN §6.

```

00100      COMMON/A/ALPHA,BETA,GAMMA,DELTA,CFLX,CFLY
00200      COMMON/B/UNOW(19,19),UTHEN(19,19)
00300      DIMENSION UINIT(19,19),LAMBDA(3)
00400      C
00500      C      PARAMETERS
00600      C
00700      T=0.0
00800      DT=0.2
00900      DX=1.0
01000      DY=1.0
01100      A=1.5
01200      B=0.5
01300      CFLX=A*DT/DX
01400      CFLY=B*DT/DY
01500      C
01600      C      INITIALISE U
01700      C
01800      DO 1 I=1,19
01900      DO 1 J=1,19
02000      1  UINIT(I,J)=2.0
02100      DO 2 I=6,19
02200      DO 2 J=6,19
02300      2  UINIT(I,J)=1.0
02400      C
02500      C      BOX SCHEME
02600      C
02700      29  DO 24 I=1,19
02800      DO 21 J=1,19
02900      UNOW(I,J)=UINIT(I,J)
03000      21  UTHEN(I,J)=UNOW(I,J)
03100      24  CONTINUE
03200      20  DO 22 K=1,20
03300      T=T+DT
03400      22  CALL BOX(1,0)
03500      WRITE(5,79)
03600      WRITE(5,80)
03700      79  FORMAT(7H BOXSWI)
03800      80  FORMAT(11H BOX SCHEME)
03900      CALL WRITE(18,18)
04000      END
04100      C
04200      C
04300      C

```

```

04400 SUBROUTINE SOXCHALL
04500 COMMON/A/ALPHA,BETA,GAMMA,DELTA,CFLX,CFLY
04600 COMMON/B/UNOW(19,19),UTHEN(19,19),UMIT(19,19)
04700 DIMENSION YIELD(19,19),XDIF(19,19),YDIF(19,19)
04800 DIMENSION NUDGE(19,19),MUDGE(19,19)
04900 DO 300 I=1,19
05000 DO 301 J=1,19
05100 NUDGE(I,J)=0
05200 MUDGE(I,J)=0
05250 IF(I.EQ.19)GOTO 200
05300 XDIF(I,J)=UNOW(I+1,J)-UNOW(I,J)
05350 200 IF(J.EQ.19)GOTO 301
05400 YDIF(I,J)=UNOW(I,J+1)-UNOW(I,J)
05420 301 CONTINUE
05440 300 CONTINUE
05460 DO 302 I=1,18
05480 DO 303 J=1,18
05500 XIELD=-0.5*BTALL*CFLX*(XDIF(I,J)+XDIF(I,J+1))
05600 YIELD(I,J)=-0.5*BTALL*CFLY*(YDIF(I,J)+YDIF(I+1,J))
05700 303 YIELD(I,J)=XIELD+YIELD(I,J)
05800 302 CONTINUE
05900 DO 307 I=1,18
06000 DO 306 J=1,18
06050 Y=YIELD(I,J)
06100 IF(ABS(YIELD(I+1,J)).GT.ABS(Y).AND.CFLX.GT.0.0)
06200 2 NUDGE(I+1,J)=1
06300 IF(ABS(YIELD(I+1,J)).LT.ABS(Y).AND.CFLX.LT.0.0)
06400 3 MUDGE(I,J)=-1
06500 IF(ABS(YIELD(I,J+1)).GT.ABS(Y).AND.CFLY.GT.0.0)
06600 4 MUDGE(I,J+1)=1
06700 IF(ABS(YIELD(I,J+1)).LT.ABS(Y).AND.CFLY.LT.0.0)
06800 5 NUDGE(I,J)=-1
06900 306 CONTINUE
07000 307 CONTINUE
07330 AMBDA=0.0
07360 CALL HEIGHT(AMBDA)
07400 DO 304 I=1,18
07500 DO 305 J=1,18
07600 Y=YIELD(I,J)
07800 UTHEN(I,J)=UTHEN(I,J)+ALPHAY
07900 UTHEN(I+1,J)=UTHEN(I+1,J)+BETAY
08000 UTHEN(I+1,J+1)=UTHEN(I+1,J+1)+GAMMAY
08200 UTHEN(I,J+1)=UTHEN(I,J+1)+DELTA*Y
08300 305 CONTINUE
08400 304 CONTINUE
08420 C
08430 C SWITCH
08440 C
08500 DO 605 I=2,18
08600 DO 606 J=2,18
08700 N=NUDGE(I,J)
08800 M=MUDGE(I,J)
08900 IF(N+M.EQ.0)GOTO 606
09000 SWA=YIELD(I,J)-YIELD(I-N,J-M)
09200 602 UTHEN(I,J)=UTHEN(I,J)-ALPHA*SWA
09400 UTHEN(I+N,J+M)=UTHEN(I+N,J+M)+ALPHA*SWA
09600 IF(N.EQ.0)GOTO 609
09650 SWD=YIELD(I,J)-YIELD(I-1,J)
09700 UTHEN(I,J+1)=UTHEN(I,J+1)-DELTA*SWD
09900 UTHEN(I+1,J+1)=UTHEN(I+1,J+1)+DELTA*SWD
10100 IF(M.EQ.0)GOTO 606
10150 609 SUB=YIELD(I,J)-YIELD(I,J-1)
10200 UTHEN(I+1,J)=UTHEN(I+1,J)-BETA*SUB
10400 UTHEN(I+1,J+1)=UTHEN(I+1,J+1)+BETA*SUB
10600 606 CONTINUE
10700 605 CONTINUE
10800 CALL UPDATE

```



```

11300      SUBROUTINE UPDATE
11400      COMMON/B/UNOW(19,19),UTHEN(19,19)
11500      DO 401 I=1,19
11600      DO 402 J=1,19
11700 402   UNOW(I,J)=UTHEN(I,J)
11800 401   CONTINUE
11900      DO 408 K=1,19
12000      U=UTHEN(4,K)
12100      UNOW(1,K)=U
12200      UNOW(2,K)=U
12300      UNOW(3,K)=U
12400      U=UTHEN(K,4)
12500      UNOW(K,1)=U
12600      UNOW(K,2)=U
12700      UNOW(K,3)=U
12800      U=UTHEN(16,K)
12900      UTHEN(17,K)=U
13000      UTHEN(18,K)=U
13100      UTHEN(19,K)=U
13200      U=UTHEN(K,16)
13300      UNOW(K,17)=U
13400      UNOW(K,18)=U
13500      UNOW(K,19)=U
13600 408   CONTINUE
13700      END
13800      C
13900      C
14000      C
14100      SUBROUTINE WRITE(LIMITI,LIMITJ)
14200      COMMON/B/UNOW(19,19),UTHEN(19,19)
14300      DO 500 J=1,LIMITJ
14400 500   WRITE(5,505)(UNOW(I,J),I=1,LIMITI)
14500 505   FORMAT(1H ,18F7.4)
14600      END
14700      C
14800      C
14900      C
15000      SUBROUTINE WEIGHT(AMBDA)
15100      COMMON/A/ALPHA,BETA,GAMMA,DELTA,CFLX,CFLY
15200      ALPHA=0.25*(1+AMBDA-CFLX-CFLY)
15300      BETA=0.25*(1-AMBDA-CFLY+CFLX)
15400      GAMMA=0.25*(1+AMBDA+CFLX+CFLY)
15500      DELTA=0.25*(1-AMBDA+CFLY-CFLX)
15600      END

```