NASA Contractor Report 172415

ICASE REPORT NO. 84-27

ICASE

FLUX-VECTOR SPLITTING AND RUNGE-KUTTA METHODS FOR THE EULER EQUATIONS

Eli Turkel

and

Bram van Leer

Contracts Nos. NAS1-16394, NAS1-17070, and NAS1-17130

June 1984

INSTITUTE FOR COMPUTER APPLICATIONS IN SCIENCE AND ENGINEERING
NASA Langley Research Center, Hampton, Virginia 23665

Operated by the Universities Space Research Association

National Aeronautics and Space Administration

Langley Research Center
Hampton, Virginia 23665

FLUX-VECTOR SPLITTING AND RUNGE-KUTTA METHODS FOR THE EULER EQUATIONS

Eli Turkel
Institute for Computer Applications in Science and Engineering
and
Tel-Aviv University, Tel-Aviv, Israel

and

Bram van Leer
Institute for Computer Applications in Science and Engineering
and
Technische Hogeschool Delft, Netherlands

Abstract

Runge-Kutta schemes have been used as a method of solving the Euler equations exterior to an airfoil. In the past this has been coupled with central differences and an artificial viscosity in space. In this study we couple the Runge-Kutta time-stepping scheme with an upwinded space approximation based on flux-vector splitting. Several acceleration techniques are also considered including a local time step, residual smoothing and multigrid.

Research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-17070. Additional support was provided for the first author under NASA Contract No. NAS1-17130. For the second author, additional support was provided under NASA Contract No. NAS1-16394 while both authors were in residence at ICASE, NASA Langley Research Center, Hampton, VA 23665.

Introduction

We wish to solve the steady-state multidimensional Euler equations with a method that is suitable for a large range of Mach numbers. At the same time we wish the method to be accurate, robust and capture shocks without excessive smearing. We also wish to reach the steady state rapidly. To achieve these goals we combine the Runge-Kutta scheme introduced in [1] with the flux-vector splitting introduced in [2].

Consider the two-dimensional system

$$w_t + f_x + g_y = 0. \tag{1}$$

We advance the numerical solution in time using a N stage algorithm

$$\begin{aligned} w^{(0)} &= w^n \\ &\vdots \\ w^{(k)} &= w^{(0)} - \alpha_k \Delta t\left(D_x f^{(k-1)} + D_y g^{(k-1)}\right) \\ &\vdots \\ w^{n+1} &= w^{(N)} \end{aligned} \tag{2}$$

where $D_x f$ and $D_y g$ are difference approximations to the flux derivatives. To check the stability we freeze coefficients and Fourier transform. The amplification matrix of (2) is then

$$G = 1 + \beta_1 z + \beta_2 z^2 + \cdots + \beta_N z^N$$

$$\beta_1 = 1 \tag{3}$$

$$\beta_k = \beta_{k-1} \alpha_{N-k+1}$$

where z is the Fourier transform of $\Delta t(D_x f + D_y g)$. When central differences are used then z lies on the imaginary axis. With upwind differences z lies on some curve in the negative real half of the complex plane.

Experience has shown that one should usually choose the parameters so that the time step is maximal. For central differences this implies that

$$C\Delta t/\Delta x < N - 1 \tag{4}$$

where C depends on $|z|_{max}$. With upwind schemes no general rules have been developed thus far for optimal parameters. At present the parameters have been chosen by experimentation. One possibility is presented in the result section.

To appreciate the connection between central differences and flux vector splitting we consider a one-dimensional example. Assume that we wish to compute a numerical flux at the cell interface $i + 1/2$. Quadratically interpolating the cell averages in zones $i-1$, i, $i+1$ yields a left-side estimate of the state variables

$$w^L_{i+1/2} = \bar{w}_i + \frac{1}{4}(\bar{w}_{i+1} - \bar{w}_{i-1}) + \frac{1}{12}(\bar{w}_{i+1} - 2\bar{w}_i + \bar{w}_{i-1}). \qquad (5)$$

Interpolating the cell averages in zones i, i+1, i+2 yields the right-sided estimate

$$w^R_{i+1/2} = \bar{w}_{i+1} - \frac{1}{4}(\bar{w}_{i+2} - \bar{w}_i) + \frac{1}{12}(\bar{w}_{i+2} - 2\bar{w}_{i+1} + \bar{w}_i). \qquad (6)$$

The difference between these values is $O((\Delta x)^3)$. We now introduce an upwind bias in the numerical flux by using from the components of $w^L_{i+1/2}$ only those characteristic combinations that are convected forward and from $w^R_{i+1/2}$ those convected backward. In the approximation of flux splitting this becomes

$$f(w^L_{i+1/2}, w^R_{i+1/2}) = (f^+)^L_{i+1/2} + (f^-)^L_{i+1/2}. \qquad (7)$$

This can be rewritten as

$$f(w^L_{i+1/2}, w^R_{i+1/2}) = \frac{1}{2}(f^L_{i+1/2} + f^R_{i+1/2}) - \frac{1}{2} Q_{i+1/2}(w^R_{i+1/2} - w^L_{i+1/2}), \qquad (8)$$

with

$$Q = \frac{df^+}{dw} - \frac{df^-}{dw};$$

$Q_{i+1/2}$ is a Roe-type [3] average of Q over the interval $(w^L_{i+1/2}, w^R_{i+1/2})$.

From (8) we see that the upwind-biased flux deviates from the average flux, used for central differencing, by a third-order term. This leads to a fourth-order viscosity with a matrix-valued coefficient. This viscosity

prevents the checkerboard instability similar to the fourth-order viscosity introduced in [1].

Equations (5) and (6) are modified before their actual use. The first order term is multiplied by a switch described in [4] while the second-order term is multiplied by its square. When $\bar{w}_{i+1} - \bar{w}_i$ is large compared with $\bar{w}_i - \bar{w}_{i-1}$ and $\bar{w}_{i+2} - \bar{w}_{i+1}$, as in a shock profile, then the limiting yields

$$w^R_{i+1/2} - w^L_{i+1/2} \simeq \bar{w}_{i+1} - \bar{w}_i. \tag{9}$$

Therefore, the viscosity term in (8) now leads locally to a second-order viscosity which guarantees a monotone profile. This is similar to the second-order artificial viscosity introduced in [1] and discussed in more detail in [5].

Limiting the high-order terms combined with upwind differencing is a robust way of preventing numerical oscillations near discontinuities. To achieve the same effect the viscosity of [1] would have to be raised to the level of the spectral radius of Q leading to excessive smearing. In practice using the code of [1] the opposite approach is used. The coefficient of viscosity is adjusted, by trial and error, so that the shock profiles are sharp and spurious entropy production in the smooth flow is minimized. For smooth flows one can achieve viscosity levels that are smaller than that of the upwind scheme with the limiter. For violent flows the artificial viscosity is much too large. In any case the coefficient of viscosity used in [1] is very problem-dependent while the present code has no adjustable parameters that need to be played with.

Flux splitting in an arbitrary multidimensional body-fitted coordinate system can be reduced to a one-dimensional problem with reference direction normal to the cell face where fluxes are computed. The resultant code is a full two-dimensional code and does not use time splitting to combine the coordinate directions. In all the interpolations we have ignored the geometric variation of the cells as was done in [1]. This can lead to errors for high angles of attack, especially near the trailing edge where cell shape and size vary strongly.

Having discussed the spatial discretization we now introduce several techniques to accelerate the convergence to a steady state. The first technique is to use a local time step. This improves the running time by an order of magnitude due to the small cells near the airfoil. The second technique is to use residual smoothing after each stage of the Runge-Kutta method. This was first introduced in [6] for the Lax-Wendroff scheme. If one uses central differencing then the scheme is unconditionally stable when the smoothing is done after every even stage. Using an upwind scheme the smoothing should be done after each stage of the algorithm. Even though the resultant scheme is unconditionally stable it is not efficient to use time steps that are too large. Time steps about three times as large as those of the explicit scheme seem to be optimal. Since the residual smoothing adds only about 10% to the running time per time step the use of the residual smoothing is advantageous.

A third acceleration technique is to use a multigrid method. Jameson [7] has proposed using the Runge-Kutta scheme coupled with central differences as a smoothing algorithm for a multigrid scheme. The parameters α_k are now

chosen to damp the high frequencies rather than achieving a maximal time step. We use the same technique with upwind differencing in the Runge-Kutta scheme instead of central differences with an artificial viscosity. In the central-difference version [7] the artificial viscosity is increased compared with the standard Runge-Kutta scheme [1]. In addition the multigrid central-difference code seems to rely on the enthalpy damping in order to achieve rapid convergence to the steady-state. With the upwind code there is no longer an artificial viscosity that can be tuned to damp the high modes. Furthermore, the steady-state total enthalpy is not preserved by the flux-vector-splitting scheme [2]. Hence, the enthalpy damping introduced in [1] and [5] cannot be used. Nevertheless the multigrid scheme does work with the upwind-biased scheme. Thus, the viscosity that is implicit in the scheme seems to be sufficient to compensate for both the artificial viscosity and the enthalpy damping of the central-difference scheme. However, the convergence rate of the upwind scheme is slower than that of the central-difference scheme, mainly because of the enthalpy damping in the latter.

Results

The upwind biased version of the code has been run on several different cases. The first case is a NACA-0012 airfoil with $M_\infty = 0.8$, and $\alpha = 1.25^\circ$. The mesh is an O-mesh generated by a sheared parabolic transformation. We use a four stage Runge-Kutta scheme with $\alpha_1 = .17$, $\alpha_2 = .273$, $\alpha_3 = .5$ and $\alpha_4 = 1$. The residual smoothing is applied after each stage with $\beta_x = .9\lambda/8$ and $\beta_y = .6\beta_x$ (λ is the local Courant number) with $\lambda = 3$. Using a 64 × 16

mesh the residual is reduced by 4 orders of magnitude after 600 steps. The C_p curve is similar to that achieved by the central-difference code [1] except that the shock profile is now sharper with one point in the middle of the shock along the airfoil. We have also run several supersonic flows about the NACA-0012. The upwind version of the code converges for a larger range of Mach numbers than does the central difference version.

Because of the flux splitting and upwind logic entering the execution the upwind code is about two times slower per time step than the central-difference version. A further slowdown is caused by the Runge-Kutta method which seems to favor the spectral distribution of central differences and which has not yet been optimized for upwind differences. Hence, the time step is about half of that for central differences. In addition, the enthalpy damping described in [1] and [5] cannot be used. Thus, the present version of the upwind scheme is about 5 times slower in reaching the steady state than the central-difference code of [1].

The multigrid version of the code has also been run using a four-stage Runge-Kutta. The original parameters were reasonable but a better set is $\alpha_1 = .15$, $\alpha_2 = .3275$, $\alpha_3 = .57$ and $\alpha_4 = 1$. We have also used a six-stage formula with $\alpha_1 = .073$, $\alpha_2 = .138$, $\alpha_3 = .22$, $\alpha_4 = .334$, $\alpha_5 = .5$ and $\alpha_6 = 1$. On a 64×16 mesh the multigrid version requires fewer iterations to converge. However, accounting for the extra work of the multigrid the two codes have about the same convergence rate per work unit. Nevertheless, if a coarser mesh is used to initialize the finer mesh, then the total number of supersonic points is predicted within 60 iterations on the fine mesh. Moreover, it is expected that for finer grids the multigrid will be more efficient. The

central-difference multigrid code is the fastest code. However, this relies heavily on the enthalpy damping. Thus, for the Navier-Stokes equations it cannot always be used. In this case the upwind multigrid scheme will be the most efficient.

References

[1] A. Jameson, W. Schmidt, and E. Turkel, AIAA Paper No. 81-1259, 1982.

[2] B. Van Leer, Lecture Notes in Physics 170, p. 507, 1982.

[3] P. L. Roe, J. Comput. Phys, Vol. 43, p. 357, 1981.

[4] G. D. Van Albada, B. Van Leer, and W. W. Roberts, Jr., Astron. Astrophys., Vol. 108, p. 76, 1982.

[5] E. Turkel, "Acceleration to a steady state for the Euler equations," to appear in Proc. of the INRIA Workshop on Numerical Methods for the Euler Equations for Compressible Flows, 1984.

[6] A. Lerat, C. R. Acad. Sci., Paris, t. 288, 1979.

[7] A. Jameson, J. Appl. Mech., Vol. 50, p. 1052, 1983.