NASA CR-179,742

NASA-CR-179742
19860023614

IMPLICIT MULTIGRID ALGORITHMS FOR THE THREE-DIMENSIONAL FLUX SPLIT EULER EQUATIONS

BY
WILLIAM KYLE ANDERSON

NF00221

NASA CR-179,742

LIBRARY COPY

NOV 6 1986

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

N86-33086#

IMPLICIT MULTIGRID ALGORITHMS FOR THE THREE-DIMENSIONAL FLUX SPLIT EULER EQUATIONS

By

William Kyle Anderson

LIBRARY COPY

NOV 6 1986

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in the Department of Aerospace Engineering

Mississippi State, Mississippi

August 1986

IMPLICIT MULTIGRID ALGORITHMS FOR THE THREE-DIMENSIONAL FLUX SPLIT EULER EQUATIONS

By

William Kyle Anderson

A Dissertation
Submitted to the Faculty of
Mississippi State University
in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
in the Department of Aerospace Engineering

Mississippi State, Mississippi

August 1986

Copyright© by William Kyle Anderson, 1986

IMPLICIT MULTIGRID ALGORITHMS FOR THE
THREE-DIMENSIONAL FLUX SPLIT EULER EQUATIONS

By

William Kyle Anderson

Professor of Aerospace Engineering
(Chairman of Committee and
Dissertation Director)

Keith Koenig
Associate Professor of Aerospace
Engineering (Graduate
Coordinator and Member of
Committee)

Professor of Aerospace Engineering
(Member of Committee)

C. Wayne Mastin
Professor of Mathematics
(Member of Committee)

Professor of Aerospace Engineering
(Member of Committee)

Charles B. Clark
Professor and Head of the
Department of Aerospace
Engineering

Dean of College of Engineering

Ralph E. Palm
Vice President for Graduate
Studies and Research

Director of Graduate Instruction
College of Engineering

August 1986

ACKNOWLEDGMENTS

The author would like to thank Dr. David L. Whitfield for his guidance throughout this research project. The financial support of the NASA-Langley Research Center is also appreciated as is the encouragement of the Analytical Methods Branch. Special thanks to James L. Thomas and Richard J. Margason for all their help and to Ms. Jacqueline Smith for the typing of the manuscript.

ABSTRACT

William Kyle Anderson, Doctor of Philosophy, 1986

Major: Engineering, Department of Aerospace Engineering

Title of Dissertation: Implicit Multigrid Algorithms for the Three-Dimensional Flux Split Euler Equations

Directed by: David L. Whitfield

Pages in Dissertation: 83. Words in Abstract: 232.

ABSTRACT

The full approximation scheme multigrid method is applied to several implicit flux-split algorithms for solving the three-dimensional Euler equations in a body fitted coordinate system. Each uses a variation of approximate factorization and are implemented in a finite volume formulation. The algorithms are all vectorizable with little or no scalar computations required. The flux vectors are split into upwind components using both the splittings of Steger-Warming and Van Leer. Results comparing pressure distributions with experimental data using both splitting types are shown. The stability and smoothing rate of each of the schemes are examined using a Fourier analysis of the complete system of equations. Results are presented for three-dimensional subsonic, transonic, and supersonic flows which demonstrate substantially improved convergence rates with the multigrid algorithm. The influence of using both a V-cycle and a W-cycle on the convergence is examined. Using the multigrid method on both subsonic and transonic wing calculations, the final lift coefficient is obtained to within 0.1 percent of its final value in as few as 15 cycles for a mesh with over

210,000 points. A spectral radius of 0.89 is achieved for both subsonic and transonic flow over the ONERA M6 wing while a spectral radius of 0.83 is obtained for supersonic flow over an analytically defined forebody. Results compared with experiment for all cases show good agreement.

TABLE OF CONTENTS

ACKNOWLEDGMENTS.....ii

ABSTRACT.....iii

NOMENCLATURE.....vi

LIST OF FIGURES.....x

LIST OF TABLES.....xii

Chapter

I. INTRODUCTION.....1

II. EULER SOLUTION METHOD.....5

2.1 Euler Equations in Generalized Coordinates.....5

2.2 Flux Vector Splitting.....7

2.3 Baseline Solution Algorithm.....15

2.4 Boundary Conditions.....23

2.5 Stability Analysis.....25

III. MULTIGRID.....30

3.1 General Algorithm.....30

3.2 Algorithm for the Euler Equations.....35

IV. RESULTS.....43

4.1 ONERA M6.....43

4.2 Analytic Forebody.....61

V. CONCLUSIONS.....65

APPENDICES.....67

A. Transformation to Generalized Coordinates.....67

B. Splitting the Flux Vectors in Generalized Coordinates.....71

C. Restriction Operators.....77

BIBLIOGRAPHY.....81

NOMENCLATURE

A, B, C	flux Jacobians, $\partial F/\partial Q$, $\partial G/\partial Q$, $\partial H/\partial Q$
$\hat{A}^{\pm}$, $\hat{B}^{\pm}$, $\hat{C}^{\pm}$	flux Jacobians, $\partial \hat{F}^{\pm}/\partial \hat{Q}$, $\partial \hat{G}^{\pm}/\partial \hat{Q}$, $\partial \hat{H}^{\pm}/\partial \hat{Q}$
$\bar{A}$	matrix from similarity transformation
a	speed of sound
CFL	Courant-Friedrichs-Lewy number
c	airfoil chord
c_{ℓ}	section lift coefficient
C_p	pressure coefficient
e	total energy per unit volume
F, G, H	fluxes of mass, momentum, and energy
f_{mass}	mass component of flux vector
f_{energy}	energy component of flux vector
G_i	grid level i
I	identity matrix
I_i^{i-1}	restriction operator used for transferring functions on grid i to grid i-1
I_{i-1}^{i}	prolongation operator used for transferring functions on grid i-1 to grid i
$\hat{I}_i^{i-1}$	restriction operator for the residual
i	$\sqrt{-1}$
$\hat{i}, \hat{j}$	x and y components of a unit vector
J	transformation Jacobian
$\hat{\xi}_x, \hat{\xi}_y, \hat{\xi}_z$	components of a unit vector normal to a ξ = constant face
k_i	denotes ξ, η, or ζ
L	nonlinear operator

M_ξ	local Mach number in ξ direction
M	implicit operator, also denotes Mach number
$\hat{N}$	unit vector normal to a k = constant face
P	forcing function added to residual
p	pressure
q_i	approximate value of conserved variables on grid i
Q	conserved variables representing mass, momentum, and total energy per unit volume
R	residual, also used as Riemann invariant
$\hat{R}$	unit vector normal to $\hat{N}$ and $\hat{T}$
$\hat{r}_x, \hat{r}_y, \hat{r}_z$	components of $\hat{R}$ in the x,y and z direction
S	forcing function
S_i	length of a cell face
T	similarity transformation matrix whose columns are the right eigenvectors of the flux Jacobians, also used as a rotation matrix
T^{-1}	similarity transformation matrix whose rows are the left eigenvectors of the flux Jacobians
$\hat{T}$	unit vector normal to $\hat{N}$ and $\hat{R}$
t	time
$\hat{t}_x, \hat{t}_y, \hat{t}_z$	components of $\hat{T}$ in the x,y, and z directions
u,v,w	Cartesian velocities, v also denotes an eigenvector for Fourier analysis
U,V,W	contravariant velocities, V also denotes the volume of a computational cell
U_0	a constant vector

V_i	correction on grid level i
x,y,z	Cartesian coordinates
α	angle of attack
α, β, γ	coefficients for stability analysis
γ	ratio of specific heats, $\gamma = 1.4$
δ_x	difference operator in x direction
ξ, η, ζ	general curvilinear coordinates
Λ	diagonal matrix of eigenvalues
$\lambda_1, \ldots, \lambda_5$	eigenvalues of Jacobian matrices
λ	amplification factor
μ	smoothing factor
ΔQ	change in dependent variables, $Q^{n+1} - Q^n$
Δt	time step
$\Delta x, \Delta y, \Delta z$	change in x,y, and z
∂	denotes partial derivative
ρ	density
$\sum$	summation
τ	time
τ_i	relative truncation error
grad(Ω)	gradient of Ω
$\lvert \chi \rvert$	magnitude of vector χ

Subscripts

i,j,k	cell indices
x,y,z	spatial derivative
∞	denotes conditions at infinity

N,N-1,...	discretization on grid N,N-1,...

Superscripts

c	denotes most current value of a quantity
n	denotes time level
±	denotes positive and negative flux and eigenvalue contributions; also denotes forward and backward spatial differencing or extrapolation
$(\bar{\ })$	denotes quantities in locally orthogonal coordinates
^	denotes fluxes and flux Jacobians in generalized coordinates; also denotes Fourier symbol

LIST OF FIGURES

Figure Page

1. Variation of Steger-Warming split mass flux with Mach number....11

2. Variation of Van Leer split mass flux with Mach number....14

3. Computational cell indexing....17

4. Computational modules

   (a) 3-factor, spatially split scheme....19

   (b) 2-factor, eigenvalue split scheme....19

   (c) 2-factor, combination split scheme....19

5. Vectorization of LU decomposition and back-substitution over multiple planes....20

6. Stability analysis of three-dimensional approximate factorization schemes; $M_\infty = 0.8$; $\alpha = 0°$.

   (a) 3-factor, spatially split scheme....28

   (b) 2-factor, eigenvalue split scheme....28

   (c) 2-factor, combination split scheme....28

7. Multigrid V-cycle....39

8. Multigrid W-cycle....40

9. Surface mesh for the ONERA M6 wing

   (a) 97 x 17 x 17 C-H mesh....44

   (b) 97 x 17 x 17 C-O mesh....44

10. Effect of multigrid on convergence; ONERA M6; 193 x 33 x 33 C-H mesh; $M_\infty = 0.84$; $\alpha = 3.06°$.

   (a) Residual history....45

   (b) Lift history....45

11. Comparison of convergence rate for the 3 schemes; ONERA M6 wing; 97 x 17 x 17 C-H mesh; $M_\infty = 0.84$; $\alpha = 3.06°$ 47

12. Comparison of convergence rate for a V-cycle and a W-cycle; ONERA M6 wing; 97 x 17 x 17 C-H mesh; $M_\infty = 0.84$; $\alpha = 3.06°$ 49

13. Upper surface variation of pressure coefficient; ONERA M6 wing; $M_\infty = 0.84$; $\alpha = 3.06°$.

    (a) 193 x 33 x 33 C-O mesh 52

    (b) 193 x 33 x 33 C-H mesh 52

14. Pressure coefficient contours on upper surface; ONERA M6 wing: $M_\infty = 0.84$; $\alpha = 3.06°$ 53

15. Comparison of experiment and inviscid calculations using Van Leer splittings; ONERA M6 wing; $M_\infty = 0.84$; $\alpha = 3.06°$ 54

16. Comparison of experiment and inviscid calculations using Steger-Warming splittings; ONERA M6 wing; $M_\infty = 0.84$; $\alpha = 3.06°$ 55

17. Multigrid convergence for C-H and C-O meshes; ONERA M6 wing; $M_\infty = 0.699$; $\alpha = 3.06°$, V-cycle 57

18. Comparison of experiment and inviscid calculations using Van Leer splittings; ONERA M6 wing; $M_\infty = 0.699$; $\alpha = 3.06°$ 59

19. Comparison of experiment and inviscid calculations using Steger-Warming splittings, ONERA M6 wing; $M_\infty = 0.699$; $\alpha = 3.06°$ 60

20. Grid for analytic forebody 62

21. Convergence history for analytic forebody; $M_\infty = 1.7$; $\alpha = 0°$ 63

22. Calculated pressure coefficient comparison with experiment at forebody symmetry plane; $M_\infty = 1.7$; $\alpha = 0°$ 64

23. Control volumes for restriction of dependent variables and residual 78

LIST OF TABLES

Table Page

1 Operation counts for solving the implicit operators.....24

2 Summary of results for ONERA M6 wing
$M_\infty = 0.84$; $\alpha = 3.06°$.....50

3 Summary of results for ONERA M6 wing
$M_\infty = 0.699$; $\alpha = 3.06°$.....58

Chapter I

INTRODUCTION

Upwind difference schemes for solving the Euler equations are becoming increasingly popular for several reasons. The time-dependent Euler equations form a system of hyperbolic equations and upwind differencing models the characteristic nature of the equations in that information at each grid point is obtained from directions dictated by characteristic theory. Some of the methods include the λ-method,[1] the split-coefficient method,[2] flux-vector splitting,[3,4,5] and flux-difference splitting.[6] These methods can be classified, in general, as upwind methods and have the advantage of being naturally dissipative. Separate spatial dissipation terms, such as are generally required in a central difference method to overcome oscillations or instabilities arising in regions of strongly varying gradients, need not be added.

While the λ-method and the split-coefficient method closely mimic the method of characteristics, they are both applied to the nonconservative form of the equations and consequently require the use of shock-fitting techniques to obtain the correct location and strength of shocks in transonic flows. Use of the conservation-law form allows shock waves to be captured as weak solutions to the governing equations and circumvents the difficulty in applying shock-fitting techniques to arbitrary flows. Both the flux-difference-splitting and flux-vector-splitting methods can be applied to the conservation-law form.

The particular upwind method used in the current work is the flux vector splitting method in which the flux vectors are split into forward and backward contributions based on an eigenvalue decomposition and

differenced accordingly. The splittings investigated include those of Steger-Warming[3,4,7] and Van Leer.[5,8,9] The advantages of flux splitting are obtained at the cost of increased computational work in comparison to unsplit methods, since two sets of fluxes are computed for each coordinate direction and implicit schemes require two sets of flux Jacobians (e.g. $\partial F^+/\partial Q$ and $\partial F^-/\partial Q$) for consistent linearization of the fluxes. In addition, the split fluxes and flux Jacobians are also generally more complicated than the unsplit terms owing to the branching involved with eigenvalue sign changes.

In order to offset the additional computational work of the upwind methods, it is highly desirable to accelerate the convergence rate, especially when only steady-state solutions are sought; the objective is to reduce the computer time required while still maintaining the high level of robustness and accuracy attained from upwind differencing. Accelerating the convergence rate becomes increasingly important as the mesh is refined since the log of the spectral radius for single grid methods generally varies linearly with the mesh size, making computations on very fine meshes impractical.

One method which has been successful in accelerating the convergence rate of elliptic problems, attaining a spectral radius independent of the mesh spacing, is the multigrid method.[10,11] Although most of the existing theory on multigrid methods pertains specifically to elliptic equations, it has been shown in several references[12-22] that the multigrid method can greatly accelerate the convergence rate of numerical schemes used for solving the Euler equations.

One of the earliest applications of multiple grids in solving the Euler equations, was presented by Ni who used coarse grids to rapidly propagate corrections throughout the domain.[12] His original idea was first incorporated into a one-step Lax-Wendroff method and later extended for use into predictor-corrector type methods by Johnson.[13] Johnson[14,15] and Chima and Johnson[16] subsequently used the method to calculate both inviscid and viscous flows over several two dimensional geometries. In 1984, Mulder applied a linear multigrid scheme to the Euler equations in two space dimensions using upwind differencing to calculate flow over a circular arc and for a weakly barred galaxy.[17] Jesperson also used upwind differencing in two spatial dimensions to calculate flow over airfoils.[18] In this approach, the Euler equations were solved by Newton iteration where the linear system arising at each step was solved using multigrid. One of the first uses of the nonlinear multigrid method in accelerating the convergence rate for both the two and three dimensional Euler equations was reported by Jameson who used central differencing in a four stage Runge-Kutta algorithm to advance the solution.[19,20] In two dimensions, recent work by Jameson and Yoon also used central differencing and incorporated the multigrid algorithm into some implicit schemes with good success.[21,22]

The purpose of the current investigation is to combine the full approximation scheme (FAS) multigrid method with flux vector splitting to obtain efficient solutions to the Euler equations in three-dimensions. The full approximation scheme for a general nonlinear problem is discussed as well as its implementation for the Euler equations. Both V and W type cycling strategies are also

investigated. Several smoothing algorithms are given involving mostly vectorizable computations on the VPS-32 supercomputer at NASA Langley. In addition, both the splittings of Steger-Warming and Van Leer are considered for splitting the flux vectors into upwind components. Numerical pressure distributions are compared with available experiment for subsonic and transonic flow over the ONERA M6 wing and supersonic flow over an analytically defined forebody.

Chapter II

EULER SOLUTION METHOD

2.1 Euler Equations in Generalized Coordinates

The governing equations are the time-dependent equations of ideal gas dynamics, i.e., the Euler equations, which express the conservation of mass, momentum, and energy for an inviscid nonconducting gas in the absence of external forces. The conservation form of the equations in generalized coordinates is given by

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = 0 \tag{2.1}$$

where

$$\hat{Q} = \frac{Q}{J} = \frac{1}{J}\begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{Bmatrix} \tag{2.2}$$

$$\hat{F} = \frac{1}{J}\begin{Bmatrix} \rho U \\ \rho U u + \xi_x p \\ \rho U v + \xi_y p \\ \rho U w + \xi_z p \\ (e + p)U \end{Bmatrix} \tag{2.3}$$

$$\hat{G} = \frac{1}{J}\begin{Bmatrix} \rho V \\ \rho V u + \eta_x p \\ \rho V v + \eta_y p \\ \rho V w + \eta_z p \\ (e + p)V \end{Bmatrix} \tag{2.4}$$

$$\hat{H} = \frac{1}{J} \begin{Bmatrix} \rho W \\ \rho W u + \zeta_x p \\ \rho W v + \zeta_y p \\ \rho W w + \zeta_z p \\ (e + p) W \end{Bmatrix} \tag{2.5}$$

The pressure p is related to the conserved variables through the ideal gas law

$$p = (\gamma - 1) \; [e - \rho \; (u^2 + v^2 + w^2)/2] \tag{2.6}$$

The equations have been generalized from Cartesian coordinates using a steady transformation of the type

$$\xi = \xi(x, y, z), \; \eta = \eta(x, y, z), \; \zeta = \zeta(x, y, z), \; \tau = t \tag{2.7}$$

where the contravariant velocity components are

$$U = \xi_x \, u + \xi_y \, v + \xi_z \, w \tag{2.8a}$$

$$V = \eta_x \, u + \eta_y \, v + \eta_z \, w \tag{2.8b}$$

$$W = \zeta_x \, u + \zeta_y \, v + \zeta_z \, w \tag{2.8c}$$

The transformation to generalized coordinates is given in Appendix A.

The equations, while written in generalized coordinates, are used in a finite-volume formulation. Equation (2.1) can be interpreted as describing the balance of mass, momentum, and energy over an arbitrary control volume. In this connection, the vectors grad(ξ)/J, grad(η)/J,

and grad(ζ)/J represent directed areas of cell interfaces in the ξ, η, and ζ directions and the Jacobian J represents the inverse of the cell volume. Likewise, the quantities $\rho U/J$, $\rho V/J$, and $\rho W/J$ represent the mass flux crossing the cell interfaces in the ξ, η, and ζ directions.

2.2. Flux Vector Splitting

The upwind differencing in the present work is effected through the technique of flux-vector splitting. The generalized fluxes $\hat{F}$, $\hat{G}$, and $\hat{H}$ are split into forward and backward contributions according to the signs of the eigenvalues of the Jacobian matrices and differenced accordingly. For example, the flux in the ξ direction can be differenced as

$$\delta_\xi \hat{F} = \delta_\xi^- \hat{F}^+ + \delta_\xi^+ \hat{F}^- \tag{2.9}$$

since $\hat{F}^+$ has all non-negative eigenvalues and $\hat{F}^-$ has all non-positive eigenvalues. For the current study, two methods of splitting the flux vectors into upwind components are considered. Although the details of each method can be found in references 3, 4, 5, 7, and 9 both methods are briefly discussed below.

The first method presented is the technique outlined by Steger and Warming in reference 3. Since the flux vectors are homogeneous functions of degree one in $\hat{Q}$ they can be expressed in terms of their Jacobian matrices. Considering the flux vector in the ξ-direction for example, $\hat{F}$ can be written as

$$\hat{F} = \hat{A}\hat{Q} = \frac{\partial \hat{F}}{\partial \hat{Q}} \hat{Q} \tag{2.10}$$

Using a similarity transformation, equation (2.10) can be rewritten as

$$\hat{F} = \hat{A}\hat{Q} = T\Lambda T^{-1}\hat{Q} \tag{2.11}$$

The matrix Λ is a diagonal matrix composed of the eigenvalues of $\hat{A}$ and is given by

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & 0 \\ 0 & 0 & 0 & \lambda_4 & 0 \\ 0 & 0 & 0 & 0 & \lambda_5 \end{bmatrix} \tag{2.12}$$

$$\begin{aligned} \lambda_{1,2,3} &= U = \xi_x u + \xi_y v + \xi_z w \\ \text{where } \lambda_4 &= U + |\text{grad}(\xi)|a \\ \lambda_5 &= U - |\text{grad}(\xi)|a \end{aligned} \tag{2.13}$$

The eigenvalues can then be decomposed into non-negative and non-positive components

$$\lambda_i = \lambda_i^+ + \lambda_i^- \tag{2.14}$$

where

$$\lambda_i^{\pm} = \frac{\lambda_i \pm |\lambda_i|}{2} \tag{2.15}$$

Similarly, the eigenvalue matrix Λ can be decomposed into

$$\Lambda = \Lambda^{+} + \Lambda^{-} \tag{2.16}$$

where Λ^{+} is made up of the non-negative contributions λ_i^{+} and Λ^{-} is constructed of the non-positive contributions λ_i^{-}. This splitting of the eigenvalue matrix, combined with equation (2.11) allows the flux vector $\hat{F}$ to be rewritten as

$$\hat{F} = T(\Lambda^{+} + \Lambda^{-})T^{-1}\hat{Q} = (\bar{A}^{+} + \bar{A}^{-})\hat{Q} = \hat{F}^{+} + \hat{F}^{-} \tag{2.17}$$

The flux vector $\hat{F}$ has three distinct eigenvalues given by (2.13) and can therefore be written as a sum of three subvectors, each of which has a distinct eigenvalue as a coefficient.[7]

$$\hat{F} = \hat{F}_1 + \hat{F}_2 + \hat{F}_3 \tag{2.18}$$

where

$$\hat{F}_1 = \lambda_1 \frac{\gamma-1}{J\gamma} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \frac{\rho}{2}(u^2 + v^2 + w^2) \end{Bmatrix} \tag{2.19}$$

$$\hat{F}_{2,3} = \lambda_{4,5} \frac{1}{J2\gamma} \begin{Bmatrix} \rho \\ \rho u \pm \rho a \hat{\xi}_x \\ \rho v \pm \rho a \hat{\xi}_y \\ \rho w \pm \rho a \hat{\xi}_z \\ e + p \pm \frac{\rho a U}{|\mathrm{grad}(\xi)|} \end{Bmatrix} \tag{2.20}$$

and the direction cosines of the directed interface in the ξ-direction are

$$\hat{\xi}_x = \xi_x / |\mathrm{grad}(\xi)| \tag{2.21a}$$

$$\hat{\xi}_y = \xi_y / |\mathrm{grad}(\xi)| \tag{2.21b}$$

$$\hat{\xi}_z = \xi_z / |\mathrm{grad}(\xi)| \tag{2.21c}$$

The forward and backward flux vectors $\hat{F}^+$ and $\hat{F}^-$ are formed from equations (2.18), (2.19), and (2.20) by inserting $\lambda_i = \lambda_i^+$ and $\lambda_i = \lambda_i^-$, respectively. It should be noted that for supersonic and sonic flow, in the ξ-direction, i.e., $|M_\xi| = |\bar{u}/a| > 1$, where $\bar{u} = U/|\mathrm{grad}(\xi)|$ represents the velocity normal to a ξ = constant face, the fluxes in this direction become

$$\begin{aligned} \hat{F}^+ &= \hat{F}, \qquad \hat{F}^- = 0, \qquad M_\xi > 1 \\ \hat{F}^- &= \hat{F}, \qquad \hat{F}^+ = 0, \qquad M_\xi < -1 \end{aligned} \tag{2.22}$$

The split fluxes in the other two directions are easily obtained by interchanging η or ζ in place of ξ.

The fluxes split in this manner above are not continuously differentiable at zeros of the eigenvalues (i.e. sonic and stagnation points).[23] This is illustrated in figure 1, where the split mass flux contributions for the one-dimensional Euler equations, non-dimensionalized by division through ρa, are shown as a function of the Mach number. The gradient discontinuities in the split fluxes are evident as the eigenvalues pass through zero. The lack of differentiability of the split fluxes has been shown in some cases to

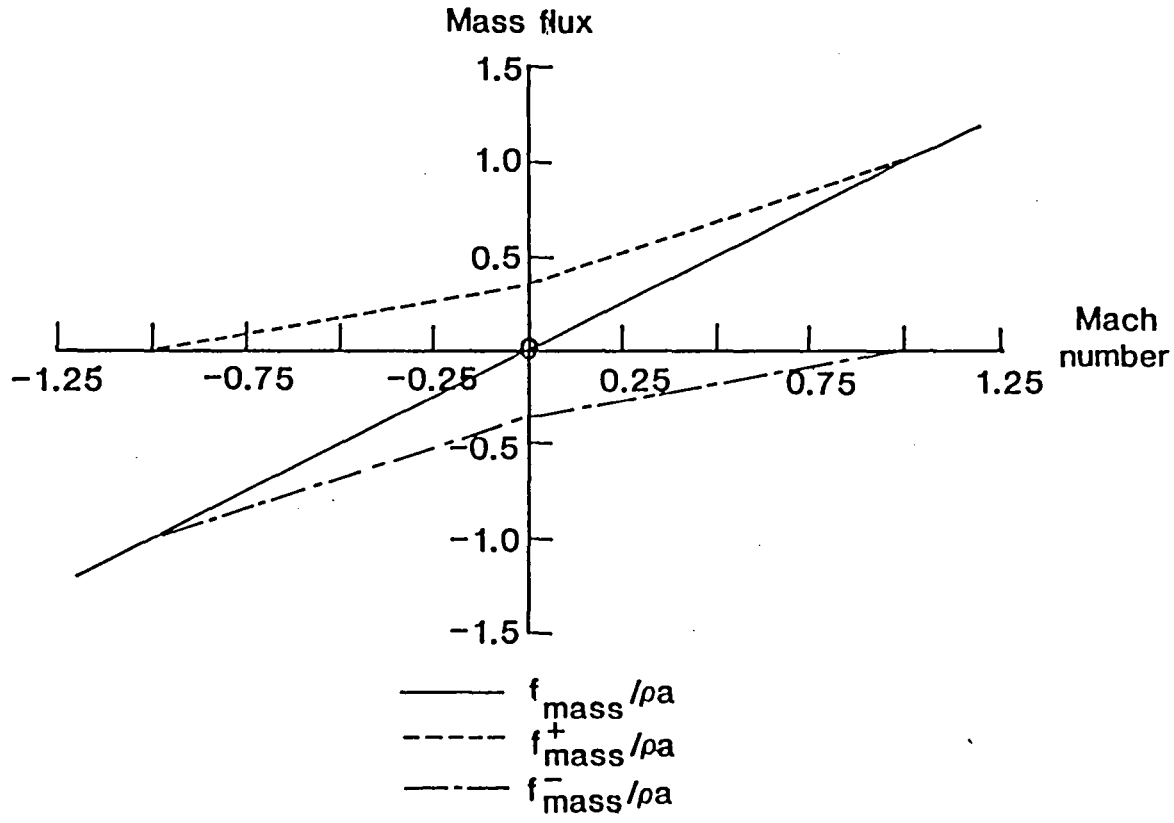


Figure 1. Variation of Steger-Warming split mass flux with Mach number

cause small oscillations or glitches at sonic points but which are rarely noticeable for most aerodynamic applications.

It should also be noted that the Jacobian matrices of $\hat{F}^+$ and $\hat{F}^-$ which are required for proper linearization for an implicit scheme do not have the same eigenvalues as $\bar{A}^+$ and $\bar{A}^-$ defined in equation (2.17).[23] However, the Jacobian matrices of $\hat{F}^+$ and $\hat{F}^-$ do have the same sign as $\bar{A}^+$ and $\bar{A}^-$ so that upwind differencing the spatial derivatives remains appropriate.[23] Although they are easier to form, the use of $\bar{A}^\pm$ in implicit schemes instead of the correct linearizations $\hat{A}^\pm$ has been shown in many cases to cause severe time step limitations.[3,24]

In 1982, a new method of splitting the flux vector was proposed by Van Leer.[5] The approach taken in the derivation was to split the fluxes so that the forward and backward flux contributions transitioned smoothly at eigenvalue sign changes, i.e., near sonic and stagnation points. Just as for the Steger-Warming splitting, it was required that the Jacobian matrices $\frac{\partial \hat{F}^+}{\partial \hat{Q}}$ have non-negative eigenvalues and $\frac{\partial \hat{F}^-}{\partial \hat{Q}}$ have non-positive eigenvalues so that upwind differencing could be used for the spatial derivatives. In addition it was required that both Jacobians have one zero eigenvalue for subsonic Mach numbers which leads to steady transonic shock structures with only two transition zones.[5] In practice, when second-order spatial differencing is used, shocks with only one interior zone are usually obtained.[9] This feature is not observed with the Steger-Warming flux splitting.

The three-dimensional splittings of Van Leer were originally given for Cartesian coordinates. The extension to generalized coordinates is given in Appendix B with the resulting split fluxes given below. Only

the splitting for the flux in the ξ-direction is given, as the others can be obtained similarly. The flux vector $\hat{F}$ is split according to the contravariant Mach number in the ξ-direction, defined above as $M_\xi = \bar{u}/a$. For supersonic flow, $|M_\xi| > 1$,

$$\hat{F}^+ = \hat{F}, \qquad \hat{F}^- = 0, \qquad M_\xi > +1 \tag{2.23}$$
$$\hat{F}^- = \hat{F}, \qquad \hat{F}^+ = 0, \qquad M_\xi < -1$$

and for subsonic flow, $|M_\xi| < 1$

$$\hat{F}^\pm = \frac{|\mathrm{grad}(\xi)|}{J} \begin{Bmatrix} f^\pm_{mass} \\ f^\pm_{mass}[\hat{\xi}_x(-\bar{u} \pm 2a)/\gamma + u] \\ f^\pm_{mass}[\hat{\xi}_y(-\bar{u} \pm 2a)/\gamma + v] \\ f^\pm_{mass}[\hat{\xi}_z(-\bar{u} \pm 2a)/\gamma + w] \\ f^\pm_{energy} \end{Bmatrix} \tag{2.24a}$$

where

$$f^\pm_{mass} = \pm \rho a(M_\xi \pm 1)^2/4 \tag{2.24b}$$

$$f^\pm_{energy} = f^\pm_{mass}[\{-(\gamma - 1)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a$$

$$+ 2a^2\}/(\gamma^2 - 1) + (u^2 + v^2 + w^2)/2] \tag{2.24c}$$

For forming $\hat{F}^\pm$, $\hat{\xi}_x$, $\hat{\xi}_y$, $\hat{\xi}_z$ are given by equation (2.21) and $\bar{u}$ is the velocity normal to a ξ = constant face. The fluxes in the other two directions are easily formed by interchanging ξ with either η or ζ. In figure 2, the non-dimensionalized mass flux, using the Van Leer splitting, is shown as a function of Mach number for the one-dimensional Euler equations. The split fluxes are continuously differentiable at

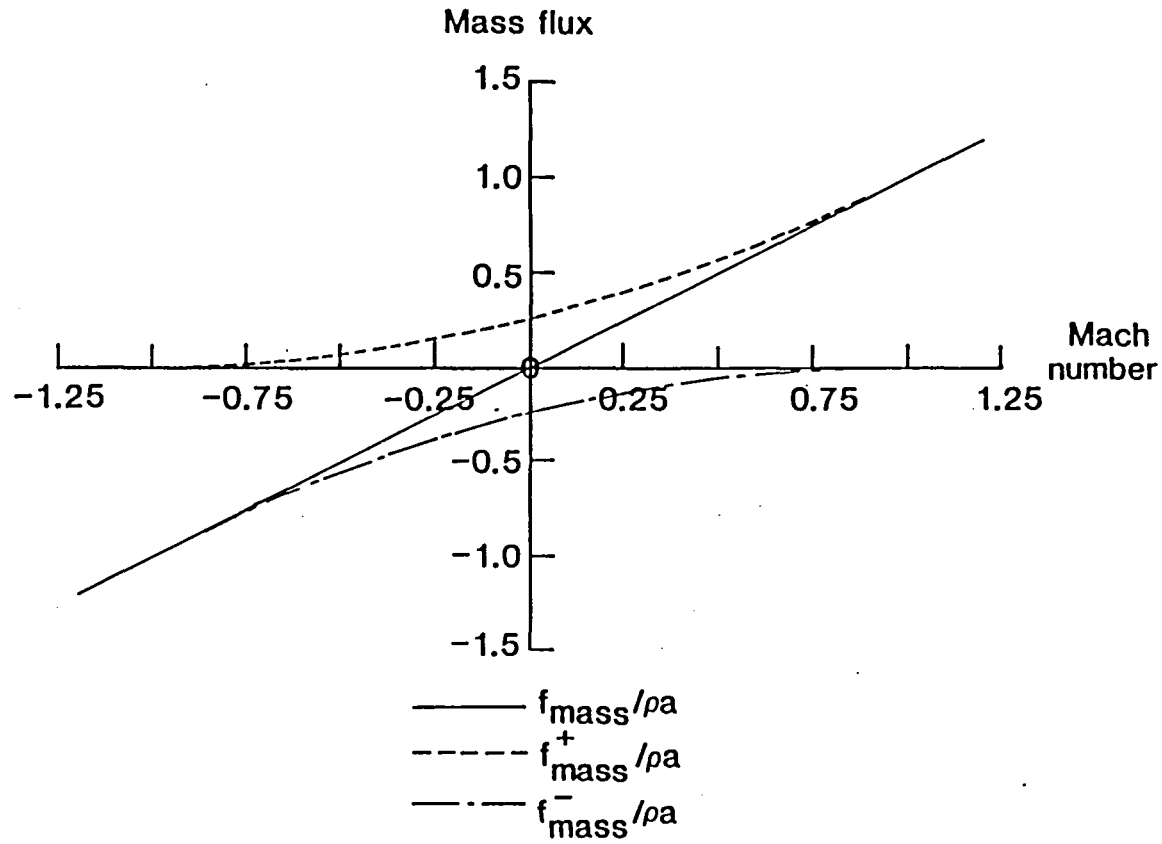


Figure 2. Variation of Van Leer split mass flux with Mach number

sonic and stagnation points; the improvement over the Steger-Warming splitting is apparent.

2.3 Baseline Solution Algorithm

The baseline algorithm for updating the steady Euler equations stems from a backward Euler time integration of the unsteady equations which yields[8]

$$[I + \Delta t(\delta_\xi^- \hat{A}^+ + \delta_\xi^+ \hat{A}^-) + \Delta t(\delta_\eta^- \hat{B}^+ + \delta_\eta^+ \hat{B}^-) + \Delta t(\delta_\zeta^- \hat{C}^+ + \delta_\zeta^+ \hat{C}^-)]\Delta\hat{Q} = -\Delta t R^n$$

where the residual at time level n is given by

$$R^n = \delta_\xi^- \hat{F}^+ + \delta_\xi^+ \hat{F}^- + \delta_\eta^- \hat{G}^+ + \delta_\eta^+ \hat{G}^- + \delta_\zeta^- \hat{H}^+ + \delta_\zeta^+ \hat{H}^- \tag{2.25}$$

The split-flux differences in equation (2.25) are implemented as a flux balance across a cell, corresponding to MUSCL type differencing (Monotone Upstream-centered Schemes for Conservation Laws).[25] For example, the flux balance in the ξ-direction across a cell centered at point i,j,k can be written as

$$\begin{aligned}\delta_\xi^- \hat{F}^+ + \delta_\xi^+ \hat{F}^- &= [\hat{F}^+(Q^-) + \hat{F}^-(Q^+)]_{i+1/2} \\ &\quad - [\hat{F}^+(Q^-) + \hat{F}^-(Q^+)]_{i-1/2}\end{aligned} \tag{2.26}$$

The notation $\hat{F}^+(Q^-)_{i+1/2}$ denotes the forward flux evaluated using the metric terms at the cell interface i+1/2 and the conserved state variables on the upwind side of the interface, obtained by a fully-

upwind second-order state variable interpolation:

$$Q^{-}_{i+1/2} = 1.5\ Q_i - 0.5\ Q_{i-1} \tag{2.27a}$$

$$Q^{+}_{i+1/2} = 1.5\ Q_{i+1} - 0.5\ Q_{i+2} \tag{2.27b}$$

As seen in figure 3, $Q^n_{i,j,k}$ denotes the average value of Q in the cell centered on (ξ_i, η_j, ζ_k) at time t^n; for simplicity, wherever the script notation is i,j,k, or n, it is most often dropped.

In equation (2.25), if second order differencing is used on both sides of the equation, Newton iteration for the steady Euler equations is obtained as Δt tends to infinity. The solution however requires the solution of a large banded block matrix at each step which is generally not feasible due to the amount of operations required to invert the system. Even if the differencing on the left hand side of the equation is reduced to first order, which would not effect the second order accuracy of the final solution, the resulting system of equations usually remains uneconomical to solve. Therefore, the solution is obtained using approximate factorization, which splits the implicit operator into a sequence of easily invertible equations.

When using flux vector splitting, there are numerous ways of factoring the implicit operator into a sequence of simpler operators.[3] For the results shown below, three ways of factoring are considered. Each of the schemes uses first order spatial differencing on the implicit side of the equation while second order differencing is maintained for the residual calculations. Since the steady state does

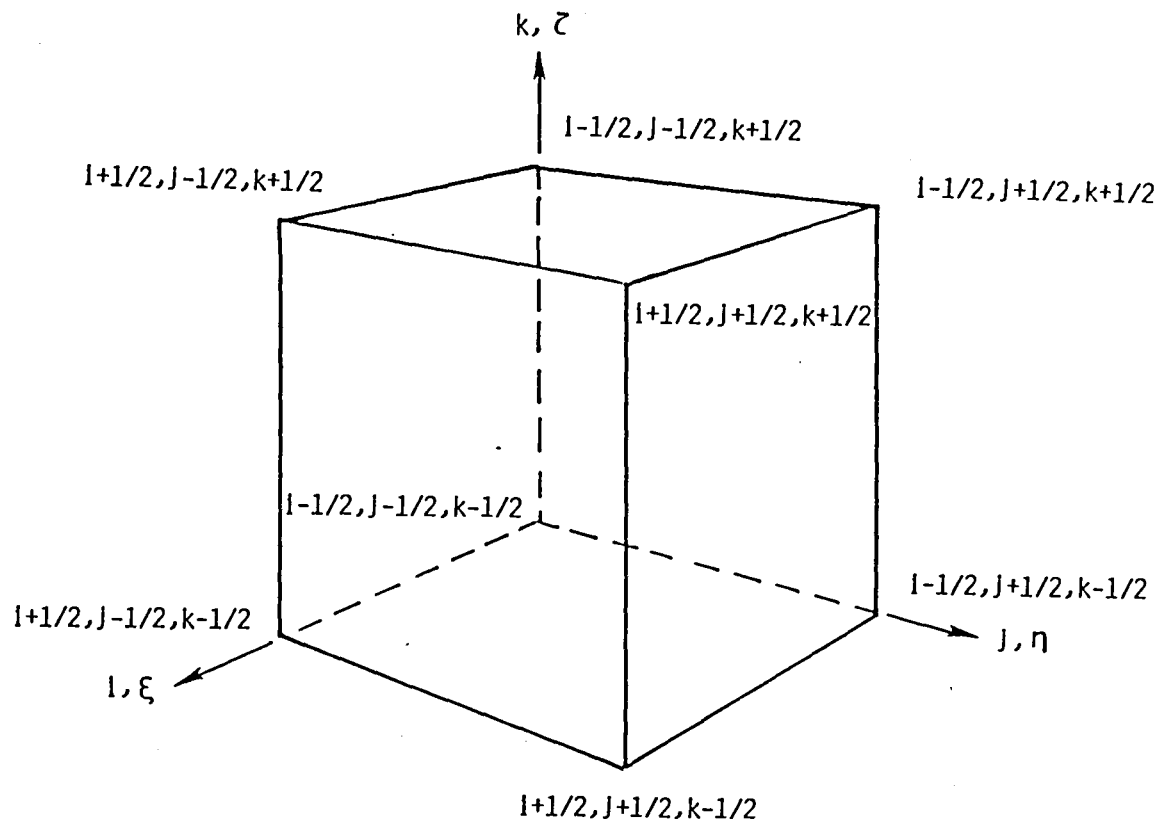


Figure 3. Computational cell indexing

not depend on the differencing of the implicit side, the final steady-state result will be spatially second order accurate. All the schemes employ simple explicit boundary conditions. Since only steady-state solutions are sought, each cell is advanced at its own time step corresponding to a given CFL number defined by

$$CFL = \Delta t[|U|+|V|+|W|+a(|grad(\xi)|+|grad(\eta)|+|grad(\zeta)|)] \tag{2.28}$$

The first scheme considered is a spatially-split algorithm given by

$$\begin{aligned} &[I+\Delta t(\delta_\xi^- \hat{A}^+ + \delta_\xi^+ \hat{A}^-)][I+\Delta t(\delta_\eta^- \hat{B}^+ + \delta_\eta^+ \hat{B}^-)] \\ &[I+\Delta t(\delta_\zeta^- \hat{C}^+ + \delta_\zeta^+ \hat{C}^-)]\Delta\hat{Q} = -\Delta t R^n \end{aligned} \tag{2.29}$$

The computational module for the implicit side of equation (2.29) is shown in figure 4a for the ξ sweep. Since the solution at each point is directly coupled to the two neighboring points, the scheme requires the solution of a system of block tridiagonals. Similarly, the other two factors also require a block tridiagonal inversion. This scheme has the advantage however of being completely vectorizable and viscous effects can be easily incorporated into the left hand side. Since the speed of the VPS-32 is much faster for long vector lengths than it is for short ones, the computations in the present implementation take advantage of the large memory available on the VPS-32 computer and solve the block matrix equations over multiple planes simultaneously, yielding longer vector lengths and faster processing rates. As seen in figure 5, the

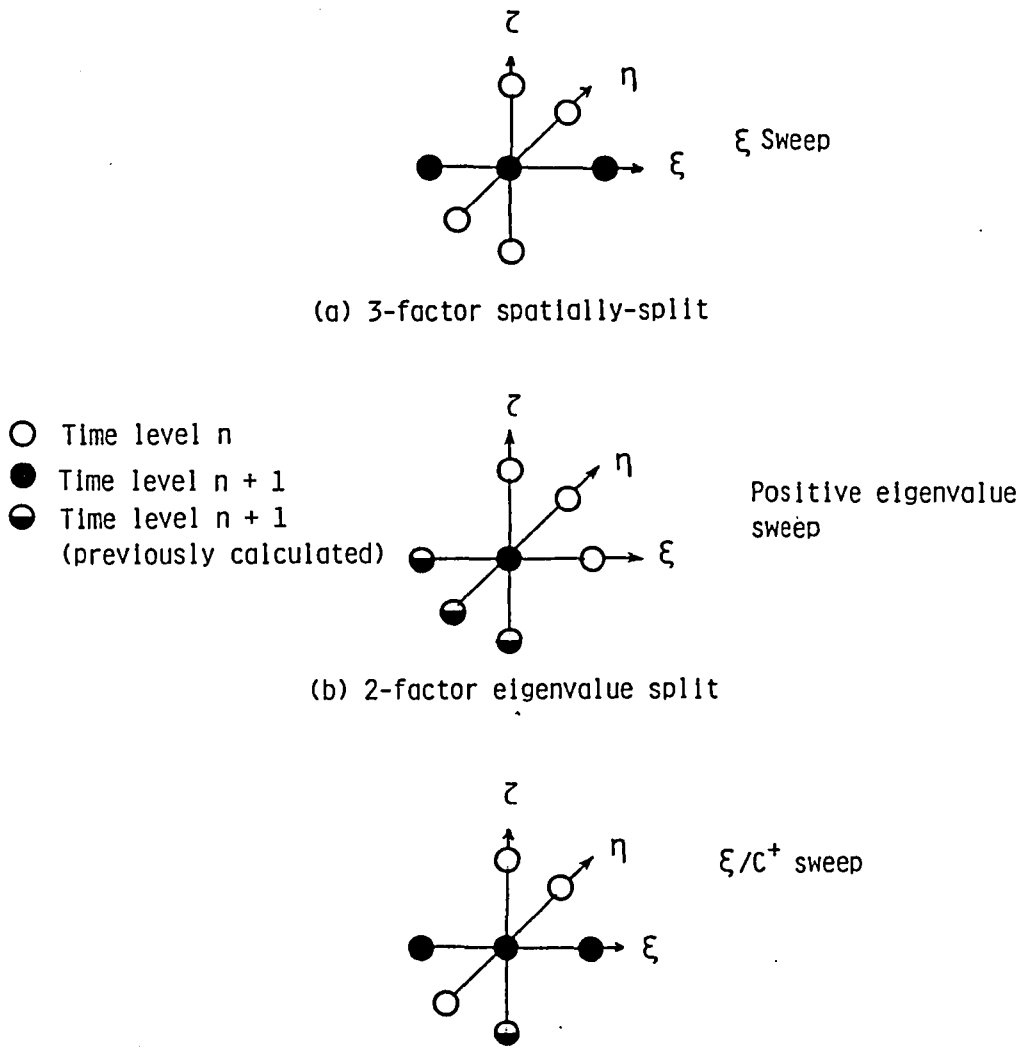


Figure 4. Computational modules

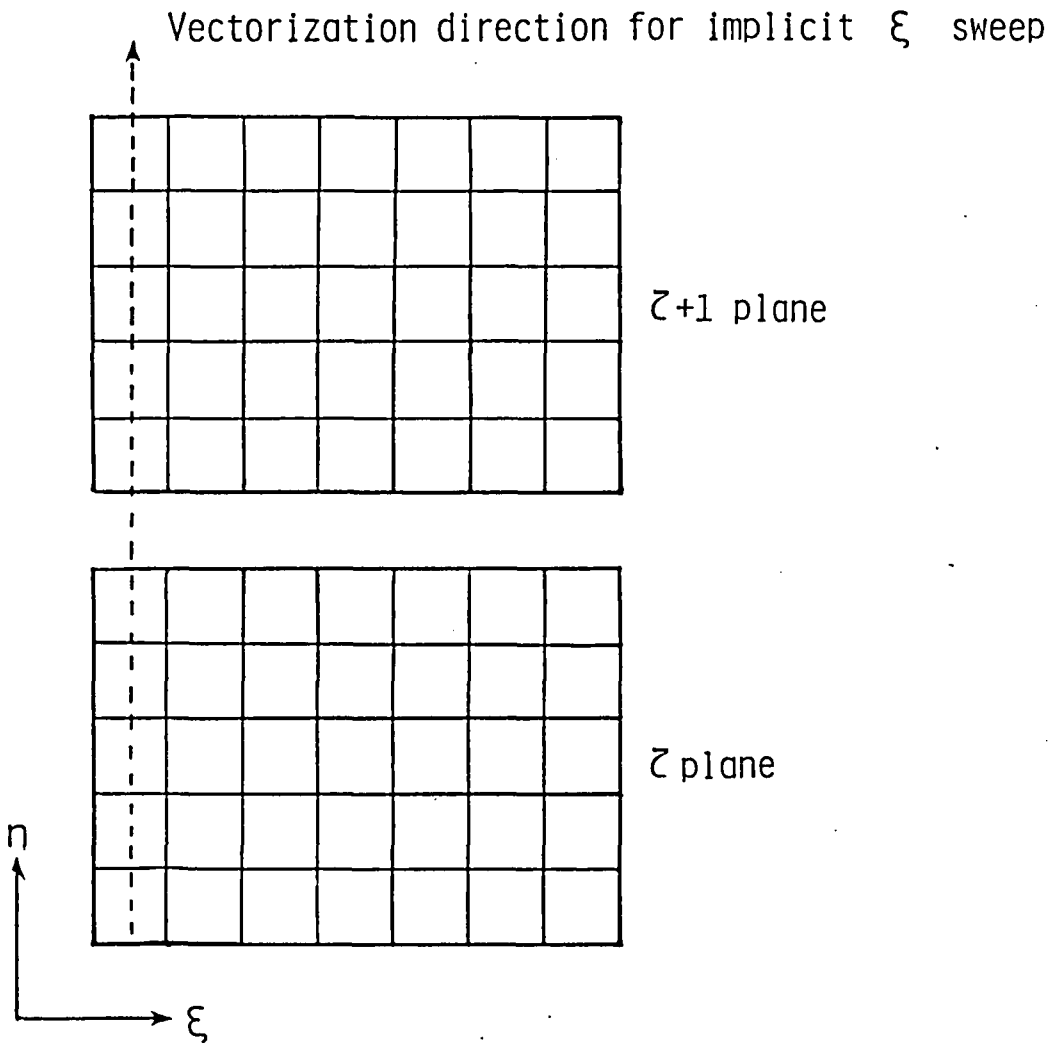


Figure 5. Vectorization of LU decomposition and back-substitution over multiple planes

block tridiagonal matrices can be solved with vector lengths corresponding to the number of lines in a plane times the number of planes taken. The residual calculations, on the other hand, can be made with vector lengths corresponding to the number of points in the grid. To decompose the implicit operator into lower and upper matrices (LU decomposition) and perform the back substitutions requires 695 multiplications and additions per factor resulting in a total of 2085 operations for each sweep through the grid.

The second method considered to factor the left hand side of equation (2.25) is a two-factor method in which the implicit operator is split such that one operator contains the Jacobians with all positive eigenvalues and the other operator contains the Jacobians with all negative eigenvalues. The scheme can be written as

$$
\begin{aligned}
&[I+\Delta t(\delta_\xi^- \hat{A}^+ + \delta_\eta^- \hat{B}^+ + \delta_\zeta^- \hat{C}^+)] \\
&[I+\Delta t(\delta_\xi^+ \hat{A}^- + \delta_\eta^+ \hat{B}^- + \delta_\zeta^+ \hat{C}^-)]\,\Delta\hat{Q} = -\Delta t R^n
\end{aligned}
\tag{2.30}
$$

This scheme only requires the solution of block lower triangular equations. Each factor is solved by starting at one corner of the grid, solving for each point by marching across the field to the opposite corner. From the computational module shown in figure 4b for the first factor in equation (2.30), the solution at the center node requires that the solution of each of the points behind it be previously obtained. These terms are taken to the right hand side of the equation and added to the residual. A similar procedure is carried out for the second

factor. For the present implementation of this scheme, only 270 operations are required for each factor to invert the left hand side and perform the back substitutions and most of the algorithm is vectorizable. This scheme requires that a 5 x 5 matrix be inverted at each point in the grid which can be carried out with vector lengths corresponding to the number of points in the grid. However, since the solution of each plane requires that the solution of the previous plane be known, the back substitution cannot be performed over multiple planes. The maximum vector lengths in the back substitution process corresponds to the number of points in a plane. The only scalar computations correspond to back substitution along a line requiring roughly a third of the total operations. Although the scheme requires only about 25 percent of the operations required for the spatially split scheme, the scalar computations degrade the processing rate significantly so that the overall processing rate on the VPS-32 is about twice as slow as for the spatially split scheme.

The last scheme considered is another two factor scheme which is spatially split in two directions, with the third direction split according to the sign of its eigenvalues. The resulting scheme, which is referred to as combination splitting, is given by

$$[I+\Delta t(\delta_\xi^-\hat{A}^+ + \delta_\xi^+\hat{A}^- + \delta_\zeta^-\hat{C}^+)][I+\Delta t(\delta_\eta^-\hat{B}^+ + \delta_\eta^+\hat{B}^- + \delta_\zeta^+\hat{C}^-)]\Delta\hat{Q} = -\Delta t R^n \qquad (2.31)$$

and the computational module for the first factor is shown in figure 4c. This scheme also requires the solution of block tridiagonal systems, requiring about 695 multiplications and divisions for each

factor. This scheme is completely vectorizable; however, as in the previous two-factor scheme, the solution of each plane requires that the solution of the previous plane be known thereby eliminating the possibility of extending the vector operations over several planes. The result is that even though this scheme requires only two-thirds of the operations of the 3-factor scheme, the computational rate is actually degraded by about ten percent. A summary of operations required to solve the left hand side for each of the three schemes is given in table 1.

2.4 Boundary Conditions

The boundary conditions for the solutions presented below are applied explicitly. On the body, the normal velocity is set to zero while the pressure and density are determined by extrapolation from the interior. In the farfield, for subsonic flow, the velocity normal to the boundary and the speed of sound are obtained from two locally one-dimensional Riemann invariants given by

$$R^{\pm} = \bar{u} \pm \frac{2a}{\gamma-1} \tag{2.32}$$

These are considered constant along characteristics defined by

$$\left(\frac{d\bar{x}}{dt}\right)^{\pm} = \bar{u} \pm a \tag{2.33}$$

For subsonic conditions at the boundary, R^- can be evaluated locally from conditions outside the computational domain, and R^+ locally from the interior of the domain. The two Riemann invariants can be added and subtracted to determine a local normal velocity and speed of sound at

Scheme	Operations required per factor and vector length		
	LU decomposition	Additional right-hand terms*	Backsubstitution
3-factor spatially split	550 (multiplane)	0	145 (multiplane)
2-factor combination split	550 (multiplane)+	50 (plane)	145 (single plane)

Scheme	Operations required per factor and vector length		
	Additional right-hand terms*	LU decomposition**	$(LHS)^{-1} \cdot (RHS)$
2-factor eigenvalue split	50 (plane) 50 (line) 50 (scalar)	75 (multiplane)++	45 (scalar)

*Terms required on right-hand side (RHS) in addition to residual

**LU decomposition of a 5 x 5 matrix only

+length corresponds to the number of lines in a plane times the number of planes

++length corresponds to the number of points in a plane times the number of planes

Table 1. Operation counts for solving the implicit operators.

the boundary respectively. Depending on the sign of the normal velocity (inflow or outflow where inflow at the boundary corresponds to $\bar{u} < 0$), the entropy and tangential velocities extrapolated from the exterior or interior of the domain are used with the speed of sound to determine the density and pressure on the boundary. The Cartesian velocities are obtained by decomposing the velocity tangential and normal to the boundary.

For supersonic free-stream conditions along inflow boundaries, quantities are extrapolated from the exterior; along outflow boundaries, quantities are extrapolated from the interior of the computational domain.

2.5 Stability Analysis

In order to examine the stability characteristics of the three-dimensional approximate-factorization algorithms considered above, a Fourier analysis is conducted on the complete system of equations in Cartesian coordinates.[26,8] Because of the mixed signs of the eigenvalues of the Euler equations and the fact that the three-dimensional Euler equations cannot be diagonalized to yield a system of convection equations, stability analysis of the scalar convection equation is not sufficient to determine stability properties of the three schemes. The complete system of equations can be written as

$$M\Delta Q = -L = -\Delta t R^n \tag{2.34}$$

where for Cartesian coordinates

$$R^n = \delta_x^- F^+ + \delta_x^+ F^- + \delta_y^- G^+ + \delta_y^+ G^- + \delta_z^- H^+ + \delta_z^+ H^- \tag{2.35}$$

and M is an implicit operator corresponding to the scheme considered. Linearizing the residual R^n as

$$\begin{aligned} R^n = {} & A^+ \delta_x^- Q^n + A^- \delta_x^+ Q^n + B^+ \delta_y^- Q^n \\ & + B^- \delta_y^+ Q^n + C^+ \delta_z^- Q^n + C^- \delta_z^+ Q^n \end{aligned} \tag{2.36}$$

and assuming that the Jacobians are locally constant, the stability can be analyzed by letting

$$Q^n = \lambda^n U_o \, e^{i\beta x} e^{i\gamma y} e^{i\alpha z} \tag{2.37}$$

where U_o is an initial constant vector. Upon substitution into equation (2.34) and dividing out the common factors, the generalized eigenvalue problem for λ, which is the vector of amplification factors, can be obtained

$$(\hat{M}-\hat{L})v = \hat{M}\lambda v \tag{2.38}$$

where $\hat{M}$ and $\hat{L}$ are the Fourier symbols of M and L, respectively. The stability characteristics are determined by cycling through a fixed number of each of the spatial frequencies, in this case sixteen frequencies, in the range

$$0 < \beta\Delta x, \ \gamma\Delta y, \ \alpha\Delta z < 2\pi$$

for a series of CFL numbers between 0.1 and 50. The generalized eigenvalue problem is solved each time using a routine from the International Mathematics and Statistics Library (IMSL).[27] Each time, the maximum eigenvalue, average eigenvalue, and the smoothing factor are determined, where the smoothing factor, defined as

$$\mu = \max_{\pi/2 < \max(\beta\Delta x,\ \gamma\Delta y,\ \alpha\Delta z) < 3\pi/2} \{|\lambda|\} \tag{2.39}$$

corresponds to the damping of the high frequencies and serves as an indication of how effectively the multigrid procedure can accelerate convergence for a given scheme.

Results are shown using the Van Leer splitting for each of the schemes given above. Identical cases were run using the Steger-Warming splittings with little change in the results. Each result was obtained by using first order differencing on the implicit side of the equation and fully-upwind, second-order differencing for the residual computations. All the calculations assume Cartesian coordinates, a Mach number of 0.8, and zero degrees yaw and angle of attack.

The average eigenvalue, the smoothing factor, and the maximum eigenvalue is shown in figure 6 for the three schemes given above. For the 3-factor, spatially split scheme, shown in figure 6a, the maximum eigenvalue indicates that this scheme is conditionally stable with a maximum CFL number of approximately 20. The minimum smoothing factor occurs at a CFL of about 5 which is somewhat less than the CFL number where the maximum eigenvalue is lowest. In contrast to the spatially split algorithm, both of the 2-factor schemes shown in figure 6b and 6c appear to be stable for all CFL numbers considered. The maximum

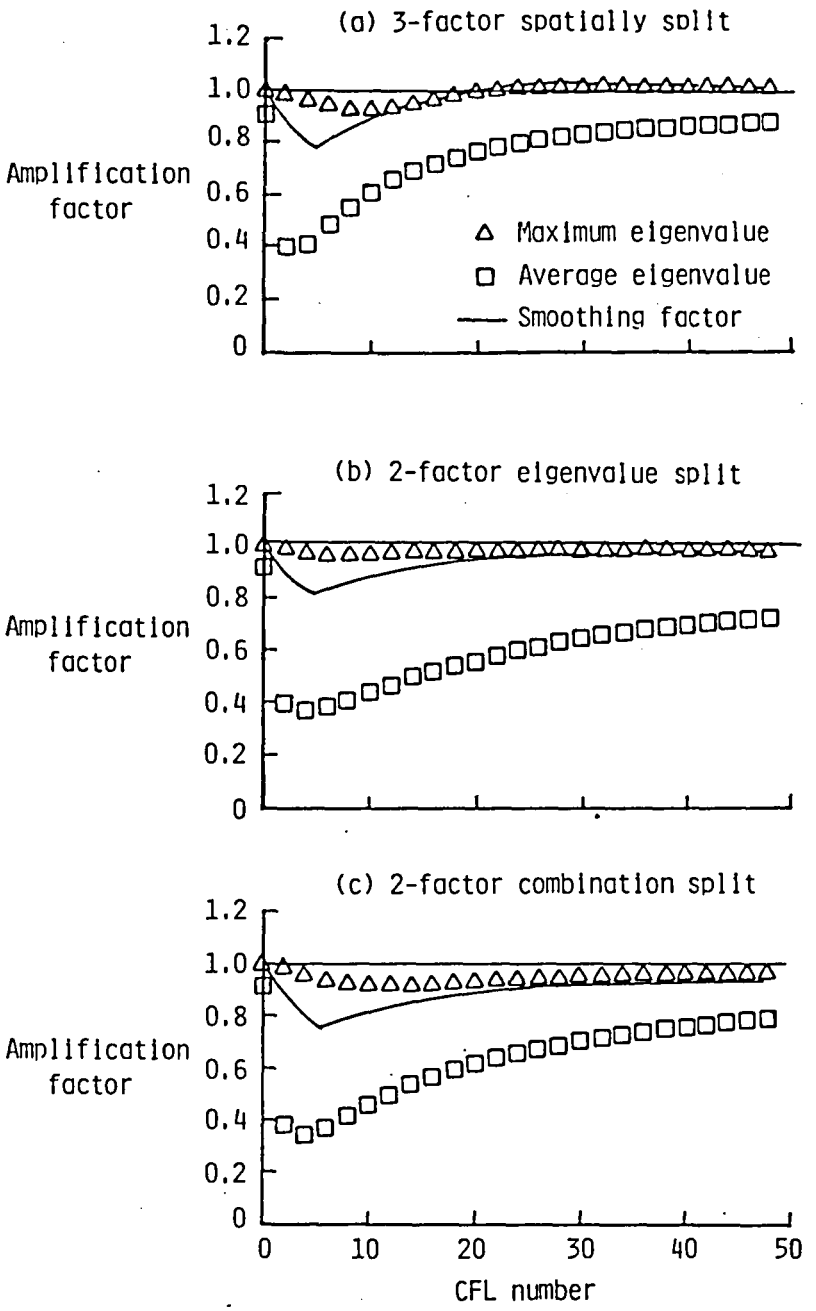


Figure 6. Stability analysis of three-dimensional approximate factorization schemes; $M_\infty = 0.8$; $\alpha = 0°$.

eigenvalues and smoothing factors also exhibit less sensitivity to the CFL number than the 3-factor scheme with minimum smoothing factors also occurring at a CFL number of about 5. However, the 2-factor eigenvalue split scheme has a somewhat higher smoothing factor than the other two schemes which are therefore more appropriate for multigrid applications.

Chapter III

MULTIGRID

3.1 General Algorithm

The multigrid method used in the current study is the full approximation scheme (FAS) which appears in many references[10,28,29,30] and is summarized below. It is most easily understood by first considering the solution of a general nonlinear equation

$$L(Q) = S \tag{3.1}$$

Equation (3.1) is to be solved numerically by dividing the domain into discrete cells yielding a system of equations to be solved simultaneously at each point as

$$L_N(Q_N) = S_N \tag{3.2}$$

where Q_N is the exact solution to the discretized system and L_N is the discrete analog of the operator L. If initial conditions are close enough to the final solution, equation (3.2) could be solved iteratively using Newton iteration. This approach however may be prohibitively expensive if the number of unknowns is large as typically occurs in multi-dimensional problems. Many other iterative schemes have therefore been devised which require significantly fewer operations. After a few iterations however, these methods generally exhibit a slow convergence rate, reducing the residuals by a very small amount each time.[29] The

reason for the slow asymptotic convergence rate is inadequate damping of the low frequency errors.[11]

The multigrid method efficiently damps the low frequency errors using a sequence of grids G_0, G_1, ... G_N where G_N denotes the finest grid from which successively coarser grids can be formed by deleting every other mesh line. In this context, the high frequency error components on a given grid are those which cannot be resolved on the next coarser mesh due to the increased grid spacing. If an iterative method is chosen which quickly damps the high frequency errors on a given grid, the remaining errors will be the lower frequency smooth components. The sequence of coarser grids can then be used in accelerating the convergence rate on the finest grid by reducing the remaining low frequency errors since some of these same frequencies appear as high frequency errors on a coarser grid. Therefore, the errors on the fine grid which are responsible for slow convergence are quickly damped using the coarser grids where the computations are relatively cheap.

In order to use the coarser grids, it is necessary to obtain an equation on the fine mesh which can be accurately represented by the coarser mesh. It is important to first realize that neither the high frequency solution or error components on the fine grid can generally be resolved on a coarser grid. The high frequency errors however can be sufficiently damped on a fine grid using a variety of iterative schemes so that the remaining errors will mainly be composed of smooth low frequency components which can be adequately represented on coarser meshes. For this reason, it is necessary to obtain an equation on the

fine mesh in terms of the errors.

When solving in an iterative fashion, equation (3.2) is solved approximately at each step as

$$L_N(q^c{}_N) = S_N + R_N \tag{3.3}$$

where $q^c{}_N$ is the most current approximation to Q_N and R_N is the residual which will only be zero when $q_N^c = Q_N$ and hence the exact solution is obtained. Subtracting (3.3) from (3.2) yields an equation on the finest grid in terms of the residual

$$L_N(Q_N) - L_N(q^c{}_N) = -R_N \tag{3.4}$$

If the high frequency errors have been previously smoothed, then the fine grid residual equation (3.4) can be adequately approximated on a coarser mesh by

$$L_{N-1}\ (Q_{N-1}) = \hat{I}_N^{N-1}(-R_N) + L_{N-1}\ (I_N^{N-1} q_N{}^c) \tag{3.5}$$

Where I_N^{N-1} and $\hat{I}_N^{N-1}$ are restriction operators for transferring the dependent variables and the residual from the fine grid to the coarse grid. Here, $I_N^{N-1} q_N{}^c$ serves as an initial approximation to the solution on the coarse mesh whereas Q_{N-1} is the exact solution which is the sum of the initial approximation and a correction.[30] Since the full solution is computed and stored on each grid level as opposed to only the corrections, this is referred to as the full approximation scheme (FAS).

On the coarser grid, equation (3.5) could possibly be solved exactly using a variety of numerical techniques to obtain Q_{N-1} from which the correction can be formed as

$$V_{N-1} = Q_{N-1} - I_N^{N-1} q_N^c \tag{3.6}$$

This can then be passed up to the fine grid and used as a correction to $q_N{}^c$ which is replaced by its previous value plus the correction

$$q^c{}_N \leftarrow q^c{}_N + I_{N-1}^N \; V_{N-1} \tag{3.7}$$

This process yields a simple FAS two level algorithm where the operations on the coarse grid (equations 3.5-3.7) which are used to update the fine grid solution are termed the coarse grid correction. Often however, the exact solution of (3.5) can still be quite expensive to obtain. Also, since the correction on the coarse grid serves only as an approximation to the fine grid correction, the exact solution of (3.5) is not required. Therefore, instead of solving (3.5) to completion, several iterations can be carried out to get a reasonable approximation to Q_{N-1}. After each iteration of equation (3.5), the equation satisfied by q_{N-1}^c is given by

$$L_{N-1}(q_{N-1}^c) = I_N^{N-1}(-R_N) + L_{N-1}(I_N^{N-1} q_N^c) + R_{N-1} \tag{3.8}$$

which differs from the solution of (3.5) only by the residual term which will be zero when $q_{N-1}^c = Q_{N-1}$. If the errors are smooth, then

subtraction of (3.8) from (3.5) yields an equation which can be well represented on yet a coarser mesh, G_{N-2}. Writing this equation on G_{N-2} yields

$$L_{N-2}(Q_{N-2}) = L_{N-2}(I_{N-1}^{N-2}\, q_{N-1}^{c}) + \hat{I}_{N-1}^{N-2}\,(-R_{N-1}) \tag{3.9}$$

where equation (3.8) defines R_{N-1}. The solution to (3.9) may be solved for exactly, approximated by several iterations, or alternately, by introducing more coarse grid levels. On all coarse grids, one or more FAS cycles (smoothing followed by coarse grid correction) are done. In this manner, each of the coarse meshes is used to obtain a correction for the solution on the next finer mesh. Since only the equations for smooth error components may be represented well on coarser grids, it is important to only pass corrections from a coarse grid up to the next finer one and not the full solution.[28] As a result of using the coarser meshes, a fast convergence rate may be obtained since the low frequency error components on the fine grid are quickly damped on the coarser meshes.

Note that equation (3.5) can be recast using equation (3.3) as

$$L_{N-1}(Q_{N-1}) = S_{N-1} + \tau_{N-1} = P_{N-1} \tag{3.10}$$

where

$$S_{N-1} = \hat{I}_{N}^{N-1}\, S_{N} \tag{3.11}$$

$$\tau_{N-1} = L_{N-1}(I_{N}^{N-1} q_{N}{}^{c}) - \hat{I}_{N}^{N-1}(L_{N}(q_{N}^{c})) \tag{3.12}$$

Here τ_{N-1} is the relative truncation error (or defect correction) between the grids so that the solution on the coarse grid is driven by the fine grid and the defect correction accounts for the difference in truncation error between the coarse and fine grids.[28] The analogous equation for (3.9) is given by

$$L_{N-2}(Q_{N-2}) = S_{N-2} + \tau_{N-2} \tag{3.13}$$

where

$$S_{N-2} = \hat{I}^{N-2}_{N-1}\, S_{N-1} \tag{3.14}$$

$$\tau_{N-2} = L_{N-2}(I^{N-2}_{N-1}\, q^{c}_{N-1}) - \hat{I}^{N-2}_{N-1}(L_{N-1}(q^{c}_{N-1})) + \hat{I}^{N-2}_{N-1}\, \tau_{N-1} \tag{3.15}$$

Note that the relative truncation error on the N-2 grid is the sum of the relative truncation error between grids N and N-1, as well as N-1 and N-2.

3.2 Algorithm for the Euler Equations

For the steady Euler equations in generalized coordinates, equation (3.2) can be written as

$$L_N(Q_N) = \delta^-_\xi \hat{F}^+ + \delta^+_\xi \hat{F}^- + \delta^-_\eta \hat{G}^+ + \delta^+_\eta \hat{G}^- + \delta^-_\zeta \hat{H}^+ + \delta^+_\zeta \hat{H}^- = 0 \tag{3.16}$$

In the multigrid solution process, a forcing function arises on the coarse grids from restricting the residual equation on a fine mesh down to the coarser one. Since for the Euler equations, S = 0 in (3.2), the forcing function is the relative truncation error between the two

meshes. The resulting equation to be solved on any mesh G_i can be written

$$L_i(Q_i) = \tau_i \tag{3.17}$$

where $\tau_i = 0$ on the finest mesh and is the relative truncation error on each of the coarser grids. The solution of equation (3.17) is updated by introducing a time derivative of the dependent variables to the left hand side so that the solution can be advanced in time using the approximate factorization methods described in Chapter Two. The resulting scheme written on mesh G_i is given by

$$M\Delta q_i^c = -\Delta t(L_i(q_i^c) - \tau_i) = -\Delta t R_i \tag{3.18}$$

where M is the implicit operator of the scheme considered and $L_i(q_i^c)$ on the right hand side is due to the linearization of $L_i(Q_i)$ from the backward Euler time integration. Note that even on the coarse meshes where τ_i is non-zero, equation (3.18) maintains the same form as the equation on the fine mesh. The result of this is that the coarse meshes can be updated using the same scheme as on the fine mesh with only a slight modification to the right hand side.

There are several strategies for deciding when to switch from one grid level to another, generally falling under the categories of fixed or adaptive cycling algorithms. The strategy used in the present study is a fixed cycling strategy in which each global cycle consists of a set number of FAS cycles on each of the coarser grids. Recall that one FAS cycle consist of a smoothing step followed by coarse grid correction.

In addition, a predetermined number of iterations is performed on each grid level to smooth the errors before applying the coarse grid correction.

The conserved variables are transferred to the next coarser grids each time by the rule

$$Q_{i-1} = I_i^{i-1} Q_i \tag{3.19}$$

where I_i^{i-1} is a volume weighted restriction operator which transfers values on the fine grid to the coarser one and is defined by

$$I_i^{i-1} Q_i = \sum VQ / \sum V \tag{3.20}$$

and the summations are taken over all the fine grid cells which make up the coarse grid cell. As shown in Appendix C, restriction of the dependent variables in this manner conserves mass, momentum, and energy in each of the cell volumes. The relative truncation error is calculated on the coarse grid as

$$\tau_{i-1} = L_{i-1}(I_i^{i-1} q_i^c) - \hat{I}_i^{i-1} R_i \tag{3.21}$$

where $\hat{I}_i^{i-1}$ is the restriction operator for the residual defined as

$$\hat{I}_i^{i-1} R_i = \sum R_i \tag{3.22}$$

where again, the summation is over the cells on the fine grid which make

up the coarse grid. By summing the residuals, the surface integral of the fluxes crossing the cell boundaries on the coarse grid are the same as would arise by integrating around all the fine grid cells making up the coarse grid (see Appendix C). On this grid, several iterations of the approximate factorization scheme can be conducted to get an approximation to the steady solution on G_{i-1} with the right hand side modified to include the relative truncation error. If only one coarse grid is used to correct the finest grid, the result is the simple FAS two-level cycle. On the other hand, if more grid levels are introduced so that one or more FAS cycles can be recursively carried out on each subsequent coarse grid level to get a better approximation to Q_{N-1}, then a multilevel algorithm results. When only one FAS cycle is carried out for each of the coarser grids, the resulting global cycling stragegy is termed a V-cycle and is depicted in figure 7. Another cycle of interest, which is shown in figure 8, is termed a W-cycle and results when two FAS cycles are used on each of the coarser meshes. Results will be shown in the next section using both types of cycles where on the coarsest mesh, three smoothing iterations are performed in lieu of solving the equations on the coarsest mesh exactly. The corrections on coarse meshes are passed to the next finer mesh using trilinear interpolation with no additional iteration steps between meshes. When a W-cycle is used however, note that an iteration is carried out at the beginning of each FAS cycle correction in order to smooth the high frequencies.

In order to further clarify the multigrid procedure, the overall process is summarized below for an exemplary case where three grid

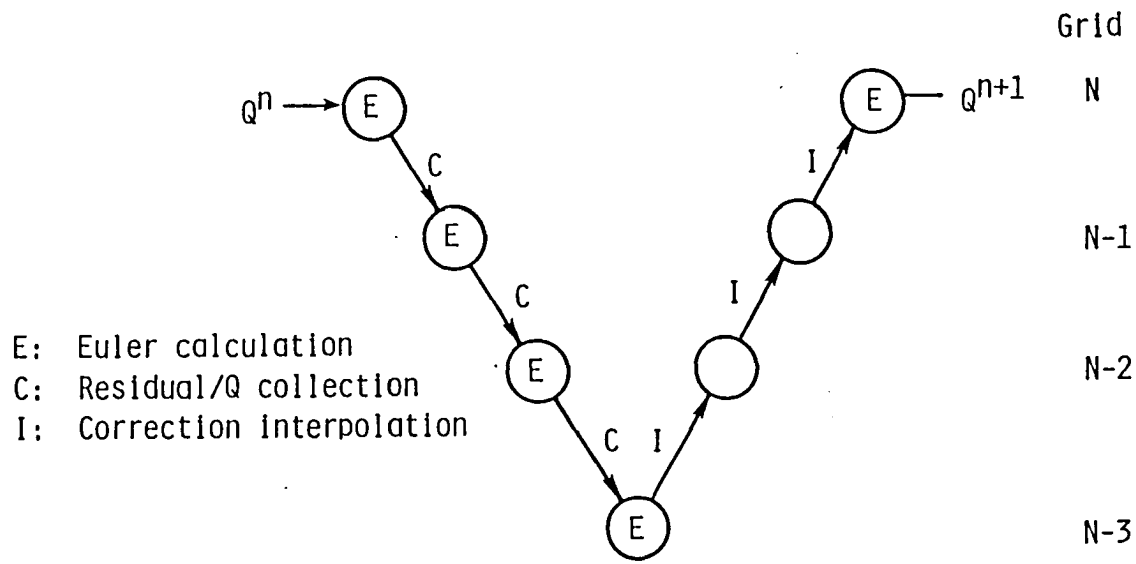


Figure 7. Multigrid V-cycle

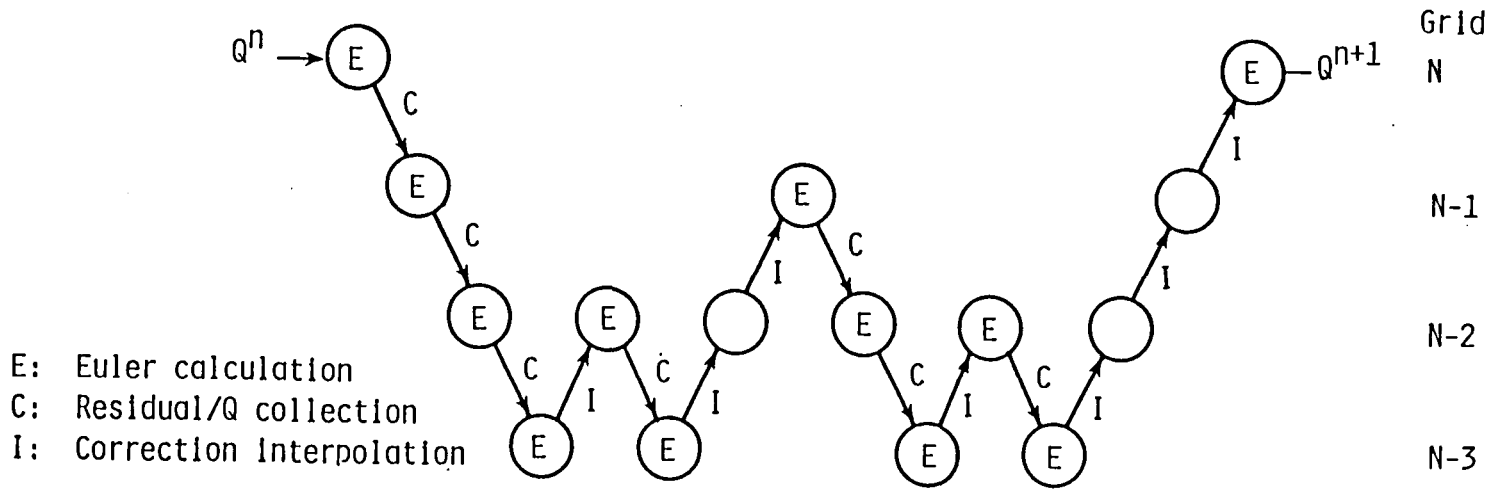


Figure 8. Multigrid W-cycle

levels are used in a V-cycle.

1. Starting on the finest grid, smooth the errors by doing one iteration of equation (3.18) with $\tau_i = 0$.
2. Calculate the residual on the fine grid from equation (3.3) where $L_N(q_N{}^c)$ is given by equation (3.16).
3. Restrict the dependent variables to the first coarse grid, G_{N-1}, using equation (3.19).
4. Restrict the residual from the finest grid to G_{N-1} using equation (3.22) and calculate the relative truncation error using (3.21).
5. Calculate the right hand side of equation (3.18) and update the solution on mesh G_{N-1}. (This serves to smooth the errors on this grid so that a coarser grid can be introduced.)
6. Calculate the residual on this mesh using equation (3.8). Note that this can be written as

   $$R_{N-1} = L_{N-1}(q^c_{N-1}) - \tau_{N-1} \tag{3.23}$$

   Since τ_{N-1} has been previously calculated, the residual is easily calculated by simply calculating $L_{N-1}(q^c_{N-1})$ from the most current values of the dependent variables on the mesh and subtracting τ_{N-1}.
7. Restrict the dependent variables on G_{N-1} to G_{N-2} using equation (3.19).
8. Restrict the residual from equation (3.23) to the N-2 grid and calculate τ_{N-2} from equation (3.21).
9. Calculate the right hand side of equation (3.18) and update the solution on this mesh. Since this is the coarsest mesh used in the present example, three iterations of equation (3.18) are done to

get an approximation to Q_{N-2}. Each step, the right hand side is updated to use the most current values of the dependent variables in $L_{N-2}(q^c_{N-2})$. Note that τ_{N-2} will not change.

10. Calculate the correction on this mesh

$$V_{N-2} = q^c_{N-2} - I^{N-2}_{N-1}\, q^c_{N-1} \tag{3.24}$$

11. Pass the correction to the next finest mesh using trilinear interpolation and update the solution

$$q^c_{N-1} \leftarrow q^c_{N-1} + I^{N-1}_{N-2}\, V_{N-2} \tag{3.25}$$

Note that steps 5-11 make up one FAS cycle on grid N-1 where steps 6-11 constitute a coarse grid correction. At this point, if a W-cycle were being employed, another FAS cycle (steps 5-11) would be done to further update q^c_{N-1}.

12 Calculate the correction on the N-1 mesh as

$$V_{N-1} = q^c_{N-1} - I^{N-1}_{N}\, q_N{}^c$$

13. Pass this correction to the finest mesh and update the solution

$$q_N{}^c \leftarrow q_N{}^c + I^{N}_{N-1}\, V_{N-1}$$

14. Do one smoothing iteration using equation (3.18) to smooth the errors.

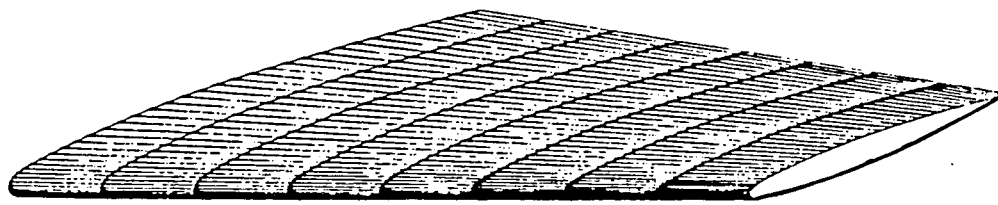
Chapter IV

RESULTS

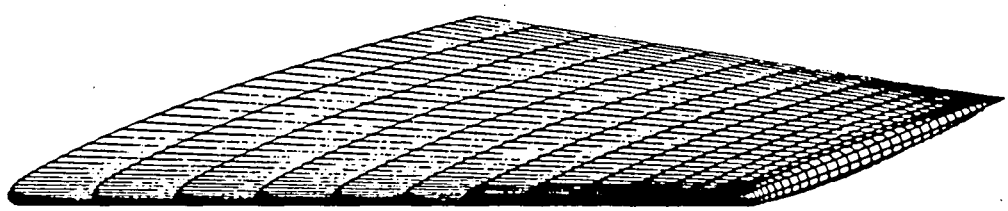
4.1 ONERA M6

Three-dimensional subsonic and transonic flow computations over the ONERA M6 wing are shown below. Comparisons are made with experimental data at a Reynolds number of 11.7 million,[31] corresponding to conditions for which viscous effects are relatively small. The wing consists of symmetrical airfoil sections with a planform swept thirty degrees along the leading edge, an aspect ratio of 3.8, and a taper ratio of 0.56. Solutions are obtained for two mesh types, C-H and C-O, both of which are C-type mesh topologies around the airfoil profile. The C-H mesh, has uniform spacing in the spanwise direction whereas the C-O mesh wraps around the wing tip, consequently leading to a more precise definition of the actual rounded tip geometry tested in the experiment. The C-O mesh has been generated with a trans-finite interpolationn procedure developed by Bruce Wedan of NASA Langley Research Center. The C-H mesh was obtained by simply stacking a series of two dimensional cross sections along the span. The surface mesh for both are shown in figure 9.

The first computation is the ONERA M6 wing at transonic conditions: Mach number of 0.84 and an angle of attack of 3.06 degrees. Figure 10 shows the effect of multigrid on the residual and lift history for a 193 x 33 x 33 C-H mesh, corresponding to 193 points along the airfoil and wake, 33 points approximately normal to the airfoil, and 33 points in the spanwise direction, 17 of which are on the wing planform. For this case, the Van Leer splittings are used with a



(a) 97 x 17 x 17 C-H mesh



(b) 97 x 17 x 17 C-O mesh

Figure 9. Surface mesh for the ONERA M6 wing

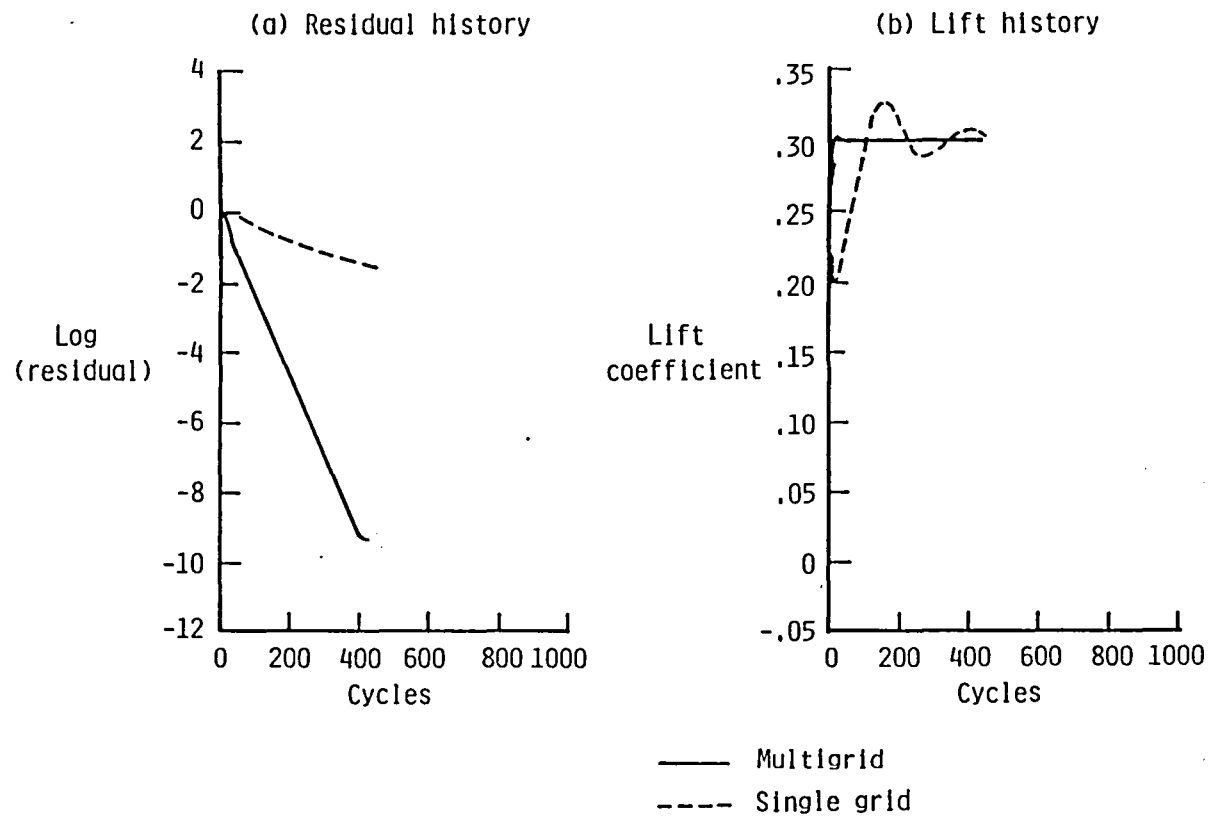


Figure 10. Effect of multigrid on convergence; ONERA M6; 193 x 33 x 33 C-H mesh; $M_\infty = 0.84$; $\alpha = 3.06°$.

V-cycle and four grid levels (a fine grid and three coarser ones). The multigrid scheme is very effective in accelerating convergence of both the residual and the lift. The residual is reduced to machine zero in 400 cycles while the single grid scheme has only reduced the residual between one and two orders of magnitude. The benefit of multigrid is especially pronounced in the lift history where the final lift value is obtained to within 0.1 percent of its final value in 41 cycles. This is a dramatic improvement over the single grid result which required more than 400 iterations to settle in on a final lift coefficient. It should be noted that for all the cases considered, several cycles (usually 5) were run with first order spatial differencing before switching to second order.

A comparison of convergence rates between the three schemes discussed in Chapter 2 is shown in figure 11 for identical conditions as given above with the exception that only every other point from the 193 x 33 x 33 mesh is used, resulting in a 97 x 17 x 17 C-H mesh. For this size mesh, only two coarser grids are used. The 3-factor, spatially-split algorithm demonstrates a higher rate of convergence than either of the two factor schemes yielding a spectral radius of approximately 0.898. The 2-factor scheme in which the implicit operator is split according to the sign of the eigenvalues displays the slowest convergence rate with a spectral radius of 0.93. It should be pointed out however that even though the spectral radius using this scheme is not as good as for the spatially split scheme, this still represents a good improvement over a corresponding single grid spectral radius of 0.98. All the runs on the 97 x 17 x 17 meshes were made at a CFL

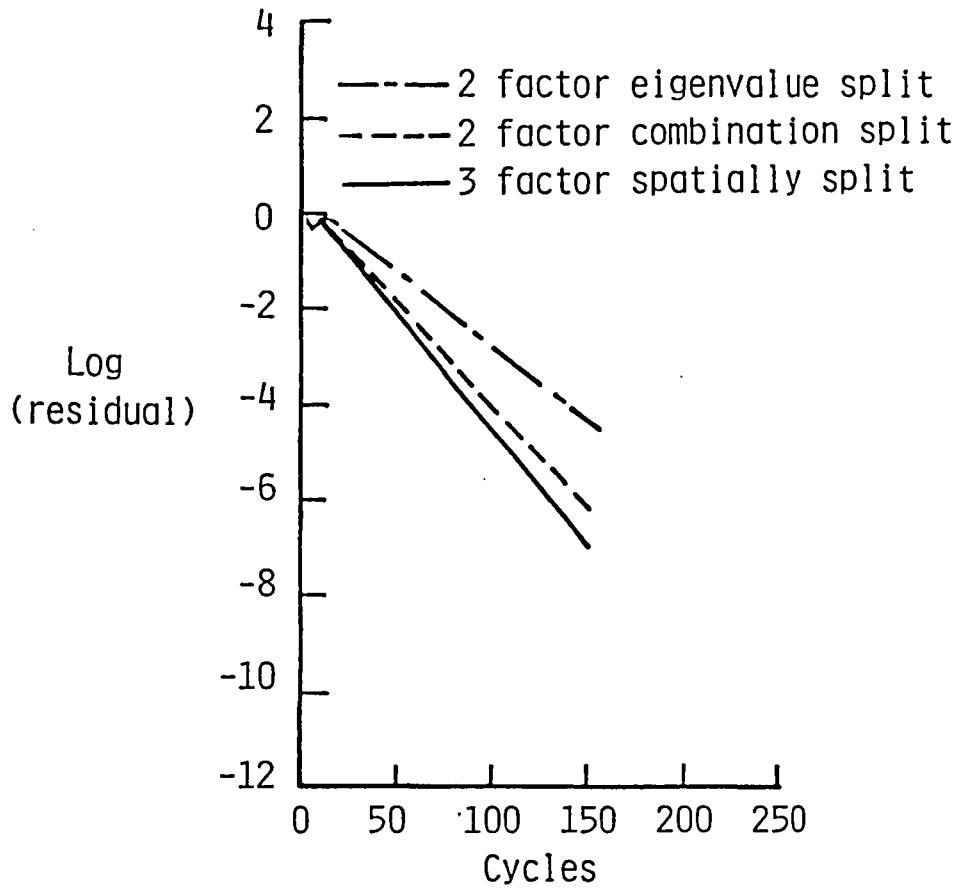


Figure 11. Comparison of convergence rate for the 3 schemes; ONERA M6 wing; 97 x 17 x 17 C-H mesh; $M_\infty = 0.84$; $\alpha = 3.06°$.

number of 7. This was determined experimentally to be about optimum and agrees well with the CFL number for best smoothing predicted by the stability analysis. In 64 bit precision, the computational rate using a V-cycle and three grid levels for the 3-factor scheme is about 75 microseconds per grid point per cycle whereas the 2-factor eigenvalue split and combination split schemes exhibit computational rates of 140 and 85 microseconds per grid point per cycle, respectively. It should be noted that the computational rate is decreased by approximately 40 percent when the computations are done in 32 bit precision. Due to the higher performance of the three-factor spatially split algorithm in both the convergence rate and the computational rate, it is used exclusively in the results that follow.

The effect of using a W-cycle over the previously used V-cycle on the residual is shown in figure 12 for the 97x17x17 C-H mesh. An improvement using a W-cycle in the convergence rate is apparent. In addition, the lift coefficient is obtained to within 0.3 percent of its final value in only 14 cycles and to under 0.1 percent in 24 cycles. This is an improvement over the V-cycle which took 37 cycles to get the error in lift below 0.1 percent. Although the work involved for a W-cycle is more than for the V-cycle due to the extra smoothing iterations on the coarser grids, the time required per cycle only increased by about 13 percent over a V-cycle. Therefore, even though more work is involved for each cycle, a net gain is still achieved by employing the W-cycle. A summary of results for this case is given in table 2 for 193 x 33 x 33 and 97 x 17 x 17 for both C-H and C-O type grids. The table includes the spectral radius based on cycles and the number of cycles

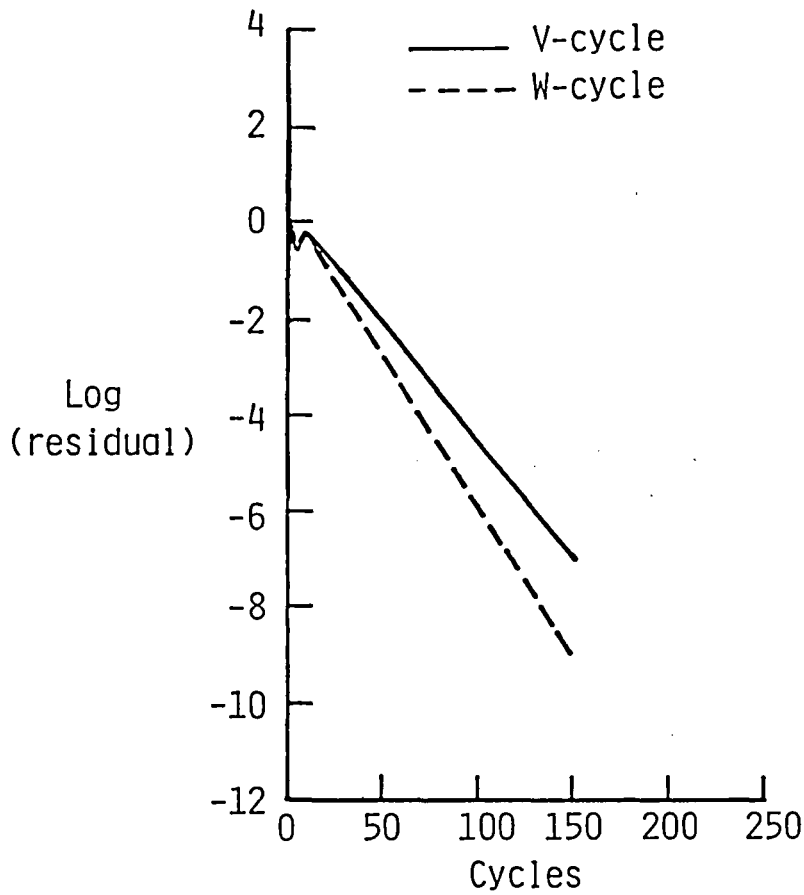


Figure 12. Comparison of convergence rate for a V-cycle and a W-cycle; ONERA M6 wing; 97 x 17 x 17 C-H mesh; $M_\infty = 0.84$; $\alpha = 3.06°$.

Mesh size and type of cycle	Cycles required to obtain			Spectral radius
	C_ℓ to 0.3% of final value	C_ℓ to 0.1% of final value	C_ℓ to 5 decimal places	
97 x 17 x 17 C-H				
V-cycle	20	37	75	0.898
W-cycle	14	24	42	0.871
97 x 17 x 17 C-O				
V-cycle	34	45	91	0.912
W-cycle	15	27	44	0.879
193 x 33 x 33 C-H				
V-cycle	37	41	153	0.948
W-cycle	12	23	47	0.923
193 x 33 x 33 C-O				
V-cycle	27	68	149	0.952
W-cycle	14	19	47	0.926

Table 2 Summary of results for ONERA M6 wing
$M_\infty = 0.84$; $\alpha = 3.06°$.

required to obtain the lift coefficient to within 0.3 and 0.1 percent of its final value as well as how many cycles were required to obtain the lift to five significant digits. Note that the number of cycles required for the W-cycle to obtain the lift coefficient is relatively insensitive to the number of grid points.

Figure 13 shows the upper surface pressure distributions on the 193 x 33 x 33 C-H mesh as well as the 193 x 33 x 33 C-O mesh. The wing under these conditions exhibits both a swept shock emanating from the apex and a nearly normal shock emanating from the root, which coalesce at about 80 percent of the span to form a single shock. Figure 14 shows upper surface pressure contours.

In figure 15, pressure coefficients obtained using the Van Leer splitting on both the 97 x 17 x 17 and 193 x 33 x 33 C-O meshes are compared with experimental data at six spanwise locations. The computations are obtained at the same spanwise locations as the experimental data by linear interpolation. The computations on both meshes agree reasonably well with experiment for each spanwise location; the effect of the finer mesh is to resolve the leading edge suction pressures and shock positions, and improve the agreement with experiment at the most outboard station. Results obtained with the Steger-Warming splitting on the same two meshes are compared with experimental data in figure 16. The computations are nearly identical to the previous ones with small differences occurring near the shock regions.

The next three-dimensional test case is the ONERA M6 wing at a freestream Mach number of 0.699 and an angle of attack of 3.06 degrees. At these conditions, the flow remains subsonic over the entire

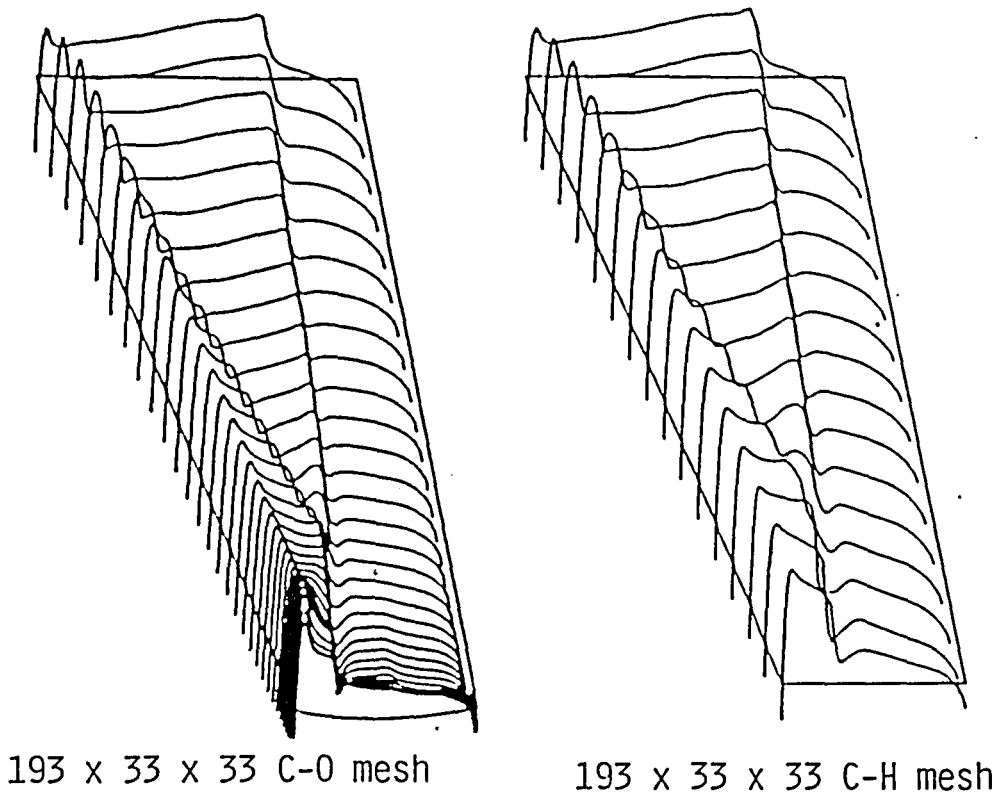


Figure 13. Upper surface variation of pressure coefficient; ONERA M6 wing; $M_\infty = 0.84$; $\alpha = 3.06°$.

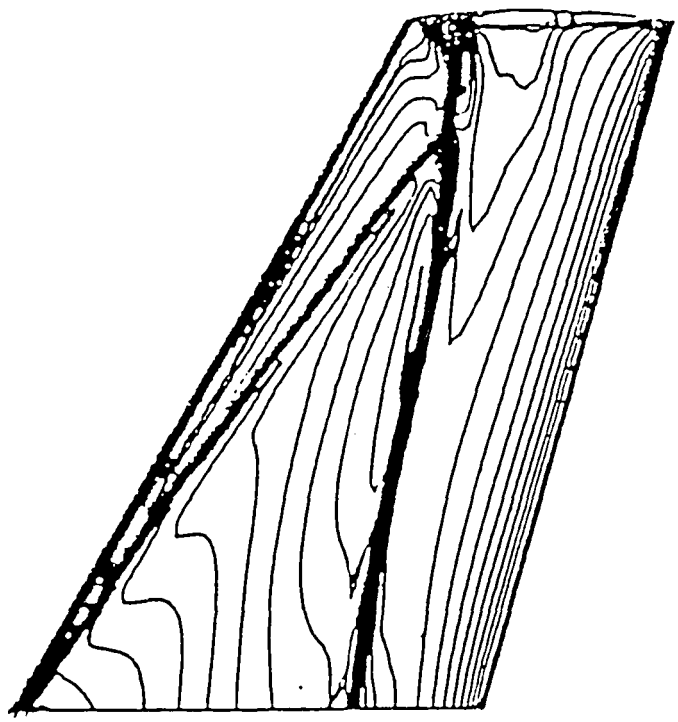


Figure 14. Pressure coefficient contours on upper surface; ONERA M6 wing: $M_\infty = 0.84$; $\alpha = 3.06°$.

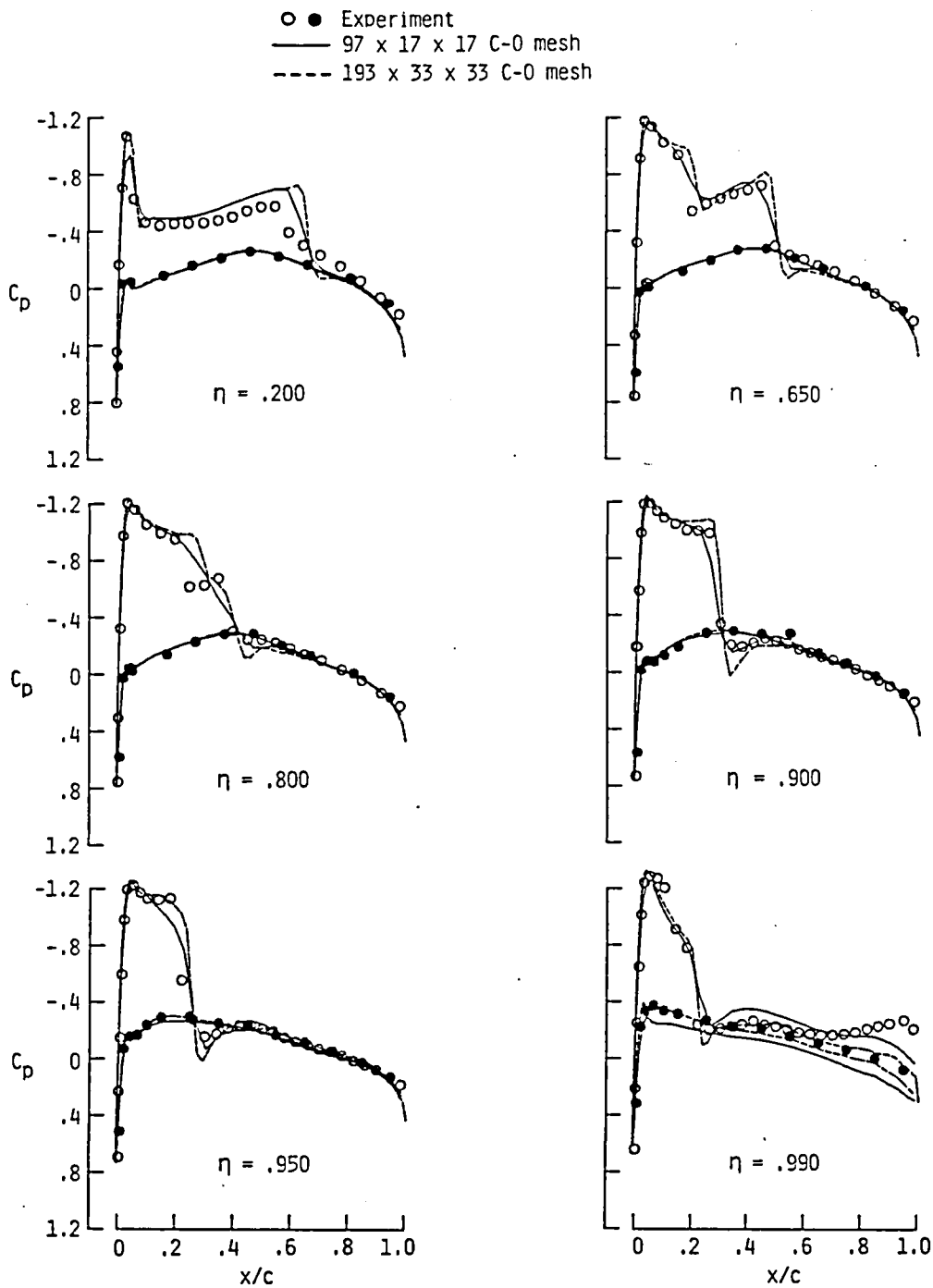


Figure 15. Comparison of experiment and inviscid calculations using Van Leer splittings; ONERA M6 wing; $M_\infty = 0.84$; $\alpha = 3.06°$.

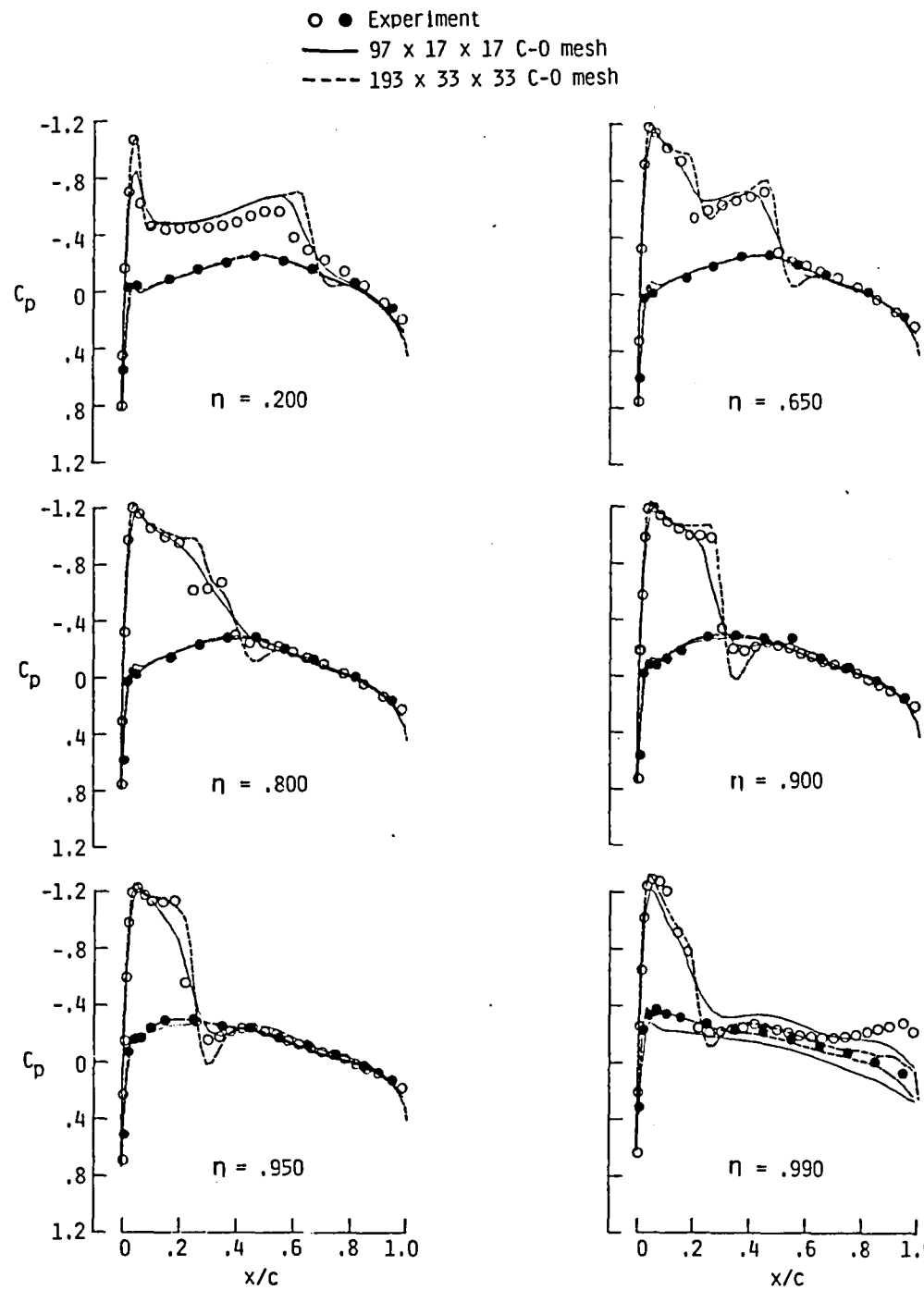


Figure 16. Comparison of experiment and inviscid calculations using Steger-Warming splittings; ONERA M6 wing; $M_\infty = 0.84$; $\alpha = 3.06°$.

wing. Results were obtained for this case on a 97 x 17 x 17 C-O mesh, a 97 x 17 x 17 C-H mesh, and both a 193 x 33 x 33 C-H mesh and C-O mesh. Figure 17 shows the residual history for both the 97 x 17 x 17 C-O and C-H meshes using the Van Leer splittings and a V-cycle. The convergence rate on the C-H mesh is slightly better than on the C-O mesh. Machine zero is reached for the C-H mesh in approximately 200 cycles while the C-O mesh requires about 300 cycles, corresponding to an asymptotic spectral radius of .891 and .926, respectively. For both meshes, the lift was obtained to less than 0.1 percent of its final value in less than 28 cycles requiring only about 46 seconds of computer time. On the 193 x 33 x 33 C-H mesh, a spectral radius of 0.929 was obtained with the multigrid algorithm while a spectral radius of 0.95 was obtained on the same sized C-O mesh. When using a W-cycle, a spectral radius of 0.866 is obtained for the 97 x 17 x 17 C-H mesh and one of 0.891 is obtained for the C-O mesh. Using the 193 x 33 x 33 mesh, the spectral radius using the W-cycle is also about 0.89 for the C-H mesh and 0.912 for the C-O mesh. A summary of results is given in table 3 similar to those shown in table 2.

The pressure distributions on the 97 x 17 x 17 C-O mesh and the 193 x 33 x 33 C-H mesh are compared with experiment at six spanwise stations in figure 18 using the Van Leer splitting and in figure 19 using the Steger-Warming splittings. At the inboard stations, the results for both meshes are essentially identical and compare well with experiment. At the outboard station, however, the pressures computed on the C-O mesh agree much closer to experiment due to the increased resolution at the tip.

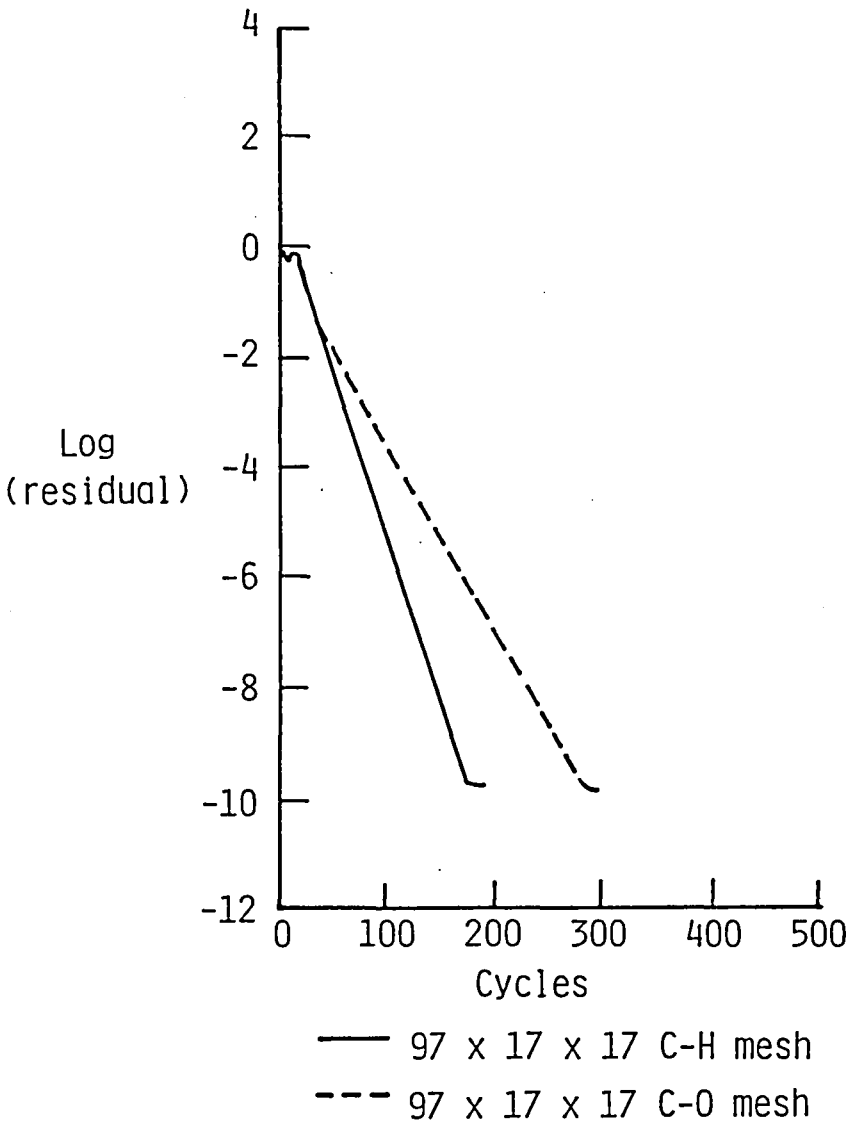


Figure 17. Multigrid convergence for C-H and C-O meshes; ONERA M6 wing; $M_\infty = 0.699$; $\alpha = 3.06°$; V-cycle.

Mesh size and type of cycle	Cycles required to obtain			Spectral radius
	C_ℓ to 0.3% of final value	C_ℓ to 0.1% of final value	C_ℓ to 5 decimal places	
97 x 17 x 17 C-H				
V-cycle	19	28	38	0.891
W-cycle	11	19	33	0.866
97 x 17 x 17 C-O				
V-cycle	21	22	48	0.926
W-cycle	13	15	31	0.891
193 x 33 x 33 C-H				
V-cycle	29	37	71	0.929
W-cycle	11	21	37	0.891
193 x 33 x 33 C-O				
V-cycle	21	38	55	0.950
W-cycle	11	15	36	0.912

Table 3 Summary of results for ONERA M6 wing
$M_\infty = 0.699$; $\alpha = 3.06°$.

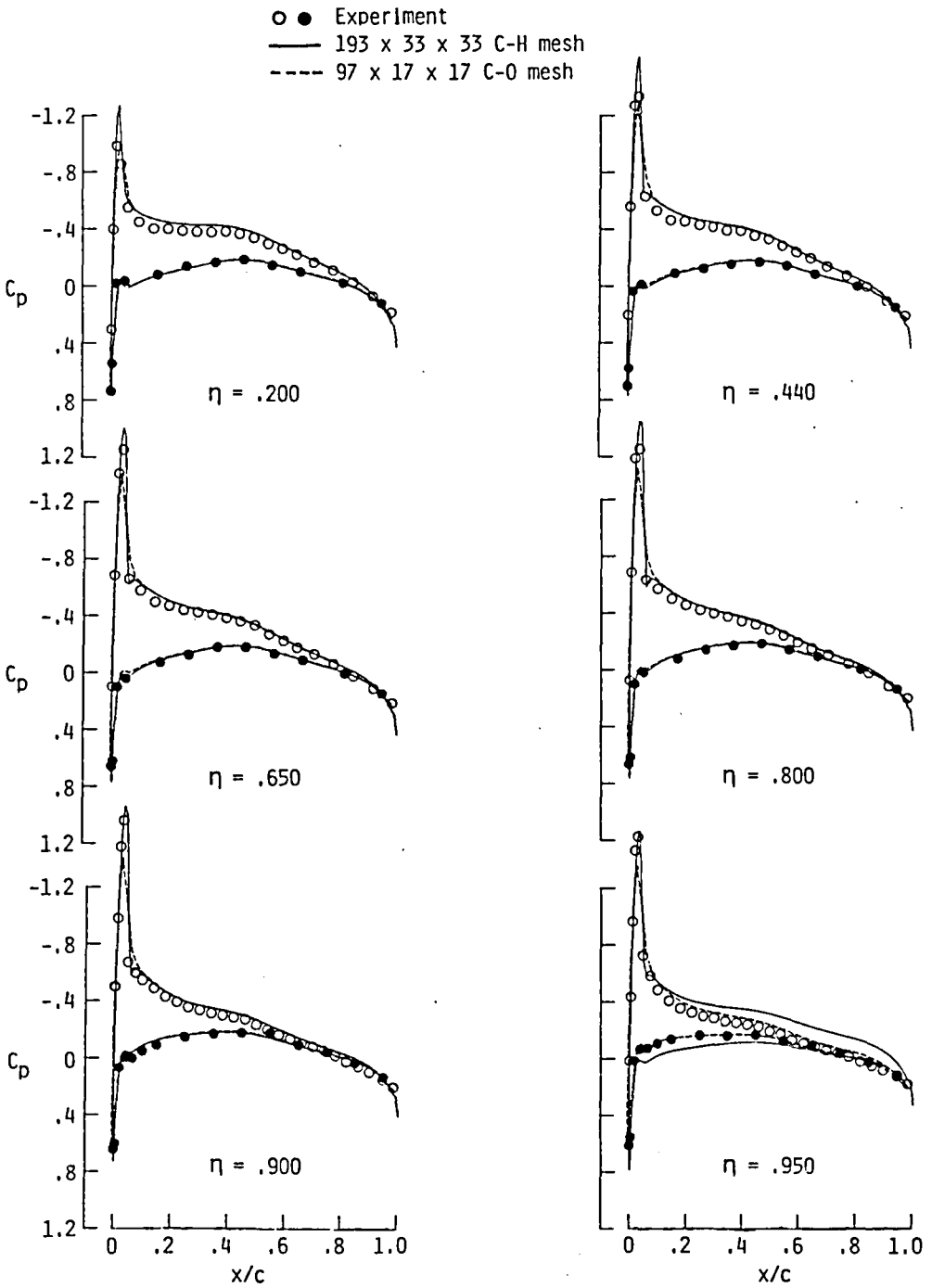


Figure 18. Comparison of experiment and inviscid calculations using Van Leer splittings; ONERA M6 wing; $M_\infty = 0.699$; $\alpha = 3.06°$.

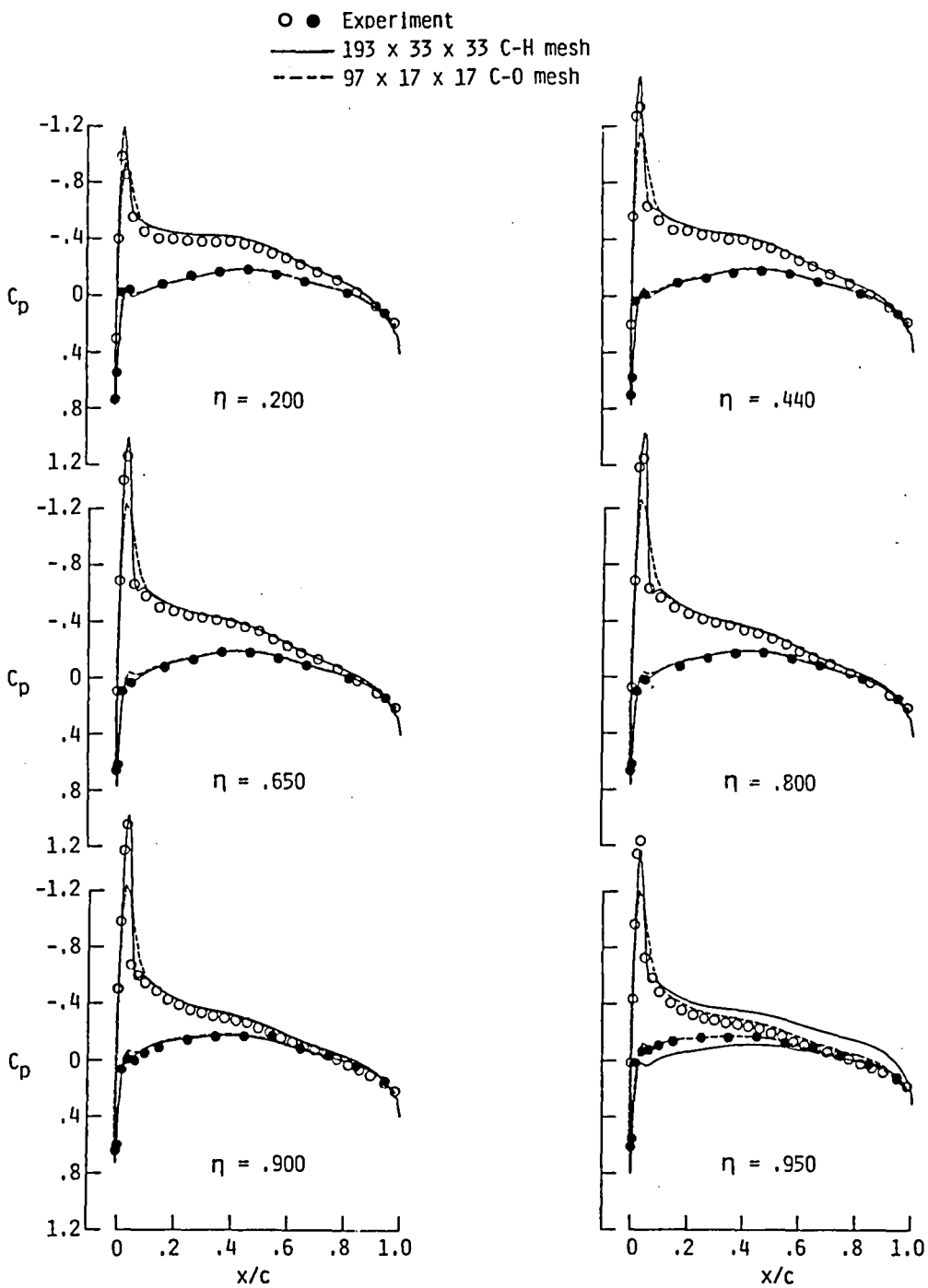


Figure 19. Comparison of experiment and inviscid calculations using Steger-Warming splittings, ONERA M6 wing; $M_\infty = 0.699$; $\alpha = 3.06°$.

4.2 Analytic Forebody

The last test case considered is an analytically defined forebody for which experimental data is available at supersonic Mach numbers.[32] The grid used, shown in figure 20 along with static density contours, was a 49 x 49 x 49 grid with a symmetry plane along the centerline. The conditions correspond to a freestream Mach number of 1.7 and an angle of attack of 0 degrees; this leads to an oblique shock at the nose and supersonic flow over the entire length of the body.

The residual and lift history obtained using a V-cycle and the Van Leer splittings are shown in figure 21. As can be seen, the residual is reduced 3 orders of magnitude in only 50 cycles (10 of which were first order accurate) and an asymptotic spectral radius of 0.83 is achieved based on the last thirty cycles. The lift is obtained to 0.3 percent error of the final value in only 22 cycles. The pressure distribution compared with experimental data at the forebody symmetry plane is shown in figure 22 over both the leeward and windward sides of the body. As seen, the pressure coefficients compare well with the experimental data over both the lower and upper surfaces of the body.

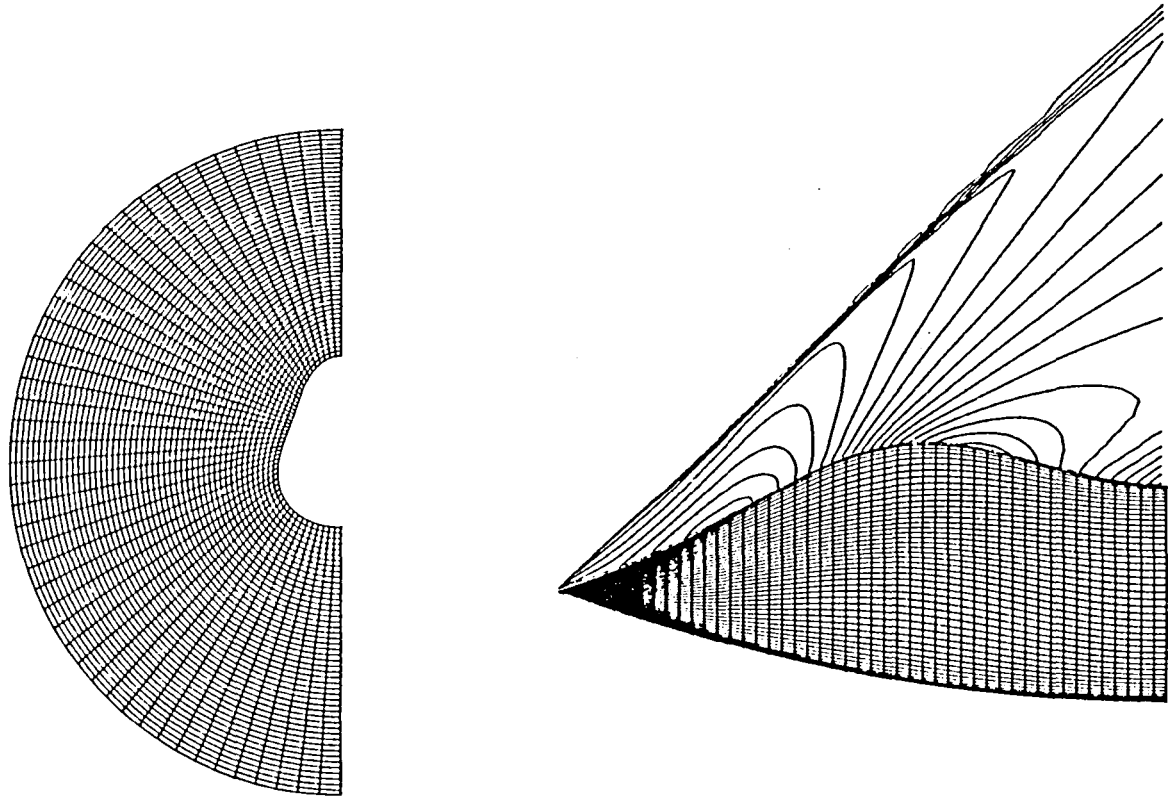


Figure 20. Grid for analytic forebody.

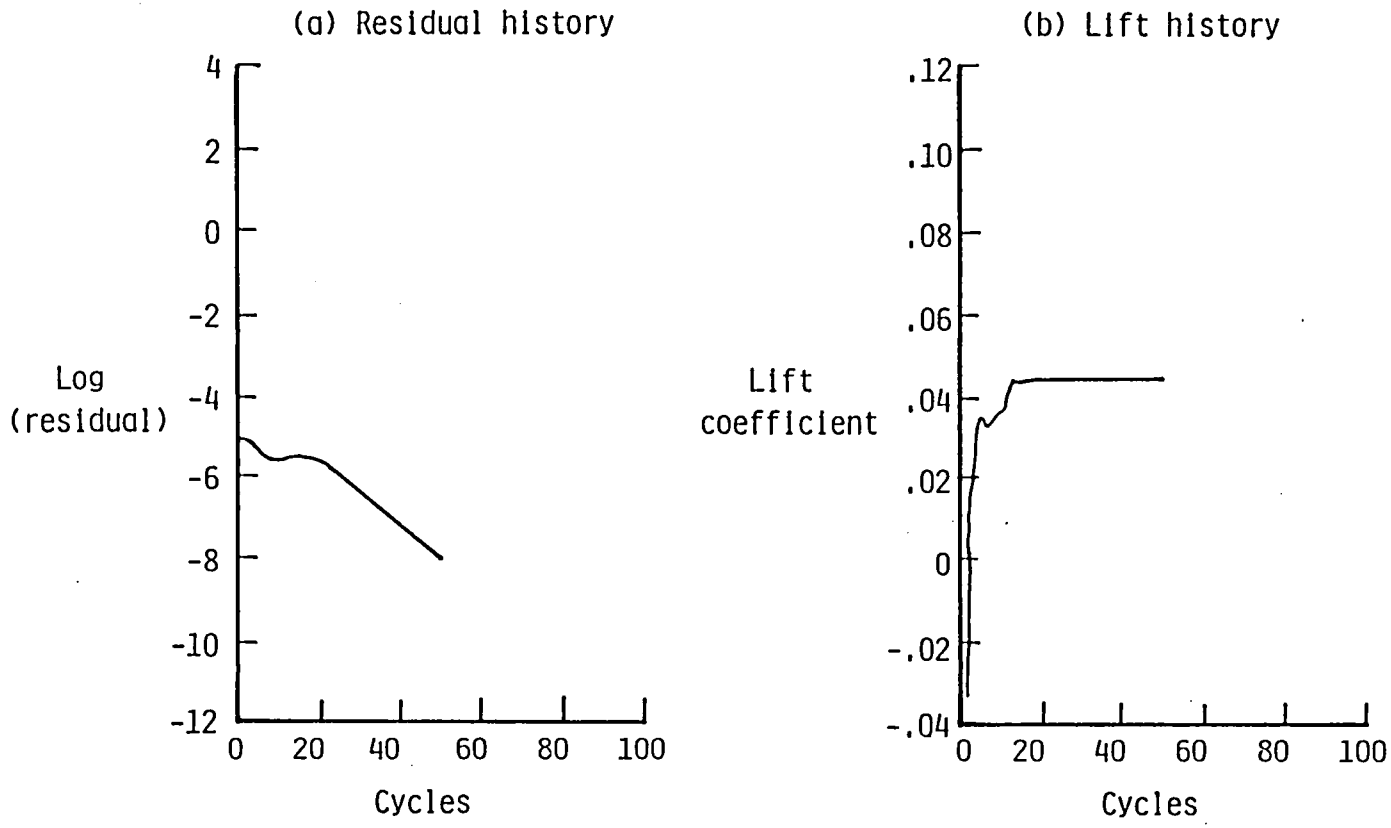


Figure 21. Convergence history for analytic forebody; $M_\infty = 1.7$; $\alpha = 0°$.

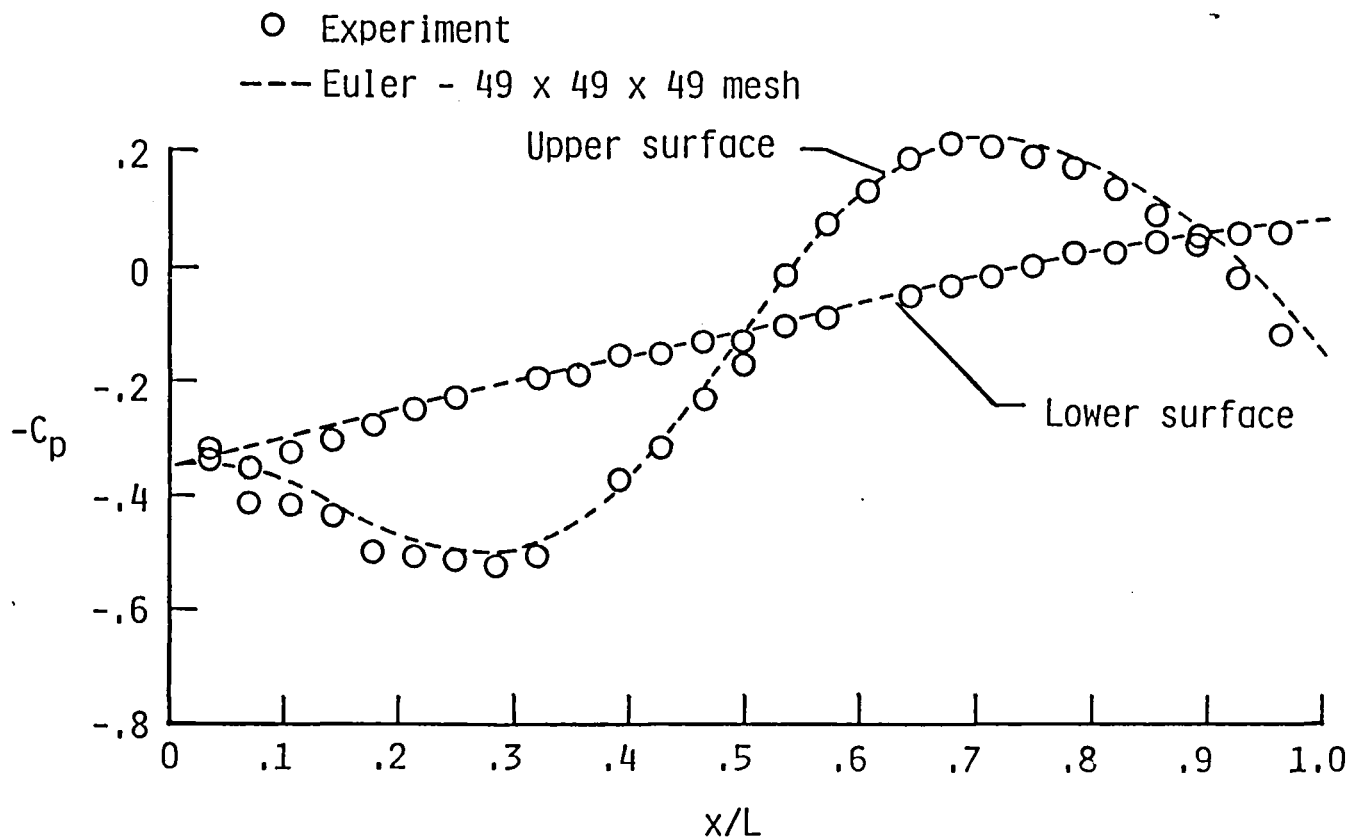


Figure 22. Calculated pressure coefficient comparison with experiment at forebody symmetry plane; $M_\infty = 1.7$; $\alpha = 0°$.

CHAPTER V

CONCLUSIONS

Multigrid acceleration has been applied to the three-dimensional flux-split Euler equations in generalized coordinates. Three implicit schemes have been used to smooth the errors at each grid level. Results from a linearized stability analysis of the coupled equations for each of the schemes agree well in overall trends with the numerical experiments and indicate that the 3-factor spatially-split algorithm is conditionally stable (up to a CFL of about 20) but offers a slightly better smoothing rate than the other two schemes and hence, the best multigrid performance. The stability analysis also indicates that the other two schemes, both two-factor schemes, are less sensitive to CFL variations. Results obtained for subsonic and transonic flow over the ONERA M6 wing and supersonic flow over an analytically defined forebody are compared with experimental data for cases with weak viscous effects. In the wing calculations, two methods of splitting the flux vectors were compared with experiment; the splitting of Steger-Warming and Van Leer. For both the subsonic and transonic cases, both methods of splitting the flux vector agree well with experiment and with each other.

Results obtained for a series of subsonic, transonic, and supersonic flows demonstrate a substantial improvement in convergence rate using the multigrid algorithm in comparison to the single grid algorithm. Using a W-cycle, solutions can be obtained in as few as 19 cycles for transonic conditions on a 193 x 33 x 33 mesh whereas a V-cycle takes about 41 cycles to reach the same level of accuracy (final

lift coefficient to within 0.1 percent). Using a V-cycle, a spectral radius of 0.891 and 0.898 is obtained for a 97 x 17 x 17 wing solution at subsonic and transonic conditions. A W-cycle for the same cases results in spectra radii of 0.866 for the subsonic case and 0.871 for the transonic case. In addition, the W-cycle is less sensitive to the size of the grid than the V-cycle for obtaining the final lift coefficient. Both of these save an order of magnitude in computing time over a single grid.

APPENDIX A

TRANSFORMATION TO GENERALIZED COORDINATES

The three dimensional Euler equations in Cartesian coordinates and strong conservation law form are given by

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} + \frac{\partial H}{\partial z} = 0 \tag{A.1}$$

where

$$Q = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{Bmatrix} \quad F = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (e+p)u \end{Bmatrix} \quad G = \begin{Bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (e+p)v \end{Bmatrix} \quad H = \begin{Bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (e+p)w \end{Bmatrix} \tag{A.2}$$

and $p = (\gamma-1)(e - \frac{1}{2}\rho(u^2 + v^2 + w^2))$

Using the chain rule and the body-fitted coordinate system given by the steady transformation

$$\xi = \xi(x,y,z) \quad \eta = \eta(x,y,z) \quad \zeta = \zeta(x,y,z) \quad \tau = t \tag{A.3}$$

the Euler equations can be recast as

$$\frac{\partial Q}{\partial \tau} + \xi_x \frac{\partial F}{\partial \xi} + \eta_x \frac{\partial F}{\partial \eta} + \zeta_x \frac{\partial F}{\partial \zeta} + \xi_y \frac{\partial G}{\partial \xi} + \eta_y \frac{\partial G}{\partial \eta} + \zeta_y \frac{\partial G}{\partial \zeta} \tag{A.4}$$

$$+ \xi_z \frac{\partial H}{\partial \xi} + \eta_z \frac{\partial H}{\partial \eta} + \zeta_z \frac{\partial H}{\partial \zeta} = 0$$

Now, again using the chain rule, the derivatives with respect to τ, ξ, η, and ζ can be written in matrix form as

$$\begin{pmatrix} \frac{\partial}{\partial\tau} \\ \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\zeta} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & x_\xi & y_\xi & z_\xi \\ 0 & x_\eta & y_\eta & z_\eta \\ 0 & x_\zeta & y_\zeta & z_\zeta \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial t} \\ \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} \tag{A.5}$$

from which Cramer's rule can be used to solve for the x,y,z and t derivatives. Using these to evaluate the metric terms gives

$$\xi_x = J(y_\eta z_\zeta - z_\eta y_\zeta) \quad \eta_x = J(z_\xi y_\zeta - y_\xi z_\zeta) \quad \zeta_x = J(y_\xi z_\eta - z_\xi y_\eta)$$

$$\xi_y = J(z_\eta x_\zeta - x_\eta z_\zeta) \quad \eta_y = J(x_\xi z_\zeta - z_\xi x_\zeta) \quad \zeta_y = J(z_\xi x_\eta - x_\xi z_\eta)$$

$$\xi_z = J(x_\eta y_\zeta - y_\eta x_\zeta) \quad \eta_z = J(X_\zeta Y_\xi - Y_\zeta X_\xi) \quad \zeta_z = J(x_\xi y_\eta - y_\xi x_\eta) \tag{A.6}$$

where $J^{-1} = x_\xi(y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi(x_\eta z_\zeta - z_\eta x_\zeta) + z_\xi(x_\eta y_\zeta - y_\eta x_\zeta)$ (A.7)

In order to regain the strong conservation law form, equation (A.4) can now be multiplied through by J^{-1} and rearranged using the chain rule on certain terms. For example, the term $J^{-1}\xi_x F_\xi$ can be rewritten as

$$J^{-1}\xi_x F_\xi = \frac{\partial}{\partial\xi}[(J^{-1}\xi_x)F)] - F(\frac{\partial}{\partial\xi}(J^{-1}\xi_x)) \tag{A.8}$$

After rewriting all the appropriate terms and noting that many parts of the resulting equation can be shown to be zero by substituting equations (A.6) for the metrics, the Euler equations can be written in generalized coordinates maintaining the strong conservation law form as

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = 0 \tag{A.9}$$

where $\hat{Q} = J^{-1}Q$

$$\hat{F} = J^{-1}(\xi_x F + \xi_y G + \xi_z H)$$

$$\hat{G} = J^{-1}(\eta_x F + \eta_y G + \eta_z H)$$

$$\hat{H} = J^{-1}(\zeta_x F + \zeta_y G + \zeta_z H) \tag{A.10}$$

Using equations (A.2) in (A.10), the flux vectors can be further written in an alternate form as

$$\hat{F} = \frac{1}{J}\begin{Bmatrix} \rho U \\ \rho uU + \xi_x p \\ \rho vU + \xi_y p \\ \rho wU + \xi_z p \\ (e + p)U \end{Bmatrix} \quad \hat{G} = \frac{1}{J}\begin{Bmatrix} \rho V \\ \rho uV + \eta_x p \\ \rho vV + \eta_y p \\ \rho wV + \eta_z p \\ (e + p)V \end{Bmatrix} \quad \hat{H} = \frac{1}{J}\begin{Bmatrix} \rho W \\ \rho uW + \zeta_x p \\ \rho vW + \zeta_y p \\ \rho wW + \zeta_z p \\ (e + p)W \end{Bmatrix} \tag{A.11}$$

where U, V, and W are the contravariant velocities defined as

$$U = \xi_x u + \xi_y v + \xi_z w$$

$$V = \eta_x u + \eta_y v + \eta_z w$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w \qquad \text{(A.12)}$$

APPENDIX B

SPLITTING THE FLUX VECTORS IN GENERALIZED COORDINATES

The Van Leer method of splitting the flux vectors was originally given only for a Cartesian coordinate system.[5] For example, the split flux vectors in the x-direction was given in terms of the local one-dimensional Mach number $M_x = u/a$. For supersonic flow, i.e., $|M_x| \geq 1$, we have

$$F^+ = F, \quad F^- = 0, \quad M_x > 1$$

$$F^+ = 0, \quad F^- = F, \quad M_x < -1 \tag{B.1}$$

and for subsonic flow, $|M_x| < 1$

$$F^{\pm} = \begin{Bmatrix} \pm\rho a\{\frac{1}{2}(M_x \pm 1)\}^2 = f^{\pm}_{mass} \\ f^{\pm}_{mass}\{(\gamma-1)u \pm 2a\}/\gamma \\ f^{\pm}_{mass} v \\ f^{\pm}_{mass} w \\ f^{\pm}_{mass}[\{(\gamma-1)u \pm 2a\}^2/\{2(\gamma^2-1)\} + (v^2 + w^2)/2] \end{Bmatrix} \tag{B.2}$$

For many applications, however, it is advantageous to construct generalized (body-fitted) coordinate systems of the type

$$\xi = \xi(x,y,z) \quad \eta = \eta(x,y,z) \quad \zeta = \zeta(x,y,z) \quad \tau = t \tag{B.3}$$

where, in the present work, the transformation is chosen so that the grid spacing in the computational domain is uniform and of unit length. In the discussion that follows, the superscript ^ indicates variables in generalized coordinates while superscript ¯ indicates variables in a locally Cartesian system. If no superscript is used, Cartesian coordinates are assumed. The strong conservation form of the Euler equations in generalized coordinates is given by

$$\frac{\partial \hat{Q}}{\partial \tau} + \frac{\partial \hat{F}}{\partial \xi} + \frac{\partial \hat{G}}{\partial \eta} + \frac{\partial \hat{H}}{\partial \zeta} = 0 \tag{B.4}$$

For the purpose of determining a generalized splitting for $\hat{F}$, only the derivatives in the ξ and t directions are considered while the η and ζ derivatives are treated as source terms. For determining the splitting of $\hat{F}$, equation (B.4) is transformed by a local rotation matrix in order to decompose the flux vector $\hat{F}$ into components normal and tangential to a ξ=constant cell face. The rotation matrix is given by

$$T = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \hat{\xi}_x & \hat{\xi}_y & \hat{\xi}_z & 0 \\ 0 & \hat{t}_x & \hat{t}_y & \hat{t}_z & 0 \\ 0 & \hat{r}_x & \hat{r}_y & \hat{r}_z & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{B.5}$$

where $\hat{\xi}_x$, $\hat{\xi}_y$, and $\hat{\xi}_z$ are components of a unit vector $\vec{N}$ normal to a ξ-constant line. The $\hat{t}_i$ and $\hat{r}_i$ are components of vectors which are

normal to $\vec{N}$ and to each other so that the three vectors form a locally Cartesian coordinate system. Note that an infinite number of vectors normal to $\vec{N}$ exists which form a locally Cartesian coordinate system. These vectors however are arbitrary and their exact specification is unnecessary. Multiplication of equation (B.4) with the matrix T then yields

$$\bar{Q}_t + \bar{F}_\xi = -T\,\hat{G}_\eta + T_t\,\hat{Q} + T_\xi\,\hat{F} - T\hat{H}_\zeta \tag{B.6}$$

where

$$\bar{Q} = T\hat{Q} = \frac{1}{J}\begin{Bmatrix} \rho \\ \rho\bar{u} \\ \rho\bar{v} \\ \rho\bar{w} \\ e \end{Bmatrix} \tag{B.7}$$

$$\bar{F} = T\hat{F} = \frac{|\text{grad } \xi|}{J}\begin{Bmatrix} \rho\bar{u} \\ \rho\bar{u}\bar{u} + p \\ \rho\bar{u}\bar{v} \\ \rho\bar{u}\bar{w} \\ (e + p)\,\bar{u} \end{Bmatrix} \tag{B.8}$$

The rotated velocity component $\bar{u}$ is the velocity normal to a line of constant ξ, representing the scaled contravariant velocity component, while $\bar{v}$ and $\bar{w}$ are normal to $\bar{u}$ and to each other

$$\bar{u} = (\hat{\xi}_x u + \hat{\xi}_y v + \hat{\xi}_z w) \tag{B.9}$$

$$\bar{v} = (\hat{t}_x\, u + \hat{t}_y v + \hat{t}_z w) \tag{B.10}$$

$$\bar{w} = (\hat{r}_x u + \hat{r}_y v + \hat{r}_z w) \tag{B.11}$$

The transformed flux $\bar{F}$ is of the same functional form as the Cartesian flux vector and thus can be split according to any splitting developed for Cartesian coordinates. Therefore, equations for both the Steger-Warming and the Van Leer splittings can be used to split the flux vector $\bar{F}$ after replacing the Cartesian velocity components u, v and w by the rotated velocity components $\bar{u}$, $\bar{v}$ and $\bar{w}$. Applying the rotation T to equation (B.4) simply allows the flux vector to be split in a one-dimensional fashion, along a coordinate axis perpendicular to the cell interface. After splitting $\bar{F}$, the appropriate splitting for $\hat{F}$ is determined by applying the inverse transformation matrix T^{-1} to equation (B.6), leading to

$$\hat{Q}_t + (\hat{F}^+ + \hat{F}^-)_\xi + \hat{G}_\eta + \hat{H}_\zeta = 0 \tag{B.12}$$

with

$$\hat{F}^\pm = T^{-1}\,\bar{F}^\pm = \frac{|\mathrm{grad}(\xi)|}{J}\begin{Bmatrix} f^\pm_{mass} \\ f^\pm_{mass}[\hat{\xi}_x(-\bar{u} \pm 2a)/\gamma + u] \\ f^\pm_{mass}[\hat{\xi}_y(-\bar{u} \pm 2a)/\gamma + v] \\ f^\pm_{mass}[\hat{\xi}_z(-\bar{u} \pm 2a)/\gamma + w] \\ f^\pm_{energy} \end{Bmatrix}$$

where

$$f^\pm_{mass} = \pm\,\rho a(M_\xi \pm 1)^2/4 \tag{B.13}$$

$$f^{\pm}_{energy} = f^{\pm}_{mass}[\{-(\gamma - 1)\bar{u}^2 \pm 2(\gamma - 1)\bar{u}a$$

$$+ 2a^2\}/(\gamma^2 - 1) + (u^2 + v^2 + w^2)/2]$$

Note that the inverse transformation restores the original form of the equations, i.e., no additional source terms arise and the form of $\hat{G}$ and $\hat{H}$ is unaffected. This allows a splitting of $\hat{G}$ and $\hat{H}$ similar to the splitting of $\hat{F}$ shown above.

Carried out with the Steger-Warming Cartesian splitting starting with equation (B.8) yields

$$\bar{F} = \bar{F}_1 + \bar{F}_2 + \bar{F}_3 \tag{B.14}$$

where

$$\bar{F}_1 = \frac{|\mathrm{grad}(\xi)|}{J} \bar{\lambda}_1 \frac{\gamma-1}{\gamma} \begin{Bmatrix} \rho \\ \rho\bar{u} \\ \rho\bar{v} \\ \rho\bar{w} \\ \frac{\rho}{2}(\bar{u}^2 + \bar{v}^2 + \bar{w}^2) \end{Bmatrix} \tag{B.15}$$

$$\bar{F}_{2,3} = \frac{|\mathrm{grad}(\xi)|}{J} \frac{\bar{\lambda}_{4,5}}{2\gamma} \begin{Bmatrix} \rho \\ \rho\bar{u} \pm \rho a \\ \rho\bar{v} \\ \rho\bar{w} \\ e + p \pm \rho a\bar{u} \end{Bmatrix} \tag{B.16}$$

$$\bar{\lambda}_1 = \frac{\lambda_1}{|\mathrm{grad}(\xi)|} = \frac{U}{|\mathrm{grad}(\xi)|} \tag{B.17}$$

where

$$\bar{\lambda}_{4,5} = \frac{\lambda_{4,5}}{|\mathrm{grad}(\xi)|} = \frac{U \pm a|\mathrm{grad}(\xi)|}{|\mathrm{grad}(\xi)|} \tag{B.18}$$

Applying the inverse transformation to (B.14) gives

$$\hat{F} = T^{-1}\,\bar{F} = \hat{F}_1 + \hat{F}_2 + \hat{F}_3 \tag{B.19}$$

$$\hat{F}_1 = \lambda_1 \frac{\gamma-1}{J\gamma} \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \frac{\rho}{2}(u^2 + v^2 + w^2) \end{Bmatrix} \tag{B.20}$$

$$\hat{F}_{2,3} = \lambda_{4,5} \frac{1}{J2\gamma} \begin{Bmatrix} \rho \\ \rho u \pm \rho a \hat{\xi}_x \\ \rho v \pm \rho a \hat{\xi}_y \\ \rho w \pm \rho a \hat{\xi}_z \\ e + p \pm \frac{\rho a U}{|\mathrm{grad}(\xi)|} \end{Bmatrix} \tag{B.21}$$

This is identical to the generalized splitting given in references 3 and 7.

APPENDIX C

RESTRICTION OPERATORS

The dependent variables are transferred from a fine mesh to a coarse mesh so that the mass, momentum, and energy contained in the coarse grid cell is the same as that contained in the fine grid cells which compose the coarser grid. Referring to figure 23, for a two dimensional example, the mass, momentum and energy in any given cell is given by Q*V where V is the volume of the cell and Q is the column vector of dependent variable representing the conserved quantities per unit volume. Since cell A is comprised of the smaller cells a, b, c, and d, a relationship is easily established for transferring the dependent variables which conserves mass, momentum, and energy.

$$Q_A = \frac{Q_a V_a + Q_b V_b + Q_c V_c + Q_d V_d}{V_A} \tag{C.1}$$

The restriction of the residuals is also guided by conservation laws. The steady Euler equations can be written in integral form as

$$\int \vec{F} \cdot \hat{n} ds = 0 \tag{C.2}$$

Here, the integral is the surface integral over the volume considered, $\vec{F}$ is the flux of mass, momentum, or energy across the boundaries, and $\hat{n}$ is the outward pointing unit vector normal to the boundary.

Considering the two dimensional case given above for simplicity, the integral around the large volume shown in figure 23 is given by

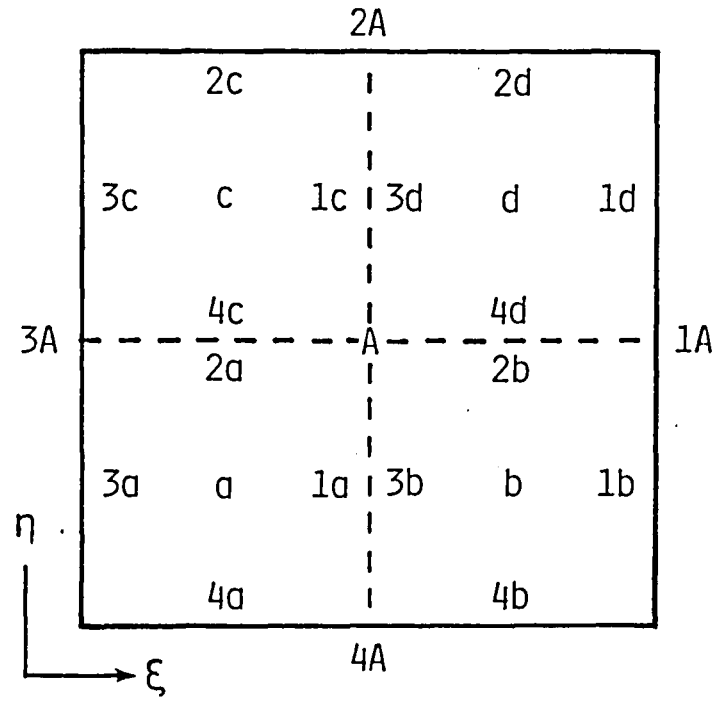


Figure 23. Control volumes for restriction of dependent variables and residual.

$$\int \vec{F} \cdot \hat{n} ds = (\vec{F} \cdot \hat{n})_{1A} S_{1A} + (\vec{F} \cdot \hat{n})_{2A} S_{2A} + (\vec{F} \cdot \hat{n})_{3A} S_{3A} + (\vec{F} \cdot \hat{n})_{4A} S_{4A} \tag{C.3}$$

Now, the integral along each of the larger faces is the sum of the two integrals along each of the smaller ones. For example

$$(\vec{F} \cdot \hat{n})_{1A} S_{1A} = (\vec{F} \cdot \hat{n})_{1b} S_{1b} + (\vec{F} \cdot \hat{n})_{1d} S_{1d} \tag{C.4}$$

Also, note that since the outward pointing normals on adjacent cell boundaries point in opposite directions, several terms which share a common boundary will cancel. For instance

$$(\vec{F} \cdot \hat{n})_{1c} S_{1c} = -(\vec{F} \cdot \hat{n})_{3d} S_{3d} \tag{C.5}$$

Therefore, by performing the integrations around each of the smaller cells and adding them together, it is seen that the integral around the larger cell is simply the sum of the integrals around each of the smaller ones.

$$\int_{\text{large cell}} \vec{F} \cdot \hat{n} ds = \sum \int_{\text{small cells}} \vec{F} \cdot \hat{n} ds \tag{C.6}$$

In order to better relate this specifically to the Euler equations, consider the steady continuity equation given by

$$\int \rho \vec{U} \cdot \hat{n} ds = 0 \tag{C.7}$$

$$\vec{U} = u\hat{i} + v\hat{j} \tag{C.8}$$

$$\hat{n} = \hat{k}_x\hat{i} + \hat{k}_y\hat{j} = \frac{\nabla k}{|\nabla k|} \tag{C.9}$$

A normal to a ξ=constant line is given by using $k = \xi$ in (C.9) and the normal to an η=constant line is obtained using η in the same manner. Now, the length of each face is given by

$$S_i = \frac{|\nabla k|}{J} \tag{C.10}$$

where k is again chosen to be ξ or η depending on which face is desired and J represents the inverse of the cell volume (i.e. the Jacobian). Using equations (C.3) (with $\vec{F} = \rho\vec{U}$), (C.7), (C.8), and (C.9), the integral around cell a in figure 23 can be written as

$$\int \rho\vec{U} \cdot \hat{n}ds = \left(\frac{\rho U}{J}\right)_{1a} - \left(\frac{\rho U}{J}\right)_{3a} + \left(\frac{\rho V}{J}\right)_{2a} - \left(\frac{\rho V}{J}\right)_{4a} \tag{C.11}$$

where

$$U = \xi_x u + \xi_y v$$

$$V = \eta_x u + \eta_y v \tag{C.12}$$

When calculated numerically, this is simply the residual for the continuity equation. Similar results are obtained for the momentum and energy equations. Therefore, the residuals on a fine grid can be transferred to a coarser grid so that the integral of the fluxes over the cell boundary is conserved simply by summing the fine grid residuals which make up the coarse grid.

BIBLIOGRAPHY

1. Moretti, G., "The λ-Scheme," Computers and Fluids, Vol. 7, 1979, pp. 191-205.

2. Chakravarthy, S. R., Anderson, D. A., and Salas, M. D., "The Split-Coefficient Matrix Method for Hyperbolic Systems of Gas Dynamic Equations," AIAA 80-0268, January 1980.

3. Steger, J. L., and Warming, R. F., "Flux Vector Splitting of the Inviscid Gasdynamic Equation with Application to Finite Difference Methods," Journal of Computational Physics, Vol. 40, No. 2, April 1981, pp. 263-293.

4. Steger, J. L., "Preliminary Study of Relaxation Methods for the Inviscid Conservative Gasdynamics Equations Using Flux Splitting", NASA CR-3415, 1981.

5. Van Leer, B., "Flux-Vector Splitting for the Euler Equations," ICASE Report No. 82-30, September 1982; also Lecture Notes in Physics, Vol. 170, 1982, pp. 501-512.

6. Lombard, C. K., Oliger, J., and Yang, J. Y., "A Natural Conservative Flux Difference Splitting for the Hyperbolic Systems of Gasdynamics," AIAA 82-0976, June 1982.

7. Janus, J. M., "The Development of a Three-Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications." Master of Science Thesis, Mississippi State University, August 1984.

8. Thomas, J. L., Van Leer, B., and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations," AIAA 85-1680, July 1985.

9. Anderson, W. K., Thomas, J. L., and Van Leer, B., "A Comparison of Finite Volume Flux Vector Splittings for the Euler Equations", AIAA 85-0122, 1985.

10. Hackbusch, W., Multi-Grid Methods and Applications, Springer-Verlag, 1985.

11 Stuben, K., and Trottenberg, U., "Multigrid Methods: Fundamental Algorithms, Model Problem Analysis and Applications" in Multigrid Methods, Proceedings of the conference held at Koln-Parz, Nov. 1981 (Lecture Notes in Mathematics 960, Springer-Verlag).

12. Ni, Ron-Ho, "A Multiple-Grid Scheme for Solving the Euler Equations", AIAA Journal, Vol. 20, No. 11, November 1982, pp. 1565-1571.

13. Johnson, G. M., "Multiple-Grid Acceleration of Lax-Wendroff Algorithms," NASA TM-82843, March 1982.

14. Johnson, G. M., "Multiple-Grid Convergence Acceleration of Viscous and Inviscid Flow Computations," NASA TM-83361, April 1983.

15. Johnson, G. M., "Convergence Accleration of Viscous Flow Computations," NASA TM-83039, October 1982.

16. Chima, R. V., and Johnson, G. M., "Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm," NASA TM 83376, July 1983.

17. Mulder, W. A., "Multigrid Relaxation for the Euler Equations," Paper presented at the 9th International Conference on Numerical Methods in Fluid Dynamics, July 25-29, 1984, Saclay, France.

18. Jespersen, D. C., "A Multigrid Method for the Euler Equations," AIAA 83-0124, January 1983.

19. Jameson, A., "Solution of the Euler Equations for Two-Dimensional Transonic Flow by a Multigrid Method," Princeton University, MAE Report No. 1613, June 1983.

20. Jameson, A., and Baker, T. J., "Multigrid Solution of the Euler Equations for Aircraft Configurations", AIAA 84-0093, January 1984.

21. Jameson, A., and Yoon, S., "Multigrid Solution of the Euler Equations Using Implicit Schemes", AIAA 85-0293, January 1985.

22. Jameson, A., and Yoon, S., "LU Implicit Schemes with Multiple Grids for the Euler Equations," AIAA 86-0105, January 1986.

23. Buning, P. G., and Steger, J. L., "Solution of the Two Dimensional Euler Equations with Generalized Coordinate Transformation Using Flux-Vector Splitting," AIAA 82-0971, 1982.

24. Van Lavante, E., Claes, D., and Anderson, W. K., "The Effects of Various Implicit Operators on a Flux Vector Splitting Method," AIAA 86-0273, January 1986.

25. Van Leer, B., "Towards the Ultimate Conservative Difference Scheme V. A Second Order Sequel to Gudonov's Method," Journal of Computational Physics, Vol. 32, 1979, pp. 101-136

26. Jespersen, D. C., and Pulliam, T. H., "Flux Vector Splitting and Approximate Newton Methods," AIAA 83-1899.

27. International Mathematical and Statistical Library, Version 9, Houston, Texas.

28. Brandt, A., "Guide to Multigrid Development, "in Multigrid Methods, Proceedings of the conference held at Koln-Porz, Nov. 1981 (Lecture Notes in Mathematics 960, Springer-Verlag).

29. Brandt, A., "Multi-Level Adaptive Solutions to Boundary-Value Problems," Mathematics of Computation, Vol. 31, No. 138, April 1977, pp. 333-390.

30. Brandt, A., "Multilevel Adaptive Computations in Fluid Dynamics," AIAA Journal, Vol. 18, No. 10, October 1980, pp. 1165-1172.

31. Schmitt, V. and Charpin, F. "Pressure Distributions on the ONERA M6-Wing at Transonic Mach Number," AGARD Advisory Report No. 138, May 1979.

32. Townsend, J. C., Howell, D. T., Collins, I. K., and Hayes, C., "Surface Pressure Data on a Series of Analytic Forebodies at Mach Numbers from 1.70 to 4.50 and Combined Angles of Attack and Sideslip," NASA TM 80062, June 1979.

End of Document