12-1991

Computation of Planar Store Trajectories Using an Adaptive Grid Procedure

William D. Hack

AFIT/GAE/ENY/91D-12

COMPUTATION OF PLANAR STORE TRAJECTORIES USING AN ADAPTIVE GRID PROCEDURE

THESIS

William D. Hack, Captain, USAF

AFIT/GAE/ENY/91D-12

AFIT/GAE/ENY/91D-12

COMPUTATION OF PLANAR STORE TRAJECTORIES USING AN ADAPTIVE GRID PROCEDURE

THESIS

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Aeronautical Engineering

William D. Hack, B.S.
Captain, USAF

December 1991

Approved for public release; distribution unlimited

Acknowledgements

The research performed in completing a thesis is very rewarding and difficult. The problems and constraints encountered can almost become insurmountable at times. But, when it is all over and the task is done, it is important to thank the people who helped you along the way.

I would like to start by thanking my wife, Ruby, for her loving support during even the toughest of times.

I thank my committee of Maj Harwood Hegna and Lt Col Jay Dejongh for their time and assistance in the completion of my thesis. I would also like to thank my advisor, Dr Philip Beran for allowing me to do the research I was truly interested in. A special thank you is required for my research sponsor, Dr Bruce Simpson at Wright Laboratory, Armament Directorate, Eglin Air Force Base and Dr Abdollah Arabshabi (Abi) at Mississippi State University for their patience and expert help on the many long-distance phone conversations we had in resolving my many road blocks. I must also thank Capt Mark Driver, Capt Mark Lutton, and Capt Ken Moran for their exceptional assistance.

I would like to conclude by dedicating this work to my late father-in-law, Erven C. Owen.

In any research,
Analyze the Past,
Consider the Present,
Visualize the Future, and
GO FOR IT!! (Tutty)

William D. Hack

Table of Contents

List of Figures

Figure

List of Tables

Nomenclature

English Letters

a	Speed of sound
A,B,C	Jacobians of the flux vectors
c	Local speed of sound
CFD	Computational fluid dynamics
COG	Center of gravity
C_d	Drag coefficient
C_l	Lift coefficient
C_m	Moment coefficient
C_p	Coefficient of pressure
D	Drag force vector
DOF	Degree-of-freedom
e	Energy per unit volume
f,g,h	Cartesian-coordinate flux-vectors
F,G,H,K	Curvilinear-coordinate flux-vectors
I	Identity operator
I_z	Z-Moment of inertia
J	Jacobian of inverse transformation; volume of cell
k	General representation of curvilinear-coordinate
L	Lift force vector
ℓ_s	Reference length
m	mass of body
M	Mach number; moment vector
P	Pressure
ρ	Angular momentum vector
Q	Curvilinear-coordinate dependent variable vector
ΔQ^n	Dependent variable vector change from time n to n+1
q	Cartesian-coordinate dependent variable vector; angular velocity
R	Residual vector
t	Time
u,v,w	Cartesian coordinate velocities
U,V,W	Contravariant velocities
W_t	Weight of body

x,y,z	Cartesian coordinates

Greek letters

α	Angle of attack
γ	Specific heat ratio; COG velocity vector
δ	Central-difference operator
λ	Eigenvalue of flux-Jacobian matrix
ψ	Accuracy parameter
Ψ	Stream function
ρ	Mass density
θ	Ellipse orientation angle
τ	Time in computational space
ξ,η,ζ	Curvilinear coordinates

Subscripts

i,j,k	Mesh point location
i	Split flux number
t	Time differentiation
x,y,z	Space differentiation
ξ,η,ζ,k	Partial differentiation
r	Reference value
∞	Freestream reference value

Superscripts

i	Eigenvalue number
n	Current time level
$n+1$	Next time level
-1	Matrix inverse
$'$	Indicates a dimensional quantity

Abstract

The objective of this research was to compare a quasi-analytical, potential flow/three-degree-of-freedom model to an implicit-Euler algorithm for the calculation of store trajectories. The implicit algorithm uses a cell-centered, finite-volume, spatial discretization applied to the Euler equations, written in time-dependent, curvilinear-coordinates. A flux-differencing Roe scheme is employed to find the split-fluxes and the Steger/Warming flux-vector method is used to calculate the flux-Jacobians. The potential flow and implicit-Euler algorithm are combined with a three-degree-of-freedom algorithm to evaluate the planar, freefall trajectories of a simple store shape. The research uses two different grid-modification techniques in the implicit algorithm evaluation.

Data collected for both grids used the minimum time-step in the three-degree-of-freedom algorithm for a Courant number of 10. Two test cases involved updating the flux-Jacobians after every time-step and only once during every 1000 iterations. The effect of multiplying the minimum time-step by factors of 2, 4, 6, 8, 10, and 100 were also examined.

The potential flow and implicit algorithm trajectories didn't compare very closely. The various Δt and Jacobian-update results matched rather closely.

COMPUTATION OF PLANAR STORE TRAJECTORIES USING AN ADAPTIVE GRID PROCEDURE

Chapter I. Introduction

Weapon separation testing is an integral part of the test and evaluation carried out on every United States Air Force Tactical and Strategic aircraft. Prior to flight test, the only reliable method of determining the trajectory of a weapon is wind-tunnel testing assisted by computer analysis (25:4.34). Accurate and robust Computational Fluid Dynamics (CFD) algorithms would make additional weapon trajectory data readily available.

Wind-tunnel testing is a very costly, time consuming, and rigid process. Wind-tunnel models take several weeks to fabricate and are very expensive. If modifications to the aircraft or weapon occur during the testing process, wind-tunnel data may be invalidated. Wind-tunnel tests generally take several weeks to plan and prepare, and once completed the data are only valid for the weapon and aircraft configurations tested. Flight test planning uses these data in determining the best and safest, separation-flight test.

Flight testing is also costly; it is hazardous as well. Before flight testing occurs, both the engineer and pilot need an idea of the trajectory of the weapon. Only wind-tunnel data provides this trajectory information. If modifications to the weapon or aircraft occur, the wind-

tunnel test data may be invalid, sometimes requiring further wind-tunnel testing or a more drawn out and hazardous flight-test program. If, however, the weapon and aircraft could be computationally modeled, changes to the aircraft or weapon could be easily incorporated and new trajectory data computed relatively quickly.

There is still much research required before CFD algorithms will be able to provide accurate weapon separation data. This research is an attempt in furthering the development of a CFD algorithm to the weapon separation problem. I will investigate the problem of computationally moving a simple geometric shape (ellipse) through a two-dimensional flow-field using Whitfield's, dynamic-grid, Euler algorithm (28). The investigation will be limited to planar trajectories and will ignore viscous effects. The work presented will exercise the Whitfield algorithm and compare its results against a quasi-analytical, potential-flow result.

The organization of this thesis is as follows. Chapter II presents the potential-flow solution and differential form of Euler equations. The transformation of the Euler equations from rectangular Cartesian-coordinates to general time-dependent, curvilinear-coordinates is explained. This transformation allows dynamic grids to be utilized. Chapter III presents the implicit, upwind, finite-volume, flux-vector scheme. This chapter also discusses the use of flux-differencing with Roe-averaging and the use of flux-limiters

to achieve schemes with higher-order accuracy. Chapter IV gives a brief discussion of the equations of motion used to generate the Three-Degree-of-Freedom (3-DOF) algorithm. The grid-manipulation technique used in this research is also presented. Chapter V presents the results of the research. The research compares the potential-flow results with results from the Whitfield algorithm. It continues by examining the effect of modifying the flux-Jacobian update methodology and the minimum time-step on the Whitfield algorithm. Chapter VI closes with conclusions and recommendations on further research.

I.1 Summary of Current Knowledge

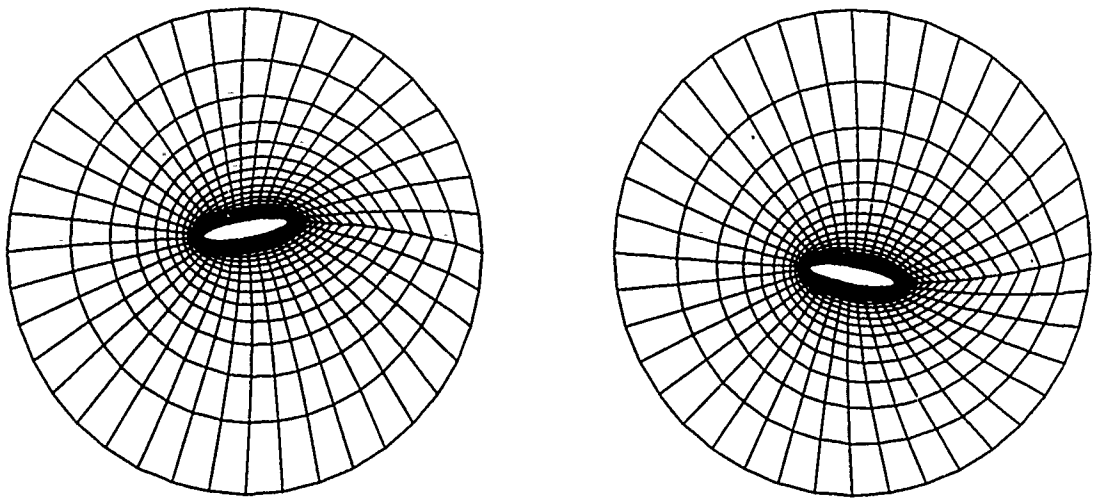
Wind-tunnel testing with computer simulation is currently the only method of resolving the trajectory of a weapon from an aircraft without flight testing. Wind-tunnel testing obtains the forces and moments on the weapon used in a Six-Degree-of-Freedom (6-DOF) program utilized by the 3246th Test Wing/TYES to compute store trajectories (17:1). The 6-DOF program determines the trajectory of a weapon very accurately when compared with flight-test data. This technique has been reliably used for many years. The trajectories, however, are limited to the aircraft and weapon configurations and flight-test conditions performed during wind-tunnel testing. Only changes in the moments-of-inertia, center-of-gravity (COG), and weight of the weapon can be accounted for in the 6-DOF algorithm. Physical

modifications to the weapon or aircraft require new aerodynamic data to account for changes in the flow-field about the weapon and aircraft. These data are in turn used to calculate the correct forces and moments present on the weapon and its trajectory.

The wind-tunnel test limitations could be overcome by the development of CFD algorithms, which could accurately model weapon separation. The Wright Laboratory, Armament Directorate, Eglin Air Force Base is developing CFD algorithms to model the separation of weapons from aircraft. The Whitfield algorithm was developed by Mississippi State University under contract as part of this work for the Armament Directorate (6:1). The algorithm accurately models the forces and moments on the weapon generated by the surrounding flow-field. In References 1, 2, 6 through 8, and 27 through 29, algorithm results have shown good agreement with wind-tunnel data for various store and airfoil shapes. These results contained data for both stable and dynamic grids in both planar and three-dimensional cases. The algorithm can account for viscous and compressibility effects in subsonic, transonic, and supersonic flow. This algorithm has the added ability of utilizing blocked grids (8:31). This allows a tailored grid to be generated around each object, such as an aircraft and weapon, which is to be modelled. These tailored grids can be coupled and a complete solution of the flow-field calculated. For example, the grid around the weapon

requires modification as the weapon trajectory is determined by the aerodynamic forces and launch conditions.

Figure 1 shows an example of how the grid can be modified as a store (an ellipse for this example) is dropped. The Whitfield algorithm has, however, only been examined for dynamic grids with specified trajectories (7:13). These specified trajectories move the entire grid. The algorithm hasn't been tested for modified grids similar to that shown in Figure 1



(a) (b)

Figure 1: Grid at t = 1 (a) and Grid at t = 2 (b)

My research extends the testing of the Whitfield algorithm into the realm of unspecified trajectories. This requires modification of the grid after each time-step to account for the new position of the ellipse as specified by the 3-DOF model. The calculated trajectory will be compared with trajectories calculated using the quasi-analytical,

potential-flow solution. This assumes a good comparison exists between low-speed, compressible flow and quasi-analytical, potential flow. By adding the 3-DOF algorithm, and eventually the 6-DOF algorithm, to the Whitfield algorithm, a true CFD algorithm could result that would fill a void in the current capabilities of weapon test and evaluation.

Chapter II. Theory

II.1 Potential-Flow Solution

The initial model developed for this research uses an ellipse to represent a weapon and an analytical, potential-flow solution. The comparison with the Whitfield algorithm solutions utilizes the analytical solution for potential flow about an ellipse in determining a stable trajectory. A stable trajectory is defined as one where the pitching motion of the store decreases as the store transitions into freefall. Stable stores generally require the COG to be forward of the center-of-pressure. The potential-flow assumption initially limits the research to cases of inviscid and incompressible flow. These assumptions are used for verifying the trajectory algorithm.

Lamb presents a potential-flow solution for the flow-field about a rotating ellipse. He used the complex potential that describes potential flow past a circular cylinder. Lamb transformed this flow, using complex variables, into flow about an ellipse (14:89). Lamb presents the following form for the stream function

$$\Psi = \left(\frac{a+b}{a-b}\right)^{1/2} \exp(-\xi)\ (Ub\ sin\eta - Va\ cos\eta)$$

$$+ \frac{1}{4}\omega(a+b)^2\ exp(-2\xi)\ cos2\eta$$

$$+ c(U\ sinh\xi\ sin\eta - V\ cosh\xi\ cos\eta) \qquad (1)$$

In Eq. 1, U and V represent the two freestream, velocity-components; ω is the angular velocity of the ellipse, measured about its center; a and b are the semi-major and semi-minor axis of the ellipse. Figure 2 shows the orientation of U, V, ω, a, and b.

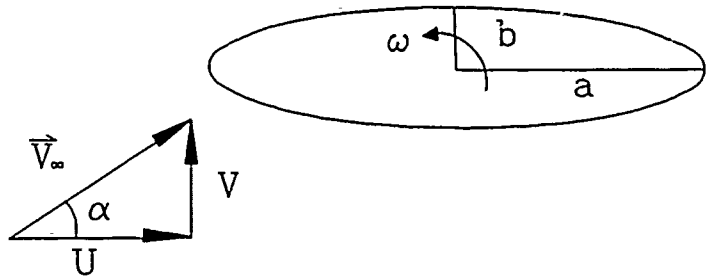


Figure 2: Velocity Component Relationship

c is a parameter defined by

$$c \equiv \sqrt{a^2 - b^2} \tag{2}$$

The semi-minor axis for the ellipse used in this research is $0.2a$.

ξ and η are the elliptic coordinates used in the complex variable transformation. The transformation equations are

$$x = c \cosh\xi \cos\eta \tag{3a}$$

$$y = c \sinh\xi \sin\eta \tag{3b}$$

The calculations of lift, drag, and moment on the ellipse use a trapezoidal integration of the surface

pressure. The calculation assumes no atmospheric pressure gradient. The surface-velocity components, u and v, are calculated from the stream function, Eq. 1, by

$$u = \frac{\partial \Psi}{\partial y} \tag{4a}$$

$$v = -\frac{\partial \Psi}{\partial x} \tag{4b}$$

or, in transformed coordinates

$$\frac{\partial \Psi}{\partial x} = \frac{1}{J}(y_\eta \Psi_\xi - y_\xi \Psi_\eta) \tag{5a}$$

$$\frac{\partial \Psi}{\partial y} = \frac{1}{J}(-x_\eta \Psi_\xi + x_\xi \Psi_\eta) \tag{5b}$$

The Jacobian, J, in Eqs. 5 is

$$J = x_\xi y_\eta - x_\eta y_\xi \tag{6}$$

In this potential-flow solution, there is no circulation about the ellipse. Thus, the resulting lift component is zero. The drag is also zero, since the pressure distribution is exactly balanced. The calculated results helped validate the trapezoidal-integration technique. The moment calculation is also utilized as a comparison of the integration accuracy. Batchelor gives an analytical solution for the moment about a non-rotating ellipse at a specified angle of attack (4:435),

$$M = -\rho\pi uv(a^2 - b^2) \tag{7}$$

This equation was used to verify the moment calculation. Appendix A contains results of the trapezoidal-integration analysis.

II.2 The Euler Equations

The Euler equations are a simplified form of the Navier-Stokes equations describing the behavior of continuum fluid flows. The Euler equations neglect the effects of viscosity, heat transfer, and body forces. The prime denotes dimensional variables in the following equations. The conservative, differential form of the Euler equations in Cartesian-coordinates are (29:1)

$$\frac{\partial q'}{\partial t'} + \frac{\partial f'}{\partial x'} + \frac{\partial g'}{\partial y'} + \frac{\partial h'}{\partial z'} = 0 \tag{8}$$

where q', f', g', and h' are defined by

$$q' = [\rho',\ \rho'u',\ \rho'v',\ \rho'w',\ e']^T \tag{9a}$$

$$f' = [\rho'u',\ \rho'u'^2 + p',\ \rho'v'u',\ \rho'w'u',\ (e'+p')\,u']^T \tag{9b}$$

$$g' = [\rho'v',\ \rho'u'v',\ \rho'v'^2,\ +\ p',\ \rho'w'v',\ (e'+p')\,v']^T \tag{9c}$$

$$h' = [\rho'w',\ \rho'u'w',\ \rho'v'w',\ \rho'w'^2 + p',\ (e'+p')\,w']^T \tag{9d}$$

The Cartesian (x', y', z') velocity-components are represented by (u', v', w'). P' and ρ' represent the pressure and density, respectively. The total energy of the flow e', is defined through the perfect gas law assumption as

$$e' = \frac{P'}{\gamma-1} + \frac{1}{2}\rho'(u'^2 + v'^2 + w'^2) \tag{10}$$

where, γ is the ratio of specific heats ($\gamma = 1.4$). More often, the Euler equations are written in nondimensional form. The nondimensional variables used in this analysis are defined as (2:7)

$$\rho = \frac{\rho'}{\rho'_\infty} \qquad P = \frac{P'}{\rho'_\infty a'^2_\infty} \qquad e = \frac{e'}{\rho'_\infty a'^2_\infty} \tag{11a}$$

$$u = \frac{u'}{a'_\infty} \qquad v = \frac{v'}{a'_\infty} \qquad w = \frac{w'}{a'_\infty} \qquad a = \frac{a'}{a'_\infty} \tag{11b}$$

$$x = \frac{x'}{l'} \qquad y = \frac{y'}{l'} \qquad z = \frac{z'}{l'} \qquad t = \frac{a'_\infty t'}{l'} \tag{11c}$$

where

$$a'_\infty = \left(\frac{\gamma P'_\infty}{\rho'_\infty}\right)^{\frac{1}{2}} \tag{12}$$

is the freestream speed of sound. The variable, l', represents any convenient reference length (2:8). The reference length is $2a$ for the ellipse.

Eq. 8 can be rewritten in nondimensional form as

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = 0 \tag{13}$$

II.2.1 Generalized Curvilinear-Coordinate Transformation

The development of the Euler equations used the Cartesian-coordinate system (x, y, z). In most applications, however, this coordinate system doesn't facilitate the accurate numerical treatment of boundary conditions. It is also insufficient in representing complex geometries, such as airfoils, aircraft, weapons, etc. Instead, a boundary-conforming, coordinate system should be incorporated. The curvilinear-coordinates are explicit functions of the Cartesian-coordinates

$$\xi = \xi(x, y, z, t) \tag{14a}$$

$$\eta = \eta(x, y, z, t) \tag{14b}$$

$$\zeta = \zeta(x, y, z, t) \tag{14c}$$

$$\tau = t \tag{14d}$$

These curvilinear-coordinates change the nondimensional form of the Euler equations to

$$\frac{\partial Q}{\partial \tau} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} + \frac{\partial H}{\partial \zeta} = 0 \tag{15}$$

where

$$Q = J \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix} \qquad F = J \begin{bmatrix} \rho U \\ \rho uU + \xi_x P \\ \rho vU + \xi_y P \\ \rho wU + \xi_z P \\ (e+P)U - \xi_t P \end{bmatrix} \tag{16a}$$

$$G = J \begin{bmatrix} \rho V \\ \rho uV + \eta_x P \\ \rho vV + \eta_y P \\ \rho wV + \eta_z P \\ (e+P)V - \eta_t P \end{bmatrix} \qquad H = J \begin{bmatrix} \rho W \\ \rho uW + \zeta_x P \\ \rho vW + \zeta_y P \\ \rho wW + \zeta_z P \\ (e+P)W - \zeta_t P \end{bmatrix} \tag{16b}$$

U, V, and W, are the contravariant velocities, in the ξ, η, and ζ-coordinate direction. They are defined as

$$U = \xi_t + \xi_x u + \xi_y v + \xi_z w \tag{17a}$$

$$V = \eta_t + \eta_x u + \eta_y v + \eta_z w \tag{17b}$$

$$W = \zeta_t + \zeta_x u + \zeta_y v + \zeta_z w \tag{17c}$$

The Jacobian, J, represents the volume of each three-dimensional cell in physical space and is

$$J = x_\xi (y_\eta z_\zeta - z_\eta y_\zeta) - y_\xi (x_\eta z_\zeta - z_\eta x_\zeta) + z_\xi (x_\eta y_\zeta - y_\eta x_\zeta) \tag{18}$$

The metric terms are

$$\xi_x = J^{-1}(y_\eta z_\zeta - z_\eta y_\zeta) \qquad \eta_x = J^{-1}(z_\xi y_\zeta - y_\xi z_\zeta) \tag{19a}$$

$$\xi_y = J^{-1}(z_\eta x_\zeta - x_\eta z_\zeta) \qquad \eta_y = J^{-1}(x_\xi z_\zeta - z_\xi x_\zeta) \tag{19b}$$

$$\xi_z = J^{-1}(x_\eta y_\zeta - y_\eta x_\zeta) \qquad \eta_z = J^{-1}(y_\xi x_\zeta - x_\xi y_\zeta) \tag{19c}$$

$$\xi_t = (-x_\tau \xi_x - y_\tau \xi_y - z_\tau \xi_z) \qquad \eta_t = (-x_\tau \eta_x - y_\tau \eta_y - z_\tau \eta_z) \tag{19d}$$

$$\zeta_x = J^{-1}(y_\xi z_\eta - z_\xi y_\eta) \tag{19e}$$

$$\zeta_y = J^{-1}(x_\eta z_\xi - z_\eta x_\xi) \tag{19f}$$

$$\zeta_z = J^{-1}(x_\xi y_\eta - y_\xi x_\eta) \tag{19g}$$

$$\zeta_t = (-x_\tau \zeta_x - y_\tau \zeta_y - z_\tau \zeta_z) \tag{19h}$$

Janus details these transformations (13:92-94).

Chapter III. Algorithm Development

III.1 The Implicit Algorithm

The algorithm that solves the Euler equations is a combination of the Roe scheme and the Steger/Warming flux-vector splitting scheme (2:35). These schemes take advantage of characteristic theory as applied to the hyperbolic Euler equations. The advantages to this method are the elimination of numerical viscosity or smoothing terms and an increase in the convergence rate. This algorithm exploits the accuracy of the Roe scheme in defining discontinuities and the relative computational ease of flux-vector splitting techniques to give accurate solutions. The order of accuracy of the algorithm is user specified by the limiter applied. Limiters and their order of accuracy are discussed in Chapter III.3.

III.1.1 Flux-Vector Splitting

Flux-vector splitting developed by Steger and Warming (Reference 24) in the 1980s has been widely used in the numerical solution of the Euler equations. The procedure accounts for the direction of information travel determined from characteristic theory to compute fluxes. The three-dimensional Euler equations (Eq. 15) are a system of five equations that have five characteristic velocities in each of the three coordinate directions. These characteristic

velocities are resolved from the quasilinear form of Eq. 15 (29:2),

$$\frac{\partial Q}{\partial \tau} + A\frac{\partial Q}{\partial \xi} + B\frac{\partial Q}{\partial \eta} + C\frac{\partial Q}{\partial \zeta} = 0 \tag{20}$$

where the flux-Jacobian matrices, A, B, and C, are

$$A = \frac{\partial F}{\partial Q} \qquad B = \frac{\partial G}{\partial Q} \qquad C = \frac{\partial H}{\partial Q} \tag{21}$$

The eigenvalues of A are the characteristic velocities in the ξ-direction. Similarly, the eigenvalues of B and C are the characteristic velocities in the η- and ζ-directions, respectively.

F, G, and H are identical except ξ appears in F, η in G, and ζ in H. We can simplify the analysis by letting K represent either F, G, or H (28:3). We can then define

$$\bar{K} = \frac{\partial K}{\partial Q} \tag{22}$$

$\bar{K}$ corresponds to A, B, and C depending on the meaning of K. The eigenvalues of the matrix K are (13:18)

$$\lambda_k^1 = \lambda_k^2 = \lambda_k^3 = k_x u + k_y v + k_z w + k_t = \beta_k \tag{23a}$$

$$\lambda_k^4 = \beta_k + c|\nabla k| = \beta_k + c(k_x^2 + k_y^2 + k_z^2)^{\frac{1}{2}} \tag{23b}$$

$$\lambda_k^5 = \beta_k - c|\nabla k| \tag{23c}$$

where here c is the speed of sound and k is either ξ, η, or ζ corresponding to A, B, or C, respectively.

The flux-vector, K, can be split into three parts, each part corresponding to the distinct eigenvalues of K given in Eqs. 23. Janus gives the details of this splitting (13:98-100). The flux-vector, K, is then written

$$K = \lambda_k^1 K_1 + \lambda_k^4 K_4 + \lambda_k^5 K_5 \tag{24}$$

where

$$K_1 = J\frac{\gamma-1}{\gamma}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \frac{\rho}{2}(u^2+v^2+w^2) \end{bmatrix} \qquad K_4 = \frac{J}{2\gamma}\begin{bmatrix} \rho \\ \rho u+\rho c\bar{k}_x \\ \rho v+\rho c\bar{k}_y \\ \rho w+\rho c\bar{k}_z \\ e+p+\rho c\bar{\theta}_k \end{bmatrix}$$

$$K_5 = \frac{J}{2\gamma}\begin{bmatrix} \rho \\ \rho u-\rho c\bar{k}_x \\ \rho v-\rho c\bar{k}_y \\ \rho w-\rho c\bar{k}_z \\ e+p-\rho c\bar{\theta}_k \end{bmatrix} \tag{25}$$

and

$$\bar{k}_x = \frac{k_x}{|\nabla k|} \qquad \bar{k}_y = \frac{k_y}{|\nabla k|} \qquad \bar{k}_z = \frac{k_z}{|\nabla k|} \tag{26}$$

$$\bar{\theta}_k = \bar{k}_x u + \bar{k}_y v + \bar{k}_z w \tag{27}$$

The sign of λ in Eqs. 23 controls the direction that information is used in determining the analogous portion of the flux, K_i, where i = 1, 4, and 5 in Eqs. 25. This flux-vector split is not unique and other techniques for splitting into positive and negative parts are possible (2:15). The splitting used by Steger and Warming is given by

$$F = F^+ + F^- \tag{28a}$$

$$G = G^+ + G^- \tag{28b}$$

$$H = H^+ + H^- \tag{28c}$$

Here, the plus superscript signifies the portion of the flux-vector associated with the non-negative eigenvalues. The minus superscript signifies the portion of the flux-vector associated with the non-positive eigenvalues (24:269).

III.1.2 Finite-Volume Formulation

This algorithm uses a finite-volume formulation, thus allowing for the modeling of arbitrary configurations. It can better manage the complicated grid-structure, than the finite-difference formulation (2:15). The computational domain is divided up into many small "non-overlapping" cells. In this way Eq. 8, can be evaluated over each cell separately or over the entire computational domain. The

resulting solution will fulfill the Euler equations over a specified volume.

The dependent variables in these equations are stored at each grid point or cell center. The flux-vectors used in this finite-volume formulation, however, are required at cell surfaces. Figure 3 shows a typical portion of the computational domain and its nomenclature.

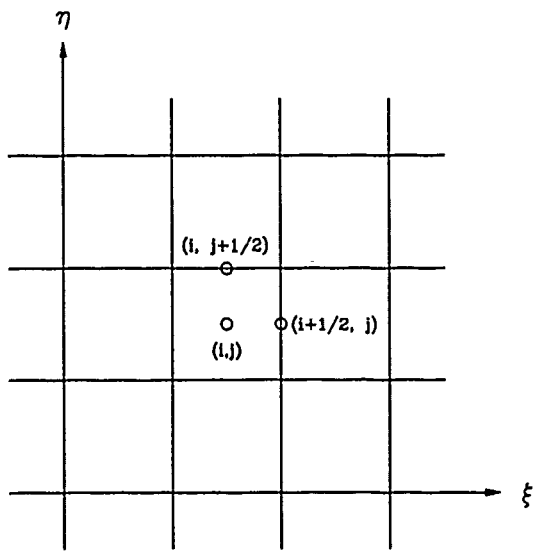


Figure 3: Indexing of Cell Centers and Surfaces

One discretized form of the three-dimensional, unsteady, Euler equations (Eq. 15) is

$$\frac{Q^{n+1}_{i,j,k}-Q^{n}_{i,j,k}}{\Delta\tau} + \frac{F^{n+1}_{i+1/2,j,k}-F^{n+1}_{i-1/2,j,k}}{\Delta\xi} + \frac{G^{n+1}_{i,j+1/2,k}-G^{n+1}_{i,j-1/2,k}}{\Delta\eta}$$

$$+ \frac{H^{n+1}_{i,j,k+1/2}-H^{n+1}_{i,j,k-1/2}}{\Delta\zeta} = 0 \qquad (29)$$

Eq. 29 can be written more compactly as

$$\Delta Q^n + \Delta\tau\left(\delta_\xi F^{n+1}+\delta_\eta G^{n+1}+\delta_\zeta H^{n+1}\right) = 0 \tag{30}$$

where

$$\Delta Q^n = Q^{n+1} - Q^n \tag{31}$$

and

$$\delta_\xi F^{n+1} = \frac{\left(F^{n+1}_{i+1/2,j,k} - F^{n+1}_{i-1/2,j,k}\right)}{\Delta\xi} \tag{32}$$

The central differences of G and H are defined similarly. The n superscript indicates values at the current time-step, while the n+1 superscript indicates unknown values at the next time-step.

The implicit, flux-split discretization of the Euler equations is formed by substituting Eqs. 28 into Eq. 30 to give,

$$\Delta Q^n+\Delta\tau\left[\delta_\xi\left(F^++F^-\right)^{n+1}+\delta_\eta\left(G^++G^-\right)^{n+1}+\delta_\zeta\left(H^++H^-\right)^{n+1}\right] = 0 \tag{33}$$

The flux-vectors F^{n+1}, G^{n+1}, and H^{n+1} are nonlinear functions of the dependent variables, Q^{n+1}. A linearization procedure using a local Taylor series expansion of the flux-vectors about the time-step, n, was suggested by Beam and Warming (5:89). The Taylor series expansion is

$$(F^+)^{n+1} = F^{+n} + A^+(Q^{n+1} - Q^n) + O(\Delta\tau^2) \tag{34a}$$

$$(F^-)^{n+1} = F^{-n} + A^-(Q^{n+1} - Q^n) + O(\Delta\tau^2) \tag{34b}$$

where

$$A^+ = \left(\frac{\partial F^+}{\partial Q}\right)^n \qquad A^- = \left(\frac{\partial F^-}{\partial Q}\right)^n \tag{35}$$

Similar linearizations are used for the split versions of G^{n+1} and H^{n+1}. Substitution of Eqs. 34 into Eq. 33 yields,

$$[I + \Delta\tau(\delta_\xi A^{+\cdot} + \delta_\xi A^{-\cdot} + \delta_\eta B^{+\cdot} + \delta_\eta B^{-\cdot} + \delta_\zeta C^{+\cdot} + \delta_\zeta C^{-\cdot})]\Delta Q^n = -\Delta\tau R^n \tag{36a}$$

where

$$R^n = [\delta_\xi(F^+ + F^-)^n + \delta_\eta(G^+ + G^-)^n + \delta_\zeta(H^+ + H^-)^n] \tag{36b}$$

R^n is called the residual. The dot in the above equation symbolizes that the difference operators apply to the product of the flux-Jacobian matrices, A, B, and C, with ΔQ^n. It is important to note that the linearization in Eqs. 34 is second-order accurate in time, so that the algorithm, as represented by Eqs. 36 remains second-order accurate.

The flux-Jacobians on the left-hand-side of Eq. 36a are determined using the flux-vector, split scheme developed by Steger and Warming (24). The residual term implements the Roe Scheme in determining the values of the split fluxes.

The flux-Jacobians would be computationally very costly to obtain using the Roe Scheme. Whitfield states, "in all results obtained thus far, the Jacobian matrices corresponding to the flux-vector split scheme gave improved convergence rates and a more robust scheme than did the Roe matrices. The reason for this is unclear (28:21)." This combination of the Steger/Warming and Roe schemes were demonstrated in Reference 2 to provide accurate, dynamic-grid results compared with wind-tunnel data.

III.1.3 Factorization

A numerical solution of Eqs. 36 at this point cannot be realistically obtained because of the large-banded matrix of the system. The left-hand-side of the equation requires the inversion of a large matrix and an exact solution would be too numerically intensive. A way to alleviate this problem is to factor the operator on the left-hand-side of Eq. 36a into several operators. Whitfield in Reference 27 and Anderson in Reference 1 have applied many such factorizations to Eqs. 36. These schemes include a six, three, and two-factor scheme. The two-factor scheme used in this research, separates the implicit operator into one factor containing all the flux-Jacobians with the non-negative eigenvalues and the other factor containing all the Jacobians with the non-positive eigenvalues. The two-factor scheme can be written as

$$[I+\Delta\tau(\delta_\xi A^{+\cdot}+\delta_\eta B^{+\cdot}+\delta_\zeta C^{+\cdot})]\,[I+\Delta\tau(\delta_\xi A^{-\cdot}+\delta_\eta B^{-\cdot}+\delta_\zeta C^{-\cdot})]\,\Delta Q^n=-\Delta\tau R^n \tag{37}$$

This equation can be solved in the following steps

$$[I + \Delta\tau(\delta_\xi A^{+\cdot} + \delta_\eta B^{+\cdot} + \delta_\zeta C^{+\cdot})]\,\Delta Q^* = -\Delta\tau R^n \tag{38a}$$

$$[I + \Delta\tau(\delta_\xi A^{-\cdot} + \delta_\eta B^{-\cdot} + \delta_\zeta C^{-\cdot})]\,\Delta Q^n = \Delta Q^* \tag{38b}$$

$$Q^{n+1} = Q^n + \Delta Q^n \tag{38c}$$

The spatial-differences on the left-hand-side of Eq. 37 are only required to be first-order accurate to retain second-order accuracy overall. The implicit algorithm is trying to drive $\Delta Q \rightarrow 0$ in the steady-state calculations, so higher-order, spatial-difference terms are not necessary on a fine grid. The spatial-difference terms are evaluated using one-point extrapolation as given by (2:21)

$$\delta_\xi(A^+\Delta Q^n)_{i,j,k} = [A^+(Q^n_{i,j,k})\,\Delta Q^n_{i,j,k} - A^+(Q^n_{i-1,j,k})\,\Delta Q^n_{i-1,j,k}] \tag{39a}$$

$$\delta_\xi(A^-\Delta Q^n)_{i,j,k} = [A^-(Q^n_{i,j,k})\,\Delta Q^n_{i,j,k} - A^-(Q^n_{i-1,j,k})\,\Delta Q^n_{i-1,j,k}] \tag{39b}$$

The metrics at appropriate cell face's are required to evaluate the Jacobian matrices. The $A^+(Q^n{}_{i,j,k})$ and $A^-(Q^n{}_{i,j,k})$ terms use the metrics at the $(i+1/2,j,k)$ and $(i-1/2,j,k)$ cell faces, respectively. The spatial-differences, $\delta_\eta(B^+\Delta Q)$, $\delta_\eta(B^-\Delta Q)$, $\delta_\zeta(C^+\Delta Q)$, and $\delta_\zeta(C^-\Delta Q)$ are similarly defined.

The two-factor scheme requires only the solution of a sparse, block lower and upper-triangular matrix at each time-step (2:21). The solution to Eq. 38a is calculated by a simple-forward substitution and the solution of Eq. 38b by a simple-backward substitution. This approach, however, requires the storage of three, large, flux-Jacobians matrices at each of the cell centers at any given time-level. Anderson presents the stability analysis for the two factor scheme in Reference 1. His analysis suggests Courant number doesn't affect the two-factor scheme and it retains stability for Courant numbers up to 35 (1:4).

III.2 Flux-Difference Splitting

Thus far, the discussion has focused on the Steger and Warming flux-vector, splitting scheme, which is used to evaluate the left-hand-side of Eqs. 36. Whitfield (Reference 28) also incorporates the flux-difference scheme developed by Roe, in the solution to the right-hand-side of Eqs. 36. The flux-difference scheme is incorporated, since it has been shown to capture shocks and contact discontinuities in as few as one cell. Whitfield's algorithm was designed to investigate a wide range of fluid flow problems, which has been enhanced by the incorporation of the Roe scheme. Bram van Leer demonstrated a first-order, Roe scheme required only one-fourth to one-half as many grid points to accurately capture shocks and contact discontinuities as a second-order, flux-vector split scheme

(2:23). These advantages are being exploited by the incorporation of the flux-difference scheme into the flux-vector scheme.

III.2.1 Riemann Problem

The very essence of the flux-difference splitting scheme is the solution of the local Riemann problem (28:3). Considering the initial-value problem for a hyperbolic conservative system of equations written

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{40}$$

$$q(x,0) = \bar{q}(x) \qquad for\ -\infty < x < \infty \tag{41}$$

where $q(x,t)$ is a column vector of m unknowns and $f(u)$, the flux, is a vector-valued function of m unknowns. The finite-volume discretization of Eq. 40 is given as

$$\frac{q_i^{n+1} - q_i^n}{\Delta t} + \frac{f'_{i+1/2} - f'_{i-1/2}}{\Delta x} = 0 \tag{42}$$

The $q^n{}_i$ term represents the average value at time, $t^n = n\Delta t$, in the interval $(i-1/2)\Delta x \le x \le (i+1/2)\Delta x$. The numerical flux-vector, $f'_{i+1/2}$, is evaluated at the grid-cell interface.

Godunov developed a procedure to advance the solution, q_i, to the next time-level by solving a set of Riemann problems at each cell interface (2:24). The Riemann problem

can be strictly applied to a set of conservative equations if initial data are given over semi-infinite intervals. For example, consider the initial data (Eq. 41) given for the initial-value problem (Eq. 40)

$$q(x,0) = \begin{pmatrix} q_L & for\ x < (i+1/2)\Delta x \\ q_R & for\ x > (i+1/2)\Delta x \end{pmatrix} \qquad (43)$$

where $q_L = q_i$ and $q_R = q_{i+1}$ for all cell interfaces, $x = (i+1/2)\Delta x$. Godunov utilized an integral average over the interval, $(i-1/2)\Delta x < x < (i+1/2)\Delta x$ to solve the Riemann problem at the cell interfaces to obtain q_i^{n+1} (2:25). The time-interval selected in Godunov's scheme must ensure no interaction between neighboring Riemann problems occurs, resulting in an exact solution to the Riemann problem over the complete domain. Approximate Riemann solvers have been developed to reduce the complex structure, but retain the important properties of the solution.

III.2.2 Roe's Approximate Riemann Solver

Roe developed a method utilizing the Riemann solver for solving a set of linear-conservation laws (2:26). He computed the exact solution of the Riemann problem for the linear-hyperbolic system, instead of finding an approximate solution to the complete equations. The linearized system is a rewritten form of Eqs. 40-41,

$$\frac{\partial q}{\partial t} + \bar{A}(q_L, q_R)\frac{\partial q}{\partial x} = 0 \tag{44}$$

$$q(x,0) = \begin{pmatrix} q_L & for\ x < 0 \\ q_R & for\ x > 0 \end{pmatrix} \tag{45}$$

where $\bar{A}(q_L, q_R)$ is a constant matrix. The constant matrix is chosen to represent local interface conditions. Roe required the matrix, $\bar{A}(q_L, q_R)$, to have the following properties (2:26):

(i) It forms a linear mapping from the vector space, q, to the vector space, f

(ii) As $q_L \rightarrow q_R \rightarrow q$, $\bar{A}(q_L, q_R)$ approaches $A(q)$, where $\bar{A} = \partial f/\partial q$

(iii) For any q_L, q_R: $\bar{A}(q_L, q_R) * (q_L - q_R) = f(q_L) - f(q_R) = f_L - f_R$

(iv) The eigenvalues of $\bar{A}$ are linearly independent.

Once the $\bar{A}$ matrix is assembled, its eigenvalues can be estimated as the wavespeeds of the Riemann problem. Eigenvalues of the matrix $\bar{A}$ are represented by λ^j. The eigenvalues are distinct and arranged in increasing order (i.e. $\lambda^1 < \lambda^2 < \ldots < \lambda^m$). The corresponding left and right eigenvectors are represented by ℓ^j and r^j, respectively. It can be shown that the change in the dependent variable is proportional to the right eigenvectors of the flux-Jacobian matrix, whether the matrix $\bar{A}$ is constant or not (2:27). This means dq across each characteristic curve, connected

with a certain eigenvalue, is proportional to the right eigenvector, r^j, associated with that eigenvalue. Thus,

$$q_R - q_L = \sum_{j=1}^{m} \alpha_j r^j \tag{46}$$

An intermediate state, q_K, between any two characteristic curves, K and $K+1$, can be computed in terms of the right eigenvectors of $\bar{A}$ in the following way

$$q_K = q_L + \sum_{j=1}^{K} \alpha_j r^j \tag{47}$$

The interface differential, dq, is proportional to the right eigenvector of the $\bar{A}$ matrix and considering that the right eigenvectors are calculated from $(\bar{A} - \lambda^j I)\, r^j = 0$, the following equation can be developed (2:27)

$$\bar{A} dq^{(j)} = \lambda^j dq^{(j)} \tag{48}$$

Therefore, the total change across all characteristic curves is calculated by

$$\bar{A}(q_R - q_L) = \sum_{j+1}^{m} \alpha_j \lambda^j r^j = f_R - f_L = \sum_{i=1}^{m} df_j \tag{49}$$

The quantity, $df_j = \alpha_j \lambda^j r^j$, represents the change in the flux-vector across the characteristic curve. The flux-vector is required at each cell interface to solve the hyperbolic system of conservation laws. Let's say, a cell

interface is located at $x = 0$; $f'_{i+1/2}$ is the numerical flux-vector at that interface, which can be computed by either of the following two expressions

$$f'_{i+1/2} = f_L + \sum_{j=1}^{m} \alpha_j \lambda^{-j} r^j \tag{50a}$$

$$f'_{i+1/2} = f_R - \sum_{j=1}^{m} \alpha_j \lambda^{+j} r^j \tag{50b}$$

where - and + represent the summation over the negative and positive eigenvalues of "wavespeeds", respectively (2:28). The average of the two flux-vector representation is

$$f'_{i+1/2} = \frac{1}{2}\left[f_R + f_L - \sum_{j=1}^{m} \alpha_j |\lambda^j| r^j\right] \tag{51}$$

where α_j is the magnitude of the jth wave. Whitfield, in Reference 28, has shown that by the hyperbolic nature of the system, the matrix $\bar{A}$ can be diagonalized through a similarity transformation as $\bar{A} = T\Lambda T^{-1}$. The diagonal matrix, Λ, is composed of the eigenvalues of $\bar{A}$. T is a matrix whose columns are the right eigenvalues of $\bar{A}$ and T^{-1} is the matrix whose rows are the left eigenvalues of $\bar{A}$. Eq. 51 can be written as

$$f'_{i+1/2} = \frac{1}{2}\left[f_R + f_L - |\bar{A}(q_L, q_R)|(q_R - q_L)\right] \tag{52}$$

where

$$|\bar{A}(q_L, q_R)| = \bar{A}^{+}(q_L, q_R) - \bar{A}^{-}(q_L, q_R) \tag{53}$$

and

$$\bar{A}^{\pm} = T\left(\frac{\Lambda \pm |\Lambda|}{2}\right) T^{-1} \tag{54}$$

An averaging process of the dependent variables is needed in order to make the $\bar{A}$ matrix satisfy Roe's requirements as cited in Chapter III.2.2. The Roe-averaged variables for the three-dimensional Euler equations incorporate information from either side of the cell interface as denoted by the subscripts L and R, (28,10)

$$\rho = (\rho_L \rho_R)^{1/2} \tag{55}$$

$$u = \frac{\rho_L^{1/2} u_L + \rho_R^{1/2} u_R}{\rho_L^{1/2} + \rho_R^{1/2}} \tag{56}$$

$$v = \frac{\rho_L^{1/2} v_L + \rho_R^{1/2} v_R}{\rho_L^{1/2} + \rho_R^{1/2}} \tag{57}$$

$$w = \frac{\rho_L^{1/2} w_L + \rho_R^{1/2} w_R}{\rho_L^{1/2} + \rho_R^{1/2}} \tag{58}$$

$$H = \frac{\rho_L^{1/2} H_L + \rho_R^{1/2} H_R}{\rho_L^{1/2} + \rho_R^{1/2}} \tag{59}$$

$$a^2 = \frac{\gamma P}{\rho} = (\gamma - 1)\left[H - \frac{u^2 + v^2 + w^2}{2}\right] \tag{60}$$

$$H = \frac{1}{\rho}(e+P) \tag{61}$$

$$e = \frac{P}{\gamma-1} + \frac{\rho}{2}(u^2+v^2+w^2) \tag{62}$$

where P is the pressure; ρ is the density; u, v, and w are the velocity components; e is the total energy; and H is the total enthalpy.

The one-dimensional analysis can be extended to three-dimensions and used with the transformed curvilinear-coordinate system following the appropriate transformations (Eqs. 14). The wave can be split into the three curvilinear-coordinate directions and the associated eigenvalues of the Euler equations determine the wave motions normal to the cell interfaces. The approximate Riemann solver is then applied at each cell interface (2:30).

III.3 Higher-Order Spatial Accuracy

The numerical solution in Eq. 52 yields a first-order accurate scheme, which we would like to improve to second- or third-order accuracy. We should simultaneously, however, maintain the desirable sharp, shock-capturing characteristics without introducing any pre-shock oscillations. A higher-order accurate, total-variation-diminishing (TVD) scheme using Roe-averaging was introduced by Osher and Chakravarthy and has been adapted for the

present work by Arabshahi (2:31). A thorough discussion of the implementation of these higher-order methods is in Reference 24. A brief discussion, however will be provided here.

A third-order accurate, numerical flux-vector at the cell interface, $i+1/2$, can be obtained by the addition of correction terms to the first-order accurate flux in Eq. 52. This calculation uses Roe-averaged variables and metric terms for the computation of eigenvalues and eigenvectors at cell interfaces. The numerical flux-vector is given as

$$f_{i+1/2} = [f(Q_i)]_{i+1/2} + \sum_{j=1}^{m} \sigma^{-}_{j,i+1/2}\, r^{j}_{i+1/2}$$

$$+ \sum_{j=1}^{m} \frac{1-\psi}{4}\left[L^{+}_{j}(-1,1) - L^{+}_{j}(3,1)\right] r^{j}_{i+1/2}$$

$$+ \sum_{j=1}^{m} \frac{1+\psi}{4}\left[L^{+}_{j}(-1,-1) - L^{-}_{j}(1,3)\right] r^{j}_{i+1/2} \qquad (63)$$

where

$$\sigma^{\pm}_{j,i+p/2} = \lambda^{\pm j}_{i+1/2}\, \alpha_{j,i+p/2} \qquad (64)$$

and

$$\alpha_{j,i-1/2} = \ell^{j}_{i+1/2} * (Q_i - Q_{i-1}) \qquad (65a)$$

$$\alpha_{j,i+1/2} = \ell^{j}_{i+1/2} * (Q_{i+1} - Q_i) \qquad (65b)$$

$$\alpha_{j,i+3/2} = \ell^{j}_{i+1/2} * (Q_{i+2} - Q_{i+1}) \qquad (65c)$$

When computing higher-order-accurate, upwind-biased schemes, overshoot and undershoot are expected in the vicinity of the shock. A limiter, L^{+}_{j}, such as the one in Eq. 63, can be implemented to reduce the scheme to a fully one-sided scheme for first-order accuracy in the shock region (2:31). This helps to eliminate the overshoot trait of higher-order schemes.

Within the Whitfield algorithm, three limiting operations are available. They are the minimum-modulus (minmod), superbee, and van Leer limiter. Each of these was used in the research.

The minmod limiter uses Eq. 63, where the limiting operator, $L^{\pm}_{j}$, is defined by

$$L^{\pm}_{j}(\ell,n) = minmod\ (\sigma^{\pm}_{j,i+\ell/2},\ b\sigma^{\pm}_{j,i+n/2}) \tag{66}$$

The operator "*minmod*" is given as

$$minmod[x,y] = sign(x) * \max(0,\ \min[|x|,\ y\ sign(x)]\ ) \tag{67}$$

The parameter, b, is called the compression parameter and is calculated from the function (28:16)

$$b = \frac{3-\psi}{1-\psi} \tag{68}$$

where ψ is the accuracy parameter. The minmod limiter contains two arguments, x and y. When the two arguments x, and y, are of opposite sign, the operator value is zero.

When the arguments are of the same sign, the operator chooses the argument with the smaller absolute value (2:33).

The subscript, i+1/2, in Eq. 63 indicates where the Roe-averaged variables and metrics are calculated. The eigenvalues, $\lambda^{\pm j}$, and left and right eigenvectors, ℓ^j and r^j, are evaluated using Roe-averaged variables. The flux-vectors, $f(Q_i)$ and $f(Q_{i+1})$, in Eq. 63 are computed using the dependent variables. The principle part of the truncation error of this higher-order scheme can be shown to be (2:32)

$$TE = -\frac{\left(\psi - \frac{1}{3}\right)}{4}(\Delta x)^2 \frac{\partial^3}{\partial x^3} f(q) \tag{69}$$

The accuracy of the scheme can be altered merely by varying the accuracy parameter, ψ. If $\psi = 1/3$, a TVD scheme of third-order accuracy results. A choice of $\psi = -1$ gives a second-order upwind TVD scheme (28:16).

The superbee limiter is another limiter used in this algorithm. The limiting operator, $L^{\pm}_j$, for this scheme is

$$L^{\pm}_j[\ell, n] = cmplim\left[\sigma^{\pm}_{j,i+\ell/2}, \sigma^{\pm}_{j,i+n/2}\right] \tag{70}$$

where

$$cmplim[x,y] = sign[x] *$$

$$\max\left(0, \min[|x|, \beta y\, sign\,(x)], \min[\beta|x|, y\, sign(x)]\right) \tag{71}$$

Here, β is the compression parameter. It is generally between $1 \leq \beta \leq 2$, and normally $\beta = 2$ is used (28:17).

The van Leer limiter is the third limiter used in the Whitfield algorithm. Here, the limiter is

$$L_j^{\pm}(\ell, n) = vanlim\left(\sigma_{j,i+\ell/2}^{\pm},\ \sigma_{j,i+n/2}^{\pm}\right) \tag{72}$$

where

$$vanlim\ [x,y] = \frac{xy + |xy|}{x+y} \tag{73}$$

The superbee and van Leer limiters are independent of the accuracy parameter, ψ.

The flux-difference scheme, with the limiters just discussed, is used to evaluate the residual on the right-hand-side of Eq. 37 in the Whitfield algorithm. The left-hand-side of Eq. 37 is calculated using the first-order, upwind, flux-vector splitting scheme. The accuracy of the solution is very good and the convergence rate for steady-state improves by combining the two schemes (2:35). When the algorithm is evaluating a dynamic problem the higher-order accuracy terms remain applicable. Further explanation of this algorithm can be found in Reference 2.

III.4 Boundary Conditions

The boundary conditions are very critical in the determination of the ultimate form of the flow-field.

Whitfield concluded the use of characteristic-variable boundary conditions produced the best results (2:36). The Euler equations must be put into characteristic form before determining the boundary conditions. The derivation of the characteristic form of the Euler equations can be found in Reference 29. There are three boundary conditions required for this investigation: subsonic inflow, outflow, and impermeable surface. Only the surface boundary condition is dependent on whether the grid is stationary or dynamic.

III.4.1 Subsonic Inflow

The subsonic inflow case is characterized by the first four eigenvalues being positive and the fifth is negative with flow in the increasing computational k-coordinate directions, ξ, η, or ζ. The first three and fifth eigenvalues are negative and the fourth is positive for flow in the decreasing computational k-coordinate direction (29:7). The subsonic inflow conditions are (29:8)

$$p_b = \frac{1}{2}\{p_a + p_1 \pm \rho_o c_o [\tilde{k}_x (u_a - u_1) + \tilde{k}_y (v_a - v_1) + \tilde{k}_z (w_a - w_1)]\} \quad (74a)$$

$$\rho_b = \rho_a + \frac{p_b - p_a}{c_o^2} \quad (74b)$$

$$u_b = u_a \pm \tilde{k}_x \frac{p_a - p_b}{\rho_o c_o} \quad (74c)$$

$$v_b = v_a \pm \tilde{k}_y \frac{p_a - p_b}{\rho_o c_o} \tag{74d}$$

$$w_b = w_a \pm \tilde{k}_z \frac{p_a - p_b}{\rho_o c_o} \tag{74e}$$

where

$$\tilde{k}_i = \frac{k_i}{(k_x^2 + k_y^2 + k_z^2)^{\frac{1}{2}}} \tag{75}$$

where i is x, y, or z, depending on which computational coordinate is being evaluated. The point a is outside the computational domain, point b is on the computational domain and point 1 is inside the computational domain (29:8). Figure 4 shows the relationship between these points.

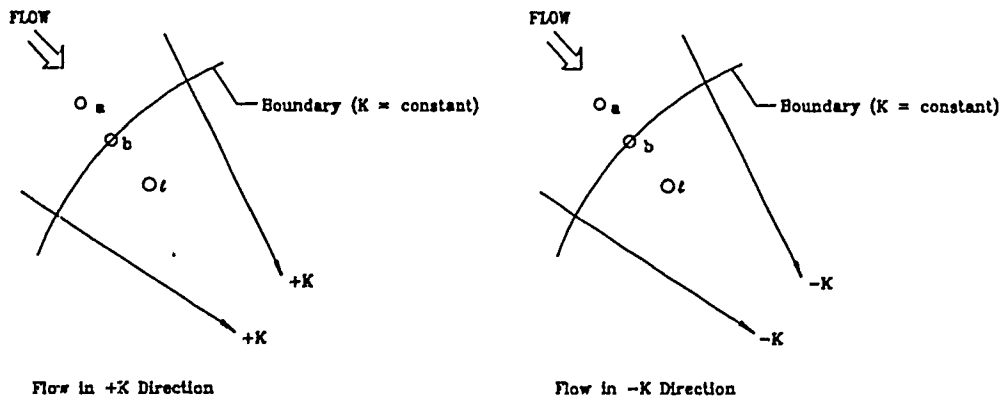


Figure 4: Computational-Coordinate Schematic for Boundary Conditions

The plus/minus sign option corresponds to the sign of the first three eigenvalues.

III.4.2 Subsonic Outflow

The subsonic outflow conditions are

$$p_b = p_1 \tag{76a}$$

$$\rho_b = \rho_a + \frac{p_b - p_a}{c_o^2} \tag{76b}$$

$$u_b = u_a \pm \tilde{k}_x \frac{p_a - p_b}{\rho_o c_o} \tag{76c}$$

$$v_b = v_a \pm \tilde{k}_y \frac{p_a - p_b}{\rho_o c_o} \tag{76d}$$

$$w_b = w_a \pm \tilde{k}_z \frac{p_a - p_b}{\rho_o c_o} \tag{76e}$$

where the point *a* is inside the computational domain and point *1* is outside the computational domain (29:9). The plus/minus signs have the same meaning as in Eqs. 74.

III.4.3 Impermeable Surface

The impermeable surface boundary condition is characterized by the first three eigenvalues being zero, the fourth is positive and the fifth is negative (13:43). The boundary conditions for a stationary, impermeable surface are

$$p_b = p_r \mp \rho_o c_o \left(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r\right) \tag{77a}$$

$$\rho_b = \rho_r + \frac{p_b - p_r}{c_o^2} \tag{77b}$$

$$u_b = u_r - \tilde{k}_x(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r) \tag{77c}$$

$$v_b = v_r - \tilde{k}_y(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r) \tag{77d}$$

$$w_b = w_r - \tilde{k}_z(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r) \tag{77e}$$

where the subscript r refers to the center of the first cell inside the boundary or the reference value. In Eq. 77a, the minus sign is used if r is in the positive k-direction from the boundary. The plus sign is used if r is in the negative k-direction from the boundary (29:9).

The dynamic impermeable surface condition varies for the stationary condition due to the addition of the time term. Eqs. 77 become (7:12)

$$\Delta p = p_b - p_r = \mp \rho_o c_o(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r + \tilde{k}_{t,b}) \tag{78a}$$

$$\rho_b = \rho_r - \frac{\Delta p}{c_o^2} \tag{78b}$$

$$u_b = u_r - \tilde{k}_x \frac{\Delta p}{\rho_o c_o} \tag{78c}$$

$$v_b = v_r - \tilde{k}_y \frac{\Delta p}{\rho_o c_o} \tag{78d}$$

$$w_b = w_r - \tilde{k}_z \frac{\Delta p}{\rho_o c_o} \tag{78e}$$

When the dynamic grid is implemented, the metrics and flux-Jacobians are normally updated after every time-step to account for the motion of the store.

The last concept to be addressed is phantom points. Phantom points can lie inside the boundary of the ellipse or outside the computational domain. Figure 5 shows the relationship between the boundaries, phantom points, and reference points.

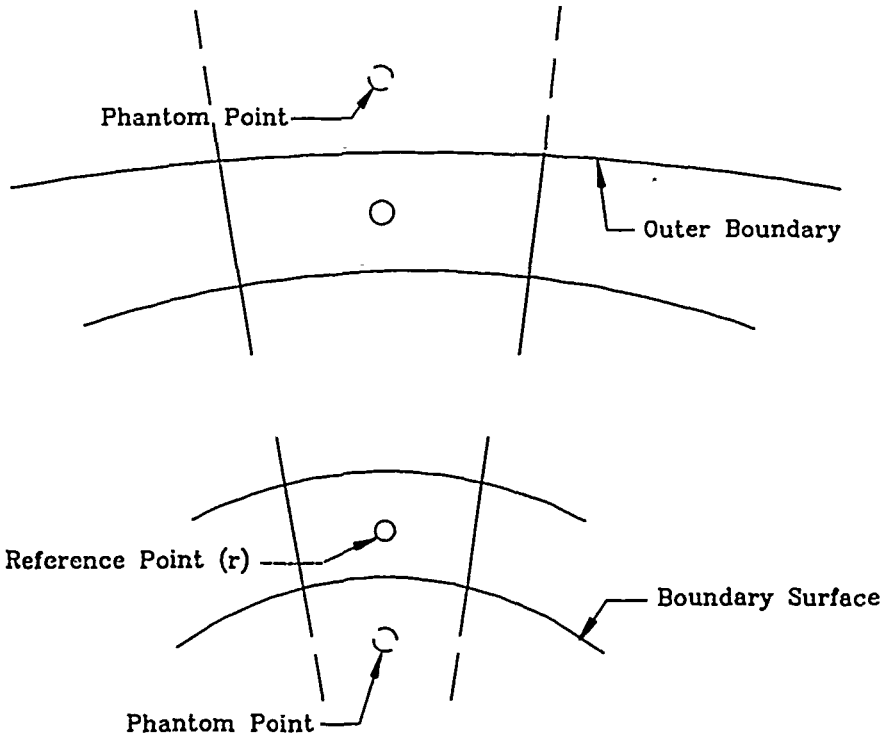


Figure 5: Reference and Phantom Point Schematic

Points *a* and *1* use the phantom point concept in the determination their values (13:45). The subscript *p* represents the phantom point and the values for the phantom points are calculated using

$$\theta_p = 2\theta_b - \theta_i \tag{79}$$

where θ can be p, ρ, u, v, or w. The i-subscript refers to the first cell inside the computational domain and can be a, l, or r depending on the boundary condition. For example, the phantom points for the impermeable surface would be calculated using (29:9)

$$p_p = p_r \mp 2\rho_o c_o \left(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r\right) \tag{80a}$$

$$\rho_p = \rho_r + \frac{2\left(p_b + p_r\right)}{c_o^2} \tag{80b}$$

$$u_p = u_r - 2\tilde{k}_x\left(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r\right) \tag{80c}$$

$$v_p = v_r - 2\tilde{k}_y\left(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r\right) \tag{80d}$$

$$w_p = w_r - 2\tilde{k}_z\left(\tilde{k}_x u_r + \tilde{k}_y v_r + \tilde{k}_z w_r\right) \tag{80e}$$

III.5 Planar Application of the Three-Dimensional Whitfield Algorithm

The three-dimensional Whitfield algorithm has been adapted for application to this planar problem. This was accomplished by generating a one-cell, three-dimensional grid, where the x- and y-locations of the grid points in each z-plane were identical. The two x-y planes where spaced a unit distance apart in the z-direction. The z-component of velocity, w, was also set to zero. This

allowed the ellipse to be treated as a two-dimensional body in the 3-DOF algorithm.

Chapter IV. Trajectory Algorithm and Grid Modification

IV.1 Equations of Motion

A 3-DOF model was added to the potential flow and Whitfield algorithms. The equations of motion for the ellipse are derived from Newton's second-law-of-motion, which states;

> A body acted upon by a force moves in such a manner that the time rate of change of momentum equals the force (11:90).

The rates of change are with respect to an inertial space. This law can be written in two vector equations;

$$\Sigma\bar{F} = \frac{d(m\bar{U})}{dt} \tag{81}$$

$$\Sigma\bar{M} = \frac{d\bar{\rho}}{dt} \tag{82}$$

where m is the mass of the body, $\bar{U}$ is the velocity vector, and $\bar{\rho}$ is the angular-momentum vector. Assuming that the earth is flat and that gyroscopic effects are negligible, Eqs. 81 and 82 reduce to a system of three equations of motion (10:32):

$$-m\frac{\partial V_{cog}}{\partial t} = D + W_t \sin\gamma \tag{83}$$

$$mV_{cog}\frac{\partial \gamma}{\partial t} = L - W_t \cos\gamma \tag{84}$$

$$I_z \frac{\partial^2 \theta}{\partial t^2} = M \tag{85}$$

where V_{cog} is the speed of the COG; D is the drag; L is the lift; M is the moment about the COG; W_t is the weight of the ellipse; I_z is the moment-of-inertia in the z-direction; γ is the angle between inertial x-axis and the velocity vector, $\vec{V}_{cog}$; and θ is the angle between the inertia axis and the body axis. The lift and drag used is calculated for the wind axis system which is parallel to the inertial axis system. Figure 6 shows the relationships between the different angles and the orientation of the ellipse between the inertial and body axes.

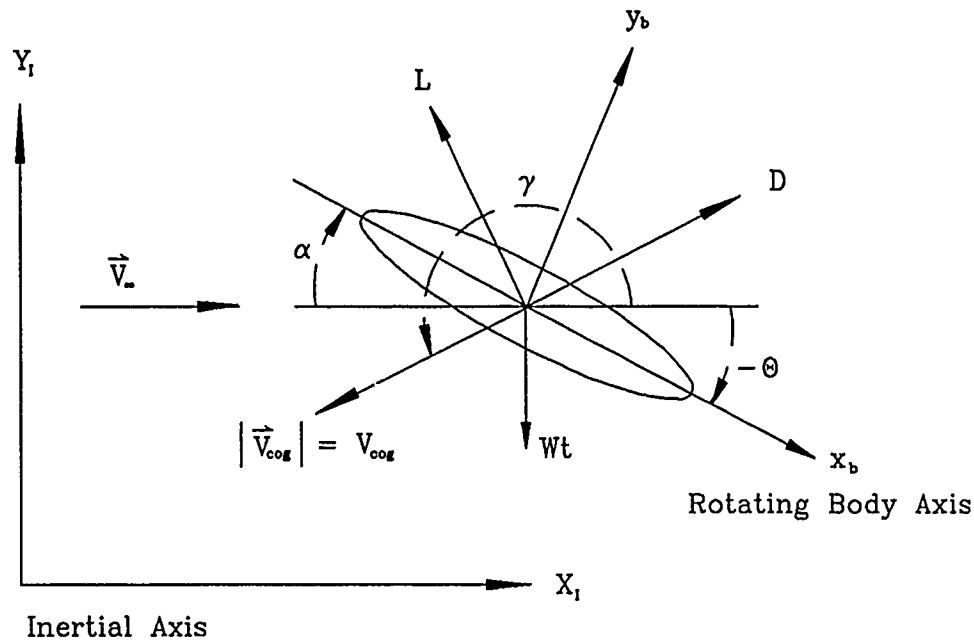


Figure 6: Orientation of Ellipse with 3-DOF Angles

θ and γ are measured positive counter-clockwise and α is measured positive clockwise.

Eqs. 83-85 represent an initial-value problem involving two first-order and one second-order nonlinear, ordinary-differential-equations. This problem can be reduced to three first-order, ordinary-differential-equations as shown in Gallaway by (10:33);

$$I_z \frac{\partial q}{\partial t} = M \tag{86}$$

where,

$$\frac{\partial \theta}{\partial t} = q \tag{87}$$

By using Eqs. 86 and 87, Eqs. 83-85 are altered to give,

$$\frac{\partial q}{\partial t} = \frac{M}{I_z} \tag{88}$$

$$\frac{\partial V_{cog}}{\partial t} = \frac{-(D + W_t \sin\gamma)}{m} \tag{89}$$

$$\frac{\partial \gamma}{\partial t} = \frac{(L - W_t \cos\gamma)}{mV_{cog}} \tag{90}$$

These three differential equations can be integrated using a variety of techniques. In Gallaway's dissertation, both Euler's explicit method and Adam-Bashforth formulae are used to solve Eqs. 88-90 (10:35). Pauletti's 6-DOF

algorithm incorporates a Runge-Kutta integration scheme to solve the equations of motion (20:22). A 6-DOF algorithm is utilized by the 3246th TW/TY to determine store trajectories using wind-tunnel data. This algorithm exercises either the Runge-Kutta or the Adam-Moulton integration formulae, which can be selected by the user (17:31).

Euler's explicit method was selected for use in this research for its simplicity. Euler's method is only first-order accurate. Gallaway found no significant difference in results using the Euler or Adam-Bashforth methods in his research (10:35). Euler's method leaves us with four equations (10:34)

$$V_{cog}^{n+1} = V_{cog}^{n} + \Delta t \frac{\partial V_{cog}^{n}}{\partial t} \tag{91}$$

$$\gamma^{n+1} = \gamma^{n} + \Delta t \frac{\partial \gamma}{\partial t}^{n} \tag{92}$$

$$q^{n+1} = q^{n} + \Delta t \frac{\partial q}{\partial t}^{n} \tag{93}$$

$$\theta^{n+1} = \theta^{n} + \Delta t \frac{\partial \theta}{\partial t}^{n} \tag{94}$$

Results from Eqs. 88-90 and Eqs. 86 and 87 are used to calculate V_{cog}^{n+1}, γ^{n+1}, q^{n+1}, and θ^{n+1}. Eq. 94 uses the results from Eq. 93 to calculate θ^{n+1}, utilizing the relationship in Eq. 87. Also, the following relationship is

used in the potential-flow algorithm to update the value of α at the new time-step (10:29).

$$\alpha = \frac{\pi}{2} - \tan^{-1}\left[\frac{V_\infty + V_{cog}|\cos\gamma|}{V_{cog}|\sin\gamma|}\right] \tag{95}$$

The relative velocity seen by the ellipse is the difference between the ellipse orientation angle, θ, and the velocity angle of the COG, γ. The new parameters calculated in Eqs. 91-95 are used in the next iteration to calculate the forces and moment on the ellipse.

The 3-DOF algorithm was developed, so the trajectory of the store is measured relative to an aircraft flying straight and level at constant velocity. The algorithm was developed in this way, so it could later be incorporated into a complete aircraft/store configuration. The initial conditions are constant velocity with no angular velocity. The store is then allowed to freefall.

For the quasi-analytical, potential-flow algorithm, the lift, drag, and moment values are calculated. They are then used in Eqs. 88-90 to calculate, $\partial q^n/\partial t$, $\partial \gamma^n/\partial t$, $\partial V^n_{cog}/\partial t$, and $\partial \theta^n/\partial t$. $\partial q^n/\partial t = \omega^n$ in Eq. 1. These results are used in Eqs. 91-93 to calculate V^{n+1}_{cog}, γ^{n+1}, and q^{n+1}. Using Eq. 87 and q^{n+1}, θ^{n+1} is calculated from Eq. 94. Eq. 95 is then used to calculate the value of α^{n+1}. V^{n+1}_{cog} and γ^{n+1} are used to compute the new location and orientation of the ellipse relative to the constant velocity aircraft. The process is repeated with the new values of α^{n+1}, V^{n+1}_{cog}, and

ω^{n+1} $(\partial q^{n+1}/\partial t)$ used in Eq. 1 to calculate the values of lift, drag, and moment at the next time-step.

The 3-DOF algorithm is similarly integrated into the implicit Euler algorithm. The lift, drag, and moment values are calculated by the implicit algorithm and V_{cog}^{n+1}, γ^{n+1}, and θ^{n+1} are calculated using the same above procedure. V_{cog}^{n+1}, γ^{n+1}, and θ^{n+1} are then used to re-orient the ellipse and the grid. The grid-manipulation procedure is discussed in depth in a later section.

IV.2 Center-of-Gravity and Moment-of-Inertia

The COG and moment-of-inertia calculations for the ellipse are required to create a stable trajectory. In general, a stable trajectory requires the COG to be forward of the center-of-pressure (26). The ellipse was assumed to be composed of two different materials having different densities. The materials are linearly distributed as shown in Figure 7.

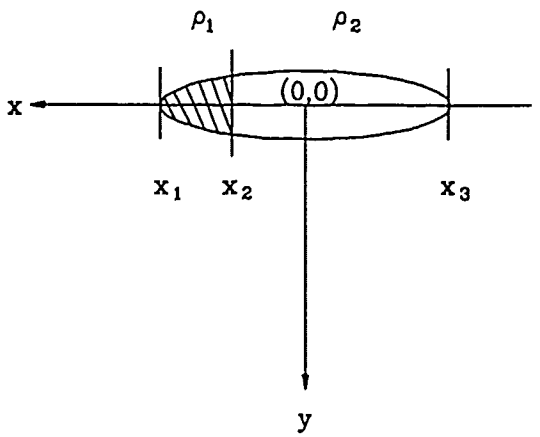


Figure 7: Mass Distribution

ρ_1 and ρ_2 are constants. This mass distribution keeps the COG in the y-axis constant at zero.

The COG was calculated using

$$x_{cg} = \frac{\int_{x_1}^{x_2} x\,dm_1 + \int_{x_2}^{x_3} x\,dm_2}{\int_{x_1}^{x_2} dm_1 + \int_{x_2}^{x_3} dm_2} \tag{96}$$

where

$$dm_1 = 2\rho_1 y(x)\,dx \tag{97}$$

$$dm_2 = 2\rho_2 y(x)\,dx \tag{98}$$

$y(x)$ is defined by the equation for an ellipse

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{99}$$

where a and b are the semi-major and semi-minor axes, respectively. Therefore, $y(x)$ is

$$y(x) = \left(b^2 - \frac{b^2x^2}{a^2}\right)^{\frac{1}{2}} \tag{100}$$

Eq. 96 can be integrated to determine the COG location, depending on the value of x_2 selected.

The moment-of-inertia is calculated using (11:204):

$$I_z = \int_{x_1}^{x_2}(x^2 + y(x)^2)\,dm_1 + \int_{x_2}^{x_3}(x^2 + y(x)^2)\,dm_2 \qquad (101)$$

The ellipse used in this research was composed of two materials with ρ_1 = 2700.0 kg/m^3, the density of aluminum, and ρ_1 = 11342.0 kg/m^3, the density of iron.

IV.3 Grid Manipulation

In this investigation, the grid is updated to reflect the ellipse's new location as it moves through an arbitrary trajectory. Grid manipulation was not required for the quasi-analytical, potential-flow algorithm, as explained in Chapter IV.2.

Two different grid-manipulation techniques were used. The first techniques moves the entire grid through the arbitrary trajectory calculated by the 3-DOF algorithm. This is the same procedure used in References 2, 6, and 8. Figure 8 shows an exaggerated example of how the entire grid translates with the ellipse. The first technique will be referred for the remainder of the text as Grid 1.

The second technique utilizes an elliptical grid-generator in determining the new grid after each time-step. The grid-generation algorithm was developed by AFWAL/FIMM (David J. Amdahl) and has been modified to run on the STELLAR ST-2000 by Capt Mark Driver. The program uses Thompson's techniques for generating grids describe in

Hoffman using Laplace's equation (12:243-305). The new location of the ellipse can be determined from the V_{cog}, γ, and θ, generated by the 3-DOF algorithm.

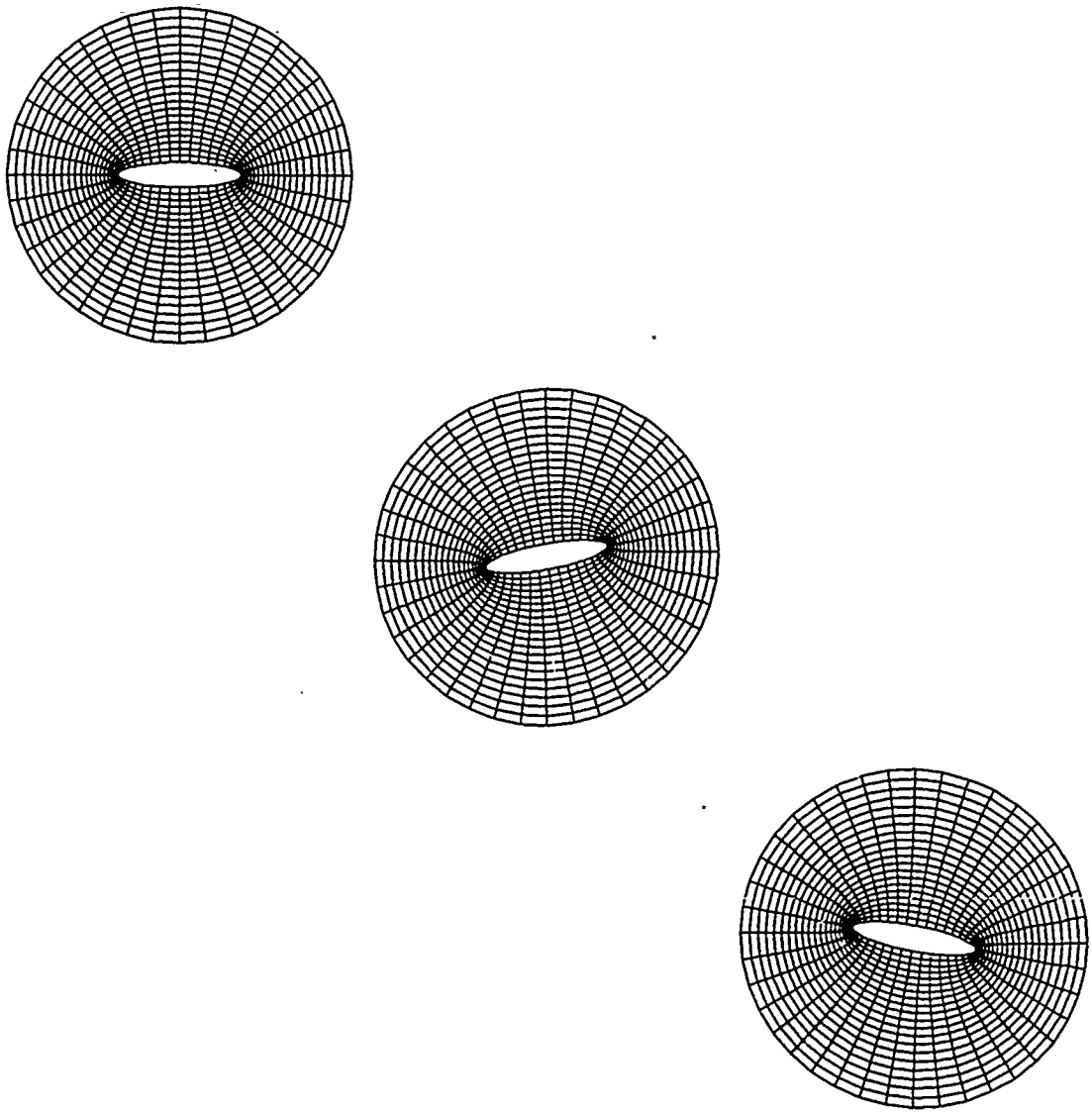
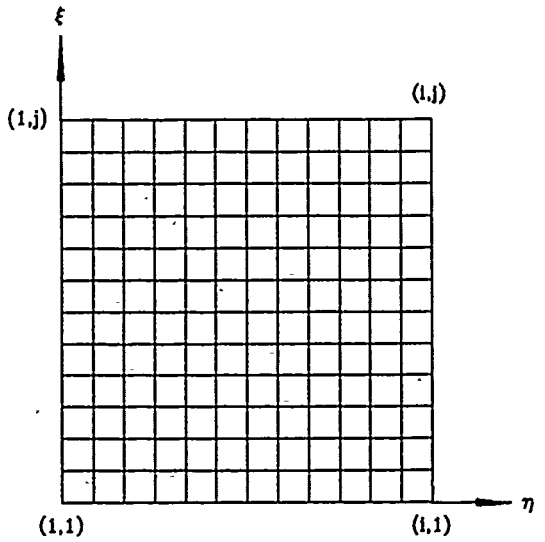


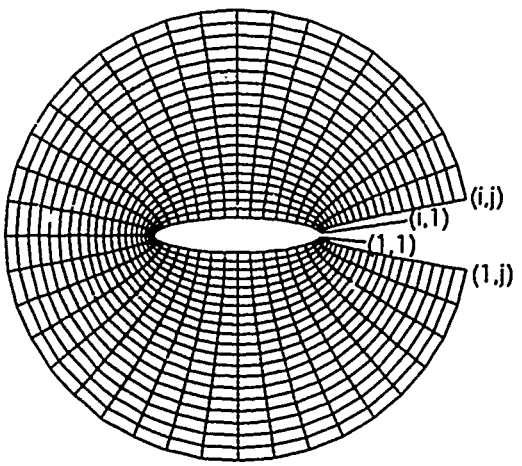
Figure 8: Example of Grid 1 Trajectory

The ellipse is moved the distance calculated within the grid. The outer boundary of the grid remains stationary and the branch-cut in the O-grid is moved using a geometric progression routine. The branch-cut in the O-grid is where

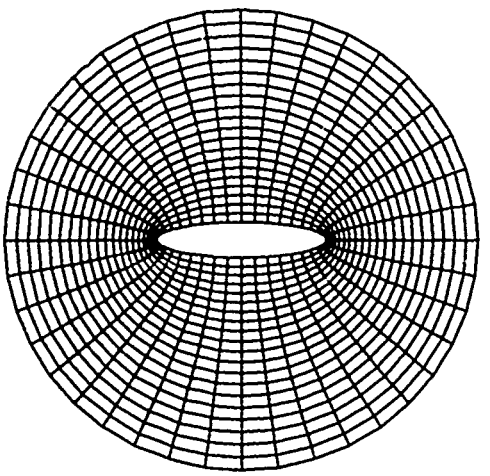
the left and right-hand-sides of the computational grid (Figure 9a) come together in the physical plane (Figure 9b) to form a continuous grid (Figure 9c).



(a)



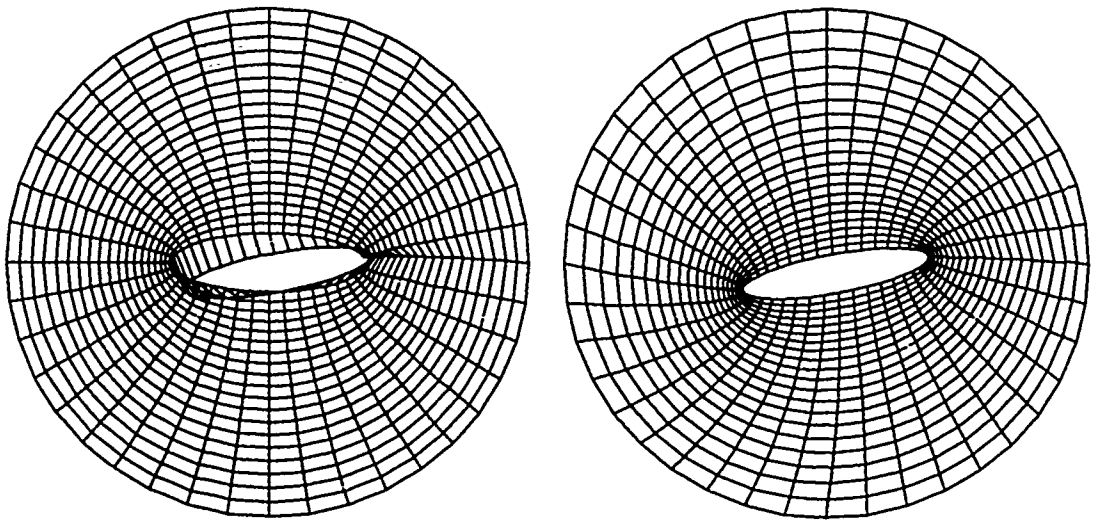
(b)



(c)

Figure 9: Conversion of Computational Grid to Physical Grid

Once, the new location of the ellipse and branch-cut are calculated the elliptic grid-generation algorithm regenerates the grid. The ellipse starts in the position similar to that in Figure 9c. Figure 10a shows an example of the ellipse and branch-cut moved as a result of the 3-DOF algorithm. Figure 10b shows the grid after it has been modified by the elliptic grid-generation algorithm. This grid-manipulation technique will be referred to as Grid 2 for the remainder of the text. The discontinuity in the cells on either side of the branch-cut is a result of the elliptic grid-generation algorithm. The discontinuity could not be removed with the grid-generation algorithm.



(a) (b)

Figure 10: Generation Procedure for Grid 2

By using the elliptic grid-generator, the uniform outer boundary can remain stationary. This allows for the

utilization of a block grid-structure in the CFD algorithm to support an aircraft/store separation problems. The Whitfield algorithm was developed for complex blocked grid-structures (8:32). The algorithm allows for the generation of a grid around a specific aircraft with "holes" in the grid where stores could be added to complete the physical domain. The only requirement is the location of the grid points along the block boundary must be coincident. A grid-generation routine similar to the Grid 2 technique could be used to keep the block-grid boundary stationary; yet still move the store through an arbitrary trajectory. Belk describes the block-grid technique in his dissertation (8:32-53).

Chapter V: Results

This section presents the data gathered for the quasi-analytical, potential-flow algorithm and two grid-modification cases. The preliminary investigation discusses the initial analyses used in the determination of the Mach number and limiter used in the research. From this investigation, no limiter was utilized in the actual trajectory calculations. Mach = 0.3 was selected for use in the implicit Euler algorithm. A comparison between the trajectory trends of the potential flow and implicit algorithm, for Grid 1 and 2, was performed. In all cases, the ellipse pitched nose-down and then begin to oscillate with varying periods. The variations depended on time-step used and the algorithm and grid type. The robustness of the algorithm and solution accuracy were also investigated in light of time-step variation and changes in the Jacobian-update methodology. Time-step variation influenced the pitching motion of the ellipse more than the trajectory of the COG. The Jacobian-update methodology, however, did not effect the COG trajectory or pitching motion of the ellipse.

V.1 Preliminary Analysis

The analysis began with a comparison of the coefficient of drag, C_d, values versus grid refinement for a stationary ellipse. An exact solution to the Euler equations would result in no C_l or C_d calculated about a symmetric body.

Any variation is a result of either numerical viscosity or numerical error introduced by the algorithm. In the initial results, negative C_d's occurred using the limiters in the Whitfield algorithm. C_d was used in determining the grid refinement necessary in generating accurate results. The initial, grid-refinement investigation used the Van Leer limiter in calculating the C_d. The maximum grid-refinement was a 201x51 grid with a $\psi = -1$. The C_d began to decrease through zero as shown in Figure 11.

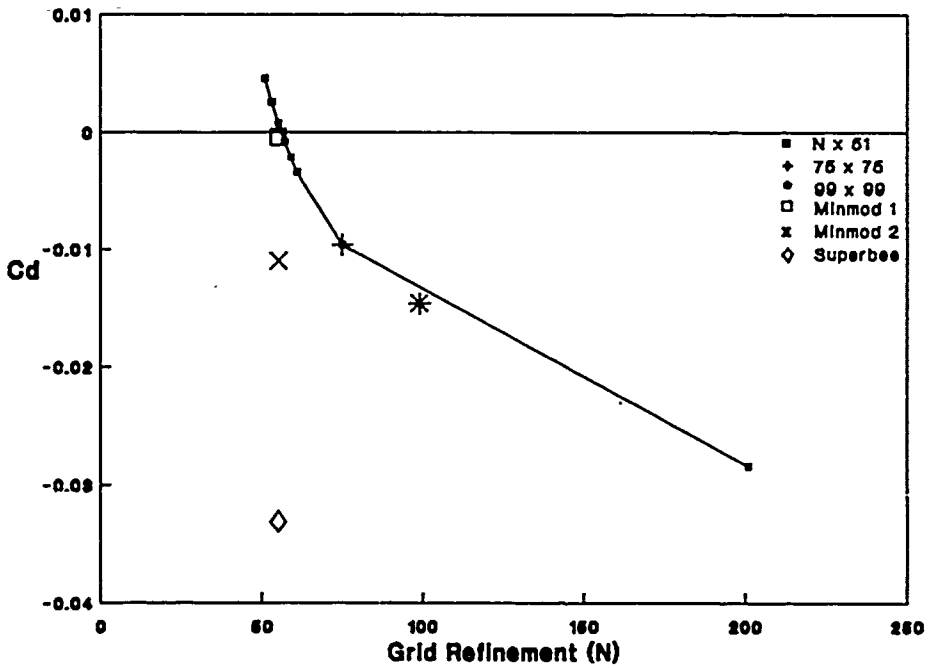


Figure 11: Cd vs Grid Refinement (with Limiter)

The minmod and superbee limiters were also investigated attempting to resolve the negative C_d problem. The minmod limiter generated a second order (MINMOD1, $\psi = -1$) and third order (MINMOD2, $\psi = 1/3$) solution. The superbee limiter

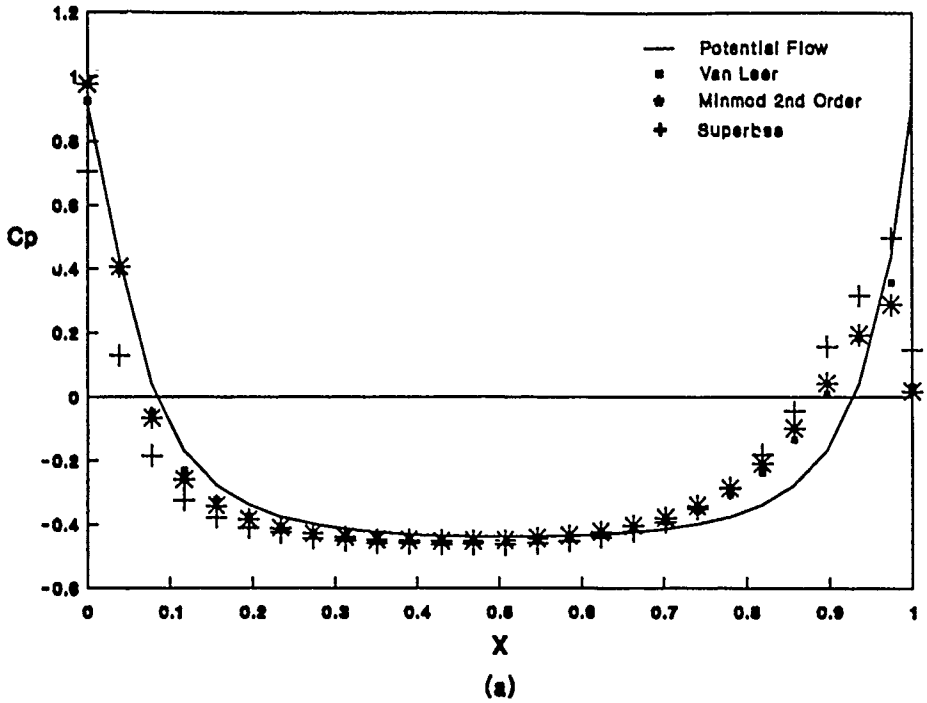
generated a second-order solution with a $\psi = 0$. Figure 11 also shows these C_d values.

Coefficient of pressure, C_p, curves are shown for the 55x51 grid. Figure 12 shows the variations in C_p with each limiter. The potential flow and no limiter case are shown for comparison.

The specific reason for these negative drag values cannot be explained by this author. A numerical error is added to the implicit algorithm by these limiters. The numerical error negatively impacts the C_p on the ellipse, which is used in the calculation of coefficient of lift, C_l and C_d. These results are not reasonable compared to what nature tells us about drag. Therefore, limiters were not used in the trajectory investigation.

Pulliam hints at the negative drag results in his AIAA article (Reference 21), but he also has no specific explanation for its occurrence (22). An explanation for the negative drag coefficient and the non-zero lift solutions as discussed by Pulliam requires further research. Multiple telephone conversations with the developers of the implicit algorithm did not bring to light any additional information into the reason for the negative drag values (3,23).

By using no limiter in the implicit algorithm, the negative C_d problem is alleviated. Figure 13 shows the C_d versus grid-refinement plot. A 201x201x2 grid was initially chosen for the research. These data was calculated using the Cray 2 supercomputer.



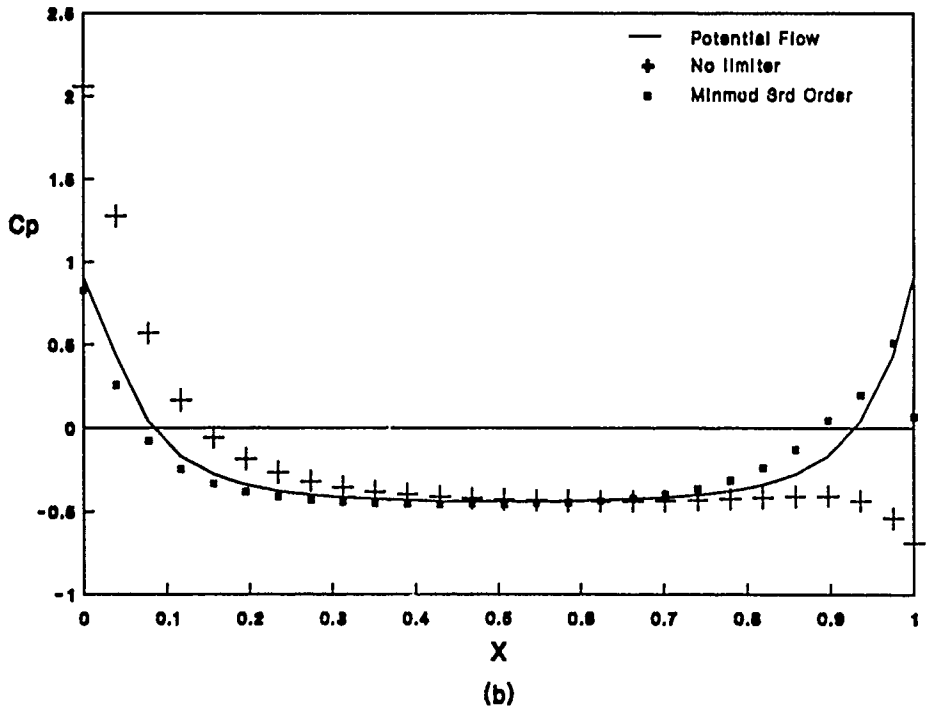


Figure 12: Cp vs X (For Different Limiter)

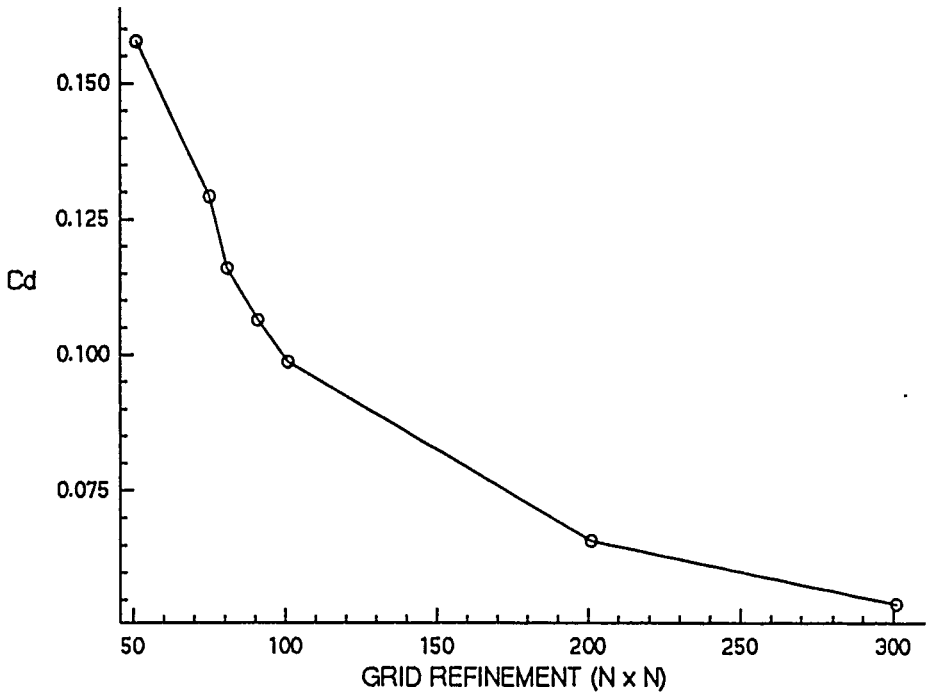


Figure 13: Cd vs Grid Refinement (No Limiter)

By evaluating the coefficient of pressure, C_p, plots for the ellipse, the Mach number used in the research was determined. From these data, the Mach number chosen for the research was 0.3. This Mach number was selected because of its relatively smooth C_p curve. Mach = 0.3 is also within the Mach range where flow can still be considered incompressible. Therefore, the implicit algorithm results should more closely match those of quasi-analytical, potential-flow results. Appendix B details these results.

A problem with computer availability required the remainder of the research to be performed on the Stellar ST-2000 computer. Memory requirements inherent to the implicit algorithm required the grid refinement to be reduced to 101x101x2.

V.1.1 Preliminary Grid Analysis

Two different grids were used during the evaluation of the implicit algorithm. Both grids have an outer radius of 10 chords. Grid 1 is a simple O-grid generated using Eqs. 3. Grid 2 was generated by taking Grid 1 and manipulating it using the elliptic grid-generator discussed in Chapter IV.3. The steady-state solution for Grid 1 and Grid 2 was calculated using the implicit algorithm. The ρ-residual (R^n) converged to a value of 1.375×10^{-8} for Grid 1 and 2.579×10^{-8} for Grid 2, the asymptotic, steady-state value. This was the starting point for the trajectories gathered in this research. Figures 14 and 15 show a close-up of Grid 1 and Grid 2 near the ellipse.

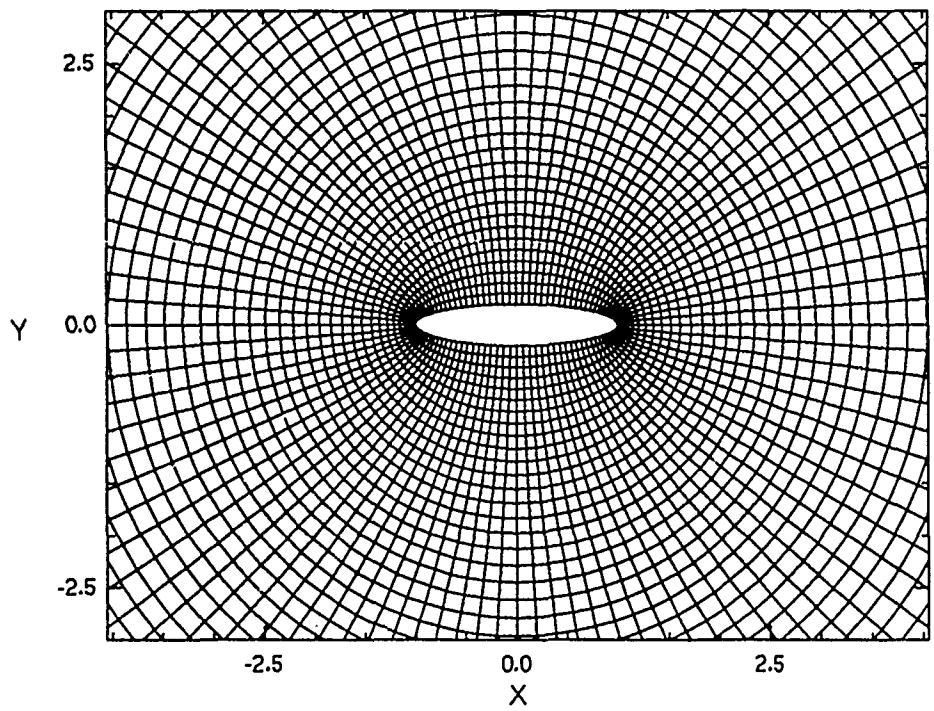


Figure 14: Grid 1, 101 x 101

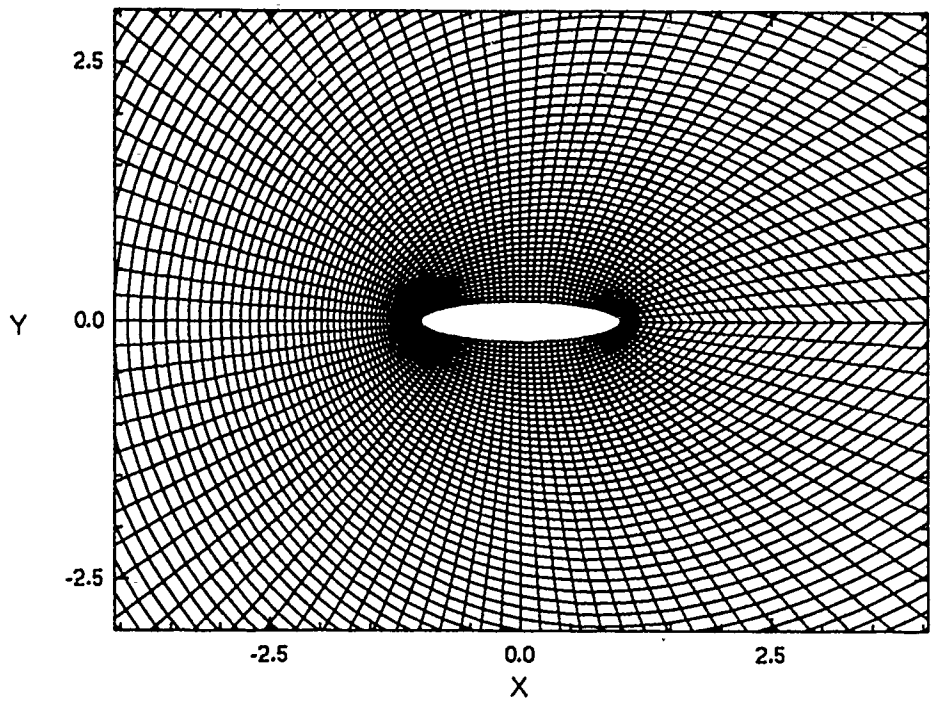


Figure 15: Grid 2, 101 x 101

The implicit solutions from both grids gave slightly different values for C_d and C_l. The ellipse in Grid 1 had a C_l = 0.0000 and a C_d = 0.1006, whereas the ellipse in Grid 2 had a C_l = -0.0032 and a C_d = 0.0958 for zero angle of attack. This variation in C_l and C_d matches Pulliam's findings for the solution of the Euler equations for subsonic flow past ellipses (21). Pulliam found minor variations in the grid refinement and algorithm type resulted in a "general lack of consistency in the results" (21:1). A numerical error is being introduced, however, the specifics are not clearly understood (22).

Figure 16 shows the C_p for both grids along the upper surface. The flow conditions are Mach = 0.3 and $\alpha = 0°$. C_p on the lower surface is graphically identical with the upper

surface; therefore plotting it would be redundant. The variation in C_p at the nose and tail between Grid 1 and Grid 2 accounts for the difference in C_d. The variation in C_l can't be graphically determined from the C_p plot.

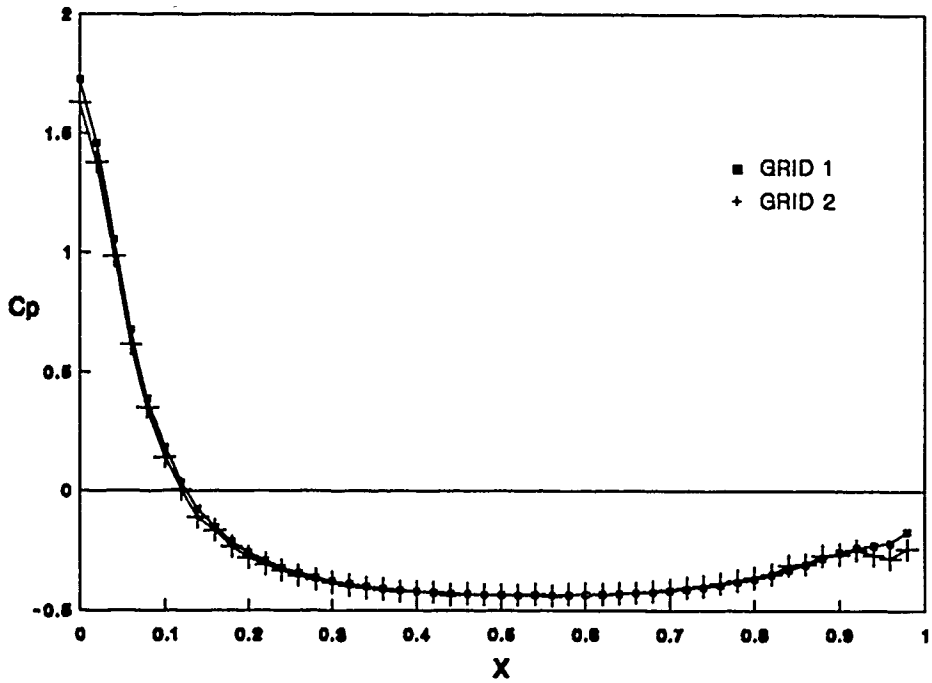


Figure 16: Cp vs X (Upper Surface)

Figure 17 and 18 show the pressure contours for Grid 1 and 2 at the same flow conditions as in Figure 16. The contour lines in both graphs are scaled to the same minimum and maximum dimensionless pressure, 0.6951 and 0.7558, respectively. These are the actual minimum and maximum pressures calculated for Grid 1.

By close inspection, one can see the higher pressure along the upper surface, thus creating the negative lift. The actual minimum and maximum pressures for Grid 2 are 0.6939 and 0.7693. The pressure range for Grid 2 is larger

than that of Grid 1. Pulliam observed similar pressure contour variations when different grids were used (21:1).

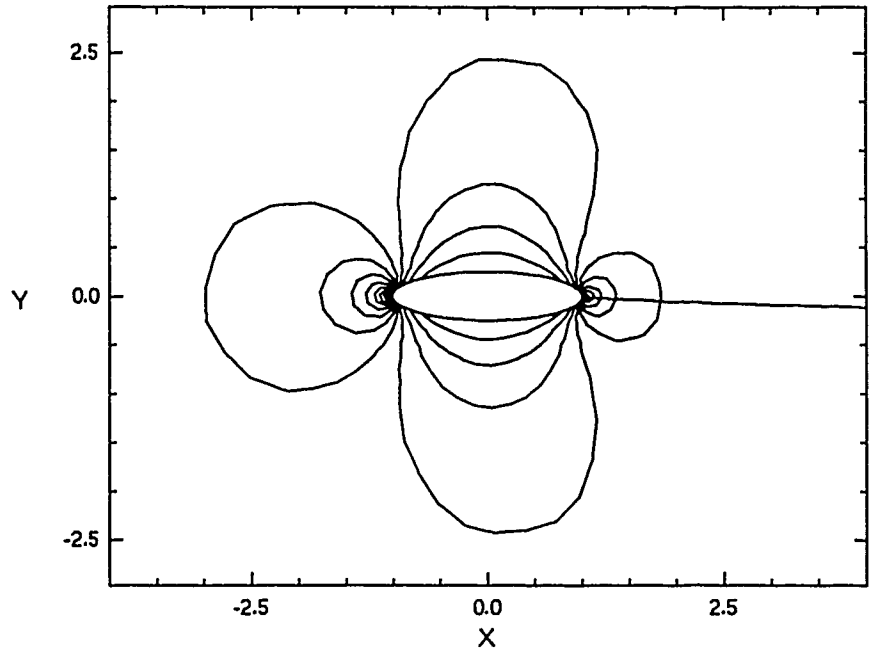


Figure 17: Pressure Contours for Grid 1

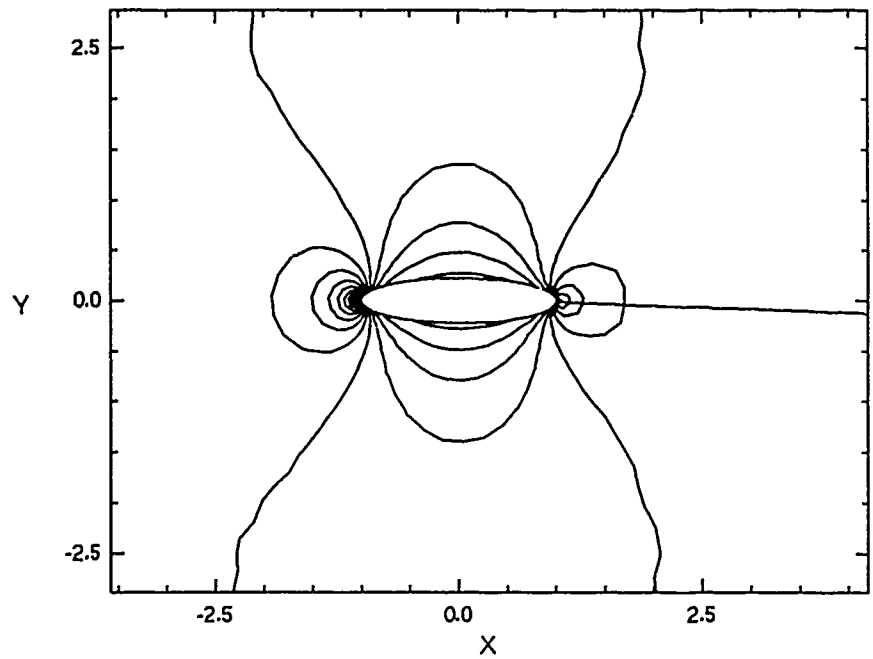


Figure 18: Pressure Contours for Grid 2

V.2 Trajectory Analysis

The trajectories for Grid 1 and 2 were compared at different time-steps and with different Jacobian-update methodologies. The initial lift, drag, and moment on the ellipse was calculated from the steady-state solution. The ellipse was then allowed to freefall through an arbitrary trajectory as calculated by the 3-DOF algorithm. The 3-DOF algorithm calculates the new position relative to an aircraft flying straight and level at constant velocity. The Δt used in the 3-DOF algorithm for both grids was calculated by using a Courant number of 10 and Eq. 102.

$$\Delta t = \frac{CFL\,(\Delta x)\,(\Delta y)\,(\Delta z)}{\max|\lambda_k'|} \tag{102}$$

The minimum time-step for Grid 1 was $\Delta t = 0.0005$ and $\Delta t = 0.0003$ for Grid 2. The varying geometries between Grid 1 and 2 caused the values of Δt to be different. The 3-DOF algorithm used the C_l, C_d, and moment coefficient, C_m, calculated after each time-step to determine the new location of the ellipse. The trajectory calculation was limited to 1000 iterations due to computer resources. This moved the ellipse approximately 2 diameters from its original location. This would be inadequate in moving a store out of the influence of an aircraft.

Three different cases were investigated, each with different initial, flow-field conditions and ellipse mass-properties. Table 1 shows these three different cases.

TABLE 1: Initial Flow Conditions and Mass-Properties

CASE:	Mach	α	X2	Xcog	Mass*	Iz**
1	0.3	0.0°	1.0	0.000	1696.5	475.0
2	0.3	0.0°	0.3	0.295	3390.2	1165.3
3	0.3	-5.0°	0.3	0.295	3390.2	1165.3

All lengths are dimensionless
* (kg)
** ($kg\text{-}m^2$)

First, the trajectory of the ellipse COG for the potential-flow solution and Grids 1 and 2 are compared using the Case 1 properties. Δt = 0.0005 was also used in the potential-flow solution. Figure 19 shows the COG dropping straight down for the potential-flow solution since, the ellipse generates no lift or drag. The small error between the potential flow and implicit algorithm COG trajectories is cause by the numerical error in the implicit algorithm, which calculates drag on the symmetric ellipse. Grid 1 and 2 show close agreement through the 1000 iteration COG trajectory. However, the original assumption was to compared the pitch angle of the ellipses between the two algorithms and not the x- and y-location. Figure 20 shows the pitch angle, θ, versus the y-COG location.

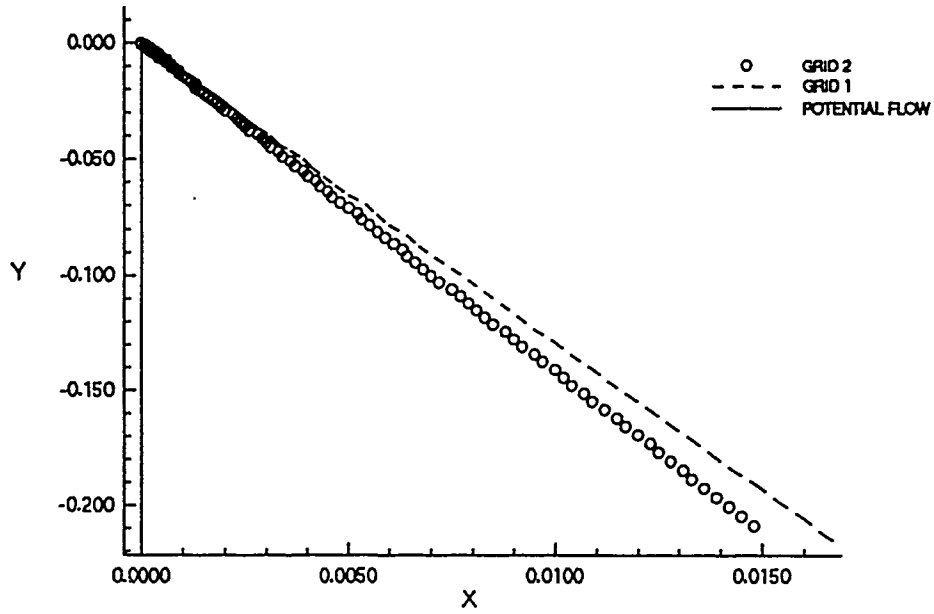


Figure 19: COG Location vs Solution Type (Case 1)

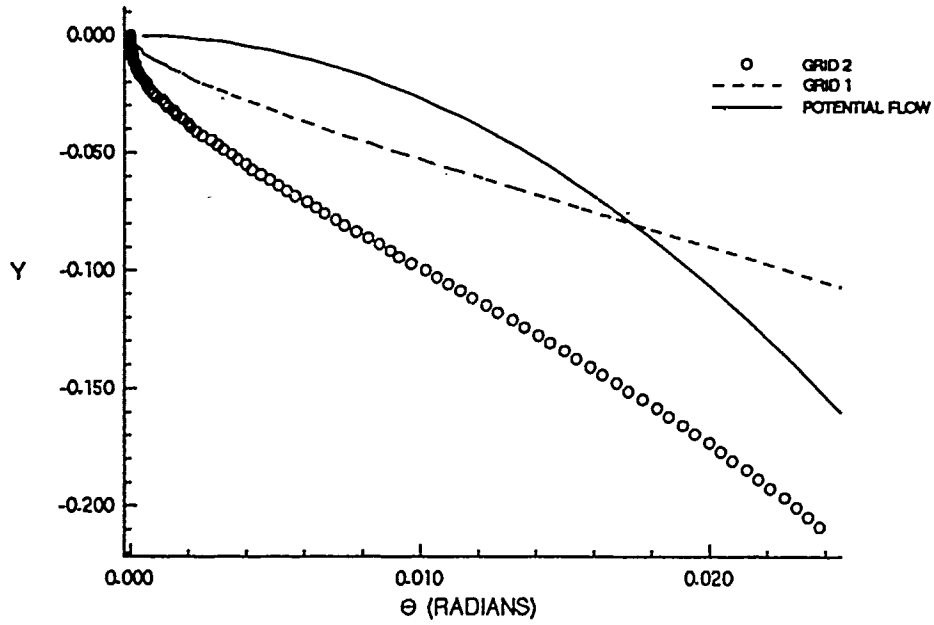


Figure 20: θ vs y-COG (Case 1)

The general trend of θ for Grid 1 and 2 is the same. The difference in magnitude can be accounted for due to the

different initial C_l and C_d values between Grid 1 and 2. The trend for the potential-flow solution shows a different initial slope than the implicit algorithm results. This variation in θ could be a result of the initial properties of the flow-field and the ellipse. The initial conditions in Case 1 are "zero" conditions, where no lift, drag, or moment is generated on the ellipse from the potential-flow algorithm. Any small error could introduce large variations in the final results.

The ρ-residual jumps from the steady-state value (10^{-8}) to a value on the order of 10^{-4} due to the initial movement for both grids. The residual then gradually decreases down to a value on the order 10^{-5} through the 1000 iterations. This trend agrees with results obtained by Simpson (23). By observing the behavior of the residual, one can access the likelihood that the implicit algorithm is generating a reasonable result. As long as the residual remains small, a value on the order of 10^{-4}, the algorithm results should be reasonable.

Case 2 is the first attempt at alleviating the discrepancy in the θ-trend between the potential and implicit algorithm results. Case 2 simply moves the COG forward of the center-of-pressure, therefore producing a more stable store. Figure 21 shows the θ versus y-COG trajectory for the Case 2 conditions.

Case 2 initial θ-trends between the potential flow and Grid 1 vary the same as they do in Case 1. The initial

slope of the θ versus y-COG for the potential-flow result is approximately -0.001; whereas the initial slope for Grid 1 is approximately -146.7. The "zero" initial conditions again could cause the variation in the trends between potential flow and Grid 1.

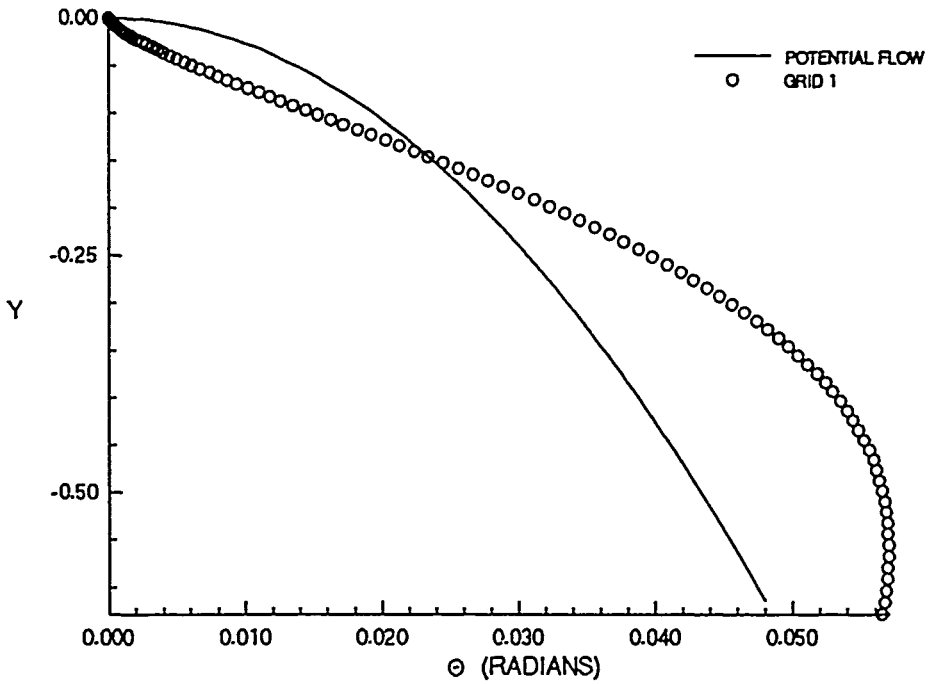


Figure 21: θ vs y-COG (Case 2)

Case 3 is an attempt at removing the "zero" initial conditions from the problem. In this case, we use the same store mass-properties as in Case 2, but start the ellipse at an $\alpha = -5°$. Figure 22 shows the variation in θ versus y-COG through a 1000 iterations. The ellipse continues to pitch down in the potential-flow case, while the ellipse in Grid 1 is trying to stabilize. Case 3 indicates there are large differences between the potential flow and Euler implicit algorithm results, which make applicable comparisons very

difficult. The reason for the Case 3 irregularity is unknown.

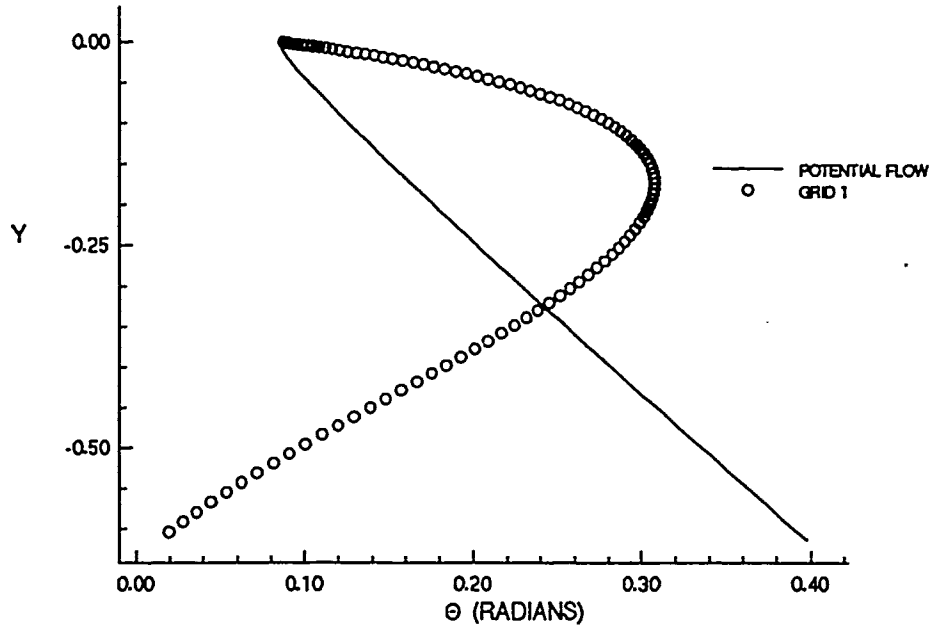


Figure 22: θ vs Y-COG (Case 3)

V.3 Implicit Algorithm Investigation

The effects of multiplying the minimum time-step, Δt, by 10 and 100 were investigated for the implicit algorithm. The purpose is to evaluate the applicability of using larger time-steps in the implicit algorithm in generating a solution. Figure 23 shows the COG trajectory results and Figure 24 shows θ versus y-COG for Grid 1 with Case 1 initial conditions.

The minimum time-step, Δt, and the Δtx10 COG trajectories compared well for Grid 1, however the Δtx100 results are extremely poor. The minimum time-step may be too large for the 3-DOF routine, since not even the first

iteration has the same slope as the other results.

Typically, 6-DOF routines (Reference 17) utilize a time-step between Δt = 0.005 to 0.01 for store separation (26). The time-step for the Δtx100 case is larger than a typical 6-DOF time-step. The minimum time-steps of 0.0003 and 0.0005 do not negatively impact the 3-DOF routine, except in requiring more iterations to reach the solution.

The trend for θ in Figure 24 matches well for the Δt and Δtx10 results, however the Δtx100 results are extremely poor. The larger period and amplitude of the Δtx10 result compared to the Δt results, can be accounted due to the larger time-step. This larger time-step is than multiplied by the lift and drag values calculated by the implicit algorithm solution causing the larger pitching amplitude.

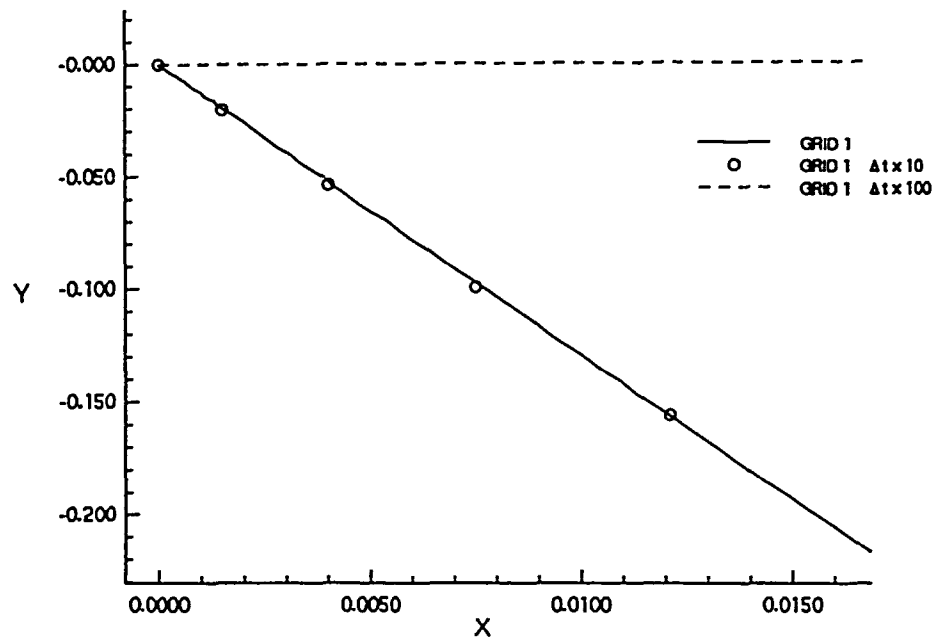


Figure 23: Grid 1 Trajectories at Δt, Δtx10, Δtx100 (Case 1)

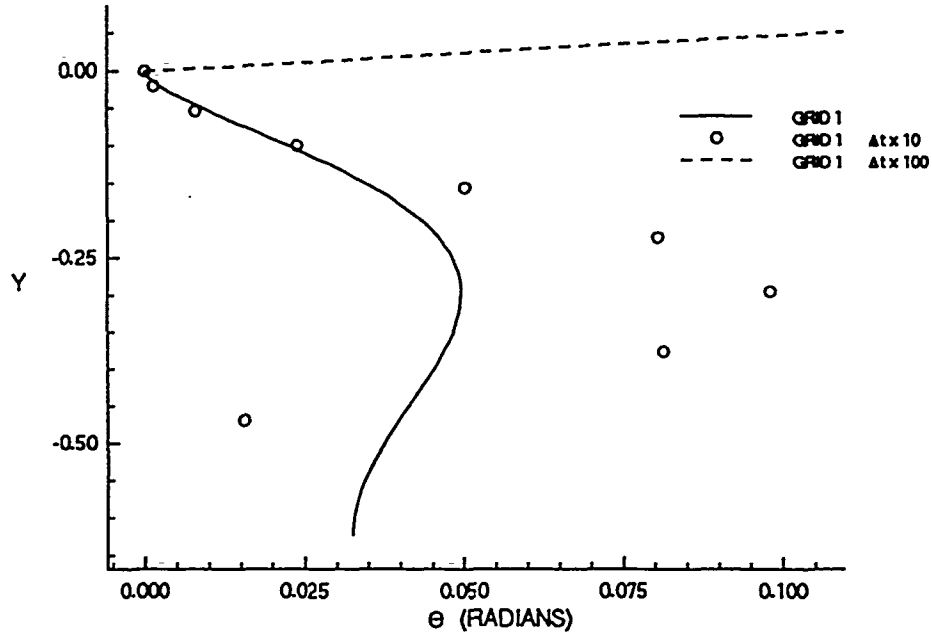


Figure 24: θ vs y-COG at Δt, Δtx10, Δtx100 for Grid 1 (Case 1)

The Δt was varied further to investigate its affect on the trajectory. Figure 25 shows the trajectories for Δtx2, Δtx4, Δtx6, and Δtx8. The results shows that the magnitude of the Δt directly affects the magnitude of the oscillation of the ellipse. Since, the "correct" trajectory is unknown, determining a valid time-step is not possible. However, the potential-flow trajectory of Δtx10 matches the original Δt trajectory, as shown in Figure 26. This is an indication that the 3-DOF algorithm is calculating valid solutions at the larger time-steps.

The ρ-residual trend for the Δtx2, 4, 6, 8, and 10 cases matched the minimum time-step trend addressed earlier in this section. The ρ-residual for the Δtx100 case jumped from the steady-state case to a value on the order 10^{-1} and

remained at this level. This is an indication that the Δtx100 time-step is too large for the implicit algorithm (23).

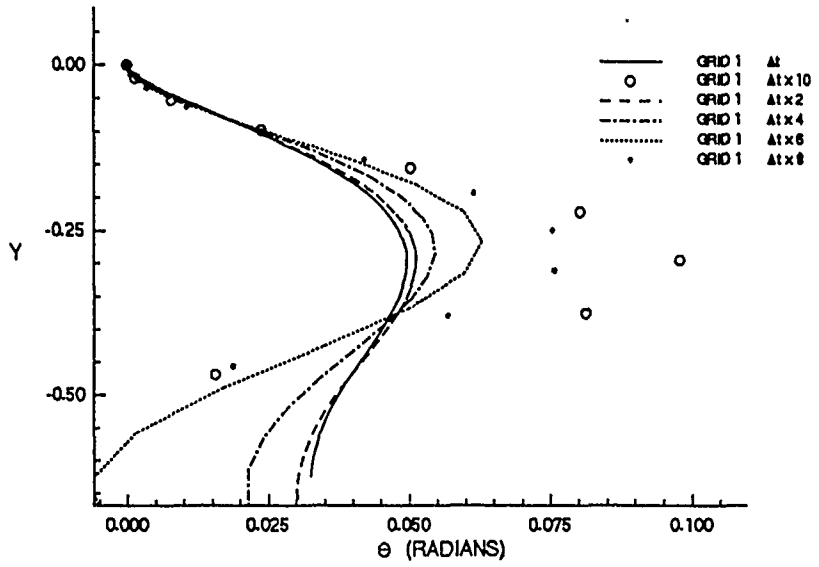


Figure 25: θ vs y-COG for Varying Δt's (Case 1)

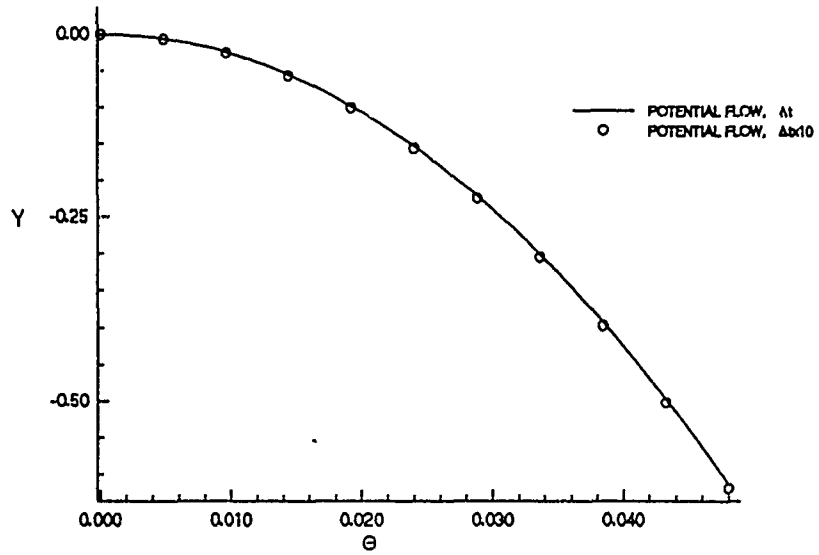


Figure 26: θ vs y-COG for the Potential Flow Solution for Δt and Δtx10 (Case 1)

The results of the Grid 1, Δtx100 case indicated the investigation of the Δtx100 case for Grid 2 would not be necessary and was not examined. Figure 27 displays COG trajectories for Grid 2, Case 1 results; there is good agreement between the Δt and the Δtx10 solutions. Figure 28 show θ vs y-COG at Δt and Δtx10 for Grid 2, Case 1. The initial θ-trend is the same as the Grid 1 results. The Δtx10 solution allows for a faster determination of the COG trajectory.

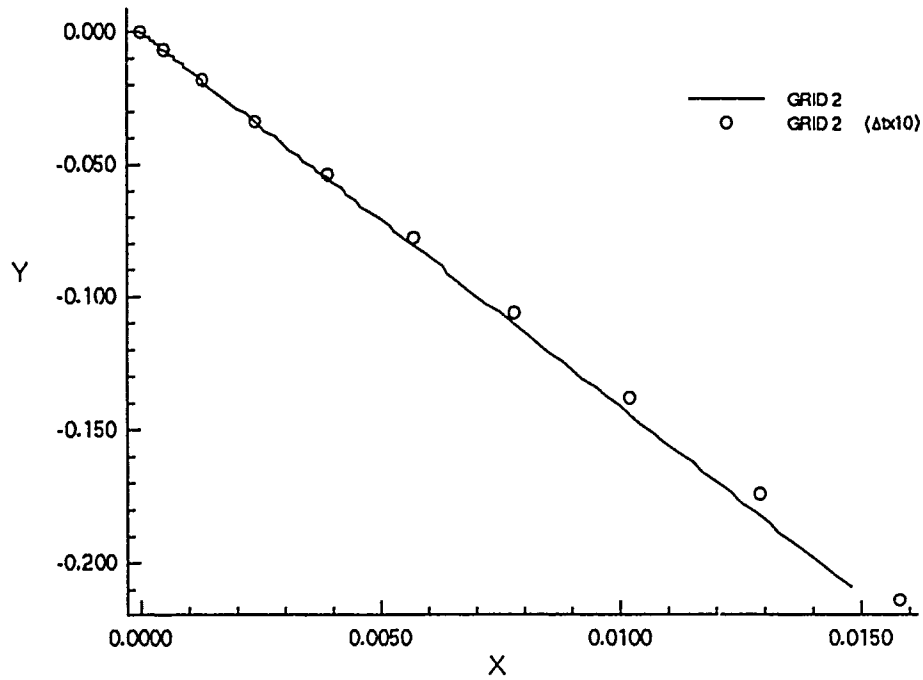


Figure 27: Grid 2 Trajectory with Δt and Δtx10 (Case 1)

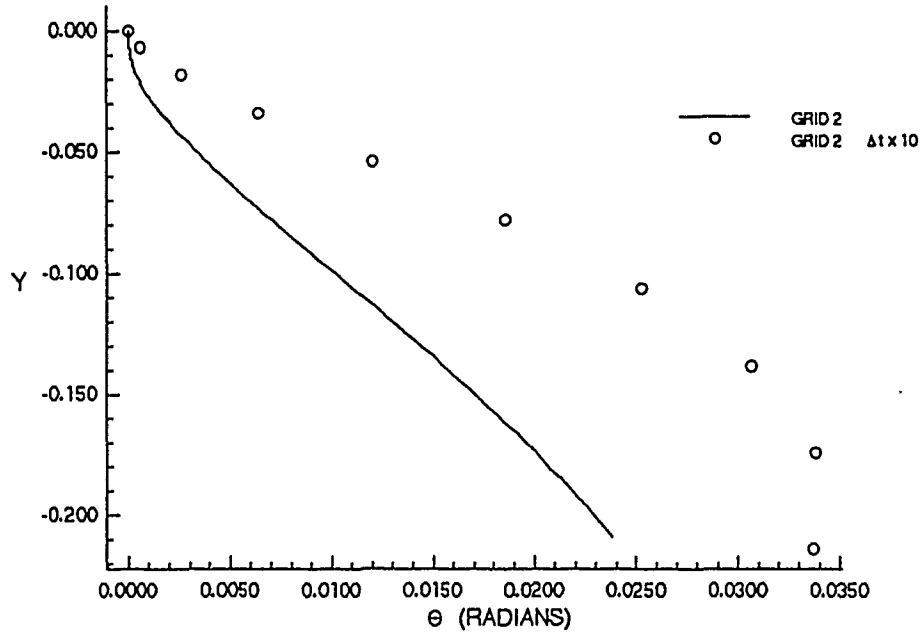


Figure 28: θ vs y-COG at Δt and Δtx10 for Grid 2 (Case 1)

The final phase of the investigation involved the examination of alternative flux-Jacobian update methodologies. In work done by Belk and Arabshabi (References 2 and 8), the flux-Jacobians were updated after every iteration during the dynamic-grid calculations. Simpson found it wasn't necessary to update the flux-Jacobians after every time-step (23). He found that by calculating the flux-Jacobians once, they could be used throughout a trajectory. Simpson had also unsuccessfully investigated updating the flux-Jacobians at varying intervals during the trajectory. It was recommended for this research the flux-Jacobians be calculated once before the trajectory begins and frozen through the trajectory. The computational time decreased by approximately 45% by

using this update methodology. Grid 1 and 2 were evaluated with the flux-Jacobians calculated once and frozen throughout a trajectory (1000 iterations). Figure 29 compares the COG trajectory for the two, Jacobian-update methodologies. The results match the trends found by Simpson (23).

As an additional investigation, the Jacobian-update methodology was also examined for the Δtx10 time-step for Grid 1 and 2. Figure 30 shows the results of this comparison. Figure 31 and 32 show θ versus y-COG; the results match for the different Jacobian-update methodologies. The increased time-step doesn't affect the trajectories using the different Jacobian methodologies. These results suggest the alternative Jacobian-update methodology is a promising one and should be investigated in future trajectory analyses.

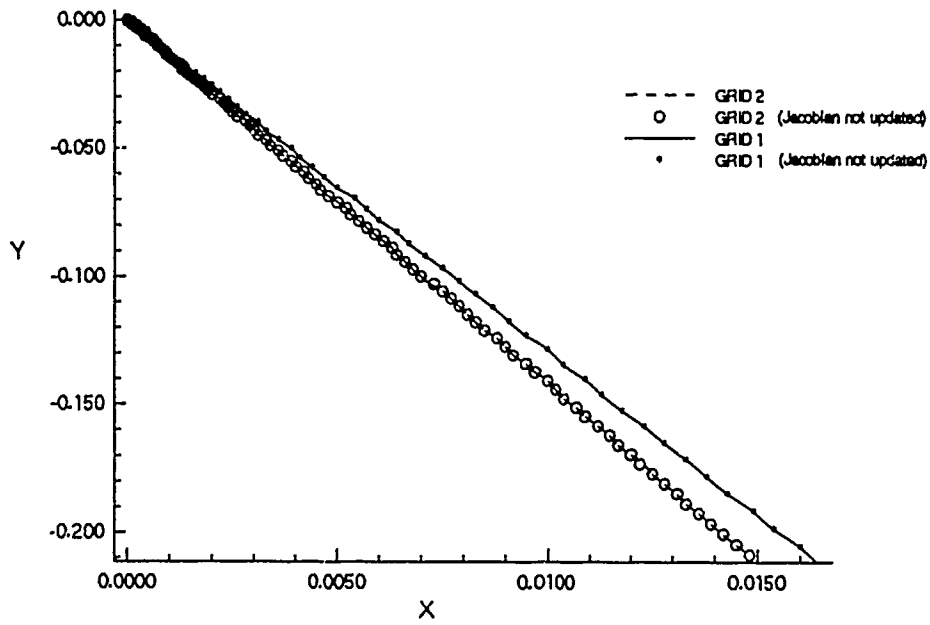


Figure 29: COG Trajectories Utilizing the Different Jacobian Update Methodologies (Case 1)

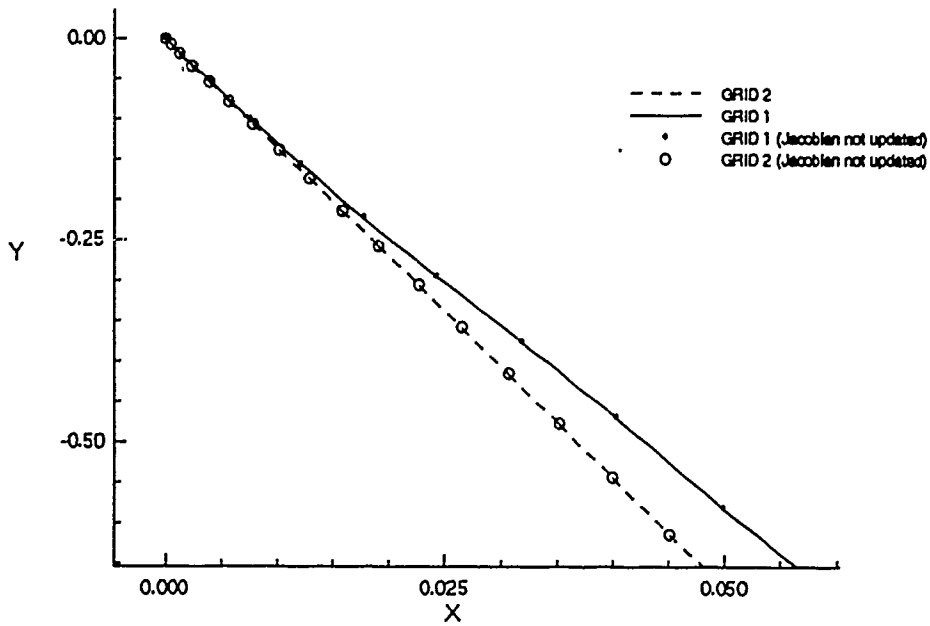


Figure 30: COG Trajectory Utilizing the Different Jacobian-Update Methodologies, Δtx10 (Case 1)

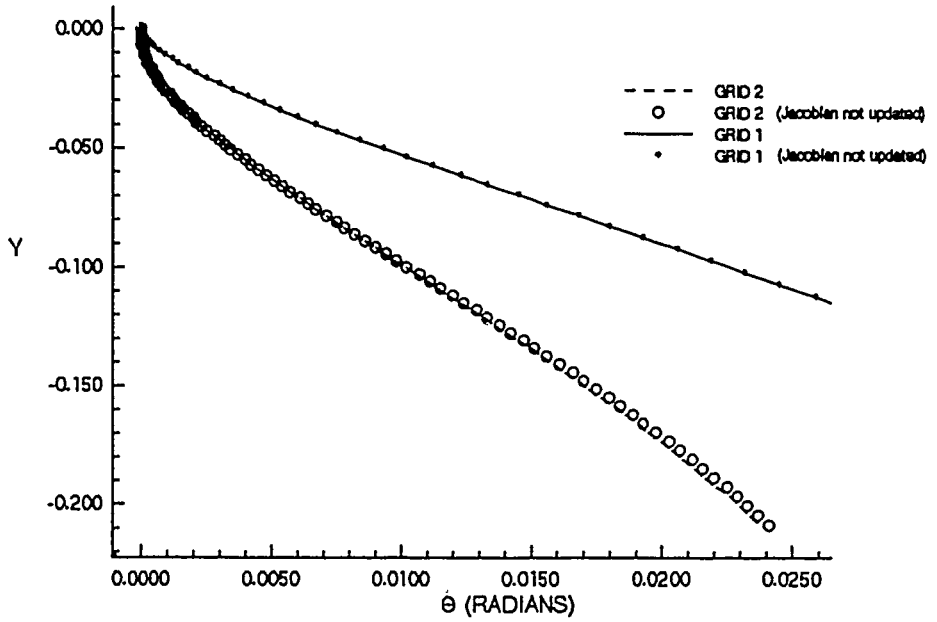


Figure 31: θ vs y-COG Utilizing the Different Jacobian Update Methodologies (Case 1)

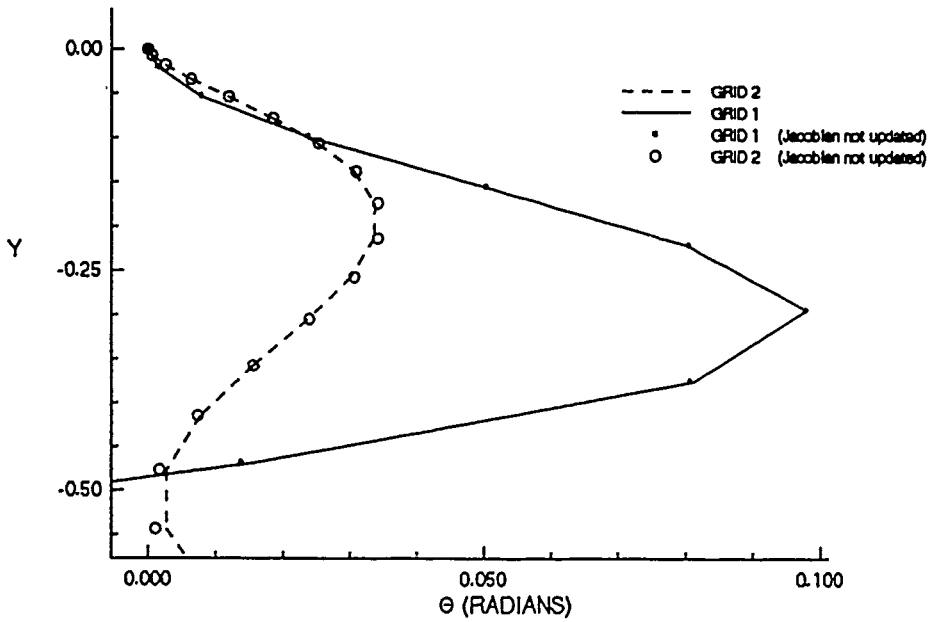


Figure 32: θ vs y-COG Utilizing the Different Jacobian Update Methodologies, Δtx10 (Case 1)

Chapter VI: Conclusions and Recommendations

VI.1 Conclusions

The Grid 1 and 2 trajectories matched well for the ellipse; the θ-trend for Grid 1 and 2 were also similar. This demonstrates the potential for using a dynamic grid, such as Grid 2 in future research. The numerical error introduced by the implicit algorithm made comparisons of the COG trajectories between the potential-flow solution and Grid 1 and 2 difficult.

The Case 1 and 2 examinations, between the potential flow and implicit algorithm solutions indicated the θ-trend is highly sensitive to the "zero" initial conditions. θ-trend results for Case 1 and 2 for the potential flow and implicit algorithm didn't match well. Case 3 was an attempt at examining the θ-trend at non-zero, initial conditions. The Case 3 results showed the potential flow and Grid 1 results still didn't correspond for the same initial conditions. In Case 3, both ellipses started to pitch nose-down, but the Grid 1 ellipse began to stabilize through the trajectory. The potential-flow solution drove the ellipse to a $\theta = 22.9°$ with no indication that it was going to stabilize. The potential-flow solution at non-zero initial conditions requires further investigation.

The Δtx10 trajectory results for Grids 1 and 2 showed comparable trends. The variation in pitch amplitude and frequency is caused by the larger time-step. The Δtx10

trajectories need to be examined for the ellipse used in Case 2 and 3. The absence of experimental data in this research doesn't allow me to quantify one trajectory better than another.

The Jacobian-update methodology resulted in a 45% reduction in computational time for the calculation of a trajectory. The close agreement in the resulting COG-trajectories and θ suggests the Jacobian-update schedule requires continued examination in future research.

The final finding of this work is the criticality of ensuring the time-step utilized falls within the limits of 3-DOF and implicit algorithms. If the Δt is larger than the algorithms can manage the results will be in error.

VI.2 Recommendations

Further investigations should examine the use of higher-order, 3-DOF routines in determining the stores trajectory. A higher-order, 3-DOF solution may provide closer results between the Grid 1 and 2 trajectories examined. Other 3-DOF algorithms may generate closer results between the potential-flow solution and the implicit algorithm.

Perturbation theory could be used to account for the Mach number effects in the potential-flow solution. These results could then be compared with results from the implicit algorithm. This analysis may uncover the origin of the numerical error present in the Euler algorithm. By

using perturbation theory in the potential-flow solution, the resulting trajectories and trends may compare better with implicit algorithm results.

Additional research should also focus on comparing the implicit algorithm/6-DOF routine with actual wind tunnel/6-DOF trajectories. The lack of experimental data for ellipse trajectories and the inherit problems with the implicit-algorithm results for the ellipse made for poor comparisons. Wind-tunnel data and the 6-DOF routine detailed in Reference 17 would be an excellent comparison tool with the implicit algorithm. Wind-tunnel data includes C_l, C_d, and C_m data for the store throughout the entire flow-field. These data are utilized in the 6-DOF routine and could be utilized in evaluating algorithm results.

Other grid-generation programs could also be investigated. The discontinuity in the cells along the branch-cut in the grid could be eliminated by other grid-generators. By not starting with the same grid, comparisons and conclusions on the results are difficult to make.

Further investigation of store trajectories isn't recommended with an ellipse. The inherit problems of the ellipse highlighted by Pullium (Reference 21) and in this research, suggests it is a difficult tool to use in trajectory analysis. Direct comparisons with experimental data is required to give valid results and conclusions on the accuracy and applicability of the CFD algorithm.

APPENDIX A: Trapezoidal Integration

The calculation of the lift, drag, and moment on the ellipse is performed using the trapezoidal-integration technique. This method is used in the potential-flow algorithm. The potential-flow algorithm calculates the pressure at each grid point along the surface of the ellipse. The surface pressure is integrated to determine the lift, drag, and moment. Trapezoidal integration averages the surface pressure calculated at each grid point to determine the pressure over the interval, P_{itl} using

$$P_{itl} = \frac{P_i + P_{i+1}}{2} \tag{103}$$

The x- and y-components of the surface pressure are summed in the body axis defined in Figure 6. The x- and y-components of force, X_b and Y_b, are determined using

$$X_b = \sum_{i=1}^{n} P_{itl_i}(x_i - x_{i+1}) \tag{104a}$$

$$Y_b = \sum_{i=1}^{n} P_{itl_i}(y_i - y_{i+1}) \tag{104b}$$

The lift and drag on the ellipse are then calculated using

$$L = Y_b \cos\alpha - X_b \sin\alpha \tag{105}$$

$$D = Y_b \sin\alpha + X_b \cos\alpha \tag{106}$$

$\alpha = \theta$ for the lift and drag calculation used in the implicit algorithm.

The moment calculation is slightly different, since the COG must be taken into account. The moment is positive in the counter-clockwise direction. It is calculated using

$$M = \sum_{i=1}^{n-1} \left(P_{itl_i} (x_{cen_i} - x_{cg}) x_{itl_i} + P_{itl_i} (y_{cen_i} - y_{cg}) y_{itl_i} \right) \tag{107}$$

where

$$x_{cen} = \frac{x_{i+1} + x_i}{2} \tag{108a}$$

$$y_{cen} = \frac{y_{i+1} + y_i}{2} \tag{108b}$$

These force calculations were validated using the zero lift and drag requirements for potential flow with no circulation. The moment calculation was validated using Eq. 7. Table 2 shows differences between the calculate and analytical solutions for different α. The values shown in Table 2 are for a 101x101 grid. By refining the grid, the Δ's decreased to values on the order of 10^{-9} for a 1001x1001 grid.

Table 2: Delta Force vs α

α (deg)	ΔL	ΔD	ΔM
0.0°	1.56×10^{-16}	3.27×10^{-16}	7.59×10^{-17}
10.0°	5.72×10^{-7}	8.39×10^{-7}	8.15×10^{-5}
20.0°	1.71×10^{-7}	2.17×10^{-7}	6.83×10^{-5}
30.0°	3.10×10^{-7}	7.58×10^{-7}	2.62×10^{-5}
40.0°	9.65×10^{-7}	9.62×10^{-7}	1.24×10^{-5}

The results in Table 2 show the trapezoidal-integration technique to be reasonably accurate compared to the analytical results.

APPENDIX B: Determination of Mach Number

The Whitfield algorithm was designed to work at subsonic, transonic, and supersonic velocities. This research focuses its attention on the subsonic region. Evaluation of the C_p on the surface of the ellipse determined the actual Mach number used to perform the store separation analyses. These values were also compared to the potential-flow solution. A 201x201 grid about an ellipse with a semi-major axis of 1.0 and a semi-minor axis of 0.2 was used to evaluate the C_p. The outer radius of the grid was 19.8 or approximately 10 chords from the center.

The algorithm computes C_p on the surface of the ellipse at seven different Mach numbers. Simpson recommended investigating values around Mach = 0.3, since the algorithm has been successfully tested at these values (23). Each solution converged until a residual value for density (ρ) of order 10^{-8} was obtained. Table 3 shows the results of the seven Mach numbers and their respective values of C_l, C_d, and C_m. Figure 33 and 34 show the C_p along the upper surface of the ellipse at an angle of attack of 0.0. The pressure coefficient curves are graphically symmetrical for the upper and lower surfaces.

Table 3: Force Coefficient at Various Mach Numbers

Mach Number:	Cl	Cd	Cm
0.10	-0.0001	0.1719	0.0000
0.15	0.0000	0.1188	0.0000
0.20	0.0000	0.0921	0.0000
0.25	0.0000	0.0762	0.0000
0.30	0.0000	0.0658	0.0000
0.35	0.0000	0.0586	0.0000
0.40	0.0000	0.0534	0.0000

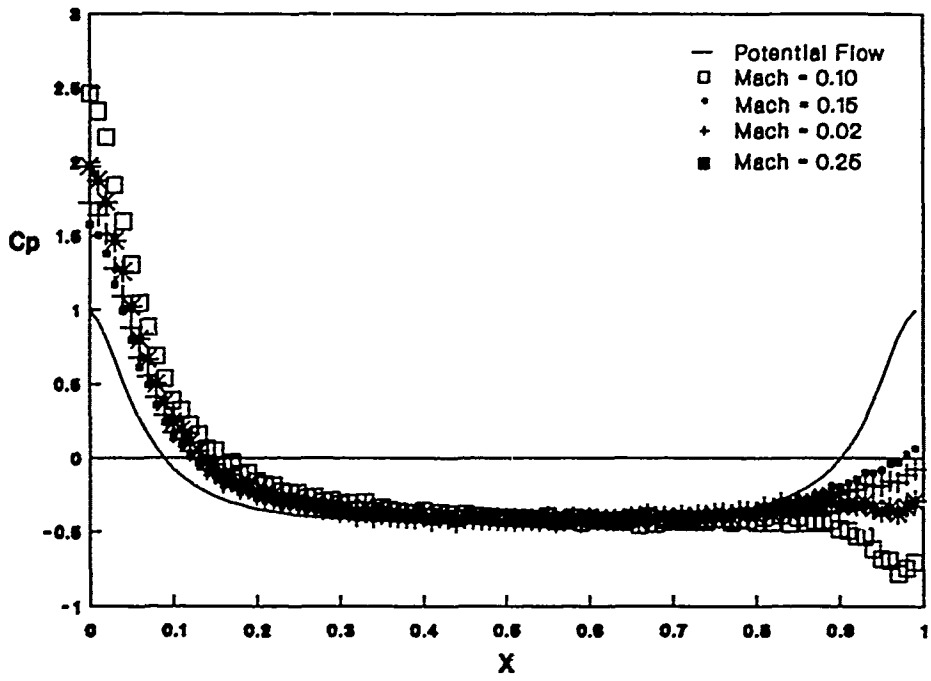


Figure 33: Cp vs X (Upper Surface)

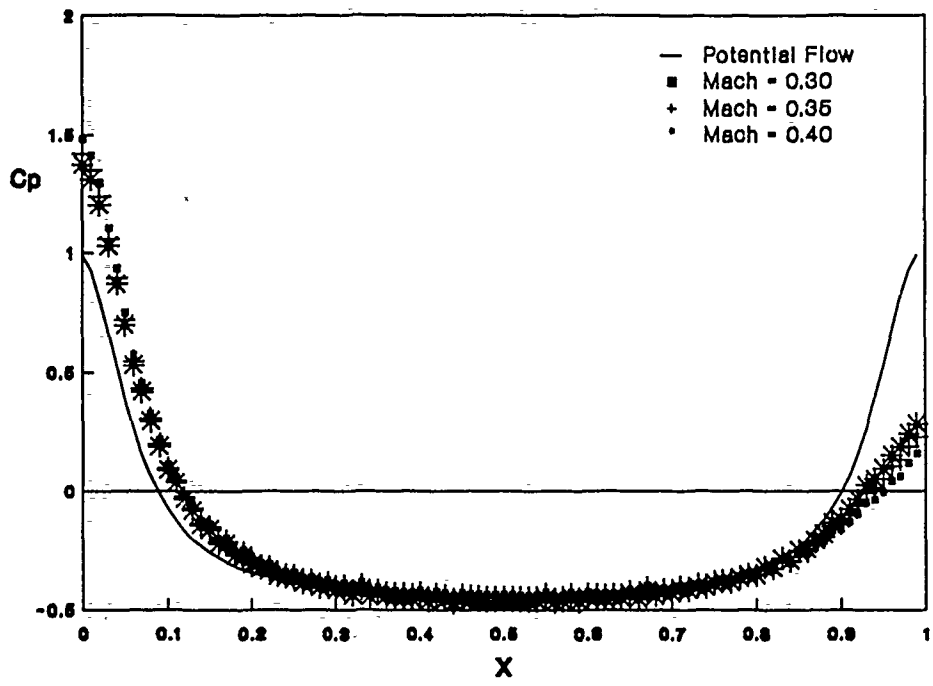


Figure 34: Cp vs X (Upper Surface)

The C_p curve at M = 0.40 is the closest to the potential-flow values, however M = 0.40 is too high for this research. The rest of the research used a Mach number of 0.30. The C_p curve at Mach = 0.30 is relatively smooth and compares well with the potential-flow result. More importantly, this Mach number is low enough so compressibility effects can still be considered negligible.

Bibliography

1. Anderson, W. Kyle, James L. Thomas, and David L. Whitfield. "Multigrid Acceleration of the Flux Split Euler Equations," AIAA 24th Aerospace Sciences Meeting. Paper No. AIAA-86-0274 New York: American Institute of Aeronautics and Astronautics, January 1986.

2. Arabshahi, Abdollah, A Dynamic Multiblock Approach to Solving the Unsteady Euler Equations about Complex Configurations. PhD Dissertation Department of Aerospace Engineering, Mississippi State University, Mississippi, May 1989.

3. Arabshabi, Dr Abdollah, Private Communication, 19 August 1991.

4. Batchelor, George Keith. An Introduction to Fluid Dynamics. Cambridge: Cambridge University Press, 1967.

5. Beam, Richard M. and R. F. Warming. "An Implicit Finite-Difference Algorithm for the Hyperbolic Systems in Conservation-Law Form," Journal of Computational Physics, Vol 22: 87-110, 1976.

6. Belk, Dave M., J. Mark Janus, and David L. Whitfield. "Three-Dimensional Unsteady Euler Equations Solutions on Dynamic Grids," AIAA 18th Fluid Dynamics and Plasmadynamics and Lasers Conference. Paper No. AIAA-85-1704. New York: American Institute of Aeronautics and Astronautics, July 1985.

7. Belk, Dave M., J. Mark Janus, and David L. Whitfield. Three-Dimensional Unsteady Euler Equations Solutions on Dynamic Grids. AFATL-TR-86-21, Air Force Armament Laboratory, Eglin Air Force Base Florida, April 1986.

8. Belk, David M. Unsteady Three-Dimensional Euler Equations Solutions on Dynamic Blocked Grids. PhD Dissertation Department of Aerospace Engineering, Mississippi State University, Mississippi, August 1986.

9. Eshbach, Ovid W. and Mott Souders. Handbook of Engineering Fundamentals. New York: John Wiley & Sons, 1975.

10. Gallaway, Charles R. Numerical Aerodynamic Analysis of a Free Falling Autorotating Plate. PhD Dissertation AFIT/DS/AA/83-2. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson Air Force Base Ohio, October 1983.

11. Halfman, Robert L. Dynamics, Volume 1: Particles, Rigid Bodies, and Systems. Reading, Massachusetts: Addison-Wesley Publishing Company, Inc, 1962.

12. Hoffman, Klaus A. Computational Fluid Dynamics for Engineers. Austin, Texas: Engineering Education Systems, 1989.

13. Janus, J Mark, The Development of a Three Dimensional Split Flux Vector Euler Solver with Dynamic Grid Applications, MS Thesis, Mississippi State University, August 1984.

14. Lamb, H. Hydrodynamics. Cambridge: Cambridge University Press, 1932.

15. Lugt, Hans J. "Autorotation of an Elliptic Cylinder about an Axis Perpendicular to the Flow," Journal of Fluid Mechanics, Vol 99: 817-840, 29 August 1980.

16. Lugt, Hans J. and Samuel Ohring. "Rotating Elliptic Cylinders in a Viscous Fluid at Rest or in a Parallel Stream," Journal of Fluid, Vol 79: 127-159, 20 January 1977.

17. Maneuvering Aircraft-Store Separation Simulation, Program Documentation Manual (Revision C). Fort Walton Beach, Florida: General Research Corporation, Contract F08637-86-D0010, Subcontract KP-0814, 3 December 1987.

18. Milne-Thomson, L.M. Theoretical Hydrodynamics. Fifth Edition, New York: The Macmillan Company, 1968.

19. Mracek, Maj Curtis P., Private Communication, 9 July 1991.

20. Pauletti, Capt Steven L. and Harbans S Sidhu. Accounting for Flexibility and Loss of Mass Using a Rigid Body Six-Degree-of-Freedom Program. Aeronautical/Structural Analysis Division, Directorate of Aeromechanics, Deputy for Engineering, AD/EN-TR-88-1004, August 1988.

21. Pulliam, Thomas H. "A Computational Challenge: Euler Solution for Ellipses," 27th Aerospace Sciences Meeting. Paper No. AIAA-89-0469. New York: American Institute of Aeronautics and Astronautics, January 1989.

22. Pulliam, Dr Thomas H., Private Communication, 18 Oct 1991.

23. Simpson, Dr L. Bruce, Private Communication, 15 Oct 1991.

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

25. "Store Certification," Systems Theory and Flight Test Techniques. Edwards Air Force Base, California: United States Air Force Test Pilot School, July 1986.

26. Welch, Ted, Separations Engineer, 3246th TW/TY, Private Communication, 19 Mar 1991.

27. Whitfield, David L. Implicit Upwind Finite Volume Scheme for the Three-Dimensional Euler Equation. MSSU-EIRS-ASE-85-1. Engineering and Industrial Research Station. Mississippi State University, Mississippi, September 1985.

28. Whitfield, David L., J. Mark Janus and L. Bruce Simpson. Implicit Finite Volume High Resolution Wave-Split Scheme for Solving the Unsteady Three-Dimensional Euler and Navier-Stokes Equations on Stationary or Dynamic Grids. MSSU-EIRS-ASE-88-2. Engineering and Industrial Research Station. Mississippi State University, Mississippi, February 1988.

29. Whitfield, David L. and J. Mark Janus. "Three-Dimensional Unsteady Euler Equations Solution Using Flux Vector Splitting." AIAA 17th Fluid Dynamics and Plasmadynamics and Lasers Conference. Paper No. AIAA-84-1552. New York: American Institute of Aeronautics and Astronautics, June 1984.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1991	Master's Thesis

4. TITLE AND SUBTITLE

COMPUTATION OF PLANAR STORE TRAJECTORIES USING A ADAPTIVE GRID PROCEDURE

5. FUNDING NUMBERS

6. AUTHOR(S)

William D. Hack, Captain, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GAE/ENY/91D-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr L. Bruce Simpson
WL/MNAA
Eglin AFB, Fl, 32542-5000

10. SPONSORING MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The objective of this research is to compare a quasi-analytical, potential flow/three-degree-of-freedom model to an implicit-Euler algorithm for the calculation of store trajectories. The implicit algorithm uses a cell-centered, finite-volume, spatial discretization applied to the Euler equations, written in time-dependent, curvilinear-coordinates. A flux-differencing Roe scheme is employed to find the split-fluxes and the Steger/Warming flux-vector method is used to calculate the flux-Jacobians. The potential flow and implicit algorithm are combined with a three-degree-of-freedom algorithm to evaluate the planar, freefall trajectories of a simple store shape. The research uses two different grid-modification techniques in the implicit algorithm evaluation.

Data collected for both grids utilized the minimum time-step in the three-degree-of-freedom algorithm for a Courant number of 10. Two test cases involved updating the flux-Jacobians after every time-step and only once during every 1000 iterations. The effect of multiplying the minimum time-step by factors of 2, 4, 6, 8, 10, and 100 were also examined. The potential flow and implicit algorithm trajectories did not compare very closely. The various Δt and Jacobian-update results matched rather closely.

14. SUBJECT TERMS

Potential Flow, Compressible Flow, Ellipse, Computational Fluid Dynamics, Store Trajectories

15. NUMBER OF PAGES

101

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements.***

Block 1. Agency Use Only *(Leave blank)*

Block 2. Report Date Full publication date including day, month, and year, if available (e.g 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	PR	- Project
G	- Grant	TA	- Task
PE	- Program Element	WU	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes Enter information not included elsewhere such as: Prepared in cooperation with ... Trans. of ... To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

- DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
- DOE - See authorities.
- NASA - See Handbook NHB 2200.2.
- NTIS - Leave blank.

Block 12b. Distribution Code.

- DOD - Leave blank.
- DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
- NASA - Leave blank.
- NTIS - Leave blank

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages

Block 16. Price Code Enter appropriate price code *(NTIS only)*

Blocks 17. - 19. Security Classifications. Self-explanatory Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information stamp classification on the top and bottom of the page

Block 20. Limitation of Abstract This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.