NEAVES, MICHAEL DEAN. Analysis of Hypersonic Aircraft Inlets Using Flow Adaptive Mesh Algorithms. (Under the direction of D. Scott McRae.)

The numerical investigation into the dynamics of unsteady inlet flowfields is applied to a three-dimensional scramjet inlet-isolator-diffuser geometry designed for hypersonic type applications. The Reynolds-Averaged Navier-Stokes equations are integrated in time using a subiterating, time-accurate implicit algorithm. Inviscid fluxes are calculated using the Low Diffusion Flux Splitting Scheme of Edwards. A modified version of the dynamic solution-adaptive point movement algorithm of Benson and McRae is used in a coupled mode to dynamically resolve the features of the flow by enhancing the spatial accuracy of the simulations. The unsteady mesh terms are incorporated into the flow solver via the inviscid fluxes. The dynamic solution-adaptive grid algorithm of Benson and McRae is modified to improve orthogonality at the boundaries to ensure accurate application of boundary conditions and properly resolve turbulent boundary layers. Shock tube simulations are performed to ascertain the effectiveness of the algorithm for unsteady flow situations on fixed and moving grids. Unstarts due to a combustor and freestream angle of attack perturbations are simulated in a three-dimensional inlet-isolator-diffuser configuration.
ANALYSIS OF HYPersonic AIRCRAFT INLETS USING FLOW ADAPTIVE MESH ALGORITHMS

by

MICHAEL DEAN NEAVES

A dissertation submitted to the Graduate Faculty of North Carolina State University in partial fulfillment of the requirements for the Degree of Doctorate of Philosophy

AEROSPACE ENGINEERING

Raleigh

2001

APPROVED BY:

Dr. D. Scott McRae - Chairman

Dr. F.R. DeJarnette

Dr. M.A. Vouk

Dr. H.A. Hassan

Dr. J.R. Edwards

Dr. J.S. Scroggs
BIOGRAPHY

Michael Dean Neaves was born in Jefferson, North Carolina on the 16th of December, 1965, and he grew up in the rural mountain county of Ashe. After graduating from Ashe Central High School, Mike earned a Bachelor of Science in Aerospace Engineering from North Carolina State University in 1988. He then earned his commission in the U.S. Navy from Officer Candidate School in Newport, Rhode Island. From there, he attended Naval Nuclear Power School, Nuclear Power Prototype Training, and Surface Warfare Officer School. His Division Officer tour was on board the USS Dwight D. Eisenhower (CVN-69) where he was the Reactor Controls Division Officer and qualified Propulsion Plant Watch Officer, Reactor Duty Officer and Officer of the Deck. Mike received the Navy Achievement Medal and the Southwest Asia Service Medal with two bronze stars for his service during Operation Desert Shield/Storm. In 1993, after honorary completion of his obligatory service, LT Neaves resigned his commission in order to return to graduate school at North Carolina State University. He finished his Master of Science in November of 1995. Mike currently works full time for the university as a Computer Network Coordinator for the College of Agriculture and Life Sciences Information Technology Department.
ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. McRae, for his support. I would also like to thank Dr. Fred DeJarnette, Dr. Hassan, and Dr. Edwards for being on my committee and teaching me on many occasions. Thanks to Dr. Scroggs and Dr. Vouk for being on my committee. I would like to thank Dr. Nd Chokani for standing in for Dr. Edwards during my oral preliminary examination. I thank the North Carolina Supercomputing Center for a grant of supercomputing resources and the Air Force Office of Scientific Research for various grants as overseen by Dr. Len Sakell. I am grateful to Kent Misegades of Computational Engineering International for assistance and instruction on making large three-dimensional animations using Ensight.

My family provided continuous support and encouragement over the years. I would like to thank them all: Gerald Dean Neaves, Rosa Lee Neaves, Anna Marie Neaves Orr, Bill Orr, William Michael Orr, and Kaitlin Rose Orr. A special thanks to Tammy Miller for her support and patience over the years.
# TABLE OF CONTENTS

**LIST OF TABLES**

v

**LIST OF FIGURES**

vi

**NOMENCLATURE**

vii

1. **INTRODUCTION**

1

2. **OUTLINE OF NUMERICAL PROCEDURE**

8

2.1 Development of a Time-Accurate Implicit Algorithm ........... 8

2.2 Development of Coupled Adaption Update Algorithm .......... 11

2.3 Adaptive Grid Algorithm .................................. 17

2.3.1 Orthogonal Adaptive Grid Algorithm ................... 18

2.3.2 Weight Function ...................................... 21

3. **RESULTS AND DISCUSSION**

23

3.1 Riemann Problem ........................................ 23

3.1.1 Fixed-Grid Riemann Solutions ......................... 24

3.1.1.1 Subiterations ................................... 24

3.1.1.2 Relaxation Parameters ........................... 25

3.1.1.3 Minmod Limiter ................................ 25

3.1.1.4 Time Step Size ................................ 26

3.1.2 Adaptive Riemann Problem ............................ 27

3.1.3 Computational Efficiency and Accuracy .................. 28

3.2 Scramjet Inlet .......................................... 30

3.2.1 Three-Dimensional Inlet Steady Results ................ 30

3.2.2 Computational Time .................................. 34

3.2.3 Three-Dimensional Inlet Unstart Calculations .......... 35

3.2.3.1 Exit Flap Induced Unstart ...................... 35

3.2.3.2 Angle of Attack Unstart ......................... 38

4. **CONCLUSIONS**

41

5. **REFERENCES**

43
LIST OF TABLES

Table 1. Error Norm and CPU Times for Riemann Problem .............. 28
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Orthogonal Adaptation Process Near Boundaries</td>
<td>46</td>
</tr>
<tr>
<td>2</td>
<td>Shock Tube Solution for Various Subiterations</td>
<td>46</td>
</tr>
<tr>
<td>3</td>
<td>Shock and Contact Surface for Various Subiterations</td>
<td>47</td>
</tr>
<tr>
<td>4</td>
<td>Unity Relaxation Factors</td>
<td>47</td>
</tr>
<tr>
<td>5</td>
<td>Minmod Limiter</td>
<td>48</td>
</tr>
<tr>
<td>6</td>
<td>Shock Tube Solutions with Various Time Steps</td>
<td>48</td>
</tr>
<tr>
<td>7</td>
<td>Shock and Contact Surface for Various Time Steps</td>
<td>49</td>
</tr>
<tr>
<td>8</td>
<td>Adapted versus Unadapted Shock Tube Results</td>
<td>49</td>
</tr>
<tr>
<td>9</td>
<td>Adapted Cell Volume and Grid Speeds</td>
<td>50</td>
</tr>
<tr>
<td>10</td>
<td>Dual-Mode Scramjet Inlet Model</td>
<td>50</td>
</tr>
<tr>
<td>11</td>
<td>Unadapted Grid and Pressure Contours</td>
<td>51</td>
</tr>
<tr>
<td>12</td>
<td>Adapted Grid and Pressure Contours</td>
<td>51</td>
</tr>
<tr>
<td>13</td>
<td>Grid from Previous Adaptation Algorithm</td>
<td>52</td>
</tr>
<tr>
<td>14</td>
<td>Mesh Near Inlet Boundary for Original (left) and Orthogonal (right) Mesh Algorithms</td>
<td>52</td>
</tr>
<tr>
<td>15</td>
<td>Grid and Pressure Contours in Aft Inlet and Isolator (Centerline)</td>
<td>53</td>
</tr>
<tr>
<td>16</td>
<td>Centerline Pressure Compared to Experimental Results</td>
<td>53</td>
</tr>
<tr>
<td>17</td>
<td>Centerline Pressure in Aft Inlet and Isolator Region</td>
<td>54</td>
</tr>
<tr>
<td>18</td>
<td>Adapted 3-D Mesh Surfaces (i=30,j=40,k=28)</td>
<td>54</td>
</tr>
<tr>
<td>19</td>
<td>Density Contours on Bottom Wall, Centerline and Selected i Surfaces</td>
<td>55</td>
</tr>
<tr>
<td>20</td>
<td>Adapted Mesh on Bottom Wall, Centerline and Selected i Surfaces</td>
<td>55</td>
</tr>
<tr>
<td>21</td>
<td>Exit Flap Unstart (0.0 ms and 1.65 ms)</td>
<td>56</td>
</tr>
<tr>
<td>22</td>
<td>Exit Flap Unstart (2.47 ms and 4.26 ms)</td>
<td>57</td>
</tr>
<tr>
<td>23</td>
<td>Exit Flap Unstart (4.67 ms and 5.49 ms)</td>
<td>58</td>
</tr>
<tr>
<td>24</td>
<td>Exit Flap Inlet Unstart in Progress I (3-D view)</td>
<td>59</td>
</tr>
<tr>
<td>25</td>
<td>Exit Flap Inlet Unstart in Progress II (3-D view)</td>
<td>59</td>
</tr>
<tr>
<td>26</td>
<td>Exit Flap Inlet Unstart after Upper Cowl Spillage (3-D view)</td>
<td>60</td>
</tr>
<tr>
<td>27</td>
<td>Angle of Attack Unstart (0.137 ms)</td>
<td>60</td>
</tr>
<tr>
<td>28</td>
<td>Angle of Attack Unstart (0.275 ms and 0.686 ms)</td>
<td>61</td>
</tr>
<tr>
<td>29</td>
<td>Angle of Attack Unstart (1.37 ms and 2.06 ms)</td>
<td>62</td>
</tr>
<tr>
<td>30</td>
<td>Angle of Attack Unstart (2.75 ms and 3.43 ms)</td>
<td>63</td>
</tr>
<tr>
<td>31</td>
<td>Angle of Attack Unstart (4.12 ms)</td>
<td>64</td>
</tr>
</tbody>
</table>
NOMENCLATURE

\[ E,F,G \] flux vectors
\[ H \] enthalpy
\[ J \] transformation Jacobian
\[ P \] pressure
\[ P,Q,R \] parametric space coordinates
\[ Re \] Reynolds Number
\[ U \] dependent variable vector
\[ V \] volume
\[ t \] time
\[ u,v,w \] velocity components
\[ x,y,z \] space coordinates
\[ \alpha_c \] curvature weighting coefficient
\[ \Delta \] finite difference operator
\[ \xi, \eta, \zeta \] computational space coordinates
\[ \rho \] density
\[ \sigma_k \] biasing coefficient
\[ \tau \] non-dimensional time
\[ \phi \] difference of adaption variables
\[ \omega \] weight function
\[ \nu \] implicit approximation parameter
\[ \varphi \] implicit approximation parameter

Subscripts
\[ i,j,k \] computational space coordinates
\[ p \] physical quantity
\[ t \] partial derivative with respect to time
\[ x,y,z \] partial derivative with respect to \( x,y,z \)
\[ \infty \] freestream quantity
\[ o \] total quantity
\[ 1 \] strong conservation form

Superscripts
\[ c \] convective component of inviscid flux
\[ I \] inviscid component of flux
\[ \ell \] subiteration index
\[ p \] pressure component of inviscid flux
\[ n \] current time level
\[ ut \] unsteady contravariant
\[ \beta \] volume adaptation control
1. INTRODUCTION

Time-accurate simulations of complex three-dimensional flow fields about or through airframe components cannot, in general, be computed with short turnaround times. For example, simulating the response of a turbulent, three-dimensional inlet flow field to a freestream, engine compressor face, or combustor perturbation can often require excessive computational resources in terms of time and/or memory requirements. The number of grid points required to properly resolve large turbulent problems can easily exceed available memory. The large number of timesteps required by most explicit time accurate flow solvers can consume hundreds, if not thousands, of hours on a supercomputer. A time accurate, implicit flow solver coupled with a solution adaptive grid algorithm could make inlet unstarts a more tractable problem.

Research to date shows computation of high-speed aircraft internal flow fields to require three-dimensional calculations. In 1993, Korte et al studied sidewall compression scramjet inlets in an inviscid parametric study. A Mach 4 swept-sidewall inlet was parametrically improved using inviscid three-dimensional calculations. A turbulent three-dimensional solution was then obtained for the optimized inlet. The resulting inlet flow field was found to be a complex three-dimensional structure with many multi-dimensional shock/shock and shock/boundary-layer interactions. The viscous solution revealed a large top-wall boundary layer separation on the optimal solution inviscid design.

During the same year, Sakata et al presented two papers on experimental and computational results for a Supersonic Air-Intake with a 5-shock system at Mach 3. The
calculations were turbulent, but only two-dimensional since the experimental model did not vary in width. Sakata concluded that the compression process is dominated by the strong viscous interactions between shock waves and the turbulent boundary layer. Comparing computational and experimental results, their most important conclusions were that the two-dimensional calculations have a tendency to overestimate the pressure recovery and that to more precisely simulate the inlet’s performance, three-dimensional calculations are required.

Paynter and Mayer et al have written extensively on flow stability issues and unstart analyses in supersonic inlets.\textsuperscript{4,5,6,7,8} Most of their work has been either quasi-steady one-dimensional analysis or time accurate two-dimensional Euler simulations with the goal of determining inlet stability based on freestream temperature, pressure, velocity and compressor pressure/flow perturbations. If these perturbations exceed the stability limits of the inlet, the normal shock is expelled from the inlet in what is known as an inlet unstart. Inlet unstarts result in a loss of propulsive efficiency due to lower pressure recovery, asymmetric pressurization of the wing, possibly sympathetic unstarts of other inlets, and suddenly increased aircraft drag which can compromise aircraft control. Euler simulations implicitly assume that the dominant mechanism for translation of the normal shock due to a disturbance is wave propagation. While the wave propagation assumption may be valid at design operating condition, due to most inlets having bleed systems which minimize shock/boundary layer interactions, many perturbations cause the normal shock to translate further into the subsonic region of the inlet which causes shock strength to increase in a region without boundary layer bleed. The combination of a stronger shock/
boundary layer interaction and no bleed could result in an non-wave propagation dominated inlet unstart. The flow in the throat is not one-dimensional, making correctly locating and calculating the throat region difficult using inviscid analysis. Mayer and Paynter conclude that future work should include viscous calculations to include throat blockage effects and to investigate inlet unstarts caused by boundary layer separation.

Trexler et al have performed numerous experiments on hypersonic-type inlets in the Mach 4 Blow Down Facility at NASA Langley. A Trexler Dual-Mode Scramjet Inlet Model is used as the basis for much of the computation results that follow. Much of their work is aimed at determining the maximum back pressure an inlet can sustain before unstarting. Burning fuel raises inlet back pressure, so throttling the exit to raise back pressure is a proxy for burning fuel. The inlet stability determines maximum back pressure which determines maximum fuel burn - i.e. maximum thrust the propulsion system can produce.

In 1995, Knight et al redesigned the NASA P2 hypersonic inlet. The original P-series inlets intended to cancel a reflected shock via surface turning at a reflection point in order to achieve higher pressure recovery and a nearly constant static pressure at the throat. The inlets were originally designed using the Method of Characteristics and a boundary layer model to account for displacement thickness. Experimental results showed that the reflected shock was not canceled by the surface turning which led to the redesign by Knight et al using a Reynolds-averaged Navier-Stokes code (NPARC).

Much of the inlet unstart analysis to date is based upon quasi one-dimensional analysis where the only boundary layer consideration is subtracting the displacement
thickness from the available flow area. The displacement thickness approach ignores any three-dimensional effects and all inviscid/boundary layer interactions. Most of the remaining work has been time accurate, two-dimensional simulations - usually inviscid. Grid adaptation has been utilized in two-dimensional unstart analysis performed by Benson and McRae\textsuperscript{12} using an explicit Reynolds-averaged Navier-Stokes flow solver. Given the inherent three-dimensional nature of turbulence, quasi one-dimensional and two-dimensional analysis have been shown to over predict pressure recovery and ignore complex three-dimensional shock/boundary layer interactions; thus making them inadequate for analyzing three-dimensional turbulent flow fields. The goal of the present work is to extend solution dynamic grid adaptation unstart calculations to large three-dimensional turbulent flow fields.

When using dynamic grid adaptation to restrain computer memory and processing times, there are two fundamentally different approaches to dynamic grid adaptation: point movement and enrichment. When constrained by computer memory resources, enrichment may not be a viable option for large three-dimensional problems due to the variable memory aspect of enrichment. Another problem regarding the enrichment of structured grids is the breakdown of the conveniently structured data set. Point movement maintains a constant maximum memory requirement and the structured data set lends itself to simple algorithm vectorization. For the large three-dimensional structured grids used in internal flow calculations, point movement was chosen as the best option for dynamic adaptation.

Given point movement on a structured grid, the next issue is time accuracy. If only a steady state solution is desired, only a steady state flow solver and grid point movement
algorithm are required. The steady state flow solver merely sees the grid point movement as a perturbation to the solution and no solution update or further coupling with the flow solver is required. In essence, the unsteady transformation terms are neglected exactly as the unsteady flow terms are neglected in most steady state flow solvers.

If time accuracy is desired while performing point movement grid adaptation, more must be done to ensure temporal accuracy while moving grid points. Two approaches exist to maintain temporal accuracy while moving the mesh: coupled and decoupled. The coupled approach was previously used in one dimension by Klopfer and McRae\textsuperscript{13} for explicit solvers and by Orkwis and McRae\textsuperscript{14} for implicit solvers. Benson and McRae\textsuperscript{15} used the coupled approach to solve the chain rule form of the governing equations in two and three dimensions using an explicit form of MacCormack’s method. The decoupled approach was introduced by Benson and McRae\textsuperscript{15} and continued by Laflin.\textsuperscript{16} In the decoupled approach, the grid adaptation and solution update algorithms are completely independent of the flow solver algorithm. The decoupled approach can use any flow solver because mesh point movement and solution update to the new grid are independent of the flow solver. The decoupled approach has the advantage of being extremely portable, but the disadvantage is a costly solution update calculation. For the present work, a coupled approach is taken to avoid the largest portion of the costly solution update. The penalty paid for using a coupled approach is a lack of portability.

In the present work, the goal is to develop a three-dimensional, adaptive mesh algorithm coupled to a time accurate, implicit, Reynolds-averaged Navier-Stokes solver in order to increase the accuracy of three-dimensional inlet calculations. The final algorithm
should be capable of calculating inlet unstarts due to engine or freestream perturbations. Three advances are discussed, the first being the development of a time accurate implicit flow solver. Time accurate subiterations of Rai\textsuperscript{17} were added to the planar Gauss-Seidel flow solver of Edwards.\textsuperscript{18} The second advance is an efficient coupling of the grid point movement transformation to the flow solver in order to solve properly the unsteady transformation governing equations. Solving the unsteady transformation governing equations is achieved by including the grid movement terms directly in the numerical formulation of the unsteady, transformed equation system. The unsteady transformation terms are easily incorporated into the Low-Diffusion Flux Splitting of Edwards.\textsuperscript{19} Grid speed terms are included in the contravariant velocities, and a pressure term is required to convert enthalpy in the upwind terms into the required energy term in the governing equations.

Lastly, a new point movement algorithm is developed which minimizes grid skewness near viscous boundaries in order to calculate turbulent boundary layers. The point movement algorithm of Benson and McRae\textsuperscript{15} has been modified to include orthogonality factors which prevent excessive grid skewness. A stretching factor (area ratio) raised to the dot product of unit normals (orthogonality factor) is used to adjust point movement coefficients as required to maintain grid quality. The weight function used in the point movement algorithm is a linear combination of scaled gradients and curvature of the dependent variables.\textsuperscript{20}

Shock tube simulations are performed to ascertain the effectiveness of the algorithm for unsteady flow situations on fixed and moving grids. A steady solution on fixed and adapted mesh is presented for a hypersonic inlet, isolator, diffuser configuration of
Trexler et al. Starting from an adapted steady solution, dynamic adaptation unstarts were calculated. To best simulate the experiment, the rear ramp in the diffuser was moved up and the exit pressure adjusted. The diffuser is primarily subsonic, so moving the ramp and increasing exit pressure will send a pressure wave forward which unstarts the inlet. A freestream perturbation unstart due to an angle of attack perturbation is also calculated and presented. Finally, time-accurate simulations of the three-dimensional unstart process are presented.

The final flow solver and adaptive mesh algorithm should provide an efficient, accurate means of simulating a three-dimensional inlet unstart. The three-dimensional unstart simulations should provide improved accuracy, giving additional insight into the shock/boundary layer mechanisms which can reduce inlet stability, thereby reducing maximum back pressure and available thrust.
2. OUTLINE OF NUMERICAL PROCEDURE

2.1 Development of a Time-Accurate Implicit Algorithm

A standard alternative to the long turnaround times of explicit flow solvers is to use an implicit flow solver with a less severe CFL\textsuperscript{21} (Courant, Friedrichs, and Lewy) stability requirement. Unfortunately, most implicit algorithms are implemented in a non-time-accurate manner for tractability and steady state convergence, and as a result the implicit algorithms contain approximate factorizations, explicit boundary conditions, relaxation, and linearization error, etc. The subiteration techniques of Rai\textsuperscript{17} can be used to restore time accuracy while still taking advantage of simplifying approximations to the implicit operator. Adding the time-accurate subiterations to the existing implicit code of Edwards\textsuperscript{18} results in a time-accurate upwind relaxation method. The existing implicit code employs a second order LDFSS(2)\textsuperscript{19} upwind discretization with planar Gauss-Seidel algorithm for time advancement.

The subiteration techniques of Rai\textsuperscript{17} can be applied to the generic form of a system of partial differential equations. For development, consider the two-dimensional Euler equations as written in equation 1,

\[ \frac{\partial U}{\partial t} + F(U) = 0 \]  \hspace{1cm} (1)

where the non-linear flux vectors are

\[ F(U) = \frac{\partial}{\partial x} E(U) + \frac{\partial}{\partial y} G(U). \]  \hspace{1cm} (2)
An implicit approximation in time for the solution of equation 1 can be expressed as

\[
\Delta t \left[ \frac{\partial U}{\partial t} + F(U) \right] = \Delta U^n + \frac{\nu \Delta t}{1 + \phi} F(U^{n+1}) = \frac{(\nu - 1) \Delta t}{1 + \phi} F(U^n) + \frac{\phi}{1 + \phi} \Delta U^{n-1} + \left( \nu - \frac{1}{2} - \phi \right) O(\Delta t^2) + O(\Delta t^3)
\]  

where the parameters \( \nu \) and \( \phi \) are chosen to provide different schemes with differing accuracy. Considering \( p \) as a subiteration index\(^2\), subtracting \( U^p \) from both sides of equation 3 yields:

\[
\Delta U^p + \frac{\nu \Delta t}{1 + \phi} F(U^{n+1}) = -(U^p - U^n) + \frac{(\nu - 1) \Delta t}{1 + \phi} F(U^n) + \frac{\phi}{1 + \phi} \Delta U^{n-1}.
\]  

Setting \( \nu = 1 \) and \( \phi = 1/2 \) results in the three-point backward scheme which is second-order accurate in time. Linearizing around \( U^p \) and substituting into equation 4 results in the following:

\[
F(U^{p+1}) = F(U^p) + \frac{\partial}{\partial U} F(U^p) \Delta U^p + O(\Delta t^2)
\]

\[
\left[ I + \frac{2 \Delta t}{3} \frac{\partial}{\partial U} F(U^p) \right] \Delta U^p = -\frac{2 \Delta t}{3} F(U^p) - \left[ U^p - \frac{4}{3} U^n + \frac{1}{3} U^{n-1} \right].
\]

Finally, divide by \( \Delta t_p \), the physical time step, and multiply by \( 3/2 \). An implicit approximation in time for the solution of equation 1 can now be expressed as:

\[
\left[ \frac{I}{\Delta t_p} + \frac{\partial}{\partial U} F(U^p) \right] \Delta U^p = F(U^p) - \frac{\left[ \frac{3}{2} U^p - 2 U^n + \frac{1}{2} U^{n-1} \right]}{\Delta t_p}.
\]
Examining equation 7 reveals the modifications required to the existing implicit code of Edwards to allow subiterating to improve time accuracy. The residual must be modified to enforce second-order time accuracy when the subiterations converge. The new time difference term was added to the residual calculation, and storage provided for the two extra time levels as required to calculate the time difference. A fixed global physical time step was implemented on the left and right hand side of the implicit solver. With these modifications, time accuracy can be improved to acceptable levels by subiterating over the modified implicit flow solver. Second-order accuracy is desired, therefore a three-point backward time difference is used with at least two subiterations.\textsuperscript{23} All forward and backward relaxation parameters in existing algorithm are set to unity. The original pressure-based limiter was required for steady state convergence but appears to be too oscillatory for the three-point backward time integration scheme. In an attempt to reduce oscillations near discontinuities, a Minmod limiter option was implemented. The modifications listed above should restore temporal accuracy to any general implicit algorithm.
2.2 Development of Coupled Adaption Update Algorithm

The coupling of a three-dimensional adaptive mesh algorithm to a time accurate upwind implicit solution algorithm is presented in this section. The goal is to couple the grid point movement and solution update more closely to the flow solver by solving the unsteady transformation governing equations using an implicit algorithm with grid movement included in the spatial upwinding. It is shown that the grid movement terms can be incorporated into any existing upwinding scheme which splits the inviscid flux into a convective and pressure contribution. The new coupled solution adaptive mesh algorithm is temporally accurate while only adding point movement and mesh speed calculation to the overall computational requirements.

The flow fields under consideration are governed by the three-dimensional compressible Navier-Stokes equations.

\[
\frac{\partial U}{\partial t} + \frac{\partial E}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial G}{\partial z} = 0
\] (8)

where \( U \) is the vector of conserved variables, \( E, F \) and \( G \) are the combined inviscid and viscous fluxes. In a curvilinear coordinate system defined by the unsteady transformation \( \xi = \xi(x, y, z, t), \eta = \eta(x, y, z, t), \zeta = \zeta(x, y, z, t), \tau = t \), the Navier-Stokes equations can be written as shown in equation 9. Multiplying by the Jacobian, \( J \), and using the standard manipulation of the spacial terms into strong conservation law form results in equation 10.
\[
\frac{\partial U}{\partial \tau} + \frac{\partial U \partial \xi}{\partial \xi \partial t} + \frac{\partial U \partial \eta}{\partial \eta \partial t} + \frac{\partial U \partial \zeta}{\partial \zeta \partial t} + \frac{\partial E \partial \xi}{\partial \xi \partial x} + \frac{\partial E \partial \eta}{\partial \eta \partial x} + \frac{\partial E \partial \zeta}{\partial \zeta \partial x} + \frac{\partial F \partial \xi}{\partial \xi \partial y} + \frac{\partial F \partial \eta}{\partial \eta \partial y} + \frac{\partial F \partial \zeta}{\partial \zeta \partial y} + \frac{\partial G \partial \xi}{\partial \xi \partial z} + \frac{\partial G \partial \eta}{\partial \eta \partial z} + \frac{\partial G \partial \zeta}{\partial \zeta \partial z} = 0
\]  
(9)

\[
J \frac{\partial U}{\partial \tau} + J \frac{\partial U \partial \xi}{\partial \xi \partial t} + J \frac{\partial U \partial \eta}{\partial \eta \partial t} + J \frac{\partial U \partial \zeta}{\partial \zeta \partial t} + \frac{\partial}{\partial \zeta} \left( J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} \right) + \frac{\partial}{\partial \eta} \left( J E \frac{\partial \eta}{\partial x} + J F \frac{\partial \eta}{\partial y} + J G \frac{\partial \eta}{\partial z} \right) + \frac{\partial}{\partial \xi} \left( J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} \right) = 0
\]  
(10)

In the present development, \( J \equiv \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \) and represents the Jacobian of the coordinate transformation; i.e. the volume of a finite volume cell. The grid movement terms in equation 10 can be incorporated as shown below:

\[
J \frac{\partial U}{\partial \tau} + U \frac{\partial J}{\partial \tau} + \frac{\partial}{\partial \zeta} \left( J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} + J U \frac{\partial \xi}{\partial t} \right) + \frac{\partial}{\partial \eta} \left( J E \frac{\partial \eta}{\partial x} + J F \frac{\partial \eta}{\partial y} + J G \frac{\partial \eta}{\partial z} + J U \frac{\partial \eta}{\partial t} \right) = 0
\]  
(11)

Now to recast in strong conservation form, combine the \( \tau \) derivative terms and note that \( t = \tau \).

\[
\frac{\partial (JU)}{\partial t} + \frac{\partial}{\partial \xi} \left( J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} + J U \frac{\partial \xi}{\partial t} \right) + \frac{\partial}{\partial \eta} \left( J E \frac{\partial \eta}{\partial x} + J F \frac{\partial \eta}{\partial y} + J G \frac{\partial \eta}{\partial z} + J U \frac{\partial \eta}{\partial t} \right) + \frac{\partial}{\partial \xi} \left( J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} + J U \frac{\partial \xi}{\partial t} \right) = 0
\]  
(12)

Further simplifications can be made by the following: \( U_1 = JU \)

\[
E_1 = J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} + J U \frac{\partial \xi}{\partial t}
\]  
(13)

\[
F_1 = J E \frac{\partial \eta}{\partial x} + J F \frac{\partial \eta}{\partial y} + J G \frac{\partial \eta}{\partial z} + J U \frac{\partial \eta}{\partial t}
\]  
(14)

\[
G_1 = J E \frac{\partial \xi}{\partial x} + J F \frac{\partial \xi}{\partial y} + J G \frac{\partial \xi}{\partial z} + J U \frac{\partial \xi}{\partial t}
\]  
(15)
The final form of the governing unsteady transformation compressible Navier-Stokes equations is given by equation 16.

\[
\frac{\partial U_1}{\partial t} + \frac{\partial E_1}{\partial x} + \frac{\partial F_1}{\partial y} + \frac{\partial G_1}{\partial z} = 0
\]  

(16)

The grid speed terms \((\xi, \eta, \zeta)\) arise from an unsteady transformation and represent the time rate of change in computational space of a point fixed in physical space. The physical grid speeds can be calculated with a second order one sided difference, for example, \(x_t = \frac{\partial x}{\partial t} = \frac{3x^{n+1} - 4x^n + x^{n-1}}{2\Delta t}\). Noting that \(\xi_x = \frac{\partial \xi}{\partial x} = \frac{1}{J} \left( \frac{\partial y \partial z}{\partial \eta \partial \zeta} - \frac{\partial z \partial y}{\partial \eta \partial \xi} \right)\), we see the metric grid speed terms can easily be calculated from the physical grid speed and metrics as shown in equation 17.

\[
\xi_t = -[x_t \xi_x + y_t \xi_y + z_t \xi_z]
\]  

(17)

Comparing the vector of conserved variables and the convective portion of the inviscid flux vector to the new grid movement terms (equation 18), the grid movement terms modify to the convective velocity in the inviscid flux and thus, may be incorporated into an existing fixed mesh upwinding scheme. In a transformed system, the grid movement term becomes part of the contravariant velocity as shown below. Thus, the unsteady grid transformation term may be implemented as a velocity correction in an existing upwinding scheme. The difference in the energy equation term results in a pressure component being subtracted from the last equation to convert from enthalpy to energy.
Now consider the general inviscid flux in the $\xi$ direction from a steady transformation written as a sum of a convective and pressure components:

$$E^{',} = E^{c} + E^{p} = J|\nabla \xi| \rho \hat{u} \tilde{E}^{c} + J|\nabla \xi| p \tilde{E}^{p}$$ (19)

where

$$\tilde{E}^{c} = \begin{bmatrix} 1 \\ u \\ v \\ w \\ H \end{bmatrix}, \quad \tilde{E}^{p} = \begin{bmatrix} 0 \\ \xi_x \\ \xi_y \\ \xi_z \\ 0 \end{bmatrix},$$ (20)

$$H = \frac{1}{\rho} (\rho_o + p) \quad \text{and} \quad \tilde{\xi}^{x, y, z, t} = \frac{\xi_x, y, z, t}{|\nabla \tilde{\xi}|}.$$ (21)

The steady contravariant velocity is given as

$$\hat{u} = \tilde{\xi}_x u + \tilde{\xi}_y v + \tilde{\xi}_z w.$$ (22)

Incorporating the $U\tilde{\xi}_t$ term into the convective portion of the inviscid flux requires calculating an unsteady transformation contravariant velocity, $\hat{u}^{ut}$ as follows:

$$\hat{u}^{ut} = \tilde{\xi}_x u + \tilde{\xi}_y v + \tilde{\xi}_z w + \tilde{\xi}_t.$$ (23)

Using $\tilde{\xi}_t$ as defined in equation 17, the unsteady transformation contravariant velocity may be further simplified:
Now, consider the resulting unsteady transformation convective flux,

\[ \hat{u}^{ut} = \xi_x (u - x_t) + \xi_y (v - y_t) + \xi_z (w - z_t). \]  

(24)

which using the definition of contravariant velocity, \( \hat{u} \) and equation 25 above can be written as

\[ E_{ut}^c = J|\nabla \xi| \rho \hat{u}^{ut} \tilde{E}^c = J|\nabla \xi| \rho [\xi_x (u - x_t) + \xi_y (v - y_t) + \xi_z (w - z_t)] \tilde{E}^c. \]  

(25)

With the exception of the energy term, the unsteady transformation convective flux calculated with \( \hat{u}^{ut} \) achieves the desired goal of including the grid speed terms \( (U \xi_t) \) in the original fixed mesh upwinding formulation. Comparing the energy term in equation 26 to the desired term \( (e_o \xi_t) \) in equation 18, one is energy while the other is enthalpy, thus they differ by a pressure term. To convert enthalpy to the desired energy term \( (e_o \xi_t) \), a pressure term \( (p \xi_t) \) must be subtracted from the energy term in the pressure component of the upwinding. The resulting unsteady transformation pressure flux is as follows:
In summary, to incorporate the unsteady transformation grid speed terms into an existing upwinding, the only changes required are to use an unsteady transformation contravariant velocity (equation 25) and an unsteady transformation pressure flux (equation 27). The upwinding scheme chosen for the present work is the Low-Diffusion Flux-Splitting (LDFSS) of Edwards.\(^{19}\)

\[
E_{ui}^p = J|\nabla \tilde{\xi}| p \begin{bmatrix}
0 \\
\tilde{\xi}_x \\
\tilde{\xi}_y \\
\tilde{\xi}_z \\
\tilde{\xi}_t
\end{bmatrix} = J|\nabla \tilde{\xi}| p \begin{bmatrix}
0 \\
\tilde{\xi}_x \\
\tilde{\xi}_y \\
\tilde{\xi}_z \\
\tilde{\xi}_t
\end{bmatrix} \begin{bmatrix}
\tilde{\xi}_x \\
\tilde{\xi}_y \\
\tilde{\xi}_z \\
\tilde{\xi}_t
\end{bmatrix}.
\]  

(27)
2.3 Adaptive Grid Algorithm

The truncation error of a finite difference solution scheme is a function of solution derivatives and grid spacing. The truncation error is large near discontinuities and rapid changes in local flow variables. Solution dynamic grid adaptation can be employed to improve overall solution accuracy by reducing grid spacing near regions of high flow gradients and/or curvature. The gradients and curvature of solution variables represent a reasonable basis for a weighting function in a center-of-mass adaptation algorithm designed to reduce solution truncation error. The center-of-mass adaptation algorithm was originally proposed by Eiseman for use in physical space. The center-of-mass algorithm applied in physical space is subject to grid crossover when applied near concave or convex boundaries. Benson and McRae alleviated the crossover problem by adapting in a parametric adaptation space - essentially a computational space grid with unity spacing in all directions. The actual movement of grid points is performed in a parametric adaption space, using a weight function defined at each vertex as a point mass in the adaptive grid algorithm. The weight function is a combination of gradients and/or curvature of flow variables. It is desirable to have as much control of the adaptation process as possible, thus the weight function is defined independently for each coordinate direction and the movement of grid points is then performed for each direction simultaneously. Once the grid is satisfactorily redistributed in the parametric space, a simple, one-step, re-transformation equation is used which returns parametric space to the physical space. The final result is that grid points will be clustered in regions of large gradients and/or curvature in flow variables.
2.3.1 Orthogonal Adaptive Grid Algorithm

An improved point movement algorithm for three-dimensional solution-adaptive gridding applications is proposed as an alternative to the center-of-mass formulation used in earlier work.12-15,16,20,25,26 Likewise, adaptation is performed in parametric space, essentially defining an interpolant for moving physical space grid points to their new locations. The original center-of-mass method, when applied within a parametric space, gives excellent adaptation to inviscid flow features when starting from uniform grids, but can result in excessive grid skewness and grid crossover when starting from the highly clustered grids required to resolve turbulent boundary layers. The original center-of-mass method is modified in two ways as follows. The diagonal nodes are ignored which allows formulation of center-of-mass partial differential equations (equations 28-30 with unity coefficients), and orthogonality restoring coefficients are added in the non-adapting directions.

Defining the interpolation variables corresponding to the computational $\xi$, $\eta$ and $\zeta$ directions as $p$, $q$ and $r$, the new orthogonal center-of-mass partial differential equations of the following form are approximately solved in parametric space to facilitate adaptation:

\[
\frac{\partial}{\partial \xi} \left( \omega_p \frac{\partial p}{\partial \xi} \right) + \left( \frac{\nabla \eta \cdot \nabla \xi}{|\nabla \xi||\nabla \eta|} \right) \frac{\partial}{\partial \eta} \left( \omega_p \frac{\partial p}{\partial \eta} \right) + \left( \frac{\nabla \zeta \cdot \nabla \xi}{|\nabla \zeta||\nabla \xi|} \right) \frac{\partial}{\partial \zeta} \left( \omega_p \frac{\partial p}{\partial \zeta} \right) = 0 \quad (p \text{ interpolant; } \xi \text{ direction})
\]

\[
\left( \frac{|\nabla \xi|}{|\nabla \eta|} \right) \frac{\partial}{\partial \xi} \left( \omega_q \frac{\partial q}{\partial \xi} \right) + \frac{\partial}{\partial \eta} \left( \omega_q \frac{\partial q}{\partial \eta} \right) + \left( \frac{\nabla \zeta \cdot \nabla \eta}{|\nabla \zeta||\nabla \eta|} \right) \frac{\partial}{\partial \zeta} \left( \omega_q \frac{\partial q}{\partial \zeta} \right) = 0 \quad (q \text{ interpolant; } \eta \text{ direction})
\]
In this, $\omega_{p,q,r}$ are the weight functions for the $p, q, r$ interpolants. The point movement equations are discretized in a fixed parametric space using standard central-difference techniques, assuming that $\nabla \xi = \nabla \eta = \nabla \zeta = 1$. The equations are then updated by a point Gauss-Seidel or Jacobi iteration subject to a reflection boundary condition and a final smoothing pass with coefficients set to unity. Adaptation is performed every time step for unsteady calculations. Point movement is accomplished through a remapping procedure developed in earlier work\textsuperscript{15} and is performed relative to a fixed background mesh. The orthogonal center-of-mass partial differential equations have non-unity coefficients of some second-derivative terms. If the new orthogonality coefficients are set to unity, the algorithm reverts to center-of-mass partial differential equations with performance similar to the original center-of-mass formulation. Considering equation 28 above, the first non-unity coefficient represents a stretching factor $(|\nabla \eta|/|\nabla \xi|)$ raised to an orthogonality factor $(|\nabla \xi \cdot \nabla \eta|/(|\nabla \zeta||\nabla \eta|))$. These new coefficients act in the non-adapting directions to weight the point movement as required to maintain or restore acceptable orthogonality. The stretching factor is a cell face area ratio squared, which is much greater than unity for the highly stretched (high aspect ratio) cells required to resolve turbulent boundary layers. The orthogonality factor is the absolute value of the dot product of surface unit vectors. It is a skewness measure and varies between zero and unity.
Consider a highly stretched and highly skewed cell with a near-unity orthogonality factor and a very large coefficient. The large coefficient dominates the other adaptation direction’s coefficients and tends to restore orthogonality. The orthogonality restoring point movement algorithm’s behavior in parametric adaption space is illustrated in Figure 1, “Orthogonal Adaptation Process Near Boundaries”. The skewness shown in parametric space would be excessive when mapped back to physical space on a highly stretched mesh. Previous center-of-mass algorithms can easily produce this type of skewness near a boundary due to considering only weight functions. The grid is highly skewed at time 1 in the $\eta$ direction adjacent to a boundary. The $\eta$ stretching factor is very large and raised to a near unity orthogonality factor which makes the $\eta$ coefficient much larger than the $\zeta$ coefficient. When the point under consideration at location 1 is moved in $\zeta$ direction subject to a reflection boundary condition, the $\eta$ direction dominates, and the point moves to location 2. The final result is a nearly orthogonal grid in parametric space which maps back to acceptable orthogonality in physical space. The new orthogonal procedure significantly reduces skewness and crossover problems in viscous regions while maintaining high resolution in the capturing of inviscid flow features.

If the grid is nearly orthogonal, the orthogonality factor is near zero, and the resulting coefficients are near unity regardless of stretching factor, causing the adaption algorithm to revert to a center-of-mass partial differential equation with unity coefficients. As desired for nearly orthogonal cells, adaptation is allowed to proceed as a function of the weight functions. Likewise, when a cell’s stretching factor is near unity, the orthogonality factor is irrelevant, and the center-of-mass partial differential equation formulation is
returned. In summary, two conditions return a center-of-mass partial differential equation which allows adaptation unconstrained by orthogonality considerations: an orthogonal grid or a rhombus-type cell. If the grid is orthogonal, adaptation should be allowed to proceed. Rhombus-type cells are indicative of grids in inviscid flow regions where skewness is allowable as it affords better grid alignment with inviscid features.

2.3.2 Weight Function

Mesh movement using the concept of a parametric space center-of-mass calculation, as developed by Benson and McRae\textsuperscript{12,15,25,26,27}, first requires calculating a weight function for each adaptation direction. The weight function is constructed as a linear combination of gradients and/or curvature of the dependent variables. The gradient of a variable is calculated as a central-difference approximation of a first derivative, and the curvature is a second derivative approximation. The weight function is formed as a linear superposition of the gradients and curvature where $\sigma_k$ is a biasing coefficient, $\phi_k$ is the magnitude of the curvature, $\phi_k$ is the magnitude of the gradient, $\alpha_c$ is a curvature weighting coefficient, and $k$ indicates a given flow variable. The weight function is scaled by the freestream values of the dependent variable, $u_k$, to determine the relative importance of the local flow consistent with the overall flow field.\textsuperscript{20} The velocity weight function is not scaled by the freestream velocity due to the local velocity approaching zero in the boundary layer and at stagnation points. The Jacobian raised to $\beta$ in the weight function provides control of minimum and maximum cell volumes while adapting.\textsuperscript{16}
After the raw weight function is calculated as shown in 31, it is smoothed and subjected to a minimum and maximum limit. The smoothing and weight function limiting prevents over adaptation to strong features, adapting to minutiae, and evacuating uniform flow regions. The upper and lower limits for the weight function are specified as a multiple/fraction of the average weight function. The lower limit is set to control the maximum cell spacing in uniform flow regions, and the maximum limit is set to control the minimum cell spacing in regions of large weight functions.

\[ \omega = \sum_k \frac{\sigma_k(\alpha_c \phi_k + (1 - \alpha_c)\phi_k)}{u_k} \times j^\beta \]  

(31)
3. RESULTS AND DISCUSSION

3.1 Riemann Problem

The Riemann (shock tube) problem appears simple but contains most of the important transient features one desires to capture with a time-accurate algorithm. The shock tube problem is useful for code validation due having an exact solution for comparison. For unsteady validation of time accuracy subiterations and unsteady mesh terms, the three-dimensional Riemann problem is solved with and without adaptation. The computational grid is uniform (201x33x33) with a length of five and unit length sides. Although the inviscid shock tube is inherently one-dimensional, solving the problem in three-dimensions is a useful code validation tool to ensure the one-dimensional nature is retained. The diaphragm is at the mid-point, and the pressure and density ratios are ten to one with constant temperature. All calculations are performed over a total time of unity with a physical time step of 0.01 unless otherwise stated. The physical time step corresponds to an explicit inviscid CFL condition of one. The following sections detail sequentially the results of the individual improvements incorporated into the original algorithm of Edwards.¹⁸
3.1.1 Fixed-Grid Riemann Solutions

3.1.1.1 Subiterations

The first modification to the implicit code of Edwards is to add time accuracy improving subiterations. The time difference in equation 7 is added to the residual calculation, a subiteration loop is added, and a global physical time step is specified. In order to evaluate the temporal accuracy of the subiterations on a fixed grid, the solutions using various numbers of subiterations are compared to the exact solution. For comparison, the solution from the original code of Edwards is also plotted in Figure 2, “Shock Tube Solution for Various Subiterations,” and Figure 3, “Shock and Contact Surface for Various Subiterations”. The locations of the contact surface and shock show the original algorithm has clearly lost time accuracy. The values of density and pressure are also incorrect. Time-accuracy errors in the original algorithm are expected due to linearization, factorization, and relaxation errors.

The subiterations are intended to restore time accuracy. In order to determine the number of subiterations required to restore time accuracy with acceptable computational effort, a series of calculations was performed with increasing numbers of subiterations. The one subiteration case improves slightly upon the original solution, and the shock speed error is reduced by half. The overall solution errors in pressure and density are reduced. As expected for a second order accurate algorithm, oscillations appear in the solution. Two subiterations show continued improvement in shock speed, pressure and density with reduced oscillations. Three and four subiterations are virtually identical with only a slight improvement over two subiterations. At three subiterations, the shock and
contact surface are captured well, and the shock speed error is less than one percent. For comparison, a solution was performed reducing each subiteration residual ten orders of magnitude, and the greater convergence provides no noticeable improvement over the three subiteration case.

3.1.1.2 Relaxation Parameters

According to Matsuno\textsuperscript{23}, a second order time-accurate scheme should require two subiterations at most. Thus, even with a zeroth order initial guess, one should obtain second order accuracy after only two subiterations. Since each subiteration is rather computationally expensive, minimizing subiterations is a driving concern in developing a practical time-accurate flow solver. Unfortunately, two subiterations do not seem to perform admirably, as three subiterations were previously required to approach the more converged residual solution. The original code of Edwards has forward and backward sweep update relaxation parameters which are less than unity for optimal multigrid steady state convergence. When time accuracy is desired, all relaxation parameters should be set to unity. The results after setting all relaxation parameters to unity are presented in Figure 4, “Unity Relaxation Factors”. The results for two subiterations with the relaxation parameters set to unity are shown to be capable of producing results comparable to the solutions with many more subiterations.

3.1.1.3 Minmod Limiter

The original code of Edwards employed a pressure-based limiter for steady state convergence which may be too oscillatory for the three point backward time integration
scheme. In an attempt to reduce oscillations near discontinuities, a Minmod limiter option was implemented. Figure 5, “Minmod Limiter,” shows the results of the Minmod limiter compared to the exact solution and the previous pressure based limiter results. The Minmod limiter reduced or eliminated all oscillations. The resulting shock speed error is also reduced to about 0.7 percent.

### 3.1.1.4 Time Step Size

In order to evaluate the new algorithm’s response to various physical time steps, additional solutions were calculated with CFL conditions corresponding to one half and two, time steps of 0.005 and 0.02 respectively. Figure 6, “Shock Tube Solutions with Various Time Steps,” and Figure 7, “Shock and Contact Surface for Various Time Steps,” showing the results compared to the previous baseline solution with a time step of 0.01, an explicit CFL of one. All calculations used two subiterations, relaxation parameters of one, and the Minmod limiter. The errors involved in a second order accurate in time scheme should be a function of the square of the time step.

The baseline CFL of unity solution seen before has a shock speed error of about 0.7 percent. As expected, reducing the time step to 0.005 results in reduced error, and the resulting shock speed error is less than 0.2 percent. Also, the resulting solution improves the capturing of the moving discontinuities and is free of any oscillations. Increasing the time step to 0.02 produces the opposite effect as the shock speed error increases to about two percent, and the oscillations grow. Since the error is a function of the time step, the physical time step size is reflected in the overall quality of the solution.
3.1.2 Adaptive Riemann Problem

Previous results showed the subiteration algorithm to be time accurate on a fixed grid. Dynamically adapting mesh shock tube calculations are required to verify the time accuracy of the unsteady grid movement terms which were incorporated into the upwinding scheme (LDFSS). The weight function for the adaptive grid calculations was calculated with all biasing coefficients set to unity, a minimum limit of 0.1, and a maximum limit of 100. A curvature weighting coefficient of zero was used to allow adapting only to gradients which provided better resolution of the moving features. After smoothing of the weight function prior to use, the curvature contribution to the weight function tended to spread the weight function beyond the important features. The time step was 0.005 and the grid was adapted during every time step.

The shock tube solutions in Figure 8, “Adapted versus Unadapted Shock Tube Results,” show the flow solver algorithm to be time accurate for the adapted calculations. The shock speed error is less than half a percent and is mainly attributed to the initialization error on a uniform grid. The adapted solution is resolved much better, with the adapted normal shock being a nearly vertical line containing only three grid points. The contact surface shows less improvement from the unadapted to the adapted case due to dissipation added by the upwinding scheme to a moving contact surface. The added dissipation prevents the extra points provided by dynamic adaptation from more significantly sharpening the contact surface.

Figure 9, “Adapted Cell Volume and Grid Speeds,” shows the corresponding cell volume normalized by the initial uniform cell volume and grid speed for the adapted solu-
tion. Near the normal shock, the grid speed is larger than the local fluid velocity. The expansion, contact surface, and normal shock can be clearly identified by the low cell volumes and higher grid speeds, indicating the movement of points into these regions as dictated by the weighting function. The minimum volume occurs at the normal shock and is about 13 percent of the original cell volume. The largest cells are only about twice the size of the original cells, and further point evacuation is undesirable for an unsteady problem. Further point evacuation leads to excessive grid speeds when an unsteady feature moves into an evacuated region, and the excessive grid speeds can cause excessive dispersion and/or dissipation.\textsuperscript{14}

### 3.1.3 Computational Efficiency and Accuracy

The Riemann problem has an exact solution which can be compared to the adapted and fixed mesh solutions. The error vector is calculated as the difference between the primitive variable solution vector and the exact solution. The L2 Norm of the error vectors are calculated and shown are shown in Table 1. The baseline case is a fixed mesh solution with 201 points in the X direction. For comparison, an adapted solution and refined

<table>
<thead>
<tr>
<th>Case</th>
<th>L2 Norm</th>
<th>CPU Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>201 points without adaptation</td>
<td>0.1538</td>
<td>319</td>
</tr>
<tr>
<td>201 points with adaptation</td>
<td>0.0659</td>
<td>322</td>
</tr>
<tr>
<td>401 points without adaptation</td>
<td>0.1200</td>
<td>636</td>
</tr>
<tr>
<td>801 points without adaptation</td>
<td>0.1072</td>
<td>1301</td>
</tr>
</tbody>
</table>
mesh solution error norms are also presented. The dynamic solution adaptive algorithm shows significant improvement over the baseline fixed mesh solution by reducing the error by more than 57 percent. For the shock tube calculations, the overhead of the adaptation algorithm is minimal at less than one percent. Refined mesh calculations are presented in an attempt to determine the Central Processing Unit (CPU) time required for a similar reduction in error on a fixed mesh. Taking advantage of the one-dimensional nature of the problem, the static grids are only refined in the X direction. Obviously for a general three-dimensional problem, the CPU time would grow geometrically instead of linearly as the mesh is refined. Increasing mesh nodes to 401 requires approximately doubling the CPU time for less than a 22 percent reduction in error. Doubling points again to 801 quadruples the original CPU time while only reducing the error norm 30 percent. For the Riemann problem, the solution dynamic grid algorithm provides significantly more error norm reduction per CPU cycle than mesh refinement.
3.2 Scramjet Inlet

3.2.1 Three-dimensional Inlet Steady Results

Steady solutions were obtained for a three-dimensional inlet/isolator component of a dual-mode scramjet geometry described by Emami et al. The experiments were performed in the Mach 4 Blowdown Facility (M4BDF) at NASA Langley Research Center. Figure 10, “Dual-Mode Scramjet Inlet Model,” illustrates the geometry and sensor locations of the experimental apparatus. Pressure data were collected along the bottom centerline and at the exit for various positions of the rear flap. The full open position of the rear flap corresponds to the 1.31 inch open position, and the closed position is at the 0.34 inch position. The steady calculations were performed with the rear flap in the 0.7 inch open position - the last position prior to the inlet unstarting in the experiment. The 0.7 inch open position case was chosen to locate the final shock near the exit of the isolator. The total inlet length is 32.7 inches with a sidewall beginning at 5.18 inches. The isolator extends from 9.77 inches to 13.25 inches. The original and adapted mesh steady solutions are presented, and the adapted mesh solution will be used as the initial solution for calculating inlet unstarts. The calculation takes advantage of the symmetry about the centerline (k=33). The original mesh was used as the background grid for the adaptation process. The weight function was calculated with unity pressure biasing coefficients, 0.01 density biasing coefficients in $\eta$ and $\zeta$, a minimum limit of 0.3, and a maximum limit of 7.0. Turbulence closure is provided by a modified version of the Spalart-Allmaras model. The one-equation model is updated in a weakly-coupled manner after each iteration.
The solution on the original fixed mesh converged three and one half orders of magnitude. Judging convergence on a moving mesh calculation can be difficult due to the small perturbations from adaptation, so convergence of adaptive mesh solutions was judged by the centerline pressure trace. The convergence for an adapting grid calculation based on residual was only about one order of magnitude due to perturbations from solution dynamic grid adaptation. When allowed to converge on a fixed adapted grid, the residual dropped four orders of magnitude from the initial residual which is slightly more converged than on the original unadapted mesh.

The inlet geometry was represented by the original 325x81x33 mesh as shown in Figure 11, clustered to the walls to resolve turbulent boundary layers. Figure 11 presents pressure contours and mesh sections for constant-index surfaces (i=20, j=40, k=33) of the three-dimensional solution on the original unadapted mesh. The inlet ramp generates an initial oblique shock followed by a cowl shock which reflects into the isolator. The aft diffuser is a large subsonic region dominated by a large recirculation region. The pressure contours show the shocks and expansions entering the isolator to be poorly resolved features on the original grid.

The adapted grid and pressure contours are presented in Figure 12. As shown for the k=33 centerplane, the adaptation algorithm results in a crisp capturing of the initial planar shock structure and the reflected shocks entering the isolator. Interpretation of the results at the i=20 and j=40 surfaces is somewhat difficult, as the mesh also deforms in the X and Y directions. It can be noted that the adaptation algorithm does respond to lateral gradients induced by the boundary layer on the inlet sidewall.
The orthogonal modifications to the adaption algorithm can be seen when compared to results from a previous version of the algorithm. No steady results from a previous adaption algorithm are presented due to excessive grid skewness which induces flow solver instabilities. Figure 13, “Grid from Previous Adaptation Algorithm,” shows adaptation from a previous version of the adaptation algorithm. A very low level of adaptation exists even at the initial strong oblique shock. The adaptation shown in Figure 13 results in the excessive skewness near a viscous boundary as shown in Figure 14, “Mesh Near Inlet Boundary for Original (left) and Orthogonal (right) Mesh Algorithms”. Figure 14 compares the adapted grid near the wall in the inlet region for the original and orthogonal mesh algorithms. The previous algorithm produces excessive skewness near the viscous boundary, and the excessive skewness impacts flow solver stability prior to achieving acceptable adaptation. By comparison, the orthogonal mesh algorithm results in a smooth, nearly orthogonal grid near the viscous wall.

Figure 15 provides a close-up view of the centerplane flow structure at the aft inlet and isolator regions for the original and adapted mesh cases. Comparing the two, the adapted cowl shock is more crisply captured, as are its initial reflections from the inlet surface. The isolator reflected shock and expansions are also better resolved on the adapted grid. The oblique shock train that propagates further into the isolator is not particularly well-captured by either case, but is completely lost in the unadapted solution. By design, as the shock train reflects down the isolator, the shock strength becomes very weak resulting in small weight function magnitudes which do not induce significant grid adaptation.
The adapted mesh better resolves the shock at the exit of the isolator and locates it further aft than the original mesh.

Wall pressure distributions at the bottom wall centerline location are shown in Figure 16. The magnitude of the pressure rise resulting from the cowl shock interaction with the bottom surface is predicted by both the adapted and unadapted mesh calculations. In the isolator shock reflections, the benefits of solution adaptation are seen more clearly, as the peak shock pressures are captured more precisely as shown in Figure 17. The adapted solution isolator exit shock location (x = 13.25) is in better agreement with experimental results. The unadapted solution exit shock is too far forward and fails to detect the last expansion and shock as seen in the experimental data.

The three-dimensional nature of the solution adaptive mesh algorithm is illustrated in Figure 18, “Adapted 3-D Mesh Surfaces (i = 30, j = 40, k = 28)”. Inspecting the i and j surfaces reveals little in surface adaptation to the initial oblique shock, but the i and j surfaces have adapted in such a way as to provide alignment with the initial oblique shock. The overall three-dimensional adaptation results in the initial oblique shock being resolved well as seen in the k surface. Aft of the i surface, the j surface mesh shows adaptation to the weak shock arising from the sidewall. Following the j and k surfaces through the isolator reveals significant three-dimensional adaptation to even the weak features found in the isolator.

Figure 19 and Figure 20 show density contours and adapted mesh on the bottom wall, centerline and selected i surfaces (i = 20, 50, 65, 110, 180). The flow features found in the density plot can be seen reflected in the adapted mesh. Inspection of the i surfaces
reveals two modes of adaptation: alignment from surface movement and reduction in grid spacing due to movement within a surface. Adaptation via alignment is more prevalent and desirable in inviscid regions and is best seen at the initial oblique shock location. The other i surfaces show more adaptation within the i surface. Adaptation within a surface is more dominant in viscous regions. The initial oblique shock has a spurious reflection off the upper boundary which is resolved and visible in the centerline plane and the second i surface. The upper cowl surface shock can be seen in the centerline plane and the third i surface. The train of weakening reflected shocks can been seen in the centerline plane. The final three i surfaces are aft of the sidewall location, and three-dimensional effects are noticeable near the wall. The i surfaces show distortions in the reflected shock train near the sidewall due to vortex and separation regions.

3.2.2 Computational Time

Implicit algorithms by nature are more computationally intense than explicit flow solvers. Adding time-accurate subiterations to an implicit algorithm even further increases the computational time required per time step. A typical explicit algorithm on a CRAY T90 takes about 9.0e-6 seconds per point per time step, whereas the modified time accurate algorithm developed here requires about 5.0e-5 seconds. The implicit time accurate scheme requires over six times the computational time per time step, but on the inlet-isolator-diffuser model under consideration, the implicit time step being used is about 70 times the explicit time step limit. Therefore, for the given case, an overall speed up of about fourteen is observed. Determining the allowable physical time step is a balance
between computational time and errors resulting from a large time step. Exceeding the inviscid CFL of unity condition results in higher subiteration residuals, and thus provides feedback for proper time step choice. The relative speed up when using time accurate sub-iterations is problem and error tolerance dependent.

Previously, the adaptation algorithm overhead for the shock tube problem was noted to be around one percent on a CRAY T90. The three-dimensional inlet calculations were preformed on a Compaq DEC Alpha. A two subiteration time step without adaptation requires approximately 77 seconds, versus 80 seconds with adaptation. Thus, the adaptation algorithm’s overhead for the inlet under consideration is about 4 percent.

3.2.3 Three-Dimensional Inlet Unstart Calculations

3.2.3.1 Exit Flap Induced Unstart

The experimental geometry was evaluated in the wind tunnel to determine the maximum back pressure the inlet could sustain before unstarting. Closing the diffuser flap to the 0.34 inch position induced an inlet unstart in the experiment. The final experimental case was the 0.34 inch open position. For the unstart calculations, the exit flap was moved from the 0.7 to 0.34 inch open position, and the exit pressure boundary condition raised. The diffuser flap movement was accomplished by moving the background grid which allowed the existing parametric space mapping to easily move the adapted grid. The initial condition for the unstart calculation is the adapted solution previously presented. For clarity in some of the following figures, only the inlet and isolator are shown.
A close up of the centerplane isolator with velocity vectors is also included. To enhance visibility, only every other velocity vector is shown.

Moving the diffuser flap and raising exit pressure sent a pressure wave forward through the diffuser region. Pressurizing the diffuser region requires approximately two milliseconds and little change was noted in the isolator during the initial pressurization time as seen in Figure 21, “Exit Flap Unstart (0.0 ms and 1.65 ms)”. At almost 2.5 milliseconds, about half way through the unstart, the pressure increase is evident in the isolator as seen in the pressure contours and shock reflection angles of Figure 22. The unstart disturbance appears to temporarily stall in the middle of the isolator as the isolator functions as designed to isolate the inlet from downstream perturbations. At time 2.47 ms, the pressure wave passing forward has increased the angle of the incident and reflected shocks which increases the severity of the shock/boundary layer interactions in the isolator. The separation region near the front lower surface of the isolator is growing. Figure 24 presents a three-dimensional view of a pressure-coded adapted mesh and shock surfaces at time 2.87ms. The shock surfaces are located by a feature detection algorithm included in a pre-release version of CEI’s Ensight visualization software package. The feature extraction algorithm is still being tuned by CEI and may over or under estimate the extent of the shock wave structures. Eventually viscous effects dominate, and the shock wave structure in the isolator deteriorates. The shock/boundary layer interactions grow and accelerate the unstart process. The growing separation and recirculation regions reduce the available flow area, choking the flow and further accelerating the unstart process. At 4.26 ms, the lower surface isolator separation region has grown and moved forward to the inlet region.
A separation region has developed on the cowl which is also moving forward. Figure 25, “Exit Flap Inlet Unstart in Progress II (3-D view),” shows the shock surfaces being lifted and pushed forward ahead of the separation regions. At 4.67 ms (Figure 23) the unstart is accelerating as the shock structures being pushed forward by the separation regions are growing to fill the available flow area. The unstart is nearing upper cowl spillage at time 5.49 ms. Figure 26, “Exit Flap Inlet Unstart after Upper Cowl Spillage (3-D view),” shows the final three-dimensional shock structure. A large oblique shock is located outside the cowl and ahead of a large recirculation region. A supersonic jet exists down the center of the isolator, as seen in the small diamond shocks in the center of the isolator. A region of reverse flow exists between the supersonic jet and the sidewall in the isolator. The unstart takes approximately 5.74 milliseconds from initiation to upper cowl spillage. After upper cowl spillage, the inlet is considered to be unstarted. Exit pressure drops significantly due to the reduced efficiency of the inlet shock system.

Initially the exit flap unstart process is dominated by wave propagation and inviscid flow dynamics. A pressure wave moves forward due to the closing of the exit flap and resulting rise in exit pressure. The pressure wave moves forward and appears to stall in the isolator. The isolator functions, as designed, to isolate the inlet from downstream disturbances. The isolator appears to be mitigating the unstart, but the shock angles in the isolator are increasing, which results in a more severe shock/boundary layer interaction. The second half of the inlet unstart process appears to be dominated by shock/boundary layer interactions, which lead to large separation and recirculation regions. The separation/recirculation zones reduce the available flow area and choke the inlet. Thus, the
shock/boundary layer interactions appear to accelerate the unstart and have a deleterious effect on inlet stability.

3.2.3.2 Angle of Attack Unstart

Although no freestream perturbation experimental data exist for the geometry under consideration, an angle of attack unstart was calculated as a code demonstration exercise. The inlet was subjected to an angle of attack perturbation of 10 degrees ramped over 0.02 ms. The steady, adapted solution previously discussed is used as an initial condition. The perturbation and resulting unstart is predominately a result of the inlet and isolator disturbances.

Figure 27 shows the inlet 0.137 ms into the angle of attack transient. The transient is apparent in the initial oblique shock as the freestream angle change is feeding into the inlet via the front and top (which is actually bottom of inlet) boundaries. The perturbation front has feed into the middle of the initial oblique shock and can be seen in the curved oblique shock. The dynamic solution adaptive mesh algorithm adapts the mesh automatically as required to follow the transient. The new freestream angle results in a steepening of the inlet oblique shock which raises pressure behind the oblique shock. The increasing pressure wave behind the oblique shock can be seen in the pressure contours. The perturbation has not reached the cowl and isolator regions. The small separations present after the shock reflections on the bottom surface are present in the steady initial condition (Figure 21).
The next unstart frames are shown in Figure 28. At 0.275 ms, the freestream angle perturbation has progressed further into the inlet as the inlet oblique shock only retains a slight curvature at the freestream boundary. The pressure contours show the resulting pressure wave to be approaching the cowl region, but no changes are yet noticeable in the isolator region. Later, at time 0.686 ms, the perturbation pressure wave has moved past the inlet and into the isolator region. The inlet oblique shock and adapted mesh are now straight and correspond to the new oblique shock angle. The pressure wave is causing significant changes in the cowl and isolator region. The cowl shock angle is increasing which indicates a strengthening shock. The cowl shock angle change moves the lower surface reflection forward causing a more severe shock/boundary layer interaction and increased separation. The first upper surface shock reflection is moving forward on the upper cowl. The next series of timesteps are shown in Figure 29. At 1.37 ms, the freestream angle perturbation has transited the isolator region. The reflected cowl shock separation region is growing and lifting the reflection further from the lower surface. The pressure increase has continued to increase the strength of the reflected shock train. Increasing the strength of an oblique shock increases the shock angle which tends to shorten a reflected shock train. The angle between the cowl and the upper isolator tends to anchor an oblique shock which prevents the shock train from migrating forward. The shortening of the existing shock train allows the formation of an extra shock reflection on the upper and lower surface prior to the isolator. At 2.06 ms, the cowl shock reflection separation region continues to grow and move forward. The first reflection on the cowl
continues to strengthen and has now formed a separation region. The aft cowl and isolator flow fields now consist of an oblique shock train with few distinct features.

Figure 30 shows the final stages of the unstart. At 2.75 ms, the first shock reflections on the inlet and cowl have decoupled from the isolator flow field. The separation region behind the cowl shock/boundary layer interaction has grown to fill approximately half the flow channel and is rapidly expelling the shock structure. The reflected shock separation region on the cowl is also growing but appears secondary as an unstart mechanism. At 3.43 ms the driving separation has nearly filled the available flow area, and the shock generated by the separation region is rapidly moving toward the upper cowl lip. By 4.12 ms (Figure 31), the inlet has upper cowl spillage and is considered to have unstarted. The flow field is dominated by the large recirculation region on the lower surface of the inlet, the shock created by the recirculation region, and several normal or near-normal shocks in the cowl and isolator regions. In the unstarted condition, pressure recovery is severely impacted by the normal shocks, and the inlet/engine assembly would provide little thrust until the inlet is restarted.

Again, the overall unstart can be divided into two distinct phases: a wave propagation and a shock/boundary layer dominated phase. During the first half of the unstart, the freestream perturbation propagates through the inlet and isolator regions raising pressure and changing shock angles. The steeper shock angles result in more severe shock/boundary layer interactions which lead to the second phase of the unstart. The second phase of the unstart is characterized by the formation of large separations and recirculation regions which ultimately unstart the inlet.
4. CONCLUSIONS

A time-accurate, implicit flow solver coupled to a solution adaptive grid algorithm has been developed. The development of a new orthogonality version of the adaptive-mesh procedure of McRae and co-workers coupled to a time-accurate implicit upwind-relaxation Navier-Stokes solver has been accomplished. Shock tube simulations show the effectiveness of the algorithm for unsteady flow situations on fixed and moving grids. The implicit algorithm requires more computational time per timestep than an explicit algorithm, but the less severe time step restriction provides a significant overall computational speedup. The solution dynamic adaptation algorithm is shown to consume negligible computational resources while providing significant reductions in error - especially when compared to global mesh refinement. When compared to fixed grid, explicit, time-accurate algorithms, the new algorithm will allow either better solution resolution for a fixed memory and time requirement or a similar solution with fewer grid points in less time - or allow any compromise between the two extremes. Steady and unsteady results were obtained for a three-dimensional Dual-Mode Scramjet Inlet flow field. The steady-state results show that the current adaptive gridding method enhances the resolution of complex flow features found in turbulent three-dimensional flow fields. Inlet unstarts due to exit and freesteam perturbations were calculated. The time-accurate simulations demonstrate the ability to calculate three-dimensional high-speed inlet unstarts with an adaptive mesh algorithm. The unstarts are characterized by a wave propagation phase followed by a shock/boundary layer interaction phase. The negative impact of the shock/
boundary layer interaction on inlet stability appears to validate the initial assumption that
turbulent three-dimensional calculations are required for inlet unstart analysis. The three-
dimensional unstart simulations should give additional insight into the shock/boundary
layer mechanisms which can reduce inlet stability, thereby reducing maximum back pres-
sure and available thrust. The final three-dimensional adaptive mesh algorithm coupled to
a time-accurate, implicit Navier-Stokes solver provides an efficient, accurate means of
simulating a three-dimensional inlet unstart which should make large three-dimensional
inlet simulations more tractable.
5. References


FIGURE 1. Orthogonal Adaptation Process Near Boundaries

FIGURE 2. Shock Tube Solution for Various Subiterations
FIGURE 3. Shock and Contact Surface for Various Subiterations

FIGURE 4. Unity Relaxation Factors
FIGURE 5. Minmod Limiter

FIGURE 6. Shock Tube Solutions with Various Time Steps
FIGURE 7. Shock and Contact Surface for Various Time Steps

FIGURE 8. Adapted versus Unadapted Shock Tube Results
FIGURE 9. Adapted Cell Volume and Grid Speeds

FIGURE 10. Dual-Mode Scramjet Inlet Model
FIGURE 11. Unadapted Grid and Pressure Contours

FIGURE 12. Adapted Grid and Pressure Contours
FIGURE 13. Grid from Previous Adaptation Algorithm

FIGURE 14. Mesh Near Inlet Boundary for Original (left) and Orthogonal (right) Mesh Algorithms
FIGURE 15. Grid and Pressure Contours in Aft Inlet and Isolator (Centerline)

FIGURE 16. Centerline Pressure Compared to Experimental Results
FIGURE 17. Centerline Pressure in Aft Inlet and Isolator Region

FIGURE 18. Adapted 3-D Mesh Surfaces (i = 30, j = 40, k = 28)
FIGURE 19. Density Contours on Bottom Wall, Centerline and Selected Surfaces

FIGURE 20. Adapted Mesh on Bottom Wall, Centerline and Selected Surfaces
FIGURE 21. Exit Flap Unstart (0.0 ms and 1.65 ms)
FIGURE 22. Exit Flap Unstart (2.47 ms and 4.26 ms)
FIGURE 23. Exit Flap Unstart (4.67 ms and 5.49 ms)
FIGURE 24. Exit Flap Inlet Unstart in Progress I (3-D view)

FIGURE 25. Exit Flap Inlet Unstart in Progress II (3-D view)
FIGURE 26. Exit Flap Inlet Unstart after Upper Cowl Spillage (3-D view)

FIGURE 27. Angle of Attack Unstart (0.137 ms)
FIGURE 28. Angle of Attack Unstart (0.275 ms and 0.686 ms)
FIGURE 29. Angle of Attack Unstart (1.37 ms and 2.06 ms)
FIGURE 30. Angle of Attack Unstart (2.75 ms and 3.43 ms)
FIGURE 31. Angle of Attack Unstart (4.12 ms)