3.0 Fundamental Equation

3.1 Nonlinear Fluid Analysis

In the absence of viscosity, heat transfer, and body forces, the Navier-Stokes equations reduce to the familiar time dependent Euler equations written in integral form for a bounded domain Ω, with boundary $\partial\Omega$, as

$$\iiint_{\Omega} \frac{\partial Q}{\partial t} dV + \oint_{\partial\Omega} F \quad N dS = 0 \tag{3.1}$$

where $Q = \{\rho \quad \rho u \quad \rho v \quad \rho w \quad \rho e_o\}^T$ is the aerodynamic state vector, and $F \quad N$ are the inviscid flux vectors normal the boundary $\partial\Omega$

$$E = F \quad N = \begin{bmatrix} \rho\Theta \\ \rho\Theta u + p\eta_x \\ \rho\Theta v + p\eta_y \\ \rho\Theta w + p\eta_z \\ (\rho e_o + p)\Theta \end{bmatrix} \tag{3.2}$$

$N = \{\eta_x \quad \eta_y \quad \eta_z\}^T$ represents the outward pointing normal to the boundary and Θ denotes the normal velocity

$$\Theta = u\eta_x + v\eta_y + w\eta_z \tag{3.3}$$

For closure, the pressure must be related to the state variables. In the current work, this is accomplished via the ideal gas law, which may be written as

$$p = (\gamma - 1)\left[\rho e_o - \rho \frac{\left(u^2 + v^2 + w^2\right)}{2}\right] \tag{3.4}$$

The above coupled set of nonlinear equations ensures the conservation of mass, momentum, and energy for an inviscid, compressible flow.

3.1.1 Finite Volume Formulation

The finite volume formulation is derived from application of the Reynolds transport equation expressed in equation 3.1. A discretization, and the subsequent solution obtained using this formulation, will ultimately satisfy the integral statement of conservation. Furthermore, the finite volume formulation is more attractive than the finite difference formulation due to its ability to handle arbitrary cell volumes. The shape of these volumes is irrelevant. The only requirement that must be adhered to is that the computational domain must be divided into a finite number of non-overlapping volumes.

An expression for the semi-discrete approximation to the governing equations may be expressed as

$$V_i \frac{\partial Q_i}{\partial t} = -R_i \tag{3.5}$$

where Q_i is the cell-averaged state variables for cell i and R_i is the residual vector containing the inviscid fluxes

$$R_i = \oint_{\partial\Omega} F \; N \, dS = \sum_{j=\kappa(i)} E_{i,j} \, A_{i,j} \tag{3.6}$$

and $A_{i,j}$ is the area of face j for cell i through which the flux passes. The number of unknowns in Eq.(3.5), i.e., the dimension of Q_i, is four times the total number of triangular cells in two-dimensions and five times the total number of tetrahedral cells in three-dimensions, respectively. Due to the random placement of cells in an unstructured

mesh, a *generalized indexing* scheme (requiring the use of a connectivity matrix which is referenced repeatedly throughout the solution process) must be used [112].

3.1.2 Upwind Discretization

Methods currently being used to construct the inviscid flux vectors, which appear in the right hand side of Eq.(3.5), are the central and upwind differencing schemes. Central difference schemes lack dissipation and are inherently unstable. Hence, to prevent oscillations near shock waves or stagnation points, artificial dissipation must be added [113]. The most popular form of this dissipation is due to Jameson et al. [114,115]. It consists of a blend of second and fourth order differences of the conserved variables. This type of dissipation, however, requires user specified second and fourth order dissipation coefficients which have been found to be case dependent.

Upwind methods overcome this deficiency by modeling the underlying physics of signal propagation as dictated by characteristic theory and, thus, are naturally dissipative. Currently there are many upwind schemes available in the literature [116-119]. A review and comparison for a number of these schemes is presented in references 120 and 121. For the computations in the present work, the flux vector splitting technique of Van Leer [117] is used exclusively.

For Van Leer's flux vector splitting scheme, the flux vectors through face *j* are split into the following contributions

$$E_j = E^+\left(Q_j^-\right) + E^-\left(Q_j^+\right) \tag{3.7}$$

where Q_j^- and Q_j^+ denote the state variables interpolated to the left and right sides of the cell interface. Determination of the appropriate fluxes are based on the Mach number normal to the cell face. This results in the possible occurrences of supersonic or subsonic flow through the face. The supersonic fluxes are evaluated as

$$E^+ = (F \quad N)^-, \quad E^- = (F \quad N)^+ = 0 \qquad M_n \geq 1 \tag{3.8a}$$

$$E^+ = (F \quad N)^- = 0, \quad E^- = (F \quad N)^+ \qquad M_n \leq -1 \tag{3.8b}$$

where $(F \quad N)^{\mp}$ are the fluxes given by Eq.(3.2) using the interpolated state variables to the left and right cell interfaces, and M_n represents the Mach number normal to the cell face and is given by

$$M_n = \frac{\Theta}{a} \tag{3.9}$$

with the local speed of sound

$$a = \left(\frac{\gamma p}{\rho}\right)^{1/2} \tag{3.10}$$

As for the subsonic occurrence $|M_n| < 1$, the split fluxes are evaluated as

$$E^{\pm} = \begin{Bmatrix} f^{\pm}_{mass} \\ f^{\pm}_{mass} f^{\pm}_{mom1} \\ f^{\pm}_{mass} f^{\pm}_{mom2} \\ f^{\pm}_{mass} f^{\pm}_{mom3} \\ f^{\pm}_{mass} f^{\pm}_{e} \end{Bmatrix} \tag{3.11}$$

with

$$f^{\pm}_{mass} = \pm \frac{\rho a}{4}(M_n \pm 1)^2 \tag{3.12}$$

$$\begin{aligned} f^{\pm}_{mom1} &= u + \eta_x(-\Theta \pm 2a)/\gamma \\ f^{\pm}_{mom2} &= v + \eta_y(-\Theta \pm 2a)/\gamma \\ f^{\pm}_{mom3} &= w + \eta_z(-\Theta \pm 2a)/\gamma \end{aligned} \tag{3.13}$$

$$f^{\pm}_{e} = \frac{(1-\gamma)\Theta^2 \pm 2(\gamma-1)\Theta a + 2a^2}{(\gamma^2 - 1)} + \frac{u^2 + v^2 + w^2}{2} \tag{3.14}$$

A point worth noting is that Van Leer's flux vector splitting is continuously differentiable, which makes it a valuable method for evaluating the inviscid fluxes for implicit time integration algorithms where the flux Jacobians are required. Moreover, it has been found in practice that steady shocks are resolved with at most two interior zones [117,120,121] with this flux splitting method.

3.1.3 Spatial Differencing

The development of a higher-order spatially accurate scheme ultimately depends on the interpolation of the state variables to the left and right of the cell interfaces. The manner in which this interpolation is accomplished depends on the grid type, and is one of the major differences between structured and unstructured grid algorithms. This is not to say that the methods used for structured grids cannot be extended to unstructured grids [122-124]; it has just been found difficult to obtain CPU efficient, accurate results. Thus, techniques which exploit the geometric properties of triangles and tetrahedra have been developed and used with success for unstructured grid algorithms [47,125-127].

A higher-order scheme is obtained by expanding the cell-centered solution to each cell face using a Taylor series expansion [125] which may be expressed as

$$Q_f^{\pm} = \left(Q + \quad Q \quad \Delta\vec{r}\right)^{\pm} \tag{3.15}$$

where the solution gradient, $\quad Q$, at the center of the cell is found using geometrically invariant features of tetrahedra. The expression for the solution gradient at the cell center may be obtained from the application of Green's theorem as

$$\left(\quad Q \quad \Delta\vec{r}\right)^{\pm} = \frac{1}{4}\begin{bmatrix}1\\3\end{bmatrix}\left(Q_{n1} + Q_{n2} + Q_{n3}\right) - Q_{n4}^{\pm}\begin{bmatrix}\\\end{bmatrix} \tag{3.16}$$

where Q_{n1}, Q_{n2}, Q_{n3} are the primitive variables at the three nodes that constitute the face through which the flux passes, and $Q_{n4}^{\pm}$ are the same variables at the appropriate fourth

node of the tetrahedra opposite to the face. In the current work, the data at the nodes are obtained from the cell center solution by using either an inverse distance [125,128] or a psuedo-Laplacian [47,127] weighting procedure. Both procedures, described in Ref. 129, attempt a multidimensional weighted averaging of the form

$$Q_n = \left(\sum_{i=1}^{nc} w_{c,i} \, Q_{c,i} \right) \Big/ \left(\sum_{i=1}^{nc} w_{c,i} \right) \tag{3.17}$$

where $w_{c,i}$ are the computed weighting factors from the desired node, n , to the surrounding nc cell centers.

The weighting factors from the inverse-distance procedure may be written as

$$w_{c,i} = \frac{1.0}{\left(\Delta x^2 + \Delta y^2 + \Delta z^2 \right)^{1/2}} \tag{3.18}$$

where $\Delta x = \left(x_{c,i} - x_n \right)$, $\Delta y = \left(y_{c,i} - y_n \right)$, and $\Delta z = \left(z_{c,i} - z_n \right)$. It should be noted that this weighting procedure has been shown to be slightly less than second-order accurate [127]. In Ref. 130, however, the data at the nodes was interpolated using the inverse-distance weighting procedure and by a linear least squares fit of the data, with no discernible differences observed between the two.

The psuedo-Laplacian weighting procedure, which has been shown to be fully second-order accurate [127], may be expressed as

$$w_{c,i} = 1 + \Delta w_{c,i} \tag{3.19a}$$

with

$$\Delta w_{c,i} = \lambda_x \, \Delta x + \lambda_y \, \Delta y + \lambda_z \, \Delta z \tag{3.19b}$$

where Δx, Δy, and Δz are given above, and λ_x, λ_y, and λ_z are obtained from the solution of an optimization problem [47,127] using the method of Lagrange multipliers, and are given by

$$\lambda_x = -\vec{\Omega}\ \left(\vec{I}^{\,y} \times \vec{I}^{\,z}\right),\ \lambda_y = \vec{\Omega}\ \left(\vec{I}^{\,x} \times \vec{I}^{\,z}\right),\ \lambda_z = -\vec{\Omega}\ \left(\vec{I}^{\,x} \times \vec{I}^{\,y}\right) \tag{3.20a}$$

where the following definitions have been used

$$\vec{\Omega} = \left\{ \Omega_x/D \quad \Omega_y/D \quad \Omega_z/D \right\},\quad \vec{I}^{\,x} = \left\{ I_{xx} \quad I_{xy} \quad I_{xz} \right\} \tag{3.20b}$$

$$\vec{I}^{\,y} = \left\{ I_{xy} \quad I_{yy} \quad I_{yz} \right\},\quad \vec{I}^{\,z} = \left\{ I_{xz} \quad I_{yz} \quad I_{zz} \right\} \tag{3.20c}$$

with

$$D = I_{xx}\left(I_{yy}I_{zz} - I_{yz}^2\right) - I_{xy}\left(I_{xy}I_{zz} - I_{xz}I_{yz}\right) + I_{xz}\left(I_{xy}I_{yz} - I_{yy}I_{xz}\right) \tag{3.20d}$$

$$\Omega_x = \sum_{i=1}^{nc}\left(x_{c,i} - x_n\right),\quad \Omega_y = \sum_{i=1}^{nc}\left(y_{c,i} - y_n\right),\quad \Omega_z = \sum_{i=1}^{nc}\left(z_{c,i} - z_n\right) \tag{3.20e}$$

$$I_{xx} = \sum_{i=1}^{nc}\left(x_{c,i} - x_n\right)^2,\quad I_{yy} = \sum_{i=1}^{nc}\left(y_{c,i} - y_n\right)^2,\quad I_{zz} = \sum_{i=1}^{nc}\left(z_{c,i} - z_n\right)^2 \tag{3.20f}$$

$$I_{xy} = \sum_{i=1}^{nc}\left(x_{c,i} - x_n\right)\left(y_{c,i} - y_n\right) \tag{3.20g}$$

$$I_{xz} = \sum_{i=1}^{nc}\left(x_{c,i} - x_n\right)\left(z_{c,i} - z_n\right) \tag{3.20h}$$

$$I_{yz} = \sum_{i=1}^{nc}\left(y_{c,i} - y_n\right)\left(z_{c,i} - z_n\right) \tag{3.20i}$$

Observe that, regardless of the method chosen to obtain the weighting factors, they are constructed solely from geometric properties of the grid. This is in direct contrast to structured grid algorithms that transform the mesh to a computational space where the upwind interpolation may proceed without the explicit need of geometric information. The mesh dependence of the interpolation in Eq.(3.15) for unstructured grids requires the computation of additional derivative terms when performing shape sensitivity analysis. These additional terms will be discussed below, and are given in the Appendices.

3.2 Linear Structural Analysis

Structural analysis primarily deals with the determination of displacement and stress distributions under given loads, temperatures, and constraints. The displacement and stress fields may be obtained by solving the basic equations of elasticity while satisfying imposed boundary conditions on forces and/or displacements. To summarize, the basic equations of elasticity for a general three-dimensional structure include 3 equations of equilibrium, 6 stress-strain equations, and 6 strain-displacement equations. Thus there are 15 equations which may be solved for the 15 unknowns, e.g., 3 displacements, 6 stresses, and 6 strains. For further information concerning the theory of elasticity, the reader is directed to the classical text of Timoshenko and Goodier [131]. In the current work, the equilibrium equations of linear static structural analysis are modeled, and only constant strain triangle (CST) membrane elements and truss members are considered. The equilibrium equation may be written in terms of the displacement field as

$$Ku - L = 0 \tag{3.21}$$

where u is the structural state vector, K is the global stiffness matrix, which is the sum of the element stiffness matrices, and L is the global load vector. A more detailed discussion of structural analysis and the finite-element method may be found in references 132-134. In the sections to follow, the element types used in the current work are presented.

3.2.1 Constant Strain Triangle Membrane Elements

In the weak form of the equilibrium equation, on which the finite-element approximation is based, the displacements are the primary variables and only first derivatives of these displacements are required for their solution. Since first derivatives of the primary variables are involved, the displacements must be approximated by the Lagrange family of interpolation polynomials [132]. The lowest order polynomial that has first derivatives in

two coordinate directions is bilinear. The simplest element that satisfies this requirement is the linear triangle, referred to as the constant strain triangle. Derivation of constant strain triangular elements may be found in any text on the finite-element method [132-134], and will therefore be omitted here. Furthermore, before transformation to a two-dimensional space, each element has 9 degrees-of-freedom, e.g., three displacement components at each node. The basic assumptions are (1) uniform thickness, (2) plane stress state, and (3) constant strain in the field.

3.2.2 Truss Members

Truss members, sometimes referred to as bar elements, by definition can only carry axial loads and deform axially. The linear, three-dimensional truss member has 6 degrees-of-freedom, e.g., three displacement components at each node. After transformation to an axial coordinate system, the element has only axial displacements at the two ends of the bar. Once again, like the constant strain triangle membrane element discussed in the previous section, truss members are well documented in the literature [132-134]. The basic assumption of this element is that the cross-sectional area of the bar remains constant along the length of the member.

3.3 Aerodynamic Sensitivity Analysis

As noted in a previous section, to determine the needed sensitivity derivatives, the sensitivity of the state vector $\partial Q/\partial \beta_k$ is required. To obtain this, the discrete residual vector (for a steady-state solution) may be recast as

$$R\left(Q\left(\beta_k\right), X\left(\beta_k\right), \beta_k\right) = 0 \tag{3.22a}$$

where the explicit and implicit dependencies of the residual on the state vector, the computational mesh, and the design variables are asserted. Similarly, the satisfaction of the boundary conditions of the problem may be expressed as

$$B\left(Q\left(\beta_k\right), X\left(\beta_k\right), \beta_k\right) = 0 \tag{3.22b}$$

At this point, one of two discrete formulations may be used to determine the sensitivity derivatives. These formulations are referred to as the direct differentiation method (Eqs.(2.6a,b)) and the adjoint variable method (Eqs.(2.7a,b)). For reasons which will be summarized below, the direct approach is used in the current work. For a more detailed discussion of both methods, and their associated boundary conditions, the reader is referred to reference 135.

3.3.1 Direct Differentiation Formulation

In this formulation Eqs.(3.21a and b) are directly differentiated with respect to the vector of design variables to produce the following linear equations

$$\frac{dR}{d\beta_k} = \frac{\partial R}{\partial Q}\frac{\partial Q}{\partial \beta_k} + \frac{\partial R}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial R}{\partial \beta_k} = 0 \tag{3.23a}$$

$$\frac{dB}{d\beta_k} = \frac{\partial B}{\partial Q}\frac{\partial Q}{\partial \beta_k} + \frac{\partial B}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial B}{\partial \beta_k} = 0 \tag{3.23b}$$

or, in terms of the state vector at interior cells, Q_o, and on the boundary, Q_b, as

$$\frac{\partial R}{\partial Q_o}\frac{\partial Q_o}{\partial \beta_k} + \frac{\partial R}{\partial Q_b}\frac{\partial Q_b}{\partial \beta_k} + \frac{\partial R}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial R}{\partial \beta_k} = 0 \tag{3.24a}$$

$$\frac{\partial B}{\partial Q_o}\frac{\partial Q_o}{\partial \beta_k} + \frac{\partial B}{\partial Q_b}\frac{\partial Q_b}{\partial \beta_k} + \frac{\partial B}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial B}{\partial \beta_k} = 0 \tag{3.24b}$$

rearranging yields the following matrix equation

$$\begin{bmatrix} \frac{\partial R}{\partial Q_o} & \frac{\partial R}{\partial Q_b} \\ \frac{\partial B}{\partial Q_o} & \frac{\partial B}{\partial Q_b} \end{bmatrix} \begin{Bmatrix} \frac{\partial Q_o}{\partial \beta_k} \\ \frac{\partial Q_b}{\partial \beta_k} \end{Bmatrix} = - \begin{Bmatrix} \frac{\partial R}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial R}{\partial \beta_k} \\ \frac{\partial B}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial B}{\partial \beta_k} \end{Bmatrix} \tag{3.24c}$$

where $\partial R/\partial Q$, $\partial R/\partial X$, $\partial B/\partial Q$, and $\partial B/\partial X$ are the Jacobian matrices evaluated at a converged flow solution, and $\partial X/\partial \beta_k$ is the grid sensitivity term. For each design variable, the number of unknowns in Eq.(3.24) is the total number of cells plus the total number of boundary faces in the mesh, e.g., *ncell+nbface*. The coefficient matrix for this equation has the dimensions of (*ncell+nbface*)^2, but is very sparse. The dimension of the above system of equations can be reduced via the implicit treatment of the boundary sensitivity, referred to as *pre-elimination* in reference 135. This is accomplished by solving Eq.(3.24b) for the boundary sensitivity and substituting it into Eq.(3.24a); grouping and rearranging terms yields

$$\frac{\partial \tilde{R}}{\partial Q}\frac{\partial Q_o}{\partial \beta_k} = -\frac{\partial \tilde{R}}{\partial \beta_k} \tag{3.25}$$

where

$$\frac{\partial \tilde{R}}{\partial Q} = \frac{\partial R}{\partial Q_o} - \frac{\partial R}{\partial Q_b}\left[\frac{\partial B}{\partial Q_b}\right]^{-1}\left[\frac{\partial B}{\partial Q_o}\right] \tag{3.26a}$$

and

$$\frac{\partial \tilde{R}}{\partial \beta_k} = \frac{\partial R}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial R}{\partial \beta_k} - \frac{\partial R}{\partial Q_b}\left[\frac{\partial B}{\partial Q_b}\right]^{-1}\left[\frac{\partial B}{\partial X}\frac{\partial X}{\partial \beta_k} + \frac{\partial B}{\partial \beta_k}\right] \tag{3.26b}$$

For each design variable the number of unknowns in Eq.(3.25) has been reduced to *ncell*, and the dimension of the coefficient matrix is now *ncell*^{2}. Since this equation is linear, the exact linearization of the residual vector and the boundary conditions must be utilized in equations 3.26a and 3.26b. When higher-order spatially accurate sensitivity analysis is desired, the number of non-zero entries in the coefficient matrix can become

prohibitively large for three-dimensional computations. The exact number of non-zero entries is mesh dependent for unstructured grids, and may vary from one mesh to the next, but is approximately (30~50)×25×*ncell*. For example, the storage requirements of Eq.(3.26a) alone, for a three-dimensional unstructured mesh containing 350,000 cells, would be approximately 260 to 437 mega-words. This memory requirement is beyond that which is currently possible on modern supercomputers.

It should be noted that the task of constructing exactly or analytically all of the required Jacobians and derivatives above by hand, and then building the software for evaluating these terms can be extremely complex. This problem is exemplified by the inclusion of even the most elementary turbulence model (for viscous flow) or use of a sophisticated grid generation package for adapting (or regenerating) the computational mesh to the latest design. A promising possible solution to this problem, however, has been found in the use of a technique known as automatic differentiation, which involves the application of a precompiler software tool called ADIFOR (Automatic DIfferentiation of FORtran, Ref. 136). This software has been utilized, with much success, to obtain complex derivatives from advanced CFD and grid generation codes for use within aerodynamic design optimization procedures [24,44,45,137-140].

In the present work, the Jacobians $\partial R/\partial Q$, $\partial R/\partial X$, $\partial B/\partial Q$, and $\partial B/\partial X$ as well as all derivatives (except for the grid sensitivity term) are constructed by hand, and are given in the Appendices. This is due to the fact that an inviscid fluid model is assumed, with the inviscid fluxes being constructed via the flux vector splitting scheme of Van Leer (a scheme which is continuously differentiable and well documented). The boundary conditions types used in the present work are inviscid surface (flow tangency) and characteristic inflow/outflow; which are both differentiable. ADIFOR, on the other hand, is used on the

unstructured grid adaptation algorithm to provide the required grid sensitivity terms. Details of this algorithm and the evaluation of grid sensitivities will be discussed in a later section.

3.3.2 Adjoint Variable Formulation

The adjoint variable formulation of the sensitivity derivatives given in Eqs.(2.7a,b) may be derived by combining Eqs.(3.23a and b) from the direct differentiation method with the sensitivity derivatives in equations 2.6a and b. From this, the adjoint vectors λ_F and λ_{C_j} may be conveniently defined such that the sensitivity of the field variables is no longer needed [135]. This is accomplished by defining the adjoint vectors as

$$\left(\frac{\partial \tilde{R}}{\partial Q}\right)^T \lambda_F = \frac{\partial \tilde{F}}{\partial Q} \tag{3.27a}$$

$$\left(\frac{\partial \tilde{R}}{\partial Q}\right)^T \lambda_{Cj} = \frac{\partial \tilde{C}_j}{\partial Q} \tag{3.27b}$$

which requires the solution of *ncon*+1 linear systems. Once again, ~ denotes that the boundary conditions have been implicitly treated or *pre-eliminated* from the equations; see Ref. 135 for further details of this procedure for the adjoint equations. Nevertheless, it can clearly be seen that the most efficient method for obtaining the sensitivity derivatives depends on the formulation of the optimization problem, i.e., the choice and number of design variables and constraints.

3.3.3 Direct vs. Adjoint Variable Method

In the current work the direct differentiation approach, as opposed to the adjoint variable approach, was used to perform the discrete sensitivity analysis. As is well known, for design problem formulations in which the number of design variables exceed the number of constraints plus one (for the objective function), the adjoint method is the preferred

approach. However, when the reciprocal is true and there are more constraints than design variables, as is typical in the case of multidisciplinary optimization, the direct approach is more attractive. For example, the aeroelastic design optimization of an aircraft wing would undoubtedly have constraints placed on the maximum allowable stress in the structural members, maximum allowable strains in the skin, and also on the dynamic pressure at which divergence occurs. To perform the discrete aeroelastic sensitivity analysis, a coupled set of adjoint equations would need to be solved for each constraint. In addition, adjoint equations would need to be defined for any aerodynamic constraints as well. Thus, it can clearly be seen that the multidisciplinary design of a configuration, which utilizes discrete sensitivity analysis, is subject to an enormous number of constraints originating from the various disciplines. Therefore since the ultimate goal of the present work is the development of a integrated multidisciplinary analysis and optimization procedure (using discrete sensitivity analysis), the direct differentiation method was adopted.