46th AIAA Aerospace Sciences Meeting and Exhibit
7 - 10 January 2008, Reno, Nevada

AIAA 2008-727

Solution of the Unsteady Discrete Adjoint for Three-Dimensional Problems on Dynamically Deforming Unstructured Meshes

Dimitri J. Mavriplis *

Department of Mechanical Engineering, University of Wyoming, Laramie WY 82071, USA

The formulation and solution of the adjoint problem for unsteady flow simulations using the Reynolds-averaged Navier-Stokes equations in the presence of dynamically deforming unstructured meshes is demonstrated. A discrete adjoint approach is used, and the full linearization is built up in a systematic and modular fashion. Discrete conservation in the analysis problem is ensured through the geometric conservation law, which is linearized consistently for the adjoint problem. An agglomeration multigrid scheme is used to solve the time-implicit problem at each time step for both the analysis problem, the adjoint problem, and the mesh and mesh adjoint problems. The methodology is demonstrated through a simple time-dependent pitching wing optimization problem.

I. Introduction

The use of the adjoint equations has become a popular approach for solving aerodynamic design optimization problems based on computational fluid dynamics.[1–5] Adjoint equations are a very powerful tool in the sense that they allow the computation of sensitivity derivatives of an objective function with respect to a set of given inputs at a cost which is essentially independent of the number of inputs.

The use of adjoint equations either in continuous or discrete form for solving steady-state aerodynamic optimization problems[1,3,5–7] is now fairly well established. However, relatively little work has been done in applying these methods for unsteady time-dependent problems.[8–10] In the context of unsteady flows, frequency domain methods which allow for reduced computational expense, particularly for problems with strong periodic behavior, have been investigated by various authors .[11,12] However, there are many problems for which time-domain approaches are preferable, such as problems with no dominant periodic behavior, and frequency domain techniques are to be viewed as complementary approaches to time domain methods.

One of the difficulties associated with time-dependent adjoint formulations is that they result in the need to integrate the adjoint equations backwards in time, from the final time value to the initial condition of the unsteady simulation, in order to compute the complete unsteady sensitivity vector. For non-linear problems, as in the case of the governing equations of fluid dynamics, this requires the complete time history of the unsteady analysis solution to be stored, for later use by the adjoint solver as it proceeds backwards in time. Another challenge lies in the formulation of the adjoint problem itself, which can become relatively complex for unsteady problems, particularly in the presence of dynamically deforming meshes, which requires the flow problem to be written in arbitrary Lagrangian Eulerian (ALE) form, and where special consideration of the Geometric Conservation Law (GCL) is required in order to ensure discrete conservation.[13–15]

In previous work, a systematic approach for formulating the discrete adjoint problem for steady-state problems on unstructured meshes has been developed and applied initially to two-dimensional[16] and subsequently to three-dimensional[6] problems. In follow-on work, this same approach was used to derive, implement and solve the unsteady adjoint problem for two-dimensional cases.[10] In the current paper, the unsteady adjoint formulation developed for two-dimensional problems is extended to three-dimensional unsteady problems involving the unsteady Reynolds-averaged Navier-Stokes equations on unstructured meshes in ALE form. This work also relies on previous developments in a unifying formulation of the discrete conservation law applicable to various time integration schemes[15] as well as previous work on efficient mesh

[*] Professor, Department of Mechanical Engineering, University of Wyoming, AIAA Associate Fellow.

deformation techniques.[17] One of the principal challenges in the three-dimensional setting is the efficient storage of the time history of the solution, as required by time-domain unsteady adjoint formulations. In this work, the time history of the unsteady analysis solution is written out to disk during the analysis run, and read back into the adjoint solver during the reverse integration procedure.

II. Analysis Problem Formulation

A. Governing Equations of the Flow Problem in ALE Form

The conservative form of the Navier-Stokes equations are used in solving the flow problem. These may be written as:

$$\frac{\partial \mathbf{w}(\mathbf{x},t)}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{w}) = 0 \tag{1}$$

Applying the divergence theorem and integrating over a moving control volume $V(t)$ yields:

$$\int_{V(t)} \frac{\partial \mathbf{w}}{\partial t} dV + \int_{dB(t)} \mathbf{F}(\mathbf{w}) \cdot \mathbf{n} dB = 0 \tag{2}$$

Using the differential identity:

$$\frac{\partial}{\partial t} \int_{V(t)} \mathbf{w} dV = \int_{V(t)} \frac{\partial \mathbf{w}}{\partial t} dV + \int_{dB(t)} (\dot{\mathbf{x}} \cdot \mathbf{n}) dB \tag{3}$$

equation (2) is rewritten as:

$$\frac{\partial}{\partial t} \int_{V(t)} \mathbf{w} dV + \int_{dB(t)} [\mathbf{F}(\mathbf{w}) - \dot{\mathbf{x}}\mathbf{w}] \cdot \mathbf{n} dB = 0 \tag{4}$$

or when considering cell-averaged values for the state $\mathbf{w}$ as:

$$\frac{\partial V \mathbf{w}}{\partial t} + \int_{dB(t)} [\mathbf{F}(\mathbf{w}) - \dot{\mathbf{x}}\mathbf{w}] \cdot \mathbf{n} dB = 0 \tag{5}$$

This is the Arbitrary-Lagrangian-Eulerian (ALE) finite-volume form of the governing equations. The equations are required in ALE form since the problem involves deforming meshes where mesh elements change in shape and size at each time-step. Here V refers to the area of the control volume, $\dot{\mathbf{x}}$ is the vector of mesh face or edge velocities, and $\mathbf{n}$ is the unit normal of the face or edge. $\mathbf{w}$ denotes the state vector of conserved variables and the flux vector $\mathbf{F}$ contains both inviscid and viscous fluxes. The single equation turbulence model of Spalart and Allmaras[18] is used to simulate turbulence effects. This equation is discretized in a similar fashion to the Navier-Stokes equations shown above, and solved in a fully coupled manner.

B. Discretization and Solution Strategies

The spatial discretization relies on a vertex-based median-dual control volume formulation which is second-order accurate in space.[6,19,20] The time derivative term in the governing equations is discretized using either the first-order accurate backward-difference formula (BDF1):

$$\frac{\partial V \mathbf{w}}{\partial t} = \frac{V^n \mathbf{w}^n - V^{n-1}\mathbf{w}^{n-1}}{\Delta t} \tag{6}$$

or a second-order accurate backwards difference formula (BDF2) requiring storage of an additional previous time level:

$$\frac{\partial V \mathbf{w}}{\partial t} = \frac{\frac{3}{2}V^n \mathbf{w}^n - 2V^{n-1}\mathbf{w}^{n-1} + \frac{1}{2}V^{n-2}\mathbf{w}^{n-2}}{\Delta t} \tag{7}$$

where the index n is used to indicate the current time-level, and $n-1$ and $n-2$ refer to previous time levels. V refers to the vector of mesh control volumes, and is indexed by the time level n, since for dynamic mesh problems these values change at each time step.

Denoting the spatially discretized terms at time level n by the operator $\mathbf{S^n}(\mathbf{w^n})$, the resulting system of non-linear equations to be solved at each time step can be denoted as (shown for the case of the BDF1 scheme for simplicity):

$$\mathbf{R}^n = \frac{V^n \mathbf{w}^n - V^{n-1}\mathbf{w}^{n-1}}{\Delta t} + \mathbf{S}^n\left(\mathbf{w}^n\right) = 0 \tag{8}$$

At each time step, an agglomeration multigrid algorithm[21] is used to converge this unsteady residual to small values. On each mesh level, a line-implicit block-tridiagonal algorithm[22] is used to further accelerate the convergence of the multigrid algorithm for highly-stretched unstructured meshes in boundary layer and wake regions.

C. Mesh Deformation Strategy

Deformation of the mesh is achieved through the linear tension spring analogy which approximates the mesh as a network of inter-connected springs. Although previous work has concentrated on the development of more robust mesh deformation techniques based on linear elasticity,[17,23,24] the spring analogy approach is used for simplicity in the current work. Three independent force balance equations are formulated for each grid point based on the displacements of neighboring grid points, using spring coefficients taken as inversely proportional to the squared length of the mesh edges. This results in a nearest neighbor stencil for the final linear system to be solved. The linear system that relates the interior vertex displacements in the mesh to known displacements on the boundaries is given as:

$$[K]\delta\mathbf{x}_{int} = \delta\mathbf{x}_{surf} \tag{9}$$

The stiffness matrix [K] is a sparse block matrix that can be treated as a constant since it is based on the initial configuration of the mesh and remains unchanged through the time-integration process. The mesh motion equations are elliptic in nature and can be solved efficiently using the same line-implicit agglomeration multigrid strategy employed for the flow equations.[6,10]

D. The Discrete Geometric Conservation Law (GCL)

The solution of the mesh motion equations (9) at each time step results in the determination of the mesh coordinates $\mathbf{x}^n$ at the new time level, based on the prescribed motion of the boundary grid points. While these values are required to generate the new mesh control volumes and other metrics at the new time level, the governing equations in ALE form also require the determination of the mesh velocities $\dot{\mathbf{x}}$, as seen from equation (5). The determination of these terms must be considered carefully in order to maintain discrete conservation in ALE form. This property is embodied in the Geometric Conservation Law (GCL), which states that a uniform flow must remain an exact solution to the discretized equations in the presence of arbitrary mesh deformation.[14,15] Substituting $\mathbf{w} = constant$ into equation (5), and noting that the integral of the flux term $\mathbf{F}(\mathbf{w})$ in this case vanishes around a closed boundary, the mathematical statement of the GCL becomes:

$$\frac{\partial V}{\partial t} = \int_{dB(t)} \dot{\mathbf{x}} \cdot \mathbf{n} dB \tag{10}$$

which provides a constraint on the method of evaluation of the grid velocity terms, which depends on the specific time discretization scheme chosen for the left hand side of this equation. In reference[15] a method for constructing the required grid velocities which discretely satisfy equation (10) is developed for BDF schemes as well as implicit Runge-Kutta time discretizations. In the case of BDF schemes, the general difference form of the BDF time discretization is first written as a sequence of incremental differences between neighboring time levels and the face-integrated grid velocities are then individually equated to the volume swept by each control volume face between neighboring time levels. The exact calculation of the volume swept by a mesh face in three-dimensions requires the use of a two point quadrature rule between each neighboring time level.[15,25] Thus, for BDF1 schemes this results in a two point time-integration rule, while for BDF2 schemes this results in a four point time-integration rule. On the other hand, for BDF1 schemes, the final functional dependence of the face-integrated grid velocities depends only on the coordinate values x^n and x^{n-1}, while for BDF2 schemes, these values depend on x^n and x^{n-1} and x^{n-1} . Note that in this formulation, it is the face-integrated grid velocities (i.e. the right-hand side of equation (10)) and not the grid velocities themselves which are computed, and used in the ALE formulation. Therefore, in the remainder of this paper, we use the notation $\dot{\mathbf{x}}$ to denote the face-integrated values in the place of the actual grid point velocities, for simplicity.

E. Functional Form of Unsteady Residual

The determination of the functional form of the unsteady residual is important for the linearization to be undertaken in the formulation of the adjoint problem discussed subsequently. By definition, for backwards difference schemes, the spatial discretization terms are to be evaluated exclusively at the most recent time level n. Noting that the volumes V are functions of the grid coordinates, we obtain for BDF2:

$$\mathbf{R}^n = \frac{3}{2\Delta t}V(x^n)\mathbf{w}^n - \frac{2}{\Delta t}V(x^{n-1})\mathbf{w}^{n-1}\frac{1}{2\Delta t}V(x^{n-2})\mathbf{w}^{n-2} + \mathbf{S}^n(\mathbf{w}^n, \mathbf{x}^n, \dot{\mathbf{x}}^n) \tag{11}$$

However, given the functional form of the face-integrated grid velocities discussed above, the spatial residual ends up with dependencies on previous time levels as well, and the entire unsteady residual functional dependence becomes:

$$\mathbf{R}^n(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{w}^{n-2}, \mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{x}^{n-2}) = \tag{12}$$

$$\frac{3}{2\Delta t}V(x^n)\mathbf{w}^n - \frac{2}{\Delta t}V(x^{n-1})\mathbf{w}^{n-1} + \frac{1}{2\Delta t}V(x^{n-2})\mathbf{w}^{n-2} + \mathbf{S}^n(\mathbf{w}^n, \mathbf{x}^n, \mathbf{x}^{n-1}, \mathbf{x}^{n-2}) \tag{13}$$

with the corresponding expression depending only on n and $n-1$ values for the BDF1 case:

$$\mathbf{R}^n(\mathbf{w}^n, \mathbf{w}^{n-1}, \mathbf{x}^n, \mathbf{x}^{n-1}) = \tag{14}$$

$$\frac{1}{\Delta t}V(x^n)\mathbf{w}^n - \frac{1}{\Delta t}V(x^{n-1})\mathbf{w}^{n-1} + \mathbf{S}^n(\mathbf{w}^n, \mathbf{x}^n, \mathbf{x}^{n-1}) \tag{15}$$

III. Sensitivity Formulation

A. General Sensitivity Formulation

Before deriving the adjoint equations for time dependent problems, we first consider the general approach to sensitivity formulation. Consider a simulation which produces various outputs $\mathbf{L}$ each of which constitutes an objective function which we wish to minimize by varying certain parameters or design variables $\mathbf{D}$ in the simulation. The entire simulation begins with the specification of the design variables $\mathbf{D}$, and may involve the evaluation of multiple functions F_i or sequential steps to finally obtain the values of the objectives $\mathbf{L}$. For steady-state problems, these various functional dependencies may include steps such as flow solution and mesh deformation, while for unsteady problems, each step may consist of the integration of current values to a new time level, either for the flow variables or the mesh motion problem, or both. Thus, the entire procedure may be written as:

$$\mathbf{L}(\mathbf{D}) = \mathbf{L}(F_{n-1}(F_{n-2}(....F_2(F_1(\mathbf{D}))....)))) \tag{16}$$

A variation in the design variables $\delta\mathbf{D}$ produces a corresponding variation the the objectives $\delta\mathbf{L}$ as:

$$\delta\mathbf{L} = \frac{d\mathbf{L}}{d\mathbf{D}}\delta\mathbf{D} \tag{17}$$

where the sensitivity derivative may be calculated as:

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial\mathbf{L}}{\partial F_{n-1}}.\frac{\partial F_{n-1}}{\partial F_{n-2}}.....\frac{\partial F_2}{\partial F_1}.\frac{\partial F_1}{\partial\mathbf{D}} \tag{18}$$

For an arbitrary number of design variables and objective functions, these sensitivity derivatives constitute a rectangular matrix which will generally be costly to evaluate. However, we consider two special cases, firstly the case where we have a single design variable and an arbitrary number of objective functions, and secondly the case where we have a single objective function and an arbitrary number of design variables. In the first case, the derivative $\frac{\partial F_1}{\partial\mathbf{D}}$ constitutes a vector which is either given, or may be assumed to be easily computable. Since only the first derivative on the right-hand side of equation (18) depends on $\mathbf{L}$, it proves economical to precompute the product of all the other derivatives as:

$$\frac{d\mathbf{L}}{d\mathbf{D}} = \frac{\partial\mathbf{L}}{\partial F_{n-1}}.\left[\frac{\partial F_{n-1}}{\partial F_{n-2}}\left[.....\left[\frac{\partial F_2}{\partial F_1}\left[\frac{\partial F_1}{\partial\mathbf{D}}\right]\right]\right]\right] \tag{19}$$

Thus, the final result in brackets is a vector which is obtained as a series of matrix-vector products. This vector may be stored and then used to compute the entire vector $\frac{d\mathbf{L}}{d\mathbf{D}}$ (one element for each individual objective function L) in a single matrix-vector multiplication. This constitutes the forward differentiation model or the tangent model.

In the case where a single objective function is specified but multiple design variables are present, the adjoint model provides the most economical approach for the calculation of $\frac{d\mathbf{L}}{d\mathbf{D}}$. For this purpose, we transpose equation (18), obtaining:

$$\frac{d\mathbf{L}}{d\mathbf{D}}^T = \frac{\partial F_1}{\partial \mathbf{D}}^T . \frac{\partial F_2}{\partial F_1}^T . \frac{\partial F_{n-1}}{\partial F_{n-2}}^T ..... \frac{\partial \mathbf{L}}{\partial F_{n-1}}^T \tag{20}$$

Noting that $\frac{\partial \mathbf{L}}{\partial F_{n-1}}^T$ is a simple vector which is either given or easily computable, and $\frac{\partial F_1}{\partial \mathbf{D}}^T$ is the only term which depends on the multitude of design variables, the corresponding strategy is to precompute the right most derivatives as:

$$\frac{d\mathbf{L}}{d\mathbf{D}}^T = \frac{\partial F_1}{\partial \mathbf{D}}^T \cdot \left[\frac{\partial F_2}{\partial F_1}^T \cdot \left[\frac{\partial F_{n-1}}{\partial F_{n-2}}^T \cdot \left[..... \left[\frac{\partial \mathbf{L}}{\partial F_{n-1}}^T\right]\right]\right]\right] \tag{21}$$

The vector of sensitivities for all design variables $\frac{d\mathbf{L}}{d\mathbf{D}}^T$ may then be obtained with a single matrix vector multiplication involving the matrix $\frac{\partial F_1}{\partial \mathbf{D}}^T$ with the precomputed vector obtained from the sequence of bracketed operations.

B. Steady-State Problems

Although the application of the above procedure to steady-state shape-optimization problems has been discussed in detail in previous work,[6] the resulting procedure is outlined in this section for completeness prior to discussing the time-dependent formulation.

In this case, the simulation objective is only a function of the steady-state flow variables and grid point coordinates, and the sensitivity vector can be written using the chain rule as:

$$\frac{dL}{dD} = \frac{\partial L}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial D} + \frac{\partial L}{\partial \mathbf{w}}\frac{\partial \mathbf{w}}{\partial D} \tag{22}$$

The last term corresponds to the flow variable sensitivities, which are determined by the constraint

$$\mathbf{R}\left(\mathbf{w}(D), \mathbf{x}(D)\right) = 0 \tag{23}$$

which merely states that the flow equation residuals must vanish at steady state. Linearization of equation (23) provides the expression for the flow sensitivities:

$$\left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]\frac{\partial \mathbf{w}}{\partial D} = -\frac{\partial \mathbf{R}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial D} \tag{24}$$

which is then substituted back into equation (22). For the adjoint problem, the resulting equation must be transposed, leading to the expression:

$$\frac{dL}{dD}^T = \frac{\partial \mathbf{x}}{\partial D}^T \left[\frac{\partial L}{\partial \mathbf{x}}^T - \frac{\partial \mathbf{R}}{\partial \mathbf{x}}^T \left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]^{-T} \frac{\partial L}{\partial \mathbf{w}}^T\right] \tag{25}$$

We next define the flow adjoint variable as

$$\mathbf{\Lambda}_w = \left[\frac{\partial \mathbf{R}}{\partial \mathbf{w}}\right]^{-T} \frac{\partial L}{\partial \mathbf{w}}^T \tag{26}$$

and note that the mesh deformation equation which relates interior grid point displacement to changes in the shape governed by the design variables D as:

$$[K]\,\delta x = \delta D \tag{27}$$

leads to the following expression for the mesh sensitivities

$$\frac{\partial \mathbf{x}}{\partial D} = [K]^{-1} \tag{28}$$

where $[K]$ represents the stiffness matrix for the mesh deformation problem. When these expressions are inserted into equation (25), the final expression becomes:

$$\frac{dL}{dD}^T = [K]^{-T} \left[\frac{\partial L}{\partial \mathbf{x}}^T - \frac{\partial \mathbf{R}}{\partial \mathbf{x}}^T \mathbf{\Lambda}_w \right] \tag{29}$$

Therefore, in order to compute the steady-state sensitivities using equation (29), a flow adjoint problem (c.f. equation (26)) must be solved, followed by the rectangular matrix-vector product defined by the last term in equation (29), and finishing with a grid deformation adjoint problem, given by the $[K]^{-T}$ operator in equation (29).

C. Application to Time-Dependent Problems

For time-dependent problems, the objective may consist of an output computed at the final time of the simulation, or may be constructed as a time-integrated quantity. We consider the formulation for a time integrated objective, since this corresponds to the most general case. Thus, our general objective can be written as:

$$L^g = \int_0^T L(t)dt \tag{30}$$

When discretized in time, the integral form becomes

$$L^g(D) = \sum_{n=0}^{n=n_f} \omega_n L^n(w^n(D), x^n(D)) \tag{31}$$

where ω_n represents the quadrature weight associated with each time step value, and L^n represents the objective evaluated at the time step n, which depends directly only on the grid coordinates and flow variables at the time level n. Using the chain rule, the sensitivity vector of this objective is given as:

$$\frac{dL^g}{dD} = \sum_{n=0}^{n=n_f} \omega_n \left[\frac{\partial L^n}{\partial \mathbf{w}^n} \frac{\partial \mathbf{w}^n}{\partial D} + \frac{\partial L^n}{\partial \mathbf{x}^n} \frac{\partial \mathbf{x}^n}{\partial D} \right] \tag{32}$$

As previously, the flow sensitivities are given by the constraint that the flow equations at each implicit time step must be satisfied. However, in the time-dependent case, the flow residuals depend on values at previous time steps in addition to values at the current time step n. Considering a BDF1 time-integration scheme for simplicity, the flow residuals at time level n can be written as:

$$\mathbf{R}^n(\mathbf{w}^n(D), \mathbf{w}^{n-1}(D), \mathbf{x}^n(D), \mathbf{x}^{n-1}(D)) = 0 \tag{33}$$

which gives the following expression for the flow sensitivities at time level n:

$$\left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^n} \right] \frac{\partial \mathbf{w}^n}{\partial D} = - \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n} \frac{\partial \mathbf{x}^n}{\partial D} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^{n-1}} \frac{\partial \mathbf{x}^{n-1}}{\partial D} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n-1}} \frac{\partial \mathbf{w}^{n-1}}{\partial D} \right] \tag{34}$$

Similarly, the mesh sensitivities at time level n can be written as a function of the mesh sensitivities at the previous time level as:

$$\frac{\partial \mathbf{x}^n}{\partial D} = [K^n]^{-1} \frac{\partial \mathbf{x}^{n-1}}{\partial D} \tag{35}$$

where $[K^n]^{-1}$ corresponds to a representation of the mesh motion in going from time level $n-1$ to time level n.

The above equations illustrate how the forward sensitivity or tangent problem reduces to a forward integration in time. Assuming the values of the flow and mesh sensitivities are known at the previous time level $n-1$, the values at the new time level n are obtained through the solution of equations (34) and (35), respectively. Substituting these values into equation (32) and proceeding to the next time step, the process is repeated until the final time level is reached, thus generating the final sensitivities for the time-integrated objective L^g.

If the process is initiated with a constant initial flow field, then the initial flow sensitivities at time level $n = 0$ vanish, whereas if the process is initiated from a steady state solution, the initial flow sensitivities correspond to the steady-state flow sensitivities which are obtained through the solution of equation (24). On the other hand, the mesh sensitivities at the initial time level must be computed through the solution of the mesh deformation equations at $n = 0$ (e.g. $[K^o]$) even though no time-dependent mesh motion is present, since even the initial mesh will vary with the values of the design variables which define the shape of the geometry. Additionally, the mesh motion between any two consecutive time steps can consist either of a mesh deformation problem, or a solid body mesh transformation, or both. In the former case, the (iterative) solution of the mesh motion equations will be required at each new time step, whereas in the latter case, the $[K^n]^{-1}$ matrix is defined explicitly by the translation/rotation matrices used to displace the mesh.

In order to formulate the time-dependent adjoint problem, the expressions given by equation (34) for the flow sensitivities are substituted into equation (32), and the entire equation is transposed. Considering, for the moment, only the contributions due to the objective value L^n at level n, we obtain:

$$\frac{\partial L^n}{\partial D} = \frac{\partial \mathbf{x}^n}{\partial D}^T \frac{\partial L^n}{\partial \mathbf{x}^n}^T - \left[\frac{\partial \mathbf{x}^n}{\partial D}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T + \frac{\partial \mathbf{x}^{n-1}}{\partial D}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^{n-1}}^T + \frac{\partial \mathbf{w}^{n-1}}{\partial D}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n-1}}^T \right] \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^n} \right]^T \frac{\partial L^n}{\partial \mathbf{w}^n}^T \tag{36}$$

If we now define a flow adjoint variable at time level n as:

$$\mathbf{\Lambda}_w^n = \left[\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^n} \right]^{-T} \frac{\partial L^n}{\partial \mathbf{w}^n}^T \tag{37}$$

equation (36) can be rearranged as:

$$\frac{\partial L^n}{\partial D} = \frac{\partial \mathbf{x}^{n-1}}{\partial D}^T [K^n]^{-T} \left[\frac{\partial L^n}{\partial \mathbf{x}^n}^T - \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T \mathbf{\Lambda}^{\mathbf{n}}_{\mathbf{w}} \right] - \frac{\partial \mathbf{x}^{n-1}}{\partial D}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^{n-1}}^T \mathbf{\Lambda}^{\mathbf{n}}_{\mathbf{w}} - \frac{\partial \mathbf{w}^{n-1}}{\partial D}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n-1}}^T \mathbf{\Lambda}^{\mathbf{n}}_{\mathbf{w}} \tag{38}$$

The first term on the right-hand side of this equation is similar in form to the steady-state sensitivities derived in equation (29). We next define a mesh adjoint variable at time level n as:

$$\mathbf{\Lambda_x}^n = [K^n]^{-T} \left[\frac{\partial L^n}{\partial \mathbf{x}^n}^T - \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T \mathbf{\Lambda}^{\mathbf{n}}_{\mathbf{w}} \right] \tag{39}$$

and note that this quantity is entirely computable at the end of the analysis run, given the solution values at time level n. When substituted into equation (38), the remaining terms depend only on values at previous time steps, thus leading to a backwards recurrence relation in time. In order to obtain expressions for the remaining terms, we substitute into this equation the constraint equation for the flow sensitivities at time level $n-1$, which is similar to equation (34), although with different time level indices, i.e.

$$\left[\frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{w}^{n-1}} \right] \frac{\partial \mathbf{w}^{n-1}}{\partial D} = - \left[\frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{x}^{n-1}} \frac{\partial \mathbf{x}^{n-1}}{\partial D} + \frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{x}^{n-2}} \frac{\partial \mathbf{x}^{n-2}}{\partial D} + \frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{w}^{n-2}} \frac{\partial \mathbf{w}^{n-2}}{\partial D} \right] \tag{40}$$

The procedure then consists of factorizing all terms multiplying $\frac{\partial \mathbf{x}^{n-1}}{\partial D}$ and premultiplying these by the mesh adjoint operator $\left[K^{n-1}\right]^T$ to obtain expressions in terms of the next preceding time level $n-2$.

However, for the full time-integrated objective given in equation (31), the contributions from the L^{n-1} sensitivities at the $n-1$ time level must also be considered, namely, by analogy with equation (36):

$$\frac{\partial L^{n-1}}{\partial D} = \frac{\partial \mathbf{x}^{n-1}}{\partial D}^T \frac{\partial L^{n-1}}{\partial \mathbf{x}^{n-1}}^T - \left[\frac{\partial \mathbf{x}^{n-1}}{\partial D}^T \frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{x}^{n-1}}^T \right] \left[\frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{w}^{n-1}} \right]^T \frac{\partial L^{n-1}}{\partial \mathbf{w}^{n-1}}^T + ... \tag{41}$$

When all these terms are substituted into equation (32) and factorized appropriately, the resulting expression can be written as:

$$\frac{dL^g}{dD} = \frac{\partial \mathbf{x}^{n-2}}{\partial D}^T \mathbf{\Lambda}_x^{n-1} + \text{...previous time step terms...} \tag{42}$$

with

$$\mathbf{\Lambda}_x^{n-1} = \left[K^{n-1}\right]^{-T} \left[\mathbf{\Lambda}_x^{n} - \frac{\partial \mathbf{R}^{n}}{\partial \mathbf{x}^{n-1}}^T \mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{n}} + \omega_{n-1} \frac{\partial L^{n-1}}{\partial \mathbf{x}^{n-1}}^T - \frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{x}^{n-1}}^T \mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{n-1}}\right] \tag{43}$$

and

$$\mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{n-1}} = \left[\frac{\partial \mathbf{R}^{n-1}}{\partial \mathbf{w}^{n-1}}\right]^{-T} \left[\omega_{n-1} \frac{\partial L^{n-1}}{\partial \mathbf{w}^{n-1}}^T - \frac{\partial \mathbf{R}^{n}}{\partial \mathbf{w}^{n-1}}^T \mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{n}}\right] \tag{44}$$

where the $\omega_n, \omega_{n-1}, ...$ quadrature weights are included in order to obtain the global time-integrated sensitivity as per equation (32). When equations (44) and (43) are solved and substituted into equation (42), an expression depending only on $n-2$ and earlier time levels is obtained, and the entire process may then be repeated to advance to the next earlier time level. Thus, the unsteady adjoint sensitivity calculation corresponds to a backwards integration in time, and requires the solution of one flow adjoint and one mesh motion adjoint problem at each time step. In this sense, the time-dependent adjoint problem at each time step is similar to a steady-state adjoint problem, although the right-hand side of the unsteady flow and mesh adjoint problems contain additional terms. For example, at a given time level $n = k$, the flow and mesh adjoint problems can be written as:

$$\mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{k}} = \left[\frac{\partial \mathbf{R}^{k}}{\partial \mathbf{w}^{k}}\right]^{-T} \left[\omega_k \frac{\partial L^{k}}{\partial \mathbf{w}^{k}}^T - \left(\frac{\partial \mathbf{R}^{k+1}}{\partial \mathbf{w}^{k}}^T \mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{k+1}}\right)\right] \tag{45}$$

and

$$\mathbf{\Lambda}_x^{k} = \left[K^{k}\right]^{-T} \left[+\omega_k \frac{\partial L^{k}}{\partial \mathbf{x}^{k}}^T - \frac{\partial \mathbf{R}^{k}}{\partial \mathbf{x}^{k}}^T \mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{k}} + \left(\mathbf{\Lambda}_x^{k+1} - \frac{\partial \mathbf{R}^{k+1}}{\partial \mathbf{x}^{k}}^T \mathbf{\Lambda}_{\mathbf{w}}^{\mathbf{k+1}}\right)\right] \tag{46}$$

respectively, where the bracketed expressions correspond to the additional terms not present in the steady flow and mesh adjoint equations, and imply a recurrence relation from later to earlier time levels. Note also that at the beginning of the adjoint time integration (i.e. at the final time step $k = n_f$), these terms vanish since no later time step values $k+1$ exist, and the adjoint problems take on a form similar to the steady-state flow and mesh adjoint problems at the final simulation time step, as given in equation (37).

D. Additional Term Formulation

As can be seen from the above analysis, the unsteady adjoint problem may be implemented as a conceptually straight-forward extension of the steady-state adjoint problem, replicating the steady adjoint flow and mesh problems as well as solution strategies at each time level, with additional source terms as outlined above. These additional source terms are constructed from linearization terms which are only present in the unsteady residual, i.e. $\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n-1}}$ and $\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^{n-1}}$. However, the other terms previously present in the steady-state formulation, i.e. $\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n}}$ and $\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^{n}}$ must also be modified to take into account the different form of the unsteady residual, as given by equation (8). In both instances, terms arising from the linearization of the residual with respect to the flow variables are relatively straight-forward. For example, the Jacobian $\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n}}$ must be augmented by the diagonal term $\frac{[I]V^n}{\Delta t}$, due to the linearization of the first term in the time discretization of equation (8). Similarly, the linearization of the unsteady residual with respect to previous time level flow variables, for a BDF1 scheme is seen to be:

$$\frac{\partial \mathbf{R}^n}{\partial \mathbf{w}^{n-1}} = [I]\,\frac{V^{n-1}}{\Delta t} \tag{47}$$

In the context of the adjoint formulation, the matrix-vector product involving this term (i.e. last term in equation (45)) simply results in a rescaling of the adjoint vector $\mathbf{\Lambda}_w^n$ by the scalar value $\frac{V^{n-1}}{\Delta t}$.

On the other hand, the modifications required for the linearization of the residual with respect to the mesh coordinates are more involved, as shown previously for two dimensions in reference.[10] As in the steady-state case,[6] the term $\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}$ must take into account the direct effect of coordinate changes on the residual, as well as changes in all mesh metrics due to changes in the mesh coordinates at the current time level. However, for the unsteady case, the face integrated grid speed terms $\dot{\mathbf{x}}$ also depend on the current grid coordinate values, and must be linearized accordingly. Additionally, the leading term of the BDF time discretization depends on the control volume values computed at the current time level $V(x^n)$. In general, $\frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}$ constitutes a sparse rectangular matrix which can have a fairly complicated structure. However, since only the product of this matrix with a field vector is required in the tangent or adjoint model, the forward matrix-vector product may be evaluated as:

$$\frac{d\mathbf{R}^n}{d\mathbf{x}^n}.\delta x^n = \frac{\partial \mathbf{R}^n}{\partial \mathbf{V}^n}.\frac{\partial \mathbf{V}^n}{\partial \mathbf{x}^n}\delta x^n + \frac{\partial \mathbf{R}^n}{\partial \dot{\mathbf{x}}^n}.\frac{\partial \dot{\mathbf{x}}^n}{\partial \mathbf{x}^n}\delta x^n + \frac{\partial \mathbf{R}^n}{\partial \mathbf{fn}^n}.\frac{\partial \mathbf{fn}^n}{\partial \mathbf{x}^n}\delta x^n + \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}\delta x^n \tag{48}$$

where δx^n represents the input vector, $\mathbf{fn}^n$ represents the vector of control volume face normals over the mesh (which are constructed using only the current time level mesh coordinates for BDF schemes), $\mathbf{V}^n$ represents the vector of mesh control volumes, and $\dot{\mathbf{x}}^n$ represents the face-integrated grid velocities, which depend both on current and previous time levels. The evaluation of this matrix-vector product is performed in a multi-step procedure, given as

$$\begin{aligned}
\delta \mathbf{V}^n &= \frac{\partial \mathbf{V}^n}{\partial \mathbf{x}^n}\delta x^n &(49)\\
\delta \dot{\mathbf{x}}^n &= \frac{\partial \dot{\mathbf{x}}^n}{\partial \mathbf{x}^n}\delta x^n &(50)\\
\delta \mathbf{fn}^n &= \frac{\partial \mathbf{fn}^n}{\partial \mathbf{x}^n}\delta x^n &(51)\\
\frac{d\mathbf{R}^n}{d\mathbf{x}^n}\delta x &= \frac{\partial \mathbf{R}^n}{\partial \mathbf{V}^n}\delta \mathbf{V}^n + \frac{\partial \mathbf{R}^n}{\partial \mathbf{fn}^n}\delta \mathbf{fn}^n + \frac{\partial \mathbf{R}^n}{\partial \dot{\mathbf{x}}^n}\delta \dot{\mathbf{x}}^{\mathbf{n}} + \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}\delta x^n &(52)
\end{aligned}$$

Note that the first step involves a matrix-vector product based on the linearization of the volume terms, while the second step involves the linearization of the the face-integrated grid velocity terms, and the third step is based on the linearization of the mesh metric routines. Finally, the last step involves only the linearization of the flow residual with respect to terms which appear directly in the residual routine (i.e. volumes, grid velocities, mesh metrics, and direct dependence on coordinates). In the case of the adjoint model, the matrix-vector product of equation (48) can be evaluated in a similar fashion. Taking the transpose of equation (48), we obtain the relation:

$$\frac{d\mathbf{R}^n}{d\mathbf{x}^n}^T\Lambda_w^n = \frac{\partial \mathbf{V}^n}{\partial \mathbf{x}^n}^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{V}^n}^T\Lambda_w^n + \frac{\partial \dot{\mathbf{x}}^n}{\partial \mathbf{x}^n}^T\frac{\partial \mathbf{R}^n}{\partial \dot{\mathbf{x}}^n}^T\Lambda_w^n + \frac{\partial \mathbf{fn}^n}{\partial \mathbf{x}^n}^T\frac{\partial \mathbf{R}^n}{\partial \mathbf{fn}^n}^T\Lambda_w^n + \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T\Lambda_w^n \tag{53}$$

which can be evaluated in a multi-step procedure as:

$$\begin{aligned}
\delta \mathbf{V}^n &= \frac{\partial \mathbf{R}^n}{\partial \mathbf{V}^n}^T\Lambda_w^n &(54)\\
\delta \dot{\mathbf{x}}^n &= \frac{\partial \mathbf{R}^n}{\partial \dot{\mathbf{x}}^n}^T\Lambda_w^n &(55)\\
\delta \mathbf{fn}^n &= \frac{\partial \mathbf{R}^n}{\partial \mathbf{fn}^n}^T\Lambda_w^n &(56)\\
\frac{d\mathbf{R}^n}{d\mathbf{x}^n}^T\Lambda_w^n &= \frac{\partial \mathbf{V}^n}{\partial \mathbf{x}^n}^T\delta \mathbf{V}^n + \frac{\partial \dot{\mathbf{x}}^n}{\partial \mathbf{x}^n}^T\delta \dot{\mathbf{x}}^n + \frac{\partial \mathbf{fn}^n}{\partial \mathbf{x}^n}^T\delta \mathbf{fn}^n + \frac{\partial \mathbf{R}^n}{\partial \mathbf{x}^n}^T\Lambda_w^n &(57)
\end{aligned}$$

As in the previous case, a (transposed) linearization of each individual routine is invoked, but in the reverse order of that used by the original discretization in the tangent model. For example, the first three steps involve the linearization of the flow residual with respect to terms appearing directly in the residual construction (i.e. volumes, grid velocities and grid metrics), while the last step incorporates the linearization of each of these individual terms with respect to the grid coordinates.

Similarly, evaluation of the matrix-vector product involving the linearization of the residual with respect to previous time level grid coordinates is performed for the forward problem as:

$$\frac{d\mathbf{R}^n}{d\mathbf{x}^{n-1}}.\delta x^{n-1} = \frac{\partial \mathbf{R}^n}{\partial \mathbf{V}^{n-1}}.\frac{\partial \mathbf{V}^{n-1}}{\partial \mathbf{x}^{n-1}}\delta x^{n-1} + \frac{\partial \mathbf{R}^n}{\partial \dot{\mathbf{x}}^n}.\frac{\partial \dot{\mathbf{x}}^n}{\partial \mathbf{x}^{n-1}}\delta x^{n-1} \tag{58}$$

and for the adjoint problem as:

$$\frac{d\mathbf{R}^n}{d\mathbf{x}^{n-1}}^T \Lambda_w^n = \frac{\partial \mathbf{V}^{n-1}}{\partial \mathbf{x}^{n-1}}^T \frac{\partial \mathbf{R}^n}{\partial \mathbf{V}^{n-1}}^T \Lambda_w^n + \frac{\partial \dot{\mathbf{x}}^n}{\partial \mathbf{x}^{n-1}}^T \frac{\partial \mathbf{R}^n}{\partial \dot{\mathbf{x}}^n}^T \Lambda_w^n \tag{59}$$

where it is seen that the only dependence on previous time levels (in this case for the BDF1 scheme) occurs through the volume term evaluated at the previous time level arising from the second contribution to the time discretization, and the face-integrated grid velocity terms present in the spatial residual.

IV. Results

A. Unsteady Test Problem

A three-dimensional pitching wing is employed as a test problem for validating and demonstrating the unsteady adjoint formulation. The geometry consists of an ONERA M6 wing which oscillates about a non-swept spanwise axis which intersects the symmetry plane at the quarter-chord of the root section. The pitching motion is sinusoidal with an amplitude of 2.5^o about a mean incidence of 5^o, and a reduced frequency of 0.1682, and the freestream Mach number is 0.3. An unstructured mesh of approximately 100,000 points containing prismatic elements in the boundary layer region and tetrahedral elements in inviscid flow regions is used to discretize the computational domain. The time-dependent mesh motion is determined by rotating the entire mesh as a solid body at each time step, in response to the prescribed wing motion. The initial condition consists of a precomputed steady-state flow-field for the wing at 5^o incidence. A total of 20 time steps are used to advance the solution for one period of the pitching motion, using the first-order accurate BDF1 time integration scheme. The unsteady Reynolds-averaged Navier-Stokes equations are solved at each time step in ALE form, using the Spalart-Allmaras turbulence model,[18] although only first-order spatial discretization is used for the convective terms in both the flow and turbulence equations. Figure 1 illustrates the ONERA M6 wing and unstructured mesh used for the calculations. In Figure 2, the time history of the computed lift and drag coefficients as a function of the pitching incidence are given.

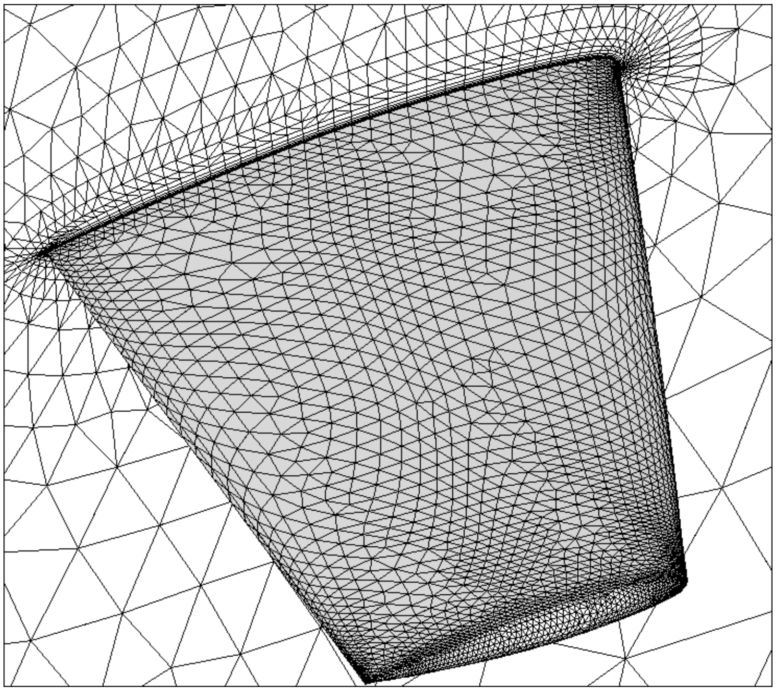


Figure 1. Illustration of mixed prismatic-tetrahedral unstructured mesh used for calculation of pitching ONERA M6 wing.

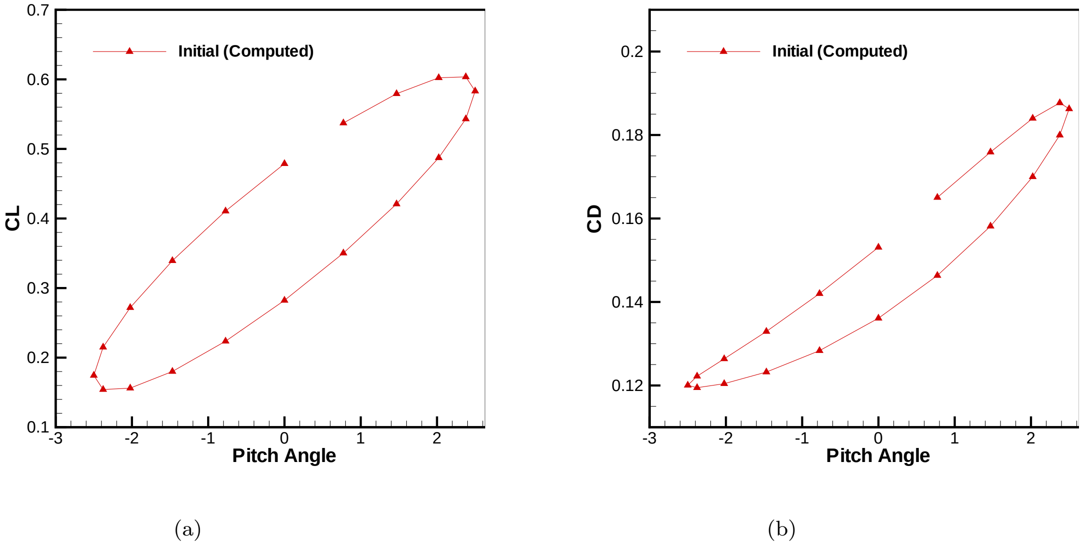


Figure 2. Computed time history of lift and drag coefficients for one cycle of pitching wing problem using 20 implicit time steps.

B. Unsteady Objective Function Formulation

An approximation to a time-integrated objective function, as defined previously by equation (31) is used in for this test case. The objective function is based on the summation of the differences between a target and a computed objective value at each time level n. Figure 3 exemplifies the formulation of this unsteady objective function. If C_L^n and C_D^n refer to the lift and drag coefficients computed on the wing at time-level n during the pitch cycle, then the local objective at time-level n is defined as:

$$L^n = (\delta C_L^n)^2 + (\delta C_D^n)^2 \tag{60}$$

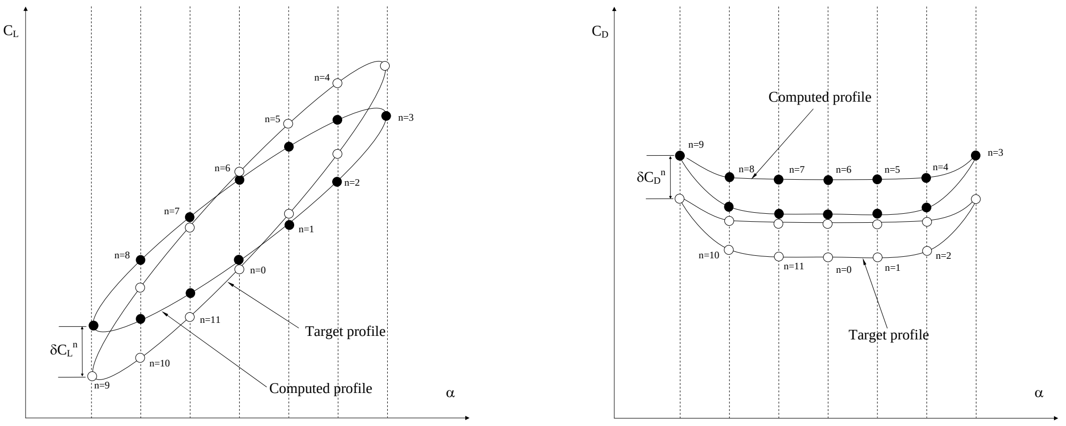
$$\delta C_L^n = (C_L^n - C_{Ltarget}^n) \tag{61}$$

$$\delta C_D^n = (C_D^n - C_{Dtarget}^n) \tag{62}$$

The global or time-integrated objective is then constructed using equal unit weights at each time step as:

$$L^g = \sum_{n=0}^{n=n_f} L^n \tag{63}$$

This construction ensures that the computed time-dependent force coefficients are driven towards the target distributions as the value of the objective function L^g is minimized.



(a) Time dependent lift profile

(b) Time dependent drag profile

Figure 3. Time dependent load profiles

C. Validation of Sensitivities

In order to validate the adjoint formulation for the calculation of sensitivities, comparisons with a finite-difference approach were performed. The finite-difference approach consists of first calculating the objective using the original geometry in analysis mode, and then repeating the calculation with a slightly perturbed configuration. The sensitivity value is then given by the difference in the objective function calculated with both configurations divided by the magnitude of the perturbation.

A finite-difference experiment was constructed by perturbing a point on the upper surface of the wing at the 10% chord and 15% span location in the normal direction, and then deforming the computational mesh in response to this perturbation via equation (9) prior to recomputing the flow solution.

Table 1 depicts the results of the validation study. The finite difference sensitivities were computed using various values of the perturbation magnitude, to investigate the effects of roundoff error, and values were obtained for steady-state lift and drag objectives, as well as unsteady time-integrated lift and drag objectives at 1 and 10 time steps. Both tangent and adjoint methods were used to calculate these sensitivities, displaying agreement to within roundoff error between these two approaches, demonstrating the duality principle, while good agreement between these values and the finite difference values is also observed.

Table 1. Validation of Adjoint and Tangent Sensitivities

	Steady State		Time = 1		Time = 10	
	C_L	C_D	C_L	C_D	C_L	C_D
FD: $\epsilon = 10^{-08}$	-0.3520915892	0.2643615382	-0.3768952150	0.2413373861	-	-
FD: $\epsilon = 10^{-09}$	-0.3520720937	0.2642965013	-0.3768726442	0.2412741950	-0.3504422307	0.2730632664
FD: $\epsilon = 10^{-10}$	-0.3521249958	0.2642880359	-0.3768896306	0.2412681165	-	-
Tangent	-0.3516050678	0.2652594347	-0.3759034567	0.2425679122	-0.3494761235	0.2746345922
Adjoint	-0.3516050677	0.2652594347	-0.3759034566	0.2425679122	-0.3494761233	0.2746345923

D. Time Dependent Optimization

The unsteady adjoint formulation is used to demonstrate a time-dependent optimization problem for the pitching ONERA M6 wing problem. The target time-dependent lift and drag profiles depicted in Figure 4 are prescribed and used to construct the objective function defined by equations (63) and (60). In order to ensure these profiles are realizable, they have been constructed by first deforming the original ONERA M6 wing surface, and computing the pitching wing problem with the deformed configuration. The optimization problem thus consists of recovering this deformed configuration, starting from the initial ONERA M6 wing geometry.

The optimization procedure follows the steepest descent approach described by Jameson.[26,27] Once the objective function sensitivities $\frac{dL}{dD}$ have been computed using the method described above, an increment in the design variables is prescribed as:

$$\delta D = -\lambda \frac{\tilde{dL}}{dD} \tag{64}$$

where λ represents a small time step, chosen small enough to ensure convergence of the optimization procedure, and $\frac{\tilde{dL}}{dD}$ represents the smoothed gradients $\frac{dL}{dD}$, obtained using an implicit smoothing technique, which is necessary to ensure smooth design shapes, as described in reference.[26] The current implementation of the steepest descent optimization procedure is not optimal, in that λ is determined empirically, but is sufficient for demonstrating the utility of the adjoint solution techniques described herein.

All computations are run on 8 processors of a Linux computer cluster. A total of 25 design cycles are used for the optimization problem. At each design cycle, an unsteady problem consisting of 20 implicit time steps is first solved, followed by the solution of the adjoint problem proceeding backwards in time. At each time level in the analysis procedure, the entire flow field and set of grid coordinates are written out to a binary file, in partitioned format, resulting in two partitions being written out to the local disk on each node of the dual core computer cluster. During the adjoint solution procedure, the partitioned solution and grid coordinate files on each node are read back in by the code at each time step, prior to the calculation of the sensitivity contributions at the given time step. Within the optimization run, the output and input of these solution files is essentially invisible to the user, since they occur at each design cycle and overwrite the previous files produced at earlier design iterations.

A total of 25 multigrid cycles were used to compute the flow and adjoint equations at each implicit time step. Figure 5(a) illustrates the convergence of the flow equations and adjoint problem at a given implicit time step, showing similar convergence rates using the line-implicit agglomeration multigrid algorithm. Due to the use of solid body mesh motion for the pitching wing, the solution of the mesh deformation problem and its adjoint are only required once for each design cycle (i.e. at the beginning of a new analysis run for the mesh deformation problem, and at the end of an adjoint run for the mesh adjoint problem). Figure 5(b) illustrates the convergence obtained for the mesh deformation and adjoint problem using the line-implicit agglomeration multigrid scheme, also showing rapid convergence for both problems.

In Figure 6 the optimization procedure is seen to result in a reduction of the time-integrated objective function by approximately three orders of magnitude over 25 design cycles. The main impediment to achieving a more rapid decrease in the objective lies in the use of a more sophisticated optimization strategy such as a line search or hessian-based approach. The current calculation required 3 hours on 8 cpus and produces a maximum file set size of 200 Mbytes, which is distributed over four computer node local disks.

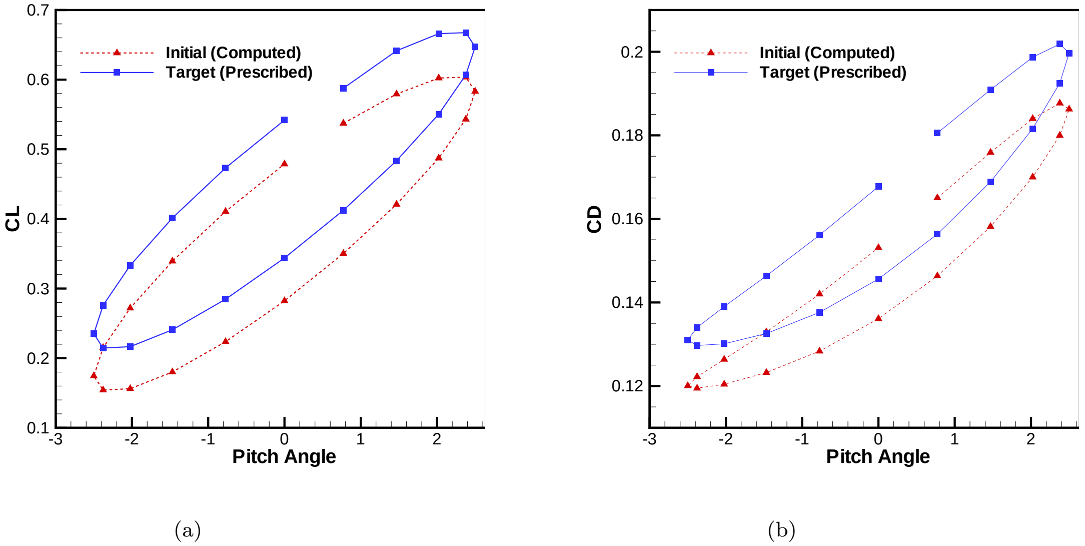


Figure 4. Target and initial computed time history of lift and drag coefficients for pitching wing optimization problem.

V. Conclusions

A time-dependent adjoint formulation for three-dimensional Reynolds-averaged Navier-Stokes flows with dynamically deforming unstructured meshes has been developed, based on previous work for three-dimensional steady-state adjoint formulations,[6] and two-dimensional unsteady formulations.[10] The current formulation has been validated by comparing adjoint-computed sensitivities with finite-difference derived values, and through the demonstration of a simple time-dependent optimization problem. Results have been confined to first-order accuracy and relatively small test cases. Work is underway to validate this approach for spatially and temporally second-order accurate simulations, and larger simulations involving several million grid points and hundreds to thousands of time steps. Concurrently, the investigation of more effective optimization strategies will be pursued in order to reduce the overall cost of time-dependent optimization problems.

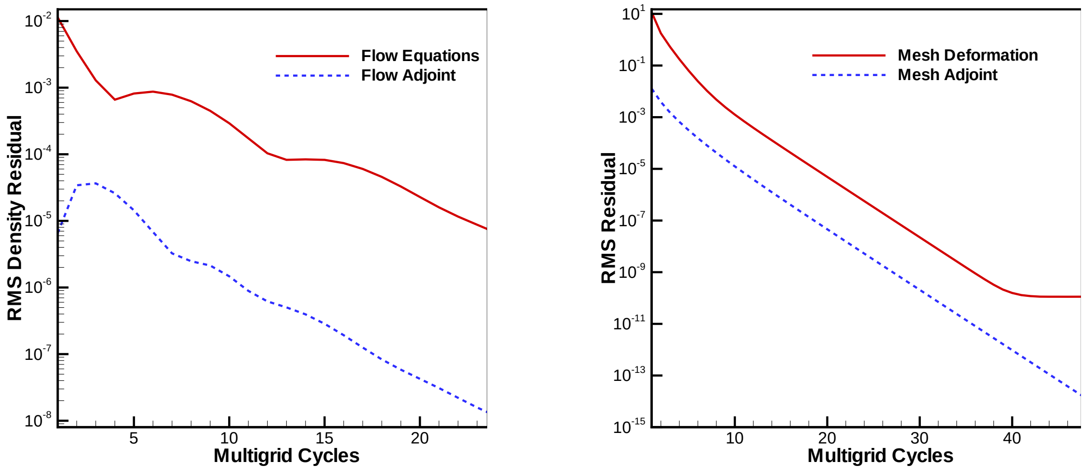
References

[1] Jameson, A., "Aerodynamic Shape Optimization using the Adjoint Method," *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics, Rhode St Genese, Belgium*, 2003.

[2] Nadarajah, S. and Jameson, A., "A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization," *Proceedings of the 38th Aerospace Sciences Meeting and Exhibit, Reno NV*, 2000, AIAA Paper 2000–0667.

[3] Nielsen, E. and Anderson, W., "Recent Improvements in Aerodynamic Optimization of Unstructured Meshes," *AIAA Journal*, Vol. 40-6, June 2002, pp. 1155–1163.

[4] Giles, M., Duta, M., and Muller, J., "Adjoint Code Developments Using Exact Discrete Approach," 2001, AIAA Paper 2001–2596.



(a)

(b)

Figure 5. (a) Multigrid convergence of time-implicit problem for flow equations and flow adjoint. (b) Multigrid convergence for mesh deformation and adjoint problem performed once per design cycle.

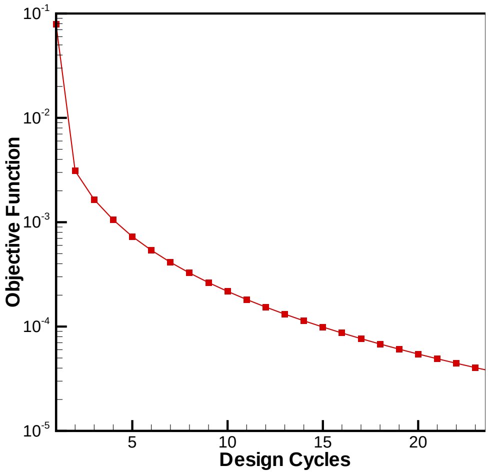
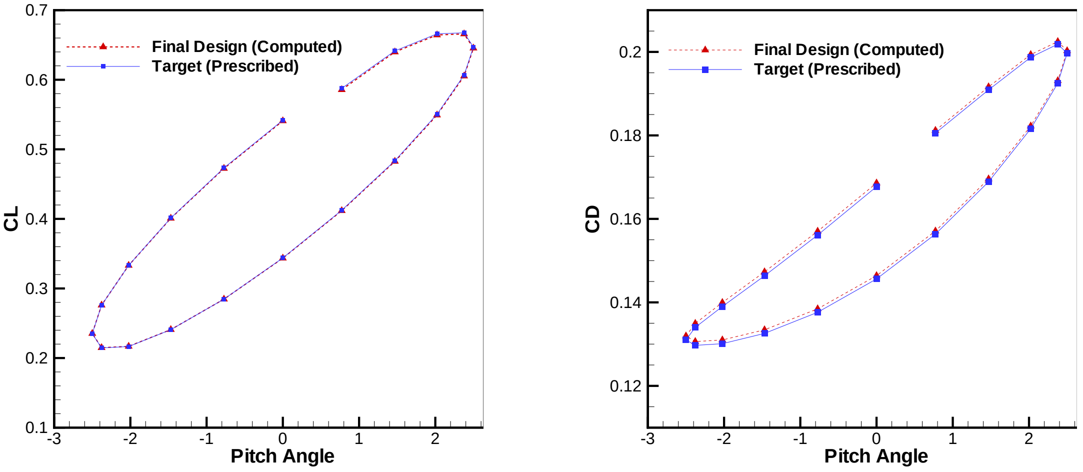


Figure 6. Convergence of Objective function for pitching wing optimization problem.



(a)

(b)

Figure 7. Optimized lift and drag profiles for pitching wing problem compared with target profiles.

[5] Elliot, J. and Peraire, J., "Practical three-dimensional aerodynamic design by optimization," *AIAA Journal*, Vol. 35-9, 1997, pp. 1479–1485.

[6] Mavriplis, D. J., "A Discrete Adjoint-Based Approach for Optimization Problems on Three Dimensional Unstructured Meshes," *AIAA Journal*, Vol. 45, No. 4, April 2007, pp. 741–750.

[7] Soto, R. and Yang, C., "An Adjoint-Based Design Methodology for CFD Optimization Problems," 2003, AIAA Paper 2003–0299.

[8] Ghayour, K. and Baysal, O., "Limit-Cycle Shape Optimization using Time-Dependent Transonic Equation," *Proceedings of the 14th Computational Fluid Dynamics Conference, Norfolk VA*, 1999, AIAA Paper 99–3375.

[9] Nadarajah, S. and Jameson, A., "Optimal Control of Unsteady Flows using A Time Accurate Method," *Proceedings of the 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference, Atlanta GA*, 2002, AIAA Paper 2002–5436.

[10] Mani, K. and Mavriplis, D. J., "An Unsteady Discrete Adjoint Formulation for Two-Dimensional Problems with Deforming Meshes," AIAA Paper 2007-0060.

[11] Nadarajah, S., McMullen, M., and Jameson, A., "Non-Linear Frequency Domain Based Optimum Shape Design for Unsteady Three-Dimensional Flow," *Proceedings of the 44th Aerospace Sciences Meeting and Exhibit, Reno NV*, 2006, AIAA Paper 2006–387.

[12] Thomas, J. P., Hall, K. C., and Dowell, E. H., "Discrete Adjoint Approach for Modeling Unsteady Aerodynamic Design Sensitivities," *AIAA Journal*, Vol. 43, No. 9, 2005, pp. 1931–1936.

[13] Thomas, P. D. and Lombard, C. K., "Geometric conservation law and its applications to flow computations on moving grids," *AIAA J.*, Vol. 17, 1979, pp. 1030–1037.

[14] Guillard, H. and Farhat, C., "On the significance of the geometric conservation law for flow computations on moving meshes," *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, 2000, pp. 1467–1482.

[15] Mavriplis, D. J. and Yang, Z., "Construction of the Discrete Geometric Conservation Law for High-Order Time Accurate Simulations on Dynamic Meshes," *Journal of Computational Physics*, Vol. 213, No. 1, April 2006, pp. 557–573.

[16] Mavriplis, D. J., "Multigrid Solution of the Discrete Adjoint for Optimization Problems on Unstructured Grids," *AIAA Journal*, Vol. 44, No. 1, Jan. 2006, pp. 42–50.

[17] Yang, Z. and Mavriplis, D. J., "A Mesh Deformation Strategy Optimized by the Adjoint Method on Unstructured Meshes," AIAA Paper 2007-0557.

[18] Spalart, P. R. and Allmaras, S. R., "A One-equation Turbulence Model for Aerodynamic Flows," *La Recherche Aérospatiale*, Vol. 1, 1994, pp. 5–21.

[19] Mavriplis, D. J. and Venkatakrishnan, V., "A Unified Multigrid Solver for the Navier-Stokes Equations on Mixed Element Meshes," *International Journal for Computational Fluid Dynamics*, Vol. 8, 1997, pp. 247–263.

[20] Mavriplis, D. J. and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *AIAA Journal of Aircraft*, Vol. 36, No. 6, Dec. 1999, pp. 987–998.

[21] Mavriplis, D., "Multigrid Techniques for Unstructured Meshes," *Notes prepared for 26th Computational Fluid Dynamics Lecture Series Program of the von Karman Institute of Fluid Dynamics, Rhode St Genese, Belgium*, 1995.

[22] Mavriplis, D. J., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *Journal of Computational Physics*, Vol. 145, No. 1, Sept. 1998, pp. 141–165.

[23] Baker, T. J., "Mesh Movement and Metamorphosis," *Engineering with Computers*, Vol. 18, No. 3, 2002, pp. 188–198.

[24] Nielsen, E. J. and Anderson, W. K., "Recent Improvements in Aerodynamic Optimization on Unstructured Meshes," *AIAA Journal*, Vol. 40, No. 6, June 2002, pp. 1155–1163.

[25] Geuzaine, P., Grandmont, C., and Farhat, C., "Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations," *Journal of Computational Physics*, Vol. 191, 2003, pp. 206–227.

[26] Jameson, A., Alonso, J. J., Reuther, J. J., Martinelli, L., and Vassberg, J. C., "Aerodynamic shape optimization techniques based on control theory," AIAA Paper 98-2538.

[27] Jameson, A., "Aerodynamic Shape Optimization using the Adjoint Method," VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics, Rhode St Genese, Belgium.