APPENDIX A

DESIGN SPACE DERIVATIVES VIA DISCRETE SENSITIVITY ANALYSIS

APPENDIX A

DESIGN SPACE DERIVATIVES VIA DISCRETE SENSITIVITY ANALYSIS

A.1. Introduction

The derivatives of an objective function $F(\vec{\beta})$ with respect to the design variables $\vec{\beta}$, often called design space derivatives, can be calculated efficiently and accurately via discrete sensitivity analysis. In discrete sensitivity analysis, the sensitivity equations are derived by differentiating the discretized system of governing partial differential equations, which involves the Jacobian matrix. For implicit codes, an approximation to the Jacobian matrix already exists, as well as the needed matrix equation solution subroutines. For explicit codes, the complex Taylor's series expansion (CTSE) method can be used to generate the Jacobian matrix from the discretized system of equations, without the need to hand-differentiate each term. CTSE can also be used to generate the numerically exact Jacobian for implicit codes. This method is discussed in Appendix D. In this appendix, the equations for the adjoint formulation and the direct formulation of discrete sensitivity analysis are developed in detail. For more detail about the derivations of discrete sensitivity analysis, see Korivi, etal[**42**] and Hou, etal[**40**]. For details about the flow prediction algorithm, see Eleshaky and Baysal [**35**].

A.2. Common Derivations

In design optimization, the objective function to be optimized is some measure of the current design's effectiveness. This objective function can be a function of the steady-state flow variables $\vec{Q}$, the grid discretization $\vec{\chi}$ and the design variables $\vec{\beta}$, or $F(\vec{Q}(\vec{\beta}), \vec{\chi}(\vec{\beta}), \vec{\beta})$. Typically, for high-fidelity simulation, a system of partial differential equations must be solve to evaluate the objective function, which can be quite computationally expensive. As a result, the evaluation of F should be avoided, whenever possible. The design space derivative, which is the derivative of F with respect to design variable $\vec{\beta}_k$, is

$$\frac{dF}{d\beta_k} = \frac{\partial F}{\partial Q}\frac{\partial Q}{\partial \beta_k} + \frac{\partial F}{\partial \chi}\frac{\partial \chi}{\partial \beta_k} + \frac{\partial F}{\partial \beta_k} \tag{A.2.1}$$

where $\frac{dF}{d\beta_k}$ represents the total variation of F with respect to the design variable β_k. Since F is an analytic function of $\vec{Q}$, $\vec{\chi}$ and $\vec{\beta}$, the vectors $\frac{\partial F}{\partial Q}$, $\frac{\partial F}{\partial \chi}$ and $\frac{\partial F}{\partial \beta_k}$ can be determined exactly. The vector $\frac{\partial \chi}{\partial \beta_k}$ can be estimated efficiently by differencing the grids generated for $\beta_k + \Delta\beta_k$ and for $\beta_k - \Delta\beta_k$, or

$$\frac{\partial \chi}{\partial \beta_k} \approx \frac{\vec{\chi}(\vec{\beta} + \Delta\beta_k) - \vec{\chi}(\vec{\beta} - \Delta\beta_k)}{2\Delta\beta_k} \tag{A.2.2}$$

where $\vec{\beta} + \Delta\beta_k = (\beta_1, \cdots, \beta_{k-1}, \beta_k + \Delta\beta_k, \beta_{k+1}, \cdots, \beta_{NDV})$ and NDV is the number of design variables. This vector can also be efficiently calculated via automatic differentiation or hand-differentiation. The vector $\frac{\partial Q}{\partial \beta_k}$ can not be approximated via finite differences without an additional steady-state simulation.

The discretized system of governing partial differential equations can be written as $W(Q(\vec{\beta}), \chi(\vec{\beta}), \vec{\beta})$, where Q is determined such that $W(Q) = 0$. Thus, the total derivative of W with respect to the design variable β_k is

$$\frac{dW}{d\beta_k} = \left[\frac{\partial W}{\partial Q}\right]\frac{\partial Q}{\partial \beta_k} + \left[\frac{\partial W}{\partial \chi}\right]\frac{\partial \chi}{\partial \beta_k} + \frac{\partial W}{\partial \beta_k} = 0 \tag{A.2.3}$$

Each term in this equation can be determined efficiently without the need for another steady-state simulation, except for the vector $\frac{\partial Q}{\partial \beta_k}$. But, since this vector is equal to the null vector, it can be manipulated so as to determine the design space derivatives without additional steady-state simulations. Rewriting,

$$\left[\frac{\partial W}{\partial Q}\right]\frac{\partial Q}{\partial \beta_k} = -\left(\left[\frac{\partial W}{\partial \chi}\right]\frac{\partial \chi}{\partial \beta_k} + \frac{\partial W}{\partial \beta_k}\right) = -\left.\frac{dW}{d\beta_k}\right|_{\text{Q fixed}} \tag{A.2.4}$$

The vector $\left.\frac{dW}{d\beta_k}\right|_{\text{Q fixed}}$ can be approximated by finite differences in the same way that $\frac{\partial \chi}{\partial \beta_k}$ is approximated.

At this point, discrete sensitivity analysis branches into two solution techniques, which are the adjoint variable formulation and the quasi-analytic (or direct) formulation. The adjoint variable formulation is usually applied when the number of design variables is greater than the number of objective functions; whereas the quasi-analytic approach is used when the number of objective functions is larger than the number of design variables. In the current research, there is only one objective function, but since it is a nonlinear least-squares function, each residual function can be considered an objective function. From this viewpoint, the quasi-analytic approach can be quite useful.

A.3. Adjoint Variable Formulation

In the adjoint variable formulation, equation (A.2.3) is multiplied by an adjoint vector λ and added to equation (A.2.1) to get

$$\frac{dF}{d\beta_k} = \left(\frac{\partial F}{\partial Q} + \lambda^T\left[\frac{\partial W}{\partial Q}\right]\right)\frac{\partial Q}{\partial \beta_k} + \frac{\partial F}{\partial \chi}\frac{\partial \chi}{\partial \beta_k} + \frac{\partial F}{\partial \beta_k} + \lambda^T\left.\frac{dW}{d\beta_k}\right|_{\text{Q fixed}} \tag{A.3.1}$$

If $\frac{\partial F}{\partial Q} + \lambda^T \left[\frac{\partial W}{\partial Q}\right] = 0$, then there is no need to calculate $\frac{\partial Q}{\partial \beta_k}$. Rearranging, the adjoint equation is

$$\left[\frac{\partial W}{\partial Q}\right]^T \lambda = -\frac{\partial F}{\partial Q} \tag{A.3.2}$$

Since our flow solver already solves $\left[\frac{\partial W}{\partial Q}\right] \Delta q = -W$, taking the transpose of $\left[\frac{\partial W}{\partial Q}\right]$ and using $-\frac{\partial F}{\partial Q}$ as the right hand side, the vector λ is easily obtained. It is important to note that the adjoint equation is independent of β_k, so λ is the same for each design variable. As a result, this equation must only be solved once for each objective function, regardless of the number of design variables. Once λ is known, the design space derivative is

$$\frac{dF}{d\beta_k} = \frac{\partial F}{\partial \chi}\frac{\partial \chi}{\partial \beta_k} + \frac{\partial F}{\partial \beta_k} + \lambda^T \frac{dW}{d\beta_k}\bigg|_{\text{Q fixed}} \tag{A.3.3}$$

A.4. Direct Formulation

The direct formulation (also called the quasi-analytic formulation) to discrete sensitivity analysis uses the flow solver to solve directly for $\frac{\partial Q}{\partial \beta_k}$ in the equation

$$\left[\frac{\partial W}{\partial Q}\right] \frac{\partial Q}{\partial \beta_k} = -\frac{\partial W}{\partial \beta_k}\bigg|_{\text{Q fixed}} \tag{A.4.1}$$

The vector $\frac{\partial Q}{\partial \beta_k}$ is then used to obtain the design space derivative via equation (A.2.1). Unfortunately, equation (A.4.1) must be solved once for each design variable. But this equation is not dependent on the objective function, so if there are many objective functions, this technique is often used.

A.5. Flow Prediction

An added benefit of discrete sensitivity analysis is that the equations involved in estimating the derivatives can also be used to predict the flow variables for the next set of design variables. Once the design space gradient $\nabla_{\vec{\beta}} F$ is approximated, and the change in design variables $\Delta\vec{\beta}$ is determined via an optimization algorithm, the design variables for the next iteration are $\vec{\beta^{n+1}} = \vec{\beta^n} + \Delta\vec{\beta^n}$. By taking a Taylor's series expansion of the flow variables at $\vec{\beta^{n+1}}$, one obtains

$$\begin{aligned} Q(\vec{\beta}^{n+1}) &= Q(\vec{\beta^n} + \Delta\vec{\beta^n}) \\ &= Q(\vec{\beta^n}) + \left[\frac{\partial Q}{\partial \beta}\right]^n \Delta\vec{\beta^n} + \text{higher order terms} \end{aligned} \tag{A.5.1}$$

or

$$\Delta Q^n \approx \left[\frac{\partial Q}{\partial \beta}\right]^n \Delta\vec{\beta^n} \tag{A.5.2}$$

Multiplying both sides by $\left[\frac{\partial W}{\partial Q}\right]$,

$$\begin{aligned} \left[\frac{\partial W}{\partial Q}\right] \Delta Q^n &\approx \left[\frac{\partial W}{\partial Q}\right] \left[\frac{\partial Q}{\partial \beta}\right]^n \Delta\vec{\beta^n} \\ &= -\left[\frac{dW}{d\beta}\right]\Bigg|_{\text{Q fixed}} \Delta\vec{\beta^n} \end{aligned} \tag{A.5.3}$$

The columns of the matrix $\left[\frac{dW}{d\beta}\right]\Big|_{\text{Q fixed}}$ consist of the vectors $\frac{dW}{d\beta_k}\Big|_{\text{Q fixed}}$, which were used in both the adjoint variable and the quasi-analytic formulation. After multiplying the matrix $\frac{dW}{d\beta}\Big|_{\text{Q fixed}}$ by the vector $\Delta\vec{\beta^n}$, the flow solver is able to solve for ΔQ^n, and the flow variables at the next design iteration are predicted. Equation (A.5.2) can be used to obtain ΔQ^n directly when the direct formulation is used, because each column in the matrix $\frac{\partial Q}{\partial \beta}$ is calculated via the direct formulation.

A.6. Variations - Using First-Order Jacobians

Often, implicit solvers use the Jacobian matrix associated with the first order discretization of the governing equations, even though the actual discretized system is of a higher order. The primary reason for this disparity is that the first order Jacobian typically has a smaller bandwidth, and thus the resulting system of equations is easier to solve computationally. By using sub-iterations, the residual vector containing the higher order terms is driven to zero at time level n via

$$\frac{\partial W_L}{\partial Q}(Q^{n,m})\Delta Q^m = -W_H(Q^{n,m}) \tag{A.6.1}$$

with $Q^{n,m+1} = Q^{n,m} + \Delta Q^m$, where W_L is the system of equations resulting from the first order discretization and W_H results from the higher order discretization. Once $\|W_H(Q^{n,m})\| < tolerance$, the sub-iterations terminate and $Q^{n+1} = Q^{n,m}$.

Since the flow solver is not equipped to handle $\frac{\partial W_H}{\partial Q}$, the matrix solution subroutine from the solver can not be used directly to solve equations (A.3.2) or (A.4.1). Following the method of Korivi, etal **[71]** to overcome this dilemma, these equations are restated as

$$\mathcal{W}^1(\lambda) = \left[\frac{\partial W_H}{\partial Q}\right]^T \lambda + \frac{\partial F}{\partial Q} = 0 \tag{A.6.2}$$

and

$$\mathcal{W}^2\left(\frac{\partial Q}{\partial \beta_k}\right) = \left[\frac{\partial W_H}{\partial Q}\right]\frac{\partial Q}{\partial \beta_k} + \left.\frac{dW}{d\beta_k}\right|_{\text{Q fixed}} = 0 \tag{A.6.3}$$

These equations are solved for λ and for $\frac{\partial Q}{\partial \beta_k}$, respectively.

Using the same sub-iteration method to solve $\mathcal{W}^1(\lambda) = 0$, one obtains

$$\frac{\mathcal{W}^1}{\partial\lambda}(\lambda^m)\Delta\lambda^m = -\mathcal{W}^1(\lambda^m) \tag{A.6.4}$$

with $\lambda^{m+1} = \lambda^m + \Delta\lambda^m$. But

$$\frac{\mathcal{W}^1}{\partial\lambda}(\lambda^m) = \left[\frac{\partial W_H}{\partial Q}\right] \tag{A.6.5}$$

which can be approximated by the first order Jacobian, so

$$\frac{\partial W_L}{\partial Q}\Delta\lambda^m = -\mathcal{W}^1(\lambda^m) \tag{A.6.6}$$

can be repeatedly solved using the same subroutines as in the flow solver to drive $\mathcal{W}^1(\lambda)$ to zero and hence solve equation (A.3.2). Equation (A.4.1) can be solved iteratively in the same fashion. This method does not remove the need for the exact Jacobian matrix for the higher-order scheme, but it does allow the more complicated problem to be solved using the lower-order Jacobian matrix.

A.7. Variations - Time-Dependent Discrete Sensitivity Analysis

When the objective function uses the flow variables from various time levels, the objective function can be expressed as

$$F(\vec{\beta}) = \sum_{n=1}^{N} f_n(Q^n, \chi, \vec{\beta}) = \sum_{n=1}^{N} f_n(Q^n(\vec{\beta}), \chi(\vec{\beta}), \vec{\beta}) \tag{A.7.1}$$

For steady-state problems, the residual vector W is only a function of the steady-state flow variables, but for time-dependent problems, the residual vector at time level n is a function of the flow variables at n and at $n-1$, or

$$W^n(Q^n, Q^{n-1}, \chi, \vec{\beta}) = 0 \tag{A.7.2}$$

If the grid χ changes in time, then additional dependencies are needed. Also, if second, or higher, order temporal accuracy is used, then the residual vector will depend on flow variables at time levels prior to $n-1$.

The design space derivative of $F(\vec{\beta})$ with respect to β_k can be written as

$$\frac{dF}{d\beta_k} = \sum_{n=1}^{N} \left(\frac{\partial f_n}{\partial Q^n}^T \frac{\partial Q^n}{\partial \beta_k} + \frac{\partial f_n}{\partial \chi}^T \frac{\partial \chi}{\partial \beta_k} + \frac{\partial f_n}{\partial \beta_k} \right) \tag{A.7.3}$$

At time level n, the derivative of the residual vector with respect to the design variables can be written as

$$0 = \frac{dW^n}{d\beta_k} = \frac{\partial W^n}{\partial Q^n}\frac{\partial Q^n}{\partial \beta_k} + \frac{\partial W^n}{\partial Q^{n-1}}\frac{\partial Q^{n-1}}{\partial \beta_k} + \frac{\partial W^n}{\partial \chi}\frac{\partial \chi}{\partial \beta_k} + \frac{\partial W^n}{\partial \beta_k} \tag{A.7.4}$$

The direct formulation solves at each time level

$$\frac{\partial W^n}{\partial Q^n}\frac{\partial Q^n}{\partial \beta_k} = -\left(\frac{\partial W^n}{\partial Q^{n-1}}\frac{\partial Q^{n-1}}{\partial \beta_k} + \left.\frac{dW^n}{d\beta_k}\right|_{Q^n,\ Q^{n-1}\ \text{fixed}} \right) \tag{A.7.5}$$

which includes the vector $\frac{\partial Q^{n-1}}{\partial \beta_k}$. Thus, the vector $\frac{\partial Q^n}{\partial \beta_k}$ for the current time level must be stored for use in the next time level. If the residual vector W is dependent on the flow variables at more time levels, then more of these vectors must be stored. For problems where the initial conditions Q^o are specified, the vector $\frac{\partial Q^o}{\partial \beta_k} = 0$, so this term does not cause any difficulties. However, if Q^o depends on the design variables $\vec{\beta}$, then this vector must be determined in another fashion; it is probably available analytically. As with steady-state discrete sensitivity analysis, the matrix equation (A.7.5) does not depend on the objective function and at each time step must be solved once for each design variable.

The derivation of the equations for the time-dependent adjoint variable equations is similar to the derivation presented by Townley and Wilson [**58**]. In the adjoint variable formulation, the derivative of the residual vector at each time level is multiplied by an adjoint vector λ^n and added to the equation (A.7.3) which becomes

$$\begin{aligned}
\frac{dF}{d\beta_k} &= \sum_{n=1}^{N}\left(\frac{\partial f_n}{\partial Q^n}^T\frac{\partial Q^n}{\partial \beta_k} + \frac{\partial f_n}{\partial \chi}^T\frac{\partial \chi}{\partial \beta_k} + \frac{\partial f_n}{\partial \beta_k}\right) \\
&+ \sum_{n=1}^{N}\left(\lambda^n\frac{\partial W^n}{\partial Q^n}\frac{\partial Q^n}{\partial \beta_k} + \lambda^n\frac{\partial W^n}{\partial Q^{n-1}}\frac{\partial Q^{n-1}}{\partial \beta_k} + \lambda^n\frac{\partial W}{\partial \chi}\frac{\partial \chi}{\partial \beta_k} + \lambda^n\frac{\partial W}{\partial \beta_k}\right) \\
&= \lambda^1\frac{\partial W^1}{\partial Q^o}\frac{\partial Q^o}{\partial \beta_k} + \sum_{n=1}^{N-1}\left(\frac{\partial f_n}{\partial Q^n}^T + \lambda^n\frac{\partial W^n}{\partial Q^n} + \lambda^{n+1}\frac{\partial W^{n+1}}{\partial Q^n}\right)\frac{\partial Q^n}{\partial \beta_k} \qquad \text{(A.7.6)} \\
&+ \left(\frac{\partial f_N}{\partial Q^N}^T + \lambda^N\frac{\partial W^N}{\partial Q^N}\right)\frac{\partial Q^N}{\partial \beta_k} + \sum_{n=1}^{N}\left(\frac{\partial f_n}{\partial \chi}^T\frac{\partial \chi}{\partial \beta_k} + \frac{\partial f_n}{\partial \beta_k} + \lambda^n\frac{\partial W}{\partial \beta_k}\right)
\end{aligned}$$

To solve for the adjoint vector λ^n's, the equation

$$\frac{\partial f_N}{\partial Q^N}^T + \lambda^N\frac{\partial W^N}{\partial Q^N} = 0 \qquad \text{(A.7.7)}$$

must be solved first. Then, the equations

$$\frac{\partial f_n}{\partial Q^n}^T + \lambda^n\frac{\partial W^n}{\partial Q^n} + \lambda^{n+1}\frac{\partial W^{n+1}}{\partial Q^n} = 0 \qquad \text{(A.7.8)}$$

are solved for the adjoint vectors, by going backwards in time. Thus, all the time dependent flow variables must be obtained before the adjoint variables can be obtained.

Once the adjoint vectors λ^n's are obtained, the design space derivative is

$$\frac{dF}{d\beta_k} = \lambda^1\frac{\partial W^1}{\partial Q^o}\frac{\partial Q^o}{\partial \beta_k} + \sum_{n=1}^{N}\left(\frac{\partial f_n}{\partial \chi}^T\frac{\partial \chi}{\partial \beta_k} + \frac{\partial f_n}{\partial \beta_k} + \lambda^n\frac{\partial W}{\partial \beta_k}\right) \qquad \text{(A.7.9)}$$

As with the steady-state adjoint variable formulation, the adjoint equations do not depend on the design variables and must be solved for each objective function F.

Finally, in regards to the choice of optimization algorithm, the Gauss-Newton algorithm works well in conjunction with the direct formulation for steady-state problems, but for time-dependent problems, it is not an efficient choice. For illustrative purposes, assume that the time-dependent objective function is of the form

$$F(\vec{\beta}) = \sum_{k=1}^{N_T} \sum_{j=1}^{N_R} f_{j,k}^2(\vec{\beta}) \tag{A.7.10}$$

where N_T is the number of time levels, N_R is the number of residual functions at each time level and N_{DV} is the number of design variables. The derivative of $F(\vec{\beta})$ can be expressed as

$$\frac{dF}{d\beta_k} = \sum_{k=1}^{N_T} \sum_{j=1}^{N_R} 2 \frac{\partial f_{j,n}}{\partial \beta_k} f_{j,k} \tag{A.7.11}$$

Using the direct formulation to calculate the derivative terms $\frac{\partial f_{j,k}}{\partial \beta_k}$ requires N_{DV} solutions of equation (A.7.5) at each time level, or $N_D V * N_T$ matrix equation solutions. Using the adjoint variable formulation to calculate these terms requires N_R solutions of equation (A.7.8) at each time level, or $N_R * N_T$ matrix equation solutions. Since one matrix equation solution is required at each time level by the flow solver, a total of N_T matrix equation solutions is required to calculate the entire set of flow variables and hence to evaluate the objective function. (The equation solved by the flow solver may be nonlinear, as with HIVEL2D, so multiple matrix equation solutions may be required at each time level.) Thus, using the direct formulation will require N_{DV} times the cost of evaluating the objective function, which is equivalent to the cost of using finite differences to evaluate the objective function. If N_R is on the same order as N_{DV}, the adjoint variable formulation fares no better computationally. As a result, the Gauss-Newton algorithm should probably not be used as the optimization algorithm for time-dependent problems. Rather, since the quasi-Newton methods also offer super-linear convergence and only require the derivative $\frac{\partial F}{\partial \beta_k}$, these methods may be a better choice for the optimization algorithm. Furthermore, the adjoint variable formulation should probably be used to estimate the design space gradient, $\nabla_{\vec{\beta}} F$.