APPENDIX B

IMPLEMENTATION AND COMPUTATIONAL COST OF DISCRETE SENSITIVITY ANALYSIS SUBROUTINES AS APPLIED TO HIVEL2D

APPENDIX B

IMPLEMENTATION AND COMPUTATIONAL COST OF DISCRETE SENSITIVITY ANALYSIS SUBROUTINES AS APPLIED TO HIVEL2D

B.1. Introduction

In this appendix, the implementations of the adjoint variable formulation and the direct formulation of discrete sensitivity analysis are presented, as well as the implementation of the flow prediction algorithm. After these implementations are presented, the computational cost is analyzed, and the reuse of code from the flow solver can be clearly seen. Regarding the computational expense, it is assumed that the steady state solution has been obtained and that the grid for the original set of design variables has been generated. The chief computational expenses are the determination of the matrix $\left[\frac{\partial W}{\partial Q}\right]$, the generation of the grid for each set of perturbed design variables, the calculation of the vector [W] for each set of perturbed design variables and the solution of one or more matrix equations. Since these activities are much more expensive than vector addition and multiplication, the cost of these simple vector manipulations is ignored. For the implementations presented below, there is only one objective function, and it is only dependent on the flow variables. If the objective function were also dependent on the grid χ or the design variables β, then additional computational cost would be needed to determine the effects of these dependencies.

B.2. Adjoint Variable Formulation

The steps involved in the implementation of the adjoint variable formulation when there is only one objective function F are presented below:

1. Determine the matrix $\left[\frac{\partial W}{\partial Q}\right]^T$ for the unperturbed grid by using the flow solver subroutines.
2. Calculate the vector $\left[\frac{\partial F}{\partial Q}\right]$. Since F is an analytic function of the flow variables Q, the vector can be determined analytically.
3. Use the flow solver subroutine to solve $\left[\frac{\partial W}{\partial Q}\right]^T [\lambda] = -\left[\frac{\partial F}{\partial Q}\right]$.
4. For each design variable β_k, these steps are necessary to estimate $\frac{dF}{d\beta_k}$:
   - 4.A. Generate grid for positively perturbed design variable, $\beta_k + \Delta\beta_k$.
   - 4.B. Determine $[W^+]$, from the flow solver.
   - 4.C. Generate grid for negatively perturbed design variable, $\beta_k - \Delta\beta_k$.
   - 4.D. Determine $[W^-]$, from the flow solver.
   - 4.E. Calculate $\left[\frac{dW}{d\beta_k}\right] = \frac{[W^+]-[W^-]}{2\Delta\beta_k}$. Store for use in flow prediction code.
   - 4.F. Calculate $\frac{dF}{d\beta_k} = [\lambda]^T \left[\frac{dW}{d\beta_k}\right]$.

The major components of the computational cost for the adjoint variable formulation is the need for 1 determination of $\left[\frac{\partial W}{\partial Q}\right]^T$, 1 solution of a matrix equation, $2N$ grid generations, and $2N$ calculations of $[W]$, where N is the number of design variables. The flow solver subroutines are used to calculate the matrix $\left[\frac{\partial W}{\partial Q}\right]^T$ and the vectors $[W^+]$ and $[W^-]$. These subroutines are also used to solve the matrix equation. Thus, the only subroutines that must be coded are those that calculate the vectors $\left[\frac{\partial F}{\partial Q}\right]$ and $\left[\frac{dW}{d\beta_k}\right]$ and the value of $\frac{dF}{d\beta_k}$.

B.3. Direct Formulation

Listed below are the steps in the implementation of the direct formulation of discrete sensitivity:

1. Calculate $\left[\frac{\partial W}{\partial Q}\right]$ for the unperturbed grid, via the flow solver.
2. Calculate the vector $\left[\frac{\partial F}{\partial Q}\right]$.
3. For each design variable β_k, do the following steps:

   3.A. Generate grid for positively perturbed design variable, $\beta_k + \Delta\beta_k$.

   3.B. Determine $[W^+]$, from the flow solver.

   3.C. Generate grid for negatively perturbed design variable, $\beta_k - \Delta\beta_k$.

   3.D. Determine $[W^-]$, from the flow solver.

   3.E. Calculate $\left[\frac{dW}{d\beta_k}\right] = \frac{[W^+]-[W^-]}{2\Delta\beta_k}$. Store for use in flow prediction code.

   3.F. Solve $\left[\frac{\partial W}{\partial Q}\right]\left[\frac{\partial Q}{\partial \beta_k}\right] = -\left[\frac{dW}{d\beta_k}\right]$, by using the flow solver.

   3.G. Calculate $\frac{dF}{d\beta_k} = \left[\frac{\partial F}{\partial Q}\right]^T \left[\frac{\partial Q}{\partial \beta_k}\right]$.

The computational cost is 1 determination of $\left[\frac{\partial W}{\partial Q}\right]$, $2N$ grid generations, $2N$ calculations of $[W]$ and N solutions of a matrix equation. The major difference in the computational cost of the two methods is that the direct formulation must solve a matrix equation for each design variable, whereas the adjoint variable formulation solves the matrix equation just once, since there is only one objective function.

When the objective function is of the form $F = \sum_{i=1}^{m} f_i^2$, then the steps 2 and 3.G are replaced by

2. Calculate the vector $\left[\frac{\partial f_i}{\partial Q}\right]$.

3.G. Calculate $\frac{df_i}{d\beta_k} = \left[\frac{\partial f_i}{\partial Q}\right]^T \left[\frac{\partial Q}{\partial \beta_k}\right]$.

3.H. $\frac{dF}{d\beta_k} = \sum_{i=1}^{m} 2 f_i \frac{\partial f_i}{\partial \beta_k}$.

3.I. $\frac{d^2 F}{d\beta_k d\beta_j} = \sum_{i=1}^{m} 2 \frac{\partial f_i}{\partial \beta_k} \frac{\partial f_i}{\partial \beta_j}$.

Steps 2 and 3.G are completed once for each residual function f_i. Note that $\left[\frac{\partial Q}{\partial \beta_k}\right]$ is calculated just once for each design variable β_k.

Once these steps are completed, the gradient ∇F and the approximate Hessian matrix $\nabla^2 F$ are used in the Gauss-Newton method to update the design variables as described in Appendix C. The additional computational expense is small, because only vector and scalar multiplication are used to obtain the additional information.

In regards to code reuse, the flow solver subroutines are again used to calculate the matrix $\left[\frac{\partial W}{\partial Q}\right]$ and the vectors $[W^+]$ and $[W^-]$. The flow solver code is also used to solve the matrix equations. The only code that must be written involves the calculation of the vector vector $\left[\frac{\partial F}{\partial Q}\right]$ or the vectors $\left[\frac{\partial f_i}{\partial Q}\right]$ and the vector manipulations involved in estimating the derivative $\frac{\partial F}{\partial \beta_k}$ or the derivatives $\frac{\partial f_i}{\partial \beta_k}$.

B.4. Flow Prediction Code

For grids whose node count and connectivity do not change as the design variables change, the flow prediction code can be used to reduce the number of iterations that are needed to obtain a steady-state solution for the grid associated with the new set of design variables. The flow prediction code is quite easy to implement once the discrete sensitivity analysis process has generated the matrix $\left[\frac{dW}{d\beta}\right]$ or $\left[\frac{\partial Q}{\partial \beta}\right]$ and the change in the design variables $\Delta\vec{\beta}$ has been determined. The columns in the matrix

$\left[\frac{dW}{d\beta}\right]$ are the vectors $\left[\frac{dW}{d\beta_k}\right]$ that were stored in the discrete sensitivity analysis process. Similarly, the columns in the matrix $\left[\frac{\partial Q}{\partial \beta}\right]$ are the vectors $\left[\frac{\partial Q}{\partial \beta_k}\right]$.

The steps in the flow prediction algorithm are

1. Obtain $\left[\frac{\partial W}{\partial Q}\right]$ for the unperturbed grid.
2. Multiply $\left[\frac{dW}{d\beta}\right] \Delta\vec{\beta}$ where $\left[\frac{\partial W}{\partial \beta}\right]$ and $\Delta\vec{\beta}$ were stored from the preceding calculations.
3. Use flow solver to solve $\left[\frac{\partial W}{\partial Q}\right] \Delta Q = -\left[\frac{dW}{d\beta}\right] \Delta\vec{\beta}$

Since $\left[\frac{\partial W}{\partial Q}\right]$, $\left[\frac{dW}{d\beta}\right]$ and $\Delta\vec{\beta}$ are available from the discrete sensitivity analysis process, the only computational expense arises from a matrix-vector multiplication and 1 solution of a matrix equation.

For the shallow water equations, the depth of flow must always be positive. Thus, in regions where the depth of flow is quite small, the flow prediction code may predict a flow depth that is not positive. In this case, the flow prediction code fails, and the flow variables are not updated.