Approximation Concepts and Sensitivity Analysis

Overview

- Design space dimensionality reduction
  - active constraint sets
  - design variable linking
- Approximate analysis
  - basis reduction
  - simplified analysis
- Explicit approximations
  - Taylor series
  - response surfaces

Design Variable Linking

- Simple reduction in the number of variables by assumed relationships

$$A_1 = A_3 = X(1)$$

$$A_2 = A_4 = X(2)$$

$$A_5 = A_6 = X(3)$$

$$A_7 = A_8 = A_9 = A_{10} = X(4)$$

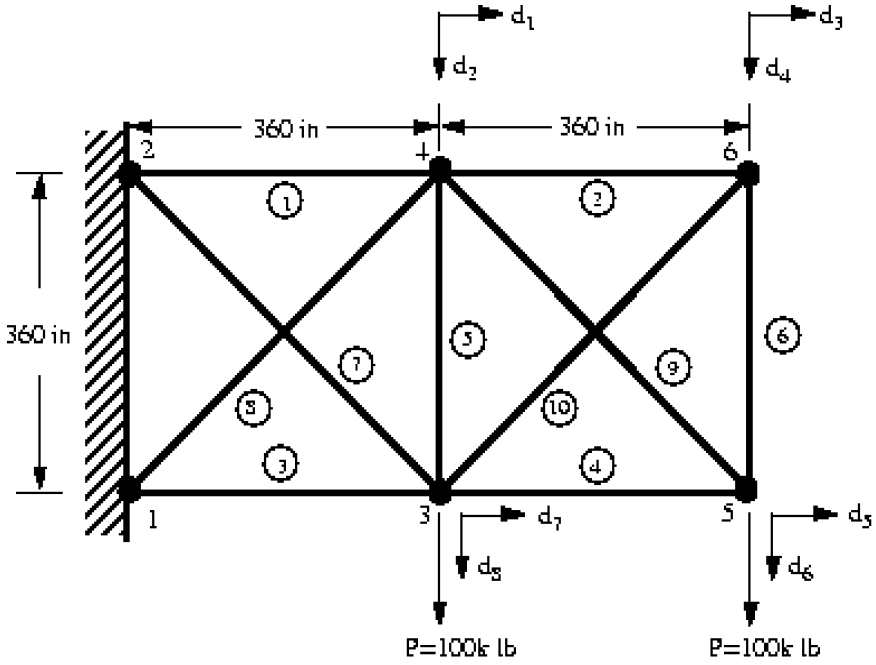
- Can also be functional relations of the type

$$A_1 = X(1) \quad A_3 = X(2)$$

$$A_2 = A_4 = 0.6X(1) + 0.4X(2)$$

$$A_5 = A_6 = X(3)$$

$$A_7 = A_8 = A_9 = A_{10} = 0.7X(3)$$



Constraint Reduction Methods

- Composite constraint representation - Kresselmeir-Stienhausser function

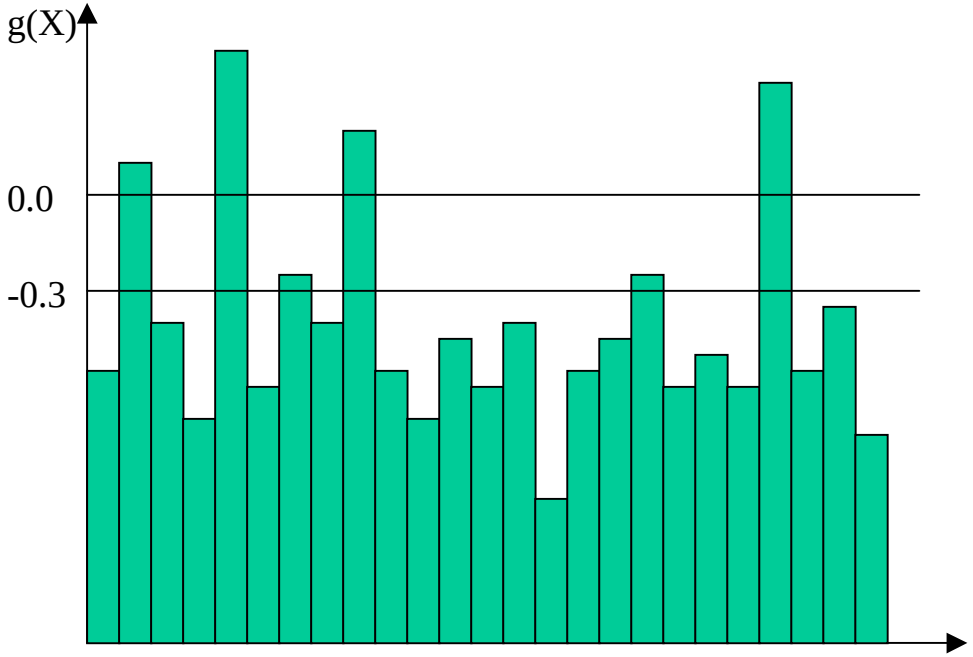
$$\Omega(X) = -\varepsilon + \frac{1}{\rho} \ln \sum_{j=1,m} \exp[\rho g_j(X)]$$

  - function is an envelope function that represents the aggregate of the most critical constraints in the set
  - choose $\rho \sim 25-30$ to make most critical constraint the dominant term in Ω

Constraint Deletion Strategy

- Many constraints are satisfied with a large margin - far from critical, and can be temporarily ignored from the optimization process
- Gradient calculations are significant source of computational expense
  - fewer constraints imply fewer gradient calculations
- Constraint deletion can be based on
  - deleting currently non-critical constraints
  - regionalization - applicable in structural design, for example

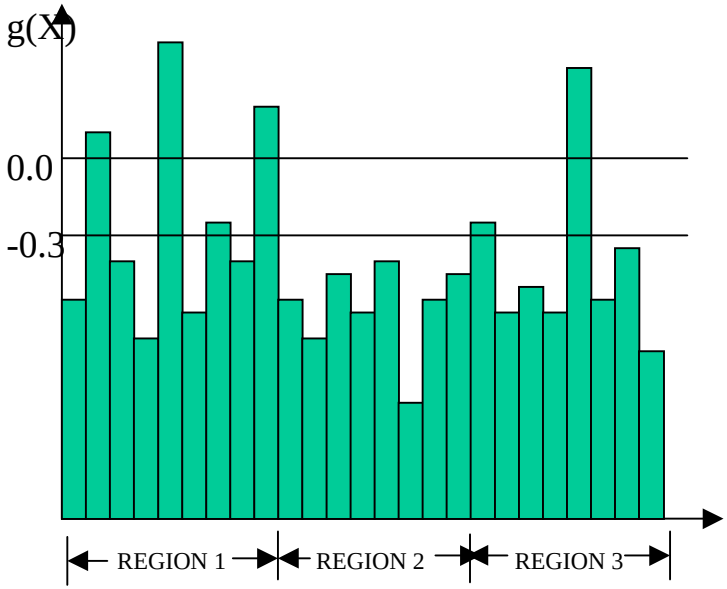
Constraint Deletion

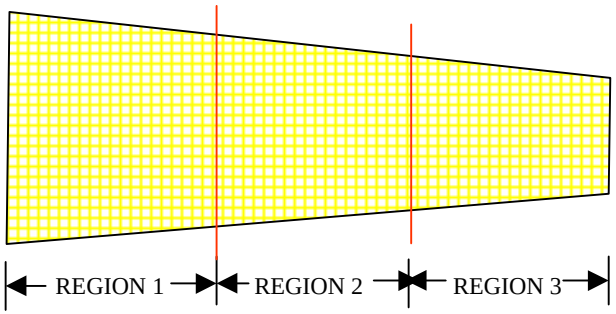


- Retain all constraints with g>-0.3 only

Constraint Deletion - Regionalization

- Only one constraint from each of the three regions of elements on the wing structure is retained
  - could be in addition to the constraint deletion based on constraint value alone





Approximate Analysis

- Basis reduction method
  - let a vector Y define the actual design
    - for example, Y could consist of 25 cross-sectional areas of truss elements
  - develop a number of design alternatives $Y^i, i = 1, N$

- The vector Y to be used in analysis is then computed as

$$Y = \sum_{i=1}^{N} X_i Y^i$$

- where X_i are the independent design variables

Basis Reduction

- Reduction in the number of independent design variables
  - will help establish an upper bound on the optimal objective function value
- Improves the conditioning of the optimization problem
- Careful selection of basis vectors greatly helps in getting to the true optimum
  - use prior designs/experience base to select basis vectors

Simplified Analysis

- Basis approach
  - create a detailed analysis model
  - create a simplified analysis model
    - compare results for the initial design using the two models and modify simplified model so that results correlate with those of exact model
  - optimize using simplified model and perform a detailed analysis of the optimal design
  - if agreement is good, stop, else modify simplified model and repeat
- Approach can be very efficient in those problems where detailed model is very expensive

Response Surface Approximations

- Particularly useful when only function information is available to create the approximations
  - if we have a number of designs X_i distributed over the design domain, we can fit a polynomial (assumed order) to these data points
    - full second order polynomial will require 1+N+N(N+1)/2 data points
  - works well for expensive analysis cases when number of variables is small
  - is also useful in those cases where function values are determined experimentally

Taylor Series Approximations

- As described earlier, one can create first or higher-order function approximations
  - first-order approximations are the basis of sequential linear programming
  - first-order approximation is preferred as second or higher order gradients are both expensive, and generally less accurate
- Assist in improving the quality of approximations by choosing variables appropriately
  - choose variables so that linear or quadratic models adequately model the response behavior

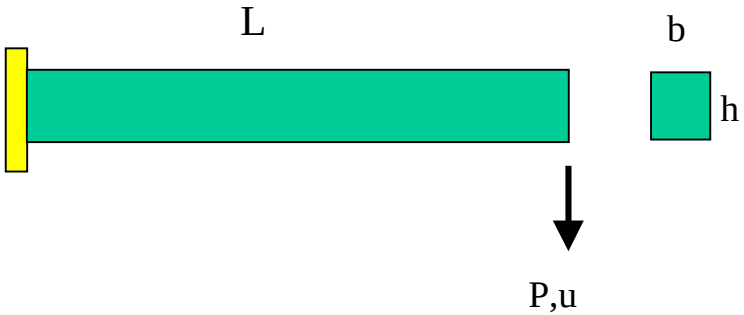
Use of Intermediate Variables

- If design variable is h, one can construct a linear approximation for u as follows (not exact)

$$\bar{u} = u_0 + \frac{\partial u}{\partial h}(h - h_0) = u_0 - \frac{12PL^3}{EBh^4}$$

- If design variable is chosen as the reciprocal moment of inertia, the approximation is exact

$$\bar{u} = u_0 + \frac{\partial u}{\partial I}\left(\frac{1}{I} - \frac{1}{I_0}\right)(-I_0^{\,2}) = \frac{PL^3}{3EI}$$



$$u_o = \frac{PL^3}{3EI} = \frac{4PL^3}{Ebh^3}$$

Use of Intermediate Response

- Consider approximation for stresses in rod elements
  - stress $\sigma = F / A$ and if A is design variable, then stress is nonlinearly varying with A
  - intermediate variable choice would be to designate 1/A as the design variable
  - alternative, approximate the internal force F as a linear function of A

$$\overline{F} = F^0 + \nabla F^T \Delta A$$

    - stresses are then computed explicitly from (nonlinear)

$$\overline{\sigma} = \frac{F^0 + \nabla F^T \Delta A}{A}$$

Conservative Approximations

- We would like to have approximate value of constraints to be more positive
  - such a conservative approximation ensures that the true constraint (when evaluated) is less likely to be violated
- This approximation is conservative with respect to a linear approximation (no guarantee with respect to actual constraint)
  - linear

$$g_L = g(X^0) + \sum_{i=1}^{N} \left.\frac{\partial g}{\partial X_i}\right|_{X^0} (X_i - X_i^{\,0})$$

  - reciprocal

$$g_R = g(X^0) - \sum_{i=1}^{N} \left.\frac{\partial g}{\partial X_i}\right|_{X^0} \left(\frac{1}{X_i} - \frac{1}{X_i^{\,0}}\right)\left(X_i^{\,0}\right)^2$$

Conservative Approximations

- If the approximation to g(X) is to be conservative
  - use linear approximation if $(X_i^0)\left.\frac{\partial g}{\partial X_i}\right|_{X^0} \geq 0$
  - use reciprocal approximation if $(X_i^0)\left.\frac{\partial g}{\partial X_i}\right|_{X^0} < 0$
- If X_i is close to zero or in the process of crossing zero, use linear approximation

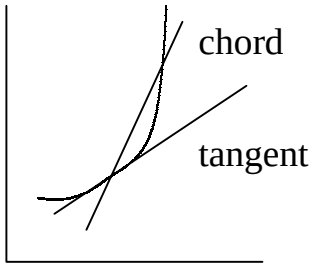
Sensitivity Analysis - Overview

- Finite difference method
- Quasi-analytical method
- Analytical method
- Coupled system sensitivity
- Problem parameter sensitivity

Finite Difference Approach

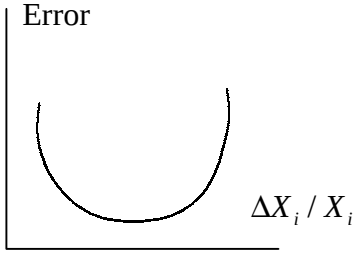
- Forward difference

$$\frac{df}{dX} = \frac{f(X+\Delta X) - f(X)}{\Delta X}$$



- Central difference

$$\frac{df}{dX} = \frac{f(X+\Delta X) - f(X-\Delta X)}{2\Delta X}$$



- Optimal step size must be chosen - optimal step size cannot simply be the smallest due to round-off errors

Quasi-Analytical Approach

- Apart from accuracy concerns, finite difference is very expensive if analysis is computationally cumbersome
  - for a N variable problem, each finite difference calculation requires N+1 function analyses
- Quasi-analytical method is based on carrying out an analytical differentiation, and alleviating computational costs and complexity by computing key derivatives by the finite difference approach
  - consider a set of equations as follows

$$F[X, u(X)] = 0$$

Quasi-Analytical Approach

- Analytically differentiating this equation with respect to the design variables X

$$\frac{\partial F}{\partial X} + \frac{\partial F}{\partial u}\frac{\partial u}{\partial X} = 0$$

- The terms $\frac{\partial F}{\partial X}$ and $\frac{\partial F}{\partial u}$ can be computed by finite difference

- As an example consider the finite element equation for static analysis [K]{u}={P}

Quasi-Analytical Approach

- Analytical differentiation with respect to X yields

$$\frac{\partial K}{\partial X}u + K\frac{\partial u}{\partial X} = \frac{\partial P}{\partial X}$$

  - or $$\frac{\partial u}{\partial X} = (K)^{-1}\left(\frac{\partial P}{\partial X} - \frac{\partial K}{\partial X}u\right)$$

- Recall that true finite difference would require new [K] matrix for each design perturbation, and factorization of this matrix to solve for corresponding displacement response

Analytical Sensitivity- Unit Load Method Applications in Structural Design

- If we need to selectively compute the sensitivity of a displacement component with respect to design variables, we could use the unit-load method
- First define [H] as

$$[H]=\left[\left\{\frac{\partial P}{\partial X_1}\right\},\cdots,\left\{\frac{\partial P}{\partial X_N}\right\}\right]-\left[\left[\frac{\partial K}{\partial X_1}\right]\{u\},\cdots,\left[\frac{\partial K}{\partial X_N}\right]\{u\}\right]$$

- Also, a desired displacement u_j is expressed as $u_j=\{Q_j\}^T\{u\}$ where $\{Q_j\}$ is a virtual load vector with unity at j-th location and zero elsewhere

Analytical Sensitivity- Unit Load Method Applications in Structural Design

- The differentiation of u_j with respect to X gives

$$\left\{\frac{\partial u_j}{\partial X}\right\}^T = \{Q_j\}^T\left[\frac{\partial u}{\partial X}\right]$$

- And since the virtual displacement corresponding to the virtual load is obtained as $[K]\{u_j{}^Q\} = \{Q_j\}$ the above equation becomes

$$\left\{\frac{\partial u_j}{\partial X}\right\}^T = \{u_j{}^Q\}^T[K]\left[\frac{\partial u}{\partial X}\right] = \{u_j{}^Q\}^T[H]$$

Coupled System Sensitivity

- What happens when finite difference approach is applied to a coupled complex system?
  - perturb one variable at a time, iterate the coupled analyses to convergence - everyone must act in concert (also very expensive)
  - small perturbations required to ensure accuracy may generate nothing more than numerical noise in a coupled system analysis - perturb isolated stringer and hope to see meaningful change in aircraft range???
  - large differences are employed but at the risk of severe loss in accuracy

Coupled System Sensitivity - GSE1

- Consider a multidisciplinary system with two subsystems A and B
  - system equations can be written in symbolic form as

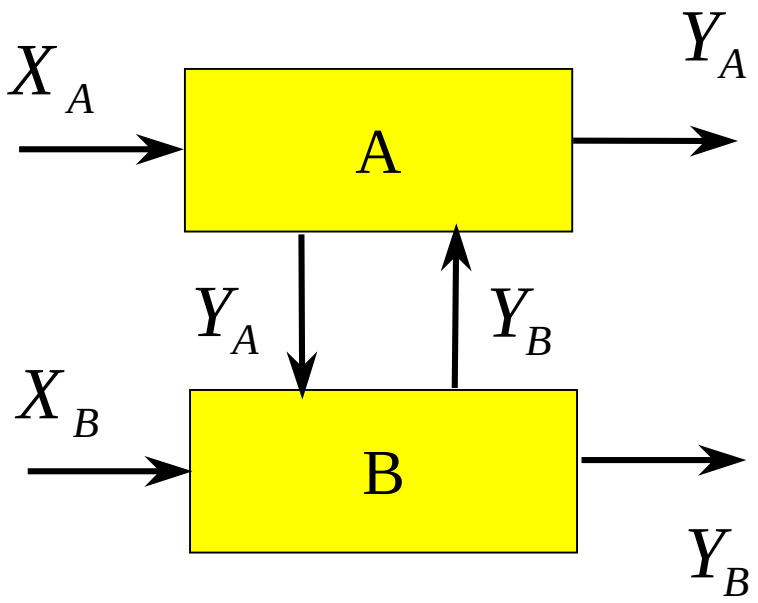
$$A[(X_A, Y_B), Y_A] = 0$$

$$B[(X_B, Y_A), Y_B] = 0$$

  - rewrite these as follows

$$Y_A = Y_A(X_A, Y_B)$$

$$Y_B = Y_B(X_B, Y_A)$$



Coupled System Sensitivity - GSE1

- First-order Taylor series representation

$$\frac{dY_A}{dX_A} = \frac{\partial Y_A}{\partial X_A} + \frac{\partial Y_A}{\partial Y_B}\frac{dY_B}{dX_A}$$
$$\frac{dY_B}{dX_B} = \frac{\partial Y_B}{\partial X_B} + \frac{\partial Y_B}{\partial Y_A}\frac{dY_A}{dX_B}$$

- And using a chain rule of derivatives

$$\frac{dY_A}{dX_B} = \frac{\partial Y_A}{\partial Y_B}\frac{\partial Y_B}{\partial X_A}$$
$$\frac{dY_B}{dX_A} = \frac{\partial Y_B}{\partial Y_A}\frac{\partial Y_A}{\partial X_A}$$

Coupled System Sensitivity - GSE1

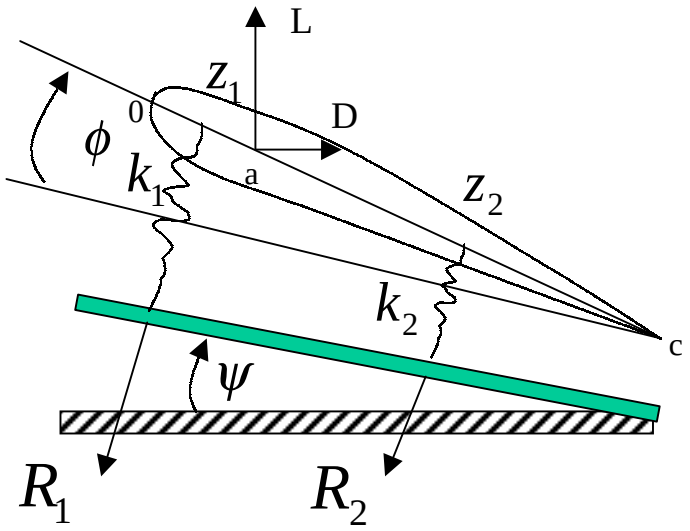
- These equations can be represented in matrix notation as

$$\begin{bmatrix} I & -\frac{\partial Y_A}{\partial Y_B} \\ -\frac{\partial Y_B}{\partial Y_A} & I \end{bmatrix} \begin{Bmatrix} \frac{dY_A}{dX_A} \\ \frac{dY_B}{dX_A} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial Y_A}{\partial X_A} \\ 0 \end{Bmatrix}$$

$$\begin{bmatrix} I & -\frac{\partial Y_A}{\partial Y_B} \\ -\frac{\partial Y_B}{\partial Y_A} & I \end{bmatrix} \begin{Bmatrix} \frac{dY_A}{dX_B} \\ \frac{dY_B}{dX_B} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \frac{\partial Y_B}{\partial X_B} \end{Bmatrix}$$

- Total derivatives can be computed if partial sensitivities computed in each subsystem are known - the latter can be computed locally within the subsystems

Example - Wing on Elastic Support



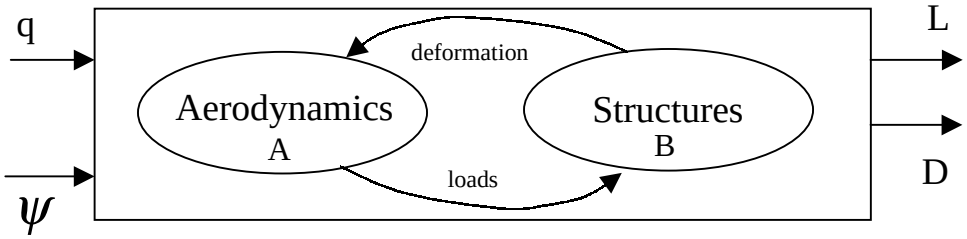
$b - span$

$c - chord$

$\overline{z_1} = z_1 / c; \overline{z_2} = z_2 / c; \overline{a} = a / c$

$\overline{h_1} = \overline{a} - \overline{z_1}; \overline{h_2} = -\overline{a} + \overline{z_2}$

$p = \overline{h_1} / \overline{h_2}$ S = b.c

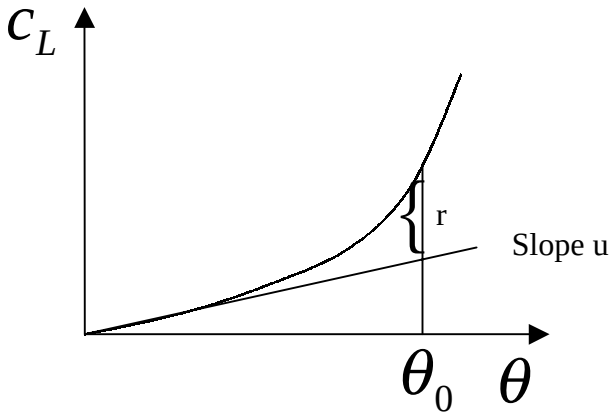


$B = 100cm, c = 10cm, \overline{z_1} = 0.2,$

$\overline{z_2} = 0.7, \overline{a} = 0.25, q = 1N / cm^2$

$k_1 = 4000N / cm, k_2 = 2000N / cm$

Example - Wing on Elastic Support



$$u = 5/rad; r = 2$$

$$\theta_0 = 0.26\,rad; \psi = 0.05\,rad$$

$$c_L = u\theta + r(1 - \cos((\pi/2)(\theta/\theta_0)))$$

$$q = \frac{1}{2}\rho V^2 \qquad d_i = R_i / k_i \quad \phi = \frac{d_1 - d_2}{z_2 - z_1}$$

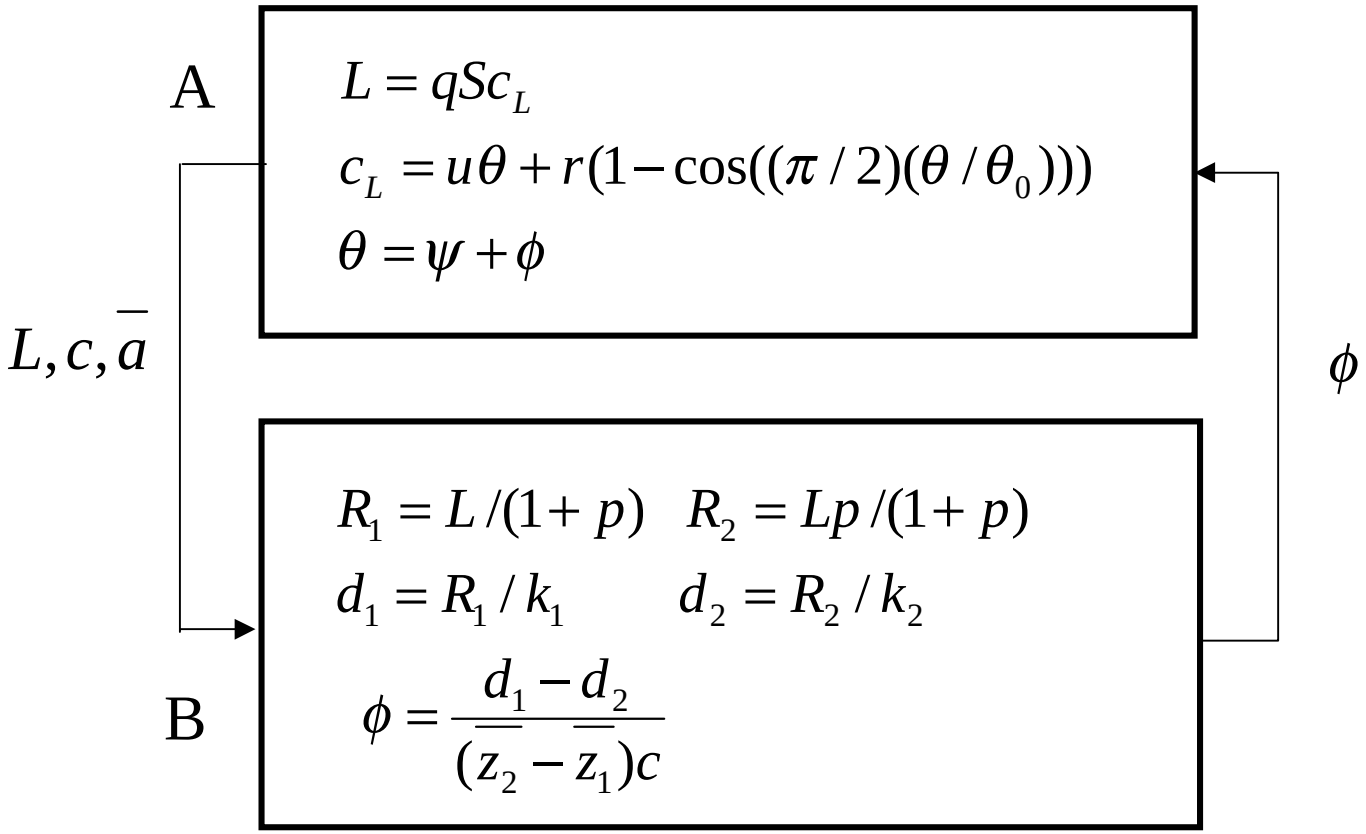
$$\theta = \psi + \phi$$

$$X = \{S, u, r, \theta_0, c, \bar{a}, \psi, \bar{z_1}, \bar{z_2}, k_1, k_2\}$$

$$Y_A = \{L, c, \bar{a}\}$$

$$Y_B = \{\phi\}$$

Wing on Elastic Support - Simplified



Wing on Elastic Support - Results

- Solution point is for L=502.3, $\phi = 0.0176, \psi = 0.05$ and we are interested in $dL/d\psi$ and $d\phi/d\psi$
- By finite differences (step size 0.0025) these are 14925.16 N/rad and 0.5221287 rad/rad
- The GSE equation becomes

$$\begin{bmatrix} I & -\frac{\partial L}{\partial \phi} \\ -\frac{\partial \phi}{\partial L} & I \end{bmatrix} \begin{Bmatrix} \frac{dL}{d\psi} \\ \frac{d\phi}{d\psi} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial L}{\partial \psi} \\ 0 \end{Bmatrix}$$

Wing on Elastic Support - Results

- From aerodynamics (locally)

$$\frac{\partial L}{\partial \psi} = 9805.105 = \frac{\partial L}{\partial \phi}$$

- From structures (locally)

$$\frac{\partial \phi}{\partial L} = 0.000035 rad / N$$

- Substituting in GGE equations and solving

$$\frac{dL}{d\psi} = 14928.12 \ \frac{N}{rad}, \ \ \frac{d\phi}{d\psi} = 0.5224841 \frac{rad}{rad}$$

Automatic Differentiation

- Contains a generic differentiation program
  - takes analysis code in C or in Fortran
  - takes user defined input on dependent and independent variables
  - introduces statements that generate sensitivity information
    - uses the chain rule to achieve this objective
  - modified program is the original + new statements that compute the required sensitivity
    - new statements are embedded calls to specific subroutines

Automatic Differentiation

- Program source increases in size
- Differentiation rules are similar to those in MACSYMA
  - while MACSYMA would have produced long symbolic expressions, this approach computes and saves only the numerical value of the differentiated expression
- Process is subject to only truncation errors - otherwise numerically exact
- Some extra storage is required to store derivative values - kept only as long as needed
- Process is neither finite differencing, symbolic, or quasi-analytical

Automatic Differentiation Example

Solve $y = x\cos(uxy)$

y – *dependendent*

x, u – *independent*

Given u & x

```
      Y = YIN
      DO 100 I = 1, N1
      YNXT = X * COS(U * X * Y)
      IF((YNXT - Y)/YNXT.LE.TOL)GOTO 101
      Y = YNXT
100   CONTINUE
101   CONTINUE
      Y = YNXT
      PRINT, Y
      END
```

Solve $y = x\cos(uxy)$

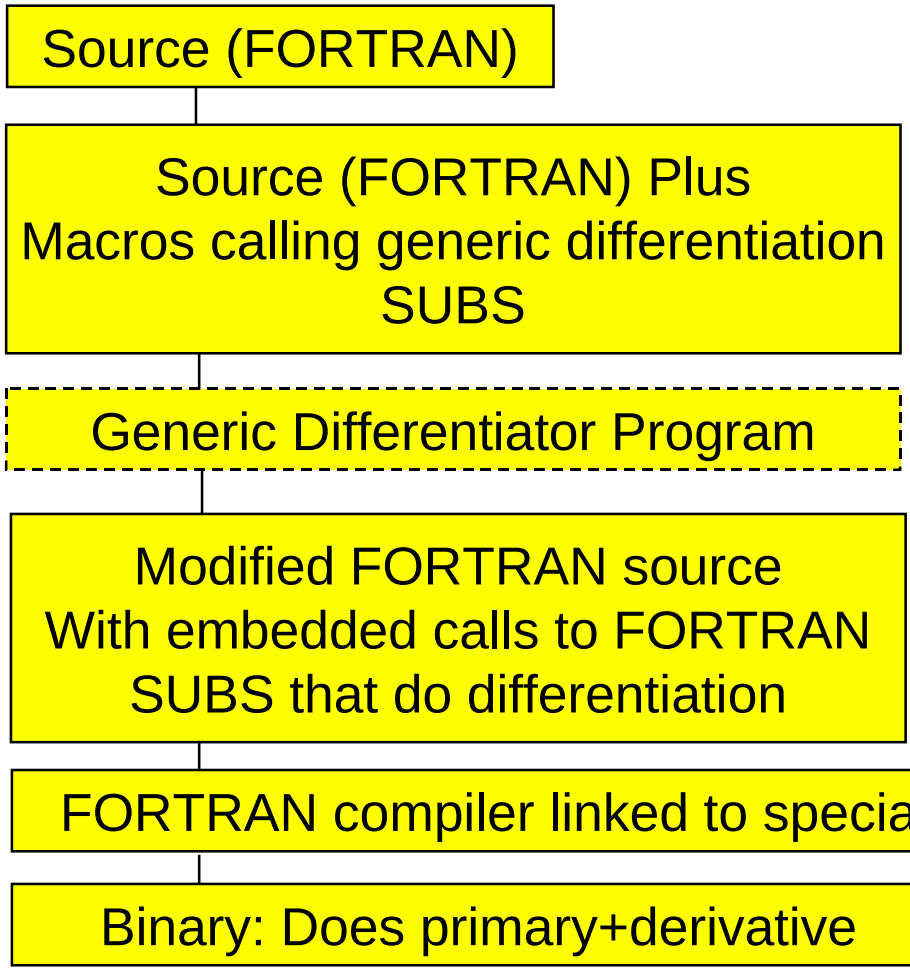
y – *dependendent*

x, u – *independent*

Given u & x

```
      Y = YIN          DY = DYIN
      DO 100 I = 1, N1
      YNXT = X * COS(U * X * Y)
      DYNXT=COS(U*X*Y) - X*U*Y*SIN(U*X*Y)
           + (-X * U * X * SIN(U * X * Y) * DY
      IF((YNXT - Y)/YNXT.LE.TOL)GOTO 101
      Y = YNXT         DY = DYNXT
100   CONTINUE
101   CONTINUE
      Y = YNXT         DY = DYNXT
      PRINT, Y
      END
```

Automatic Differentiation
Sequence of Operations



User designates dependent
and independent variables
by means of macros

Program that recognizes where to
differentiate and to apply chain rule

Specific FORTRAN SUBS “Know”
differentiation

Specific libraries to satisfy
Calls to SUBS

FINISHED PRODUCT

Sensitivity Based on Complex Variables

- Designating X as a vector of complex numbers $X_i = X_i^r + ih$

- Carry out a Taylor series expansion with a complex step size

$$f(x+i\Delta x) = f(x)+i(\Delta x)f'(x)-(\Delta x)^2 f''(x)/2!-i(\Delta x)^3 f'''(x)/3!+\ldots$$

- Using the imaginary part of the function yields

$$f'(x) \sim \mathrm{Im}[f(x+i\Delta x)]/\Delta x$$

- Has truncation errors of $O(\Delta x^2)$

Sensitivity Based on Complex Variables

- Given an analysis code with inputs x1, x2, x3 and output y
- Simply declare x and y to be both complex numbers
- For step length Δx initialize Rex1 + Δx, Rex2+i0, and Rex3+i0, and compute the complex output y
- Compute derivative dy/dx1=Im(y)/ Δx
- Reset with step size on variable x2, and repeat.

- Note that this approach can be used with selected xi components only.

Sensitivity Based on Complex Variables

- Advantages
  - No need to code anything except dy/dx1=Im(y)/ Δx
  - Insensitive to h
- Disadvantage
  - Cost about the same as in ordinary finite differencing with added overhead for handling complex numbers

Sensitivity Based on Complex Variables

- Carry out a Taylor series expansion with a complex step size

$$f(x+i\Delta x) = f(x) + i(\Delta x) f'(x) - (\Delta x)^2 f''(x)/2! - i(\Delta x)^3 f'''(x)/3! + \ldots$$

- Using the imaginary part of the function yields

$$f'(x) \sim \mathrm{Im}[f(x+i\Delta x)]/\Delta x$$

- Has truncation errors of $O(\Delta x^2)$

Sensitivity Based on Complex Variables

- Given an analysis code with inputs x1, x2, x3 and output y
- Simply declare x and y to be both complex numbers
- For step length Δx initialize Rex1 + Δx, Rex2+i0, and Rex3+i0, and compute the complex output y
- Compute derivative dy/dx1=Im(y)/ Δx
- Reset with step size on variable x2, and repeat.

- Note that this approach can be used with selected xi components only.

Sensitivity of Optimum WRT Preassigned Problem Parameters

- Useful in obtaining estimates of new optimal design due to a perturbation in one or more problem parameters
- This sensitivity is of practical value in hierarchical decomposition strategies

$$\textit{Minimize} \quad f(X, p)$$

$$\textit{Subject to} \quad g_j(X, p) \le 0 \qquad j = 1, m$$

- Optimal solution $f^*(X^*, p)$ *and* $X^*(p)$, $g_k(X^*, p) = 0$ are active constraints at the optimum
- Sensitivities *df*/dp and dX*/dp* can be obtained by differentiation of the Kuhn-Tucker conditions of optimality
  - requires active constraints remain active after perturbation
  - requires second-order sensitivity information

Sensitivity of Optimum WRT Preassigned Problem Parameters

- Assume that an optimal solution X* has been obtained - the K-T conditions for optimality

$$\nabla f(X^*) + \sum_{j \in J} \lambda_j \nabla g_j(X^*) = 0$$

$$g_j(X^*) = 0, \ \lambda_j > 0, \ \ j \in J$$

- For some small change in problem parameter, we require that K-T conditions remain valid - differentiating these conditions wrt p one obtains

$$\begin{bmatrix} A_{nxn} & B_{nxJ} \\ B^T{}_{Jxn} & O_{JxJ} \end{bmatrix} \begin{Bmatrix} \delta X \\ \delta \lambda \end{Bmatrix} + \begin{Bmatrix} c_{nx1} \\ d_{Jx1} \end{Bmatrix} = 0$$

Sensitivity of Optimum WRT Preassigned Problem Parameters

- Where the submatrices and sub-vectors are defined as follows

$$A_{ik} = \frac{\partial^2 f}{\partial X_i \partial X_k} + \sum_{j \in J} \lambda_j \frac{\partial^2 g_j}{\partial X_i \partial X_k} \qquad B_{ij} = \frac{\partial g_j}{\partial X_i}, \; j \in J$$

$$c_i = \frac{\partial^2 f}{\partial X_i \partial p} + \sum_{j \in J} \lambda_j \frac{\partial^2 g_j}{\partial X_i \partial p} \qquad d_j = \frac{\partial g_j}{\partial p}, \; j \in J$$

$$\delta X = \begin{Bmatrix} \frac{\partial X_1}{\partial p} \\ \bullet \\ \bullet \\ \frac{\partial X_n}{\partial p} \end{Bmatrix} \qquad \delta \lambda = \begin{Bmatrix} \frac{\partial \lambda_1}{\partial p} \\ \bullet \\ \bullet \\ \frac{\partial \lambda_n}{\partial p} \end{Bmatrix}$$