A COUPLED-ADJOINT METHOD FOR HIGH-FIDELITY AERO-STRUCTURAL OPTIMIZATION

Joaquim Rafael Rost Ávila Martins

Department of Aeronautics and Astronautics
Stanford University
Outline

• Introduction
  – High-fidelity aircraft design
  – Aero-structural optimization
  – Optimization and sensitivity analysis methods

• Complex-step derivative approximation

• Coupled-adjoint method
  – Sensitivity equations for multidisciplinary systems
  – Lagged aero-structural adjoint equations

• Results
  – Aero-structural sensitivity validation
  – Optimization results

• Conclusions
High-Fidelity Aerodynamic Shape Optimization

- Start from a baseline geometry provided by a conceptual design tool.

- High-fidelity models required for transonic configurations where shocks are present, high-dimensionality required to smooth these shocks.

- Accurate models also required for complex supersonic configurations, subtle shape variations required to take advantage of favorable shock interference.

- Large numbers of design variables and high-fidelity models incur a large cost.
Aero-Structural Aircraft Design Optimization

- Aerodynamics and structures are core disciplines in aircraft design and are very tightly coupled.

- For traditional designs, aerodynamicists know the spanload distributions that lead to the true optimum from experience and accumulated data. What about unusual designs?

- Want to simultaneously optimize the aerodynamic shape and structure, since there is a trade-off between aerodynamic performance and structural weight, e.g.,

\[
\text{Range} \propto \frac{L}{D} \ln \left( \frac{W_i}{W_f} \right)
\]
The Need for Aero-Structural Sensitivities

- Sequential optimization does not lead to the true optimum.

- Aero-structural optimization requires coupled sensitivities.

- Add structural element sizes to the design variables.

- Including structures in the high-fidelity wing optimization will allow larger changes in the design.
minimize \( I(x) \)
\[ x \in \mathbb{R}^n \]
subject to \( g_m(x) \geq 0, \ m = 1, 2, \ldots, N_g \)

- \( I \): objective function, output (e.g. structural weight).
- \( x_n \): vector of design variables, inputs (e.g. aerodynamic shape); bounds can be set on these variables.
- \( g_m \): vector of constraints (e.g. element von Mises stresses); in general these are nonlinear functions of the design variables.
Optimization Methods

- **Intuition:** decreases with increasing dimensionality.

- **Grid or random search:** the cost of searching the design space increases rapidly with the number of design variables.

- **Genetic algorithms:** good for discrete design variables and very robust; but infeasible when using a large number of design variables.

- **Nonlinear simplex:** simple and robust but inefficient for more than a few design variables.

- **Gradient-based:** the most efficient for a large number of design variables; assumes the objective function is “well-behaved”.

Ph.D. Oral Examination, Stanford University, September 2002
• Analysis computes objective function and constraints (e.g. aero-structural solver)

• Optimizer uses the sensitivity information to search for the optimum solution (e.g. sequential quadratic programming)

• Sensitivity calculation is usually the bottleneck in the design cycle.

• Accuracy of the sensitivities is important, specially near the optimum.
Sensitivity Analysis Methods

- **Finite Differences**: very popular; easy, but lacks robustness and accuracy; run solver $N_x$ times.

\[
\frac{df}{dx_n} \approx \frac{f(x_n + h) - f(x)}{h} + \mathcal{O}(h)
\]

- **Complex-Step Method**: relatively new; accurate and robust; easy to implement and maintain; run solver $N_x$ times.

\[
\frac{df}{dx_n} \approx \frac{\text{Im} [f(x_n + ih)]}{h} + \mathcal{O}(h^2)
\]

- **Algorithmic/Automatic/Computational Differentiation**: accurate; ease of implementation and cost varies.

- **(Semi)-Analytic Methods**: efficient and accurate; long development time; cost can be independent of $N_x$. 

Ph.D. Oral Examination, Stanford University, September 2002
Research Contributions

- **Complex-step derivative approximation**
  - Contributed new insights to the theory of this approximation, including the connection to algorithmic differentiation.
  - Automated the implementation of the complex-step method for Fortran codes.
  - Demonstrated the value of this method for sensitivity validation.

- **Coupled-adjoint method**
  - Developed the general formulation of the coupled-adjoint for multidisciplinary systems.
  - Implemented this method in a high-fidelity aero-structural solver.
  - Demonstrated the usefulness of the resulting framework by performing aero-structural optimization.
The History of the Complex-Step Method

• Lyness & Moler, 1967:
  \( n^{th} \) derivative approximation by integration in the complex plane

• Squire & Trapp, 1998:
  Simple formula for first derivative.

• Newman, Anderson & Whitfield, 1998:
  Applied to aero-structural code.

• Anderson, Whitfield & Nielsen, 1999:
  Applied to 3D Navier-Stokes with turbulence model.

• Martins et al., 2000, 2001:
  Automated Fortran and C/C++ implementation. 3D aero-structural code.
Finite-Difference Derivative Approximations

From Taylor series expansion,

\[ f(x + h) = f(x) + hf'(x) + h^2 \frac{f''(x)}{2!} + h^3 \frac{f'''(x)}{3!} + \ldots. \]

Forward-difference approximation:

\[ \frac{df(x)}{dx} = \frac{f(x + h) - f(x)}{h} + O(h). \]

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(x) )</td>
<td>1.234567890123484</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( f(x + h) )</td>
<td>1.234567890123456</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \Delta f )</td>
<td>0.0000000000000028</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ph.D. Oral Examination, Stanford University, September 2002
Complex-Step Derivative Approximation

Can also be derived from a Taylor series expansion about \( x \) with a complex step \( ih \):

\[
f(x + ih) = f(x) + ihf'(x) - h^2 \frac{f''(x)}{2!} - ih \frac{f'''(x)}{3!} + \ldots
\]

\[
\Rightarrow f'(x) = \frac{\text{Im}[f(x + ih)]}{h} + h^2 \frac{f'''(x)}{3!} + \ldots
\]

\[
\Rightarrow f'(x) \approx \frac{\text{Im}[f(x + ih)]}{h}
\]

No subtraction! Second order approximation.
Simple Numerical Example

Estimate derivative at $x = 1.5$ of the function,

$$f(x) = \frac{e^x}{\sqrt{\sin^3 x + \cos^3 x}}$$

Relative error defined as:

$$\varepsilon = \frac{|f' - f'_{ref}|}{|f'_{ref}|}$$
Algorithmic Differentiation vs. Complex Step

Look at a simple operation, e.g. \( f = x_1 x_2 \),

<table>
<thead>
<tr>
<th>Algorithmic</th>
<th>Complex-Step</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \Delta x_1 = 1 )</td>
<td>( h_1 = 10^{-20} )</td>
</tr>
<tr>
<td>( \Delta x_2 = 0 )</td>
<td>( h_2 = 0 )</td>
</tr>
<tr>
<td>( f = x_1 x_2 )</td>
<td>( f = (x_1 + i h_1)(x_2 + i h_2) )</td>
</tr>
<tr>
<td>( \Delta f = x_1 \Delta x_2 + x_2 \Delta x_1 )</td>
<td>( f = x_1 x_2 - h_1 h_2 + i(x_1 h_2 + x_2 h_1) )</td>
</tr>
<tr>
<td>( df/dx_1 = \Delta f )</td>
<td>( df/dx_1 = \text{Im } f/h )</td>
</tr>
</tbody>
</table>

Complex-step method computes one extra term.

- Other functions are similar:
  - Superfluous calculations are made.
  - For \( h \leq x \times 10^{-20} \) they vanish but still affect speed.
Implementation Procedure

• Cookbook procedure for any programming language:
  – Substitute all real type variable declarations with complex declarations.
  – Define all functions and operators that are not defined for complex arguments.
  – A complex-step can then be added to the desired variable and the derivative can be estimated by $f' \approx \Im[f(x + ih)]/h$.

• Fortran 77: write new subroutines, substitute some of the intrinsic function calls by the subroutine names, e.g. abs by c_abs. But ... need to know variable types in original code.

• Fortran 90: can overload intrinsic functions and operators, including comparison operators. Compiler knows variable types and chooses correct version of the function or operator.

• C/C++: also uses function and operator overloading.
Fortran Implementation

- complexify.f90: a module that defines additional functions and operators for complex arguments.

- Complexify.py: Python script that makes necessary changes to source code, e.g., type declarations.

- Features:
  - Compatible with many platforms and compilers.
  - Supports MPI based parallel implementations.
  - Resolves some of the input and output issues.

- Application to aero-structural framework: 130,000 lines of code in 191 subroutines, processed in 42 seconds.

Outline

• Introduction
  – High-fidelity aircraft design
  – Aero-structural optimization
  – Optimization and sensitivity analysis methods

• Complex-step derivative approximation

• Coupled-adjoint method
  – Sensitivity equations for multidisciplinary systems
  – Lagged aero-structural adjoint equations

• Results
  – Aero-structural sensitivity validation
  – Optimization results

• Conclusions
Objective Function and Governing Equations

Want to minimize scalar objective function,

\[ I = I(x_n, y_i), \]

which depends on:

- \( x_n \): vector of design variables, e.g. structural plate thickness.
- \( y_i \): state vector, e.g. flow variables.

Physical system is modeled by a set of governing equations:

\[ R_k(x_n, y_i(x_n)) = 0, \]

where:

- Same number of state and governing equations, \( i, k = 1, \ldots, N_R \)
- \( N_x \) design variables.
Sensitivity Equations

Total sensitivity of the objective function:

\[ \frac{dI}{dx_n} = \frac{\partial I}{\partial x_n} + \frac{\partial I}{\partial y_i} \frac{dy_i}{dx_n}. \]

Total sensitivity of the governing equations:

\[ \frac{dR_k}{dx_n} = \frac{\partial R_k}{\partial x_n} + \frac{\partial R_k}{\partial y_i} \frac{dy_i}{dx_n} = 0. \]
Solving the Sensitivity Equations

Solve the total sensitivity of the governing equations

\[
\frac{\partial R_k}{\partial y_i} \frac{dy_i}{dx_n} = -\frac{\partial R_k}{\partial x_n}.
\]

Substitute this result into the total sensitivity equation

\[
\frac{dI}{dx_n} = \frac{\partial I}{\partial x_n} - \frac{\partial I}{\partial y_i} \left[ \frac{\partial R_k}{\partial y_i} \right]^{-1} \frac{\partial R_k}{\partial x_n},
\]

where \( \Psi_k \) is the adjoint vector.
Adjoint Sensitivity Equations

Solve the adjoint equations

\[
\frac{\partial \mathcal{R}_k}{\partial y_i} \Psi_k = -\frac{\partial I}{\partial y_i}.
\]

Adjoint vector is valid for all design variables.

Now the total sensitivity of the the function of interest \( I \) is:

\[
\frac{dI}{dx_n} = \frac{\partial I}{\partial x_n} + \Psi_k \frac{\partial \mathcal{R}_k}{\partial x_n}
\]

The partial derivatives are inexpensive, since they don’t require the solution of the governing equations.
Aero-Structural Adjoint Equations

Two coupled disciplines: Aerodynamics ($A_k$) and Structures ($S_l$).

\[ R_{k'} = \begin{bmatrix} A_k \\ S_l \end{bmatrix}, \quad y_i' = \begin{bmatrix} w_i \\ u_j \end{bmatrix}, \quad \Psi_{k'} = \begin{bmatrix} \psi_k \\ \phi_l \end{bmatrix}. \]

Flow variables, $w_i$, five for each grid point.

Structural displacements, $u_j$, three for each structural node.
Aero-Structural Adjoint Equations

\[
\begin{bmatrix}
\frac{\partial A_k}{\partial w_i} & \frac{\partial A_k}{\partial u_j} \\
\frac{\partial S_l}{\partial w_i} & \frac{\partial S_l}{\partial u_j}
\end{bmatrix}^T \begin{bmatrix}
\psi_k \\
\phi_l
\end{bmatrix} = - \begin{bmatrix}
\frac{\partial I}{\partial w_i} \\
\frac{\partial I}{\partial u_j}
\end{bmatrix}.
\]

- \( \partial A_k / \partial w_i \): a change in one of the flow variables affects only the residuals of its cell and the neighboring ones.
- \( \partial A_k / \partial u_j \): wing deflections cause the mesh to warp, affecting the residuals.
- \( \partial S_l / \partial w_i \): since \( S_l = K_{lj} u_j - f_l \), this is equal to \( -\partial f_l / \partial w_i \).
- \( \partial S_l / \partial u_j \): equal to the stiffness matrix, \( K_{lj} \).
- \( \partial I / \partial w_i \): for \( C_D \), obtained from the integration of pressures; for stresses, its zero.
- \( \partial I / \partial u_j \): for \( C_D \), wing displacement changes the surface boundary over which drag is integrated; for stresses, related to \( \sigma_m = S_{mj} u_j \).
Lagged Aero-Structural Adjoint Equations

Since the factorization of the complete residual sensitivity matrix is impractical, decouple the system and lag the adjoint variables,

\[
\frac{\partial A_k}{\partial w_i} \psi_k = - \frac{\partial I}{\partial w_i} - \frac{\partial S_l}{\partial w_i} \tilde{\phi}_l, \\
\text{Aerodynamic adjoint}
\]

\[
\frac{\partial S_l}{\partial u_j} \phi_l = - \frac{\partial I}{\partial u_j} - \frac{\partial A_k}{\partial u_j} \tilde{\psi}_k, \\
\text{Structural adjoint}
\]

Lagged adjoint equations are the single discipline ones with an added forcing term that takes the coupling into account.

System is solved iteratively, much like the aero-structural analysis.
Total Sensitivity

The aero-structural sensitivities of the drag coefficient with respect to wing shape perturbations are,

\[
\frac{dI}{dx_n} = \frac{\partial I}{\partial x_n} + \psi_k \frac{\partial A_k}{\partial x_n} + \phi_l \frac{\partial S_l}{\partial x_n}.
\]

- \(\partial I/\partial x_n\): \(C_D\) changes when the boundary over which the pressures are integrated is perturbed; stresses change when nodes are moved.

- \(\partial A_k/\partial x_n\): the shape perturbations affect the grid, which in turn changes the residuals; structural variables have no effect on this term.

- \(S_l/\partial x_n\): shape perturbations affect the structural equations, so this term is equal to \(\partial K_{lj}/\partial x_n u_j - \partial f_l/\partial x_n\).
3D Aero-Structural Design Optimization Framework

- Aerodynamics: FLO107-MB, a parallel, multiblock Navier-Stokes flow solver.
- Structures: detailed finite element model with plates and trusses.
- Coupling: high-fidelity, consistent and conservative.
- Geometry: centralized database for exchanges (jig shape, pressure distributions, displacements.)
- Coupled-adjoint sensitivity analysis: aerodynamic and structural design variables.
**Sensitivity of** $C_D$ **wrt Shape**

![Graph showing sensitivity of $C_D$ with respect to shape](image)

- **Coupled adjoint**
- **Complex step**
- **Coupled adjoint, fixed displacements**
- **Complex step, fixed displacements**

Avg. rel. error = 3.5%

**Design variable, $n$**

1 2 3 4 5 6 7 8 9 10
Sensitivity of $C_D$ wrt Structural Thickness

![Graph showing sensitivity of $C_D$ with respect to structural thickness. The graph plots the design variable $n$ against the ratio $dC_D/dx_n$, with design variables 11 to 20 along the x-axis and the ratio values along the y-axis. Two lines are shown: one for coupled adjoint and another for complex step, with the average relative error being 1.6%.]
Structural Stress Constraint Lumping

To perform structural optimization, we need the sensitivities of all the stresses in the finite-element model with respect to many design variables.

There is no method to calculate this matrix of sensitivities efficiently.

Therefore, lump stress constraints

\[ g_m = 1 - \frac{\sigma_m}{\sigma_{\text{yield}}} \geq 0, \]

using the Kreisselmeier–Steinhauser function

\[ KS(g_m) = -\frac{1}{\rho} \ln \left( \sum_m e^{-\rho g_m} \right), \]

where \( \rho \) controls how close the function is to the minimum of the stress constraints.
Sensitivity of KS wrt Shape

Design variable, $n$

-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Avg. rel. error = 2.9%

Coupled adjoint
Complex step
Coupled adjoint, fixed loads
Complex, fixed loads

Ph.D. Oral Examination, Stanford University, September 2002
Sensitivity of KS wrt Structural Thickness

Avg. rel. error = 1.6%

Ph.D. Oral Examination, Stanford University, September 2002
Computational Cost vs. Number of Variables

- Coupled adjoint
- Finite difference
- Complex step

Normalized time vs. Number of design variables ($N_x$)

- $2.1 \times 0.92 N_x$
- $1.0 + 0.38 N_x$
- $3.4 + 0.01 N_x$
Computational Cost Breakdown

\[ \frac{\partial A_k}{\partial w_i} \psi_k = - \frac{\partial I}{\partial w_i} - \frac{\partial S_l}{\partial w_i} \phi_l \]

0.60

\[ \frac{\partial S_l}{\partial u_j} \phi_l = - \frac{\partial I}{\partial u_j} - \frac{\partial A_k}{\partial u_j} \psi_k \]

< 0.001

\[ \frac{dI}{dx_n} = \frac{\partial I}{\partial x_n} + \psi_k \frac{\partial A_k}{\partial x_n} + \phi_l \frac{\partial S_l}{\partial x_n} \]

0.01Nx
Supersonic Business Jet Optimization Problem

Minimize:

\[ I = \alpha C_D + \beta W \]

where \( C_D \) is that of the cruise condition.

Subject to:

\[ KS(\sigma_m) \geq 0 \]

where KS is taken from a maneuver condition.

With respect to: external shape and internal structural sizes.

Natural laminar flow
supersonic business jet
Mach = 1.5, Range = 5,300nm
1 count of drag = 310 lbs of weight
Baseline Design

\[ C_D = 0.007395 \]
\[ \text{Weight} = 9,285 \text{ lbs} \]

Surface density (cruise)

\[
\begin{array}{c|c|c}
0.5 & \text{green} & 1.4 \\
\end{array}
\]

Von Mises stresses (maneuver)

\[
\begin{array}{c|c|c}
0.0 & \text{blue} & 1.0 \\
\end{array}
\]
Design Variables

Total of 97 design variables

10 Hicks-Henne bumps

TE camber

Twist

LE camber

6 defining airfoils

9 bumps along fuselage axis

10 skin thickness groups
Aero-Structural Optimization Convergence History

Ph.D. Oral Examination, Stanford University, September 2002
Aero-Structural Optimization Results

\[ C_D = 0.006922 \]

Weight = 5,546 lbs

Surface density (cruise)

0.5 \quad \textcolor{red}{1.4}

Von Mises stress (maneuver)

0.0 \quad \textcolor{red}{1.0}
### Comparison with Sequential Optimization

<table>
<thead>
<tr>
<th></th>
<th>$C_D$ (counts)</th>
<th>$\frac{\sigma_{\text{max}}}{\sigma_{\text{yield}}}$</th>
<th>Weight (lbs)</th>
<th>Objective</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All-at-once approach</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>73.95</td>
<td>0.87</td>
<td>9,285</td>
<td>103.90</td>
</tr>
<tr>
<td>Optimized</td>
<td>69.22</td>
<td>0.98</td>
<td>5,546</td>
<td>87.11</td>
</tr>
<tr>
<td><strong>Sequential approach</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aerodynamic optimization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>74.04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimized</td>
<td>69.92</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Structural optimization</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>0.89</td>
<td></td>
<td>9,285</td>
<td></td>
</tr>
<tr>
<td>Optimized</td>
<td>0.98</td>
<td></td>
<td>6,567</td>
<td></td>
</tr>
<tr>
<td>Aero-structural analysis</td>
<td>69.95</td>
<td>0.99</td>
<td>6,567</td>
<td>91.13</td>
</tr>
</tbody>
</table>
Conclusions

- Shed a new light on the theory behind the complex-step method.
- Showed the connection between complex-step and algorithmic differentiation theories.
- The complex-step method is excellent for validation of more sophisticated gradient calculation methods, like the coupled-adjoint.
Conclusions

- Developed the general formulation for a coupled-adjoint method for multidisciplinary systems.
- Applied this method to a high-fidelity aero-structural solver.
- Showed that the computation of sensitivities using the aero-structural adjoint is extremely accurate and efficient.
- Demonstrated the usefulness of the coupled adjoint by optimizing a supersonic business jet configuration.
Long Term Vision

Continue work on a large-scale MDO framework for aircraft design
Acknowledgments
Acknowledgments
Acknowledgments
CFD and OML grids
Displacement Transfer
Mesh Perturbation

1. Baseline
2. Perturbed
3. 1
4. 2, 3
Load Transfer

OML point

CFD mesh point

\( u \)

\( v \)
Aero-Structural Iteration

\[ w^{(0)} = w_\infty \]

\[ u^{(0)} = 0 \]

Load transfer

Displacement transfer