Adjoint Approaches in Aerodynamic Shape Optimization and MDO Context I/II

Nicolas Gauger 1), 2)

1) DLR Braunschweig
   Institute of Aerodynamics and Flow Technology
   Numerical Methods Branch

2) Humboldt University Berlin
   Department of Mathematics

http://www.mathematik.hu-berlin.de/~gauger
Collaborators

With contributions to this lecture:

• DLR: N. Kroll, J. Brezillon, A. Fazzolari,
  R. Dwight, M. Widhalm

• HU Berlin: A. Griewank, J. Riehme

• Fastopt: R. Giering, Th. Kaminski

• TU Dresden: A. Walther, C. Moldenhauer

• Uni Trier: V. Schulz, S. Hazra
Content of lecture

- Why adjoint approaches?
- What is an adjoint approach?
- Continuous and discrete adjoint approaches / solvers
- Validation and Application in 2D and 3D, Euler and Navier-Stokes
- Algorithmic / Automated Differentiation (AD)
- Coupled aero-structure adjoint approach
- Validation and application in MDO context
- One shot approaches
Requirements on CFD

• high level of physical modeling
  – compressible flow
  – transonic flow
  – laminar - turbulent flow
  – high Reynolds numbers (60 million)
  – large flow regions with flow separation
  – steady / unsteady flows

• complex geometries
• short turn around time
Consequences

- solution of 3D compressible Reynolds averaged Navier-Stokes equations
- turbulence models based on transport equations (2 – 6 eqn)
- models for predicting laminar-turbulent transition
- flexible grid generation techniques with high level of automation (block structured grids, overset grids, unstructured/hybrid grids)
- link to CAD-systems
- efficient algorithms (multigrid, grid adaptation, parallel algorithms...)
- large scale computations ( ~ 10 - 25 million grid points)
- …
**Structured RANS solver FLOWer**
- block-structured grids
- moderate complex configurations
- fast algorithms (unsteady flows)
- design option
- adjoint option

**Unstructured RANS solver TAU**
- hybrid grids
- very complex configurations
- grid adaptation
- fully parallel software
- adjoint option
Physical model
- 3D compressible Navier-Stokes equations
- arbitrarily moving bodies
- steady and time accurate flows
- state-of-the-art turbulence models (RSM)

Grid strategy
- block-structured grids
- discontinuous block boundaries
- overset grids (Chimera)
- deforming grids

Numerical algorithms
- 2nd order finite volume discretization (cell centered & cell vertex option)
- central and upwind schemes
- multigrid
- implicit treatment of turbulence equations
- implicit schemes for time accurate flows
- preconditioning for low speed flow
- vectorization & parallelization
- adjoint solver
Reynolds-Averaged Navier-Stokes Solver TAU

Physical model
- 3D compressible Navier-Stokes equations
- arbitrarily moving bodies
- steady and time accurate flows
- state-of-the-art turbulence models

Grid strategy
- unstructured/hybrid grids
- semi-structured sublayers
- overset grids (Chimera)
- deforming grids
- grid adaptation (refinement, de-refinement)

Numerical algorithms
- 2nd order finite volume discretization based on dual grid approach
- central and upwind schemes
- multigrid based on agglomeration
- implicit schemes for time accurate flows
- preconditioning for low speed flow
- optimized for cash and vector processors
- MPI parallelization
Dual grid approach

- solver independent of cell types of primary grid
- efficient edge-based data structure
- agglomeration of dual cells for coarser meshes (multigrid)
Local Mesh Adaptation

- local grid refinement and de-refinement depending flow solution
- reduction of total number of grid points
- efficient simulation of complex flow phenomena

Overlapping grid technique

- efficient approach for simulation of complex configurations with movable control surfaces \((m\text{aneuvering aircraft})\)
- separate grids for movable surfaces
- parallel implementation
• $M_\infty = 0.85$, $Re = 32.5 \times 10^6$
• coupled CFD/structural analysis for wing deformation at $\alpha \approx 1.5^\circ$
• FLOWer, $k_\omega$ turbulence model, fully turbulent

3.5 million grid points
• $M_\infty = 0.85$, $Re = 32.5 \times 10^6$
• coupled CFD/structural analysis for wing deformation at $\alpha \approx 1.5^\circ$
• FLOWer, $k_\omega$ turbulence model, fully turbulent

Validation
HiReTT Wing/Body Configuration

3.5 million grid points

$C_L(C_{D,net})$, $C_L(C_D)$, $C_L(\alpha)$

wing deformation computed by RWTH Aachen

exp. ETW
def. pre-estimated
def. computed for $\alpha \approx 1.5^\circ$
Aerodynamic Shape Optimization

Requirements

- complex configurations
- compressible Navier-Stokes equations
  with accurate models for turbulence and transition
- validated and efficient CFD codes
- multi-point design, multi-objective optimization, MDO
- large number of design variables
- physical and geometrical constraints
- meshing & mesh deformation techniques ensuring grid quality
- efficient optimization algorithms
- automatic framework
- parameterization based on CAD model
Aerodynamic Shape Optimization

Requirements

- complex configurations
- compressible Navier-Stokes equations with accurate models for turbulence and transition
- validated and efficient CFD codes
- multi-point design, multi-objective optimization
- large number of design variables
- physical and geometrical constraints
- meshing & mesh deformation techniques
- efficient optimization algorithms
- automatic framework
- parameterization based on CAD model

⇒ Sensitivity based deterministic optimization strategies !!!
Aerodynamic Shape Optimization

Parametrized airfoil

Search direction

\[
\nabla I = -\left(\frac{\delta I}{\delta P_i}, \ldots\right)^T_{i=1,\ldots,n}
\]

control points/ control polygon
original curve
B-spline

Design space

Line search

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Governing Equations and Aerodynamic Coefficients

Compressible 2D Euler-Equations

\[ \frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0 \]

while

\[
\begin{pmatrix}
\rho \\
\rho u \\
\rho v \\
\rho E
\end{pmatrix}
, \quad
\begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho u v \\
\rho u H
\end{pmatrix}
, \quad
\begin{pmatrix}
\rho v \\
\rho u v \\
\rho v^2 + p \\
\rho v H
\end{pmatrix}
\]

Dimensionless pressure

\[ C_p = \frac{2(p - p_\infty)}{\gamma M_\infty^2 p_\infty} \]

Drag, lift, pitching moment coefficients

\[ C_D = \frac{1}{C_{ref}} \int_C C_p (n_x \cos \alpha + n_y \sin \alpha) dl \]
\[ C_L = \frac{1}{C_{ref}} \int_C C_p (n_x \cos \alpha - n_x \sin \alpha) dl \]
\[ C_m = \frac{1}{C_{ref}^2} \int_C C_p (n_y (x - x_m) - n_x (y - y_m)) dl \]

Pressure (ideal gas)

\[ p = (\gamma - 1) \rho (E - \frac{1}{2} \vec{v}^2) \]
Finite Differences

Variation of i-th design variable

\[ \delta C_D = \frac{2}{\gamma M_\infty^2 p_\infty C_{\text{ref}}} \int_C \delta P (n_x \cos \alpha + n_y \sin \alpha) dl \]

\[ + \frac{1}{C_{\text{ref}}} \int_C C_p (\delta n_x \cos \alpha + \delta n_y \sin \alpha) dl, \]

Metric sensitivities → pressure variation → aerodynamic sensitivity

i-th component of cost function's gradient

• Finite Differences

n design variables require n+1 flow calculations
Motivation of Adjoint Approach

High number of design variables

- Finite Differences  \( \rightarrow \) n design variables require \( n+1 \) flow calculations

- Adjoint Approach  \( \rightarrow \) n design variables require 1 flow and 1 adjoint flow calculation

  Independent of number of design variables

  High accuracy
Dual or Adjoint (Linear) Problem

Let be \( A \in \mathbb{R}^{n \times m} \), \( h \in \mathbb{R}^m \), \( \varphi \in \mathbb{R}^m \) and \( b \in \mathbb{R}^n \).

We define the primal linear problem:

\[
\text{evaluate } I = h^T \varphi , \quad \text{while } A \varphi = b. \tag{1}
\]

Furthermore, \( \psi \in \mathbb{R}^n \) fulfills:

\[
A^T \psi = h. \tag{3}
\]

Then eqs. (2) and (3) imply

\[
h^T \varphi = (A^T \psi)^T \varphi = (A^T \psi, \varphi) = (\psi, A \varphi) = \psi^T A \varphi = \psi^T b \quad \forall \varphi, \psi \tag{4}
\]

and we have the equivalent dual or adjoint linear problem:

\[
\text{evaluate } I = \psi^T b , \quad \text{while } A^T \psi = h. \tag{5}
\]

The vector \( \psi = (\psi_i)_{i \in \{1, \ldots, n\}} \) is called the vector of adjoint variables \( \psi_i \).
Continuous Adjoint

We define now the scalar product

\[(h, \varphi) := \int_{\Omega} h^T \varphi \, dx.\]  \hspace{1cm} (7)

Let \( \varphi \) be the solution of the PDE

\[L\varphi = b\]  \hspace{1cm} (8)

in the domain \( \Omega \), which fulfills the homogeneous boundary conditions on \( \partial\Omega \).

Then \( L^* \), the dual or adjoint operator of \( L \), is defined as:

\[L^*: (\psi, L\varphi) = (L^*\psi, \varphi) \quad \forall \varphi, \psi.\]  \hspace{1cm} (9)

Furthermore, \( \psi \), the vector(-field) of adjoint variables, solves the dual or adjoint PDE

\[L^*\psi = h\]  \hspace{1cm} (10)

in the domain \( \Omega \) and again fulfills the homogeneous boundary conditions on \( \partial\Omega \).

Then finally we have as before:

\[(h, \varphi) = (L^*\psi, \varphi) = (\psi, L\varphi) = (\psi, b).\]  \hspace{1cm} (11)
Examples of Adjoint Operators

Let’s take e.g. the convection-diffusion equation

\[ L \varphi \equiv \frac{d \varphi}{dx} - \epsilon \frac{d^2 \varphi}{d x^2}, \quad 0 < x < 1, \]  \hspace{1cm} (12)

with homogeneous boundary conditions \( \varphi(0) = \varphi(1) = 0 \).

Integration by parts yields \((\varphi, \psi) \in C^2\):

\[
(\psi, L \varphi) = \int_0^1 \psi \left( \frac{d \varphi}{dx} - \epsilon \frac{d^2 \varphi}{dx^2} \right) \, dx \\
= \int_0^1 \left( -\frac{d \psi}{dx} - \epsilon \frac{d^2 \psi}{dx^2} \right) \varphi \, dx + \left[ \psi \varphi - \epsilon \psi \frac{d \varphi}{dx} + \epsilon \varphi \frac{d \psi}{dx} \right]_0^1 \\
= \int_0^1 \left( -\frac{d \psi}{dx} - \epsilon \frac{d^2 \psi}{dx^2} \right) \varphi \, dx + \left[ -\epsilon \psi \frac{d \varphi}{dx} \right]_0^1. \hspace{1cm} (15)
\]
Examples of Adjoint Operators

For the adjoint convection-diffusion equation

\[ L^*\psi \equiv -\frac{d\psi}{dx} - \epsilon \frac{d^2\psi}{dx^2}, \]  

(16)

with homogeneous boundary conditions \( \psi(0) = \psi(1) = 0 \), the boundary term (15) vanishes and it holds (11):

\[ (h, \varphi) = (L^*\psi, \varphi) = (\psi, L\varphi) = (\psi, b). \]

Some examples:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Adjoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convection-Diffusion Eq.</td>
<td>( \frac{d\varphi}{dx} - \epsilon \frac{d^2\varphi}{dx^2} )</td>
</tr>
<tr>
<td>Wave Eq.</td>
<td>( \frac{d\varphi}{dt} - \frac{d^2\varphi}{dx^2} )</td>
</tr>
<tr>
<td>Convection Eq.</td>
<td>( \frac{d\varphi}{dt} + \frac{d\varphi}{dx} )</td>
</tr>
</tbody>
</table>
Different adjoint approaches

- **Continuous Adjoint**
  - optimize then discretize
  - hand coded adjoint solvers
  - time consuming in implementation
  - efficient in run and memory

- **Discrete Adjoint / Algorithmic Differentiation (AD)**
  - discretize then optimize
  - hand coding of adjoint solvers or ...
  - ... more or less automated generation
  - memory effort increases (way out e.g. check-pointing)
How to get the gradient using adjoint theory

Let the optimization problem be stated as

$$\min_{D} I(W, X, D),$$

and with the governing equations

$$R(W, X, D) = 0$$

with $W$ the flow variables, $X$ the mesh and $D$ the design variables.

The goal here is to determine the derivatives of $I$ with respect to $D$.

We define the Lagrangian which is identical to $I$ and its derivatives with respect to the design variables $D$

$$L = I + \Lambda^T R$$
The derivatives of $L$ with respect to the design variables $D$ are:

$$\frac{dL}{dD} = \frac{d}{dD}
\left( I(W, X, D) + \Lambda^T R(W, X, D) \right)$$
How to get the gradient using adjoint theory

The derivatives of $L$ with respect to the design variables $D$ are:

$$\frac{dL}{dD} = \frac{d}{dD} \left( I(W, X, D) + \Lambda^T R(W, X, D) \right)$$

$$= \left\{ \frac{\partial I}{\partial W} \frac{dW}{dD} + \frac{\partial I}{\partial X} \frac{dX}{dD} + \frac{\partial I}{\partial D} \right\} + \Lambda^T \left\{ \frac{\partial R}{\partial W} \frac{dW}{dD} + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} \right\}$$
The derivatives of $L$ with respect to the design variables $D$ are:

\[
\frac{dL}{dD} = \frac{d}{dD} \left( I(W, X, D) + \Lambda^TR(W, X, D) \right)
\]

\[
= \left\{ \frac{\partial I}{\partial W} \frac{dW}{dD} + \frac{\partial I}{\partial X} \frac{dX}{dD} + \frac{\partial I}{\partial D} \right\} + \Lambda^T \left\{ \frac{\partial R}{\partial W} \frac{dW}{dD} + \frac{\partial R}{\partial X} \frac{dX}{dD} + \frac{\partial R}{\partial D} \right\}
\]

\[
= \left\{ \frac{\partial I}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} \right\} \frac{dW}{dD} + \left\{ \frac{\partial I}{\partial X} + \Lambda^T \frac{\partial R}{\partial X} \right\} \frac{dX}{dD} + \left\{ \frac{\partial I}{\partial D} + \Lambda^T \frac{\partial R}{\partial D} \right\}
\]
How to get the gradient using adjoint theory

The derivatives of $L$ with respect to $D$ are:

$$\frac{dL}{dD} = \left\{ \frac{\partial I}{\partial X} + \Lambda^T \frac{\partial R}{\partial X} \right\} \frac{dX}{dD} + \left\{ \frac{\partial I}{\partial D} + \Lambda^T \frac{\partial R}{\partial D} \right\} + \left\{ \frac{\partial I}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} \right\} \frac{dW}{dD}$$

$$= 0$$

The expensive component can be canceled by solving the adjoint equation.

- Metric sensitivities relatively inexpensive with finite differences
- Partial variations according to the design variables relatively inexpensive
- Variations w. r. t. the flow variables expensive to evaluate
After solving the adjoint equation,

\[ \frac{\partial I}{\partial W} + \Lambda^T \frac{\partial R}{\partial W} = 0 \]

the derivatives of \( L \) with respect to \( D \) are evaluated according to

\[ \frac{dL}{dD} = \left\{ \frac{\partial I}{\partial X} + \Lambda^T \frac{\partial R}{\partial X} \right\} \frac{dX}{dD} + \left\{ \frac{\partial I}{\partial D} + \Lambda^T \frac{\partial R}{\partial D} \right\} \]
\[ D \quad \text{flow field domain} \]
\[ B \quad \text{far field} \]
\[ C \quad \text{wall} \]
\[ \partial D := B \cup C \quad \text{flow field boundary} \]
\[ \vec{S} := (S_x, S_y) \quad \text{normal vector } \perp \partial D \]
\[ \vec{n} := (n_x, n_y) \quad \text{normal unit vector } \perp \partial D \]
\[ \alpha \quad \text{angle of attack} \]
\[ C_D \quad \text{drag coefficient} \]
\[ C_L \quad \text{lift coefficient} \]
\[ \rho \quad \text{pressure} \]
\[ M \quad \text{Mach number} \]
\[ \rho_\infty \quad \ldots \text{at free stream} \]
\[ \gamma \quad \text{ratio of specific heats} \]
\[ S_{ref} \quad \text{area of airfoil} \]
\[ \frac{2(p - p_\infty)}{\gamma M^2 p_\infty} =: C_p \quad \text{pressure coefficient} \]
2D Euler Equations in body fitted coordinates

**Cartesian coordinates:**

\[
\frac{\partial w}{\partial t} + \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} = 0
\]

\[
w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u H \end{pmatrix}, \quad g = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v H \end{pmatrix}
\]

\[p = (\gamma - 1)\rho(E - \frac{1}{2}(u^2 + v^2)), \quad \rho H = \rho E + p\]

**Body fitted transformation:**

\[(x, y) \mapsto (\xi(x, y), \eta(x, y)),\]

\[J = \det \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{pmatrix}, \quad \begin{pmatrix} U \\ V \end{pmatrix} = \frac{1}{J} \begin{pmatrix} \frac{\partial \xi}{\partial x} & -\frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix}\]

**Body fitted coordinates:**

\[
\frac{\partial W}{\partial t} + \frac{\partial F}{\partial \xi} + \frac{\partial G}{\partial \eta} = 0
\]

\[W = J \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad F = J \begin{pmatrix} \rho u \\ \rho u^2 + \frac{\partial \xi}{\partial x} p \\ \rho v + \frac{\partial \xi}{\partial y} p \\ \rho U H \end{pmatrix}, \quad G = J \begin{pmatrix} \rho v \\ \rho v^2 + \frac{\partial \eta}{\partial x} p \\ \rho v + \frac{\partial \eta}{\partial y} p \\ \rho V H \end{pmatrix}\]
Derivation of the continuous adjoint Euler equations

In the case of steady state it holds for the perturbed geometry

\[
\frac{\partial}{\partial \xi} (F + \delta F) + \frac{\partial}{\partial \eta} (G + \delta G) = 0
\]

\[\Rightarrow \quad (1) \quad \frac{\partial}{\partial \xi} (\delta F) + \frac{\partial}{\partial \eta} (\delta G) = 0.\]

Furthermore

\[\delta F = \delta \left( J \frac{\partial \xi}{\partial x} \right) f + \delta \left( J \frac{\partial \xi}{\partial y} \right) g + J \frac{\partial \xi}{\partial x} \frac{\partial f}{\partial \omega} \delta \omega + J \frac{\partial \xi}{\partial y} \frac{\partial g}{\partial \omega} \delta \omega \]

and

\[\delta G = \delta \left( J \frac{\partial \eta}{\partial x} \right) f + \delta \left( J \frac{\partial \eta}{\partial y} \right) g + J \frac{\partial \eta}{\partial x} \frac{\partial f}{\partial \omega} \delta \omega + J \frac{\partial \eta}{\partial y} \frac{\partial g}{\partial \omega} \delta \omega.\]
Derivation of the continuous adjoint Euler equations

Together with (1) and the fundamental lemma of variational calculus it holds

\[ \int_D \psi^T \left( \frac{\partial}{\partial \xi} (\delta F) + \frac{\partial}{\partial \eta} (\delta G) \right) d\xi d\eta = 0 \]

for any Lagrangian multiplier \( \psi \).

If \( \psi \) is differentiable one obtains together with Greens formula

\[ - \int_D \left( \frac{\partial \psi^T}{\partial \xi} \delta F + \frac{\partial \psi^T}{\partial \eta} \delta G \right) d\xi d\eta + \int_D (n_1 \psi^T \delta F + n_2 \psi^T \delta G) d\xi - \int_C (n_1 \psi^T \delta F + n_2 \psi^T \delta G) d\xi = 0. \]
Derivation of the continuous adjoint Euler equations

Now the variation of the cost function can be expressed as

$$
\delta C_D = \frac{2}{\gamma M_\infty^2 p_\infty S_{ref}} \int_C \delta p (S_x \cos \alpha + S_y \sin \alpha) \, d\xi - \int_D \left( \frac{\partial \psi^T}{\partial \xi} \delta F + \frac{\partial \psi^T}{\partial \eta} \delta G \right) d\xi d\eta
$$

$$
+ \int_B (n_1 \psi^T \delta F + n_2 \psi^T \delta G) d\xi - \int_C \left( \frac{n_1 \psi^T \delta F + n_2 \psi^T \delta G}{\psi_0, n_1=0} \right) d\xi
$$

$$
+ \frac{1}{S_{ref}} \int_C C_p (\delta S_x \cos \alpha + \delta S_y \sin \alpha) \, d\xi.
$$

Along $C$ it holds $V = 0$ and yields

$$
G = J \begin{pmatrix}
0 \\
\frac{\delta \eta}{\partial \tau} p \\
\frac{\delta \eta}{\partial \eta} p \\
0
\end{pmatrix}, \quad \delta G = J \begin{pmatrix}
0 \\
\frac{\delta \eta}{\partial \tau} \delta p \\
\frac{\delta \eta}{\partial \eta} \delta p \\
0
\end{pmatrix} + p \begin{pmatrix}
0 \\
\delta \left( J \frac{\delta \eta}{\partial \tau} \right) \\
\delta \left( J \frac{\delta \eta}{\partial \eta} \right) \\
0
\end{pmatrix}.
$$
Derivation of the continuous adjoint Euler equations

Together with (2) and (3) one obtains

\[
\delta C_D = \frac{2}{\gamma M^2_{\infty} P_{\infty} S_{ref}} \int_C \delta p (S_x \cos \alpha + S_y \sin \alpha) \, d\xi \\
- \int_D \frac{\partial \psi^T}{\partial \eta} \left( \delta \left( J \frac{\partial \xi}{\partial x} \right) f + \delta \left( J \frac{\partial \xi}{\partial y} \right) g + J \frac{\partial \xi}{\partial x} \frac{\partial f}{\partial w} + J \frac{\partial \xi}{\partial y} \frac{\partial g}{\partial w} \right) + \frac{\partial \psi^T}{\partial \eta} \left( \delta \left( J \frac{\partial \eta}{\partial x} \right) f + \delta \left( J \frac{\partial \eta}{\partial y} \right) g + J \frac{\partial \eta}{\partial x} \frac{\partial f}{\partial w} + J \frac{\partial \eta}{\partial y} \frac{\partial g}{\partial w} \right) d\xi \, d\eta \\
- \int_C \psi_2 \left( J \frac{\partial \eta}{\partial x} \delta p + p \delta \left( J \frac{\partial \eta}{\partial x} \right) \right) + \psi_3 \left( J \frac{\partial \eta}{\partial y} \delta p + p \delta \left( J \frac{\partial \eta}{\partial y} \right) \right) d\xi \\
+ \int_B n_1 \psi^T \delta F + n_2 \psi^T \delta G d\xi + \frac{1}{S_{ref}} \int_C C_p (\delta S_x \cos \alpha + \delta S_y \sin \alpha) \, d\xi.
\]

If the adjoint Euler equations

\[
\frac{\partial \psi^T}{\partial \xi} \left( J \frac{\partial \xi}{\partial x} \frac{\partial f}{\partial w} + J \frac{\partial \xi}{\partial y} \frac{\partial g}{\partial w} \right) + \frac{\partial \psi^T}{\partial \eta} \left( J \frac{\partial \eta}{\partial x} \frac{\partial f}{\partial w} + J \frac{\partial \eta}{\partial y} \frac{\partial g}{\partial w} \right) = 0 \iff \left( \frac{\partial f}{\partial w} \right)^T \frac{\partial \psi}{\partial x} + \left( \frac{\partial g}{\partial w} \right)^T \frac{\partial \psi}{\partial y} = 0
\]
Derivation of the continuous adjoint Euler equations

... are fulfilled in the domain $D$ with the boundary conditions

$$
\frac{2}{\gamma M_{\infty}^2 p_{\infty} S_{\text{ref}}} (S_x \cos \alpha + S_y \sin \alpha) = \frac{-S_x \psi_2 - S_y \psi_3}{-\frac{\partial \psi_2}{\partial x} + \frac{\partial \psi_3}{\partial y} = \frac{\partial \psi_2}{\partial x} + \frac{\partial \psi_3}{\partial y}}
$$

on the airfoil $C$ (dependent on the cost function!) and

$$
\delta \left( J \frac{\partial \xi}{\partial x} \right), \ldots, \delta \left( J \frac{\partial \eta}{\partial y} \right) \to 0 \quad \psi^T J \frac{\partial \xi}{\partial x} \frac{\partial f}{\partial w} \delta w = 0, \ldots, \psi^T J \frac{\partial \eta}{\partial y} \frac{\partial g}{\partial w} \delta w = 0
$$

at the far field $B$ one can simplify $\delta C_D$ to

$$
\delta C_D = -\int_D \frac{\partial \psi^T}{\partial \xi} \left( \delta \left( \frac{\partial y}{\partial \eta} \right) f - \delta \left( \frac{\partial x}{\partial \eta} \right) g \right) \frac{\partial \eta^T}{\partial \eta} \left( -\delta \left( \frac{\partial y}{\partial \xi} \right) f + \delta \left( \frac{\partial x}{\partial \xi} \right) g \right) d\xi d\eta
$$

$$
-\int_C p (\delta S_x \psi_2 + \delta S_y \psi_3) d\xi + \frac{1}{S_{\text{ref}}} \int_C C_p (\delta S_x \cos \alpha + \delta S_y \sin \alpha) d\xi.
$$
Continuous Adjoint Approach

**Adjoint Euler-Equations:**

\[
- \frac{\partial \psi}{\partial t} - \left( \frac{\partial f}{\partial w} \right)^T \frac{\partial \psi}{\partial x} - \left( \frac{\partial g}{\partial w} \right)^T \frac{\partial \psi}{\partial y} = 0
\]

\(\Psi\): Vector of adjoint variables

**Boundary conditions:**

**Wall:** \(n_x \psi_2 + n_y \psi_3 = -d(I)\)

**Farfield:** \(\delta x_\xi, ..., \delta y_\eta = 0, \delta w = 0\)

**Adjoint volume formulation of cost function’s gradient:**

\[
\delta I = -\int_C p \left( -\psi_2 \delta y_\xi + \psi_3 \delta x_\xi \right) dl + K(I)
\]

\[-\int_D \psi_T^{\xi} \left( \delta y_\eta f - \delta x_\eta g \right) + \psi_T^{\eta} \left( -\delta y_\xi f + \delta x_\xi g \right) dA\]
Continuous Adjoint Approach

\[ d(C_D) = \frac{2}{\gamma M_c^2 p_c C_{ref}} (n_x \cos \alpha + n_y \sin \alpha) \]

Drag

\[ K(C_D) = \frac{1}{C_{ref}} \int_C C_p (\delta n_x \cos \alpha + \delta n_y \sin \alpha) \, dl \]

\[ d(C_L) = \frac{2}{\gamma M_c^2 p_c C_{ref}} (n_y \cos \alpha - n_x \sin \alpha) \]

Lift

\[ K(C_L) = \frac{1}{C_{ref}} \int_C C_p (\delta n_y \cos \alpha - \delta n_x \sin \alpha) \, dl \]

\[ d(C_m) = \frac{2}{\gamma M_c^2 p_c C_{ref}^2} (n_y (x - x_m) - n_x (y - y_m)) \]

Pitching moment

\[ K(C_m) = \frac{1}{C_{ref}^2} \int_C C_p \delta(n_y (x - x_m) - n_x (y - y_m)) \, dl \]
Continuous adjoint

- Euler implemented in FLOWer & TAU
- surface formulation for gradient evaluation
- one shot method (FLOWer)
- coupled aero-structure adjoint (FLOWer)
- Navier-Stokes (frozen $\mu$) implemented in FLOWer, robustness problems

Discrete adjoint

- implemented in TAU
- Euler & RANS with several turbulence models
- currently high memory requirements
- experience with automatic differentiation (FLOWer and TAUijk)
Continuous adjoint
- Euler implemented in FLOWer & TAU
- surface formulation for gradient evaluation
- one shot method (FLOWer)
- coupled aero-structure adjoint (FLOWer)
- Navier-Stokes (frozen $\mu$) implemented in FLOWer, robustness problems

Discrete adjoint
- implemented in TAU
- Euler & RANS with several turbulence models
- currently high memory requirements
- experience with automatic differentiation (FLOWer and TAUijk)

comparison of gradients (3-airfoil, viscous)
Continuous adjoint Euler solver TAU

Runge-Kutta versus LUSGS

- **flow solution**
  - Rae2822
  - $M = 0.734$
  - $\alpha = 2.0^\circ$
- **drag optimization**
  - adjoint solution

N. C
Intro to Optimization and MDO, VKI, March 6-10, 2006
Continuous adjoint solver FLOWer

Adjoint solver on block-structured grids

- continuous adjoint approach
- implemented in FLOWer
- cost functions: lift, drag & moment and combinations
- adjoint solver based on multigrid
- Euler & Navier-Stokes (frozen $\mu$)

![Convergence history, FLOWer](image)

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Validation of continuous adjoint solver in FLOWer

Adjoint approach vs. finite differences' gradient

finite differences:
51 calls of FLOWer MAIN

adjoint approach:
1 call of FLOWer MAIN
50 design variables (B-spline)

RAE2822
$M_{\infty} = 0.73, \alpha = 2.0^\circ$

Validation

Adjoint approach compared to finite differences for lift, drag, and moment:
- Lift:
  - Adjoint: 1 call of FLOWer MAIN
  - Finite Differences: 51 calls of FLOWer MAIN
- Drag:
  - Adjoint: 3 calls of FLOWer ADJOINT
  - Finite Differences: 51 calls of FLOWer MAIN
- Moment:
  - Adjoint: 3 calls of FLOWer ADJOINT
  - Finite Differences: 51 calls of FLOWer MAIN

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Validation of adjoint gradient based optimization

Objective function
- Drag reduction for RAE 2822 airfoil
- \( M_\infty = 0.73, \alpha = 2.00^\circ \)

Constraints
- Constant thickness

Approach
- FLOWer Euler Adjoint
- Deformation of camberline (20 Hicks-Henne functions)

Optimizer
- Steepest Descent
- Conjugate Gradient
- Quasi Newton Trust Region

Drag reduction by constant thickness
RAE 2822 - \( M_\infty = 0.72, \alpha = 2^\circ \)
Validation of adjoint gradient based optimization

Objective function
- Drag reduction for RAE 2822 airfoil
- $M_\infty = 0.73$, $\alpha = 2.00^\circ$

Constraints
- Constant thickness

Approach
- FLOWer Euler Adjoint
- Deformation of camberline (20 Hicks-Henne functions)

Optimizer
- Steepest Descent
- Conjugate Gradient
- Quasi Newton Trust Region
Content of lecture

Adjoint Approaches in Aerodynamic Shape Optimization and MDO Context II

- Why adjoint approaches?
- What is an adjoint approach?
- Continuous and discrete adjoint approaches / solvers
- Validation and Application in 2D and 3D, Euler and Navier-Stokes
- Algorithmic / Automated Differentiation (AD)
- Coupled aero-structure adjoint approach
- Validation and application in MDO context
- One shot approaches
Treatment of Constraints

Orthogonal projection

\[ \nabla C_L - \nabla C_D \]

In direction \( r^{(k)} \) the drag is reduced while the lift is held constant

\[ \frac{dC_L(X^{(k)})}{dr^{(k)}} = (\nabla_{X^{(k)}} C_L)^T \frac{r^{(k)}}{\|r^{(k)}\|} = 0 \]

\[ C_L(r) \approx C_L(X^{(k)}) \]

Schmidt - orthogonalization

\[ \{a_1, a_2, a_3\} = \{\nabla C_L, \nabla C_m, -\nabla C_D\} \]

\[ \{b_1, b_2, b_3\} : \]

\[ b_1 = a_1 , \]

\[ b_{l+1} = a_{l+1} - \sum_{i=1}^{l} \frac{b_i^T a_{l+1} b_i}{\|b_i\|^2} \quad l = 1,2. \]

it holds

\[ a_i^T b_3 = 0, \quad i = 1,2 \]

\[ b_3 = -\nabla C_D + \sum_{i=1}^{2} \frac{b_i^T \nabla C_D b_i}{\|b_i\|^2} \]

In direction \( b_3 \) the drag is reduced while the lift and pitching moment are held constant
Treatment of Constraints

Orthogonal projection

\[ \nabla C_L - \nabla C_D \]

Schmidt - orthogonalization

\[ \{a_1, a_2, a_3\} = \{\nabla C_L, \nabla C_m, -\nabla C_D\} \]

\[ \{b_1, b_2, b_3\} : \]

In direction \( r^{(k)} \) the drag is reduced while the lift is held constant

\[ \frac{dC_L(X^{(k)})}{dr^{(k)}} = (\nabla X^{(k)} C_L)^T \frac{r^{(k)}}{\|r^{(k)}\|} = 0 \]

\[ C_L(r) \approx C_L(X^{(k)}) \]

A lot of other strategies and commercial packages are available !!!

it holds

\[ a_i^T b_3 = 0, \quad i = 1,2 \]

\[ b_3 = -\nabla C_D + \sum_{i=1}^{2} \frac{b_i^T \nabla C_D}{\|b_i\|^2} b_i \]

In direction \( b_3 \) the drag is reduced while the lift and pitching moment are held constant
Multi-constraint airfoil optimization RAE2822

Objective function
- Drag reduction for RAE 2822 airfoil
- $M_\infty = 0.73$, $\alpha = 2.0^\circ$

Constraints
- Lift, pitching moment and angle of attack held constant
- Constant thickness

Approach
- FLOWer Euler Adjoint
- Constraints handled by feasible direction
- Deformation of camberline

Graph:
- $\Delta C_D > 60 \%$
- $51.9$ drag counts
Multi-constraint airfoil optimization RAE2822

Objective function

- Drag reduction for RAE 2822 airfoil
- $M_{\infty} = 0.73, \alpha = 2.0^\circ$

Constraints

- Lift, pitching moment and angle of attack held constant
- Constant thickness

Approach

- FLOWer Euler Adjoint
- Constraints handled by feasible direction
- Deformation of camberline

![Surface pressure distribution](image)
Multipoint airfoil optimization RAE2822

Objective function

- Reduction of drag in 2 design points

Design points

- 1 : \(M_\infty = 0.734, \ CL = 0.80, \ \alpha = 2.8^\circ, \ Re=6.5 \times 10^6, \ x_{trans}=3\%, \ W_1=2\)
- 2 : \(M_\infty = 0.754, \ CL = 0.74, \ \alpha = 2.8^\circ, \ Re=6.2 \times 10^6, \ x_{trans}=3\%, \ W_2=1\)

Constraints

- No lift decrease, no change in angle of incidence
- Variation in pitching moment less than 2% in each point
- Maximal thickness constant and at 5% chord more than 96% of initial
- Leading edge radius more than 90% of initial
- Trailing edge angle more than 80% of initial

\[
I = \sum_{i=1}^{2} W_i C_d(\alpha_i, M_i)
\]
Parameterization
- 20 design variables changing camberline, Hicks-Henne functions

Optimization strategy
- Constrained SQP
- Navier-Stokes solver FLOWer, Baldwin/Lomax turbulence model
- Gradients provided by FLOWer Adjoint, based on Euler equations

Results

<table>
<thead>
<tr>
<th>Pt</th>
<th>$\alpha$</th>
<th>$M_i$</th>
<th>$C_l^t$</th>
<th>$C_d^t$ ($10^{-4}$)</th>
<th>$C_l$</th>
<th>$C_d^t$ ($10^{-4}$)</th>
<th>$\Delta C_d/C_d^t$</th>
<th>$\Delta C_l/C_l^t$</th>
<th>$\Delta C_m/C_m^t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.8</td>
<td>0.734</td>
<td>0.811</td>
<td>197.1</td>
<td>0.811</td>
<td>135.5</td>
<td>-31.2%</td>
<td>0%</td>
<td>+1.6%</td>
</tr>
<tr>
<td>2</td>
<td>2.8</td>
<td>0.754</td>
<td>0.806</td>
<td>300.8</td>
<td>0.828</td>
<td>215.0</td>
<td>-27.4%</td>
<td>+2.7%</td>
<td>+2.0%</td>
</tr>
</tbody>
</table>
Multipoint airfoil optimization RAE2822

1. design point

$M_\infty = 0.734$, $\alpha = 2.8^\circ$

shape geometry

2. design point

$M_\infty = 0.754$, $\alpha = 2.8^\circ$
Adjoint gradient formulations

Volume formulation:

\[ \delta I = - \int_{C} p (-\psi_2 \delta y_\xi + \psi_3 \delta x_\xi ) \, dl + K(I) \]
\[ - \int_{D} \psi_\xi^T (\delta y_\eta f - \delta x_\eta g) + \psi_\eta^T (-\delta y_\xi f + \delta x_\xi g) \, dA \]

High accuracy but unpractical for 3D multi-block!

Way out:

Surface formulations:

I.
\[ \delta I = - \int_{C} w_H^T \psi (\delta n_x u + \delta n_y v) \, dl + K(I) \]
\[ w_H^T = (\rho, \rho u, \rho v, \rho H) \]

II.
\[ \delta I = \int_{C} \text{div}(\vec{k}(I) + (w_H^T \psi) \vec{v}) \cdot (n_x \delta x + n_y \delta y) \, dl \]

others …

\[ \text{e.g. } \vec{k}^T (C_D) = C_p / C_{\text{ref}} (\cos \alpha, \sin \alpha) \]
RAE2822
\(M_\infty = 0.73, \, \alpha = 2.0^\circ\)
50 design variables
(B-spline)

Adjoint gradient formulation: Volume formulation vs. surface formulation (I.)

- \(\text{drag}\)
- \(\text{lift}\)
- \(\text{moment}\)
Objective function
  ▶ drag reduction by constant lift

Design point
  ▶ Mach number = 2.0
  ▶ lift coefficient = 0.12

Constraints
  ▶ fuselage incidence
  ▶ minimum fuselage radius
  ▶ wing planform unchanged
  ▶ minimum wing thickness distribution in spanwise direction
Optimization of SCT Configuration

Approach

- FLOWer code in Euler mode with target lift option
- Lift kept constant by adjusting angle of attack
- FLOWer code in Euler adjoint mode
- Adjoint gradient formulation: Surface formulation (II.)
- Structured mono-block grid (MegaCads), 230,000 grid points

Optimization strategy

- Quasi-Newton Method (BFGS algorithm)
Optimization of SCT Configuration

Design variables

- fuselage: 10 parameters
- twist deformation: 10 parameters
- camberline (8 sections): 32 parameters
- thickness (8 sections): 32 parameters
- angle of attack: 1 parameter

85 parameters

Fuselage

10 sections controlled by Bezier nodes
Optimization of SCT Configuration

Design variables

• fuselage: 10 parameters
• twist deformation: 10 parameters
• camberline (8 sections): 32 parameters
• thickness (8 sections): 32 parameters
• angle of attack: 1 parameter

85 parameters

Camberline

Deformation in 8 sections

Deformation in 8 sections
Design variables

- fuselage: 10 parameters
- twist deformation: 10 parameters
- camberline (8 sections): 32 parameters
- thickness (8 sections): 32 parameters
- angle of attack: 1 parameter

Thickness and camberline
Optimization of SCT Configuration

Optimized geometry

Baseline geometry

14.6 Drag Counts

11 times faster than classical approach

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Optimization of SCT Configuration

- Optimized geometry
- Baseline geometry

14.6 Drag Counts

11 times faster than classical approach

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Optimization of SCT Configuration

Radius of the fuselage in freestream direction

and Area Rule

Body radius [m]

Area [m²]

Freestream direction [m]
Wing section and pressure distribution

\( \eta = 0.24 \)

\( \eta = 0.49 \)

\( \eta = 0.92 \)

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Validation of Discrete Adjoint Solver in TAU

viscous flow around RAE2822 airfoil, M=0.73, \( \alpha=2.8^0 \), Re=6.5x10^6
Shape Optimization Based on Discrete Adjoint

Objective

- drag minimization for RAE 2822 airfoil at constant lift, pitching moment and AoA
- projected steepest descent strategy
- flow solver: viscous TAU-Code, SA model
- adjoint solver: viscous discrete TAU adjoint

\[ M_\infty = 0.73, \, \alpha = 2.80^\circ, \, Re = 6.5 \times 10^6 \]

20 design parameters
CD reduction at constant CL with varying angle-of-attack.
Takeoff configuration, Re=14.7x10^6, Ma=0.1715, RANS+SAE
Parameterized is only the “hidden” nose of the flap ~10 design vars.
Exact adjoint gradients with Conjugate Gradient optimization.
Drag reduction of 9 counts – lift unchanged.
Discrete Adjoint Solver

Advantage
- exact discrete adjoint in TAU for most commonly used models and discretizations
- solution via Krylov method requires 5% - 10% of time needed for flow solution

Problem
- memory requirement for large scale application
  - efficient storage strategy (recalculation of terms)

Approximations of discrete adjoint
- 1st order discretization (FOA)
- assumption of constant coefficients in the JST scheme (CCA)
- gradients based on Euler solution
- adjoint solution based on thin layer viscous fluxes
- assumption of constant eddy viscosity

<table>
<thead>
<tr>
<th></th>
<th>TAU Main</th>
<th>+ Jacobian storage</th>
<th>+ linear sol. storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory (bytes)</td>
<td>25M</td>
<td>165M</td>
<td>290M</td>
</tr>
<tr>
<td>Factor increase</td>
<td>x1.0</td>
<td>x6.6</td>
<td>x11.6</td>
</tr>
<tr>
<td>points in 1GB</td>
<td>2x106</td>
<td>300x103</td>
<td>170x103</td>
</tr>
</tbody>
</table>
Discrete Adjoint Solver

Advantage
- exact discrete adjoint in TAU for most commonly used models and discretizations
- solution via Krylov method requires 5% - 10% of time needed for flow solution

Problem
- memory requirement for large scale application
  - efficient storage strategy (recalculation of terms)

Approximations of discrete adjoint
- 1st order discretization (FOA)
- assumption of constant coefficients in the JST scheme (CCA)
- gradients based on Euler solution
- adjoint solution based on thin layer viscous fluxes
- assumption of constant eddy viscosity

<table>
<thead>
<tr>
<th></th>
<th>TAU Main</th>
<th>+ Jacobian storage</th>
<th>+ linear sol. storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory (bytes)</td>
<td>25M</td>
<td>165M</td>
<td>290M</td>
</tr>
</tbody>
</table>
Algorithmic Differentiation (AD)

Work in progress and results

• ADFLOWer generated with TAF (3D Navier-Stokes, k-w), first verifications and validation

• Adjoint version of TAUij (2D Euler) + mesh deformation and parameterization with ADOL-C, validated versus finite differences and first applications

• First and second derivatives of a “FLOWer-Derivate” (2D Euler) + mesh deformation and parameterization generated with TAPENADE, used for One Shot (Piggy Back)
www.autodiff.org

AD - Tools for Fortran and C

- ADOL-C, REVOLVE: C, C++, Open Source
- ADIFOR 2.0 / 3.0: Fortran 77/90/95, Licensed, Closed Source
- Tapenade: Fortran 77/90/95, (some) C, free, Closed Source
- TAF / TAC (FastOpt GbR): Fortran 77/90/95, (some) C, commercial, maybe free for educational
- NAGWare Fortran 95, NAG Ltd., Oxford, UK: AD-enabled version in beta status, not available for the public
- OpenAd: Fortran 77/90/95, (some) C, Open Source

Other tools

for Fortran, C, C++    for Matlab    for ADA    for ...
Main Properties of Automatic Differentiation:

- No Truncation Errors!!!!
- Chain Rule applied to Numbers
- Applicability to "Arbitrary Programs".

A priori bounded and/or adjustable costs:
- Total Operations Count
- Maximal Memory Requirement
- Total Memory Traffic

always relative to original function.
Simple Example

$$y = \left[ \sin\left(\frac{x_1}{x_2}\right) + \frac{x_1}{x_2} - \exp(x_2) \right] \times \left[ \frac{x_1}{x_2} - \exp(x_2) \right]$$

Evaluation of Simple Example:

<table>
<thead>
<tr>
<th>$v_{-1}$</th>
<th>$x_1$</th>
<th>$1.5000$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_0$</td>
<td>$x_2$</td>
<td>$0.5000$</td>
</tr>
<tr>
<td>$v_1$</td>
<td>$v_{-1}/v_0$</td>
<td>$1.5000/0.5000$</td>
</tr>
<tr>
<td>$v_2$</td>
<td>$\sin(v_1)$</td>
<td>$\sin(3.0000)$</td>
</tr>
<tr>
<td>$v_3$</td>
<td>$\exp(v_0)$</td>
<td>$\exp(0.5000)$</td>
</tr>
<tr>
<td>$v_4$</td>
<td>$v_1 - v_3$</td>
<td>$3.0000 - 1.6487$</td>
</tr>
<tr>
<td>$v_5$</td>
<td>$v_2 + v_4$</td>
<td>$0.1411 + 1.3513$</td>
</tr>
<tr>
<td>$v_6$</td>
<td>$v_5 \times v_4$</td>
<td>$1.4924 \times 1.3513$</td>
</tr>
<tr>
<td>$y$</td>
<td>$v_6$</td>
<td>$2.0167$</td>
</tr>
</tbody>
</table>
Geometric Interpretation – Forward Mode

\[ F \ldots \text{Original program } F \]
\[ \dot{x} \ldots \text{Tangent direction for input } x \]
\[ \dot{F} \ldots \text{Tangent version of } F \text{ (generated by Forward Mode AD)} \]
\[ \dot{y} \ldots \text{Tangent of output } y: \]

\[ \dot{y} = \dot{F}(x, \dot{x}) = F'(x) \cdot \dot{x} \]
### Forward Derived Evaluation Trace of Simple Example

<table>
<thead>
<tr>
<th>Expression</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_0 ) = x_2</td>
<td>0.5000</td>
</tr>
<tr>
<td>( v_0 ) = \dot{x}_2</td>
<td>0.0000</td>
</tr>
<tr>
<td>( v_1 ) = ( \frac{v_0}{v_0} )</td>
<td>3.0000</td>
</tr>
<tr>
<td>( v_1 ) = ( \frac{v_0 - v_1 + v_0}{v_0} )</td>
<td>2.0000</td>
</tr>
<tr>
<td>( v_2 ) = \sin(v_1)</td>
<td>0.1411</td>
</tr>
<tr>
<td>( v_2 ) = \cos(v_1) \cdot \dot{v}_1</td>
<td>-1.9800</td>
</tr>
<tr>
<td>( v_3 ) = \exp(v_0)</td>
<td>1.6487</td>
</tr>
<tr>
<td>( v_3 ) = v_3 \cdot \dot{v}_0</td>
<td>0.0000</td>
</tr>
<tr>
<td>( v_4 ) = v_1 - v_3</td>
<td>1.3513</td>
</tr>
<tr>
<td>( v_4 ) = \dot{v}_1 - \dot{v}_3</td>
<td>2.0000</td>
</tr>
<tr>
<td>( v_5 ) = v_2 + v_4</td>
<td>1.4924</td>
</tr>
<tr>
<td>( v_5 ) = \dot{v}_2 + \dot{v}_4</td>
<td>0.0200</td>
</tr>
<tr>
<td>( v_6 ) = v_5 \cdot v_4</td>
<td>2.0167</td>
</tr>
<tr>
<td>( v_6 ) = v_5 \cdot v_4 + v_5 \cdot \dot{v}_4</td>
<td>3.0118</td>
</tr>
<tr>
<td>( y ) = v_6</td>
<td>2.0100</td>
</tr>
<tr>
<td>( \dot{y} ) = \dot{v}_6</td>
<td>3.0110</td>
</tr>
</tbody>
</table>
Geometric Interpretation – Reverse Mode

\[ F \] \quad \text{Original program } F

\[ \bar{y} \] \quad \text{Adjoint direction for output } y

\[ \bar{F} \] \quad \text{Adjoint version of } F \text{ (generated by Reverse Mode AD)}

\[ \bar{x} \] \quad \text{Adjoint of input } x:\n
\[ \bar{x} = \bar{F}(x, \bar{y}) = \bar{y} \cdot F'(x) \]

\[ \bar{v}_i \] \quad \text{Adjoint variable } \frac{dy}{dv_i}
\[ \begin{align*}
\nu_{-1} &= x_1 = 1.5000 \\
\nu_0 &= x_2 = 0.5000 \\
\nu_1 &= \nu_{-1} / \nu_0 = 1.5000 / 0.5000 = 3.0000 \\
\nu_2 &= \sin(\nu_1) = \sin(3.0000) = 0.1411 \\
\nu_3 &= \exp(\nu_0) = \exp(0.5000) = 1.6487 \\
\nu_4 &= \nu_1 - \nu_3 = 3.0000 - 1.6487 = 1.3513 \\
\nu_5 &= \nu_2 + \nu_4 = 0.1411 + 1.3513 = 1.4924 \\
\nu_6 &= \nu_5 \times \nu_4 = 1.4924 \times 1.3513 = 2.0167 \\
y &= \nu_6 = 2.0167 \\
\bar{\nu}_6 &= y = 1.0000 \\
\bar{\nu}_5 &= \bar{\nu}_6 \times \nu_4 = 1.0000 \times 1.3513 = 1.3513 \\
\bar{\nu}_4 &= \bar{\nu}_5 + \nu_5 = 1.0000 \times 1.4924 = 1.4924 \\
\bar{\nu}_4 &= \bar{\nu}_4 + \bar{\nu}_5 = 1.4924 + 1.3513 = 2.8437 \\
\bar{\nu}_2 &= \bar{\nu}_3 + \bar{\nu}_5 = 1.3513 \\
\bar{\nu}_3 &= -\nu_4 = -2.8437 \\
\bar{\nu}_1 &= \bar{\nu}_4 = 2.8437 \\
\bar{\nu}_0 &= \bar{\nu}_3 \times \nu_3 = -2.8437 \times 1.6487 = -4.6884 \\
\bar{\nu}_1 &= \bar{\nu}_1 + \bar{\nu}_2 \times \cos(\nu_1) = 2.8437 + 1.3513 \times (-0.9900) = 1.5059 \\
\bar{\nu}_0 &= \bar{\nu}_0 - \bar{\nu}_1 \times \nu_1 / \nu_0 = -4.6884 - 1.5059 \times 3.0000 / 0.5000 = -13.7239 \\
\bar{\nu}_{-1} &= \nu_1 / \nu_0 = 1.5059 / 0.5000 = 3.0118 \\
\bar{x}_2 &= \nu_0 = -13.7239 \\
\bar{x}_1 &= \nu_{-1} = 3.0118
\end{align*} \]
Test configuration
- 2d NACA12
- k-omega (Wilcox) turbulence model
- cell-centred metric
- 2 time steps on fine grid
- target sensitivity: $\frac{d \text{lift}}{d \text{alpha}}$

Steps
- Modifications of FLOWer code (TAF Directives, slight recoding, etc...)
- tangent-linear code (verification + useful per se small dimensional design problems)
- adjoint code
- efficient adjoint code

Major challenge
- memory management (all variables in one big field 'variab') complicates detailed analysis and handling of deallocation
### TAF CPUs

<table>
<thead>
<tr>
<th>Case</th>
<th>Code lines</th>
<th>solve rel CPU</th>
<th>solve memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominal</td>
<td>166000</td>
<td>1.0</td>
<td>57</td>
</tr>
<tr>
<td>tangent</td>
<td>293</td>
<td>3.3</td>
<td>75</td>
</tr>
<tr>
<td>adjoint</td>
<td>253</td>
<td>6.3</td>
<td>489</td>
</tr>
</tbody>
</table>

**Accuracy of Sensitivity**

(Adjoint - Finite Difference Approximation) in Test Configuration

- Usually better for larger configurations

Ma = 0.734

α = 2.8°

Re = 6x10^6

kw turbulence model

---

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Demonstrates convergence of discrete sensitivities including turbulence (tangent linear model)

- Same sensitivity for Euler adjoint
- Same sensitivity for Navier-Stokes adjoint once it runs for 2000 time steps

\[
\begin{align*}
\text{Ma} &= 0.734 \\
\alpha &= 2.8^\circ \\
\text{Re} &= 6 \times 10^6 \\
\text{kw turbulence model}
\end{align*}
\]
• Demonstrates convergence of discrete sensitivities including turbulence (tangent linear model)

• Same sensitivity for Euler adjoint

• Same sensitivity for Navier-Stokes adjoint once it runs for 2000 time steps

Ma = 0.734
α = 2.8°
Re = 6x10^6
kw turbulence model
Differentiate entire design chain

- Adjoint version of entire design chain by ADOL-C:
  TAUij (2D Euler) + mesh deformation + parameterization
- Validated versus finite differences

Design vector \( P \) → defgeo \( x_{new} \) → difgeo \( dx \) → meshdefo \( m \) → TAUij → \( C_D \)

Surface grid (static) → grid (static)

\[
\frac{dC_D}{dP} = \frac{\partial C_D}{\partial m} \cdot \frac{\partial m}{\partial (dx)} \cdot \frac{\partial (dx)}{\partial x_{new}} \cdot \frac{\partial x_{new}}{\partial P}
\]

And

\[
\frac{\partial (dx)}{\partial x_{new}} = \frac{\partial (x_{new} - x_{old})}{\partial x_{new}} = Id
\]

TAUij_AD → meshdefo_AD → defgeo_AD

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
• RAE2822
  \( Ma = 0.73 \)
  \( \alpha = 2.0^\circ \)
  (mesh 161x33)

• Design variables:
  20 Hicks-Henne for camberline deformation

• Run time
  - primal: 2 minutes
  - adjoint: 16 minutes

• Run time memory
  - primal: 8 MB
  - adjoint: 45 MB

Differentiate entire design chain

Validation
Differentiate entire design chain

Application

RAE2822
Ma = 0.73
α = 2.0°
(mesh 161x33)
Different adjoint approaches

• Continuous Adjoint
  - optimize then discretize
  - hand coded adjoint solvers
  - time consuming in implementation
  - efficient in run and memory

• Discrete Adjoint / Algorithmic Differentiation (AD)
  - discretize then optimize
  - hand coding of adjoint solvers or …
  - … more or less automated generation
  - memory effort increases (way out e.g. check-pointing)

• Hybrid Adjoint
  - use source to source AD tools
  - optimize differentiated code
  - merge “continuous and discrete” routines
Coupled Aero-Structure Adjoint

Motivation

Wing deflection up to 7% of wing span!

Deflected aerodynamic optimal shape can be worse than the initial ...

Boeing 737-800 at ground and in cruise (Ma = 0.76)
Coupled Aero-Structure Adjoint

**AMP wing**

15 design variables  
(shape bumping functions based on Bernstein polynomials)

Ma = 0.78  
alpha = 2.83

Drag reduction by constant lift

Taking into account static deformation

**NASTRAN**  
shell/beam model  
126 nodes

**FLOWer MAIN/ADJOINT**  
15 design variables  
Ma = 0.78  
alpha = 2.83  
(300,000 cells)
Coupled Aero-Structure Adjoint

Conventional Gradient:

\[
\frac{dC_D}{dP} = \frac{\partial C_D}{\partial P} + \frac{\partial C_D}{\partial w} \frac{\partial w}{\partial P} + \frac{\partial C_D}{\partial d} \frac{\partial d}{\partial P}
\]

Aero/Structure Adjoint System:

\[
\begin{align*}
\left( \frac{\partial R_A}{\partial w} \right)^T \psi_A &= \frac{\partial C_D}{\partial w} & \left( \frac{\partial R_S}{\partial w} \right)^T \tilde{\psi}_S \\
\left( \frac{\partial R_S}{\partial d} \right)^T \psi_S &= \frac{\partial C_D}{\partial d} & \left( \frac{\partial R_A}{\partial d} \right)^T \tilde{\psi}_A
\end{align*}
\]

Adjoint Gradient:

\[
\frac{dC_D}{dP} = \frac{\partial C_D}{\partial P} - \psi_A^T \frac{\partial R_A}{\partial P} - \psi_S^T \frac{\partial R_S}{\partial P}
\]

Aerodynamics, e.g. Euler Eqn.:

\[ R_A = 0 \]

Structure:

\[ R_S = Kd - a = 0 \]

K: Symmetric stiffness matrix
a: Aerodynamic force
d: Displacement vector
P: Vector of Design variables

\[ \psi_A : \text{Aerodynamic Adjoint} \]

\[ \psi_S : \text{Structure Adjoint} \]

\[ \sim : \text{Lagged ...} \]
Coupled Aero-Structure Adjoint

\[ \frac{\partial R_A}{\partial d}, \frac{\partial R_A}{\partial P}, \frac{\partial C_D}{\partial d}, \frac{\partial C_D}{\partial P} \] : perturb shape by \( d,P \) \( \rightarrow \) calculate change in CFD residual

\[ \frac{\partial C_D}{\partial w} \] : perturb shape by \( d,P \) \( \rightarrow \) calculate change in drag coefficient

\[ \int_C \ldots \frac{\partial p}{\partial w} (n_x \cos \alpha + n_y \sin \alpha) \ldots \rightarrow \text{boundary condition} \]

\[ \ldots \text{has been derived in the last lecture!} \]

\[ \frac{\partial R_S}{\partial w} = \frac{\partial (Kd - a)}{\partial w} = -\frac{\partial a}{\partial w} \quad \text{: treat} \quad \int_C \ldots \frac{\partial p}{\partial w} \ldots \rightarrow \text{boundary condition} \]

\[ \frac{\partial R_S}{\partial d} = \frac{\partial (Kd - a)}{\partial d} = K = K^T \]

\[ \frac{\partial R_S}{\partial P} = \frac{\partial (Kd - a)}{\partial P} = \frac{\partial K}{\partial P} d - \frac{\partial a}{\partial P} \]
Coupled Aero-Structure Adjoint

Finite Differences:
Perturb the shape by each design variable and converge the aero-elastic loop until stationary behavior

Coupled Aero-Structure Adjoint:
Each 100 iterations the lagged $\tilde{\psi}_S$ is updated ...

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Coupled Aero-Structure Adjoint

Validation of Adjoint Gradient

\[
\frac{dC_D}{dP} = \frac{\partial C_D}{\partial P} - \psi_T A \frac{\partial R_A}{\partial P} - \psi_T S \frac{\partial R_S}{\partial P}
\]

NASTRAN shell/beam model
126 nodes
15 design variables
Ma=0.78
alpha=2.83
(300,000 cells)

AMP wing

N. Gauger et al.
Intro to Optimization and MDO, VKI, March 6-10, 2006
Validation of Adjoint Gradient

\[ \frac{dC_L}{dP} = \frac{\partial C_L}{\partial P} - \psi_A^T \frac{\partial R_A}{\partial P} - \psi_S^T \frac{\partial R_S}{\partial P} \]

NASTRAN shell/beam model
126 nodes
Ma=0.78
alpha=2.83
(300,000 cells)

AMP wing
Coupled Aero-Structure Adjoint

AMP wing

240 design variables (control points free form deformation)

Ma=0.78
alpha=2.83

Drag reduction by constant lift

\[ \Delta C_D = 24.9\% \]

\[ \Delta C_L = 0.1\% \]

feasible direction method
Coupled Aero-Structure Adjoint

AMP wing

240 design variables (control points free form deformation)

Ma=0.78
alpha=2.83

Drag reduction by constant lift
Coupled Aero-Structure Adjoint

AMP wing

240 design variables
(control points free form deformation)

Ma=0.78
alpha=2.83

Drag reduction by constant lift

Comparison of numerical effort:
(PC Pentium IV, 2.6 GHz, 2GB RAM)

• Coupled adjoint: 15 days
  (11 gradient and 91 state evaluations)

• Finite differences: 227 days
Range R:

\[ R \propto \frac{C_L}{C_D} \ln \frac{W}{W - F} \]

Fuel Weight F

Weight W:

\[ W = W_0 (1 + \lambda k_s) \]

Kreisselmeier-Steinhauser:

\[ k_s = \frac{1}{\beta} \ln \left( \sum_n \exp \left( \beta \frac{\sigma_n - \sigma_0}{\sigma_0} \right) \right) \]

\( \lambda = 0.2 \), \( \sigma_0 = 30.000 \) and \( \beta = 40 \)
Aero-Structure MDO

Range R:

\[ R \propto \frac{C_L}{C_D} \ln \frac{W}{W - F} = \frac{C_L}{C_D} \ln \left( \frac{1 + \lambda k_s}{1 + \lambda k_s - \frac{F}{W_0}} \right) \]

Fuel Weight F

Weight W:

\[ W = W_0 (1 + \lambda k_s) \]

Kreisselmeier-Steinhauser:

\[ k_s = \frac{1}{\beta} \ln \left( \sum_n \exp \left( \beta \frac{\sigma_n - \sigma_0}{\sigma_0} \right) \right) \]

\( \lambda = 0.2 \), \( \sigma_0 = 30.000 \) and \( \beta = 40 \)

Kreisselmeier-Steinhauser:

\[ \frac{d}{dP} = \frac{\partial k_s}{\partial P} + \psi^T \frac{\partial R_A}{\partial P} \]

adjoint b.c.

\[ n_x \psi_2 + n_y \psi_3 + n_z \psi_4 = -\frac{\partial k_s}{\partial P} \]
AMP wing

240 design variables
(control points free form deformation)

Ma=0.78
alpha=2.83

Range maximization by constant lift

\[ \Delta R = +37 \% \]
\[ \Delta ks = -10 \% \]
\[ \Delta C_D = -25 \% \]

feasible direction method
Adjoint Based Optimization

\[ \min I(w,x) \]

\[ \text{s.t. } R(w,x) = 0 \]

\[ \dim x = M \]

Adjoint solver

\[ R^*(w,\psi^k, x^n) = 0 \]

\[ \psi \]

\[ x^0 \]

start geometry

\[ x^0 \]

RANS solver

\[ R(w^k, x^n) = 0 \]

\[ w \]

\[ \nabla \]

\[ \int_V M_n dV \]

\[ (\nabla I)_m = \int_V i(w,\psi, (\delta x^n)_m) dV \]

\[ n=1,\ldots,N \]

\[ \text{m-loop} \]

\[ m=1,\ldots,M \]

\[ \nabla I \]

\[ x^{n+1} \]

optimization strategy

All at once?
Simultaneous Pseudo-Time stepping
- One Shot Approach -

\[
L(w, x, \psi) = I(w, x) - \psi^T R(w, x)
\]

\[
\nabla_w L(w, x, \psi) = 0 \quad \text{(adjoint equation)}
\]

\[
\nabla_x L(w, x, \psi) = 0 \quad \text{(design equation)}
\]

\[
R(w, x) = 0 \quad \text{(state equation)}
\]

\[
\begin{bmatrix}
    w + \Delta w \\
    x + \Delta x \\
    \psi + \Delta \psi
\end{bmatrix} =
\begin{bmatrix}
w \\
x \\
\psi
\end{bmatrix} -
\begin{bmatrix}
    L_{ww} & L_{wx} & \left(\frac{\partial R}{\partial w}\right)^T \\
    L_{xw} & L_{xx} & \left(\frac{\partial R}{\partial x}\right)^T \\
    \frac{\partial R}{\partial w} & \frac{\partial R}{\partial x} & 0
\end{bmatrix}^{-1}
\begin{bmatrix}
    \nabla_w L \\
    \nabla_x L \\
    R
\end{bmatrix}
\]

\[
\begin{bmatrix}
    \Delta w \\
    \Delta x \\
    \Delta \psi
\end{bmatrix} =
\begin{bmatrix}
    0 & 0 & I \\
    0 & B & \left(\frac{\partial R}{\partial x}\right)^T \\
    I & \frac{\partial R}{\partial x} & 0
\end{bmatrix}^{-1}
\begin{bmatrix}
    -\nabla_w L \\
    -\nabla_x L \\
    -R
\end{bmatrix}
\]

\text{KKT}

\text{Newton SQP method}

\text{inexact Newton rSQP method}

\text{simultaneous preconditioned pseudo time stepping}
Simultaneous Pseudo-Time stepping
- One Shot Approach -

\[ \psi^{k+1} = \psi^k - \Delta t \cdot R^*(w^{k+1}, \psi^k, x^k) \]

\[ w^{k+1} = w^k - \Delta t \cdot R(w^k, x^k) \]

\[ x^{k+1} = x^k - \Delta t \{ B_k^{-1} \nabla_x L - B_k^{-1} \left( \frac{\partial R}{\partial x} \right)^T \nabla_w L \} \]

\[ (\nabla_x L)_m = \int_V l(w^{k+1}, \psi^{k+1}, (\partial x^k)_m) dV \]

\[ \nabla_x L \]

\[ B_k \] – BFGS updates of reduced Hessian \( L_{xx} \)
Optimization problem
• drag reduction for RAE 2822
• inviscid flow
• \( M=0.73, \alpha=2^0 \)

Tools
• FLOWer
• FLOWer adjoint
Simultaneous Pseudo-Time stepping
- One Shot Approach -

Optimization at the cost of 4 flow simulations!
Optimal Design Scenario
Piggy–Back Approach

■ **Problem:** \( \min f(y,u) \) s.t. \( c(y,u) = 0, \)
where \( y \in \mathbb{R}^n \) and \( u \in \mathbb{R}^m \) are state and design variables

■ **Available:**

\[
\text{Code for } f(y,u) \text{ and } G(y,u) \approx y - \left( \frac{\partial}{\partial y} c(y,u) \right)^{-1} c(y,u)
\]

■ **Assumption:**

\( G, f \in C^{2,1}(\mathbb{R}^{n+m}) \) and \( \left\| \frac{\partial}{\partial y} G(y,u) \right\| \leq \rho < 1 \)

■ **Notation:**

\( N(\bar{y}, y, u) \equiv f(y,u) + \bar{y}G(y,u) \equiv \text{Lagrangian} + yy, \)
where the Lagrangian is formed w.r.t. \( c(y,u) \equiv G(y,u) - y \equiv 0 \)
Piggy–Back Approach
Single-step-one-shot

\[ y_{k+1} = G(y_k, u_k) \rightarrow \text{primal feasibility at } y_* \]

\[ \bar{y}_{k+1} = N_y(y_k, \bar{y}_k, u_k) \rightarrow \text{dual feasibility at } \bar{y}_* \]

\[ u_{k+1} = u_k - H_k^{-1} N_u(y_k, \bar{y}_k, u_k) \rightarrow \text{optimality at } U_* \]

where \( N_u = \bar{y}G_u + f_u \approx \text{reduced gradient} \)

and \( H_k \) is a suitable preconditioner
Spectral Analysis
Piggy–Back Approach

\[ \frac{\partial (y_{k+1}, \bar{y}_{k+1}, u_{k+1})}{\partial (y_k, \bar{y}_k, u_k)} = \begin{pmatrix} G_y & 0 & G_u \\ N_{yy} & G_y^T & N_{yu} \\ -H^{-1}N_{uy} & -H^{-1}G_u^T & I - H^{-1}N_{uu} \end{pmatrix} \]

has at \((y_*, \bar{y}_*, u_*)\) as eigenvalues \(\lambda\) the roots of

\[ P(\lambda) \equiv \text{det}[H(\lambda) + (\lambda - 1)H] \]

where

\[ H(\lambda) \equiv [Z(\lambda)^T, I] \nabla_{(y,u)}^2 N[Z(\lambda)^T, I]^T \]

\[ Z(\lambda)^T \equiv -G_u^T(G_y^T - \lambda I)^{-1} \]

Rows of \([Z(\lambda)^T, I]\) span tangent space of \(\{G(y,u) = \lambda y\}\)
Contractivity in convex case

Piggy–Back Approach

\[ \lambda < 1 \iff H > 0 \quad \text{i.e. positive definite} \]

\[ \lambda > -1 \iff H > H(-1)/2 \]

Numerical experience shows:

Reduced Hessian \( H \equiv H(1) \Rightarrow \) immediate blow-up

Projected Hessian \( H \equiv H(-1) \Rightarrow \) full-step convergence
Transonic case: NACA 0012 at $Ma = 0.8$ with $\alpha = 2^\circ$

Cost function: glide ratio

“FLOWer-Derivate” (2D Euler) + mesh deformation + parameterization

First and second derivatives by AD tool TAPENADE

Geometric constraint: constant thickness

Camberline/Thickness decomposition, 20 Hicks-Henne coefficients define camberline
min $\frac{C_D}{C_L}$ (Inverse Glide Ratio)

NACA 0012
$Ma = 0.8$ with $\alpha = 2^\circ$

Wing Shapes
Thanks for your attention!