Development of a Consistent Discrete Adjoint Solver for the SU² Framework

Tim Albring, Max Sagebaum, Nicolas Gauger

Chair for Scientific Computing
TU Kaiserslautern

16th Euro-AD Workshop, Jena
December 8, 2014
Open-source multiphysics suite SU²

Some key facts

- Developed at Stanford University (with global collaboration).
- Comprises complete self-contained optimization environment.
- Finite-Volume Discretization of (U)RANS equations and turbulence models using a highly modular C++ code-structure.
- Convergence Acceleration using Multigrid and local time-stepping methods.
Demands for a (discrete) Adjoint Solver

- Automatic advancement with the flow solver:

  Number of commits per week to the SU^2 repository.

- Smooth integration into the optimization framework (reuse of existing routines).

- Robustness: Inheritance of convergence properties of the flow solver.

- Consistency: Resulting gradients should constitute as the exact derivatives based on the particular numerical methods used in the flow solver.
Generic problem formulation

Given the state vector $y \in Y \subset \mathbb{R}^N$, the discrete design vector $\chi$, the computational mesh $u$ and the state equation $c(y, u) = 0$, the optimization problem can be formulated by the following generic problem description:

$$\min_{\chi} \quad J(y(u(\chi)), u(\chi))$$

subject to

$$R(y(u(\chi)), u(\chi)) = 0.$$  

- Gradient computation of $J$ is based on an implicitly defined state $y$.
- $f$ can be e.g. $C_D$, $C_L$ or $C_{Mz}$.
- Usually, there are additional constraints.
Aerodynamic Design Chain

Components

\[ \Delta u_{\text{surf}}(x) = \sum_{i=1}^{n} \chi_i \sin \left( \pi x \frac{\log 5}{\log t_i} \right) \]

Mesh Deformation

- Surface Deformation using Hicks-Henne functions
- Mesh Movement based on the Linear Elasticity Method

Flow Solver

- JST-based centered spatial discretization
- Implicit Euler temporal discretization
- SA or SST turbulence model
For Newton-type methods, $R(y, u)$ can be transformed into fixed-point equation $y = G(y, u)$ by using

$$G(y, u) := I - P(y, u)R(y, u)$$

with some suitable preconditioner $P$.

Feasible solutions $y = y(u)$ are then computed from the iteration

$$y_{n+1} = G(y_n, u).$$

It is natural to assume that $G$ is stationary only at feasible points, i.e.

$$R(y_*, u) = 0 \text{ iff. } y_* = G(y_*, u).$$

But: For a complex flow solver, the explicit structure of $P$ (and $G$) is unknown!
The Discrete Adjoint Solver

Lagrange viewpoint

Mesh Deformation is included by adding a constraint \( M(\chi) = u \) to the optimization problem*.

Lagrangian function:

\[
L(\chi, y, u, \bar{y}, \bar{u}) = J(y, u) + \bar{y}^T (G(y, u) - y) + \bar{u}^T (M(\chi) - u)
\]

\[=: N, \text{ Shifted Lagrangian} \]

KKT conditions give equations for \( \bar{y}, \bar{u} \) and \( dL/d\chi \):

\[
\bar{y} = \frac{\partial}{\partial y} J(y, u) + \frac{\partial}{\partial y} G^T(y, u)\bar{y} = \frac{\partial}{\partial y} N^T(y, \bar{y}, u)
\]

\[
\bar{u} = \frac{\partial}{\partial u} J(y, u) + \frac{\partial}{\partial u} G^T(y, u)\bar{y} = \frac{\partial}{\partial u} N^T(y, \bar{y}, u)
\]

\[
\frac{dL}{d\chi} = \frac{d}{d\chi} M^T(\chi)\bar{u}
\]

Comparison of Standard Form and Fixed-Point Form*

System of equations is solved using the iteration

\[
\left( D^n + \frac{\partial R^n}{\partial y} \right) \Delta y^n = -R^n, \quad y^{n+1} = y^n + \Delta y^n
\]

Comparison of Standard Form and Fixed-Point Form

System of equations is solved using the iteration

\[
\left( D^n + \frac{\partial R^n}{\partial y} \right) \Delta y^n = -R^n, \quad y^{n+1} = y^n + \Delta y^n
\]

Approximation to \( \frac{\partial R^n}{\partial y} \) due to
- first order spatial discretizations only,
- inconsistent boundary conditions,
- approximate solutions of the linear system.

Comparison of Standard Form and Fixed-Point Form

System of equations is solved using the iteration

\[
\begin{pmatrix}
D^n + \frac{\partial R^n}{\partial y}
\end{pmatrix} \Delta y^n = -R^n, \quad y^{n+1} = y^n + \Delta y^n
\]

Approximation to \( \frac{\partial R^n}{\partial y} \) due to
- first order spatial discretizations only,
- inconsistent boundary conditions,
- approximate solutions of the linear system.

\[
\left( \frac{\partial R^*}{\partial y} \right)^T \lambda = \frac{\partial J^*}{\partial y}
\]

Comparison of Standard Form and Fixed-Point Form*

System of equations is solved using the iteration

\[
\left( D^n + \frac{\partial R^n}{\partial y} \right) \Delta y^n = -R^n, \quad y^{n+1} = y^n + \Delta y^n
\]

Approximation to \( \frac{\partial R^n}{\partial y} \) due to
- first order spatial discretizations only,
- inconsistent boundary conditions,
- approximate solutions of the linear system.

No approximations allowed in this formulation to get correct (consistent) gradient.

Comparison of Standard Form and Fixed-Point Form*

System of equations is solved using the iteration

\[ \left( D^n + \frac{\partial R^n}{\partial y} \right) \Delta y^n = -R^n, \quad y^{n+1} = y^n + \Delta y^n \]

\[ G^* := y^* - \left( D^* + \frac{\partial R^*}{\partial y} \right)^{-1} R^* \]

\[ \bar{y}^{n+1} = \left( \frac{\partial G^*}{\partial y} \right)^T \bar{y}^n + \frac{\partial J^*}{\partial y} \]

Uses same approximative Jacobian, thus features convergence to consistent gradient with the same rate as state iteration.

Approximation to \( \frac{\partial R^n}{\partial y} \) due to

- first order spatial discretizations only,
- inconsistent boundary conditions,
- approximate solutions of the linear system.

The Discrete Adjoint Solver
Notes on the Implementation

Based on Algorithmic Differentiation for the gradient computation:
- Uses a modified version of the Adept C++ software library* (also prepared for dco and ADOL-C).
- Requires almost no modifications in the flow solver.
- Highly efficient in runtime:

\[
\frac{\text{Time for adjoint solution}}{\text{Time for flow solution}} \approx 1.2 - 2.5
\]

Actual value depends on the numerical methods used.
- Parallelized using AdjointMPI.

Performance Tuning

In the AD-context

1. Solve linear system in Reverse Sweep using analytic formulation:

\[ u = W^{-1}v \]

Reverse Mode

\[ s = W^{-T} \bar{u} \]
\[ \bar{W} = s \cdot u^T \]
\[ \bar{v} += s \]
\[ \bar{u} = 0 \]

2. Assume a passive preconditioner \( P \) in the non-linear fixed-point solver:

\[
\frac{\partial G(y, u)}{\partial y} = I - P(y, u) \frac{\partial R(y, u)}{\partial y} + \frac{\partial P(y, u)}{\partial y} R(y, u)
\]

Reasonable to drop this term since \( R(y_k, u) \) goes to zero.

Both methods can efficiently be combined.
Setting and Performance

- GMRES + LU-SGS Preconditioner (BCGSTAB also possible).
- Matrix stored using Block Compressed Row Storage.
- Linear System in Reverse Sweep solved using same solver with differentiated Matrix-Vector product.
- Multi-grid disabled.

**Flow Solver performance:** 0.21s per Iteration (5 Iter. of Linear Solver), 85 MB peak memory

<table>
<thead>
<tr>
<th>Linear Iter.</th>
<th>Full Taping</th>
<th>Active Prec.</th>
<th>Passive Prec.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runtime</td>
<td>1.42</td>
<td>2.21</td>
<td>1.46</td>
</tr>
<tr>
<td>Memory</td>
<td>30.04</td>
<td>24.42</td>
<td>23.52</td>
</tr>
</tbody>
</table>

Relative performance of Adjoint Solver.

Development of a Consistent Discrete Adjoint Solver
Convergence - Adjoint Solver

\( \rho_{rms} = 10^{-4} \)

Passive Preconditioner leads to small difference in gradient (error \( \approx 0.05\% \)). Much faster with reduced number of linear iterations (despite slightly higher number of outer iterations).
Convergence - Adjoint Solver

\( \rho_{rms} = 10^{-7} \)

(c) Convergence \( \bar{y}_\rho \)  
(d) Convergence \( \bar{u} \)
### Inviscid Flow

**Test case definition**

**Baseline design:** NACA0012

<table>
<thead>
<tr>
<th>Flow Conditions</th>
<th>Numerical Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Ma_\infty = 0.8$</td>
<td>Mesh: 10216 Elements</td>
</tr>
<tr>
<td>$\alpha = 1.25^\circ$</td>
<td>Multigrid: 3 Level W-Cycle, $CFL = 4.0$.</td>
</tr>
<tr>
<td></td>
<td>38 Hicks-Henne functions</td>
</tr>
</tbody>
</table>

**Problem statement:**

$$\min_{\chi} \quad C_D(\chi)$$

subject to:

- $C_L(\chi) \geq 0.326$
- $C_{Mz}(\chi) \geq 0.0336$. 
Inviscid Flow

Validation of gradient with Finite Differences resulted in $l_2$-Norm distance of 0.068%.

(e) Optimization History

(f) Pressure Distribution $C_p$
Viscous Flow - SA

Test case definition

**Baseline design:** RAE2822

**Flow conditions**
- $Ma_\infty = 0.729$
- $\alpha = 2.31^\circ$
- $Re = 6.5 \text{ Mio}$

**Numerical Settings**
- Mesh: 59976 Elements
- 3 Level W-Cycle, $CFL = 4.0$
- 38 Hicks-Henne functions

**Problem statement**

$$\min_{\chi} \quad C_D(\chi)$$

subject to:

- $C_L(\chi) = 0.707$
- $C_{Mz}(\chi) \geq -0.088$
- Area = Area_{initial}$
Viscous Flow - SA

Validation

(g) Drag Gradient

(h) Lift Gradient
Viscous Flow - SA
Results - Optimization History

(i) Discrete Adjoint

(j) Continuous Adjoint
Viscous Flow - SA

Results - $C_p$

(k) Pressure Distribution

- Experimental Data - RAE2822
- Baseline Design - RAE2822
- Final Design - Disc. Adj.
- Final Design - Cont. Adj.
Viscous Flow - SST

Results

(1) Optimization History

(m) Pressure Distribution
Viscous Flow

Pressure Contours

(n) Baseline

(o) Optimized - Discrete Adjoint
Figure: Magnitude of $\bar{u}_{surf}$ for the Onera-M6 test case using the SST turbulence model.
Summary & Outlook

Summary

- Differentiation of the entire design chain of SU².
- Consistent gradients of complex flow models.
- Efficient and easy maintainable/extensible adjoint solver.

Outlook

- Further performance optimizations.
- Application to 3D and unsteady flows.
- Release of the software suite to the public (early 2015).
- Extension to multi-disciplinary optimization (aeroacoustics, aero-structure coupling).
Thank you for your attention!