Optimization of Turbine Engine Cycle Analysis with Analytic Derivatives

Tristan Hearn, Eric Hendricks, Jeffrey Chin, Justin Gray, Kenneth T. Moore

NASA Glenn Research Center, Cleveland, OH

June 16th, 2016
Optimization of a separate flow turbofan design was performed with analytic derivatives using the cycle analysis code Pycycle.

Computation cost on average was $1/3$ that of an optimization performed on an NPSS implementation, with finite-difference derivatives.
Pycycle Overview

Pycycle is a 1D cycle modeling tool similar to NPSS, but with an extra level of decomposition\(^1\)

This allows for the implementation of analytic derivatives

---

OpenMDAO computes coupled derivatives for complex multidisciplinary models automatically.

Forward:

$$\frac{dF}{dx_i} = \frac{\partial F}{\partial x_i} - \frac{\partial F}{\partial y} \left( \frac{\partial R}{\partial y} \right)^{-1} \frac{\partial R}{\partial x_i}$$  \hspace{1cm} (1)$$

Adjoint:

$$\frac{dF_i}{dx} = \frac{\partial F_i}{\partial x} - \left( \frac{\partial R^T}{\partial y} \right)^{-1} \frac{\partial F_i^T}{\partial y} \frac{\partial R}{\partial x}^T,$$  \hspace{1cm} (2)$$
Analytic derivative benefits

Analytic derivatives provide significant computational savings for gradient based optimization

Computational Cost vs # of Design Variables

- ALPSO
- SNOPT - FD
- SNOPT - Fwd. Analytic
- SNOPT - Adjoint Analytic

Optimization of Turbine Engine Cycle Analysis with Analytic Derivatives
A separate flow turbofan model was built in both Pycycle and NPSS and optimized in OpenMDAO

Minimize:

TSFC

With respect to:

1 ≤ FPR ≤ 2
1 ≤ CPR ≤ 30
1 ≤ BPR ≤ 12
1 ≤ W ≤ 2000 lbm/s

Such That:

OPR = 30
$F_n = 25,000$ lbf
$T_4 ≤ 3000^{\circ}R$

Flight condition: 35,000 ft, 0.8 MN
Pycycle and NPSS based optimizations drove towards the same answer

<table>
<thead>
<tr>
<th></th>
<th>Baseline</th>
<th>Optimized (Pycycle)</th>
<th>Optimized (NPSS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPR</td>
<td>1.5</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>CPR</td>
<td>10.3</td>
<td>15.0</td>
<td>15.0</td>
</tr>
<tr>
<td>BPR</td>
<td>5.0</td>
<td>12.0</td>
<td>12.0</td>
</tr>
<tr>
<td>$W$</td>
<td>500.0</td>
<td>1069.2</td>
<td>1032.40</td>
</tr>
<tr>
<td>TSFC</td>
<td>0.612</td>
<td>0.331</td>
<td>0.320</td>
</tr>
</tbody>
</table>

Mass flow and TSFC vary between codes due to a thermodynamic discrepancy.
Both internal solver tolerances were set to $10^{-5}$

Pycycle converged to much tighter tolerances overall

<table>
<thead>
<tr>
<th></th>
<th>Pycycle</th>
<th>NPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. constraint violation</td>
<td>$3.5 \cdot 10^{-15}$</td>
<td>$1.2 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>ShaftL_{net pwr.}</td>
<td>$1.64 \cdot 10^{-6}$</td>
<td>$-0.022$</td>
</tr>
<tr>
<td>ShaftH_{net pwr.}</td>
<td>$6.11 \cdot 10^{-8}$</td>
<td>$2.826 \cdot 10^{-6}$</td>
</tr>
</tbody>
</table>
Analytic Derivatives give fewer iterations and lower wall time on average

<table>
<thead>
<tr>
<th></th>
<th>Pycycle</th>
<th></th>
<th>NPSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>FD step size</td>
<td>-</td>
<td>$10^{-5}$</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$0.99 \cdot 10^{-3}$</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>SNOPT iterations</td>
<td>44</td>
<td>120</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td></td>
<td>721</td>
<td>11</td>
</tr>
<tr>
<td>Run time (s)</td>
<td>3753</td>
<td>30912</td>
<td>12796</td>
</tr>
<tr>
<td></td>
<td></td>
<td>131581</td>
<td>1071</td>
</tr>
<tr>
<td></td>
<td></td>
<td>18788</td>
<td></td>
</tr>
</tbody>
</table>

- NPSS optimizations were highly sensitive to step size
- Difference in compute cost is primarily due to the difference in the cost of computing derivatives
- Tight tolerance requires more iterations for each FD step
Conclusions

- Results suggest analytic derivatives are suitable for optimization of engine cycle analysis.
- Optimizations performed using engine cycle analysis outperform analyses performed using finite-difference derivatives.
- Access to analytic adjoint derivatives will enable more ambitious MDO problems (propulsion-airframe, propulsion-mission, etc.).
Acknowledgments

- TAC Transformational Tools and Technologies Project
- Thomas Lavelle, NASA GRC
- Christopher Snyder, NASA GRC