Higher Fidelity Analysis in
Wind Turbine Multi-disciplinary Design Optimization

Michael McWilliam

Danish Technical University
Outline

• Direct Optimization at Higher Fidelity
  • Medium Fidelity Analysis Tools
  • The FE Based Vortex Dynamics
  • Optimization Results

• Multi-fidelity Design Optimization
  • The AMMF Algorithm
  • Structural Design Case Study

• Closing Statements
Direct Optimization with Higher Fidelity Analysis
• Trends show wind turbines are getting larger
  • Higher turbines better winds
  • Improved economies of scale (e.g. offshore)

• Future growth will require advanced designs
  • Bend-twist coupling, curved blades, active load alleviation, winglets, coning, etc.

• Multidisciplinary Design Optimization (MDO)
  • Simultaneously optimize multiple disciplines (e.g. aero, structural, control, etc.)
  • Optimization based on holistic metrics (e.g. cost of electricity)
  • Wind turbine design constrained by unsteady loads (i.e. strong gusts and fatigue)
Medium Fidelity Analysis Tools
Direct Optimization with Higher Fidelity Analysis

Analysis Tools

- Conventional preliminary design tools
  - Blade Element Momentum Theory and Linear beam theory
  - Fast and efficient, but lacks the fidelity required by advanced designs
- High fidelity analysis
  - Grid-based CFD and Shell and Brick based FEM
  - Excellent fidelity, very expensive for optimization
- Need medium fidelity analysis (improved fidelity, still efficient)
  - Vortex Dynamics (VD)
  - Nonlinear beam theory (GEBT)
  - Anisotropic Cross Section Analysis (VABS)
• Aeroelastic model with Conventional VD, GEBT and VABS
• Obtained optimization results with
  • Pure aerodynamic
  • Aero-elastic with fixed wake
• Failed to obtain aeroelastic results with free wake simulations
  • Pure vortex methods are fundamentally chaotic
  • Numerical noise spoils the gradients and optimization
• Conventional VD not suitable for aero-elastic optimization

The Finite Element Based Vortex Dynamics
• Vortex position in the wake defined by interpolating splines:

\[ \mathbf{x} = \sum_j \eta_j(\tau) \mathbf{X}_{xj} \quad \dot{\mathbf{x}} = \sum_j \dot{\eta}_j(\tau) \mathbf{X}_{xj} \]

• Can have an arbitrary number of influence elements and control points
  • Can add more influence elements to improve accuracy
  • Can remove control points to accelerate calculations
• Convergence defined by a residual:

\[ r_x \equiv \dot{x} + \Omega \times (x - x_0) - u_\infty - u_\gamma \]

• Mapped to control points through Galerkin projection:

\[ R_{xj} = \int_{\tau_0}^{\tau_f} \zeta_j(\tau) r_x(\tau) d\tau \]

• Solved with a Newton iteration
  • Adaptive relaxation required to get reliable convergence
  • See Video for example

• Best results with a far-wake model
  • Avoids singularities
  • Eliminates wake-truncation errors
Optimization Results
Direct Optimization with Higher Fidelity Analysis

Optimization Convergence with FEM-Based VD

- Used analytic gradients
- Explicit VD residual definition predicts changes in state
- Tight optimization tolerances
- Small changes avoid singularities
Aerodynamic Only Optimization:

Aero-elastic Optimization:

- Aeroelastic optimization created more efficient designs
Multi-fidelity Design Optimization
Multi-fidelity Design Optimization

The Multi-Fidelity concept

- Uses both a high fidelity and low fidelity model
  - Less expensive by using fewer high fidelity results
  - Reduces surrogate error with low-fidelity results

- Fidelity could be based on:
  - Formulation (e.g. RANS vs. BEM)
  - Grid resolution (e.g. fine vs. course)
  - Type of simulation (e.g. unsteady vs. steady)
  - etc.

- Low fidelity just needs to show similar trends
The AMMF Algorithm
Multi-fidelity Design Optimization

The AMMF Algorithm

- High fidelity used for accuracy
- Low fidelity is used for speed
- Correction for first order consistency

\[ \tilde{f}(x) = f_l(x) + \beta(x) \]

\[ \beta(x) = f_{h0} - f_{l0} + (\nabla f_{h0} - \nabla f_{l0}) \Delta x \]

- Trust-region for robustness
Multi-fidelity Design Optimization

The Trust Region Algorithm

- The trust-region defines the region where we can “trust” our approximation
- Constrained to stay within the trust-region
- Re-centered at every major iteration
  - Only when an improved is found
- Trust region is resized
  - If the approximation gives excellent agreement then it grows
  - If the trust region gives poor agreement then it shrinks
    - If the inner optimization fails to find an improvement, it will repeat within the smaller trust region
    - Similar to the line search algorithm
  - Otherwise maintain the trust region
Constraints in the AMMF Algorithm

• Constraints are corrected in the same way
• The constraints are present in the low fidelity optimization
• Constraints receive special treatment in Approximation and Model Management Framework (AMMF)

First an estimated Lagrangian is calculated

\[ \Phi = f + \tilde{\lambda}_e \cdot |c| + \tilde{\lambda}_i \cdot \max(0, -c_i) \]

• \( \tilde{\lambda} \) are the Lagrange multipliers estimated from previous iterates.
• \( \tilde{\lambda} \) is specified for the first iteration

• New iterate only accepted when \( \Phi_i < \Phi_{i-1} \)
• Trust region is expanded or contracted based on \( M \):

\[ M = \frac{\Phi_{i-1} - \Phi_i}{\Phi_{i-1} - \tilde{\Phi}_i} \]

• Trust region expanded if \( M \) is close to 1
• Trust region contracts if \( M \) is far from 1
Multi-fidelity Structural Design Optimization
## Summary of Low Fidelity Tools

<table>
<thead>
<tr>
<th>Position</th>
<th>EA</th>
<th>Elx</th>
<th>Ely</th>
<th>GJ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05</td>
<td>0.0</td>
<td>2.6</td>
<td>-4.9</td>
<td>-5.4</td>
</tr>
<tr>
<td>0.15</td>
<td>0.5</td>
<td>1.1</td>
<td>-3.0</td>
<td>-0.8</td>
</tr>
<tr>
<td>0.25</td>
<td>-0.4</td>
<td>-1.8</td>
<td>2.1</td>
<td>-1.4</td>
</tr>
<tr>
<td>0.35</td>
<td>-0.7</td>
<td>-2.6</td>
<td>1.7</td>
<td>-3.1</td>
</tr>
<tr>
<td>0.45</td>
<td>-0.7</td>
<td>-3.1</td>
<td>1.0</td>
<td>-5.5</td>
</tr>
<tr>
<td>0.55</td>
<td>-0.9</td>
<td>-3.1</td>
<td>-0.3</td>
<td>-7.7</td>
</tr>
<tr>
<td>0.65</td>
<td>-0.8</td>
<td>-2.9</td>
<td>-1.7</td>
<td>-9.3</td>
</tr>
<tr>
<td>0.75</td>
<td>-0.6</td>
<td>-2.2</td>
<td>-2.2</td>
<td>-9.2</td>
</tr>
<tr>
<td>0.85</td>
<td>-0.6</td>
<td>-1.7</td>
<td>-3.5</td>
<td>-5.9</td>
</tr>
<tr>
<td>0.95</td>
<td>-0.1</td>
<td>-1.2</td>
<td>-2.0</td>
<td>-2.0</td>
</tr>
</tbody>
</table>

Table: Percent Error with BECAS

- **Low fidelity cross section tool**
  - Thin-walled cross section assumption
  - Rigid cross section (Euler-Bernoulli)
  - Classic laminate theory
  - Written in C++
  - Python bindings with Swig
  - Will have analytic gradients
  - Within 10% compared to BECAS

- **High fidelity cross section tool**
  - Based on BECAS
  - BECAS uses an FE formulation
  - Solves the warping field
  - Gives fully populate matrix
## Summary of Low Fidelity Tools

<table>
<thead>
<tr>
<th>Operation</th>
<th>Calculation time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Beam Model</td>
<td>0.0035</td>
</tr>
<tr>
<td>LF cross section model</td>
<td>0.0074</td>
</tr>
<tr>
<td>BECAS</td>
<td>200.1866</td>
</tr>
</tbody>
</table>

Table: Speed Comparison of Low Fidelity Tools

- **Linear Beam Model**
  - C++ code from my PhD
  - Analytic gradients wrt.
    - Positions
    - Orientation
    - Cross section properties
    - Applied forces
  - Solves equivalent forces for given deflection

- **Speed comparison:**
  - With python bindings
  - Calculation for whole blade
  - 19 elements
  - DTU 10MW
Problem Description

- Minimize DTU 10MW Blade Mass
- Varying spar cap thickness
- Subject to:
  - Tip deflection constraint
- Analysis based on the equivalent static problem (i.e. Frozen loads)
- Compared pure BECAS, pure CLT and AMMF
- Looked at various AMMF configurations:
  - Additive vs. Multiplicative corrections
  - Trust region size
  - Initial Lagrange multiplier (i.e. Penalty parameter)
Multi-fidelity Design Optimization

Optimization Results

- Low fidelity model is not conservative
  - Will produce infeasible solutions
- AMMF reproduced the BECAS solution
  - AMMF had better constraint resolution
- AMMF gives accurate corrections
- Additive vs multiplicative corrections:
  - Gives similar solutions
  - Similar performance
Multi-fidelity Design Optimization

Optimization Convergence

- AMMF converges 12 times faster
  - Just 2 major iterations
- AMMF had smoother convergence
  - Only 1 iteration with constraint violation
  - BECAS optimization ended due to maximum iterations
- Low fidelity models more suitable for optimization
AMMF guards against poor approximations

- Unconstrained has all protections disabled
  - Large violations
  - Fails to converge
- Trust region is most robust
  - Same progress as ideal configuration
- Large penalties work without trust region
  - No large violations
  - More searching
Closing Statements
Conclusions

• Higher fidelity in direct optimization is challenging but possible
  • Underlying tools may be non-smooth
  • Tools may need to be re-written or re-formulated (optimization proof)
  • Developed a totally new formulation for vortex methods based on FEM
  • Successfully obtained aero-elastic optimization results with vortex methods

• Higher fidelity through multi-fidelity design optimization is promising
  • Effective when low fidelity gives similar trends much faster
  • Achieved a 12 times speed up using multi-fidelity techniques
  • The AMMF algorithm is robust in handling errors
  • Ongoing case studies focusing on difficult problems
This work was supported by

- The University of Victoria
- Natural Sciences and Engineering Research Council of Canada
- The Technical University of Denmark
- The Danish Energy Technology Development and Demonstration Program
Closing Statements

Thank-you for your interest

Comments or Questions?