Towards high-fidelity wind turbine analysis
S.F. van den Broek
Background

Increase in wind energy

Global cumulative wind energy production [MW]

(Source: Global Wind Energy Council Global Wind Statistics 2012)
Background

Wind turbine size

Airbus A380-800: 79.8 m

30 m 1990
  e.g. Vestas
  V27-225 kW

80 m 1998
  e.g. Vestas
  V80-2.0 MW

112 m 2002
  e.g. Vestas
  V112-3.0 MW

164 m 2015
  e.g. Vestas
  V164-8.0 MW

(Source: Identification of wind energy systems, by G.J. van der Veen)
Mono-discipline design

No or simplified controller
Aeroelasticity
Aeroelasticity of trees

- Mono-disciplinary analysis:
  - Static structure $\rightarrow$ very high loads
  - Constant high loads $\rightarrow$ overestimated deflection/damage of/to tree.
- Reality: Reduction of aerodynamic loads
  - Deflection of leaves
  - Deflection of branches
Why multi-disciplinary design analysis?

- Aeroelastic effects
  - Deformation effect on the aerodynamic performance and v.v.
- Embed controller
  - Controller design earlier in the design phase
- Closer to real-world results

(Haghighat et al)
Current tools, low fidelity

- **Structure**
  - Low fidelity, beam or modal representation
  - No or simplified bend-twist coupling
  - Depends on programs to calculate equivalent properties (Chen et al.)

- **Aerodynamics**
  - Low fidelity, usually Blade Element Momentum
  - Many assumptions, e.g. no spanwise flow, infinite amount of blades
Necessity of high-fidelity

- Reduce dependence on beam property extraction programs
- Fewer assumptions, increased accuracy
- Composite bend-twist coupling
- Geometric bend-twist coupling
- Analysis of non-traditional aerodynamics

(Morphing wing structure with controllable twist based on adaptive bending–twist coupling Wolfram Raither et al 2013 *Smart Mater. Struct.* 22 065017)
Goal - The first steps

- Couple an aerodynamic code with a high-fidelity structural code
  - Study interfaces
  - Couple codes
- Check results
  - Find verification model
  - Do model-to-model verification
MDOLab codes

- TACS (Toolkit for the Analysis of Composite Structures)
  - C++ code
  - Structural solver
  - Shell elements
  - Composites
  - Derivatives
- pyAeroStruct
  - Aerostructural managing software
  - Python-level
  - Four different solution methods
    - Linear block Gauss-Seidel
    - Non-linear block Gauss-Seidel
    - Coupled Krylov
    - Coupled Newton-Krylov
  - Designed to be modular for aerodynamic solvers

[Kennedy and Martins, AIAA MAO, 2012]
AeroDyn

- Developed by the National Renewable Energy Lab
- FORTRAN code
- Widely used in combination with FAST and ADAMS
- Uses low-fidelity Blade Element Momentum
- Many corrections to BEM built in
- Interface is well documented
The codes and their relation

- **NASTRAN input file (.bdf)**
- **AeroDyn input file**
- **Material properties & Analysis settings**
- **FORTRAN to Python (F2py)**
- **Output**

### Codes:
- **NASTRAN**: Input file (.bdf)
- **TACS**: C++ to Python (SWIG)
- **pyTACS**: Python wrapper
- **AeroDyn**: Analysis tool
- **pyAeroDyn**: Python wrapper
- **Aerowrap**: FORTRAN to Python (F2py)
- **F2py**: Python wrapper
forces this would require. Pressure energy can be extracted in a step-like manner, however, and all wind turbines, whatever their design, operate in this way. The presence of the turbine causes the approaching air, upstream, gradually to slow down such that when the air arrives at the rotor disc its velocity is already lower than the free-stream wind speed. The stream-tube expands as a result of the slowing down and, because no work has yet been done on, or by, the air its static pressure rises to absorb the decrease in kinetic energy.

As the air passes through the rotor disc, by design, there is a drop in static pressure such that, on leaving, the air is below the atmospheric pressure level. The air then proceeds downstream with reduced speed and static pressure – this region of the flow is called the wake. Eventually, far downstream, the static pressure in the wake must return to the atmospheric level for equilibrium to be achieved. The rise in static pressure is at the expense of the kinetic energy and so causes a further slowing down of the wind. Thus, between the far upstream and far wake conditions, no change in static pressure exists but there is a reduction in kinetic energy.

3.2 The Actuator Disc Concept

The mechanism described above accounts for the extraction of kinetic energy but in no way explains what happens to that energy; it may well be put to useful work but some may be spilled back into the wind as turbulence and eventually be dissipated as heat. Nevertheless, we can begin an analysis of the aerodynamic behaviour of wind turbines without any specific turbine design just by considering the energy extraction process. The general device that carries out this task is called an actuator disc (Figure 3.2).

Upstream of the disc the stream-tube has a cross-sectional area smaller than that of the disc and an area larger than the disc downstream. The expansion of the stream-tube is because the mass flow rate must be the same everywhere. The mass of air which passes through a given cross section of the stream-tube in a unit length of time is \( rAU \), where \( r \) is the air density, \( A \) is the cross-sectional area and \( U \) is the flow velocity. The mass flow rate must be the same everywhere along the stream-tube and so \( rA1U1 = rAdUd = rwUw \) (3.1).

The symbol \( 1 \) refers to conditions far upstream, \( d \) refers to conditions at the disc and \( w \) refers to conditions in the far wake.

It is usual to consider that the actuator disc induces a velocity variation which must be superimposed on the free-stream velocity. The stream-wise component of this induced flow at the disc is given by \( /C0aU1 \), where \( a \) is called the axial flow induction factor, or the inflow factor. At the disc, therefore, the net stream-wise velocity is \( Ud = U1(1/C0a) \) (3.2).

3.2.1 Momentum theory

The air that passes through the disc undergoes an overall change in velocity, \( U1/Uw \) and a rate of change of momentum equal to the overall change of velocity times the mass flow rate:

\[
\text{Rate of change of momentum} = (U1/Uw)rA Ud (3.3)
\]

The force causing this change of momentum comes entirely from the pressure difference across the actuator disc because the stream-tube is otherwise completely surrounded by air at atmospheric pressure, which gives zero net force. Therefore, \( (p+td/C0pd/C0d)Ad = (U1/Uw)rAdU1(1/C0a) \) (3.4).

To obtain the pressure difference \( (p+td/C0pd/C0d) \) Bernoulli's equation is applied separately to the upstream and downstream sections of the stream-tube; separate equations are

\[
\text{Stream-tube}
\]

\[
\text{Velocity}
\]

\[
\text{Pressure}
\]

\[
\text{Actuator disc}
\]

\[
\text{Velocity}
\]

\[
\text{Pressure}
\]

\[
\text{Velocity and pressure drop}
\]

\[
\theta
\]

\[
\text{Velocity and pressure drop}
\]

Idealized flow through wind turbine

Blade split in to elements

Forces are a function of the inflow speed
Inputs for AeroDyn

- Initialization:
  - Wind conditions
  - Undeformed configuration
  - Options

- Per calculation
  - Position
  - Orientation
  - Translational velocity
  - Rotational velocity
Meshes
Meshes

- Deflected structure
- Deflected intermediate
- Undeflected structure
Force application

\[ F_x = 9757 \text{ N/m} \]
\[ M_z = -3025 \text{ N \cdot m/m} \]

\[ f_{mean,x} + f_{m,1} + f_{m,2} = f_{mc,1} + f_{mc,2} \]
Non-linear block Gauss-Seidel

AD → ST → AD → ST → ST → AD

Converged?

AD: Aerodynamic solver
ST: Structural solver
Convergence (iteration 1)
Convergence (iteration 2)
Convergence (iteration 3)
Convergence (iteration 4)
Convergence (iteration 5)
Convergence (iteration 6)
Challenge the future

Convergence (iteration 7)
Convergence (iteration 8)
Choice of verification model

- No suitable experimental data publicly available.
- Limited amount of high-fidelity models

NREL 5 MW
- Cross sectional properties

Sandia 5 MW
- Higher-fidelity reversed engineered NREL 5 MW blade.
- Export option to FAST
- Stiffness and mass distribution are similar to the NREL model.
Verification model

- Remodeled manually
- Small changes in spar orientation

** PIC OF SPARS **

<table>
<thead>
<tr>
<th></th>
<th>NREL</th>
<th>Sandia (ANSYS)</th>
<th>Sandia (FAST)</th>
<th>TACS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass [kg]</td>
<td>17740</td>
<td>17700</td>
<td>16878</td>
<td>17249</td>
</tr>
</tbody>
</table>
Power control in wind turbine

- Below rated wind speed
  - Change rotational speed of blade

- Above rated wind speed
  - Change pitch of blade
Results

Power output at set wind speed.

Power [kW]

0 5 10 15 20 25

-3000 -2000 -1000 0 1000 2000 3000 4000 5000 6000

TACS

FAST
Twisting

Twist of blade in z-direction [deg] at 11 m/s wind speed
Hypothesis

(Source: http://www.mathworks.com/help/physmod/powersys/ref/windturbine.html)
Manually “tuned” turbine

Pitch angle at set wind speed

Power output at wind speed wind speed.
Conclusions

- Results indicate that the torsional stiffness has a very significant effect.
- This becomes more important as blades become more flexible.
- Beam model
  - Anisotropic beam models
  - Beam properties
- Additional verification recommended with similar fidelity model
Future work

- Additional verification
- Validation
- Add derivative computation to (py)AeroDyn
- Perform optimization
- Add time-domain analysis
- Add higher fidelity aerodynamics (CFD)
Thank you

• Questions?
Derivative computation

Figure 9: Gradient evaluation cost for first-order finite differencing and the coupled adjoint method versus number of design variables; one unit of normalized time corresponds to one aerostructural solution for each geometric design variable. Nevertheless, each additional design variable requires only 0.005% of the aerostructural solution time.

It is worth comparing the current results with the previous work of Martins et al.\[31\]. In that work, the coupled adjoint cost was found to scale with the number of design variables according to $3.4 + 0.01N_x$. Since the constant term in the equation includes the aerostructural solution, the coupled adjoint solution had a baseline cost of 2.4. The present method scales according to $1.68 + 5 \times 10^{-5}N_x$, as indicated in Figure 9. This corresponds to a baseline cost for the coupled adjoint of 0.67, i.e., a 72% reduction relative to the previous implementation. This is primarily due to the elimination of the finite differencing that was used to compute the off-diagonal coupled adjoint terms. This improvement is even more significant in absolute terms because the aerostructural solution of the new implementation is also much more efficient. Additionally, the slope in the dependency on the number of design variables has been reduced by over two orders of magnitude. This is achieved by eliminating the use of finite-difference derivatives in the total-derivative equation (15). We have shown that the new implementation of the coupled adjoint method exhibits extremely good design-variable scaling. The coupled computational cost can be considered practically independent of the number of design variables, and it is now feasible to compute gradients with respect to 35 of 42