Wind Energ. Sci. Discuss., community comment CC1
https://doi.org/10.5194/wes-2021-116-CC1, 2021

Comment on wes-2021-116

Mads Pedersen

Community comment on "FLOW Estimation and Rose Superposition (FLOWERS): an integral approach to engineering wake models" by Michael J. LoCascio et al., Wind Energ. Sci. Discuss., https://doi.org/10.5194/wes-2021-116-CC1, 2021

Thank you for a very interesting paper. I really like the idea of fitting the wind rose with a fourier series to avoid the wake rays which is clearly seen when applying the tophat model for a limited number of wind directions.

I have a few comments, questions and corrections:

- Is it possible to include local flow conditions without a time-consuming iterative approach?
- As I understand it, you will try to fix the power integral and allow a wind-speed dependent cp.
  Will this solution capture the fact that a annual wind distribution normally also includes wind speeds above rated with lower cp and is it possible to also allow a wind-speed dependent ct

- I assume the codes are implemented in python using numpy. The problem is that a python loop is so much slower than a numpy vector operation.
  The time comparison is therefore only fair if the none of the codes contain extra python loops compared to the other, which could be vectorized.
- line 48: 180 > 270
- section 3.1: You indicate that you are using local waked wind speed instead of free-stream wind speed when scaling the Jensen deficit, is that correct?
- line 177: "*FLOWERS is only computing the velocity through each turbine at a single point instead of an array of points on the rotor area.*". How many points are evaluated with the Jensen model
- line 209: speed > direction?
- line 245: "*One reason why the Jensen optimizer favors this type of solution is that wake-added turbulence is included in the modeling framework, so maximizing the streamwise spacing of the turbines improves the wake recovery for downstream turbines.*"
  You claim to have identical wake expansion factor, which I assume means a constant factor. This conflicts with improved wake recovery due to wake-added turbulence.
  As far as I can see, the Jensen simulation gives a interference mesh of wake "fingers", which results in many local optima, where the optimizer gets stuck as you also write in section 4.1
- line 253: "corresponds to Case 3". It looks more as Case 2
- Is it correct understod that all 4x10 runs in figure 8, 9, 11 and 12 would end up with

the same AEP gain (same optimized layout) if the optimizations were perfect?
Maybe the same 10 cases could be compared in the one figure for the five different approaches (Jensen_72, Jensen_360, FLOWERS, Gaus_360, Gaus_9) if it does not spoil your nice flow of the story