GPU Based Fast Free-Wake Calculations For Multiple Horizontal Axis Wind Turbine Rotors

M Türkal \(^{4,b}\), Y Novikov \(^{1,d}\), S Üşenmez \(^{4,e}\), N Sezer-Uzol \(^{3,e}\), O Uzol \(^{1,2,a}\)

1 Department of Aerospace Engineering, Middle East Technical University (METU), Ankara, TURKEY
2 METU Center for Wind Energy (METUWIND), Ankara, TURKEY
3 TOBB University of Economics and Technology, Ankara, TURKEY
4 Aerotim Ltd., Ankara, TURKEY
E-mail: \(^{a}\)uzol@metu.edu.tr, \(^{b}\)turkal@aerotim.com.tr, \(^{c}\)nsezeruzol@etu.edu.tr,
\(^{d}\)yaroslav.novikov@metu.edu.tr, \(^{e}\)usenmez@aerotim.com.tr

Abstract. Unsteady free-wake solutions of wind turbine flow fields involve computationally intensive interaction calculations, which generally limit the total amount of simulation time or the number of turbines that can be simulated by the method. This problem, however, can be addressed easily using high-level of parallelization. Especially when exploited with a GPU, a Graphics Processing Unit, this property can provide a significant computational speed-up, rendering the most intensive engineering problems realizable in hours of computation time. This paper presents the results of the simulation of the flow field for the NREL Phase VI turbine using a GPU-based in-house free-wake panel method code. Computational parallelism involved in the free-wake methodology is exploited using a GPU, allowing thousands of similar operations to be performed simultaneously. The results are compared to experimental data as well as to those obtained by running a corresponding CPU-based code. Results show that the GPU based code is capable of producing wake and load predictions similar to the CPU-based code and in a substantially reduced amount of time. This capability could allow free-wake based analysis to be used in the possible design and optimization studies of wind farms as well as prediction of multiple turbine flow fields and the investigation of the effects of using different vortex core models, core expansion and stretching models on the turbine rotor interaction problems in multiple turbine wake flow fields.

1. Introduction

Wind turbine placement in a wind farm for power optimization requires specialized tools that can provide accurate results in short times. These problems have not yet been tackled efficiently by conventional Computational Fluid Dynamics (CFD) codes, which require too long of a computational time to offer any feasible attempt to solve wind farm wake interaction and layout optimization problems. The blade element momentum theory (BEM), an alternative, provides an exceptional performance from the standpoint of computational time. However, attainable accuracy restricts its use only to preliminary sizing and performance analysis of wind turbines. The advantages of both methods, i.e. high accuracy and low computational time, are offered by free-wake panel methods.

\(^1\) To whom any correspondence should be addressed.
Free-wake methods provide an efficient means to solve computationally intensive aerodynamics problems relatively fast due to discretization of only the geometry rather than discretization of the entire solution domain as required by conventional CFD methods. Nevertheless, such difficult engineering problems as wind farm optimization can still require weeks of computation to obtain a meaningful solution. Applications of free-wake methodology to wind turbine rotors and wakes can be found in the literature (Gupta and Leishman [1, 2], Hallissy and Chattot [3], Dossing [4], Sezer-Uzol and Uzol [5], Roura et al. [6]). Utilization of the free-wake methodology to multiple turbine interaction problems is very limited. One of the main reasons is due to the uncertainty in vortex core modeling in interaction problems (Uzol and Sezer-Uzol [7]). The other main reason is that solution of turbine interaction problems generally require long run times due to the need for unsteady solution of the entire wake flow fields for many revolutions of each and every turbine present in the domain. These unsteady solutions involve computationally intensive vortex filament interaction calculations, which generally limit the total amount of simulation time or the number of turbines that can be simulated by the method. This problem, however, can be addressed easily using high-level of parallelization. Especially when exploited with a GPU, a Graphics Processing Unit, this property can provide a significant computational speed-up, rendering the most intensive engineering problems realizable in hours of computation time.

This paper presents the results of the simulation of the flow field for the NREL Phase VI turbine using a GPU-based in-house free-wake panel method code. Computational parallelism involved in the free-wake methodology is exploited using a GPU, allowing thousands of similar operations to be performed simultaneously. The results are compared to the experimental data as well as to those obtained by running a corresponding CPU-based code.

2. Free-wake methodology

The solver is an in-house 3-D unsteady vortex-panel method potential-flow solver and is based on a free-vortex-wake methodology. Any number and any combination of fixed and/or rotating wings as well as non-lifting bodies can be included in the solution domain and corresponding interactions and their impact on wing/rotor performances can be analyzed. The suction and pressure surfaces of the blades as well as the tip and root surfaces are discretized using quadrilateral panel elements, and vortex ring elements are placed within each element. The leading edge of each vortex ring is placed on the ¼ chord of each panel, and the collocation point is placed at the ¾ chord of the panel element. Enforcing the surface Dirichlet boundary condition (i.e. the velocity vector in the surface normal direction is set to zero at each and every collocation point) results in a set of linear system of equations that has to be solved at each time step. During an unsteady run, at each time instant, the influence coefficients of surface panels, i.e., the left-hand-side of the equation system, are calculated using the induced velocity values from each vortex segment on the collocation points using Biot-Savart law.

Once the vortex strengths of each vortex ring are obtained as a result of the solution, the blades are moved according to their respective kinematic motion to the next time step, and a row of wake panels are shed from the wing/blade trailing edges with vortex strengths being equal to the difference between vortex strengths of the suction and pressure side panels at the trailing edge, hence satisfying the unsteady Kutta condition at the trailing edge. Various vortex core models are implemented in the code, such as Vatistas, Scully and Oseen-Lamb models. More information on the code can be found in Sezer-Uzol and Uzol [5].
3. GPU Implementation

For the calculations, an NVIDIA GeForce 770 GTX GPU coordinated by an Intel i7 3770 running at 3.40 GHz has been used. The selected GPU shows a peak processing power of 3213 GFLOPS for single precision floating numbers with 1536 CUDA cores available for parallel processing. Geometry transformations, boundary and influence matrix calculations, linear system factorization and solution, free vortex shedding, relaxation steps are all computed on GPU, whereas geometry generation, system coordination and CUDA kernel dispatch algorithms are handled by a single CPU core. Force and moment calculations for the given lifting bodies in the solution space are done on the CPU due to negligible performance requirement. However this part of the algorithm can also be parallelized on the GPU in a straightforward manner. Maximum number of free vortices in the solution space increases with each time step and reaches a maximum number specified by the configuration of the solution space. In order to minimize calculation time during early steps of the simulation, a parameter for defining the number of free vortices that can exist within the solution space is used. Instead of operating on the maximum number of free vortices for a solution for a given timestep, a value for the number of free vortices shed up to the current timestep is calculated on every iteration. With this approach, computation time gradually increases until this value is equal to maximum number of free vortices defined for the solution. CUDA kernel dispatch grid and block dimension parameters are calculated and selected according to the number of bound vortices found within the solution space. In order to get maximum grid and block occupancy from the executed kernels, number of bound and free vortices in the solution space are constrained and aligned to orders of 16 during geometry generation step. This in effect both prevents the threads from spending time calculating on non-zero or undefined solution space objects and allows the software to get maximum throughput during the calculations.

4. Results

The simulations are performed for the two-bladed National Renewable Energy Laboratory (NREL) Phase VI Horizontal Axis Wind Turbine (HAWT) rotor, in accordance with the NREL Unsteady Aerodynamics Experiments (UAE). Blades have the NREL S809 airfoil section from root to tip. Pitch is defined at 75% radius and the pitch axis is at the 30% chord line. A linearly tapered and nonlinearly twisted blade geometry with a rotor radius of 5.029 m and a flat tip is used. The blade geometry has 0° twist at 75% span (and −2° twist at the tip) and a 5° pitch angle. The blade has a root chord of 0.737 m and tip chord of 0.356 m with a taper ratio of 2.1. Figure 1 shows the two-bladed rotor used in the current simulations. D=10.058 m is the rotor diameter. The turbine rotates around positive x direction and is exposed to a 7 m/s freestream wind speed. In all simulations, each one of the rotor blades is discretized using 32 chordwise and 15 spanwise panel segments. The rotor blades are rotating with 72 rpm rotational speed, and the time increment is chosen such that the rotor blades advance 6° at each time step therefore one revolution is completed in 60 time steps. Oseen-Lamb vortex core model is used in the simulations.
Figure 1. NREL Phase VI two-bladed rotor used in the CPU and GPU based simulations

Figure 2. Comparison of wake geometry after 3 revolutions obtained using free-wake methodology. Left column: CPU-based code. Right column: GPU-based code. Simulation times are 3360 s and 33 s for the CPU and GPU codes, respectively.

Figure 2 shows the comparison of the predicted wake geometries by the CPU and GPU codes after three revolutions. The differences in the predicted wake shapes, tip and root vortex trajectories are minimal. One should note that for the results presented in Figure 2, the simulation time for three revolutions took 3360 seconds on the CPU and about 33 seconds on the GPU.

Figure 3 presents the predicted surface pressure coefficient contours on the suction and pressure sides of the blades. Again the differences in the distributions are minimal indicating that similar results to the CPU-based code can be obtained by the GPU code in substantially reduced amount of time.
consistency in the surface pressure distributions in Figure 3 also indicates that the GPU code is not only able to predict the wake shape and geometry but the load predictions can now be obtained much faster compared to a typical CPU-based run.

This is also illustrated in Figure 4 in which quantitative comparisons are presented for the thrust per blade and torque per blade predictions obtained by the CPU and GPU codes. In this figure the results are also compared to the test data obtained in the NREL experiments. As is evident both codes overpredict the thrust and underpredict the actual torque levels. Considering the numerical differences between the CPU and the GPU code predictions, these are most probably due to differences in implementation and optimization of mathematical procedures between CPU’s and GPU’s. For example, the linear system solvers used for matrix inversion calculations are different (Gauss-Jordan in CPU vs. Preconditioner Based System Solver in GPU), blade advance methodologies are different, etc. Keeping in mind that the GPU code is still in the development phase, there might be other bugs in the software causing these differences. Figure 4 bottom row quantitatively shows variations of differences between the CPU and the GPU predictions in thrust and torque using an RMS difference percentage calculated using,

\[
RMS \text{ Difference } \% \text{ in Thrust} = \sqrt{\frac{(\text{Thrust}_{CPU} - \text{Thrust}_{GPU})^2}{\text{Thrust}_{CPU}^2}} \times 100
\]

\[
RMS \text{ Difference } \% \text{ in Torque} = \sqrt{\frac{(\text{Torque}_{CPU} - \text{Torque}_{GPU})^2}{\text{Torque}_{CPU}^2}} \times 100
\]

It can be seen that the maximum RMS differences in thrust and torque predictions are around 0.3% and 5%, respectively, between the GPU and CPU codes. One should note however that, although there are differences between the CPU and the GPU codes, the deviation from the experimental data is much bigger than the differences between the CPU and the GPU codes in thrust and torque predictions (13% overprediction in thrust and 27% underprediction in torque compared to the experimental data).

\[\text{Figure 3. Comparison of surface pressure coefficient distributions after 3 revolutions obtained using free-wake methodology. Left column: CPU-based code. Right column: GPU-based code. Upper row: Blade suction side. Lower row: Blade pressure side.}\]
Table 1 presents the time durations (wall-clock time) for simulating 3, 9 and 27 revolutions of the NREL Phase VI rotor. As is evident, GPU code based results are obtained more than 65 times faster than the CPU code. The simulation times increase as the number of revolutions increase however 27 revolution simulation of the rotor can be obtained in less than an hour whereas it would have taken substantially higher amount of time for the CPU. It is reported previously by using the same CPU-based code, 10 revolutions of data in a serial run took 60 hours on one core of the TR-Grid cluster, which has four AMD Opteron 6172 processors (with 48 cores each; Advanced Micro Devices, Inc., Sunnyvale, CA) with total 128 GB DDR3 memory and two QDR InfiniBand network (80 Gbps) (Sezer-Uzol and Uzol [5]).

Table 1. Comparison of time durations for the simulations of different number of revolutions for the NREL Phase VI rotor using the CPU and GPU-based codes

| No. of Revolutions | Simulation Time-GPU (s) | Simulation Time-CPU (s) |
|--------------------|-------------------------|-------------------------|
| 3                  | 33                      | 2160                    |
| 9                  | 351                     | -                       |
| 27                 | 2495                    | -                       |
Figure 5. Multiple turbine simulation example: Unsteady run for 6 NREL Phase VI rotors for 6 revolutions. Simulation time 2040 s using the GPU-based code.

Figure 6. Comparison of thrust/blade and torque/blade results for the case presented in Figure 5. Upstream turbine is located at x=0 and y=0, and the downstream turbine is located at x=20 y=0.

Figure 5 shows the capability of the GPU-code for simulating multiple wind turbines. This case consists of 6 NREL Phase VI rotors arranged in the shown configuration. Rotors are separated by two diameters in the streamwise and transverse directions. The simulation is performed for six revolutions of the rotors and the total simulation time for this case using the GPU code is 2040 s. Figure 6 shows the impact on thrust and torque predictions when the upstream turbine wakes start to hit the downstream turbines. Reduction in thrust and torque levels after about 3.5 revolutions and related unsteady fluctuations can be observed. Investigating the validity of free-wake based analysis in wind farm interactions problems would require detailed parametric analysis of the modeling of viscous effects, which are usually done through vortex core expansion and vortex stretching models. In our previous publications with the CPU code we have shown that these models can have significant impact on wake and load calculations (Sezer-Uzol and Uzol, [5,7]). However, the main problem in terms of parametrically investigating the impact of viscous effects modeling in free-wake simulations for wind farm interaction problems is that one would need a really fast code due to the need for investigating many different combinations of vortex core models, expansion and stretching models. This was one of our main objectives in developing the GPU code. In the context of the current paper, of course our objective is not to demonstrate the validity of this approach in terms of wind farm simulations or...
viscous wake modeling but it is to demonstrate the fast calculation capability of the GPU solver in multi-turbine configurations so that it could be used in the future for the parametric investigation of viscous wake modeling to be used in wind farm interaction simulations.

5. Conclusions

Results obtained by simulating the flow field of the NREL Phase VI two-bladed rotor using a CPU and GPU based free-wake code are presented. GPU based code is capable of producing wake and load predictions similar to the CPU-based code and in a substantially reduced amount of time. This capability could allow free-wake based analysis to be used in the possible design and optimization studies of wind farms as well as prediction of multiple turbine flow fields and the investigation of the effects of using different vortex core models, core expansion and stretching models on the turbine rotor interaction problems in multiple turbine wake flow fields.

Acknowledgments

We would like to thank all the discussions and insight provided by Assoc. Prof. Dr. Ilkay Yavrucuk of METU Department of Aerospace Engineering and Onur Tarimci of Aerotim Ltd. This study is conducted under a project supported by the Scientific and Technological Research Council of Turkey (TÜBİTAK) under the project number 7120764 of the 1507 Research Support Program for SMEs.

6. References

[1] Gupta, S. and Leishman, J.G., 2006, “Performance Predictions Of NREL Phase VI Combined Experiment Rotor Using A Free-Vortex Wake Model,” AIAA-2006-390, 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, Jan. 9-12.
[2] Gupta, S., and Leishman, J., 2005, “Comparison of Momentum and Vortex Methods For the Aerodynamic Analysis of Wind Turbines” AIAA 2005-594, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 10-13.
[3] Hallissy, J., and Chattot, J., 2005, “Validation of Helicoidal Vortex Model with the NREL Unsteady Aerodynamic Experiment,” AIAA 2005-1454, 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, January 10-13.
[4] Dossing, M., “Vortex Lattice Modelling of Winglets on Wind Turbine Blades,” Risø-R-1621(EN) August 2007.
[5] Sezer-Uzol, N., Uzol, O., 2013, “Effect of Steady and Transient Wind Shear on the Wake Structure and Performance of a Horizontal Axis Wind Turbine Rotor,” Wind Energy, V. 16, pp. 1-17.
[6] Roura, M., Cuerva, A., Sanz-Andres, A., Barrero-Gil, A., 2009, “A Panel Method Free Wake Code for Aeroelastic Rotor Predictions,” Wind Energy.
[7] Sezer-Uzol, N., Uzol, O., 2012, “Free-wake Calculations of Wind Turbine Wake-Rotor Interactions: The Effect of Vortex Filament Core Size, Expansion and Stretching on Performance Predictions,” Science of Making Torque from Wind, Oldenburg, Germany, 9-11 October.