UCF

Stands For Opportunity

Department of Mechanical, Materials and Aerospace Engineering

Center for Advanced Turbines and Energy Research (CATER)

Laboratory for Turbine Aerodynamics, Heat Transfer and Durability

CFD Analysis of a Traffic-Driven Wind Generator (TDWG) System

Final Report

Principal Investigator (P.I.): Dr. Son Ho (UCF)

Co-P.I.: Prof. Dr. Jay Kapat (UCF)

Research Assistant: Perry Johnson (UCF)

Sponsor: Theodore Wiegel (Wind Age, LLC)

March 2011

Executive Summary

This study performed a Computational Fluid Dynamics (CFD) analysis in conjunction with solid body kinematics to predict the transient output of a helical-blade vertical axis wind turbine when exposed in close proximity to passing traffic on an interstate highway. The CFD software package Fluent (Ansys, Inc.) was used to solve the unsteady Reynolds-Averaged Navier-Stokes (RANS) equations for the fluid domain. The domain consisted of three regions: a stationary surroundings region, a rotating region encompassing the turbine, and a translating region encompassing the passing vehicle. A parametric study was carried out to determine the effects of initial rotational velocity of the turbine, velocity of the passing vehicle, clearance between the turbine and passing vehicle, placement of false numerical boundaries, and the addition of shrouding to the turbine. A brief extension of the truck-driven methodology was made to a high-speed rail-driven scenario.

First, the CFD procedure was validated for a typical Savonius wind turbine in steady wind by comparison with experimental data. Then the twisted helical turbine was simulated in steady wind and compared against its manufacturer specifications to determine the friction. The specifications were also used to model the electrical generator, which was important in the transient simulations. The CFD results were used as input forces for a solid body kinematics simulation, which was coded using the C programming language. Various python scripts were used for the manipulation and control of data.

The output of the baseline case was around 0.1 W-hr per truck per turbine with the truck passing at 70 mph and a clearance of one foot. This value was enhanced when the turbine was moved closer to the truck and the truck was given a higher velocity. A lower output was achieved when the turbine was moved further away, the truck approached at a lower velocity, or multiple turbines were included in close proximity, and any shrouding techniques were added.

The study was extended to include an application with a turbine on the side of a passing High Speed Rail (HSR) with a velocity of 175 mph. The output for the HSR simulations was between 0.2-0.5 W-hr per train per turbine. It is expected that this output would be similar for a turbine placed above the HSR. In this way energy could be captured from multiple directions coming from the high speed rail.

Terminology

CFD	Computational fluid dynamics: general term for the numerical modeling of fluid flows
discretization scheme	Method of approximating a continuous partial differential equation (PDE) as a discrete algebraic set of equations that can be solved iteratively
domain	The extent of space computed in the mathematical model
grid	The decomposition of the domain into discrete control volumes at which the transport equations are solved algebraically.
HAWT	Horizontal-axis wind turbine
incompressible	Having constant density, a fair assumption for low Mach number flows
interface	A numerical boundary condition indicating the boundary of two domains that are moving with respect to each other, requiring interpolation between control surfaces at the boundary, thus diminishing the spatial resolution of the discretization scheme.
k-ω SST	A two-equation turbulence model achieving mathematical closure for the RANS equation, known to be excellent for aerodynamic flows.
moment	Used synonymously with 'torque' indicating a rotational force on the turbine
pressure inlet	A numerical boundary condition at which a pressure is specified and no velocity is specified.
radians (rad)	Measurement of angular position: 6.28 radians = 1 rotation
RANS	Reynolds-Averaged Navier-Stokes: time-averaged form of the Navier-Stokes equations (governing equations for fluid flows) used to facilitate realistic numerical solution requirements; require a turbulence model for closure due to the effects of turbulent motions.
transport equation	A partial differential equation (PDE) governing the motion of a particular quantity (mass, momentum, energy, etc…) throughout a fluid flow; governed by Reynolds Transport Theorem.
VAWT	Vertical-axis wind turbine
Velocity inlet/outlet	A numerical boundary condition at which the velocity is specified and no pressure is specified.
wall	A numerical boundary condition indicated that the velocity components at the boundary are zero and special turbulent modeling is applied.

Introduction

Wind turbines generate electrical power by converting the kinetic energy of moving atmospheric air into rotational energy for turning an electrical generator. The dynamic pressure of the air is used to create a moment which acts to turn the turbine blades. There are two general classifications of wind turbines: Horizontal Axis Wind Turbines (HAWTs) and Vertical Axis Wind Turbines (VAWTs). Currently, HAWTs are more commonly used due to their higher performance. Certain circumstances and geometric constraints may call for use of VAWTs. The present investigation concerns the performance testing of two general VAWTs geometries using Computational Fluid Dynamics (CFD).

Because of the high upfront cost of laboratory wind tunnel testing of VAWT configurations, preliminary numerical simulations serve to predict the performance of given geometries by solving the governing equations of fluid motion under certain simplifying assumptions [1]. The numerical model allows for relatively quick data acquisition and engineering estimation of turbine performance.

A particularly common VAWT design is the Savonius turbine. The Savonius wind turbine geometry consists of two or three semi-cylindrical blades with constant radius of curvature. Because of the simplicity of the geometry, the Savonius wind turbine provides an excellent means of validating the current numerical procedure, as outlined below. Once the procedure is shown to match experimental results satisfactorily, more complicated schemes can be simulated with confidence. Blackwell et al. [2] report experimental data for a large number of Savonius wind turbine tests performed at Sandia National Laboratories. Figure 1 shows a typical Savonius VAWT.
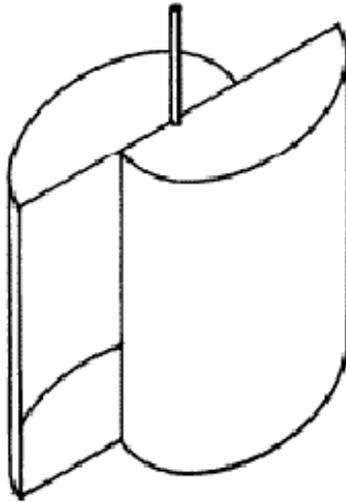
Figure 1 Typical Savonius VAWT[2].

A typical helical VAWT is shown in Figure 2. The helical shape of the three blades creates an inherently three-dimensional geometry, as opposed to the two-dimensional nature of the Savonius. This model VAWT is 4 feet in diameter and 12 feet long. This project uses this geometry as the basis for the simulations.

Figure 2 Reference helical VAWT.

Nomenclature

a	Radius of curvature for semi-circular Savonius cups [m]
A	Projected area of the turbine in the stream-wise direction [m^2]
ΔA	Differential area along blade surface
C_M	Moment coefficient
C_P	Power coefficient
D	Turbine diameter (=2*R) [m]
J	Moment of inertia
p	pressure
P	Power
M	Moment (or torque)
r	Position with respect to the center of rotation
R	Turbine radius [m]
t	Time [sec]
Δt	Time step of simulations [sec]
T	Total time of simulation [sec]
U_∞	Freestream approach wind velocity
ε	Angular acceleration
θ	Angular position
λ	Tip speed ratio (= $\omega R/U_\infty$)
ρ	Fluid density
ω	Angular velocity, specific dissipation rate

Scope

The purpose of this study was to use CFD to predict the performance of a helical wind turbine in capturing energy from a truck passing underneath a highway overpass and provide preliminary performance regarding a High Speed Rail (HSR) application. In addition, the study aimed to investigate the sensitivity of the performance to relevant variables. A reference geometry of a helical VAWT was utilized in this study. The specifications shown in Figure 3 were used to establish realistic inputs for the numerical simulations. Note that there is an error in the rated wind speed conversion from mph to m/s: 31 mph is equivalent to 14 m/s.
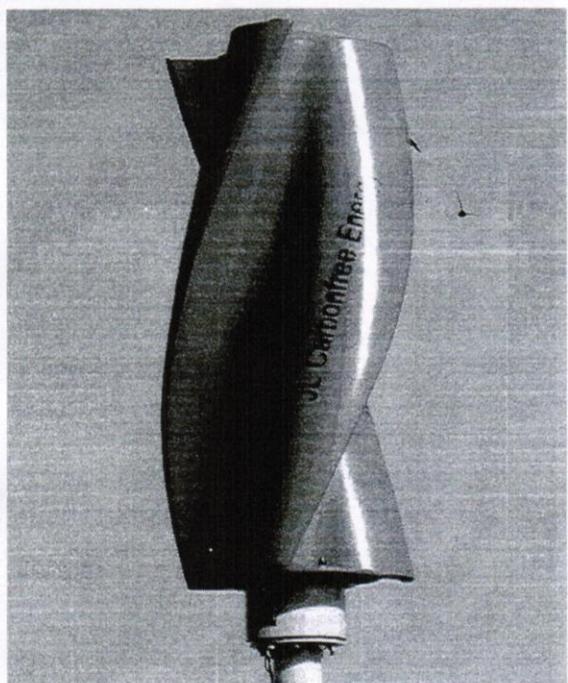
Parameter	
Rated power	1.5KW
Max power	1.7KW
Output voltage	AC:220/110 DC:48V
Rated wind speed	31mph(12m/sec)
Rotor diameter	4Ft(1.2m)
Blade Length	12Ft(3.7m)
Blade material	PE
Start-up Wind Speed	4.5mph(2m/sec)
Cut-in wind speed	6.3mph(2.8m/sec)
Work wind speed	2-35m/s
Rotation speed	110r/min
Output/Year	2420kWh@12mph(5.4m/s)
Number of blades	1
Turbine weight	130kg
Tower height	6m
Package Size (m)	4.2*1.4*1.4m
Noise level	< 20dB

Figure 3 The specifications provided for the reference helical VAWT.

The scope of the present study was divided into a sequence of tasks with differing procedures and features. There were two different main types of procedures used. First, a Standalone procedure was used to evaluate the performance of various schemes under steady operation and steady wind. This procedure was developed in a previous report [1] sponsored by the Space Alliance Technology Outreach Program (SATOP). Table 1 shows the two tasks, labeled '0a' and '0b', completed in this preliminary study. First, the current CFD methodology was validated using the Standalone procedure (Task 0a) on a well-documented Savonius wind turbine [2]. The helical turbine was then simulated (Task 0b) and compared against the specifications in Figure 1.

Table 1 Scope of SATOP-sponsored report.

Task	Description
0a	Standalone Savonius turbine
0b	Standalone helical turbine

The second type of procedure was the simulation of a truck passing underneath a horizontally mounted helical turbine. This procedure was inherently transient and included motion of both the turbine rotor and the passing truck. Three of the four tasks in the current investigation used this truck-passing procedure, as outlined in Table 2. First, the bare turbine (from standalone Task 0b) was simulated with the truck passing in Task 1. Then, Task 2 involved the inclusion of a shroud in an attempt to focus the incoming wind from the truck and enhance the output of the turbine. Task 3 provided a geometric variation in which a hood was used around the turbine rather than a shroud. Finally, Task 4 extended the current methodology for predicting a HSR application.

Table 2 Scope of current study.

Task	Description
1	Bare helical turbine with truck passing
2	Shrouded helical turbine with truck passing (Figs. 70, 76)
3	Hooded helical turbine with truck passing (Fig. 82)
4	Bare helical turbine with high-speed rail passing

The majority of the project focused on Task 1, studying the effect of a variety of parameters on the performance of a bare turbine. These parameters included: the effect of artificial boundary conditions necessary for numerical analysis; the clearance distance between the truck's upper surface and the rotor; the velocity of the oncoming truck; the benefit of initial rotational velocity; the decrease in performance in the case of multiple turbine interaction; extension to a high-speed rail scenario. Task 2 was briefly studied with respect to the extent of the shroud and Task 3 comprised only one case.

Standalone Procedure

Two types of simulations were performed for the standalone procedure, which is flow-charted in Figure 4. First, a static test was run with the blades in fixed positions ($\omega = 0$) to determine the static torque loading on the turbine, which can be interpreted as the torque available for start-up. The static moments are evaluated at different fixed angles at which the turbine can lie with respect to the incoming wind. This simulation provided validation against experimental static tests for Task 0a. The utility of this simulation for Tasks 0b was the determination of the friction opposing the turbine rotation. Secondly, dynamic simulations were run in which the turbine geometry was rotated using a sliding mesh technique. With steady inflowing wind velocity used as the inlet boundary condition, the performance of the turbine was

evaluated over a range of tip speed ratios. The friction estimation from the static simulation was used in the dynamic simulation to correct for frictional losses.
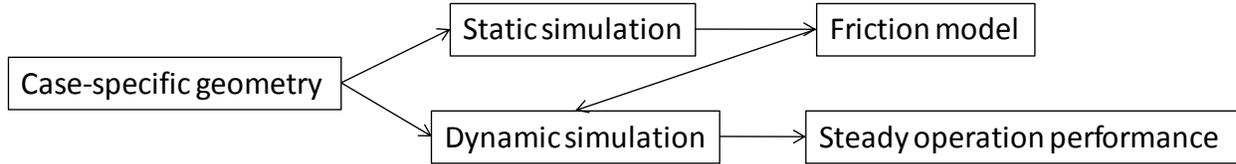

Figure 4 Flow-chart outlining Standalone procedure.

For both static and dynamic CFD simulations, Ansys Fluent 12 software [6] is used to solve the governing equations in the stationary and rotating domains. The unsteady incompressible two-dimensional RANS equations for mass and momentum conservation, Eqs. 1a and 1b respectively, were solved alongside the two turbulence model equations for the turbulent kinetic energy and specific dissipation rate. The Shear Stress Transport (SST) version of the k-ω model [4] was used due to its suitability for aerodynamic wall-bounded flows. The governing equations were discretized using second-order upwind differencing on the non-linear convective terms and second-order central differencing on the diffusive terms. Time was discretized with a fully-implicit second-order scheme. The segregated pressure-based SIMPLE algorithm was used for pressure-velocity coupling. The solution was iterated within each time step until convergence was achieved. Convergence for the Standalone cases was defined as normalized residuals of 1e-5 for the continuity and turbulence equations and 1e-6 for the momentum equations.

Because of the turbulent nature of the fluid flows at high Reynolds numbers, a procedure of modeling the Reynolds stresses is needed, leading to the well known closure problem in turbulent flows [3]. The Boussinesq hypothesis, Eq. 2, is used to model the effects of turbulence as an 'eddy viscosity'. The k-ω SST [4] model is used achieve closure. This model uses two transport equations in addition to the RANS equations to solve for the turbulent kinetic energy (k $[m^2/s^2]$), Eq. 3, and the specific dissipation rate (ω $[s^{-1}]$), Eq. 4, where ω is the ratio of turbulent dissipation rate (ε $[m^2/s^3]$) to the turbulent kinetic energy (k $[m^2/s^2]$). The turbulent viscosity, μ_t, is a function of k and ω.   In all, a total of five transport equations are solved throughout the domain.

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i}(\rho u_i) = 0$$

(1a)

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial u_l}{\partial x_l}\right)\right] + \frac{\partial}{\partial x_j}(-\rho\overline{u_i'u_j'})$$

(1b)

$$-\rho\overline{u'_i u'_j} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \left(\rho k + \mu_t \frac{\partial u_k}{\partial x_k} \right) \delta_{ij} \tag{2}$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_i}(\rho k u_i) = \frac{\partial}{\partial x_j}\left(\Gamma_k \frac{\partial k}{\partial x_j} \right) + \tilde{G}_k - Y_k + S_k \tag{3}$$

$$\frac{\partial}{\partial t}(\rho\omega) + \frac{\partial}{\partial x_i}(\rho\omega u_i) = \frac{\partial}{\partial x_j}\left(\Gamma_\omega \frac{\partial\omega}{\partial x_j} \right) + G_\omega - Y_\omega + D_\omega + S_\omega \tag{4}$$

The solution to the five transport equations above results in the two velocity components, the pressure, and the two turbulence parameters at each cell.

For all simulations, the moment (or torque) is calculated as the sum over the surface edges of the cross products of the coordinate locations with respect to the origin and the local pressure force:

$$M(t) = \Sigma_i \left[\vec{r_i}(t) \times \left(p_i(t) \right) (\Delta\vec{A_i}) \right] \tag{5}$$

Using the specified scales and the moment coefficients reported by the Fluent solver [6], the moment acting on the turbine due to the wind can be calculated for any time (t). The time-averaged moments are taken for comparison with the data from [1]. The average moments are calculated by:

$$\overline{M} = \frac{\int_{t_0}^{t_0+T} M(t)dt}{T} \tag{6}$$

The performance of a rotational wind turbine is commonly evaluated using the torque, (C_M) which is defined as:

$$C_M = \frac{M_{wind}}{\frac{1}{2}\rho U_\infty^2 AR} \tag{7}$$

The area (A) is the projected area in the streamwise direction and is equal to the product of the diameter of the turbine and its height. The length scale used to non-dimensionalize the moment is taken to be the maximum moment arm of the flowing fluid, which is the turbine radius (R), or half the diameter (D). The freestream density (ρ) and velocity (U_∞) are used in this formulation.

The rotational speed is typically non-dimensionalized using the tip speed ratio:

$$\lambda = \frac{\omega R}{U_\infty} \tag{8}$$

The power resulting from the moment (or torque) passed to the generator at a given rotational speed is given by:

$$P = M_{generator}\,\omega \tag{9}$$

The power delivered to the generator can be non-dimensionalized by the power coefficient:

$$C_p = \frac{P}{\frac{1}{2}\rho U_\infty^{\ 3} A} \tag{10}$$

Comparing the power and moment coefficients, it can be shown:

$$C_p = \lambda\, C_M \tag{11}$$

The quantity of interest is the torque generated by the flow field on the turbine surface. Applying the conservation of angular momentum to the turbine, the rotational motion is described by:

$$\sum M = M_{wind} - M_{friction} - M_{generator} = J\varepsilon \tag{12}$$

Under steady operation, Eq. 12 becomes:

$$\sum M = M_{wind} - M_{friction} - M_{generator} = 0 \tag{13}$$

Before the generator starts up, Eq. 13 can be further simplified to:

$$\sum M = M_{wind} - M_{friction} = 0 \tag{14}$$

Because of this reasoning, a static simulation (no rotation) can be run to quantify startup torque ($M_{friction}$) for a given wind velocity using Equation 14. Using this value, Equation 13 is used for a simulation with rotation. The torque passed to the generator ($M_{generator}$) is proportional to the power generated by the turbine.

After initialization of the static simulation, steady calculations were performed with the turbine in a fixed position. Because a vortex-shedding phenomenon is observed, the torque experienced by the turbine at a fixed position oscillates at the frequency of the shedding. An unsteady simulation was performed, initialized by the steady (but oscillating) solution. The unsteady simulation was run until convergence of the time-averaged torques for each static position before advancing to the next position. The static test data from [2] allows for direct comparison of the Task 0a simulation with experimental data. The Task 0b static simulation allows for the calculation of friction using data from Figure 3.

The dynamic simulation is then performed by applying a fixed angular velocity to the rotating sub-domain. The unsteady solution is temporally advanced with a fixed time step (Δt) for a total time (T) long enough for time-averaged torque coefficients to converge. This is repeated for a range of angular velocities.

Validation is available for both moment and power coefficients as a function of tip speed ratio (λ). Tip speed ratio is varied from 0.2 to 2.0 as done in the experiments of [2].

Truck-Passing Procedure

The second type of procedure used included the transient effects of the passing truck. This inclusion introduced further complication into the modeling procedure. The procedure used for Tasks 1-4 in the current study is flow-charted in Figure 5. The final simulation is a rigid body kinematics code using the equations of motion for a rotating mass (see previous section). The various torques acting on the turbine are inputs to this simulation, which outputs the details of the rotor motion and power output. The main input torque of the wind is calculated using a surface interpolation technique which draws data from a database produced by a number of individual CFD simulations. These CFD simulations represent the wind forces on the turbine at a given angular velocity and rotor position. The rigid body simulator, for a given point in the progression with certain angular velocity, rotor position, and position of the truck, reads the nearest neighboring values from the database and interpolates to determine the input wind torque at that time step. The friction is determined as in the standalone procedure above. The generator model determines the opposing torque of the generator on the rotor as it extracts mechanical power and converts it into electrical power. The model assumes that the generator torque varies linearly with angular velocity with known points at zero angular velocity and the angular velocity and power output from the specifications in Figure 3.
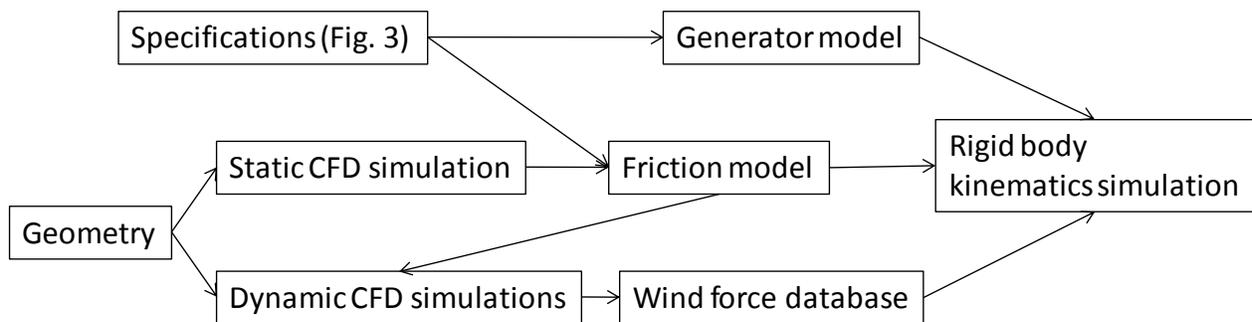
Figure 5 Flow chart outlining the truck-passing simulation procedure.

Dynamic CFD simulations

Ansys Fluent 12 [6] is used to solve the governing equations in the stationary and rotating domains. The unsteady incompressible two-dimensional RANS equations for mass and momentum conservation were solved alongside the two turbulence model equations for the turbulent kinetic energy and specific dissipation rate. The Shear Stress Transport (SST) version

of the k-ω model [4] was used due to its suitability for aerodynamic wall-bounded flows. The governing equations were discretized using second-order upwind differencing on the non-linear convective terms and second-order central differencing on the diffusive terms. Time was discretized with a fully-implicit, second-order scheme. The segregated, pressure-based SIMPLE algorithm was used for pressure-velocity coupling. The solution was iterated within each time step until convergence was achieved. Convergence for the standalone cases was defined as normalized residuals of 1e-5 for the continuity and turbulence equations, and 1e-6 for the momentum equations.

The domains of these simulations are comprised of three sections. First, the main part of the domain includes the task-specific geometry of the surroundings away from the turbine and truck. Secondly, a circular domain surrounding the turbine is inserted into the task-specific domain and allowed to rotate at a given angular velocity. Finally, a rectangular domain around the truck is taken, which translates with respect to the other two domains at the speed of the oncoming truck. The task-specific domain and mesh are different for each case, while the truck and turbine surrounding domains are nearly constant for all tasks. Only minor changes were made to these two domains to accommodate changes in clearance height or geometry change (shrouding/hooding).

A simulation of just the truck geometry traveling at the given truck velocity is used to initialize the domain. The truck starts in a position far enough away that the turbine does not yet feel any effects from the truck. As time progresses, the truck translates past the turbine as it rotates and pressure data is taken at the surface of the truck and turbine. The pressure distribution of the wall is used to calculate the torque on the turbine at each time step:

$$M(t) = \sum_i \left[\vec{r_i}(t) \times \left(p_i(t) \right) (\Delta \vec{A_i}) \right]$$

The calculated moments at each time step are stored in a database for use by the interpolating kinematics simulation. The typical solution time for one CFD simulation was about three days on two nodes of the Ariel cluster using the Ansys Fluent software [6].

Friction and Generator Models

The frictional torque from the equivalent Standalone simulation (Task 0b) is used for Tasks 1-4. The generator is modeled as providing an opposing torque that varies linearly with the rotational velocity of the turbine. The linear relationship is fit through the datum point (zero force at zero rotational velocity) and the design point specified in Figure 3. The generator provides zero torque underneath the kick-in velocity specified.

Kinematic Simulations

The actual kinematic simulation is left to the last step in order that a quick parametric look at the variability of the turbine dynamics with respect to initial conditions, friction and

generator models, and rotor inertia can be studied in a time-efficient manner. The kinematic code, as described below, executes in about one second on a typical desktop computer.

The kinematic solver was coded for the specific problem at hand using the C computer language. The solver code with its typical output is included in Appendix A, and auxiliary python scripts are included in Appendix B. The inputs to the solver include the database of wind forces on the turbine, initial conditions, and various models and properties of the turbine. The solver is a basic rigid body solver using the angular momentum conservation principle, which is given by Equation 12.

From a given initial angle and rotational velocity, the solver steps through time with the step size of the CFD solution. This ensures that the wind torques from the database align with the present simulation in time. At each time step, the solver uses a first-order accurate interpolation with respect to angular position and rotational velocity to find an appropriate wind torque. Next in the time step, the solver enters an iteration loop in which the new rotational velocity is found for the next time step. The generator torque is dependent on this new angular velocity calculation, and thus an iterative process is used. Once the new rotational velocity and torques have all been determined, the new angular position of the turbine is found by multiplying the new angular velocity by the size of the time step. The energy produced by the generator at the time step is calculated by multiplying the generator torque by the rotational velocity and the size of the time step. The data from the time step can be output to a file or the screen before moving on to the next time step and repeating the above procedure. At the end of the solution, the total energy produced is reported in Watt-hours per truck-turbine.

Task 0

The first efforts in this ongoing project were completed under the SATOP program. The data resulting from this initial study is reported here as 'Task 0' because this data is fundamental to the presently reported data.

Case 0a (standalone Savonius VAWT)

The geometry of the Savonius turbine itself was constructed to duplicate the wind tunnel models from the Sandia National Laboratory report [2]. The turbine consists of two semicircular cups with a prescribed radius a=0.25m given a spacing of s/D = 0.1. A small thickness is given to the blades to match the geometries of the Sandia report [2]. Figure 6 illustrates the geometric features of the Savonius turbine.

A two-dimensional structured mesh is generated using Gambit software [5], with this geometry embedded in the computational domain. The grid was generated by mapping quadrilateral cells in logical rectangles. The final grid cell count was around 35,000. Discretization was achieved using second-order upwind schemes for the non-linear convective terms of the governing equations with second-order central differencing techniques for the

diffusive terms. The overall grid is shown in Figure 7 and a close-up of the region near the turbine is shown in Figure 8.
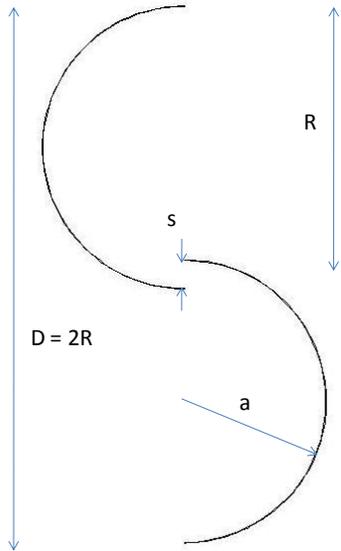
Figure 6 Geometry of the Savonius rotor.
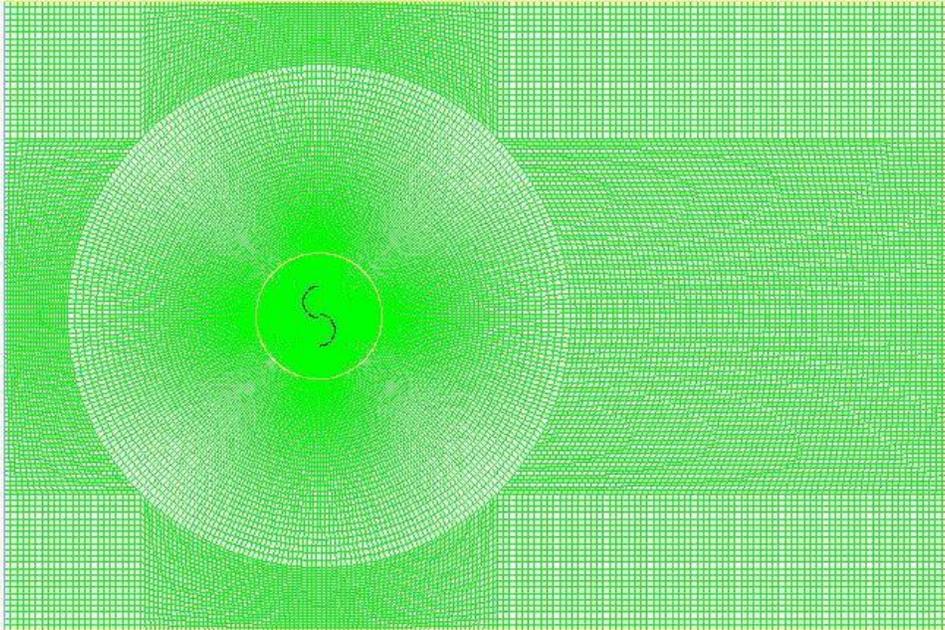
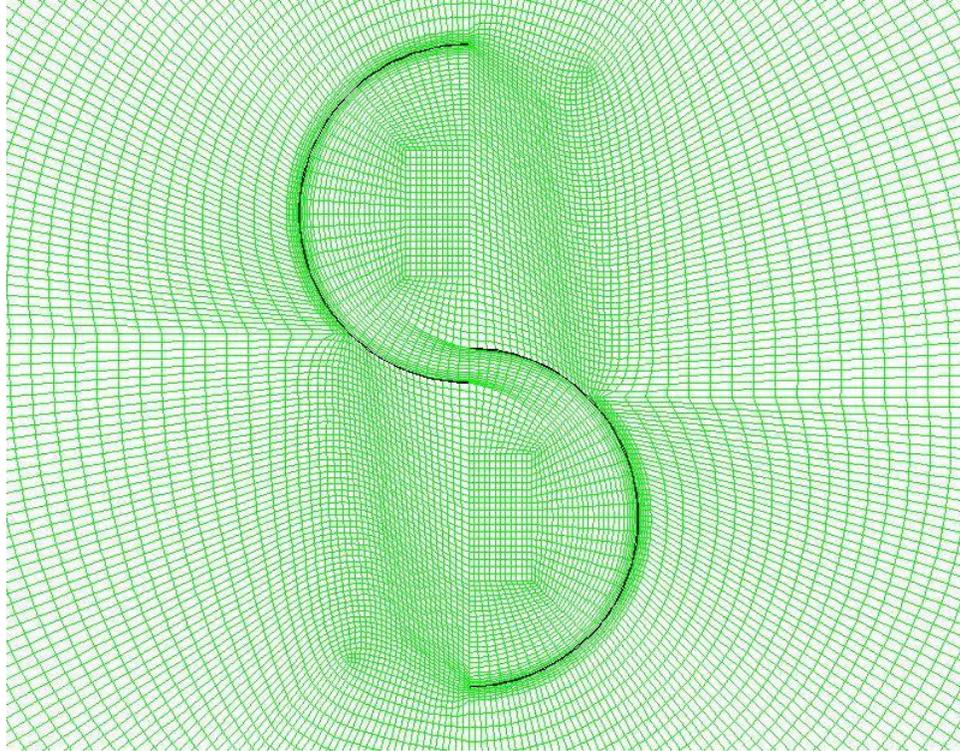**Figure 7 Structured grid for Savonius simulations.**

Figure 8 Grid in the near-region of the Savonius rotor.

Two types of simulations were performed for each geometry. First, a static test was run with the blades in fixed positions (ω = 0) to determine the static torque loading on the turbine, which can be interpreted as the torque available for start-up, that is, to overcome the friction in beginning the rotation of the turbine from a stationary state. The static moments are evaluated at different fixed angles at which the turbine can lie with respect to the incoming wind. Starting at the position shown in Figures 6 and 7, the turbine is rotated in increments of 20° for a total of ten static tests. The origin (0°) is taken to be the initial position. This is identical to the definition in the Sandia report [2].

Ansys Fluent 12 [6] is used to solve the governing equations in the stationary and rotating domain. The unsteady incompressible RANS equations were used in conjunction with Shear Stress Transport (SST) formulation of the k-ω turbulence model [4]. After first initialization, steady calculations were performed with the turbine fixed position. Because a vortex-shedding phenomenon was observed, the torque experienced by the turbine at a fixed position oscillates at the frequency of the shedding. An unsteady simulation was performed, initialized by the steady (but oscillating) solution. The unsteady simulation was run until convergence of the time-averaged moments for each static position, before advancing to the next position. The static test data from [1] allows for direct comparison of the current two-dimensional simulation with experimental data.

A sample instantaneous velocity contour plot for θ = 0° is shown in Figure 9. The accompanying pressure contours are shown in Figure 10. For this angular position, the high pressure region (low velocity region) is concentrated mostly on the lower half of the turbine, generating a positive moment. On the upper half, the convex surface smoothly directs the flow around the blade, similar to a cylinder in crossflow. As the flow accelerates around this curve, a low pressure region is observed as expected. The stagnation region on the convex upper blade is shifted down from the leading edge as the flow seeks the path of least resistance (greater flow resistance around the lower half of the blade than the upper).
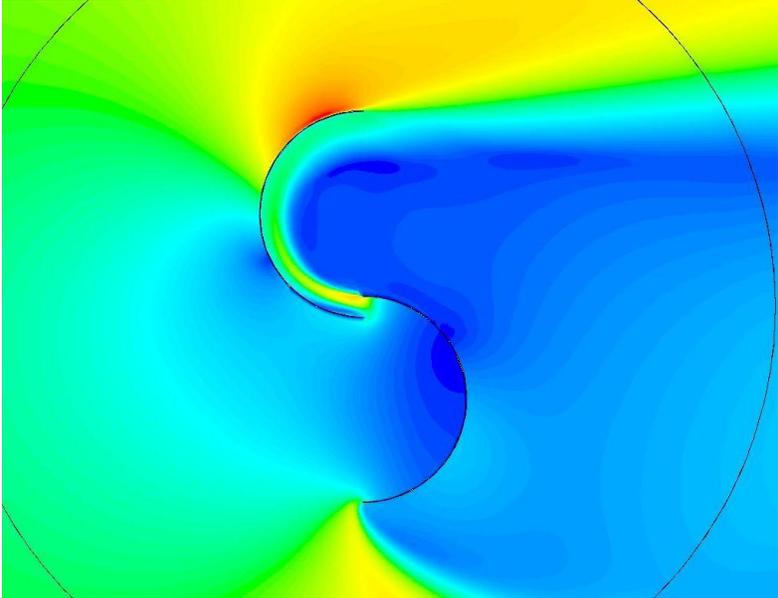
Figure 9 Instantaneous velocity magnitude contours for static test.
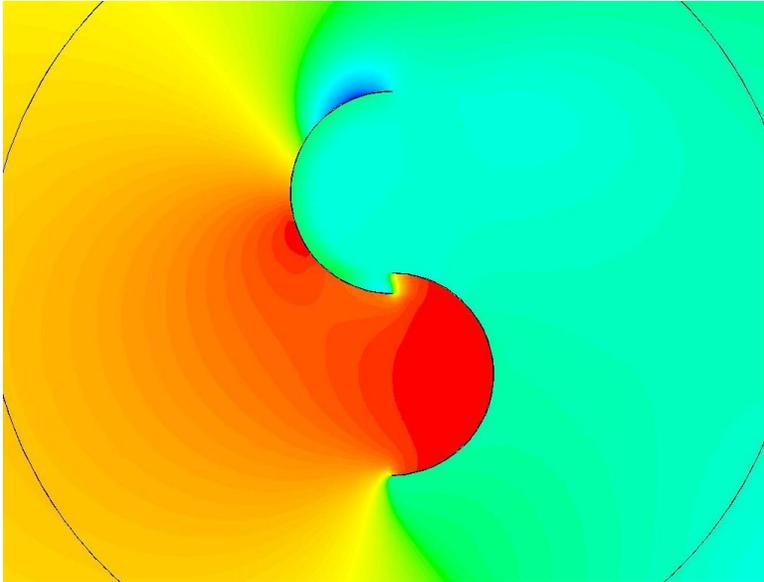
Figure 10 Instantaneous pressure contours for static test.

16

Two static simulations were performed with two different physical time steps. Recall that the static simulations were unsteady calculations due to the shedding phenomenon observed. Figure 11 compares the numerical results of the Savonius simulation with the experimental finding of [1]. The difference between the time step sizes, which vary by multiple orders of magnitude, is very small aside from the first three results. It is hypothesized that the transient effects were not fully eradicated from the large time step simulation for the first few positions. This is further supported by the difference between the 0° and 180° moments in the larger time step simulation. It is apparent that the larger time step converges to agreement with the simulation data from the finer time step after the first few data points. The size of the time step for the static solutions, then, is seen to have small influence on the moment results, given that the solution is given time to become stationary.

The numerical results compare well to the experimental results regarding the trend of the moment coefficient with respect to the angular position. There is noticeable over-prediction at all positions. The current simulation assumes a two-dimensional flow, which is not truly an accurate reproduction of the original experiment. The 2-D assumption essentially simulates an infinite-span turbine, which experiences no losses due to the ends of the blades. For this reason, the over-prediction of the moments can be explained.
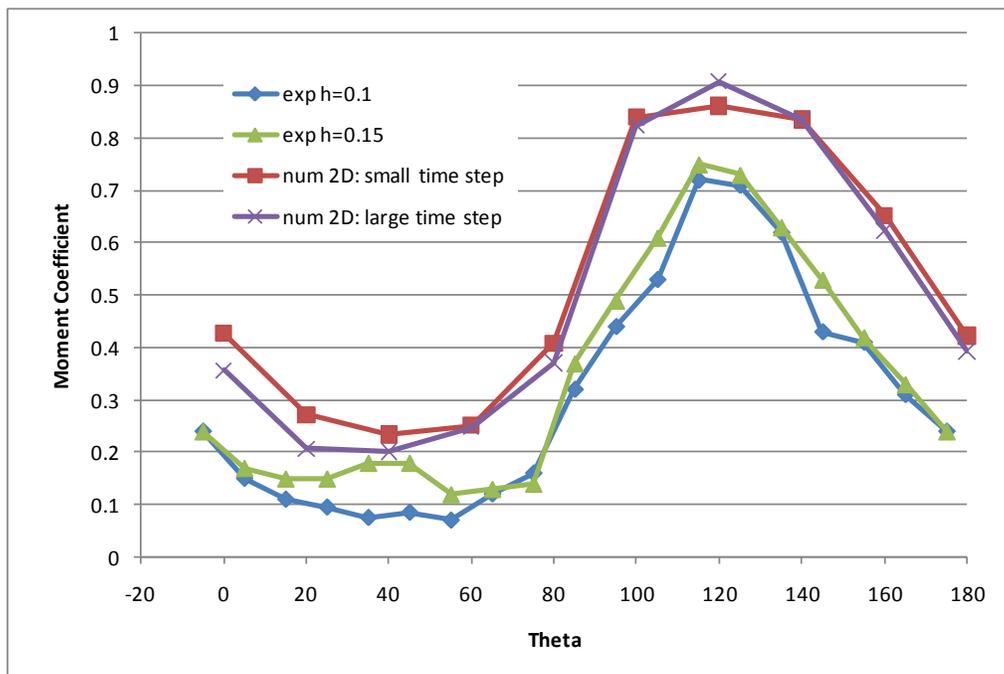
Figure 11 Savonius static test validation for two time step sizes.

Dynamic testing is then performed by applying a fixed angular velocity to the rotating sub-domain. The unsteady solution is time advanced with a given time step (Δt) for a total time (T) long enough for time-averaged moment coefficients to converge.

Validation is available for both moment and power coefficients as a function of tip speed ratio (λ). Tip speed ratio for the Savonius rotor simulation is varied from 0.2 to 2.0 as done in the experiments of [2]. For the helical rotor simulation, λ is varied from 0.2 o 1.8. The stopping point of the simulation is the 'run-away' point. This point is defined where the torque on the blade opposes the direction of rotation.
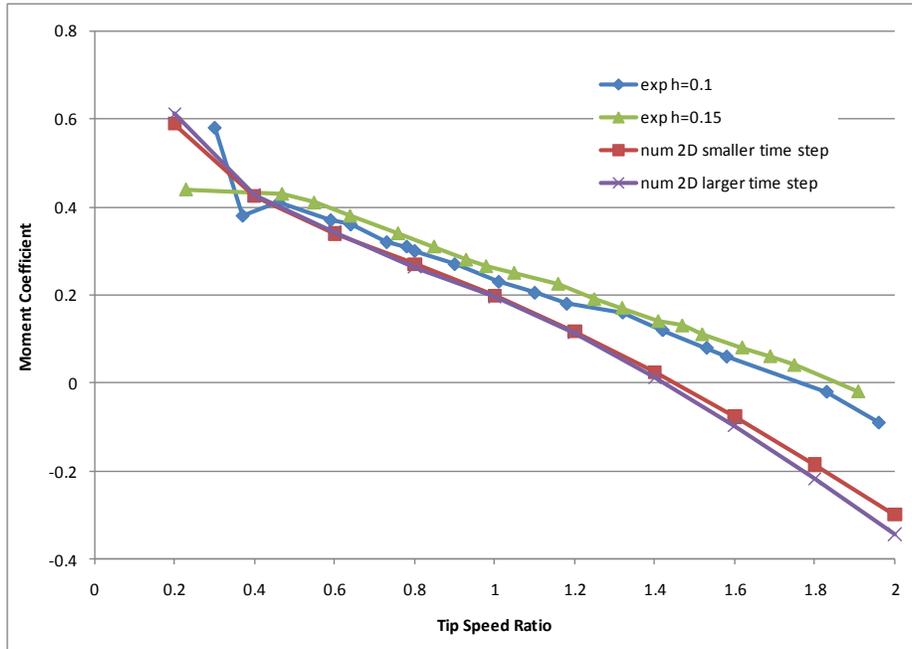
Figure 12 Moment coefficient validation for dynamic test for simulation data.
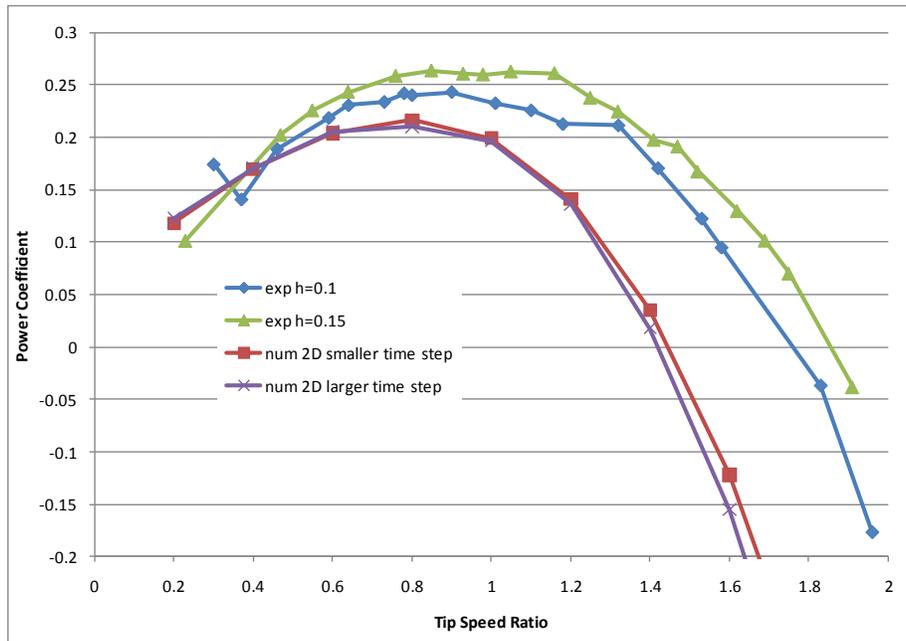
Figure 13 Power coefficient validations for dynamic tests.

Similarly, two dynamic simulations were run with a larger and smaller time step. The resulting moment coefficients are plotted in Figure 12 against tip speed ratio from 0.2 to 2.0. It is seen that the numerical results slightly under-predict the experimental results at low tip speed ratios. This under-prediction becomes exaggerated at high tip speed ratios. This trend is repeated in Figure 13, which shows the power coefficient as a function of tip speed ratio.

Case 0b (standalone helical VAWT)

For the helical-blade turbine, the twisted nature allows for each of the two-dimensional profiles to coexist at any position of the blade with respect to wind direction. In this manner, the torque produced by the wind at any given time is equal to the average of all positions of the two-dimensional profile.

The geometry of the helical turbine is more complex than the simple semi-circles of the Savonius configuration. To approximate a cross-section of the helical turbine, a piecewise fit of arcs and fillets was used based on the non-technical visual data available. The rotor diameter of the blade is 1.2 m, so that the maximum radius for the torque is 0.6 m. The geometry for the helical VAWT simulation is shown in Figure 14.
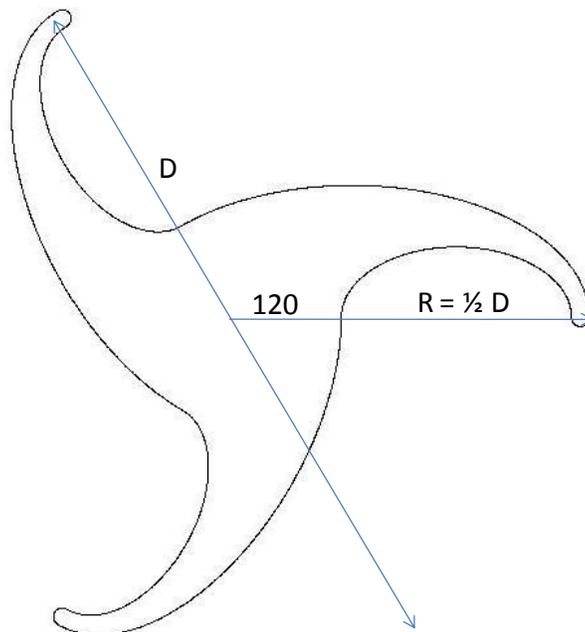
Figure 14 Geometry of the helical wind turbine.

The two-dimensional structured grid was carefully constructed using the Gambit [5] meshing software. Figure 15 displays the overall structure of the grid for the entire domain of the helical blade (Case 0b). Figures 16 shows the grid structure in the immediate vicinity of the blades, where the mesh is the finest. The grid totaled around 85,000 cells. A grid of nearly 300,000 cells (not shown) was also created for the helical turbine geometry to test grid convergence. This finer grid had the same basic structure as the 85,000-cell grid.

19

Qualitative results are shown in Figures 17-19. Figure 17 shows the pressure distribution around the blade, with its accompanying velocity distribution in Figure 18. Two distinct stagnation regions are barely seen, no nearly as distinguished as in the Savonius results above. Again, the pressure distribution lends itself to a counter-clockwise (positive) torque on the turbine. Figure 19 shows the pressure contours in the wake of the turbine, clearly outlining the vortex shedding pattern present, creating an oscillatory moment, even in the static cases.
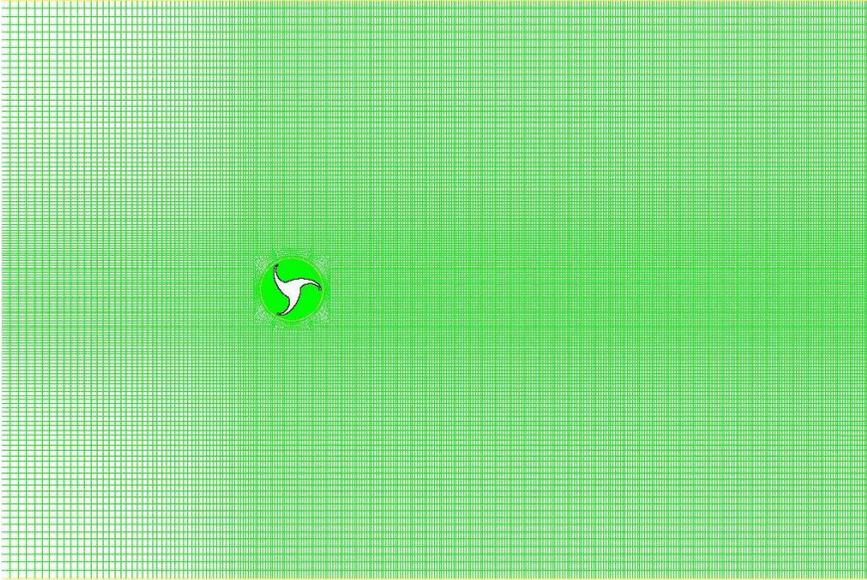
Figure 15 Quadrilateral grid for helical turbine simulations.
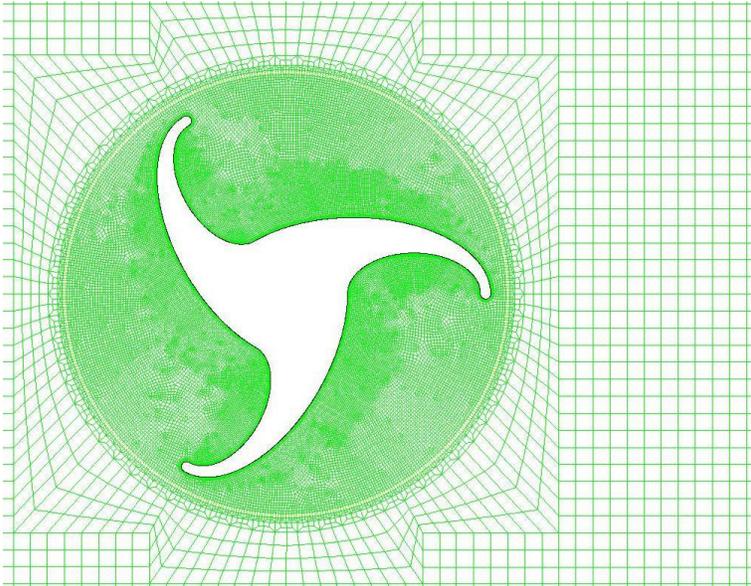
Figure 16 Grid in the near-region of the helical rotor.
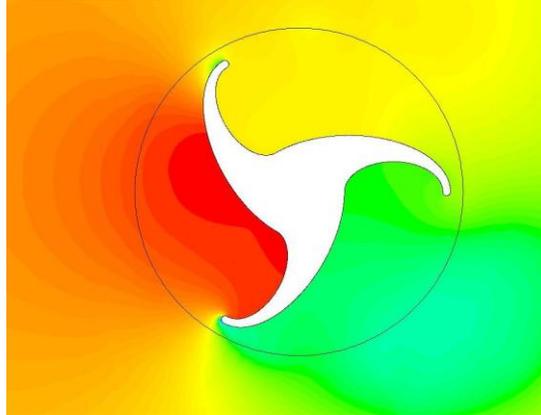
Figure 17 Pressure contours around the helical turbine.
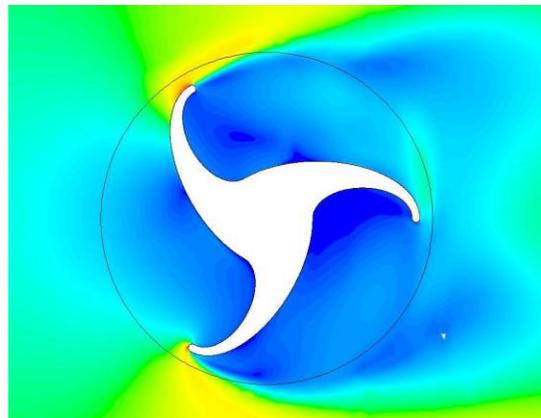
Figure 18 Velocity magnitude contours around the helical turbine.
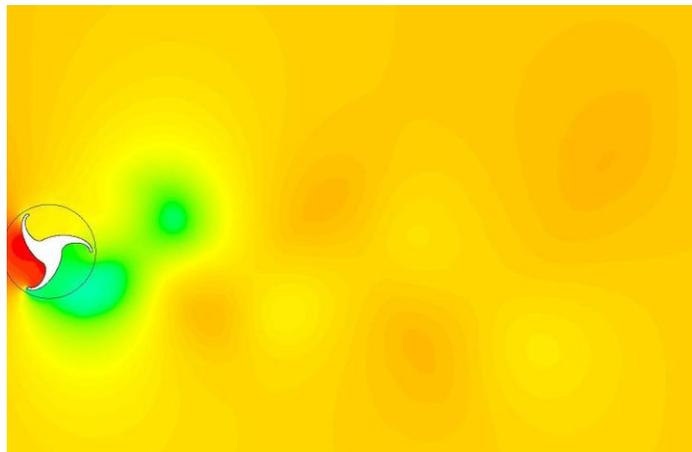
Figure 19 Pressure in the wake of the turbine, showing the alternating shedding-like pattern.

The effect of spatial and temporal discretization is more formally studied for the helical turbine blade. Discretization tests are run at a tip speed ratio of ($\lambda = 0.5$). The 85,000-cell grid is used to simulate the power production of the turbine using 6 different time steps spanning an

order of magnitude. The finer grid, due to its increased computational expense, is simulated at the two extreme time steps. The resulting power predicted is shown in Figure 20. The coarser time steps yield a clear under-prediction of the torques when compared to the finer time steps. Following about 50 time steps per radian, the prediction is relatively unchanged. Concerning spatial discretization, the change in prediction between the coarse and fine grids is small, less than 10%.

Following the discretization study, the 85,000 cell grid is determined to be appropriately accurate for the simulations and 50 time steps per radian is found to be acceptable. Using these parameters, static and dynamic simulations are run for comparison with the rated power of the helical turbine.

The static tests are conducted identical to those of the Savonius simulations. For the helical turbine, however, the rated startup speed of 2 m/s is used rather than the operating wind speed of 14 m/s. This allows for simulations applicable to the operation of the helical turbine in situations where it must start from rest with the minimum wind velocity of 2 m/s. This allows approximation of the minimum starting torque needed to overcome static friction. Figure 21 shows the static test results over the 120-degree span from one blade to the next. The average moment coefficient was slightly less than 0.2, which corresponds to a torque of about 1.2 N-m.
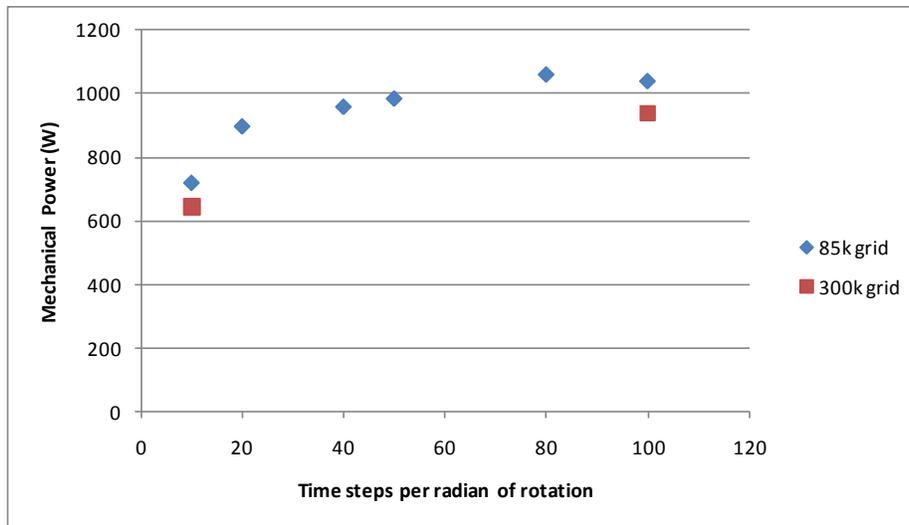
Figure 20 Spatial and temporal discretization study for helical turbine geometry.
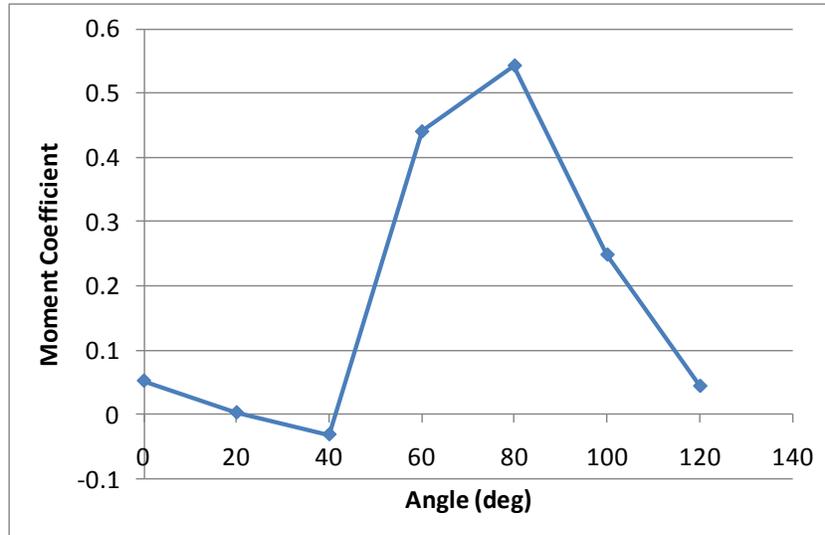
Figure 21 Coefficient of moment for the static simulations at different orientation angles.

Dynamic tests for varying tip speed ratio were run at 14 m/s to match the rated wind speed of the helical turbine, Figure 22. Tip speed ratio was varied from 0.2 to 1.8. The coefficient of moment was monitored throughout the unsteady run and averaged for each tip speed ratio. It is emphasized for the helical blades that the time-averaged moments for a given tip speed ratio in the two-dimensional simulations are applicable to the instantaneous moments experienced by the three-dimensional turbine. since the three-dimensional has a continuous distribution of cross-sectional positions. The averaged moment data is presented in Figure 22. The maximum power coefficient around 0.142 is found at $\lambda = 0.6$, which corresponds to about 134 rpm. The power coefficient calculated from the rated power, Figure 3, is about 0.201 (corresponding to 1.5 kW). The numerical simulation is seen to under-predict the rated power by about 30%. As a note, the torque from friction (~1.2 N-m) corresponds to a moment coefficient less than 0.004.
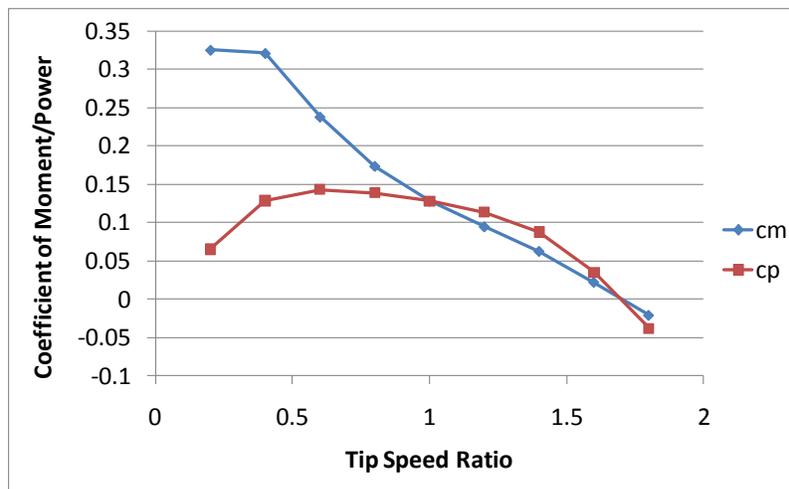
Figure 22 Coefficients of moment and power for the 14 m/s simulation.

A simulation was also carried out for a velocity of 8 m/s for the same tip speed ratios. This simulation was carried out to explore the effect of wind velocity on the power coefficient. Theoretically, the power produced by a wind turbine is proportional to the cube of the wind velocity, as accounted for in the power coefficient. The results of this second dynamic simulation are shown in Figure 23. It is seen that the coefficients of torque and power are relatively unchanged by the change in velocity, confirming that the power produced varies proportional to the third power of the approach velocity, as in the definition of power coefficient.
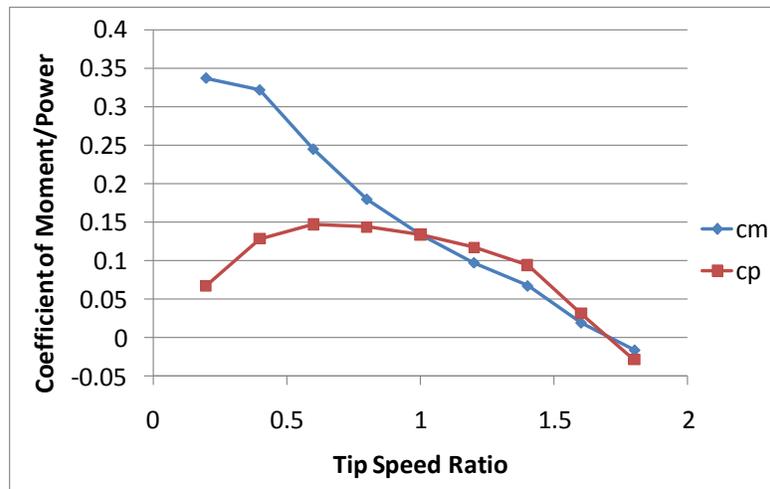
Figure 23 Coefficients of moment and power for the 8 m/s simulation.

Task 1

The first task comprised the majority of the analysis performed for this Report. The baseline parameters for all cases (except where otherwise specified) are shown in Table 3.

Table 3   Baseline parameters for Task 1.

Parameter	Value
Turbine rotational velocity	0 – 44 rad/s
Truck velocity	31.3 m/s (70 mph)
Train velocity	78.2 m/s (175 mph)
Turbine/ truck vertical clearance	0.3 m (1 ft)
Number of turbines	1
Initial turbine rotational velocity	0-5 rad/s
Duration of simulation	1.5 sec

A list of all cases run for the bare turbine configuration of Task 1 is shown in Table 4. A total of nine cases were simulated in this task.

Table 4   Cases included in Task 1.

Case	Description
1a	Uncorrected boundary conditions (small domain, short wake)
1b	Boundary conditions corrected for longer wake
1c	Corrected boundary conditions with large domain, longer wake
1d	Turbine lowered to 0.5 ft clearance
1e	Turbine raised to 2 ft clearance
1f	Truck speed lowered to 65 mph
1g	Truck speed raised to 75 mph
1h	Inclusion of a second turbine

The first three simulations reflect the consideration of influential boundary conditions on the physics of the model. The original model was the smallest domain. It was first tested to see if a domain extension behind the passing truck resulted in more energy predicted. The gains were minimal but the length extension (Case 1b) was kept for subsequent cases. The second change in the boundary (resulting in Case 1c) led to a large difference in the power produced by the turbine. This change, which was largely adaptive to the indications of the first two simulations, was held to be the more accurate boundary.

The subsequent simulations (Cases 1d-h) reflected physical variations in the simulation. The first parameter of study was the clearance of the turbine above the truck (Cases 1c-e). Then the effect of truck speed was studied (Cases 1c,f,g). Finally, a second turbine was included in Case 1h to study how multiple turbines would influence each other compared with the single turbine case (Case 1c,h).

Case 1a (original domain)

Case 1a was the first simulation performed with the model including the passing truck. The two-dimensional domain consisted of a stationary region representing the ambient wind, a translating region containing the truck geometry, and a rotating region containing the bare turbine. The dimensions of the geometry are shown in Figure 24. Notice that the domain only models four meters above the truck, roughly representing a corrected distance between the truck and the overpass (corrected for three-dimensional effects). All baseline parameters from Table 3 are used, except the simulation duration.
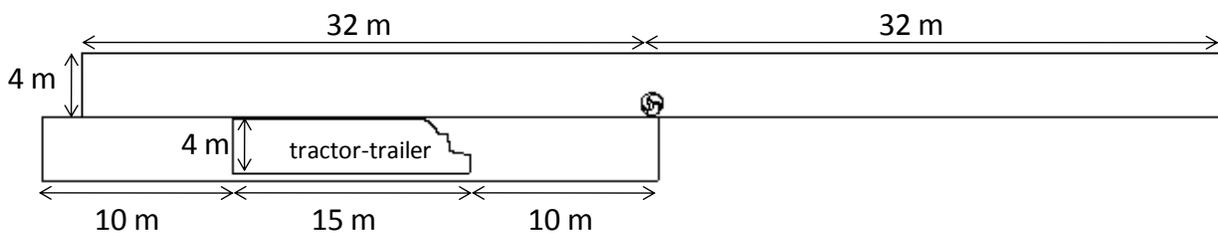
Figure 24 Dimensions of original domain (Case 1a).

It is noted here, an applicable for all subsequent cases, that the truck geometry used in this investigation included a cabin roof spoiler commonly used to reduce drag. A vehicle without this feature would likely produce more power when passing the turbine due to the fact that it is supplying greater energy to the ambient air.

The quadrilateral numerical grid for Case 1a is shown in Figure 25. Elements were clustered near the walls of the turbine, truck, and ground to resolve the viscous sublayer ($y^+{\sim}1$). The total mesh consisted of about 65,000 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 26 and 27, respectively. The domain consisted of three regions: a stationary surroundings region, a translating region encompassing the truck, and a rotating region around the turbine. A sliding mesh was used for the two interfaces: between the stationary and rotating regions, and between the stationary and translating regions. Figure 28 shows a closer view of the truck/turbine interaction.
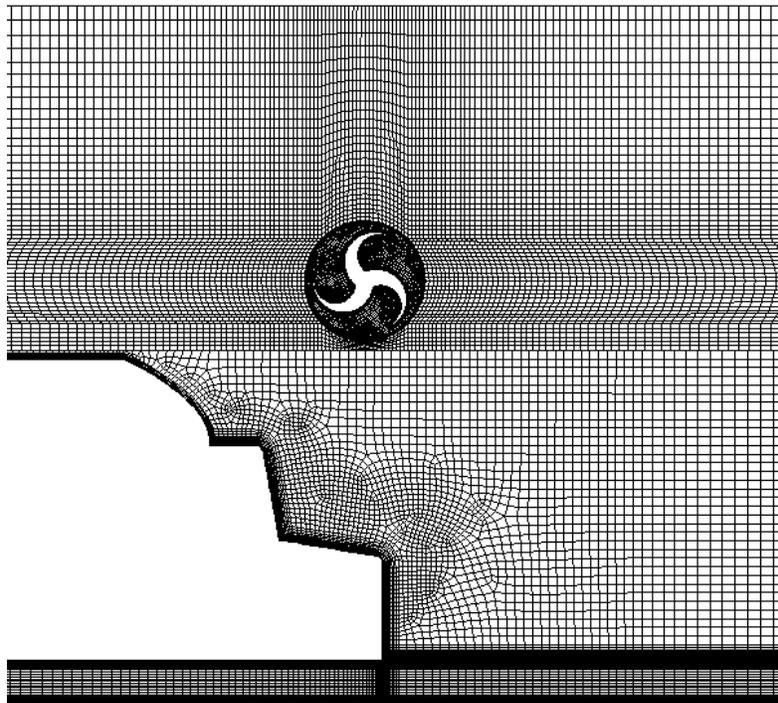
Figure 25 Quadrilateral grid for Case 1a.
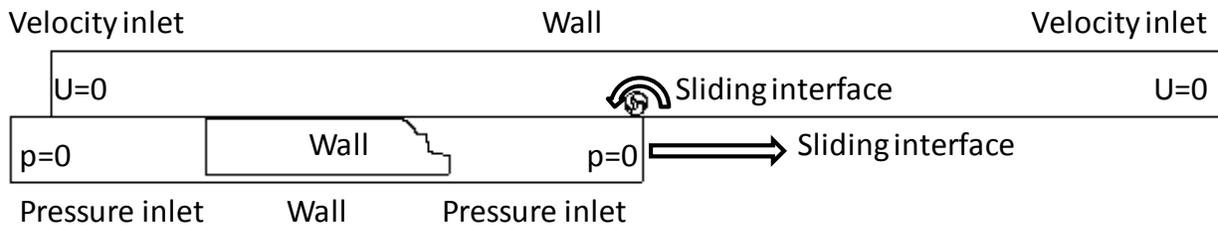
Figure 26 Starting position of the Case 1a simulation.

Velocity inlet   Wall   Velocity inlet

U=0   Sliding interface   U=0

Sliding interface ⟹   p=0   Wall   p=0

Pressure inlet   Wall   Pressure inlet
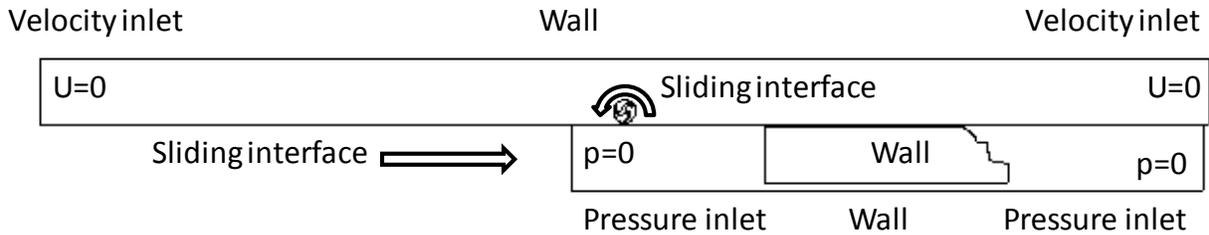
Figure 27 Final position of the Case 1a simulation.

The simulation duration was 1.14 seconds, with 3800 time steps of 0.0003 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 29. Notice that the main direction of the air hitting the turbine is upward (perpendicular to the direction of the truck). It is during the small period of time where the turbine is above the front of the truck that a large blast of upward moving air meets the turbine. Due to the close proximity of the upper bounds of the domain, a large backward moving air mass is required to satisfy continuity. In this way, a physically unrealistic back-draft (Fig. 29) continues to pass energy to the turbine as the truck passes. This effect was found to lead to unrealistically high predictions of the energy production from the turbine. Note that in Fig. 29 the strength of the velocity is such that the vectors cover the geometry of the truck at some places; this is merely a consequence of the velocity vector scaling for this particular representation and does not represent any change in the geometry of the truck.

Wall

Sliding interface

Wall

Wall   Sliding interface

Wall
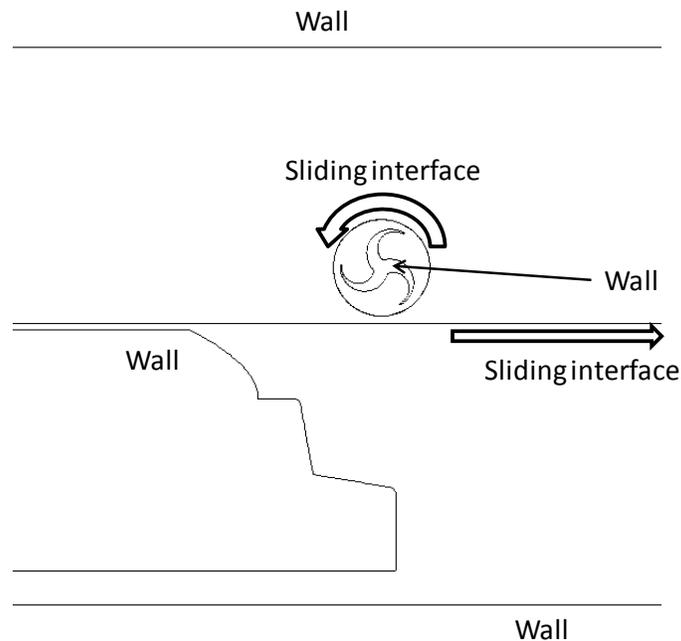
Figure 28 Snapshot of the geometry involved in the truck/turbine interaction for Case 1b.

It was also noticed that the length of the domain in the aft of the truck was insufficient to allow the prediction of the additional energy supplied to the turbine by the spreading wake of the truck, leading to a slight under-prediction in that regard (but still a net over-prediction as will be seen later). Figure 30 shows the energy production as a function of initial angular velocity of the turbine (which could be due to ambient wind or residual angular velocity from a previous truck). Even accounting for the extra kinetic energy in the initial condition of the turbine, the initial angular velocity dramatically improves the output of the turbine (nearly 100% gain).
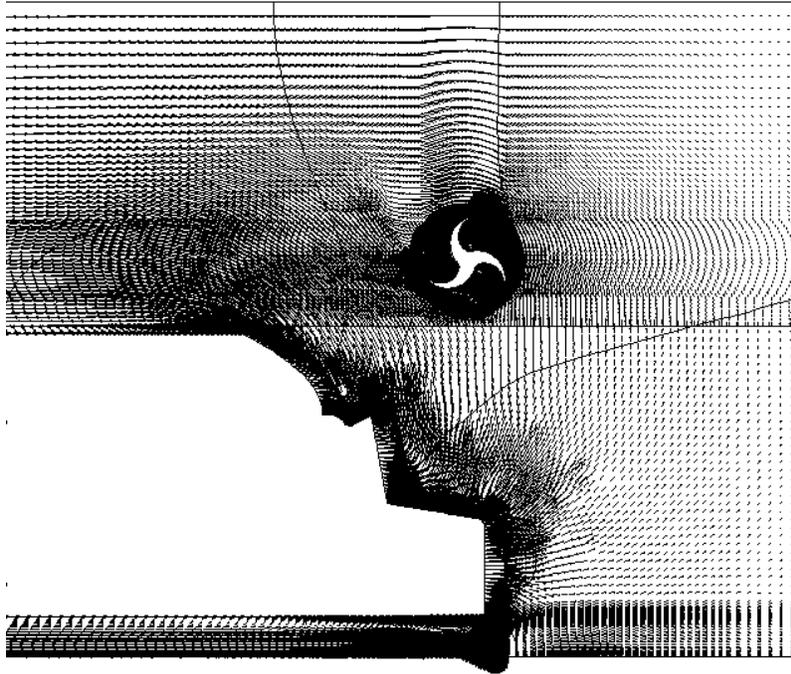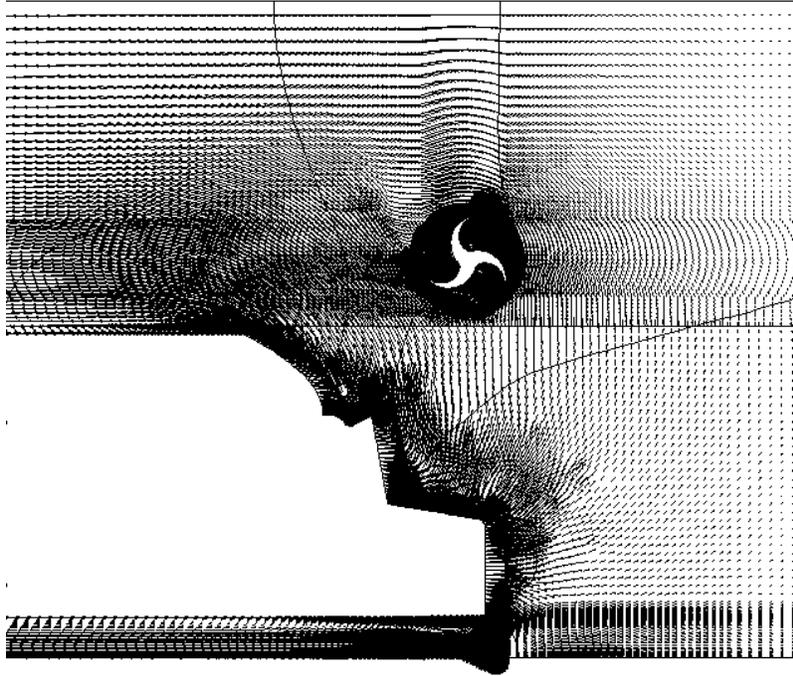

Figure 29 Velocity vectors with pressure contours at the snapshot showing truck/turbine interaction for Case 1a.
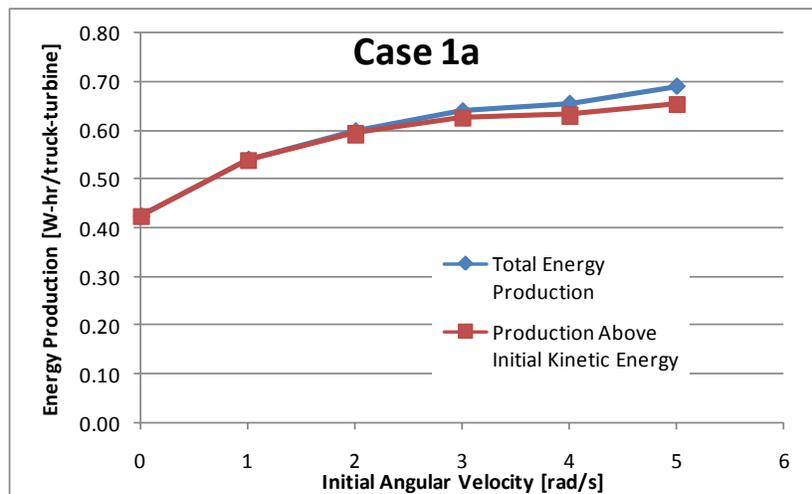
Figure 30 Energy production as a function of turbine initial angular velocity for Case 1a.

Case 1b (extended wake)

Before leaving any baseline parameters, it was necessary to establish the effect that the domain was having on the numerical prediction. Case 1b set out to establish the effect of domain extension in the wake of the truck. This region was extended from 10m to 50m long, as shown in Figure 31. In all other aspects, this simulation was equivalent to Case 1a, including the overpass placement at 4 meters above the truck. All baseline parameters from Table 3 were used, except the duration of the simulation.
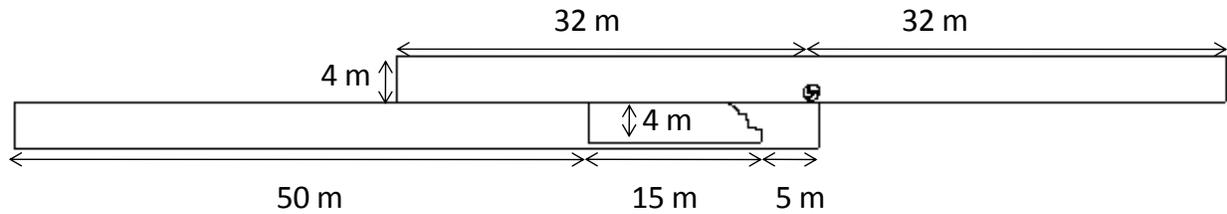
Figure 31 Dimensions of lengthened domain (Case 1b).

The quadrilateral grid created for Case 1b is shown in Figure 32. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 61,600 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 33 and 34, respectively. The domain consisted of three regions: a stationary surroundings region, a translating region around the truck, and a rotating region around the turbine. The sliding mesh technique was used for the two interfaces: between the stationary and rotating regions, and between the stationary and translating regions. Figure 35 shows a closer view of the domain at the point where the truck and turbine interact.
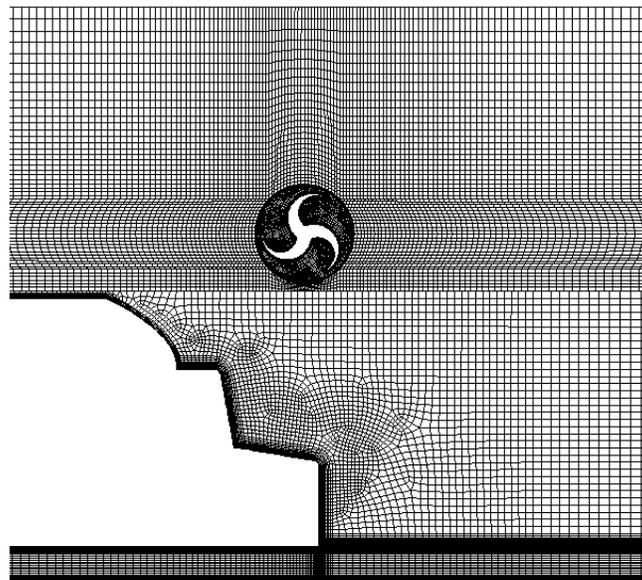
Figure 32 Quadrilateral grid for Case 1b.
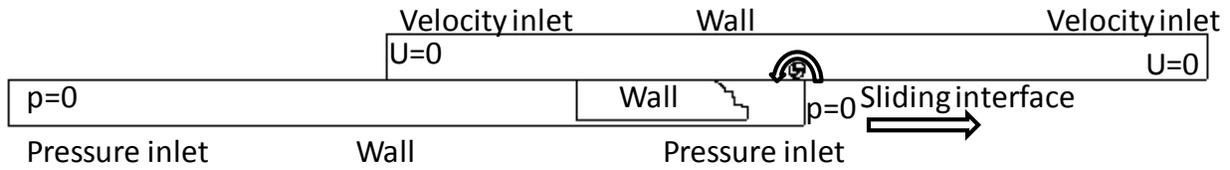
Figure 33 Starting position of the Case 1b simulation.
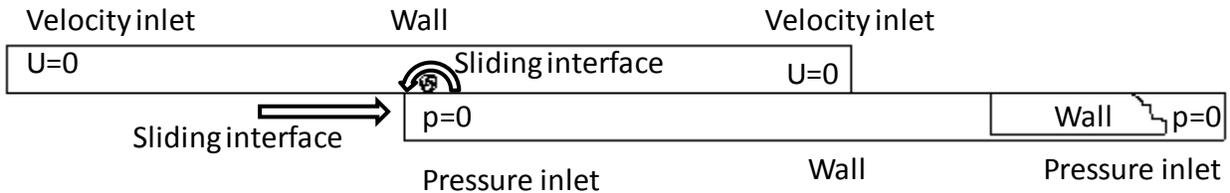
Figure 34 Final position of the Case 1b simulation.
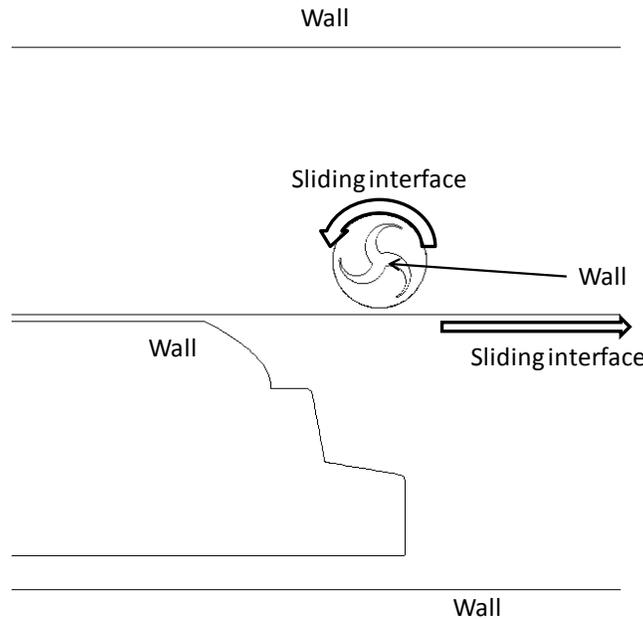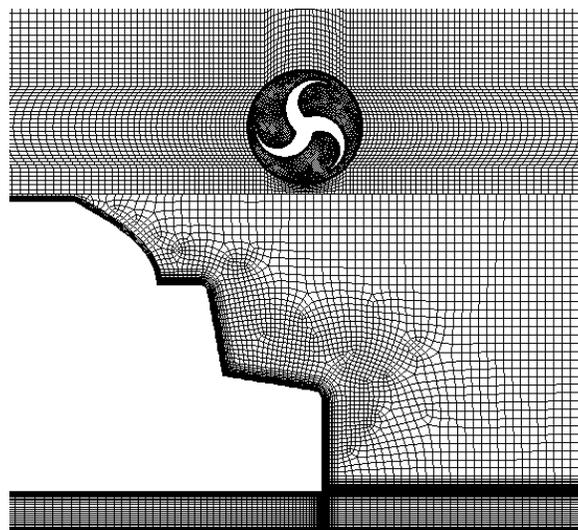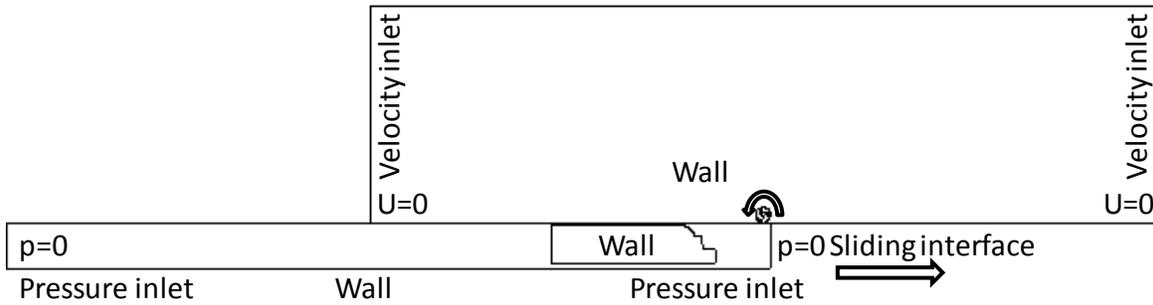

Figure 35 Snapshot of the geometry involved in the truck/turbine interaction for Case 1b.

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 2.2 seconds long with 2200 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 36. The upward blast followed by backward-draft of air was again noticed. The lengthening of the wake region allowed for a noticeable change in the energy output, Figure 37. For all subsequent simulations, therefore, a wake region of 50 meters was used. The domain above the truck was still only four meters, and therefore the results of this case were again considered an over-prediction. A comparison of Case 1a and Case 1b results shows that Case 1b represents about a 50% increase in energy production in the case with no initial turbine rotational velocity.

The increase remains about constant (~ 0.2 W-hr/truck-turbine) for all initial angular velocities. The truck was seen to pass across the end of the stationary domain before the end of the solution, making the pressure field unstable. Therefore, all subsequent cases stopped before this occurred.
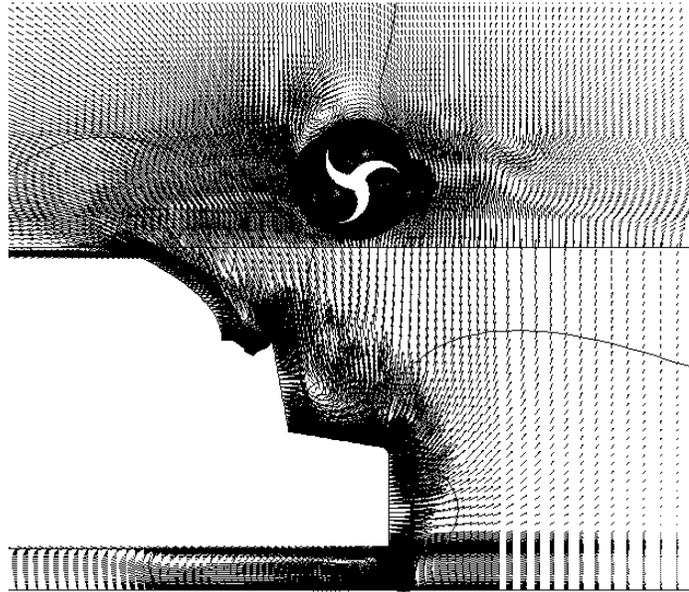

Figure 36 Velocity vectors with pressure contours at the snapshot showing truck-turbine interaction for Case 1b.
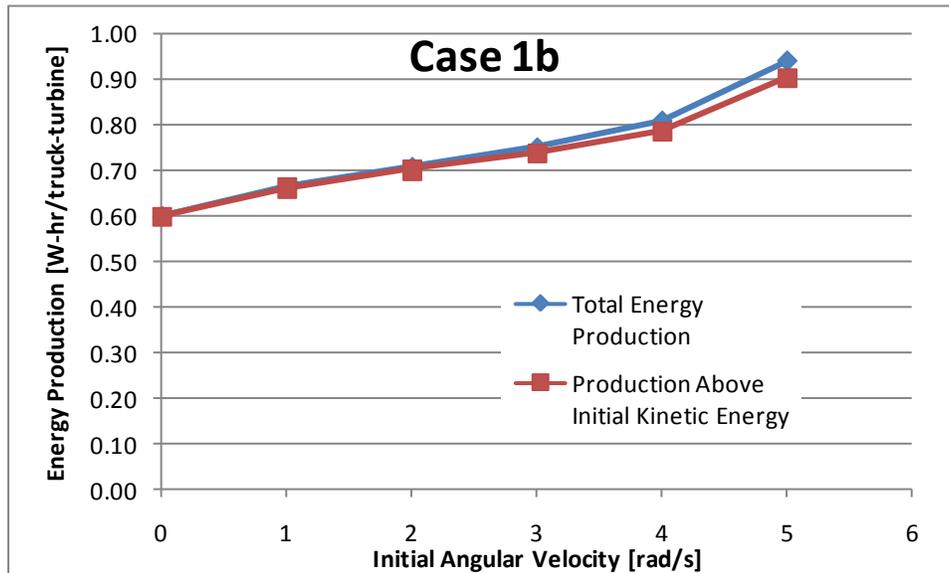
Figure 37 Energy production as a function of initial angular velocity for Case 1b.

Case 1c (baseline)

To establish the squeezing effect of the first two simulations (with the upper bound of the domain placed 4 meters above the truck's upper surface), Case 1c extended the domain above the truck. The two-dimensional length above the truck was set to 20 m, which was an estimate of the effective open flow area around a truck passing under an overpass in three dimensions. The dimensions for Case 1c are shown in Figure 38. The change from 4 m to 20 m represents a significant change. For this reason, it was the only major change to the simulation methodology between Cases 1b and 1c.
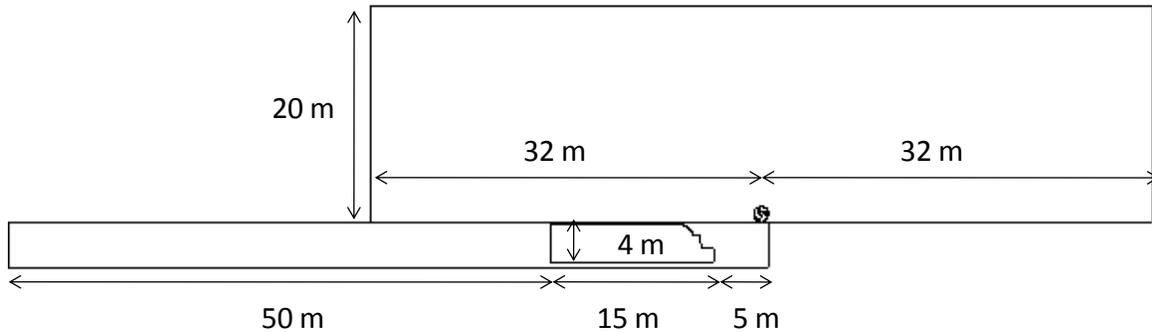
Figure 38 Dimensions of enlarged domain (Case 1c and following).

The quadrilateral grid for Case 1c is displayed in Figure 39. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 76,000 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 40 and 41, respectively, and consist of the same three-region composition as the previous cases. Figure 42 shows a closer view of the domain at the point where the truck and turbine interact.

Figure 39 Quadrilateral grid for Cases 1c, 1f, 1g.

Figure 40 Starting position of the Case 1c simulation.
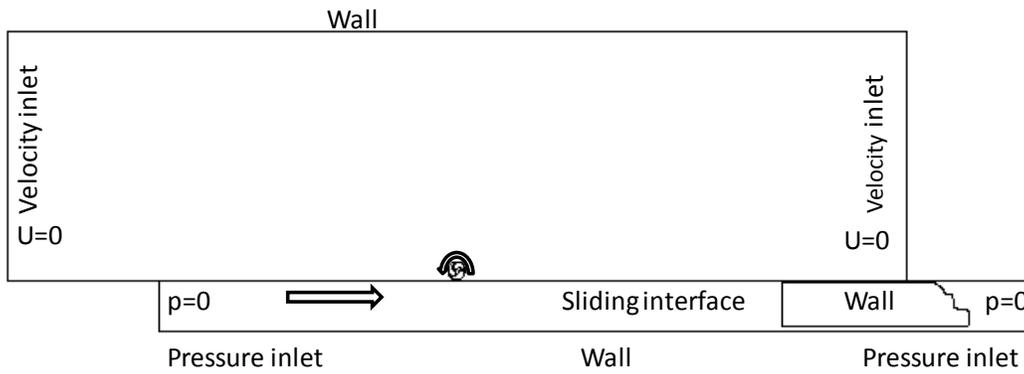
Figure 41 Final position of the Case 1c simulation.
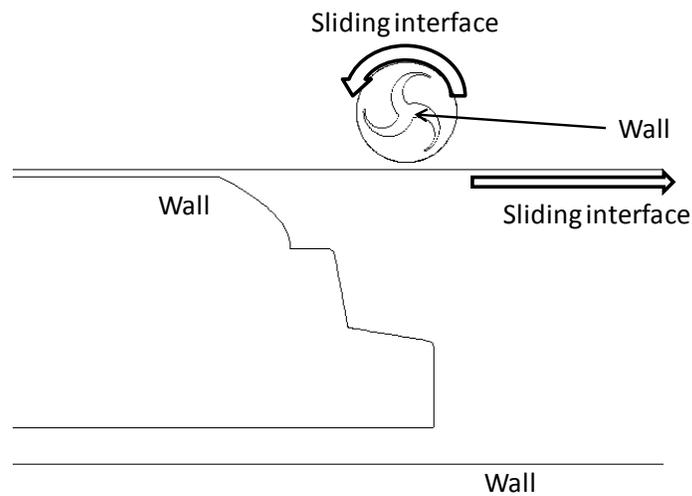
Figure 42 Snapshot of the geometry involved in the truck/turbine interaction for Case 1c.

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long, with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 43. The upward blast of air was again noticed, but the backward draft of air was

dramatically diminished due to the added area for satisfying the continuity equation. This had the effect of dramatically reducing the output of the turbine, Figure 44. The 'from rest' case shows a decrease from 0.61 to 0.11 W-hr/truck-turbine. The increase due to initial angular velocity is more dramatic, however. The output doubles with an initial turbine angular velocity of 5 rad/s, even after the initial kinetic energy of the turbine has been taken out.

Figure 43 Velocity vectors with pressure contours at the snapshot showing truck/turbine interaction for Case 1c.
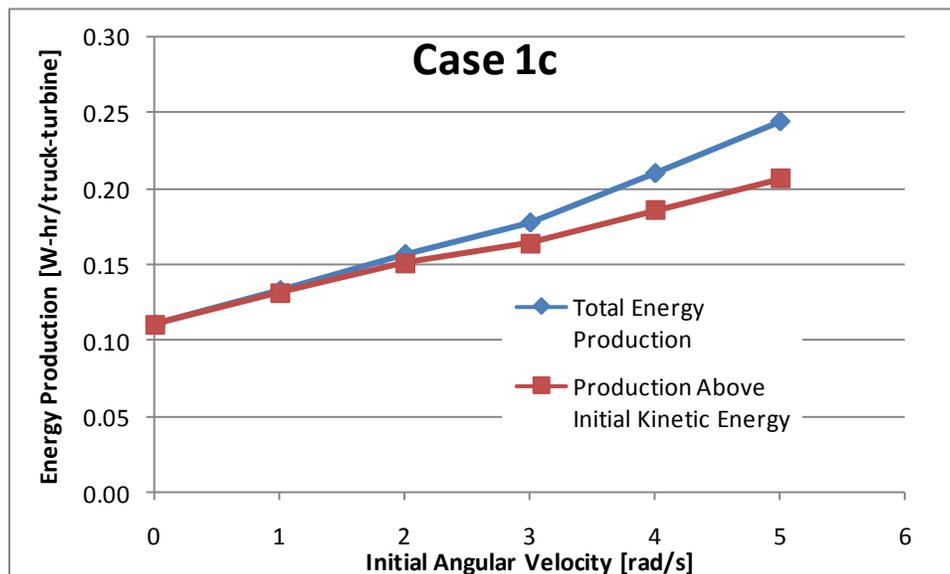
Figure 44 Energy production as a function of initial angular velocity for Case 1c.

Case 1d (6-inch clearance)

With the domain parameters of Case 1c established for the remainder of the simulations, the baseline parameters of Table 3 were changed in subsequent simulations to establish the effect of various parameters on the turbine energy output. For Case 1d, the turbine was moved from 1ft above the truck to 0.5 ft above the truck. This decrease in clearance was expected to enhance the output of the truck by moving the turbine into closer interaction with the truck. All other baseline parameters from Table 3 were retained.

The quadrilateral grid for Case 1d is displayed in Figure 45. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 70,300 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 46 and 47, respectively, consisting of the same three-region scheme as the previous cases. Figure 48 shows a closer view of the domain at the point where the truck and turbine interact.
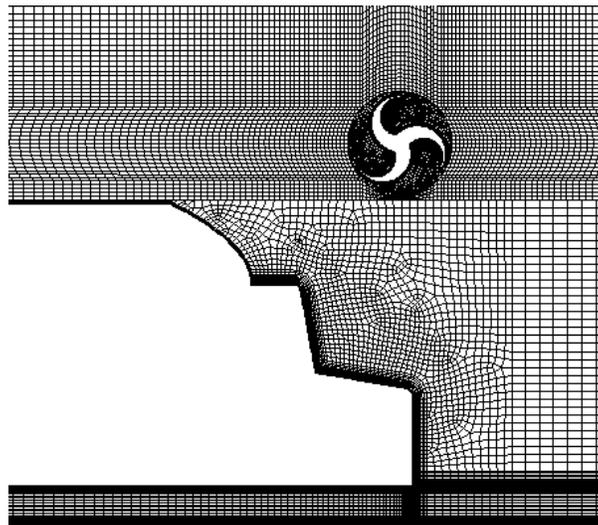
Figure 45 Quadrilateral grid for Case 1d (6" clearance).
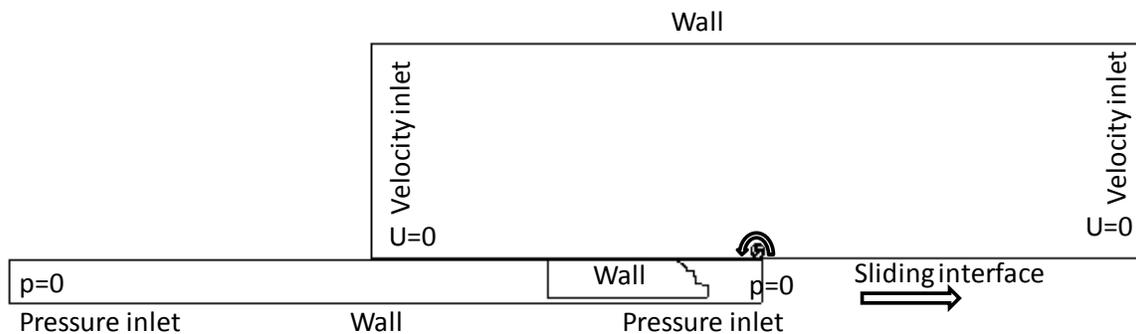
Figure 46 Starting position of the Case 1d simulation (6" clearance).
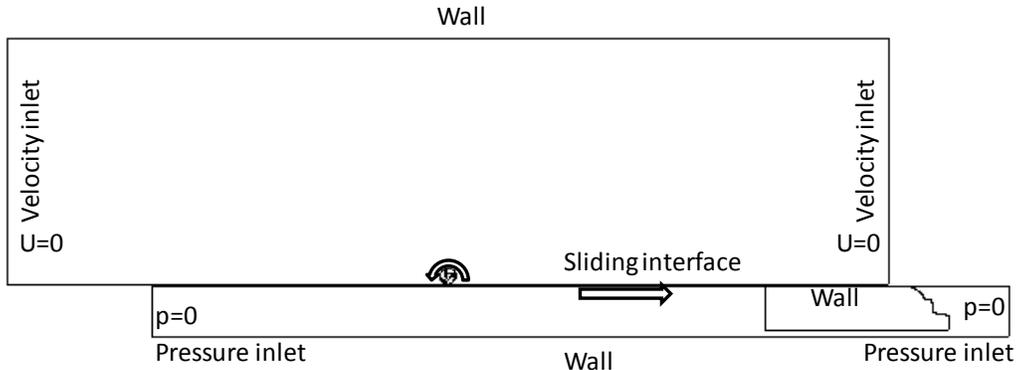
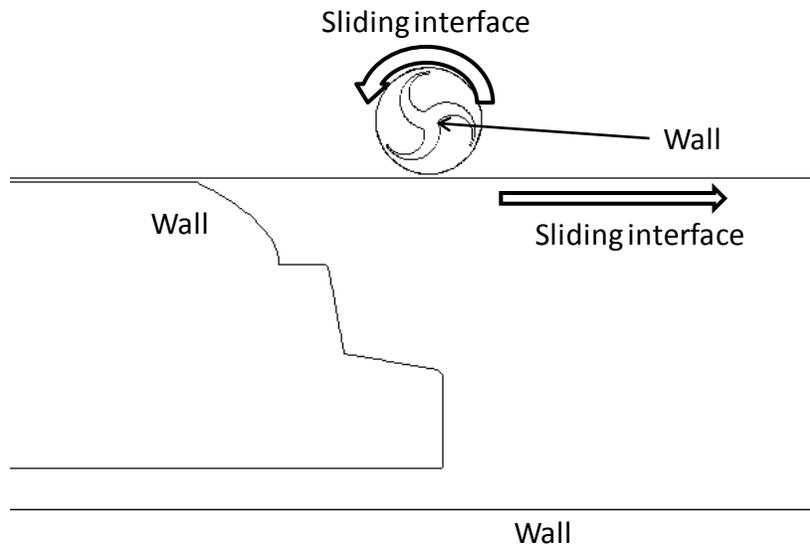**Figure 47 Final position of the Case 1d simulation.**

Figure 48 Snapshot of the geometry involved in the truck/turbine interaction for Case 1d.

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 49. The upward blast of air was felt more strongly by the closer turbine, leading to a higher output of energy, Figure 50. The output starts at 0.16 W-hr/truck-turbine for the turbine starting from rest and increases linearly to nearly double the 'from rest' value. In comparison with Case 1c, the output increases from about 0.11 to 0.16 W-hr/truck-turbine for the zero initial angular velocity case.

36
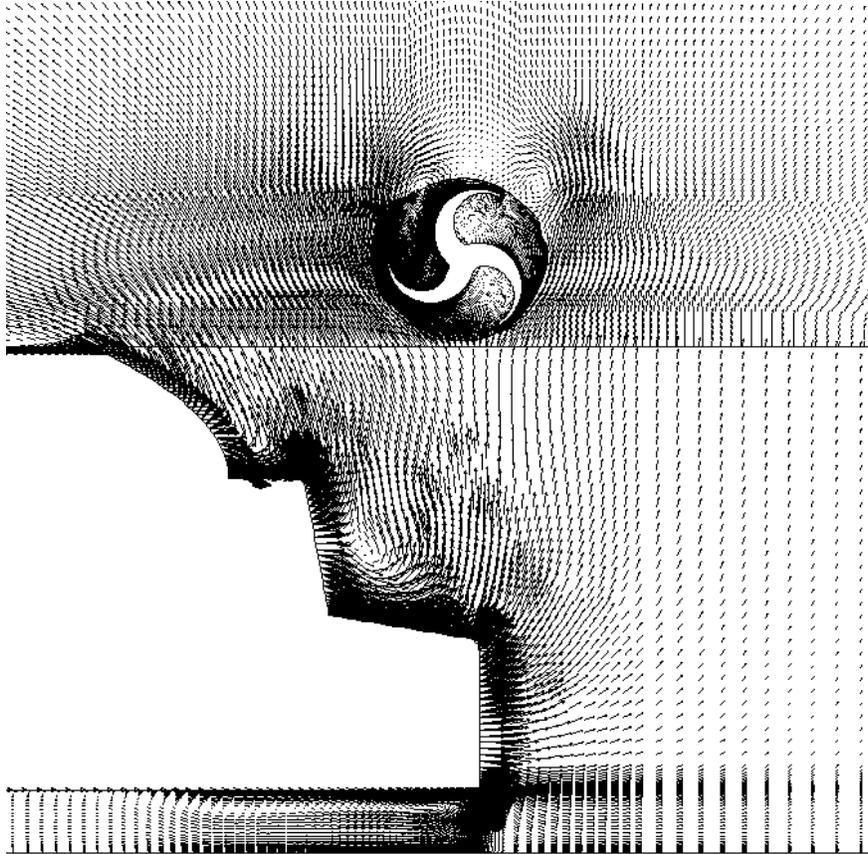
Figure 49 Velocity vectors with pressure contours at the snapshot showing truck/turbine interaction for Case 1d (6" clearance).
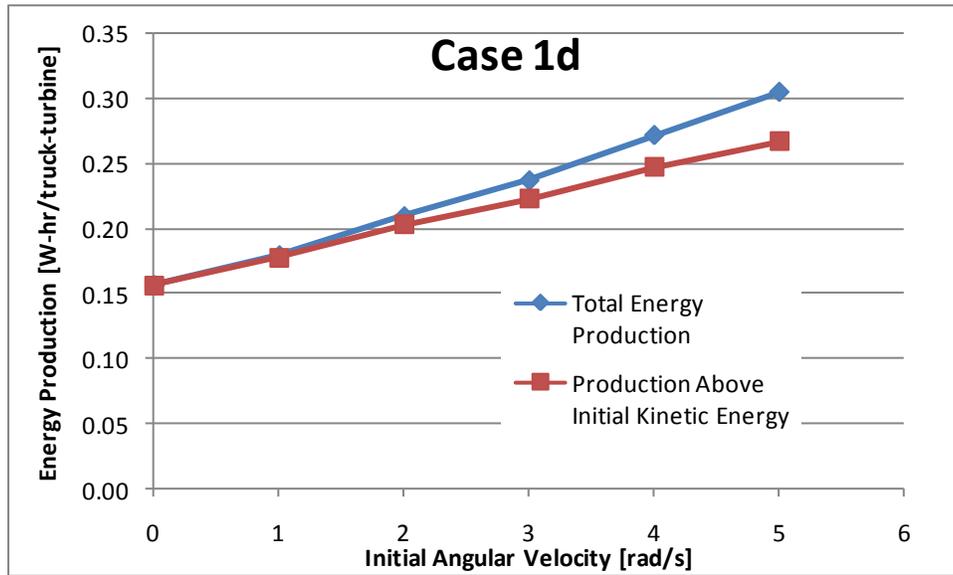

Figure 50 Energy production as a function of initial angular velocity for Case 1d (6" clearance).

Case 1e (2-foot clearance)

For Case 1e, the turbine was moved from 1ft above the truck to 2 ft above the truck. This represents the opposite effect of Case 1d. This increase in clearance was expected to diminish the output of the truck by moving the turbine further away from the truck. All other baseline parameters from Table 3 were retained.

The quadrilateral grid for Case 1d is displayed in Figure 51. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 79,100 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 52 and 53, respectively, and consist of the same three-region scheme as the previous cases. Figure 54 shows a closer view of the domain at the point where the truck and turbine interact.
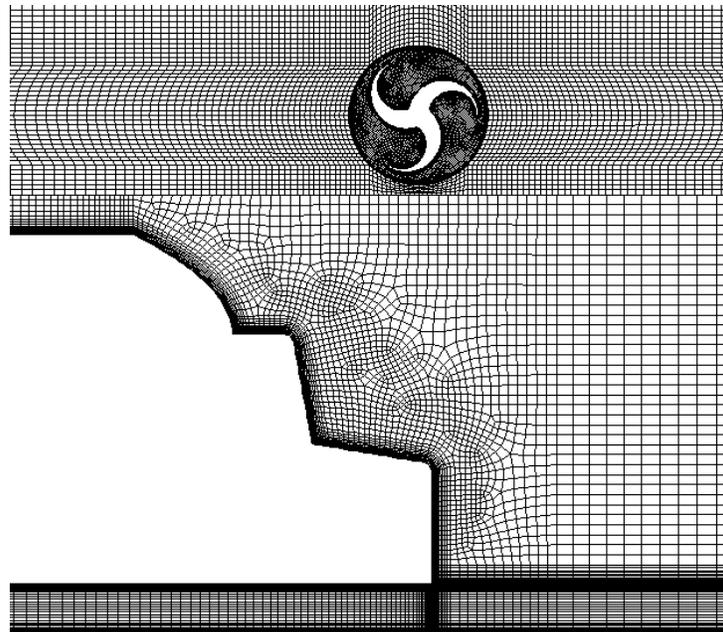
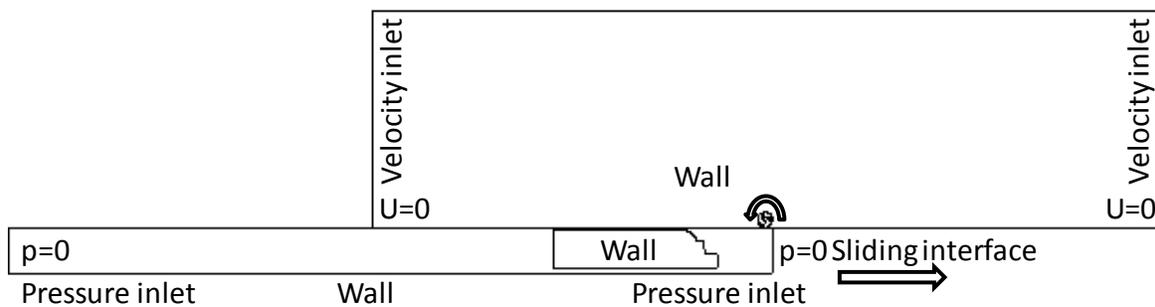**Figure 51 Quadrilateral grid for Case 1e (2' clearance).**

Figure 52 Starting position of the Case 1e simulation (2' clearance).
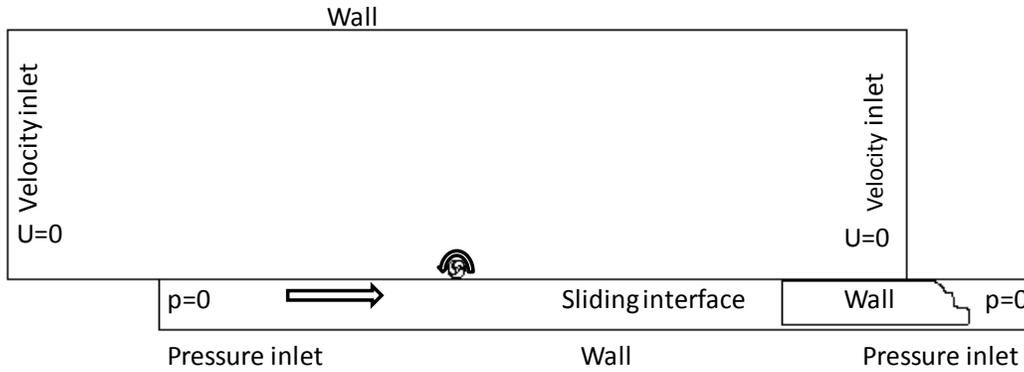
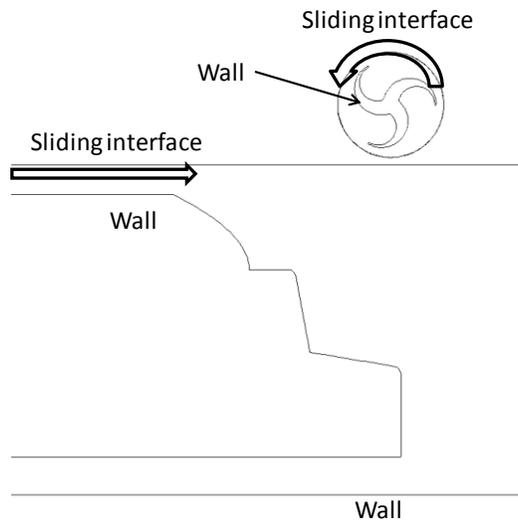**Figure 53 Final position of the Case 1e simulation (2' clearance).**

Figure 54 Snapshot of the geometry involved in the truck/turbine interaction for Case 1e (2' clearance).

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 55. The upward blast of air was not as strong for the turbine, leading to a lower output of energy, Figure 56. The output starts at 0.09 W-hr/truck-turbine for the turbine starting from rest, and increases linearly to nearly double the 'from rest' value. In comparison with Case 1c, the output decreases from about 0.11 to 0.09 W-hr/truck-turbine for the zero initial angular velocity case. The turbine output, as expected, increases when moved closer to the truck and decreases when moved further away. The simulations suggest that the trend with spacing is not linear.
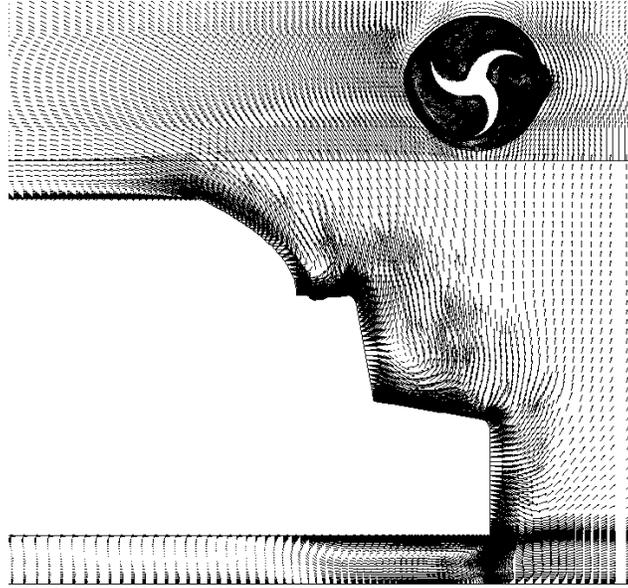
Figure 55 Velocity vectors at the snapshot showing truck/turbine interaction for Case 1e (2' clearance).
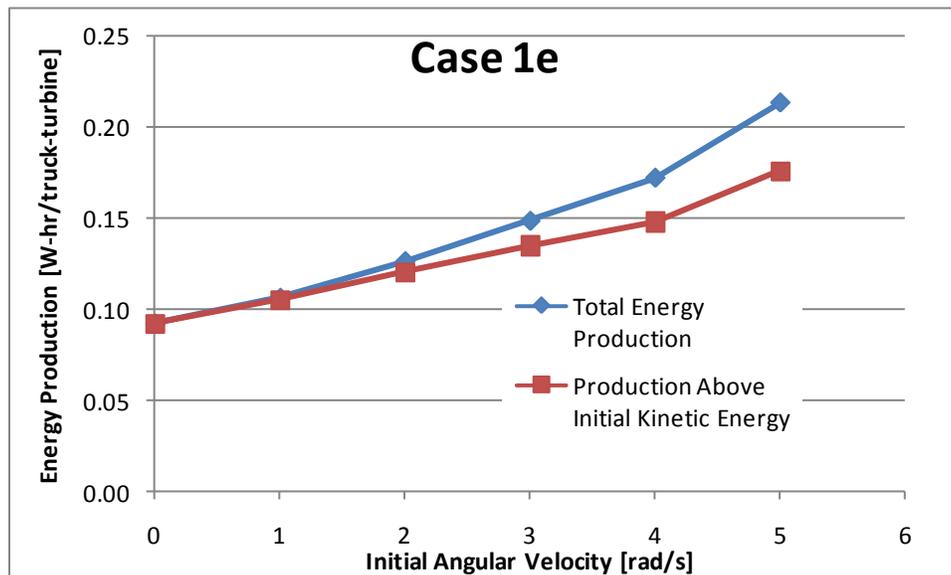

Figure 56 Energy production as a function of initial angular velocity for Case 1e (2' clearance).

Case 1f (truck speed 65 mph)

With the effect of numerical domain and turbine proximity to the truck determined, the next two simulations were designed to establish the effect of truck velocity on the turbine energy output. For Case 1f, the truck velocity was decreased to 65 mph. This decrease in truck velocity was expected to decrease the output of the truck by decreasing the intensity with which the air was pushed aside from the truck. All other baseline parameters from Table 3 were retained.

The quadrilateral grid for Case 1f is displayed in Figure 39 (same as baseline Case 1c). Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 76,000 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 40 and 41, respectively, consisting of the same three-region scheme as the previous cases. Figure 42 shows a closer view of the domain at the point where the truck and turbine interact.

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 57. The upward blast of air was not as strong for this case because of the lower truck velocity, with the output graphed in Figure 58. The output starts at 0.09 W-hr/truck-turbine for the turbine starting from rest and increases linearly to nearly double the 'from rest' value. In comparison with Case 1c, the output decreases from about 0.11 to 0.09 W-hr/truck-turbine for the zero initial angular velocity case. This decrease in output is about equivalent to moving the turbine further away (with the magnitudes specified in this report).
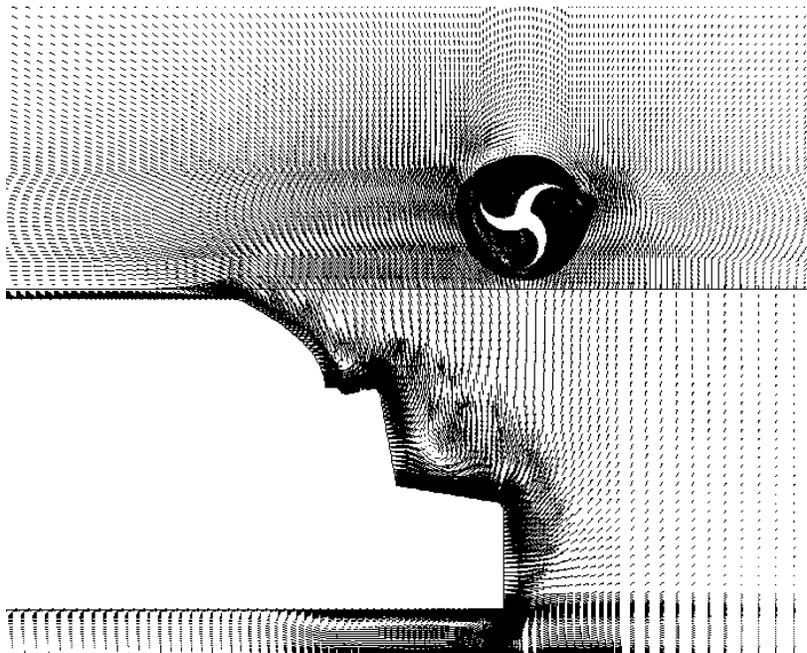

Figure 57 Velocity vectors at the snapshot showing truck/turbine interaction for Case 1f (65 mph truck).
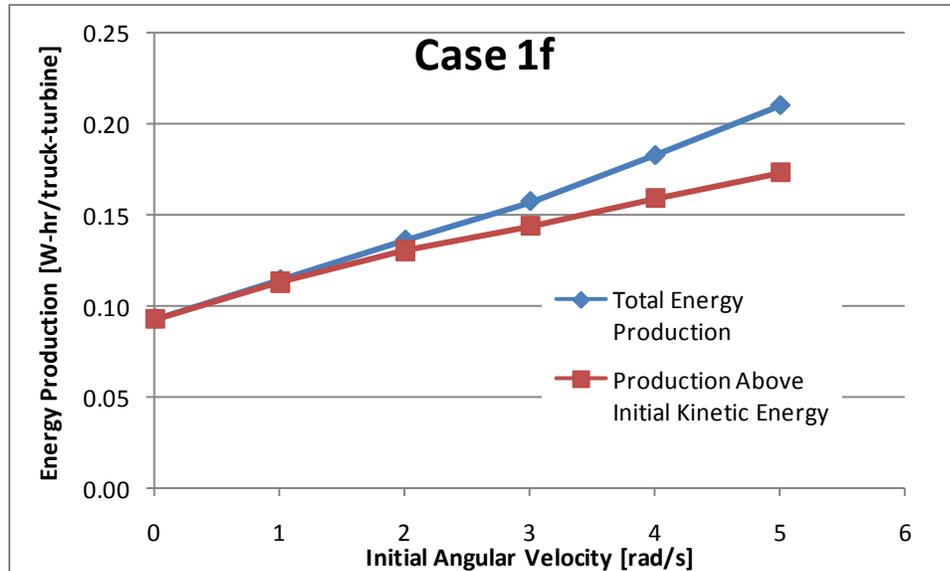
Figure 58 Energy production as a function of initial angular velocity for Case 1f (65 mph truck).

Case 1g (truck speed 75 mph)

For Case 1g, the truck velocity was increased to 75 mph. This represents the opposite of Case 1f. The increase in truck velocity was expected to increase the output of the truck by increasing the intensity with which the air was pushed aside by the truck. All other baseline parameters from Table 3 were retained.

The quadrilateral grid for Case 1g is displayed in Figure 39. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 76,000 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 40 and 41, respectively, consisting of the same three-region scheme as in the previous cases. Figure 42 shows a closer view of the domain at the point where the truck and turbine interact.

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 59. The upward blast of air was stronger for this case because of the higher truck velocity, Figure 60. The output starts at 0.13 W-hr/truck-turbine for the turbine starting from rest and increases linearly to nearly double the 'from rest' value. In comparison with Case 1c, the output increases from about 0.11 to 0.13 W-hr/truck-turbine for the zero initial angular velocity case. Cases 1f and 1g confirm that the output increases as truck velocity increases. The trends shows that this increase is super-linear, as would be expected given that constant wind turbine power is proportional to the cube of velocity. The trend is not as dramatic, however, as the third order increases seen in the standalone cases.
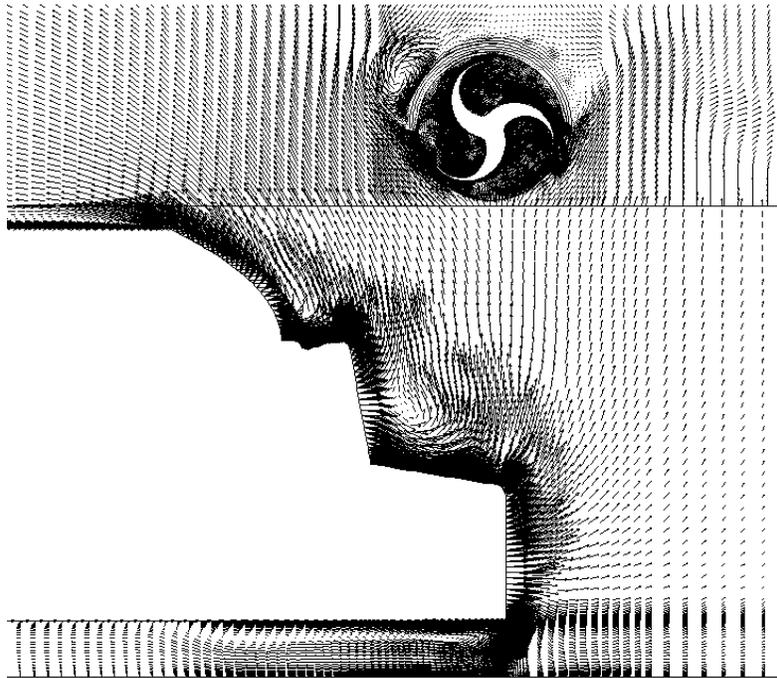
42

Figure 59 Velocity vectors at the snapshot showing truck/turbine interaction for Case 1g (75 mph truck).
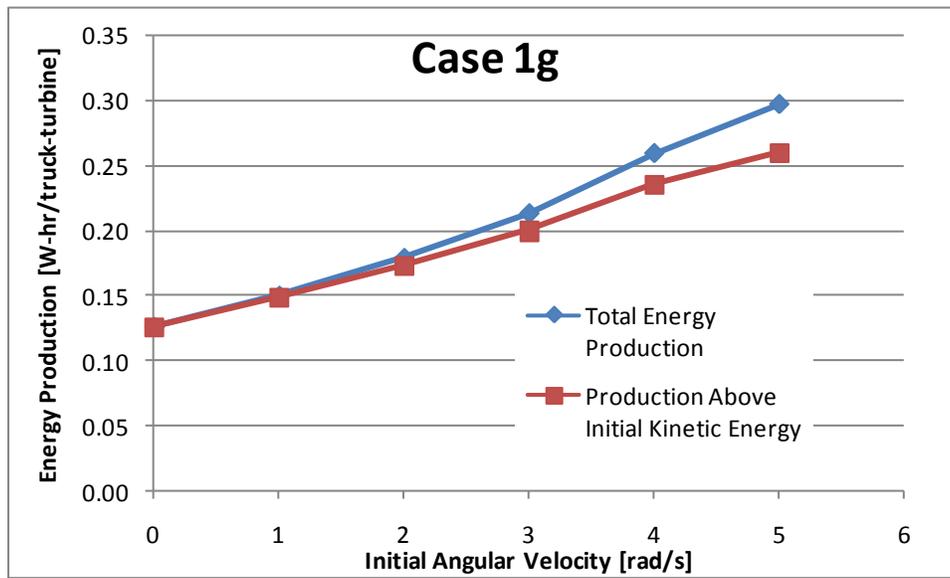

Figure 60 Energy production as a function of initial angular velocity for Case 1g (75 mph truck).

Case 1h (multiple turbines)

All cases previous to Case 1h simulate a single turbine. The objective of Case 1h is, for the conditions of Case 1c, to extend the simulation to include two turbines (still 4ft in diameter) spaced with 4 feet in between the outermost edges of each. Given the general order of magnitude for the energy production mentioned thus far, multiple turbines would be beneficial for energy output under a busy highway overpass. With the need for multiple turbines spaced closely together, it is important to evaluate the impact the turbines may have on each other. Case 1h was designed to explore this impact. All the baseline parameters of Table 3 are retained, except for the added second turbine.

The quadrilateral grid for Case 1g is displayed in Figure 61. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 88,600 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 62 and 63, respectively, consisting of the same three-region scheme as the previous cases, with a fourth region added for the second turbine. This fourth region, as with the first turbine, is rotational and shares a sliding mesh interface with the stationary domain. Figure 64 shows a closer view of the domain at the point where the truck and turbines interact.
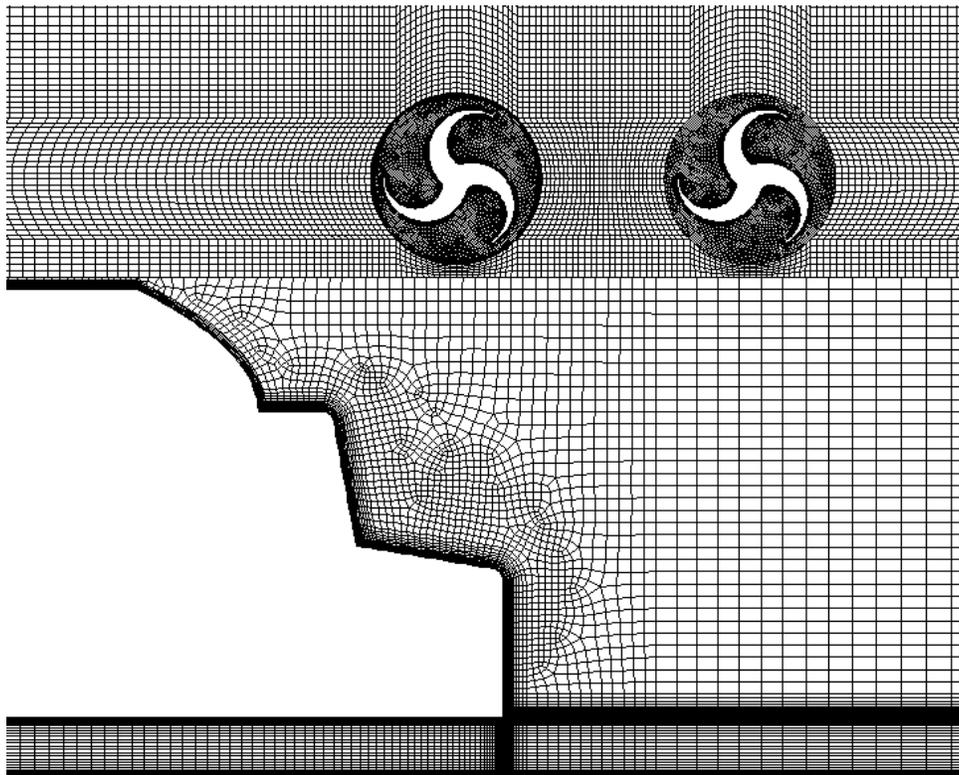

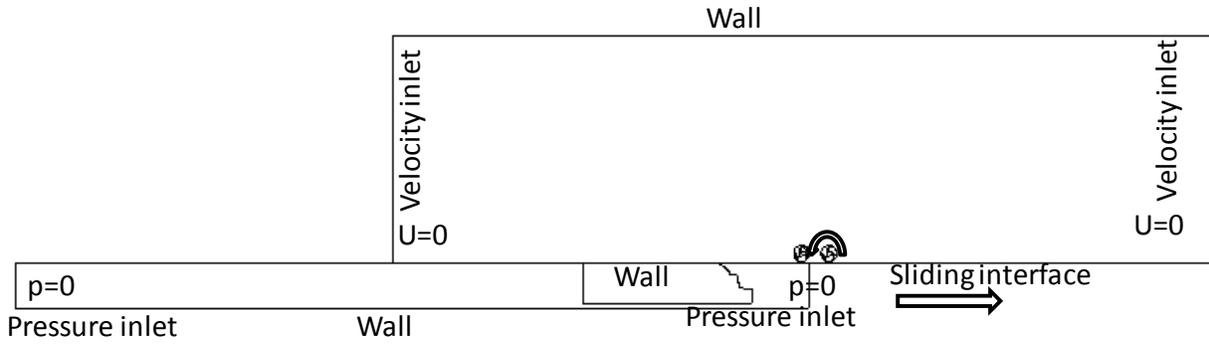**Figure 61 Quadrilateral grid for Case 1h.**

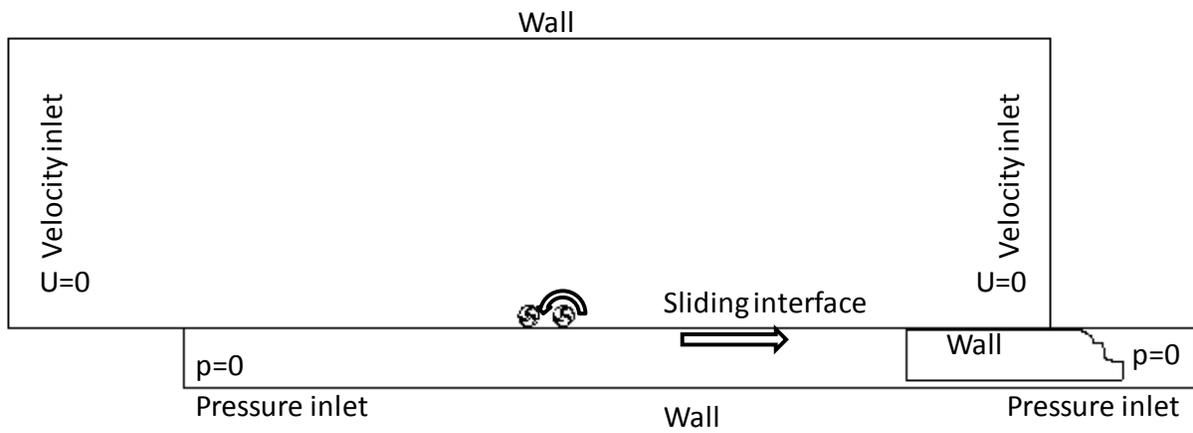**Figure 62 Starting position of the Case 1h simulation.**

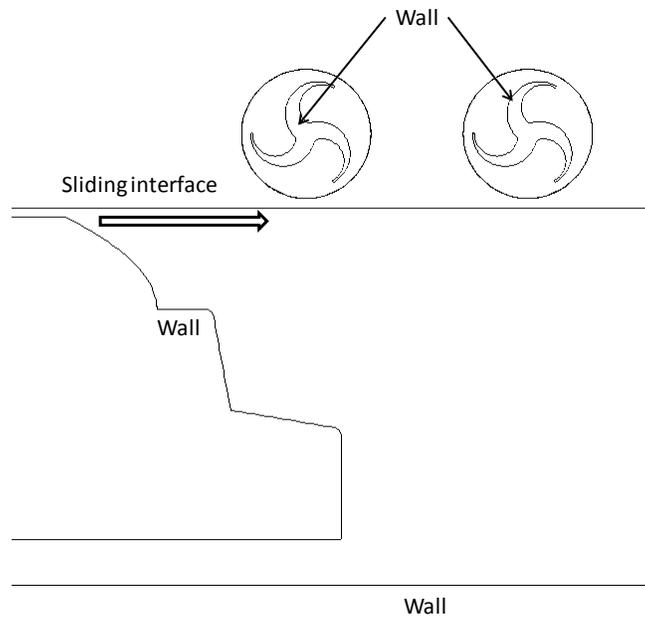**Figure 63 Final position of the Case 1h simulation.**

Figure 64 Snapshot of the geometry involved in the truck/turbine interaction for Case 1h.

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 65. The energy production of the second turbine is shown in Figure 66. It is seen that the output significantly drops for the second turbine, to about half of the output of the single turbine. Therefore, the interaction between two turbines reflects a significant hindrance to energy production. Possible methods of weakening this interaction (and increasing the output of the second turbine) include further spacing the turbines or staggering the turbines such that there is only partial overlapping of adjacent turbines from the view of the approaching truck.
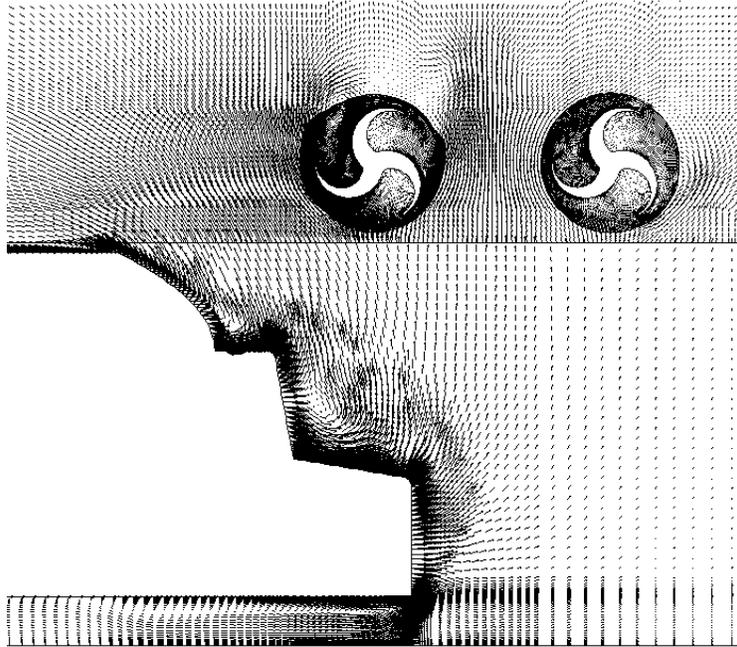

Figure 65 Velocity vectors at the snapshot showing truck/turbine interaction for Case 1h.
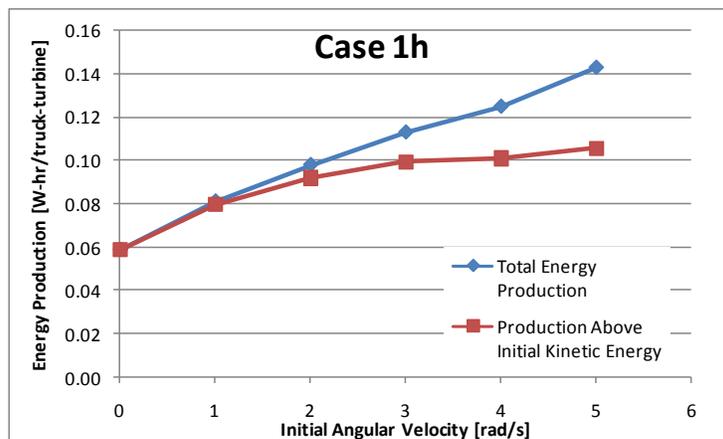

Figure 66 Energy production as a function of initial angular velocity for Case 1h.

Task 2

Task 1 aimed to explore the bare turbine and the effect that important parameters had on its energy output. Tasks 2 and 3 were designed to explore the effect of added components in an attempt to focus the truck-produced air stream into the turbine and increase energy output. Task 2 focuses on turbine shrouding.

Case 2a

Shrouding wind turbines is one method for improving their energy output. The objective of Case 2a is to use a 225-degree shroud around the turbine and observe the changes in flow pattern and energy output. All conditions mimic the conditions of Case 1c, except that the geometry of the domain is altered to incorporate the shroud. All the baseline parameters of Table 3 are retained (1' clearance, 70 mph truck).

The quadrilateral grid for Case 2a is displayed in Figure 67. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 77,800 cells. The domain configurations at the beginning and end of the transient simulation are shown in Figures 68 and 69, respectively, consisting of the same three region scheme as the previous cases. Figure 70 shows a closer view of the domain at the point where the truck and turbine interact. The shroud covers the upper half of the turbine from the front and wraps around the top, covering most of the rear of the turbine. One disadvantage to this design is that extension to multiple turbines would not be as efficient because of the added rearward air blockage.
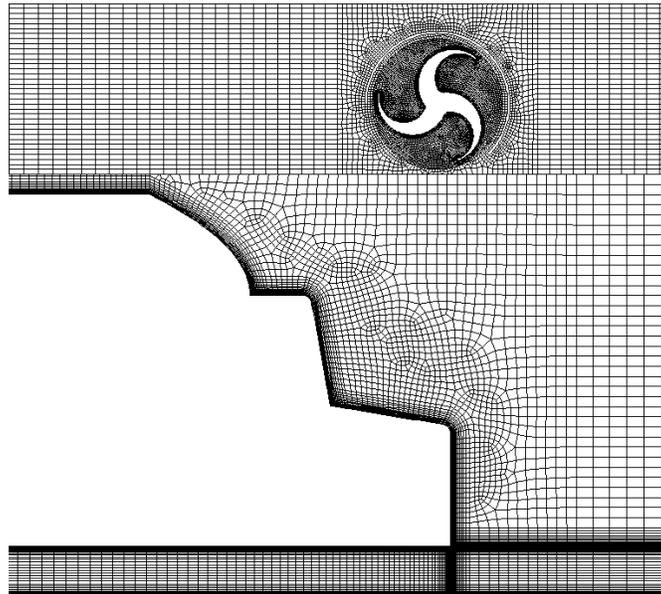

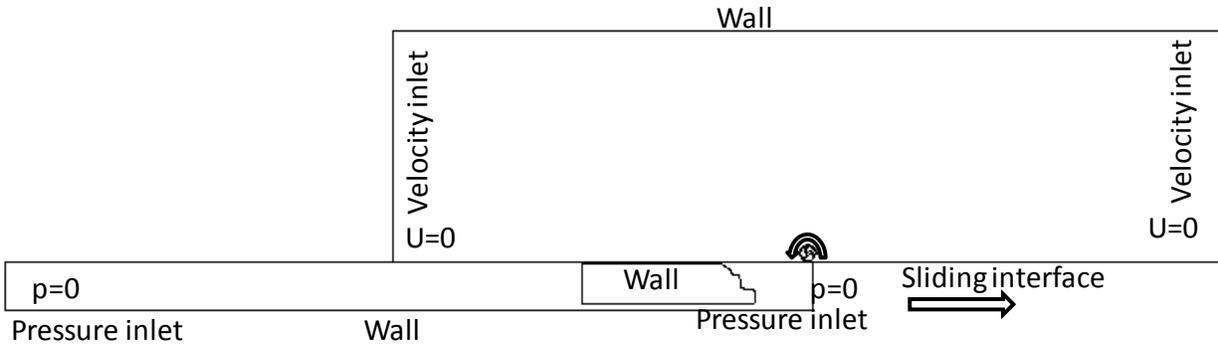**Figure 67 Quadrilateral grid for Case 2a (225-deg shroud).**

Figure 68 Starting position of the Case 2a simulation (225-deg shroud).
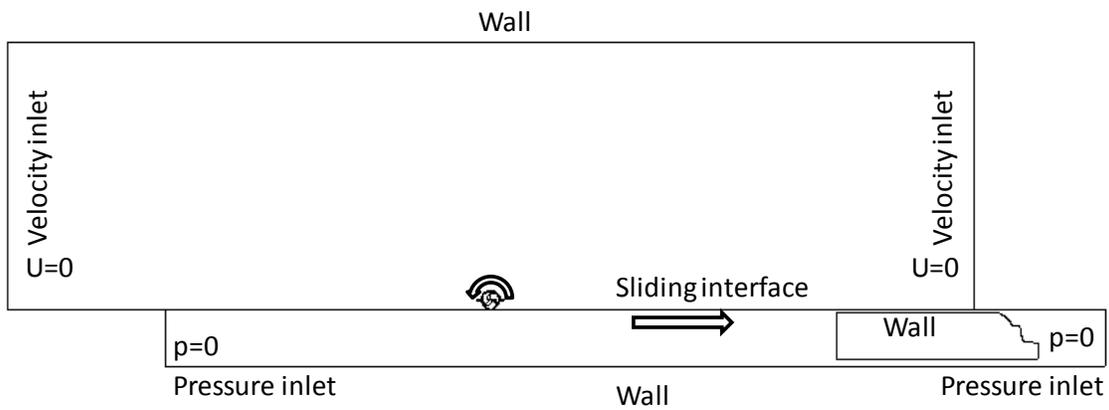

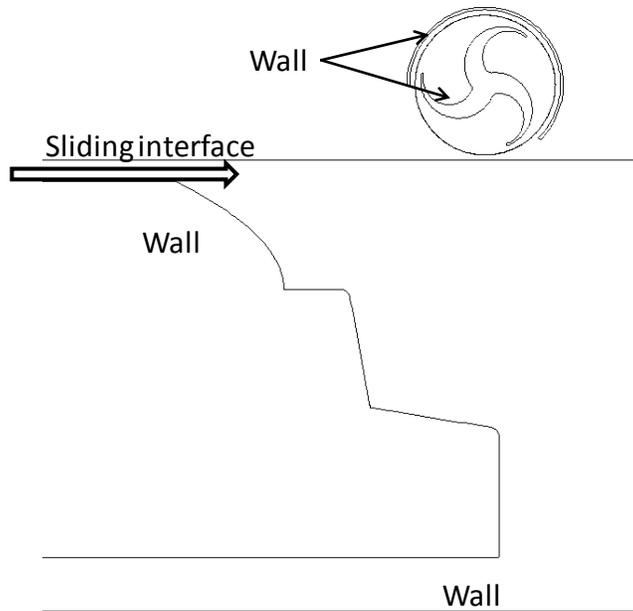**Figure 69 Final position of the Case 2a simulation (225-deg shroud).**

Figure 70 Snapshot of the geometry involved in the truck/turbine interaction for Case 2a (225-deg shroud).

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long, with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 71. The upward blast of air is significantly slowed by the blockage effect in the vertical direction (across the top of the turbine). This leads to a sharp decline in energy output, Figure 72. Evidently, because the direction of the main blast of air is vertical, a typical shrouding design such as in Case 2a will not work. The output of the shrouded turbine is dwarfed by the output of the bare turbine (Case 1c) in every respect.
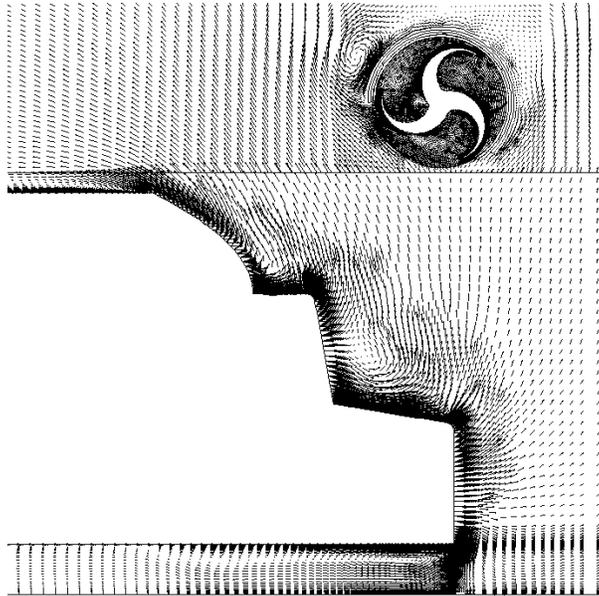

Figure 71 Velocity vectors at the snapshot showing truck/turbine interaction for Case 2a (225-deg shroud).
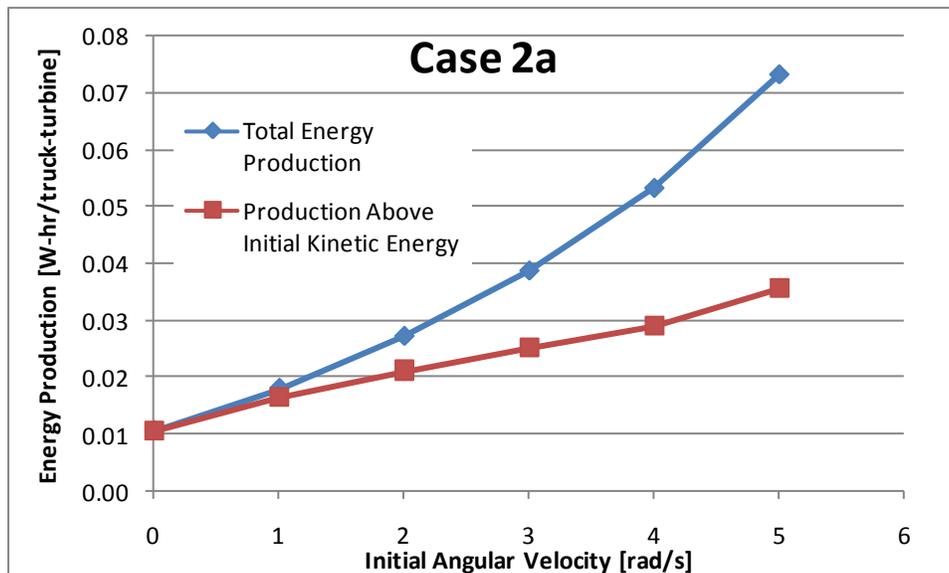

Figure 72 Energy production as a function of initial angular velocity for Case 2a (225-deg shroud).

49

Case 2b

The objective of Case 2b is to use a 180-degree shroud around the turbine and observe the changes in flow pattern and energy output compared with the 225 degree shroud. All conditions mimic the conditions of Case 1c, except that the geometry of the domain is altered to incorporate the shroud. All the baseline parameters of Table 3 are retained (1' clearance, 70 mph truck). It was hoped that, with this design, the negative effect of Case 2a would be diminished, and the benefits of shrouding could be realized.

The quadrilateral grid for Case 2b is displayed in Figure 73. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 77,800 cells, being only a slight variation of the Case 2a mesh. The domain configurations at the beginning and end of the transient simulation are shown in Figures 74 and 75, respectively, consisting of the same three region scheme as the previous cases. Figure 76 shows a closer view of the domain at the point where the truck and turbine interact. The shroud covers the upper half of the turbine from the front and wraps around the top covering half of the rear of the turbine.
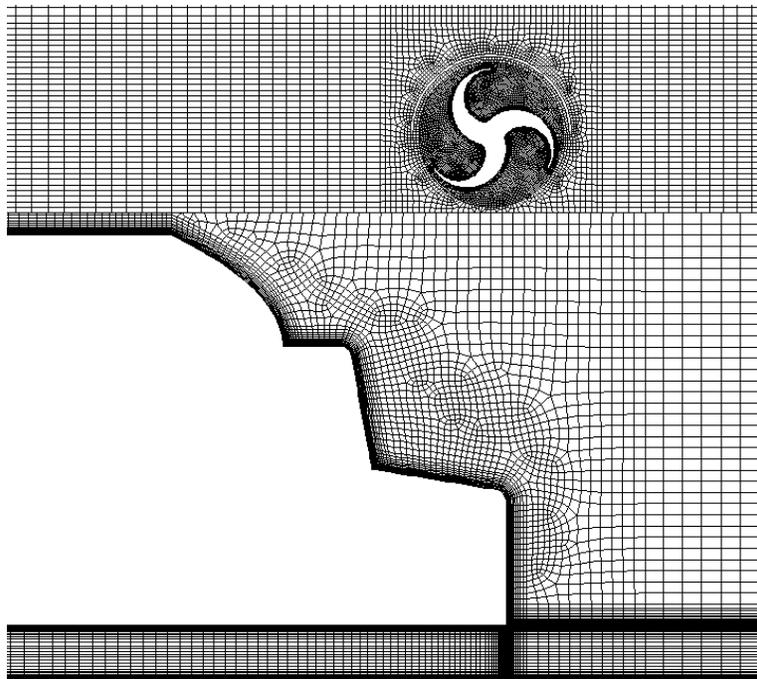

Figure 73 Quadrilateral grid for Case 2b.
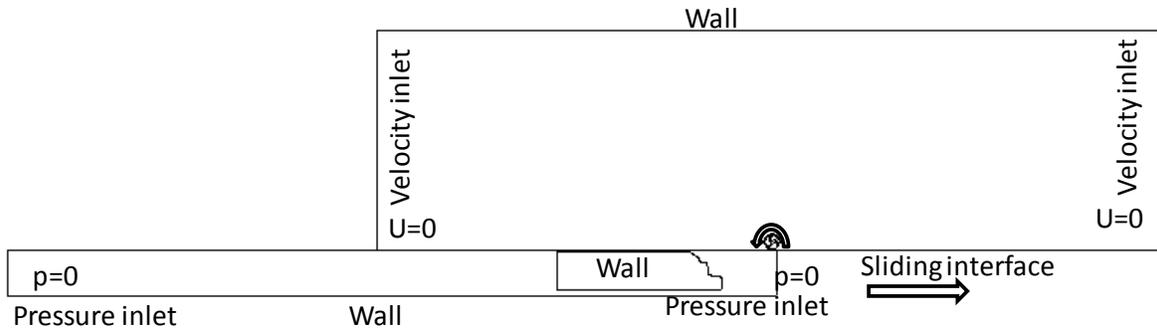
Figure 74 Starting position of the Case 2b simulation (180-deg shroud).
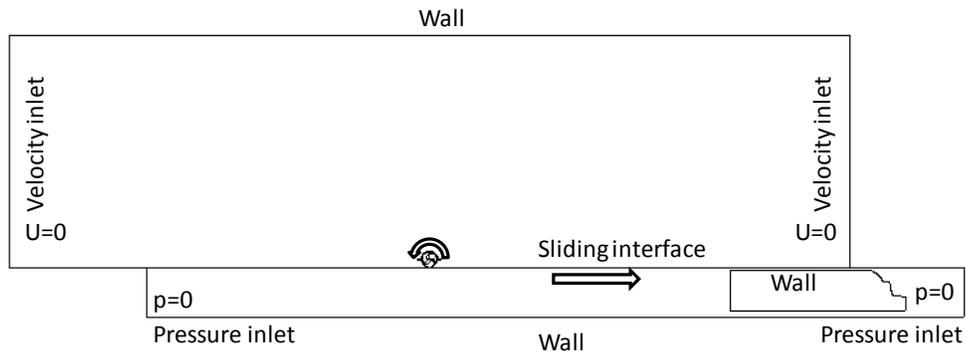
Figure 75 Final position of the Case 2b simulation (180-deg shroud).
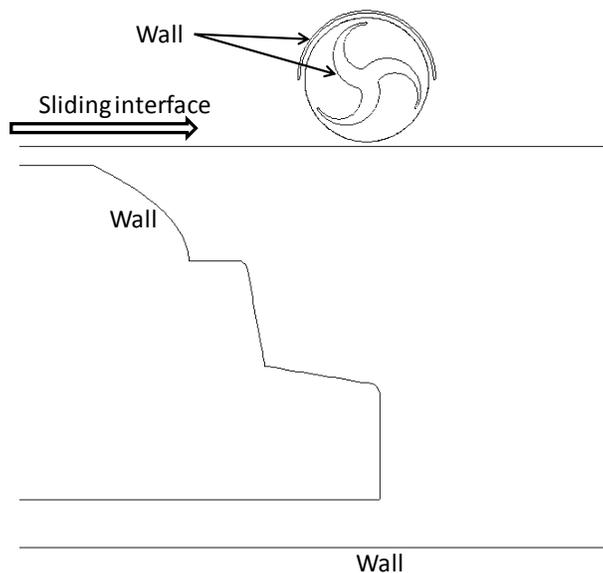
Figure 76 Snapshot of the geometry involved in the truck/turbine interaction for Case 2b (180-deg shroud).

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure

51

77. The upward blast of air is still significantly slowed by the blockage effect in the vertical direction (across the top of the turbine). This leads to a sharp decline in energy output, Figure 78. Case 2b shows little improvement over Case 2a and the main fluid mechanical effect of the shroud is still vastly harmful to the output of the turbine. The output of either shrouded turbine is still dwarfed by the output of the bare turbine (Case 1c) in every respect.
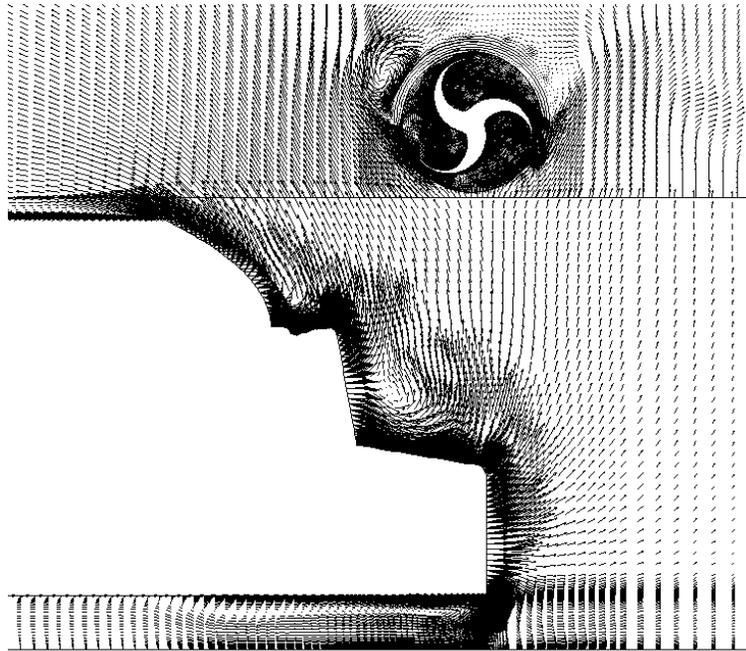

Figure 77 Velocity vectors at the snapshot showing truck/turbine interaction for Case 2b (180-deg shroud).
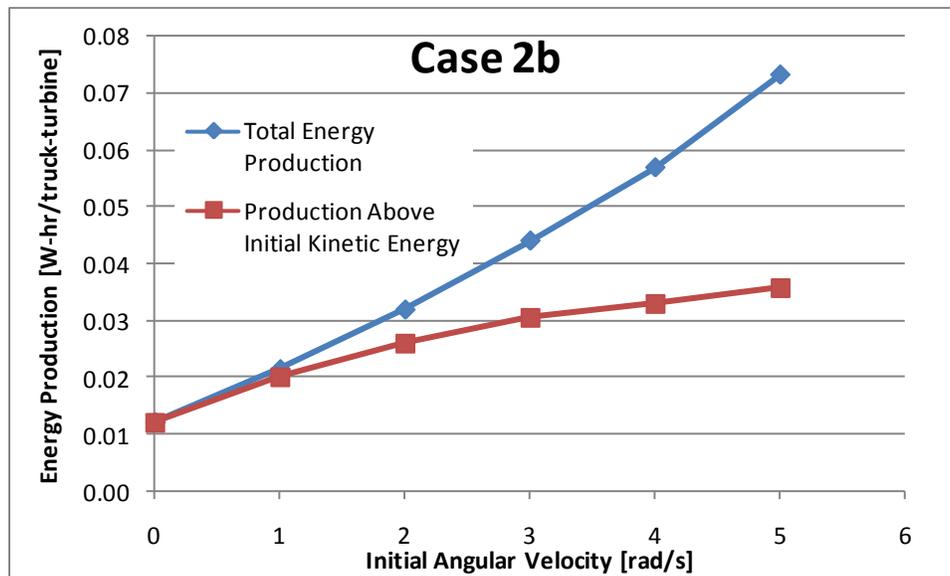

Figure 78 Energy production as a function of initial angular velocity for Case 2b (180-deg shroud).

Task 3

The objective of Case 3 (the only case within Task 3) is to show the effect of extending the shroud of Case 2b to include a hooding effect. This could, ideally, combat the negative blockage effect seen in Task 2. All conditions mimic the conditions of Case 1c, except that the geometry of the domain is altered to incorporate the shroud. All the baseline parameters of Table 3 are retained (1' clearance, 70 mph truck).

The quadrilateral grid for Case 3 is displayed in Figure 79. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 77,800 cells, being only a slight variation of the Task 2 meshes. The domain configurations at the beginning and end of the transient simulation are shown in Figures 80 and 81, respectively, consisting of the same three-region scheme as the previous cases. Figure 82 shows a closer view of the domain at the point where the truck and turbine interact. The hood covers the upper half of the turbine from the front and wraps around the top covering half of the rear of the turbine. The hood, in the case of multiple turbines, would extend to the next turbine continuously. In the case of one turbine, it has been cut off as shown in Figure 82.
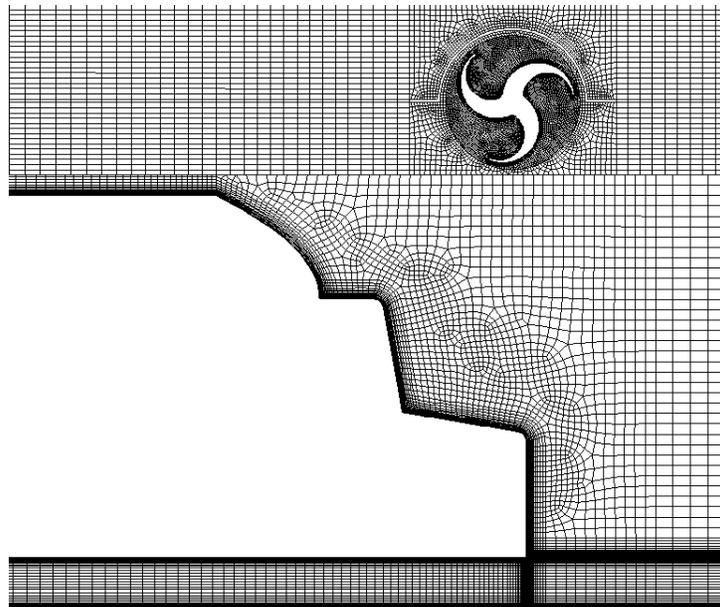

Figure 79 Quadrilateral grid for Case 3 (hooded).
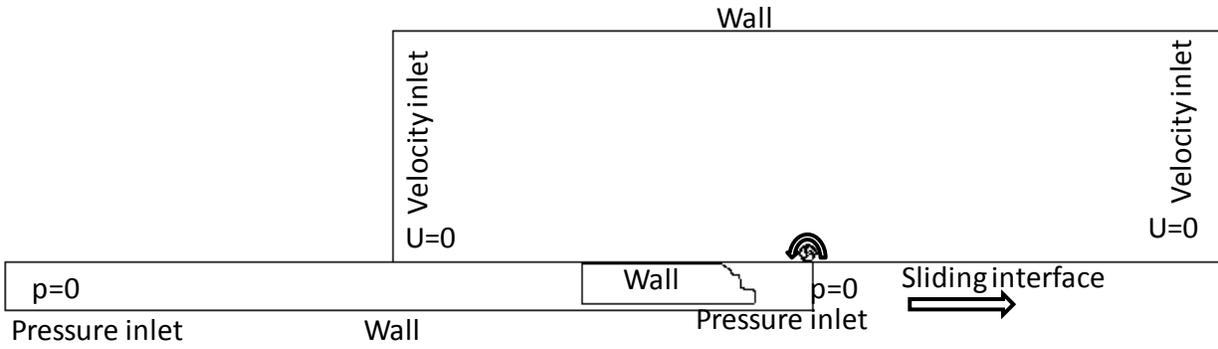
Figure 80 Starting position of the Case 3 simulation (hooded).

Figure 81 Final position of the Case 3 simulation (hooded).
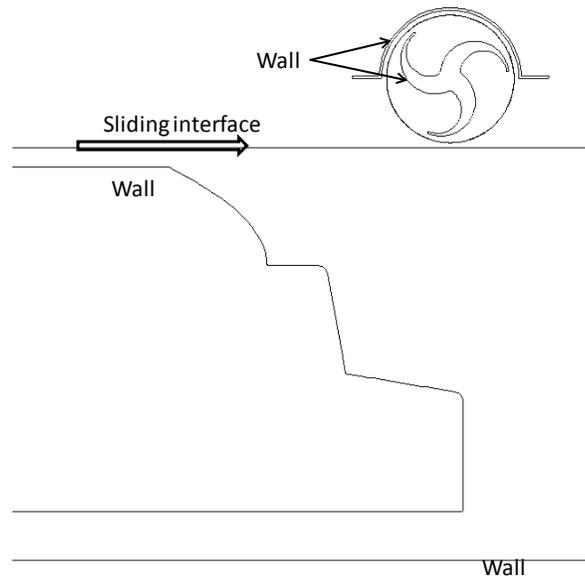
Figure 82 Snapshot of the geometry involved in the truck/turbine interaction for Case 3 (hooded).

54

The simulation duration began with 40 time steps of 0.0003 seconds to assure stability. The bulk of the simulation was 1.5 seconds long with 1500 time steps of 0.001 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 83. The upward blast of air is still significantly slowed by the blockage effect in the vertical direction (across the top of the turbine). This leads to the sharp decline in energy output noticed in Task 2, Figure 78. Case 3 shows no noticeable improvement over Case 2b and the main fluid mechanical effect of the shroud is still vastly harmful to the output of the turbine. The output of the hooded turbine is still dwarfed by the output of the bare turbine (Case 1c) in every respect.
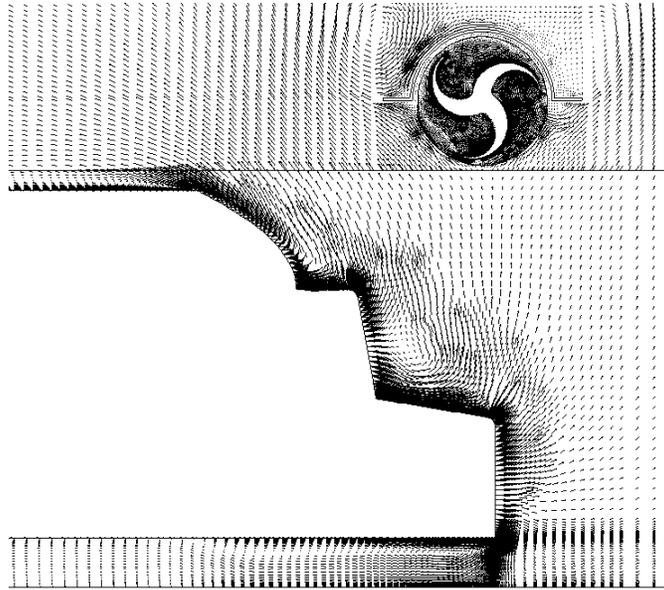
Figure 83 Velocity vectors at the snapshot showing truck/turbine interaction for Case 3 (hooded).
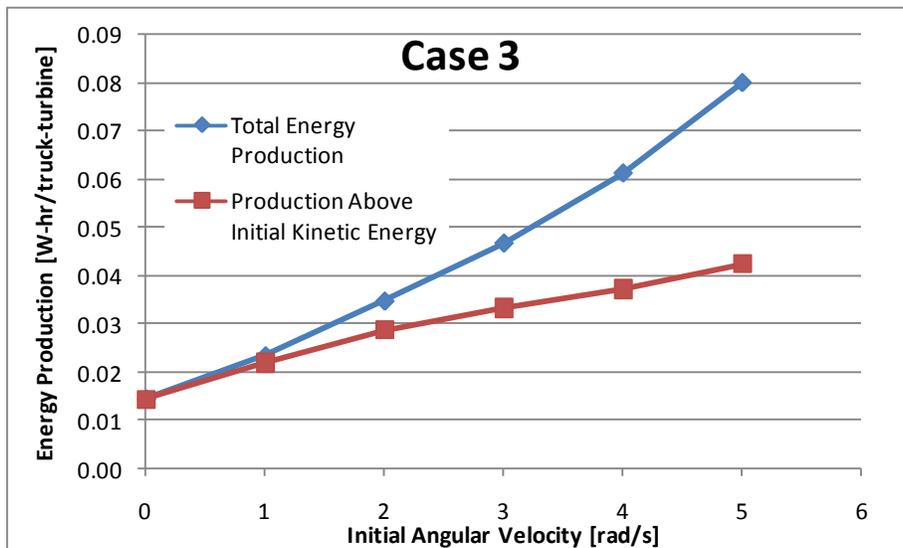
Figure 84 Energy production as a function of initial angular velocity for Case 3 (hooded).

Task 4

Task 4 extends the methodology from the first three tasks to the case with a turbine alongside a High-Speed Rail (HSR). The turbine output is simulated at two different clearances from the passing train.

Case 4a

Case 4a represents the HSR equivalent to the baseline Case 1c. The train is simulated at a speed of 175 mph, with a noticeable geometry change not only due to the change from the truck to train shape, but also in consideration that the turbine is to the side of the passing train, as opposed to above the passing truck. All other baseline parameters of Table 3 are retained (1' clearance).

The quadrilateral grid for Case 4a is displayed in Figure 85. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer. The total mesh consisted of about 91,700 cells, noticeably more than the previous meshes because of the larger domain on the opposite side of the train from the turbine (which was bounded by the ground in the case of the truck simulations). The domain configurations at the beginning and end of the transient simulation are shown in Figures 86 and 87, respectively, consisting of the same three region scheme as the previous cases. Figure 88 shows a closer view of the domain at the point where the train and turbine interact.

Figure 85 Quadrilateral grid for Case 4a (HSR at 1').

Figure 86 Starting position of the Case 4a simulation (HSR at 1').
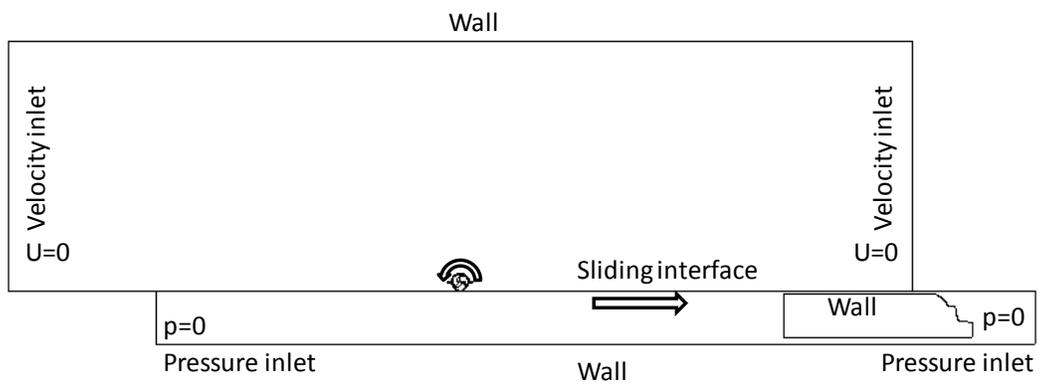

Figure 87 Final position of the Case 4a simulation (HSR at 1').

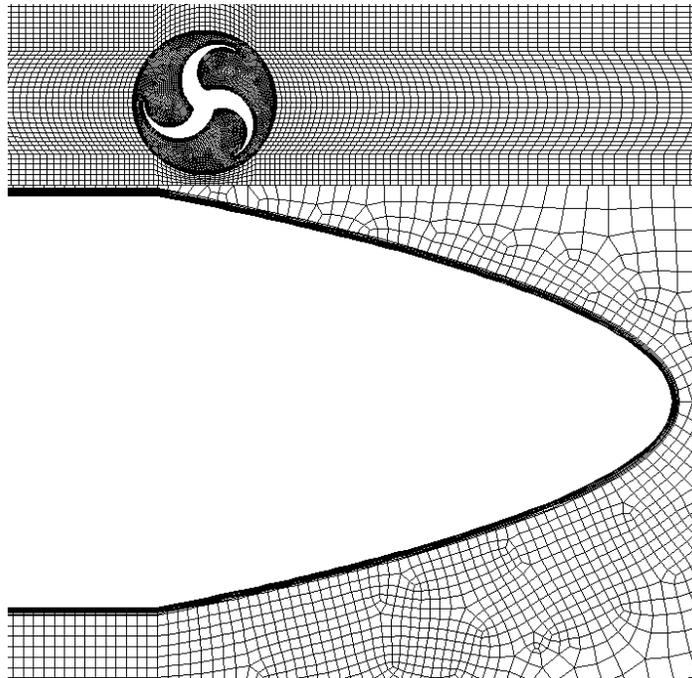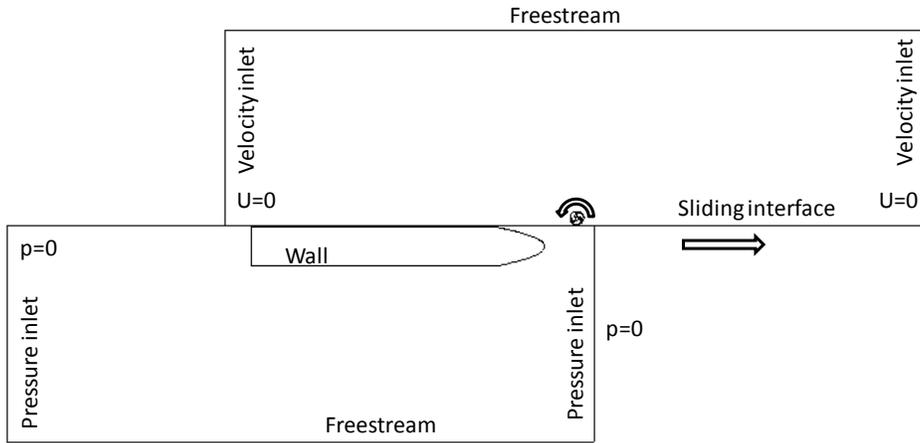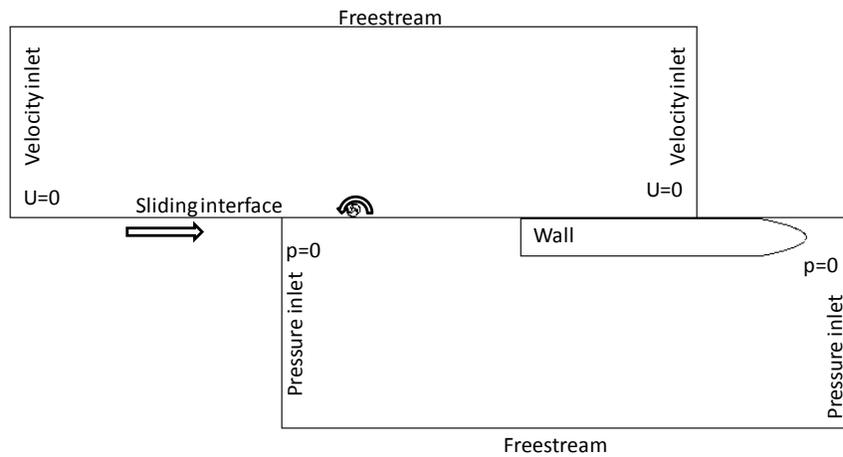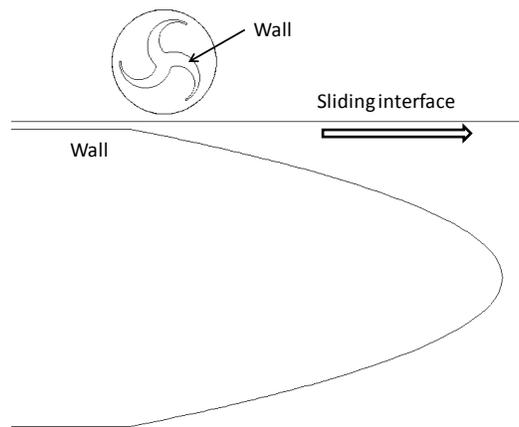**Figure 88 Snapshot of the geometry involved in the HSR/turbine interaction for Case 4a (HSR at 1').**

The simulation was ~0.5 seconds long with 1540 time steps of 0.0003 seconds. Velocity vectors captured at a time of high interaction between the truck and turbine are shown in Figure 89. The perpendicular blast of air is still the main force in energizing the turbine. As with the truck-driven outputs, the train is helped when an initial angular velocity is given to the turbine. The steady increase in output with increasing initial angular velocity from ambient wind is seen in Figure 90.

With the increase in velocity, it might be expected that the train-driven turbine would well outperform its truck-driven counterpart. This is not seen to be as dramatic as anticipated. The output is about doubled when switching from the truck to the train. A number of factors contribute to this. First, the truck pushed the majority of the air toward the top, as it was strictly bounded by the ground in the other direction. The train pushes the air to either side equally, cutting down on the effective momentum given to the air blast experienced by the turbine. Also important is the comparative aerodynamic shaping of the truck and train. The train reflects a much more carefully streamlined geometry when compared to the truck. This is a key feature of high speed trains, since the drag force represents a serious limiting factor in the energy efficiency of vehicles traveling at such high speeds. For this reason, less momentum is transferred from the train to the air in the current case than would be if the high-speed train had the more blunt geometry of the truck. These two primary factors limit the increase in energy output when comparing the 70 mph truck with the 175 mph train. A final consideration is the duration of the numerical simulation, which may have truncated the turbine's ability to capture the spreading wake of the train (similar to Case 1a). Therefore, it is possible that the train simulation represents a significant under-prediction.

Figure 89 Velocity vectors at the snapshot showing HSR/turbine interaction for Case 4a (HSR at 1').
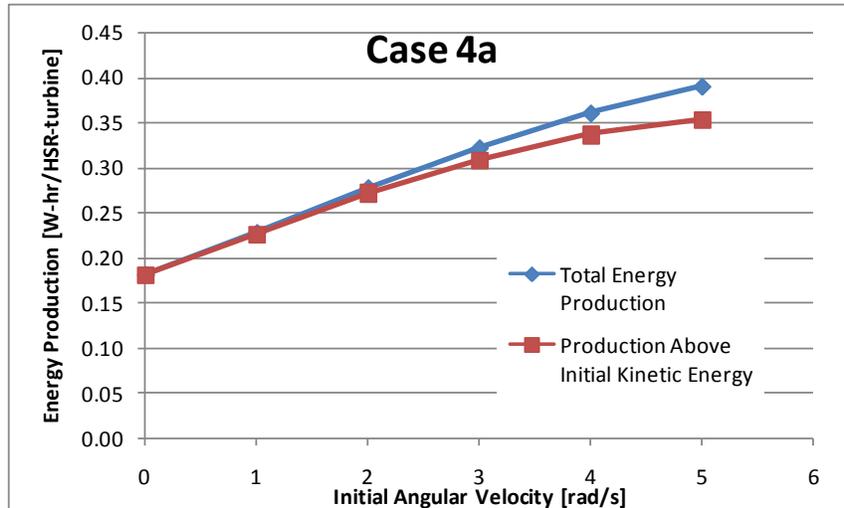
Figure 90 Energy production as a function of initial angular velocity for Case 4a (HSR at 1').

Case 4b

Case 4b moves the turbine to a clearance of 0.5 feet from the train. The other parameters of the simulation are kept identical to the Case 4a parameters. The quadrilateral grid for Case 4b is displayed in Figure 91. Again, the near-wall regions are clustered tightly with cells to resolve the viscous sub-layer for both the turbine and the train. The total mesh consisted of about 84,600 cells, being a slight reduction from the Case 4a mesh. The domain configurations at the beginning and end of the transient simulation are shown in Figures 92 and 93, respectively, consisting of the same three-region scheme as the previous cases. Figure 94 shows a closer view of the domain at the point where the train and turbine interact.
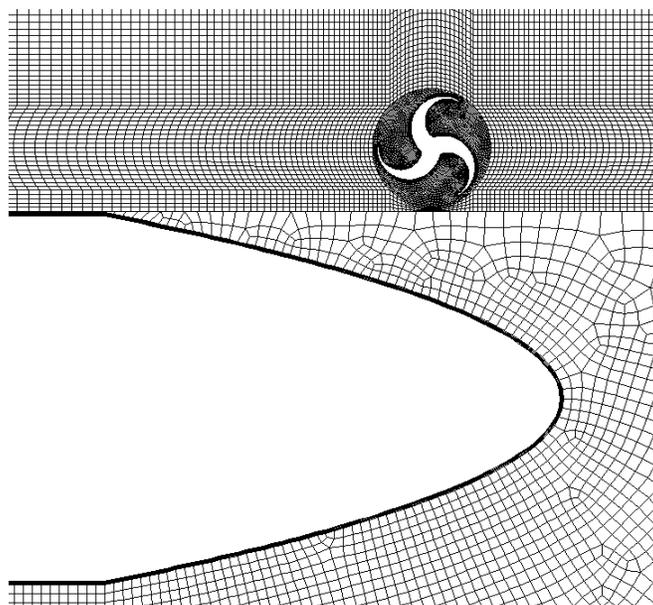
Figure 91 Quadrilateral grid for Case 4b (HSR at 6").
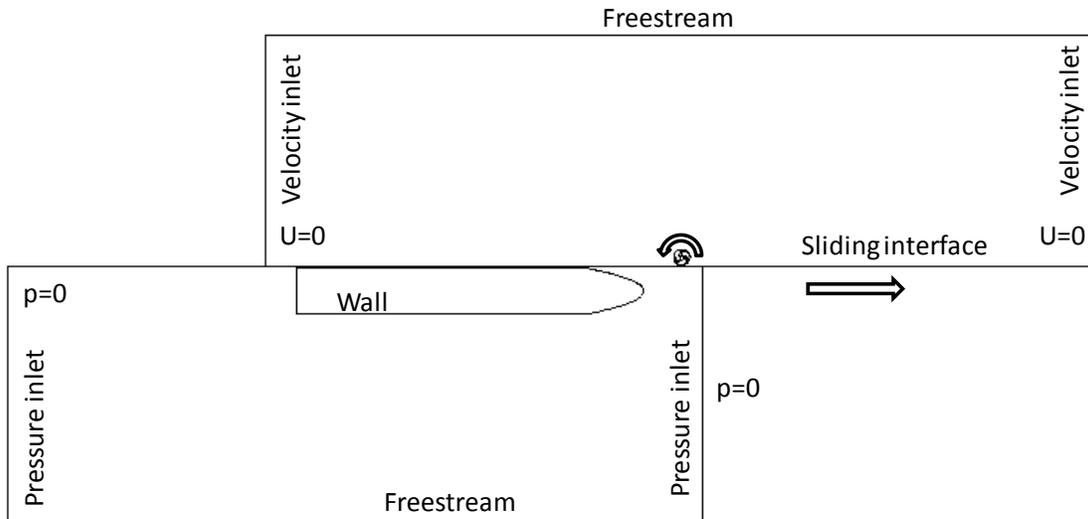
Figure 92 Starting position of the Case 4b simulation (HSR at 6").
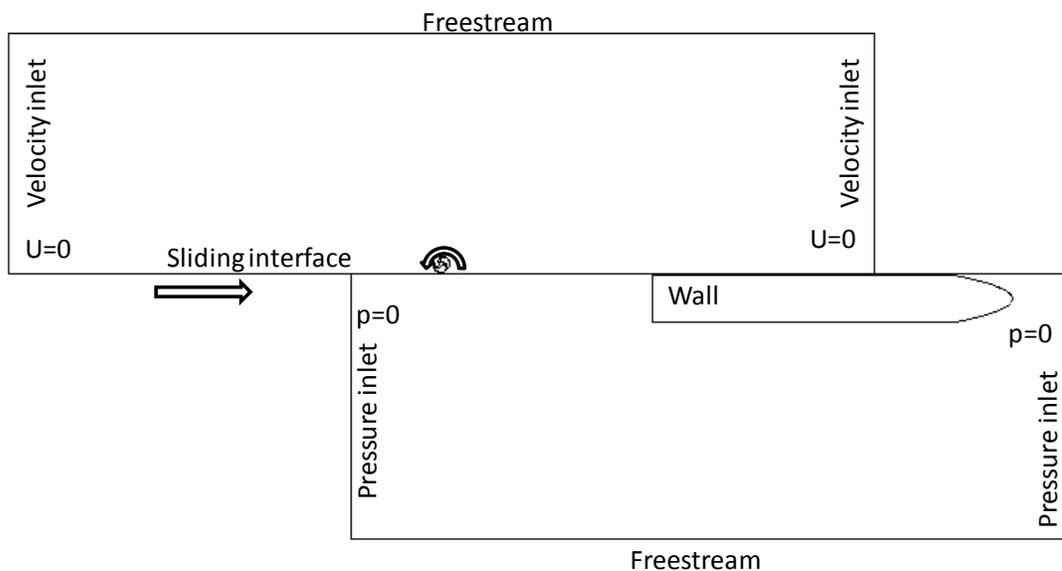
Figure 93 Final position of the Case 4b simulation (HSR at 6").

The simulation was ~0.5 seconds long with 1540 time steps of 0.0003 seconds. Velocity vectors captured at a time of high interaction between the HSR and turbine are shown in Figure 95. The perpendicular blast of air is still the main force in energizing the turbine. As with the truck driven outputs, the train output is helped when an initial angular velocity is given to the turbine. The steady increase in output with increasing angular velocity is seen in Figure 96. Case 4b shows that the output is close to doubled by moving the turbine from 1 foot to a clearance of 0.5 feet. This result is consistent with the results of clearance effect studies for the passing truck. Case 4b interestingly shows that the energy production levels off above an initial turbine angular velocity of around 2 rad/s.

60
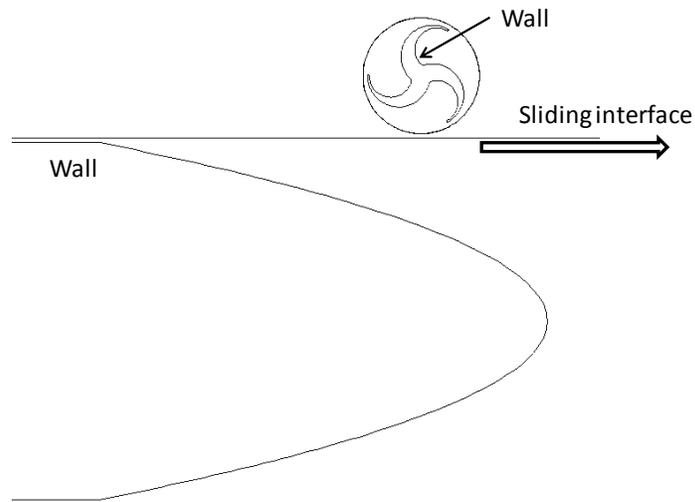
Figure 94 Snapshot of the geometry involved in the HSR/turbine interaction for Case 4b (HSR at 6").
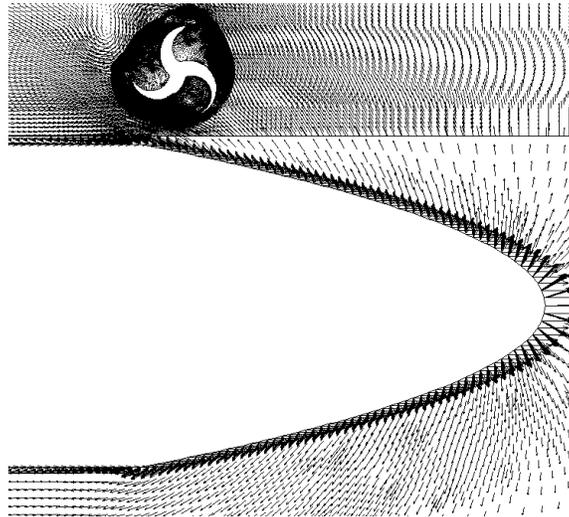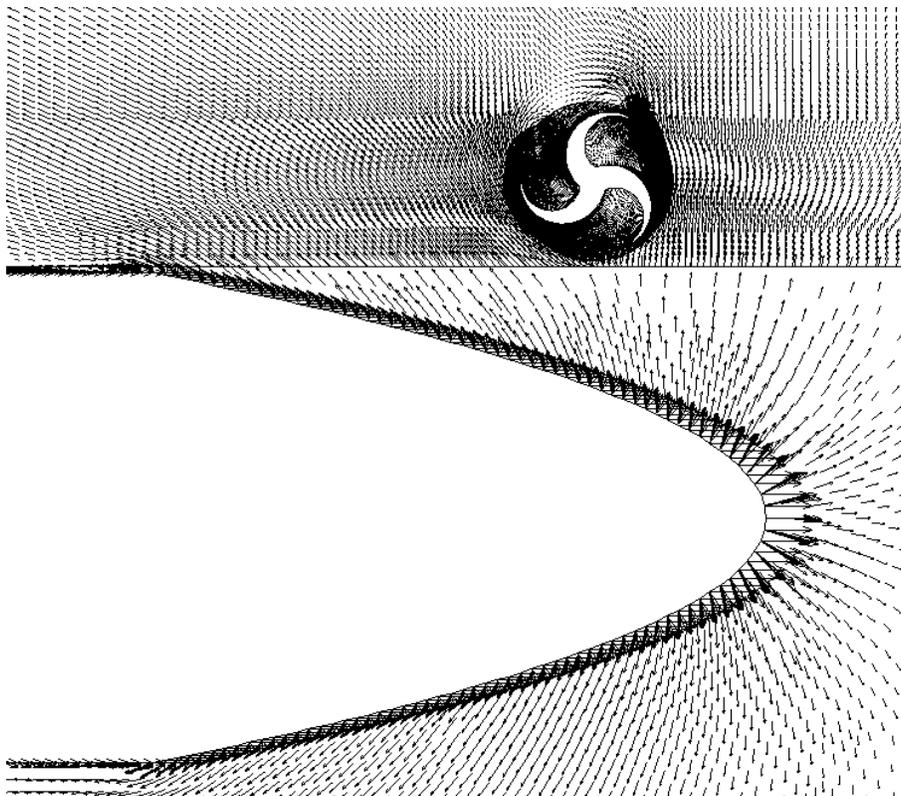

Figure 95 Velocity vectors at the snapshot showing HSR-turbine interaction for Case 4b (HSR at 6").
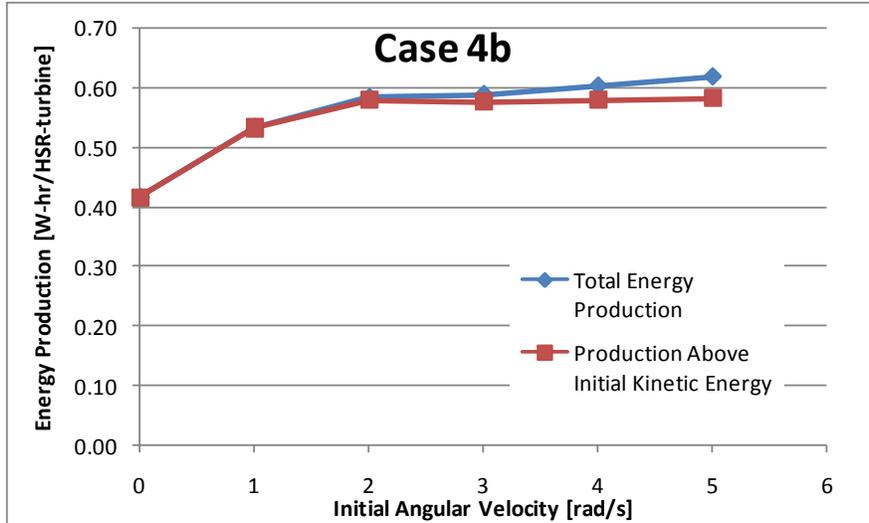
Figure 96 Energy production as a function of initial turbine angular velocity for Case 4b (HSR at 6").

Discussion

The various effects studied in this project are summarized and discussed in this section. Cases 1a, 1b, and 1c studied the effect of the numerical domain extent on the prediction for a single bare turbine with 1 foot of clearance experiencing the air movement induced by the passing of a 12-meter-long tractor-trailer at 70 mph. Cases 1d and 1e explored the effect of changing the distance between the truck and the turbine. Cases 1f and 1g determined the impact of changing the truck's velocity as it passed the turbine. Case 1h studied the effect of one turbine on another in a closely spaced arrangement. Cases 2a, 2b, and 3 showed the effect of attempts to geometrically enhance the performance of the turbine. Finally, Case 4 extended the scenario to a high speed rail configuration with a velocity of 175 mph.

The results of the first three simulations illustrate the strong dependence of the CFD prediction on the numerical domain. For the two-dimensional assumption made in this report, such dependence could be disheartening (Figure 97). However, Case 1c represents the attempt to capture the three-dimensional effect of the surroundings in two dimensions. The ambient wind-induced angular momentum for initial conditions shown in the following charts is 3 rad/s, which represents a tip speed ratio of 0.5 in an 8 mph wind.

62
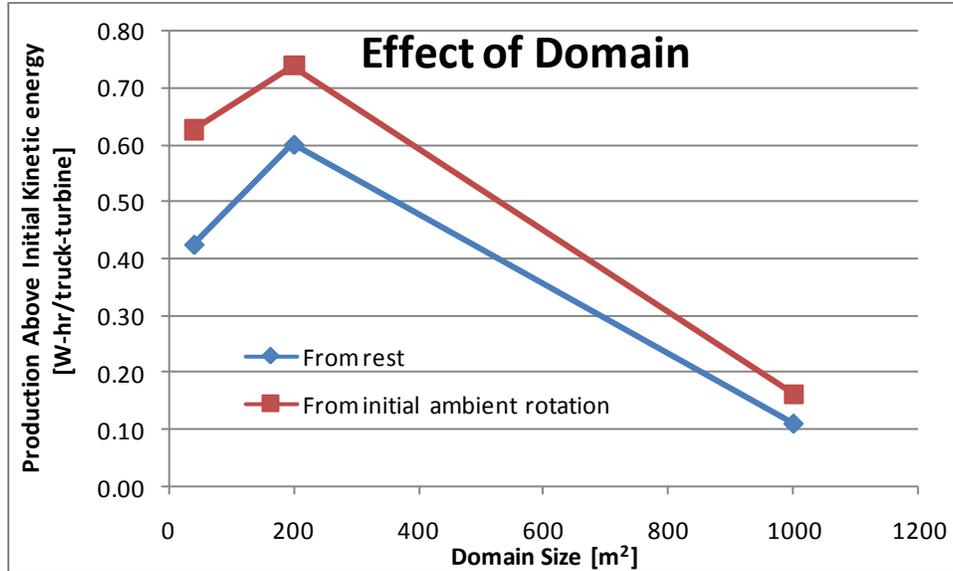
Figure 97 Effect of domain size on the predicted energy output.

Cases 1c, 1d, and 1e represent three difference clearance heights between the truck and turbine, with all other parameters held constant (Figure 98). The trend is as expected, with the closer turbine producing the most power. The effect appears to be highly non-linear for the three cases studied. In other words, the closer the turbine is to the truck, the higher its sensitivity to changes in clearance.
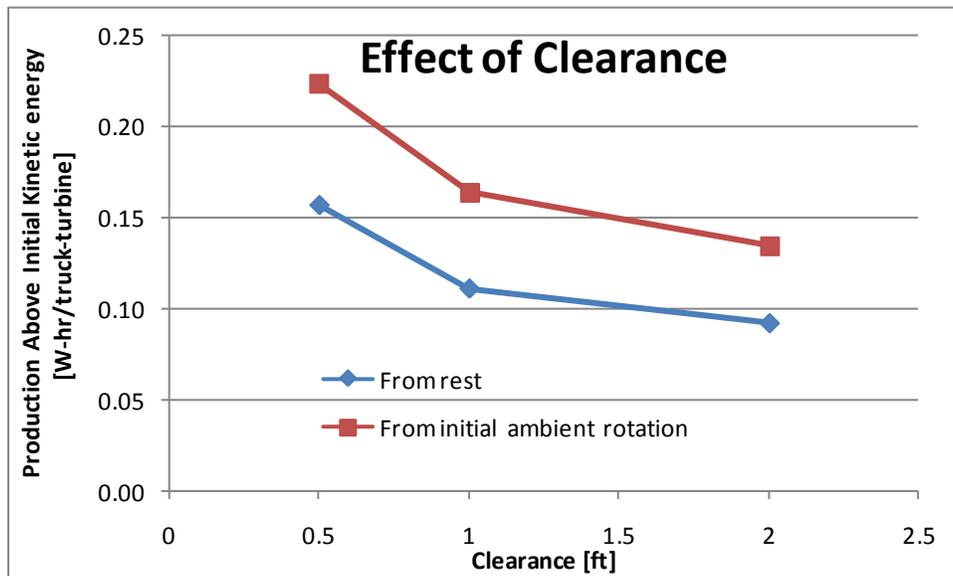
Figure 98 Effect of clearance on the predicted energy output.

Similarly, the effect of the oncoming truck's velocity followed expected trends. Case 1c, 1f, and 1g represent three simulations with different truck velocities, but with all other parameters held constant. Although the trend appears to be linear in Figure 99, this is merely a visual effect, and the numbers indicate a stronger dependence on the truck velocity.
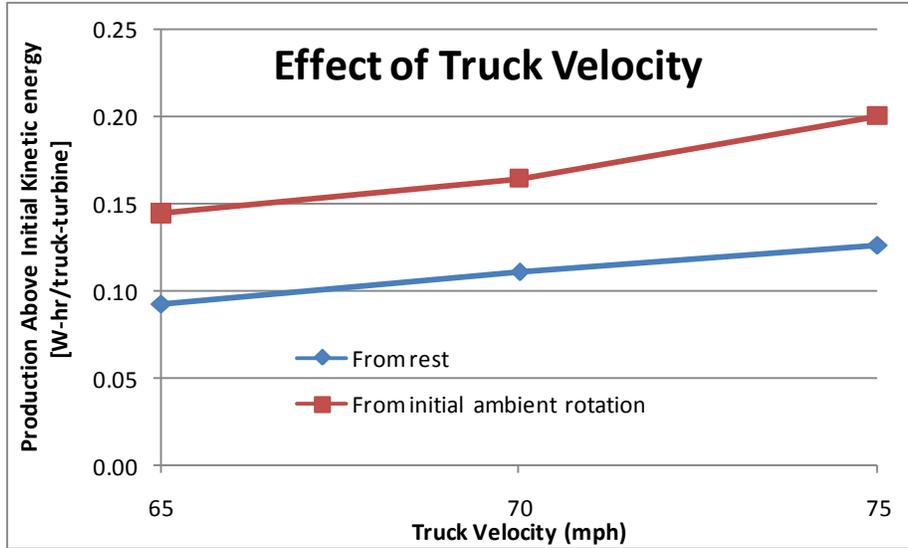
Figure 99 Effect of truck velocity on the predicted energy output.

Case 1h added a second turbine and recorded the change in the energy production caused by the interaction of multiple turbines. There was a significant decrease in output in this case, Figure 100. To accommodate multiple turbines, either generous spacing should be provided, or the turbines should be staggered such that they interfere minimally with each other.
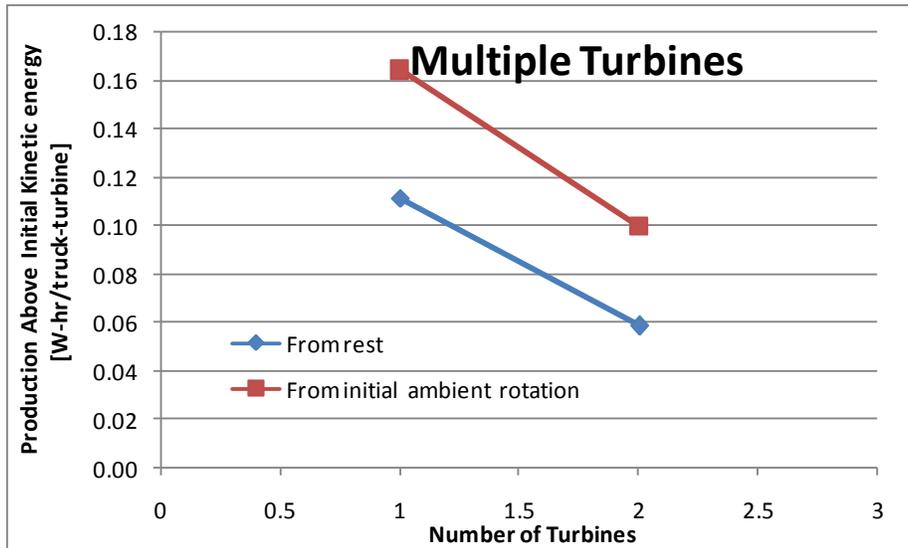
Figure 100 Effect of having multiple turbines closely spaced.

Cases 2a, 2b, and 3 studied the effect of shrouding the turbine. The parameters for these simulations are the same as Case 1c, to which it is compared in Figure 101. The effect of shrouding is highly negative in all simulations. The blockage of any flow direction, due to the transient and oscillatory pattern of flow induced by the truck-turbine interaction, can be assumed to be harmful to the output of the turbine. The results of this study strongly suggest against any attempt to geometrically enhance the output of a wind turbine in this scenario, especially when coupled with the opportunity to make use of ambient wind from any direction.

64
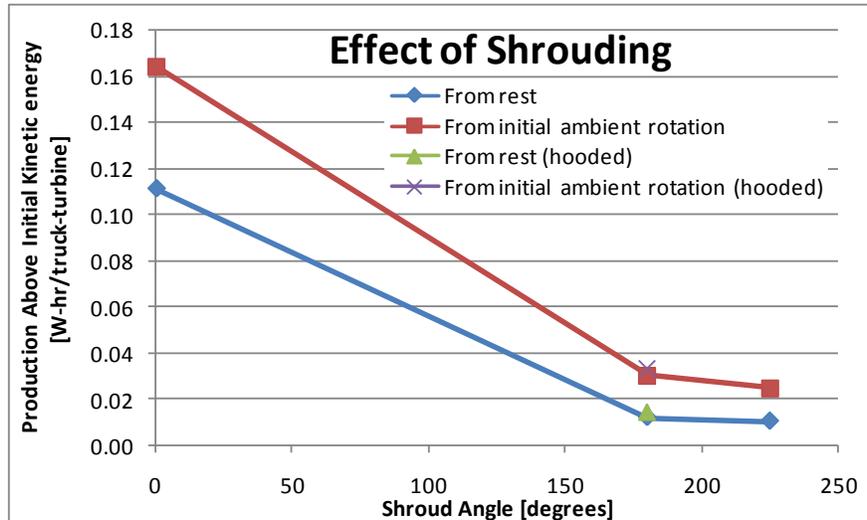
Figure 101 Effect of shrouding on the predicted energy output.

Cases 4a and 4b studied the clearance effect of the turbine in the High Speed Rail scenario. As in the case of the truck, the closer turbine collected more energy than the one further from the train, Figure 102. The increase in energy production is mostly caused by the increase in velocity, but this is diminished by the more streamlined body of the high-speed train (compared with the truck) and the two-sided nature of the scenario with the turbine placed to one side. Clearance again is an important parameter controlling the output of the turbine. The train was 30 meters long, twice the length of the truck.
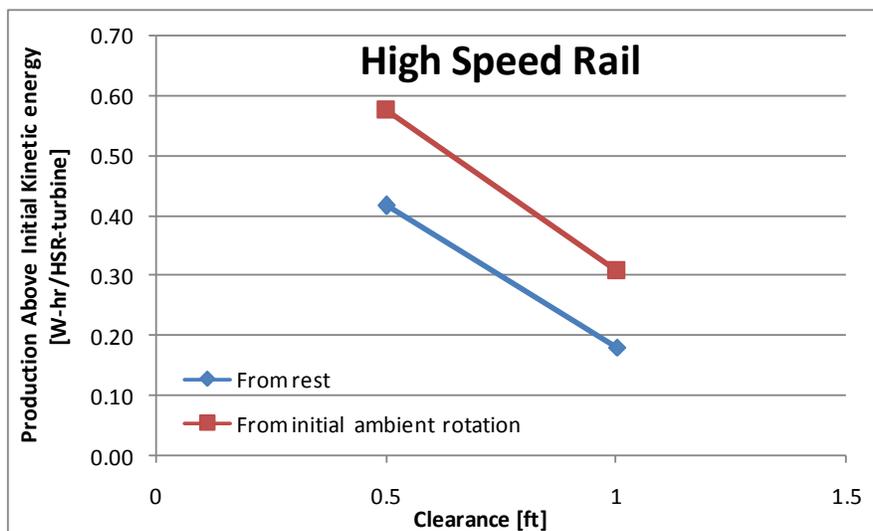
Figure 102 Effect of clearance on the predicted energy output next to high speed train.

An important concern for the implementation of this design is the increase in drag the turbine may induce on the passing vehicle, resulting in increased energy costs for the vehicle. This lowers the effective net energy output of the turbine. This topic was not rigorously addressed in the current methodology and is deferred to future studies.

Conclusion

A parametric CFD study of a traffic driven wind generator was carried out with two-dimensional meshes using the Fluent unsteady RANS solver with the k-omega SST turbulence model [4]. The baseline output of the turbine was about 0.1 W-hr/truck-turbine with no initial angular velocity of the turbine (which has a cut-in at 2.8 m/s of steady wind). The following conclusions were found through this study:

- Initial angular momentum allowed the turbine to produce more energy over and above the original kinetic energy because of the cut-in effect from the generator. Therefore, the ideal implementation of this system would be in places of high traffic (where passing trucks will be closely spaced) or high average ambient wind.
- The predictions on the CFD results were highly sensitive to the extent of the numerical domain.
- The turbine is able to increase its energy output when placed closer to the passing vehicle. This is a non-linear effect, so close placement (0.5 feet) is highly desirable for increasing energy output.
- The turbine output is also sensitive to the speed of the passing vehicle, with a significant improvement when increased to 75 mph over the baseline 70 mph truck. This effect, likewise, is non-linear, probably owing to the fact that output increases with the cube of velocity. The average output of the turbine over a number of trucks would not necessarily be the output of a truck traveling at the average velocity.
- A second turbine is seen to produce only half of the power as the first turbine when spaced one diameter (4ft) apart. Further spacing and/or staggering of multiple turbines should allow for the diminishing of this effect.
- Adding any shrouding, hooding, or ducting in an attempt to increase the ability of the turbine to collect the energy results instead in decreased output. The blockage of the flow in any direction is detrimental to the flow around the blades.
- A High Speed Rail produces more energy than a passing tractor-trailer truck due to its significantly higher speed. This is diminished by geometry and surroundings effects, however. In the HSR scenario, the turbines could be place above and beside the passing train, extending the ability of the TDWG system to produce energy.
- The domain extent as well as the two-dimensionality represented a trade-off between accuracy and computational expense. The scope of the current project focused on a parametric study of the physics rather than a high-fidelity model. Future investigations should aim to employ a three-dimensional domain and robust boundary conditions.

References

[1] "CFD Analysis of a Vertical Axis Wind Turbine" SATOP RTA #3498

[2] Blackwell, B.F., Sheldahl, R.E., and Feltz, L.V., 1977, "Wind Tunnel Performance Data for Two- and Three-Bucket Savonius Rotors," Sandia Laboratories Report, SAND76-0131.

[3] Tennekes, H, and Lumley, J. L., *A First Course in Turbulence*, MIT Press, 1972.

[4] Menter, F., 1993, "Zonal Two Equation k-ω Turbulence Models for Aerodynamic Flows", AIAA Paper 93-2906.

[5] Gambit, 2008, "Gambit Version 2.3.16", USA, ANSYS Inc

[6] Ansys Fluent, 2008, "Ansys Fluent Version 12", USA, ANSYS Inc

Appendix A: Solid Body Kinematics Code

C language code

```c
#include <stdio.h>

#include <math.h>


#define addedlength 0


char filename[50] = "data.txt";

char filenamed[50] = "drag.txt";

char filenamewr[50] = "output.txt";

//char filenamewr[50] = "outputlong.txt";

double function(double, double, double, double*);

double functiond(double, double, double, float*);

double *sim(double, double, double*, float*, FILE *fp);

double dt = 0.001;

double Vt = 31.3;

// omega step size and number of steps

double dw = 4.;

int wMax = 12;

// number of time steps (and x-locations)

int tSteps = 1540;

int xMax;

// initial theta steps and step size

int qMax = 6;

double dq = 20.;
```

```c
// kick in rotational velocity

double wkickin = 2.8;

double ans[2];


main()

{

  double *ptr, *ptr2, *ptr3;

  /* Read turbine torques*/

  xMax = tSteps;

  int vectLength = xMax*(qMax+1)*wMax;

  printf("%i\n", vectLength);

  double data[vectLength];

  float drag[vectLength];

  double number;

  FILE *fp;

  if((fp=fopen(filename, "r"))!= NULL)

  {

        printf("File opened successfully!\n");

        int size = 0;

        while(fscanf(fp,"%lf",&number) != EOF)

        {

                //printf("problem not yet %i\n", size);

                data[size] = number * 6.5268;

                size++;

        }
```

```
        fclose(fp);

        //for(t=0;t<size;t++)

        //{

        //      printf("%lf\n", data[t]);

        //}

        printf("size=%i\nexpectedsize=%i\n", size, vectLength);

        system("PAUSE");

        if(size!=vectLength)

        {

                printf("The input vector is not the right size!\n");

                system("PAUSE");

                exit(1);

        }

        else

        {

         printf("Data loaded successfully!\n");

        }

}

else

{

 printf("Unable to open file...\n");

 system("PAUSE");

 exit(1);

}

fclose(fp);
```

```c
if((fp=fopen(filenamed, "r"))!= NULL)

{

        printf("File (drag) opened successfully!\n");

        int size = 0;

        while(fscanf(fp,"%lf",&number) != EOF)

        {

                //printf("problem not yet %i\n", size);

                drag[size] = (number-525.0) * 10.878;

                //if(size==135){printf("drag = %.3f", drag[size]);}

                size++;

        }

        fclose(fp);

        //for(t=0;t<size;t++)

        //{

        //        printf("%lf\n", data[t]);

        //}

        printf("size=%i\nexpectedsize=%i\n", size, vectLength);

        system("PAUSE");

        if(size!=vectLength)

        {

                printf("The input drag vector is not the right size!\n");

                system("PAUSE");

                exit(1);

        }

        else
```

```c
        {
          printf("Drag loaded successfully!\n");

        }

}

else

{

 printf("Unable to open drag file...\n");

 system("PAUSE");

 exit(1);

}

fclose(fp);

if((fp=fopen(filenamewr, "w")) != NULL)

{

        printf("File successfully opened for writing!\n");

        system("PAUSE");

}

else

{

 printf("Unable to open writing file...\n");

 system("PAUSE");

 exit(1);

}

int k;

int sizer = 120/10;

double thetain[sizer], powerout[sizer], dragin[sizer];
```

```c
double sumpower = 0, sumloss=0;

int wi;

for(wi = 0; wi < 6; wi++)

{

    sumpower=0.0;

    sumloss = 0.0;

    for(k=0;k<sizer;k++)

    {

        thetain[k] = 5 + 120 / sizer * k;

        //fprintf(fp,"Theta = %lf:\n", thetain[k]);

        ptr = sim(wi*1.0, thetain[k], data, drag, fp);

        //powerout[k] = sim(0.0, thetain[k], data);

        //dragin[k] = sim(0.0, thetain[k], drag);

        powerout[k] = ptr[0];

        dragin[k] = ptr[1];

        sumpower += powerout[k];

        sumloss += dragin[k];

    }

double avgpower = sumpower / k;

double avgloss = sumloss / k;

fprintf(fp, "Run#\tInitial Angle\tPower Produced\tDrag Power Loss\n");

for(k=0;k<sizer;k++)

{

        fprintf(fp,"%i\t%e\t%e\t%e\n", k, thetain[k], powerout[k], dragin[k]);

}
```

73

```c
    fprintf(fp, "Initial rotational velocity of %.1lf radians per second\n", wi*1.0);

    fprintf(fp, "Overall power prediction:\n%e\n", avgpower);

    fprintf(fp, "Overall drag loss prediction:\n%e\n\n", avgloss);

    //printf("%.3f\n", functiond(16.0,4.0,20,drag));

    //system("PAUSE");

    }

    fclose(fp);

    return 0;

}


double *sim(double omega, double theta, double *data, float *drag, FILE *fp)

{

    int t,k,iter;

    double Mwind=0., Mfr=1., Mgeno=130., J = 10.8, x=0, Mgen, omegaguess, Mnet, power=0.,
omegaold, Mgenold, Mwindold;

    float sumdrag=0., bdrag=0., Fdrag=0.;

    // read data vector from file

    // specify initial conditions

    // double omega = 0;

    // double theta = 5;

    double th;

    // time step loop

    for(t=100;t<tSteps;t++)

    {

            x = Vt*t*dt;
```

```
//printf("Iterating...\n");

th = theta - 120.*floor(theta/120.);

//printf("\n%lf\n", th);

omegaold = omega;

Mwindold = Mwind;

Mgenold = Mgen;

for(iter=0;iter<5;iter++)

{


        Mwind = function(omega, th, x, data);

        //Mwind = 0.9 * Mwind + 0.1 * Mwindold;

        //printf("looking up");

        Mgen = Mgeno * (omega-wkickin) / (11.52-wkickin);

        //Mgen = 0.9 * Mgen + 0.1 * Mgenold;

        //printf("%lf\t%lf\n", omega, Mwind);

        if(omega>wkickin)

        {

                Mnet = Mwind - Mfr - Mgen;

                omega = omegaold + dt*(Mnet)/(J);

        }

        else

        {

          Mgen = 0;

          Mnet = Mwind - Mfr;

          omega = omegaold + dt*(Mnet)/(J);
```

75

```
        }

        if(omega<0)

        {

            omega = 0.;

        }

        if(omega>(dw*(wMax-1)))

        {

            omega=44.0;

            //printf("rotational velocity exceeded!\nomega=%f\n", omega);

            //system("PAUSE");

            //exit(1);

        }

    }

    Fdrag = functiond(omega, th, x, drag);

    //printf("dragforce = %f", Fdrag);

    theta += omega*180/6.283*dt;

    power += omega*Mgen*dt;

    if(t<640){sumdrag += Vt*dt*Fdrag/3600000.;}

    //bdrag += Vt*dt*525.*10.878/3600000.;

    //Mwind = (Vt/3 - omega*1.2)*0.3+Mwind*0.7;

    #if addedlength

    //output to file

    if(t%10==1)

    {

        fprintf(fp,"time\tpos\tMwind\tMgen\tomega\tpower\tDragLoss\n");
```

```c
        }

        if(t%1==0)

        {

            fprintf(fp,"%.4lf\t%.1lf\t%.1lf\t%.1lf\t%.3lf\t%.1lf\t%.6f\n", t*dt, x, Mwind, Mgen, omega, power, sumdrag);

        }

        #endif

    }

    #if addedlength

    fprintf(fp,"Total energy produced: %e kW-hr\n", (power+0.5*J*omega*omega)/3600000.);

    #endif

    //fprintf(fp,"Total energy produced: %e kW-hr\n", (power)/3600000.);

    //system("PAUSE");

    ans[0]=(power+0.5*J*omega*omega)/3600000.;

    ans[1]=(sumdrag);

    return ans;

}


#if 1

double function(double omega, double theta, double x, double *data)

{

    #if 1

    float dx, xn, w[3], q[5], qi[6], M[6], Minterp;

    int k, kk, Nxn, first, second, third, index, Nw[2], Nqi[3];

    w[2] = omega;
```

```c
q[3] = theta;

dx = Vt * dt;

/* calculate nearest neighbor to the current position of truck, zeroth order interp */

Nxn = round(x/dx);

xn = dx * Nxn;

/* calculate the bounding omegas */

Nw[0] = floor(w[2]/dw);

Nw[1] = ceil(w[2]/dw);

w[0] = dw * Nw[0];

w[1] = dw * Nw[1];

//printf("%f\n", w[0]);

//calculate the const omega equivalent initial angles based on current angle for the bounding
omegas

for(k=0;k<=1;k++)

{

    qi[k] = q[3] - (w[k]*xn/Vt);

}

// use the equivalent initial angles for the two bounding omegas to find four bouding initial
thetas

for(k=0;k<=1;k++)

{

    Nqi[2*k+2] = floor(qi[k]/dq);

    Nqi[2*k+3] = ceil(qi[k]/dq);

    qi[2*k+2]= dq * Nqi[2*k+2];

    qi[2*k+3]= dq * Nqi[2*k+3];

}
```

// calculate the current angles for the four bounding equivalent initial angles

```c
for(k=0;k<=3;k++)

{

    kk = k/2;

    q[k] = qi[k+2] + (w[kk]*xn/Vt);

}
```

// using the bounding omegas, initial thetas, and current position, look up the bounding moments

```c
for(k=0;k<4;k++)

{

    first = Nw[k/2];

    second = Nqi[k+2];

    third = Nxn;

    // translate to vector index

    index = third + xMax*second + xMax*qMax*first;

    M[k+2] = data[index];

    //if(k==3){printf("%i\n\n", Nxn);}

}
//printf("%f\n", w[1]);

//printf("%f\n", w[0]);

// interpolate bounding moments

M[0] = M[2];

if(q[1] != q[0])

{

    M[0] += ((q[3]-q[0]) * (M[3]-M[2]) / (q[1]-q[0]));
```

```
    }

    M[1] = M[3];

    if(q[3] != q[2])

    {

        M[1] += ((q[3]-q[2]) * (M[5]-M[3]) / (q[3]-q[2]));

    }

    Minterp = M[0];

    if(w[1] != w[0])

    {

        Minterp += ((w[2]-w[0]) * (M[1]-M[0]) / (w[1]-w[0]));

    }

    // find

    // output to screen

    //for(k=0;k<=5;k++)

    //{

    //      printf("%lf\n", M[k]);

    //}

    //printf("Float=%f\n", M[0]);

    //printf("Int=%i\n", first);

    #endif

    return Minterp;

}
#endif


#if 1
```

```
double functiond(double omega, double theta, double x, float *data)

{

  #if 1

  float dx, xn, w[3], q[5], qi[6], M[6], Minterp;

  int k, kk, Nxn, first, second, third, index, Nw[2], Nqi[3];

  w[2] = omega;

  q[3] = theta;

  dx = Vt * dt;

  /* calculate nearest neighbor to the current position of truck, zeroth order interp */

  Nxn = round(x/dx);

  xn = dx * Nxn;

  /* calculate the bounding omegas */

  Nw[0] = floor(w[2]/dw);

  Nw[1] = ceil(w[2]/dw);

  w[0] = dw * Nw[0];

  w[1] = dw * Nw[1];

  //printf("%f\n", w[0]);

  //calculate the const omega equivalent initial angles based on current angle for the bounding omegas

  for(k=0;k<=1;k++)

  {

      qi[k] = q[3] - (w[k]*xn/Vt);

  }

  // use the equivalent initial angles for the two bounding omegas to find four bouding initial thetas

  for(k=0;k<=1;k++)
```

81

```c
{

    Nqi[2*k+2] = floor(qi[k]/dq);

    Nqi[2*k+3] = ceil(qi[k]/dq);

    qi[2*k+2]= dq * Nqi[2*k+2];

    qi[2*k+3]= dq * Nqi[2*k+3];

}

// calculate the current angles for the four bounding equivalent initial angles

for(k=0;k<=3;k++)

{

    kk = k/2;

    q[k] = qi[k+2] + (w[kk]*xn/Vt);

}

// using the bounding omegas, initial thetas, and current position, look up the bounding
moments

for(k=0;k<4;k++)

{

    first = Nw[k/2];

    second = Nqi[k+2];

    third = Nxn;

    // translate to vector index

    index = third + xMax*second + xMax*qMax*first;

    M[k+2] = data[index];

    //if(k==3){printf("%i\n\n", Nxn);}

}

//printf("%f\n", w[1]);
```

```c
//printf("%f\n", w[0]);

// interpolate bounding moments

M[0] = M[2];

if(q[1] != q[0])

{

    M[0] += ((q[3]-q[0]) * (M[3]-M[2]) / (q[1]-q[0]));

}

M[1] = M[3];

if(q[3] != q[2])

{

    M[1] += ((q[3]-q[2]) * (M[5]-M[3]) / (q[3]-q[2]));

}

Minterp = M[0];

if(w[1] != w[0])

{

    Minterp += ((w[2]-w[0]) * (M[1]-M[0]) / (w[1]-w[0]));

}

// find

// output to screen

//for(k=0;k<=5;k++)

//{

//      printf("%lf\n", M[k]);

//}

//printf("Float=%f\n", M[0]);

//printf("Int=%i\n", first);
```

```
    #endif

    return Minterp;

}

#endif
```

Typical output

Run#	Initial Angle	Power Produced	Drag Power Loss
0	5.000000e+000	1.011884e-004	6.529566e-003
1	1.500000e+001	1.348816e-004	6.637689e-003
2	2.500000e+001	1.631715e-004	6.520518e-003
3	3.500000e+001	1.565371e-004	6.562683e-003
4	4.500000e+001	1.108553e-004	6.612322e-003
5	5.500000e+001	7.763362e-005	6.616157e-003
6	6.500000e+001	6.488635e-005	6.446677e-003
7	7.500000e+001	7.348895e-005	6.185268e-003
8	8.500000e+001	1.624437e-004	5.950056e-003
9	9.500000e+001	1.999076e-004	6.259993e-003
10	1.050000e+002	1.666901e-004	6.278036e-003
11	1.150000e+002	1.079391e-004	6.335700e-003

Initial rotational velocity of 0.0 radians per second

Overall power prediction:

1.266353e-004

Run#	Initial Angle	Power Produced	Drag Power Loss
0	5.000000e+000	1.452512e-004	6.472439e-003

1	1.500000e+001	1.847771e-004	6.568172e-003
2	2.500000e+001	1.756591e-004	6.550048e-003
3	3.500000e+001	1.514047e-004	6.592850e-003
4	4.500000e+001	1.037001e-004	6.592637e-003
5	5.500000e+001	8.127142e-005	6.507317e-003
6	6.500000e+001	8.395392e-005	6.367926e-003
7	7.500000e+001	1.636305e-004	5.896890e-003
8	8.500000e+001	2.097618e-004	6.182639e-003
9	9.500000e+001	2.242370e-004	6.215631e-003
10	1.050000e+002	1.487110e-004	6.413891e-003
11	1.150000e+002	1.347719e-004	6.297375e-003

Initial rotational velocity of 1.0 radians per second

Overall power prediction:

1.505941e-004

Run#	Initial Angle	Power Produced	Drag Power Loss
0	5.000000e+000	2.239851e-004	6.480590e-003
1	1.500000e+001	2.096788e-004	6.628414e-003
2	2.500000e+001	1.806262e-004	6.573578e-003
3	3.500000e+001	1.396597e-004	6.581760e-003
4	4.500000e+001	1.039734e-004	6.501907e-003
5	5.500000e+001	9.438412e-005	6.322922e-003
6	6.500000e+001	1.261844e-004	5.859993e-003
7	7.500000e+001	2.238461e-004	6.119485e-003
8	8.500000e+001	2.544046e-004	6.182237e-003

9	9.500000e+001	2.274209e-004	6.151305e-003
10	1.050000e+002	1.616327e-004	6.310643e-003
11	1.150000e+002	2.080605e-004	6.028493e-003

Initial rotational velocity of 2.0 radians per second

Overall power prediction:

1.794880e-004

Run#	Initial Angle	Power Produced	Drag Power Loss
0	5.000000e+000	2.649584e-004	6.683281e-003
1	1.500000e+001	2.161417e-004	6.612711e-003
2	2.500000e+001	1.735884e-004	6.574485e-003
3	3.500000e+001	1.305744e-004	6.515781e-003
4	4.500000e+001	1.046381e-004	6.384134e-003
5	5.500000e+001	9.512362e-005	6.150680e-003
6	6.500000e+001	2.368421e-004	5.958653e-003
7	7.500000e+001	2.742891e-004	6.217461e-003
8	8.500000e+001	2.956041e-004	6.000711e-003
9	9.500000e+001	2.277100e-004	6.175672e-003
10	1.050000e+002	2.396251e-004	5.988553e-003
11	1.150000e+002	3.061599e-004	6.401873e-003

Initial rotational velocity of 3.0 radians per second

Overall power prediction:

2.137712e-004

Run#	Initial Angle	Power Produced	Drag Power Loss

0	5.000000e+000	2.683801e-004	6.674075e-003
1	1.500000e+001	2.134009e-004	6.558558e-003
2	2.500000e+001	1.630244e-004	6.542638e-003
3	3.500000e+001	1.281021e-004	6.427295e-003
4	4.500000e+001	1.089630e-004	6.222336e-003
5	5.500000e+001	2.368604e-004	5.797573e-003
6	6.500000e+001	2.954716e-004	6.239013e-003
7	7.500000e+001	3.214347e-004	6.013370e-003
8	8.500000e+001	3.461225e-004	6.086545e-003
9	9.500000e+001	3.342298e-004	6.192034e-003
10	1.050000e+002	3.562503e-004	6.334213e-003
11	1.150000e+002	3.462916e-004	6.609159e-003

Initial rotational velocity of 4.0 radians per second

Overall power prediction:

2.598776e-004

Run#	Initial Angle	Power Produced	Drag Power Loss
0	5.000000e+000	2.615863e-004	6.600679e-003
1	1.500000e+001	2.042634e-004	6.490333e-003
2	2.500000e+001	1.550364e-004	6.456301e-003
3	3.500000e+001	1.264311e-004	6.316158e-003
4	4.500000e+001	2.056982e-004	5.811977e-003
5	5.500000e+001	3.125355e-004	6.114315e-003
6	6.500000e+001	3.494388e-004	6.092338e-003
7	7.500000e+001	3.929979e-004	6.019463e-003

8	8.500000e+001	3.881534e-004	6.321920e-003
9	9.500000e+001	4.182670e-004	6.356466e-003
10	1.050000e+002	4.028291e-004	6.522054e-003
11	1.150000e+002	3.524197e-004	6.781114e-003

Initial rotational velocity of 5.0 radians per second

Overall power prediction:

2.974714e-004

Typical detailed output

Appendix B: Python scripts

Output file manipulators

```python
omegas = 14

thetas = 6

domega = 4

dtheta = 20


for i in range(omegas):
  for j in range (thetas):
    filename = 'post\cm_turbine_a'+str(j*dtheta)+'_w'+str(i*domega)+'.txt'
    input_file = open(filename, 'r')
    lines = input_file.readlines()
    input_file.close()
    #print lines[0]
    if lines[0] == '"Moment Convergence"\n':
      print 'cutting!'
      output_file = open(filename, 'w')
      n = 0
      for line in lines:
        n += 1
        if n > 2:
          output_file.write(line)
      output_file.close()
    del lines
```

Vector concatenation data manipulators

```python
import numpy as np

omegas = 12

thetas = 7

domega = 4

dtheta = 20


data = np.array([0])

for i in range(omegas):

  for j in range (thetas):

    if j*dtheta == 120: j = 0

    print j

    filename = 'post\cm_turbine_a'+str(j*dtheta)+'_w'+str(i*domega)+'.txt'

    print filename

    x = np.fromfile(filename,dtype='float',sep='\n')

    #print x

    x.shape=-1,2

    x = np.hsplit(x,2)

    #print x[1]

    data = np.vstack((data,x[1]))

    print len(x[1])

    del x

print data

np.savetxt('post-sim\data.txt', data, fmt='%1.4f', newline='\n')
```