Scientific Technical Report (STR) for DOE/EERE

Project Title: Real-time POD-CFD Wind-Load Calculator for PV Systems

Award Number: DE-SC0010161

Project Period: 06/12/2013-03/09/2014

Reporting Period: 06/12/2013-03/09/2014

Submission Date: 03/23/2014

Recipient: Central Technological Corporation (CENTECORP)
932 Larson Drive, Altamonte Springs, FL 32714
www.centecorp.com

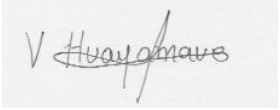
Subcontractor: Florida Solar Energy Center (FSEC)
1679 Clearlake Road, Cocoa, FL 32922
www.fsec.ucf.edu

PI: Victor Huayamave
Development Vice-President and CCO
Phone: 772-631-0457
Email: vhuayamave@centecorp.com

Team Members: Eduardo Divo (Centecorp)
Andres Ceballos (Centecorp)
Carolina Barriento (Centecorp)
Ronald Eaglin (Centecorp)
Stephen Barkaszi (FSEC)
Hubert Seigneur (FSEC)

DOE Project Manager: Mike Cliggett

DOE Project Officer: Scott Morgan



03/23/2014

Signature Date

Table of Contents:

1. Executive Summary

The primary objective of this project is to create an accurate web-based real-time wind-load calculator. This is of paramount importance for (1) the rapid and accurate assessments of the uplift and downforce loads on a PV mounting system, (2) identifying viable solutions from available mounting systems, and therefore helping reduce the cost of mounting hardware and installation. Wind loading calculations for structures are currently performed according to the American Society of Civil Engineers/ Structural Engineering Institute Standard ASCE/SEI 7; the values in this standard were calculated from simplified models that do not necessarily take into account relevant characteristics such as those from full 3D effects, end effects, turbulence generation and dissipation, as well as minor effects derived from shear forces on installation brackets and other accessories. This standard does not include provisions that address the special requirements of rooftop PV systems, and attempts to apply this standard may lead to significant design errors as wind loads are incorrectly estimated. Therefore, an accurate calculator would be of paramount importance for the preliminary assessments of the uplift and downforce loads on a PV mounting system, identifying viable solutions from available mounting systems, and therefore helping reduce the cost of the mounting system and installation. The challenge is that although a full-fledged three-dimensional computational fluid dynamics (CFD) analysis would properly and accurately capture the complete physical effects of air flow over PV systems, it would be impractical for this tool, which is intended to be a real-time web-based calculator. CFD routinely requires enormous computation times to arrive at solutions that can be deemed accurate and grid-independent even in powerful and massively parallel computer platforms.

This work is expected not only to accelerate solar deployment nationwide, but also help reach the SunShot Initiative goals of reducing the total installed cost of solar energy systems by 75%. The largest percentage of the total installed cost of solar energy system is associated with balance of system cost, with up to 40% going to “soft” costs; which include customer acquisition, financing, contracting, permitting, interconnection, inspection, installation, performance, operations, and maintenance. The calculator that is being developed will provide wind loads in real-time for any solar system designs and suggest the proper installation configuration and hardware; and therefore, it is anticipated to reduce system design, installation and permitting costs.

2. Background

The proposed solution would not only take advantage of the great detail and accuracy of a grid-converged 3D CFD analysis, but also calculate in real-time the loads that result from wind-induced drag and lift forces on PV arrays. A real-time response framework based on the Proper Orthogonal Decomposition (POD) method will be developed. The key to this approach is to generate, beforehand, and off-line an extensive set of solutions using CFD within our defined design space (various module sizes, wind speed, topography, roof dimensions and pitch, etc.) and store them in a database. The web-based wind-load calculator would then access the database of known solutions and generate in real-time an approximated solution by means of the POD method, which can be thought of as a multifaceted interpolation that preserves the physics of the problem. Because the POD method can produce a low-order approximation of the solution field with minimal loss of accuracy and fidelity, it serves as a reliable, fast and accurate response surface within the design space that can enable a real-time web-based calculator. In addition, the POD method is favorable for this application as it provides many

desirable numerical features such as model reduction, error filtration, and regularization. Furthermore, as the POD algorithms rely uniquely on algebraic manipulation of previously generated field data, the entire POD modeling framework can be implemented in modest platforms such as laptops and tablets while allowing for 'true' real-time prediction.

Specialized engineers and scientists with strong background in the areas of solar energy, computational fluid dynamics (CDF), structural analysis using finite element method (FEA), and numerical methods are currently working together on this project. All the work will be performed utilizing the existing assets of both Central Technological Corporation (Centecorp) and the associated research group at the Florida Solar Energy Center (FSEC) of the University of Central Florida (UCF). Centecorp also owns commercial licenses for CFD and FEA software to analyze the model in hand and obtain the desired solutions. A tool framework will be developed which can be used by engineers and solar panel installers to predict loads over solar panels given several different design parameters.

3. Significant Achievements or Major Concerns

Matching funds provided by the Florida High-Tech Corridor Council (FHTCC) were used to perform an experimental data validation, to carry out a wind-burst analysis, and to hire several engineering firms to perform wind-load calculations on roof-mounted PV systems using the current ASCE 7 tables. In addition, a working version prototype of the web-based Wind-Load Calculator has been developed and deployed to illustrate the function and capacity of the eventual final product. These tasks were not part of the original scope of Phase I of this project but thanks to the matching funds the scope was expanded to achieve these additional goals. The details of these tasks are explained in detail in the following sections.

4. Project Progress

The progress of the Phase I work has been completed according to what it was proposed in the performance schedule. The following is a list of tasks and studies conducted during the project performance period as well as the technical accomplishments.

4.1. Study of Non-hardware ("Soft") Costs Reduction of the Proposed Tool

The SunShot Initiative aims to reduce the total installed cost of solar energy systems by 75% between 2010 and 2020. To that end, there is a significant effort in trying to reduce both hardware and soft costs. This project will significantly impact non-hardware cost reduction for residential and small commercial solar photovoltaics. In Q4 2012 and within this category (less than 10KW systems), balance-of-system (BOS) cost was estimated to be 70% ($2.6/W) of the total installed cost ($3.69/W), **with *soft cost alone corresponding to about 60% of the total installed cost ($2.13/W)*** [1].

The good news is that reported system pricing fell 6-14% from 2011-2012 and is expected to continue to decrease. According to a recently published soft cost reduction roadmap [2], targeted soft-cost areas for cost reductions were: (1) customer acquisition; (2) permitting, inspection, and interconnection (PII); (3) installation labor; and (4) financing. Table 1 shows target values for each soft cost area. Green means realizable, yellow-low uncertainty, orange-medium uncertainty, and red-high uncertainty.

Table 1. Residential PV Soft-Cost Reduction Roadmap

	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020
Customer Acquisition ($/W)	$0.67	—	—	$0.53	$0.49	$0.45	$0.41	$0.36	$0.28	$0.19	$0.12
PII ($/W)	$0.20	—	—	$0.18	$0.16	$0.15	$0.13	$0.11	$0.10	$0.06	$0.04
Installation Labor ($/W)	$0.59	—	—	$0.51	$0.46	$0.42	$0.36	$0.30	$0.24	$0.19	$0.12
Other Soft Costs ($/W)	$1.86	—	—	$1.30	$1.14	$0.97	$0.82	$0.68	$0.56	$0.48	$0.37
Financing (WACC %-real)	—	—	9.9%	9.4%	8.8%	8.2%	7.7%	7.7%	4.8%	3.4%	3.0%
Total Soft Costs ($/W)	$3.32	—	—	$2.52	$2.25	$1.99	$1.72	$1.45	$1.18	$0.92	$0.65
Total System Costs ($/W)	$6.60	—	—	$4.99	$4.49	$3.99	$3.49	$3.00	$2.50	$2.00	$1.50

Primary Impact

The real-time web-based wind-load calculator is anticipated to create a path for achieving PII soft cost reduction below $0.10/W, which is currently deemed highly uncertain. Take for example the state of Florida; engineering fees for wind loading calculations and design vary between $1,000 and $2,000, a pre-requisite in many counties for permitting. Therefore, for a 5kW system, the fees are the equivalent of $0.2/W and $0.4/W respectively [2]. The wind load calculator would provide the same service for a flat fee of $40. If one assumes a 5kW installation on average, then the total engineering cost would amount to $0.008/W or a cost reduction of up to 98%. Furthermore, the median installed price for residential and small commercial systems in the state of Florida was $4.6/W or $23,000 for a 5kW system; therefore, $1000 to $2000 in savings from engineering fees would be substantial.

Secondary Impacts

For Installers:

- ***Customer acquisition*** – The calculator can have an impact on customer acquisition. 3D graphics can show customers the effect of high winds on their systems and increase their confidence on the safety of the installation.
- ***Simplified permitting process*** – The calculator can help streamlining the permitting process independent of the jurisdiction (see next section)
- ***Faster and cheaper installation*** – The calculator intends to provide rapid identification of viable solutions from available mounting systems based on load calculation with associated cost and installation time comparisons.

For Permitting:

- ***Lack of standardization in permitting*** – Standardization of the procedure for wind load calculations is a logical prerequisite for the standardization of requirements (comparing apples to apples). The calculator provides a path towards the standardization of the procedure by eliminating engineer error.
- ***Transparency of requirements*** – It was found that 36% of installers limit or avoid sales efforts in certain jurisdictions due to cumbersome permitting processes. Each jurisdiction with access to the software can provide their desired design requirements through the web interface. The software automatically takes these into account when recommending design based on load calculations.
- ***Online permit application submittal*** – Wind loads obtained from the web-based calculator and design can easily be submitted electronically in a standard/requested format. Each jurisdiction with access to the software can provide their desired submission format through the web interface.

For PV Module Manufacturers:

- ***Innovative labor-saving products*** – Equipment providers can use the software to obtain rapid and accurate assessments of wind load, shear stress, etc. during product development
- ***Durability and reliability*** – Equipment providers can use the calculator to obtain initial data module natural frequency, displacement, thermal cycling effects, etc.
- ***Marketing*** – 3D graphics can enhance product marketability

For Universities and Certifications:

- ***Research and education*** – The calculator can be used in an academic setting for doing research or in the classroom for educational purposes.
- ***Professional training*** – The calculator can be used to train installers and perhaps be added as a new module for the PV systems installation course at FSEC. The calculator could be included also in the curriculum for professional certifications such as NABCEP.

4.2. Parameterization of PV System Configuration and Modeling

This task consists of generating solid models and the fluid domain for CFD analysis of roof-mounted PV systems as a function of two design variables: module arrangement and pitch. Sets of representative values of these two design variables or parameters were selected so as to cover a wide design space. That is, selected standard PV module configuration arranged in a rectangular fashion as well as a discrete set of pitch angles including 18 degrees, 22.5 degrees, and 26.5 degrees. The angles were selected because they are a subset of typical roof slopes used in ASCE 7 design tables.

A parametric model for a house and a specific solar arrangement was created using standard values for currently available PV modules and hardware and typical dimensions for basic residential construction. The solid models were built using SolidWorks® 2011 (Dassault Systemes, Concord, MA). Every component was built based on industry and code standards. The following components were modeled to include in the analysis:

Commercial PV Modules

PV modules currently used for residential installations have power output of 200 to 260 Watts. This class of modules has dimensions of 60” to 66” by 36” to 39” and a frame thickness of 1.25” to 2”. Given these ranges, nominal module dimensions were selected to be 60” x 39” x 1.5”.

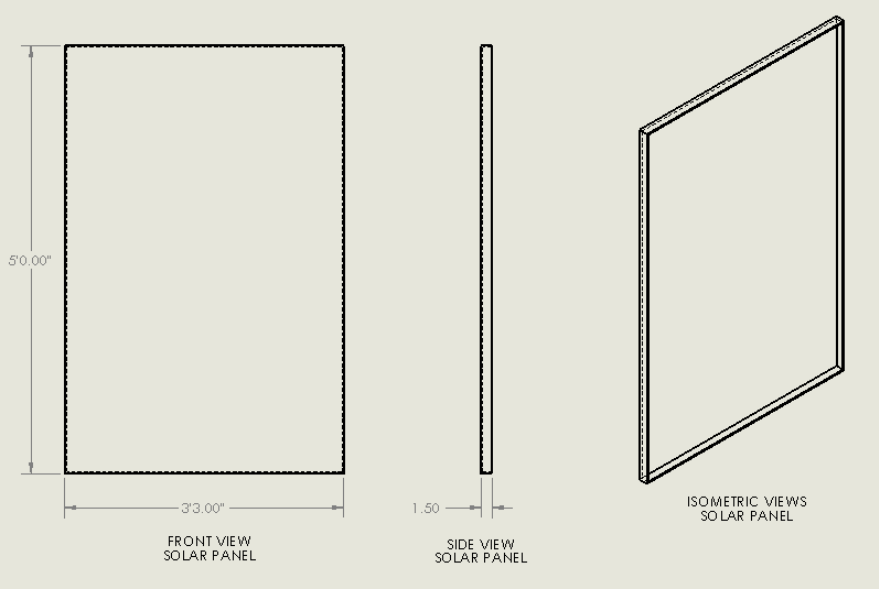


Figure 1. Solar Panel Dimensions

Arrangement of solar panels

The number of modules for the PV array was arbitrarily selected but based on common PV array configurations and the layout was partially constrained by the building dimensions. A total of 27 modules were arranged in 3 rows and 9 columns. A system with this number of modules would have a nominal output rating of 5 to 7 kW. This is also in the range of typical residential installations.

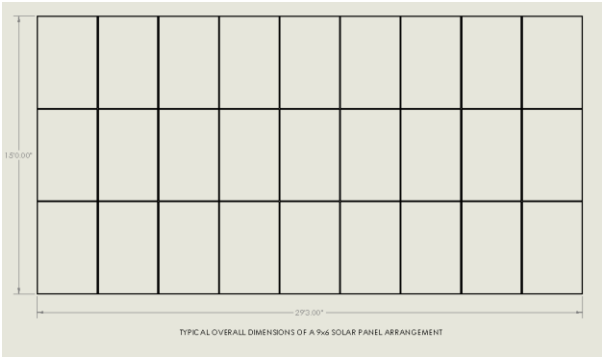


Figure 2. Solar Panel Arrangement

Standard house angles

The base case building model is configured a single-story building with a rectangular shape and a gable roof. Three different roof slopes (18 degrees, 22.5 degrees, 26.5 degrees) were selected for the initial evaluations. These represented angles used in the ASCE 7 wind load tables and are common roof slopes found in high wind regions

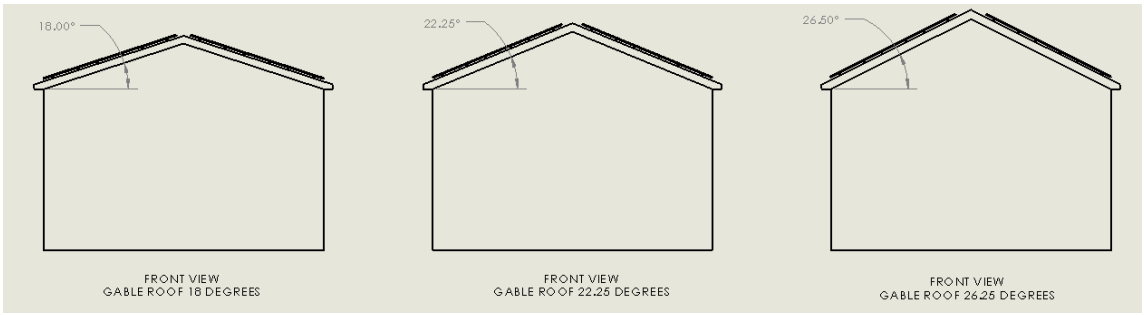


Figure 3. Gable roof pitch configurations

Assembly of solar panels with standard house

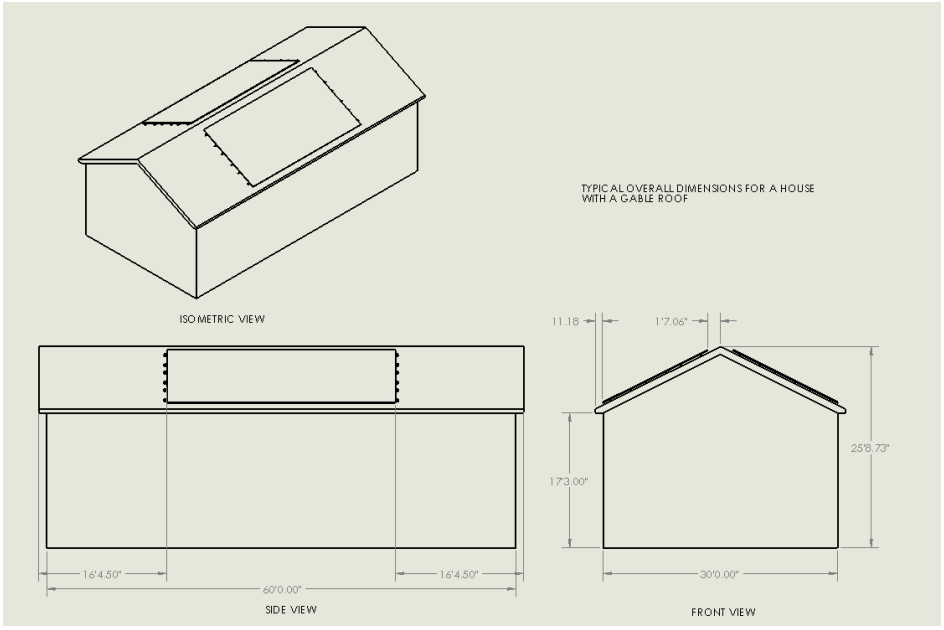


Figure 4. Overall house dimensions

4.3. CFD analysis of the Parameterized PV system Configurations

This task was specifically concerned with solving the incompressible viscous flow equations that govern the behavior of airflow over the PV system and yield the sought-after values of wind-load. These are the incompressible Navier-Stokes equations, which, in the Eulerian frame of reference, have the following form:

$$\nabla \cdot \vec{V} = 0$$

$$\rho \frac{\partial \vec{V}}{\partial t} + \rho(\vec{V} \cdot \nabla)\vec{V} = -\nabla p + \mu \nabla^2 \vec{V}$$

Where ρ is the density and μ is the dynamic viscosity. The field variables are the velocity field, $\vec{V}$, and the pressure, p. All field variables are functions of space and time, in a fixed domain Ω surrounded by a closed boundary Γ. The explicit space–time dependency of each dependent variable has been omitted for simplicity of notation. The fluid domain of the generated solid models were discretized using a combination of tetrahedral and hexahedral cells (to properly capture the boundary layer) and tested for grid-convergence using StarCCM+ (CD-Adapco, NY). A realizable κ-ω turbulent model was implemented and commonly used values of inlet turbulent intensity were imposed. The wind speed and incidence angle were introduced as the design variables or parameters.

The grid-converged CFD solution over each of the parameterized PV systems was then to be post-processed to integrate the pressure and shear stress over the entire surface of the PV system to yield the sought after values of wind-loads (uplift or downforce). Due to the nonlinear nature of the fluid flow phenomena, it is expected that the wind-loads will exhibit a highly nonlinear relationship with respect to the design parameters and possibly a non-monotonic one that can lead to multiple minima and/or maxima.

Initial CFD runs to determine the appropriateness of the chosen computational domain size, mesh, quality, and physics models were performed. In an iterative approach, a suitable set of the aforementioned parameters was reached, with the opportunity of further fine-tuning for the requirements of this particular analysis.

For the analysis, an incompressible flow model with κ-ω SST turbulence with two-layer, all Y+ wall treatment scheme was used. One initial case was run with the κ-ε model to initialize subsequent runs with the κ-ω model. The κ-ω SST model offers improved performance for boundary layers under adverse pressure relative to the κ-ε model. The κ-ω SST model has seen fairly wide application in the aerospace industry, where viscous flows are typically resolved and turbulence models are applied throughout the boundary layer. Potentially more accurate models such as Large Eddy Simulation (LES) and Detached Eddy Simulation (DES) are not considered for the time being since they require excessively higher mesh resolution and computational expense.

The analysis was initially run in steady state, after which the inherently unstable nature of the flow prompted the use of an unsteady (transient) solver. A first-order implicit unsteady solver was used. A second order up-winding for convective derivatives (with flux limiters) was used in all cases. The time-accurate simulations were carried out using the STOKES cluster at the University of Central Florida, with 48 parallel processes per case.

The defined computational domain for the 3D time-accurate turbulent Computational Fluid Dynamics (CFD) can be seen in Figure 5 and the final mesh was arrived at through a grid-

convergence study and consisted of over 10 million cells. A detailed view of the mesh can be seen in Figure 6.

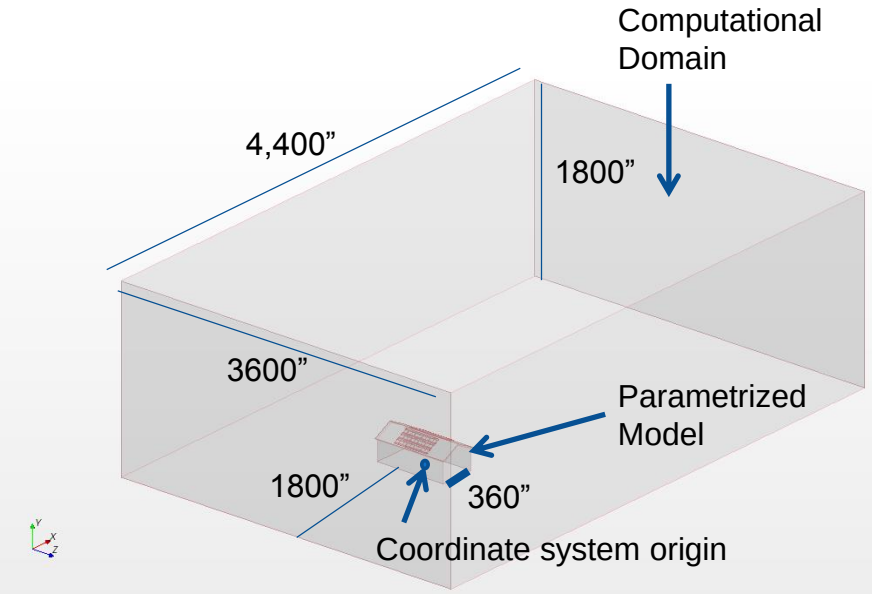


Figure 5. Computational domain.

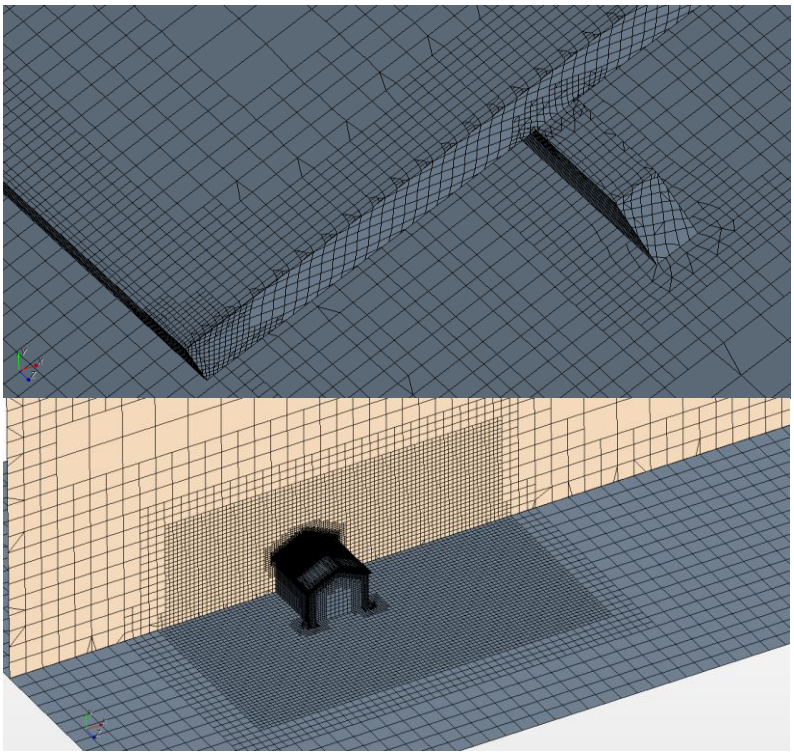


Figure 6. Computational mesh detail.

A total of eighty-four (84) configurations for CFD analysis were defined by altering two input parameters: wind speed (80mph to 200mph, in 20mph increments) and wind angle (360° around, in 30° increments). The number of CFD runs configurations was reduced to only twenty-eight (28) by taking advantage of symmetry and by placing PV panels on either side of the gable roof in order to render two or four solutions in one CFD run. The CFD analysis configuration schedule is shown in Table 2 indicating the task completion of every case.

Table 2. CFD analysis configuration schedule

Wind Speed\Angle	0˚/180˚	30˚/330˚ 150˚/210˚	60˚/300˚ 120˚/240˚	90˚/270˚
80 mph	OK	OK	OK	OK
100 mph	OK	OK	OK	OK
120 mph	OK	OK	OK	OK
140 mph	OK	OK	OK	OK
160 mph	OK	OK	OK	OK
180 mph	OK	OK	OK	OK
200 mph	OK	OK	OK	OK

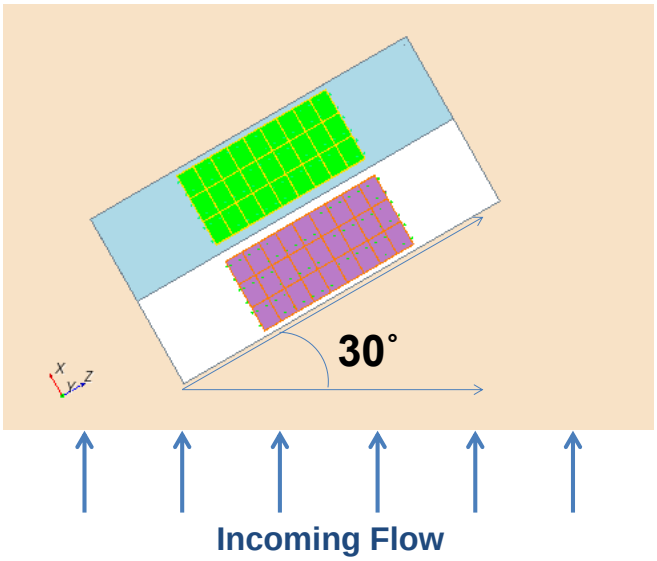


Figure 7. Wind angle nomenclature, 30˚ example

Figure 8 and Figure 9 show the forces due to wind load on the panels as a function of wind speed for the 0˚/180˚ and 30˚/330˚--150˚/210˚ configurations. Figure 10 through Figure 12 illustrate the flow field using velocity streamlines and pressure contours for sample cases in the directions: 0˚/180˚, 30˚/330˚--150˚/210˚ and 60˚/300˚--120˚/240˚, at 160 mph.

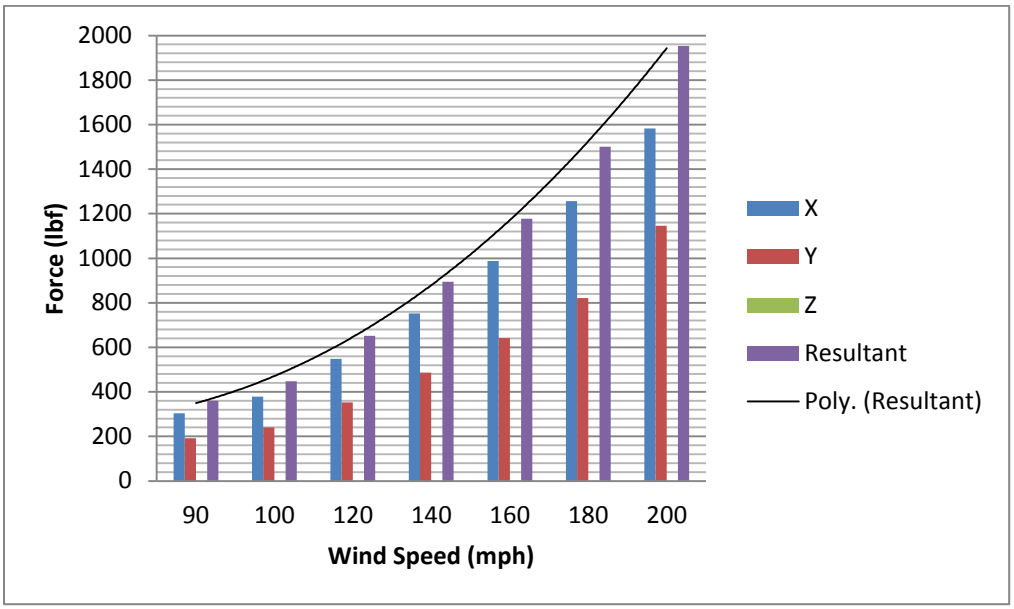


Figure 8. Wind forces on panels as a function of speed, 0˚/180˚configuration

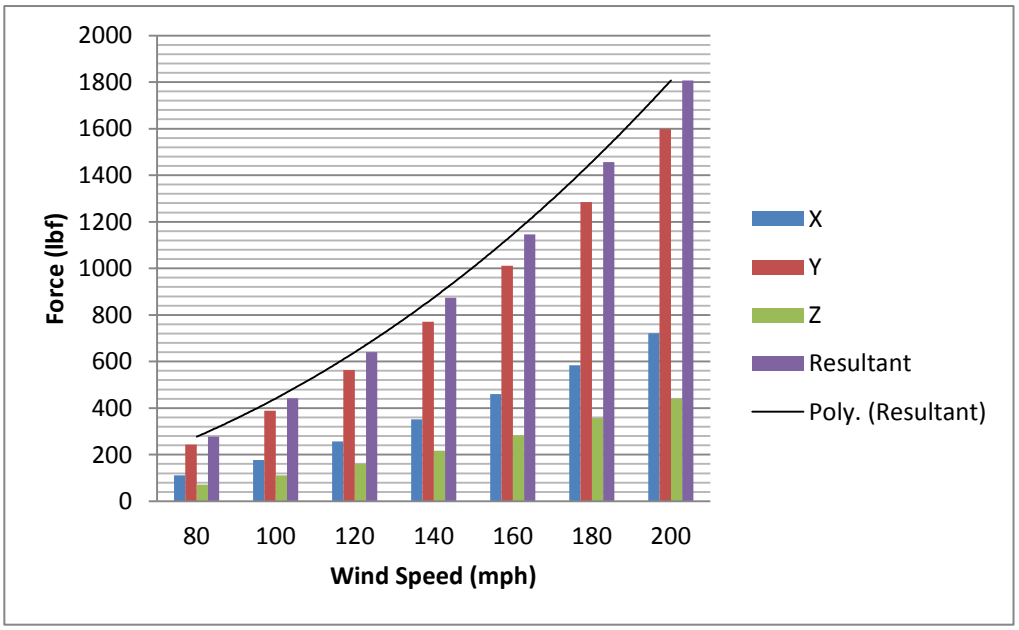


Figure 9. Wind forces on panels as a function of speed, 30˚/330˚--150˚/210˚ configuration

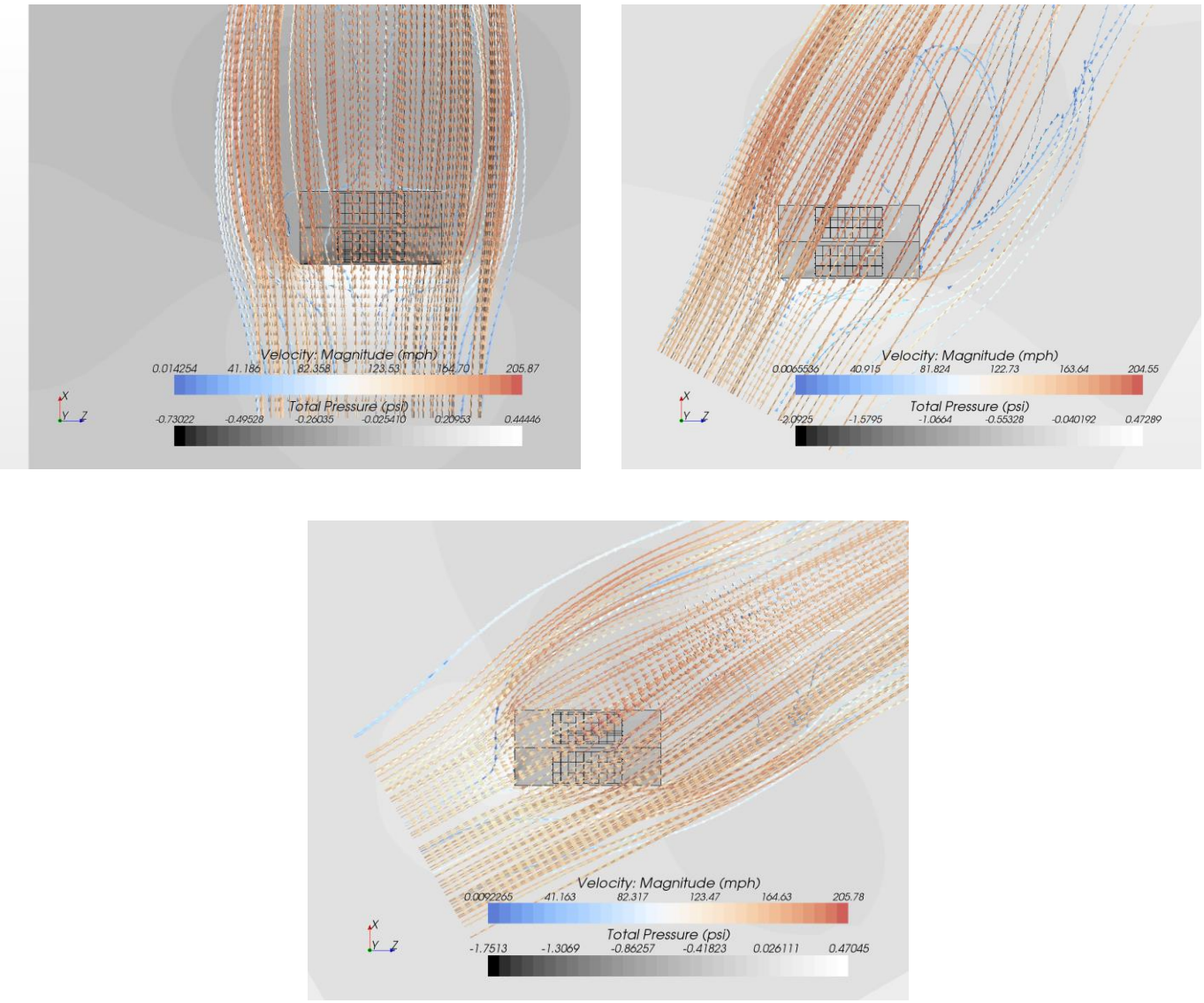


Figure 10. Streamlines colored by velocity and pressure contour plot of panel assembly, roof, and ground. Configurations are: 0˚ (Top Left), 30˚ (Top Right), 60˚ (Bottom).

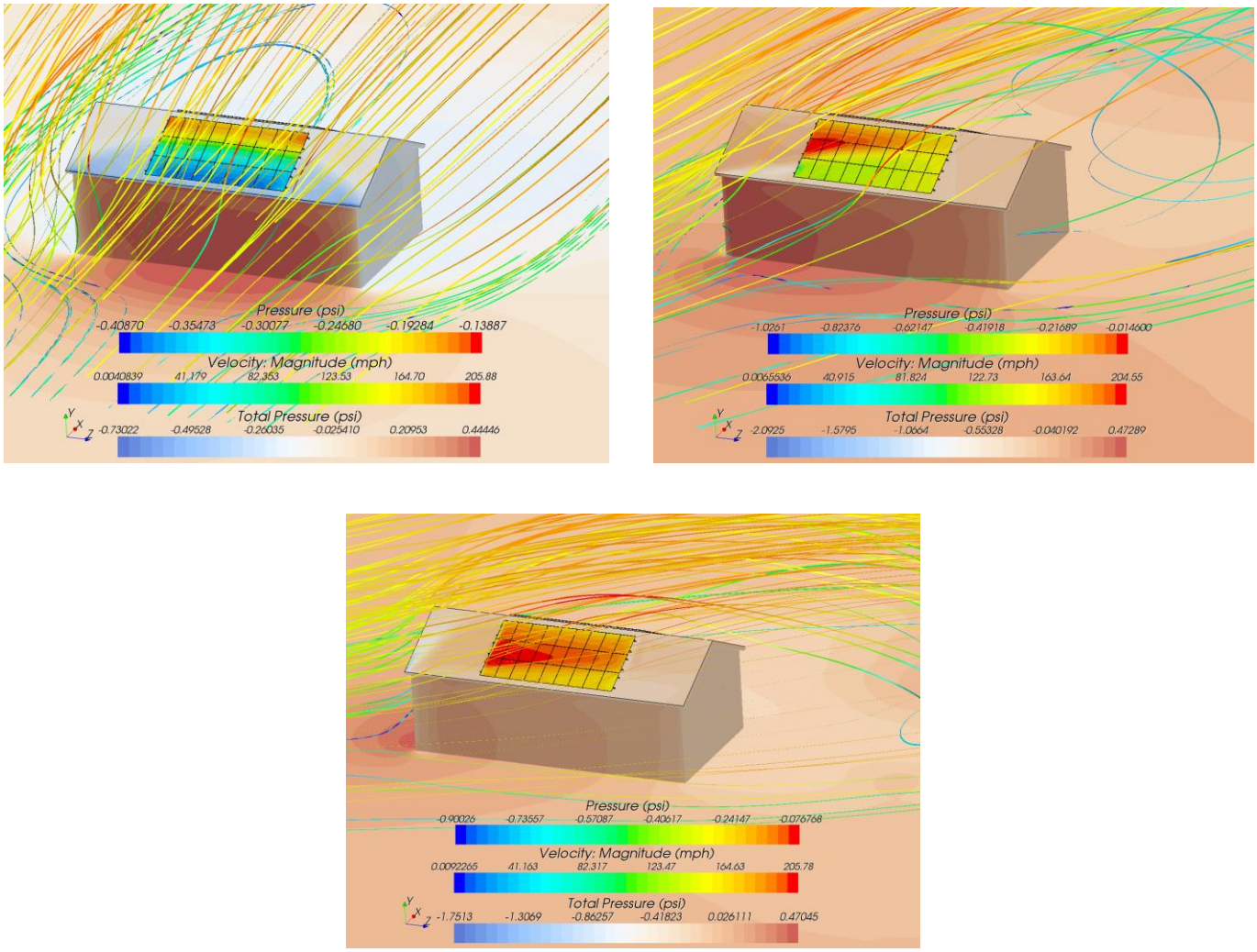


Figure 11: Streamlines colored by velocity and pressure contour plot of panel assembly, roof, and ground. Configurations are: 0˚ (Top Left), 30˚ (Top Right), 60˚ (Bottom).

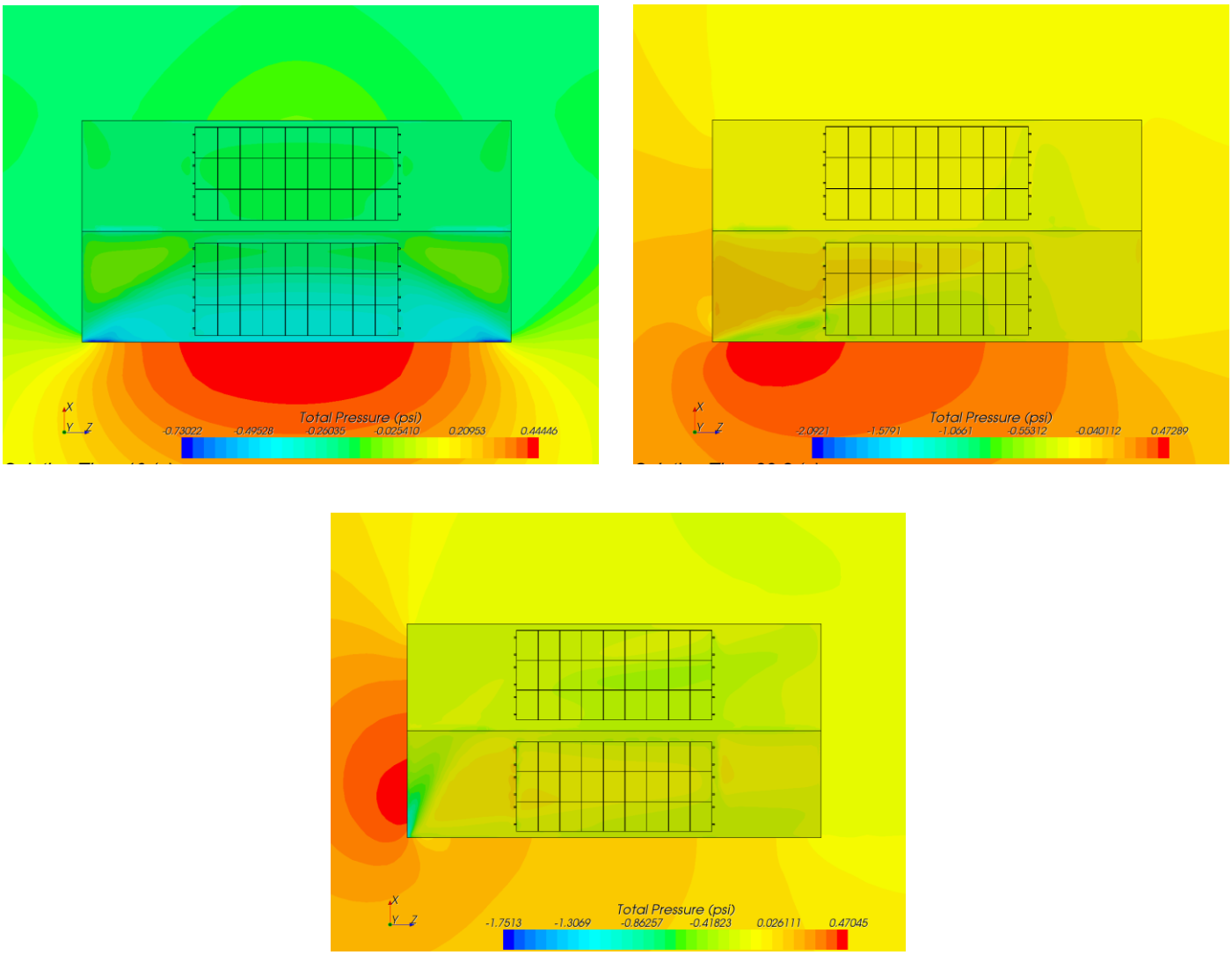


Figure 12: Total pressure contour plots of panel assembly, roof, and ground. Configurations are: 0˚ (Top Left), 30˚ (Top Right), 60˚ (Bottom). Top view.

The analysis configuration schedule is shown in Table 3 with the resulting values of average PV panel pressure in psf. Again, these values of average pressure on the PV system were rendered at a specific time value of the time-accurate CFD analysis when the total force generated on the panel (monitored during the CFD run) was at a maximum value.

Table 3. CFD analysis configuration schedule (POD-RBF)

Pressure (psf) Speed\Angle	0˚	30˚ (330˚)	60˚ (300˚)	90˚ (270˚)	120˚ (240˚)	150˚ (210˚)	180˚
80 mph	**-10.39**	**-8.89**	**-5.44**	**-7.57**	**-8.36**	**-8.35**	**-6.44**
100 mph	**-16.25**	**-13.89**	**-8.43**	**-11.83**	**-13.07**	**-13.05**	**-10.05**
120 mph	**-23.42**	**-20.01**	**-12.15**	**-17.03**	**-18.90**	**-18.79**	**-14.47**
140 mph	**-31.90**	**-27.24**	**-16.74**	**-23.17**	**-25.91**	**-25.58**	**-19.69**
160 mph	**-41.69**	**-35.59**	**-21.83**	**-30.26**	**-33.87**	**-33.42**	**-25.72**
180 mph	**-52.79**	**-45.05**	**-27.60**	**-38.29**	**-43.46**	**-42.31**	**-32.55**
200 mph	**-65.20**	**-55.63**	**-34.50**	**-47.26**	**-53.76**	**-52.25**	**-40.19**

4.4. Experimental Data Validation

A validation study of the CFD analysis model employed was conducted using experimental wind tunnel data found in the literature, [3]. The intention was to leverage the goals of this work to obtain a limited set of empirical data that can be used to validate at least one case. This proof of concept will be very useful in designing the experiments required for future development. The experimental data validation was not part of the original scope of Phase I but was included due to relevance of the results in the analysis. This was possible because of the matching funds obtained from the Florida High Tech Corridor Council (FHTCC) Industry Match Program.

In order to perform the CFD validation a new 3D computational model was developed to match the experimental setup using the following geometrical conditions:

- A PV module was modeled with the following dimensions: 157.2cm X 95.1cm X 4.1cm (length x width x thickness).
- The house had a 5:12 gable roof model, with 22.6˚ slope.
- PV module was mounted parallel to the roof.

- The base of the house is 3.7m X 3.7m.
- The roof has 0.3m overhangs.
- The mean height of the roof was equal to 3.2m, (height of top apex + height of edge) / 2. (Figure 13)

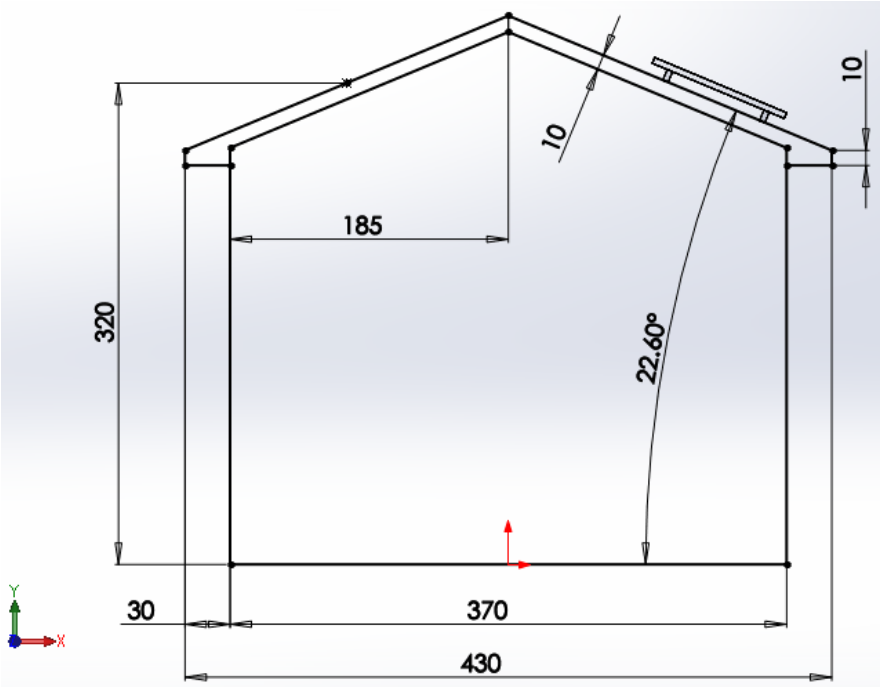


Figure 13. PV module position and geometry details of the 3D model (side view).

Also the following assumptions were made for the computational model:

- The thickness of the PV module was 4.1 cm, and the inside was considered hollow with a thickness of 1cm.
- The PV panel was mounted parallel to the roof of the house. The edge of the panel is aligned with the edge of the wall as shown in Figure 13 and centered along the side of the house as shown in Figure 14.
- The gap between the roof and the panel is 7.6 cm.
- Panel supports are modeled as 4 cylinders of 3.81 cm with their centers forming a 131.8 x 69.7 cm rectangle centered with respect of the PV panel sides.
- The model is imposed with a horizontal wind speed of 120mph which corresponds to a category 3 Hurricane.

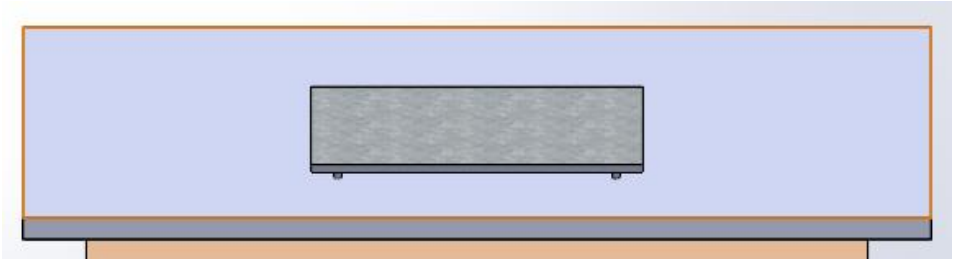


Figure 14. Solar panel position relative to the roof.

After the solid model was completed and matched with the experimental conditions, a computational domain was created to perform the CFD analysis. Three-dimensional coarse meshes (Figure 15 and Figure 16) and fine meshes (Figure 17 and Figure 18) were developed in order to arrive at a grid-converged solution.

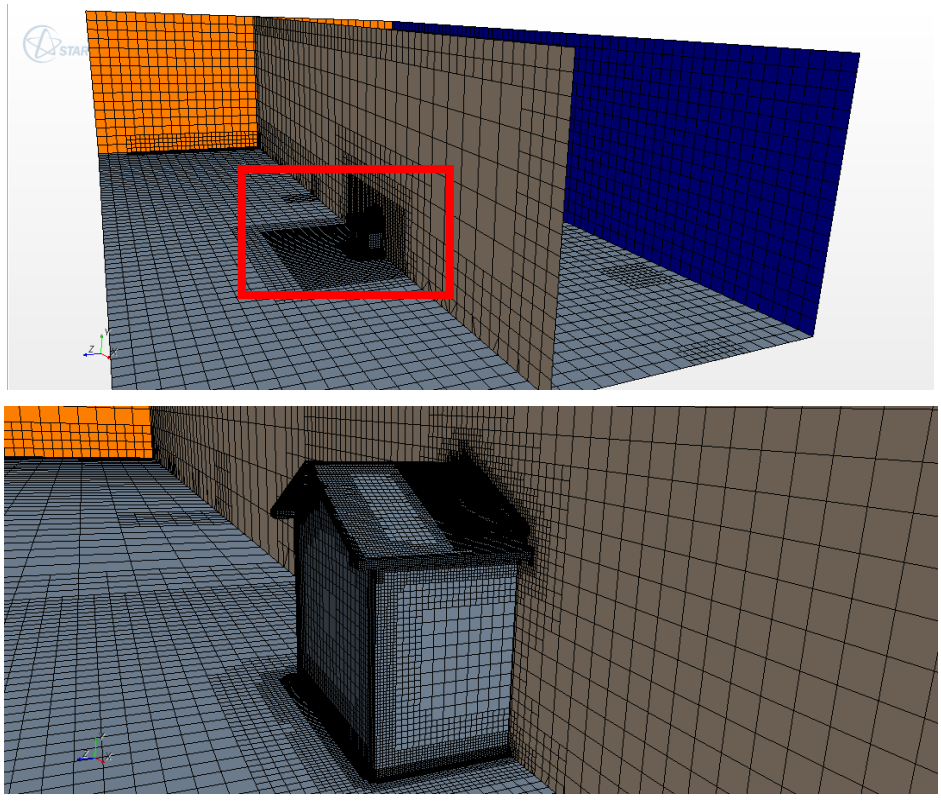


Figure 15. Mesh details of the coarse mesh configuration

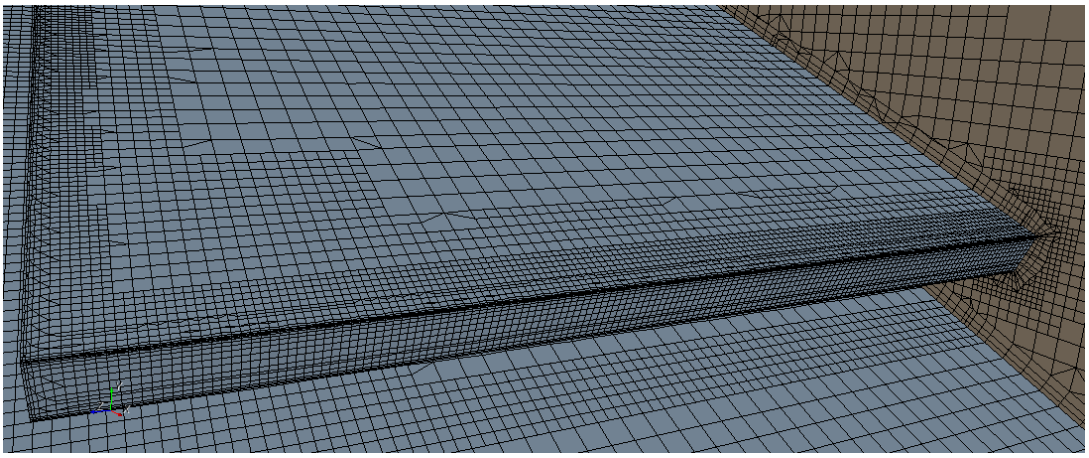


Figure 16. Coarse mesh detail on PV module

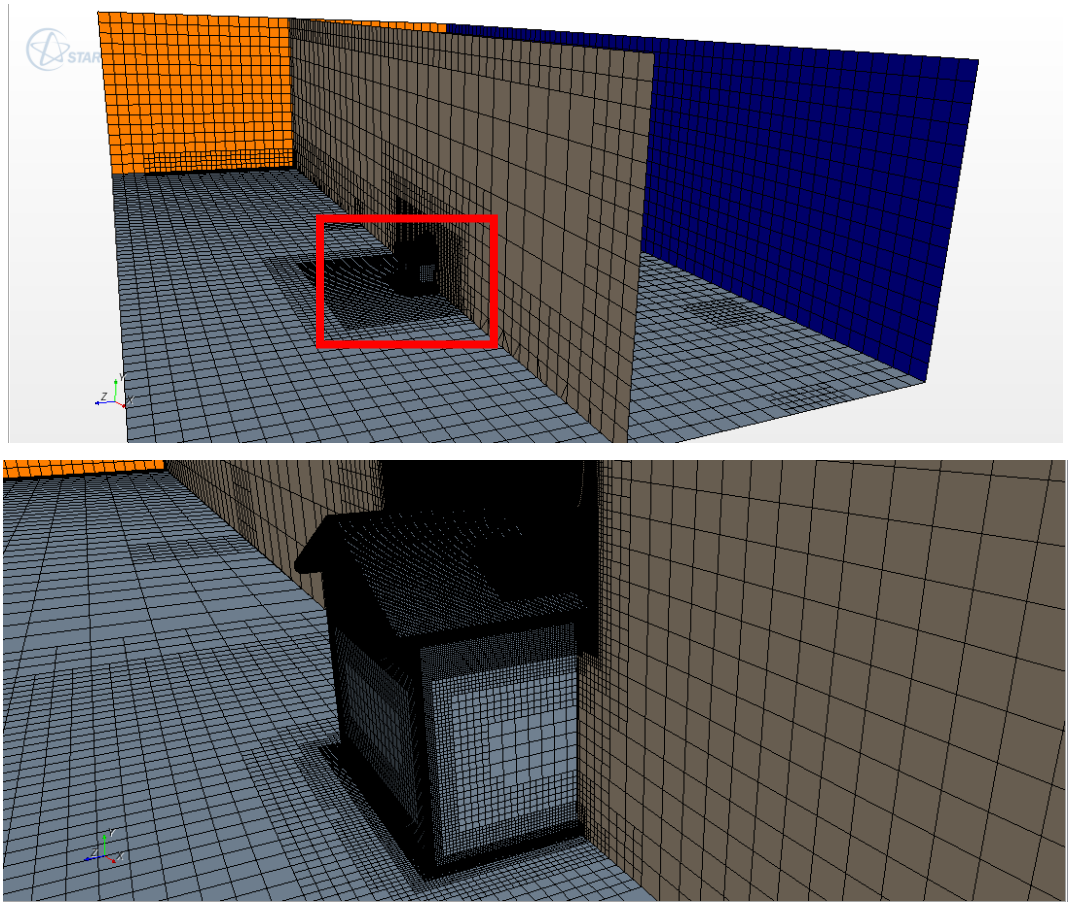


Figure 17. Fine mesh configuration detail

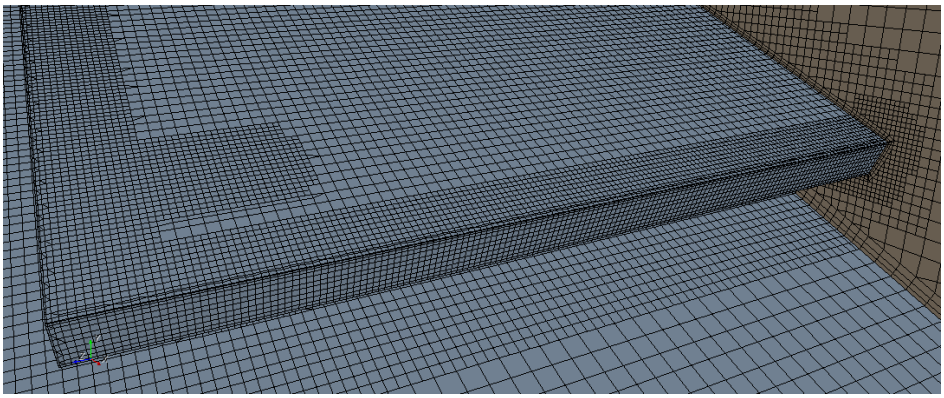


Figure 18. Fine mesh detail on PV module.

After the mesh was optimized, steady and transient CFD analyses were performed. The κ-ω SST turbulence model was again employed in both the coarse and the fine mesh configurations. The Y+ values were adjusted in the fine mesh configuration in order to arrive at higher accuracy solutions. Figure 19 through Figure 24 show pressure and velocity contours as well as velocity vectors for different views of the experimental validation case.

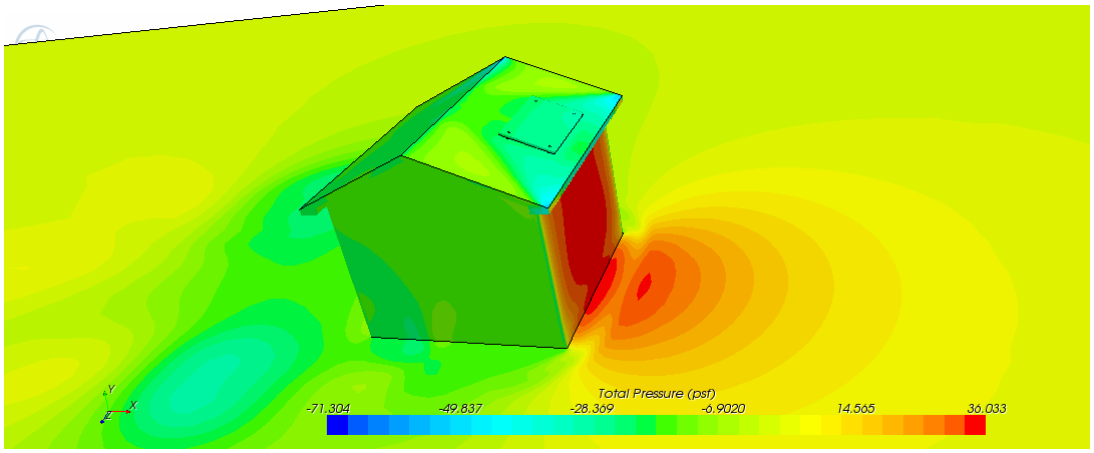


Figure 19. Total pressure surface plot (Coarse mesh 3D model)

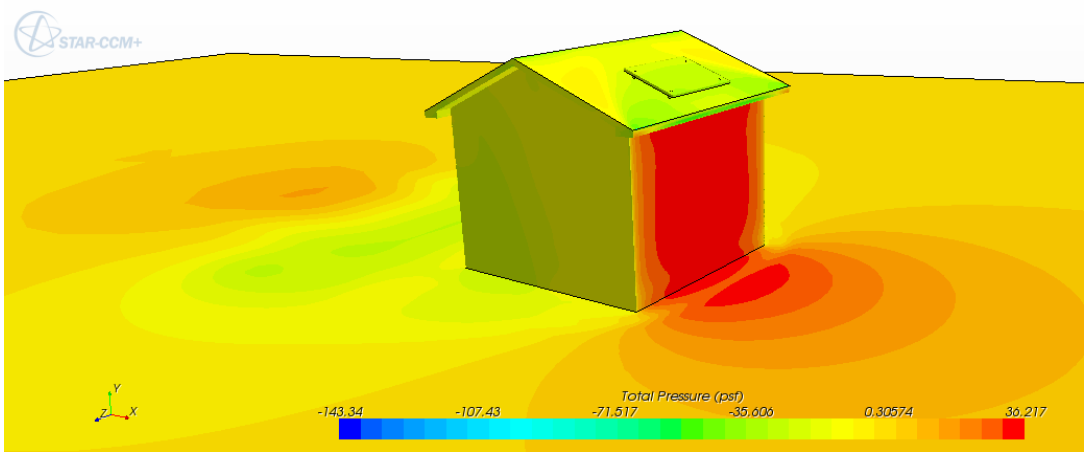


Figure 20. Total pressure surface plot (Fine mesh 3D model)

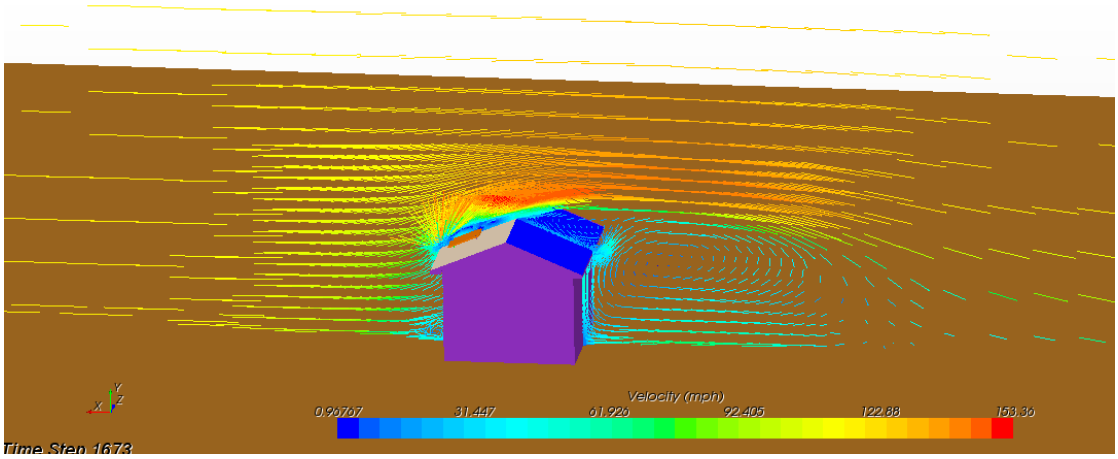


Figure 21. Symmetry plane streamline velocities (Coarse mesh 3D model)

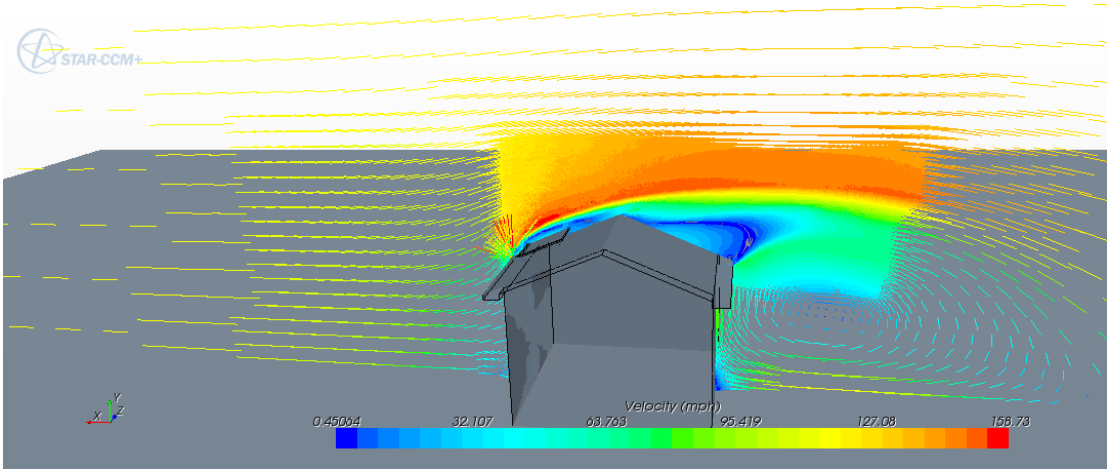


Figure 22. Symmetry plane streamline velocities (Fine mesh 3D model)

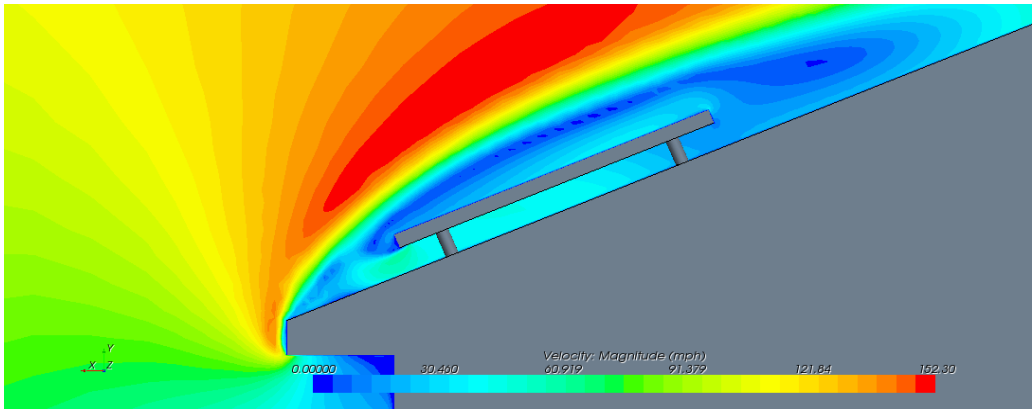


Figure 23. Symmetry plane velocity contour plot (Coarse mesh 3D model)

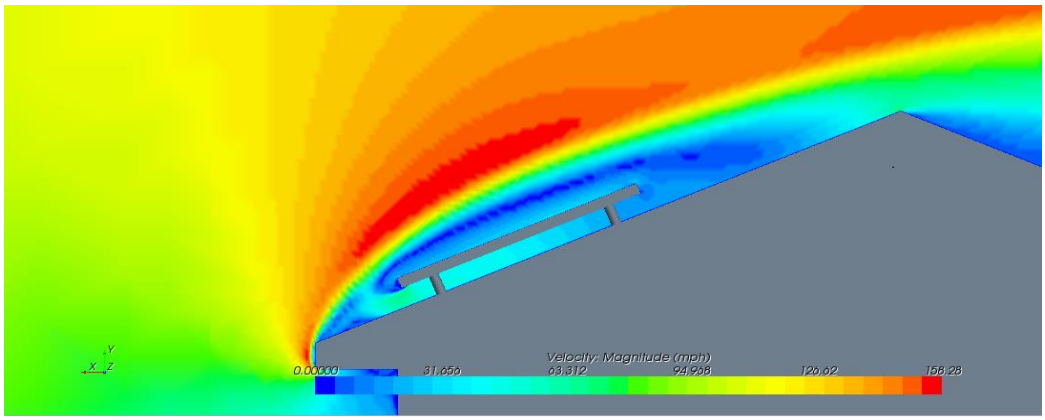


Figure 24. Symmetry plane velocity contour plot (Fine mesh 3D model)

The final results were measured in global coordinates and also using local coordinates aligned with the PV module. The resulting forces in the Z direction on the global direction (normal direction to side of the module) were very small (-0.76lbf to 0.15lbf) and therefore not reported. Table 4 shows the CFD results for all the different simulations performed.

Table 4. CFD-computed forces over experimental PV panel

Total Force (lbf)		Global Coordinates		Local Coordinates	
		X	Y	Tangential	Normal
Coarse Mesh	*Steady*	25.426	44.595	0.402	51.804
	Transient	27.019	48.946	-6.052	55.585
Fine Mesh	*Steady*	29.959	47.344	-27.006	30.980
	Transient	29.578	51.749	-24.403	27.735

It is clearly shown in Table 4 that there is a large discrepancy in the results provided by the coarse and fine meshes, in particular when shifting reference frames to the local coordinate system (normal and tangential to the panel). Therefore, the fine mesh calculations were adopted as those to be compared to the experimental data. To compare these results to the experimental values, pressure coefficients must be calculated for the steady and transient cases of the fine mesh configuration. The following relations were used in the experiment to calculate the pressure coefficient:

$$C_f = \frac{F}{qA} \qquad q = \frac{1}{2}\rho v^2$$

Were F is the force acting over the panel, q is the dynamic pressure, ρ is the density of air, and v is the air stream velocity. The area A is defined as the projected area of the PV module normal to the applied force:

For the horizontal force $\mathrm{A} = 157.2\left(95.1\sin(22.6°) + 4.1\cos(22.6°)\right) = 6340\ cm^2$

For the vertical force $\mathrm{A} = 157.2\left(95.1\cos(22.6°) + 4.1\sin(22.6°)\right) = 14049\ cm^2$

C_{fx} and C_{fy} were calculated for the steady and transient fine mesh configurations and the results are tabulated on Table 5. CFD-Experimental validation results including: CFD data, wall on wind (WoW) data, and wind tunnel data.

Table 5. CFD-Experimental validation results

	CFD Data	WoW Data	Wind Tunnel Data
Variable	Predicted average	Predicted avg. over 1hr	Predicted avg. over 1hr
C_{fx}	**0.13945**	**0.19500**	**0.15500**
C_{fy}	**0.10475**	**0.03500**	**0.15500**

It can be seen on Table 5. CFD-Experimental validation results that the lateral and uplift load coefficients for the 5:12 gable roof are consistent with the solutions obtained by numerical analysis considering these experimental results did not take into account hollow PV modules.

4.5. POD Analysis and RBF Interpolation Network

This task involves the generation of Proper Orthogonal Decomposition (POD) basis that will serve as the response surface for interpolation within the design space. The first step in the implementation of the POD is the creation of the *snapshot* which is the collection of *N* sampled values of ***u*** **-** the field under consideration. In this case, ***u*** stores the discrete values of pressure. The panel configuration consists of 9x3 PV modules arranged as shown in Figure 25.

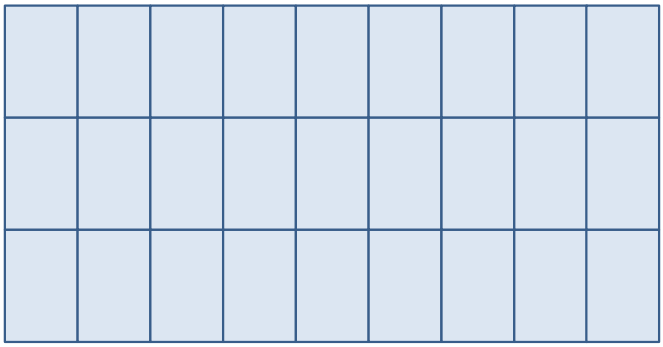


Figure 25. Panel configuration of 9x3 standard PV modules.

Each standard PV module is 39”x60”, so a point distribution for POD sampling was setup with 13x20 points per PV module separated 3” with a margin of 1.5”. This point configuration and spacing is shown in Figure 26. Therefore the number of sampled values of pressure on the panel is *N=(9x3)x(13x20)=7,020.* This *N* value provides sufficient resolution for the pressure distribution on the panel as well as for its integration to render the lift force.

Next, a collection of *M* snapshots denoted as $\mathbf{u}^j$ (*for j = 1, 2 … M*) are generated by altering the parameter(s) upon which the field depends on. In the current scope, these refer to the wind speed (80mph to 200mph, in 20mph increments) and wind angle (360° around, in 30° increments) for a total number of snapshots generated is *M=91*. Each $\boldsymbol{u}^j$ is then stored inside a rectangular *N x M* matrix **U** denoted as the snapshot matrix. Note that the dimensions of the snapshot matrix **U** are 7,020x91.

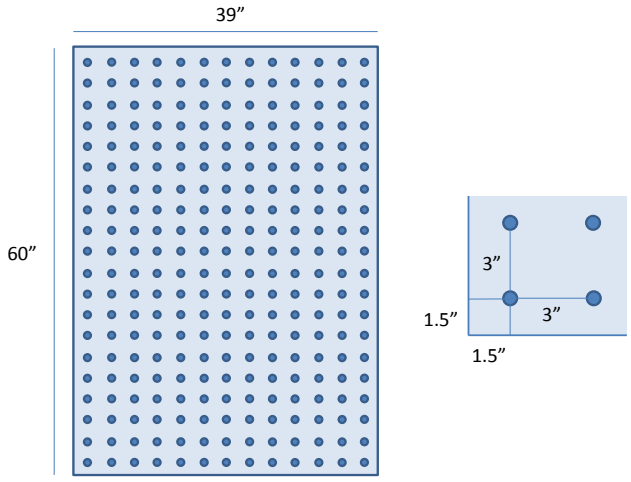


Figure 26. Point distribution and spacing on PV module for POD sampling.

Then the goal of the POD is to establish a set of orthonormal vectors $\boldsymbol{\Phi}^j$ (*for j = 1, 2 … M*) resembling the snapshot matrix **U** in an optimal way. The matrix **Φ** is commonly referred to as the POD basis and is given by:

$$\mathbf{\Phi} = \mathbf{U} \cdot \mathbf{V}$$

Where **V** represents the eigenvectors of the covariance matrix **C** and can be easily derived using the nontrivial solution of the general eigenvalue problem denoted as:

$$\mathbf{C} \cdot \mathbf{V} = \mathbf{\Lambda} \cdot \mathbf{V}$$

Λ represents a diagonal matrix that stores the eigenvalues λ of the covariance matrix **C**, which is defined as:

$$\mathbf{C} = \mathbf{U}^{\mathrm{T}} \cdot \mathbf{U}$$

Note that the dimensions of the covariance matrix **C** are *MxM* or *91x91*, which renders the eigenvalue decomposition process very fast. It may also serve to note that **C** is symmetric and positive definite and, therefore, all the eigenvalues λ are real and positive. Typically the eigenvalues λ are sorted in a descending order and can be attributed to the energy of the POD mode (base vector). This energy decreases rapidly with the increasing mode number.

Since higher modes hold little energy (or data) of the system they can be discarded without influencing the accuracy of representation. This is known as the truncation of the POD basis and is accomplished by deciding which fraction of the energy of the system can be neglected in later calculations. The resulting POD basis $\widehat{\mathbf{\Phi}}$, referred to as the *truncated* POD basis consists of *K<M* vectors and is given by:

$$\widehat{\mathbf{\Phi}} = \mathbf{U} \cdot \widehat{\mathbf{V}}$$

This also corresponds to the truncation of the eigenvector matrix, denoted as $\widehat{\mathbf{V}}$, which stores the first K^{th} eigenvectors of **C**. The truncated POD basis is also orthogonal $\widehat{\mathbf{\Phi}}^{\mathrm{T}} \cdot \widehat{\mathbf{\Phi}} = \mathbf{I}$ and presents optimal approximation properties. Once $\widehat{\mathbf{\Phi}}$ is known, the snapshot matrix **U** can be regenerated and approximated as $\mathbf{U} = \widehat{\mathbf{\Phi}} \cdot \mathbf{A}$, where **A** stands for the amplitudes associated with $\boldsymbol{u}^{j}$. Now referring to the orthogonality of $\widehat{\mathbf{\Phi}}$, the amplitudes can be determined from $\mathbf{A} = \widehat{\mathbf{\Phi}}^{\mathrm{T}} \cdot \mathbf{U}$.

At this time, data may begin to be extrapolated for information on the current problem. To do this, consider a vector **p** which stores the parameters on which the solution depends (wind speed and angle). Next, the amplitudes **A** are defined as a nonlinear function of the parameter vector **p**. The unknown constant coefficients of the current combination are gathered in a matrix **B**, as $\mathbf{A} = \mathbf{B} \cdot \mathbf{F}$, where **F** is defined as the matrix of interpolation functions, where the set of interpolation functions $f_i(\mathbf{p})$ can be chosen arbitrarily. For instance, Radial-basis functions (RBF) can be used as the interpolating functions of choice due to their spectral convergence properties. One of such RBFs is the inverse Hardy Multiquadric defined as:

$$f_i(\mathbf{p}) = f_i(\left|\mathbf{p} - \mathbf{p}^{i}\right|) = \frac{1}{\sqrt{\left|\mathbf{p} - \mathbf{p}^{i}\right|^{2} + c^{2}}}$$

Where *c* is defined as the RBF shape or smoothing parameter and $\mathbf{p}^i$ corresponds to the specific values of wind speed and angle used to generate $\mathbf{u}^i$ (for *i = 1, 2 … M*). The matrix of coefficients **B** can be evaluated by simple inversion as $\mathbf{B} = \mathbf{A} \cdot \mathbf{F}^{-1}$, where **F** is the matrix of interpolation functions defined as a set of *M=91* vectors ***f(p)*** defined as $\{\mathbf{f}\}_j = f_j\left(\left|\mathbf{p} - \mathbf{p}^j\right|\right)$. At this point it should be stressed that the matrix of amplitudes **A** and the matrix of coefficients **B** are known using the above relations. Now the following equation is arrived at:

$$\hat{\mathbf{\Phi}}^{\mathrm{T}} \cdot \mathbf{U} = \mathbf{B} \cdot \mathbf{F}$$

Using the orthogonality of $\hat{\mathbf{\Phi}}$, it can easily be seen that the snapshot matrix **U** can be approximated as $\mathbf{U}(\mathbf{p}) \approx \hat{\mathbf{\Phi}} \cdot \mathbf{B} \cdot \mathbf{F}(\mathbf{p})$, such that after the coefficient matrix **B** is evaluated, a low dimensional model can be set in vector form as:

$$\mathbf{u}(\mathbf{p}) \approx \hat{\mathbf{\Phi}} \cdot \mathbf{B} \cdot \mathbf{f}(\mathbf{p})$$

This model will now be referred to as the *trained* POD-RBF network and is completely capable of reproducing the unknown pressure field that corresponds to any arbitrary set of parameters **p** (wind speed and angle). This can be thought of as a numerical eingenfunction expansion of the solution reminiscent of the variation of parameters (or integral transform) method for the analytical solution of partial differential equations.

After the CFD computations were performed for all cases and the 7,020x91 POD snapshot matrix **U** was formed, the decomposition was performed and tested. The 91x91 covariance matrix **C** was formed as $\mathbf{C} = \mathbf{U}^{\mathbf{T}} \cdot \mathbf{U}$ followed by a standard eigenvalue decomposition which produced the results shown in Figure 27. Note that the 91 eigenvalues are displayed in logarithmic scale showing a maximum value of about 10^9 and a minimum value of about 10^{-5}. More importantly, the eigenvalues decrease very rapidly from the largest value to less than 10^3 after the first 12 eigenvalues, indicating that most of the system information (energy) is contained and can be extracted from the first few eigenvalues using a truncated POD basis.

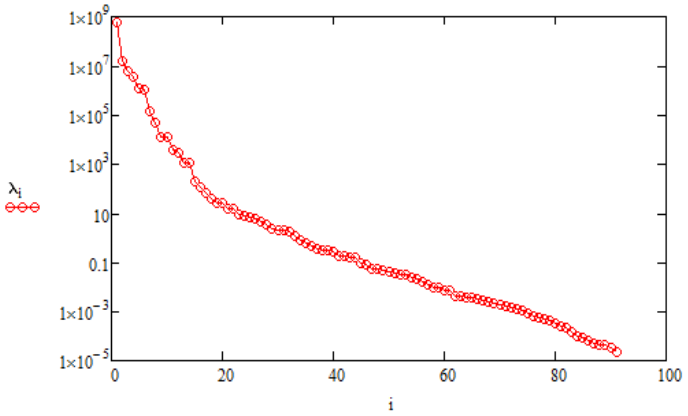


Figure 27. Eigenvalues of the POD covariance matrix sorted in descending order.

A truncated POD basis $\hat{\Phi}$ was built by using only the first 12 eigenvalues and eigenvectors $\hat{\mathbf{V}}$ as $\hat{\Phi} = \mathbf{U} \cdot \hat{\mathbf{V}}$. Then the POD snapshot matrix was reconstructed as $\mathbf{U}_w = \hat{\Phi} \cdot \mathbf{A}$ where the amplitudes are: $\mathbf{A} = \hat{\Phi}^T \cdot \mathbf{U}$. The pressure field is then obtained from each row of the reconstructed POD snapshot matrix $\mathbf{U}_w$.

To illustrate the accuracy of the POD truncation a test is performed comparing the CFD-generated pressure distribution and the POD-generated pressure distribution truncated after 12 eigenvalues for a wind speed of 140 mph and a wind angle of 30°. This comparison is shown in Figure 28(a) and Figure 28(b) respectively revealing an almost perfect qualitative accuracy. The relative RMS error between these two solutions was found to be just 0.025%, again revealing the high accuracy of the truncated POD approximation.

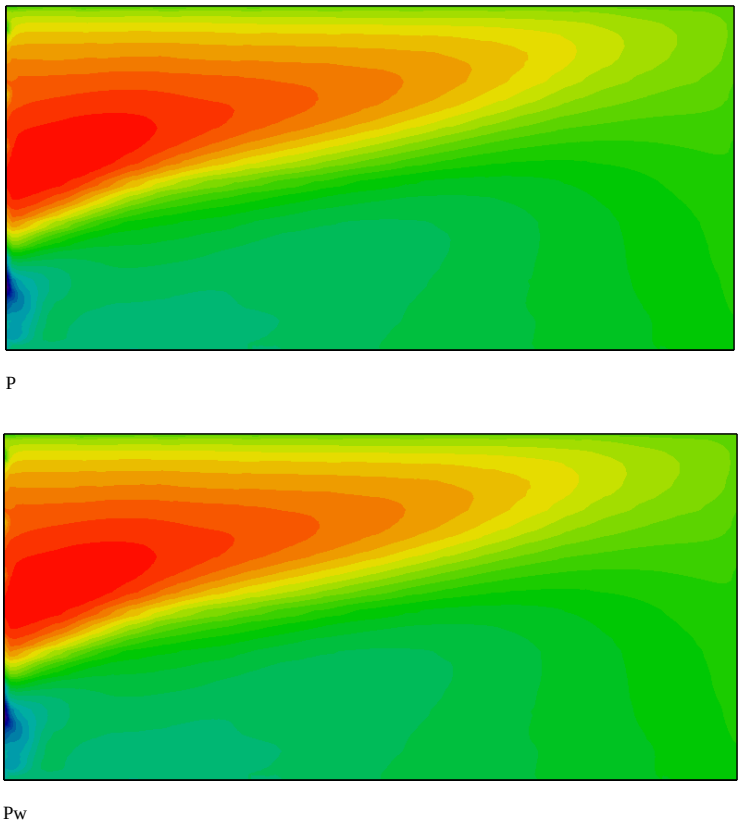


Figure 28. Pressure contours on PV panel at 140 mph and 30°. (a) CFD-generated and (b) Truncated POD-generated.

The trained POD-RBF interpolation network was implemented and tested. The design parameters **p** where in this case chosen to be the wind speed $p_1=V$ and the wind angle $p_2=\beta$ in this particular order. In addition, the RBF smoothing factor *c* in was chosen so as to produce a well-conditioned interpolation capable of generating smooth evaluations outside the collocation points p_1={80,100…200}, p_2={0,30…360}. This is accomplished by selecting a value of *c* that produces an interpolation matrix **F** with a high condition number but within the range of the precision used for the floating-point representation of the variables (double-precision in this case). For this case, the smoothing factor was selected to be:

$$c = \frac{1}{4}\sqrt{(200 - 80)^2 + (360)^2}$$

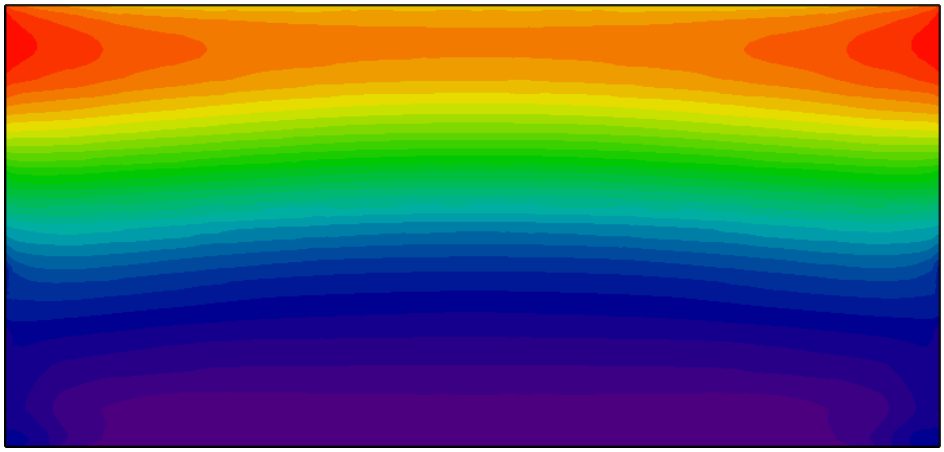
This value of the smoothing factor resulted in an interpolation matrix **F** exhibiting a condition number based on the L-infinity norm of $C = 6 \cdot 10^7$, which is within the range of double-precision floating point representation.

Once the interpolation matrix **F** was formed, the interpolation coefficients **B** were obtained using the POD amplitude matrix **A** as: $\mathbf{B} = \mathbf{A} \times \mathbf{F}^{-1}$. These interpolation coefficients were then employed with the RBF interpolation formula to evaluate the pressure distribution on the 7,020 points distributed on the panel at any arbitrary value of the design parameters p_1: wind speed and p_2: wind angle.

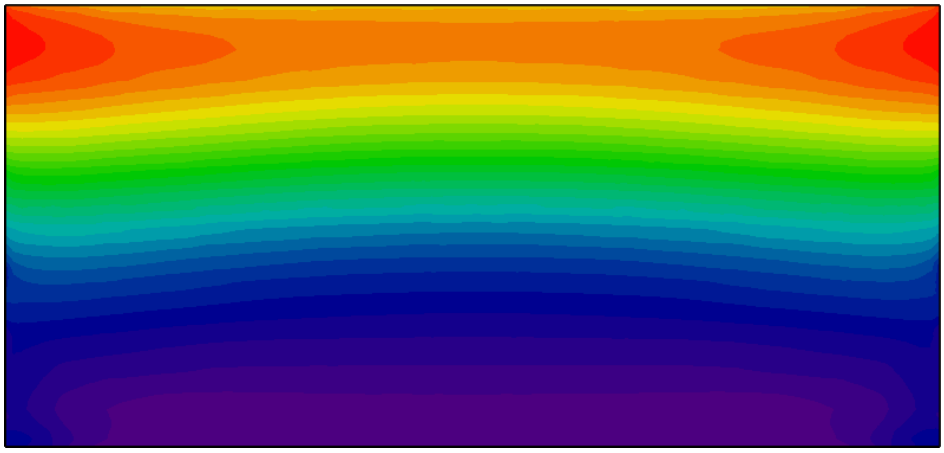
The POD-RBF interpolation network was tested using two CFD solutions that were not originally used as part of the POD snapshots. First, Figure 29(a) and Figure 29(b) show the CFD-generated ($P_{average}$=-13.15psf) and POD-RBF-rendered ($P_{average}$=-13.12psf) pressure distributions at a wind speed of 90mph and an angle of 0°. The qualitative comparison of the two solutions shows virtually no error while a quantitative comparison through a relative RMS error reveals a difference of 0.34%, demonstrating excellent agreement between the POD-RBF-rendered pressure distribution and the CFD-generated one, with the difference that the POD-RBF network produces instant results in any platform while the CFD solution requires several hours (or days) of computation to yield a grid-converged results in a high-end cluster.

The second test was carried out at a wind speed of 90mph and an angle of 180°. Figure 30(a) and Figure 30(b) show the CFD-generated ($P_{average}$=-8.14psf) and the POD-RBF-rendered ($P_{average}$=-8.14psf) pressure distributions at a wind speed of 90 mph and an angle of 180°. Again, the qualitative comparison of the two solutions shows virtually no error while a quantitative comparison through a relative RMS error reveals a difference of 0.98%, demonstrating very good agreement between the POD-RBF-rendered pressure distribution and the CFD-generated one.

These two comparison cases provide the validation and confidence necessary to implement the POD-RBF interpolation network to predict wind load distributions over PV panels at arbitrary wind velocities and angles dictated by installation requirements and codes.

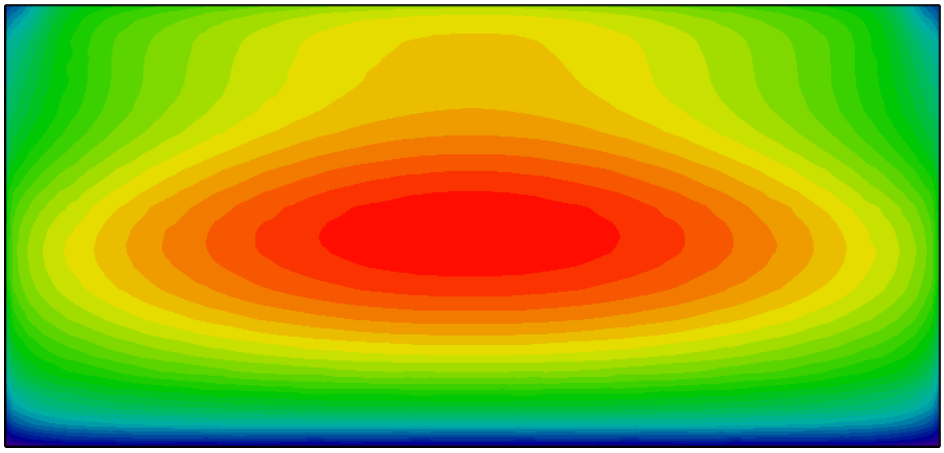


P_{CFD}

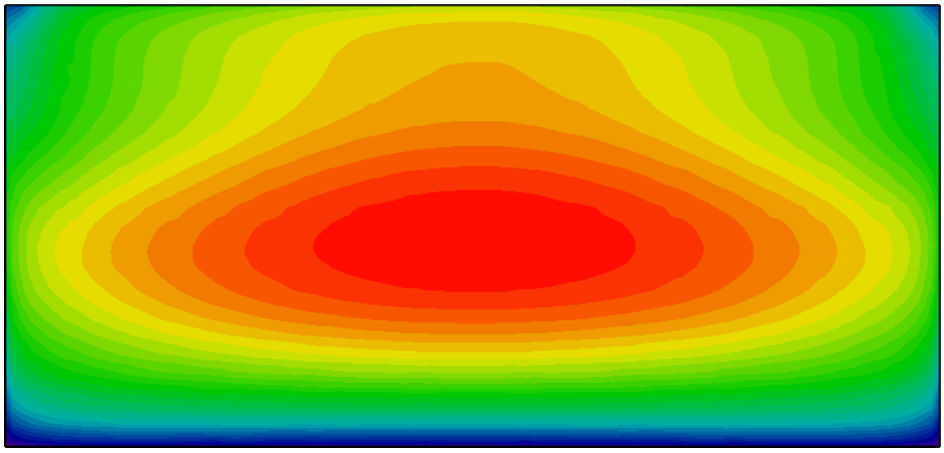


P_{POD}

Figure 29. Pressure contours on PV panel at 90 mph and 0°. (a) CFD-generated and (b) POD-RBF-generated.



P_{CFD}



P_{POD}

Figure 30. Pressure contours on PV panel at 90 mph and 180°. (a) CFD-generated and (b) POD-RBF-generated.

4.6. Wind-Load Calculator Graphical User Interface Prototype

A Graphical User Interface (GUI) for the Wind-Load Calculator has been developed to take advantage of the trained POD-RBF interpolation network in order to provide the pressure distribution on the panel at arbitrary values of wind speed and wind angle. In addition, it provides the average pressure as well as the resulting uplift force on the PV panel. This GUI was developed in *JavaScript* so that it can be rendered in HTML5 format from any Web browser on a desktop computer, laptop, tablet, or even smartphone. The layout is automatically refined to appropriately fit in desktop or mobile displays with legends as well as options for contour coloring. The Wind-Load Calculator (WLC) is currently being hosted at http://centecorp.com/wlc. A rendering of the Wind-Load Calculator Interface can be seen in Figure 31.

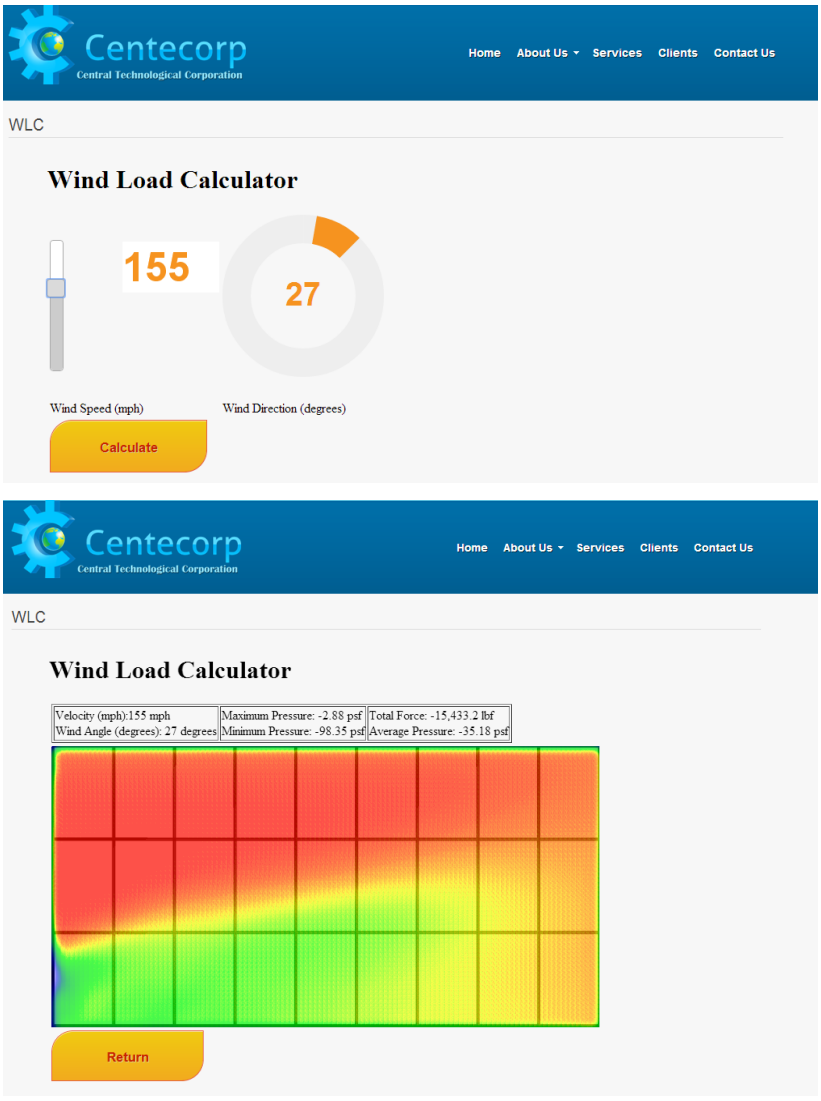


Figure 31. Rendering of the Wind-Load Calculator Interface

4.7. ASCE 7-10 Wind-Load Calculations

Validating a CFD model is a critical step in acceptance and use by the design community and authorities. There are design guides and codes in use for structures that are based on years of design experience, detailed analysis, and wind tunnel testing. ASCE 7 is the primary standard recognized for determining design loads for structures. The current version is ASCE 7-10 (although some jurisdictions may not have adopted this version yet and are using ASCE 7-05). Using the guidance provided in these standards was identified as a potential means to check the results.

Independent verification of the calculations was sought from different sources. ***Two engineering firms were selected to conduct an analysis, the published prescriptive procedure from a module manufacturer was followed, and one rack manufacturer's design tool was used.*** This task was not part of the original Phase I scope but it was added thanks to the matching funds obtained from the Florida High-Tech Corridor Council (FHTCC).

A base case structure and PV array was developed to replicate those used for the CFD analysis. The building dimensions were arbitrarily selected but based on a residential or small commercial two-story structure with a rectangular floor plan and trussed gable roof with a moderate roof slope. The array is comprised of all identical PV modules based on modules currently available and arranged in a typical fashion on the roof. The dimensions and other details are provided in Table 6.

Table 6. Base Case Structure and PV Array

Building Characteristics			
Building perimeter dimensions	30' x 60'	Ridge Height	<30'
Roof Slope	7°<**θ**< 27°	Eave Height	17'3"
Array Characteristics			
Module Dimensions	60" x 39" (16.25sf)	Module Weight	45 lbs
Array Dimensions	15' x 29'3" (438.75sf)	Rows x Columns	3 x 9

General wind load parameters were specified for the base case analysis. Common PV installations were considered when selecting the values. Table 7 details the load parameters. These criteria were the basis for the ASCE 7-10 analysis and were supplied to the engineers or used as inputs to calculation procedures that were identified.

Table 7. ASCE 7-10 General Requirements.

Basic Wind Speed V_{ult}	150mph	Location: Cocoa, FL 32922
Risk Category	II	Typical residential building
Exposure Category	B	Typical suburban setting
Adjustment Factor (λ)	1.0	Factor for building height and exposure
Topographic Factor (K_{zt})	1.0	Typical landscape with no major variations

A summary of the P_{net} (net pressure) results received from the engineers and other methods including the Wind-Load Calculator developed in this project are compiled in Table 8. These pressure values are used in the design of the array racking system and determine the number

and size of rails requires, number of point attachments to the roof, roof truss loads, and other critical components in the load path.

Table 8. P_{net} Design Pressures

	Velocity (mph)	Tributary Area (ft^2)	Zone 1 (psf)	Zone 2 (psf)	Zone 3 (psf)
Engineer 1	116	16.25	**13.1** **-21.8**	13.1 -36.5	13.1 -54.6
Engineer 2	150	100	**16.5** **-33.6**	16.5 -47.3	16.5 -74.8
Module Mfr Procedure	150	10	**23.3** **-37.0**	23.3 -64.5	23.3 -95.4
Rack Mfr Design Tool	121	n/a	**25.4*** **-34.2***	n/a n/a	n/a n/a
Wind-Load Calculator	150	n/a	**22.60** **-36.66**	n/a n/a	n/a n/a

* Converted from point load values and array area

The computed net pressures varied by as much as 76% between the different sources. Engineer 1 used a factor of $\sqrt{6}$ to convert the ultimate design wind speed (V_{ult}) to V_{asd} (allowable stress design) and a tributary area of 16.25 sf (module area) while Engineer 2 used V_{ult} and a tributary area of 100sf (~array area). The published procedure also indicated that V_{ult} was the wind speed but the tributary area was rounded down from the module area to a more conservative value at 10sf. 150mph was input to the rack manufacturer's tool as the ultimate wind speed and that seemed to be converted to a wind velocity of 121mph for the calculations. Other parameters could not be determined and the output loads from the design tool were converted to design pressures based on the number of point attachments and the array area.

The initial concept for the ASCE 7 exercise was to have a basis for comparison with the results rendered by the Wind-Load Calculator developed in this project and to document the cost of performing such calculations. ***Results from the Wind-Load Calculator fall within the range of calculated values.*** However, it is difficult to assess the validity of the comparison since there is so much variability in the interpretations of the ASCE 7 standard. Assumptions made by the engineer, design professional, or user can significantly impact the results of the calculations. Clearly, there is a desperate need for improved guidance, consistency, and accuracy in determining the wind loads on PV arrays. A CFD calculator that is calibrated and validated would significantly impact the speed, accuracy, and cost of PV system design and permitting.

The costs for obtaining engineering for the Base Case Structure ranged from $1,800 to $2,500. The manufacturers procedure and tool were available for free but it is not known if this documentation would be sufficient to satisfy permit requirements for any jurisdiction without endorsement from a design professional.

5. Final remarks and Phase 2 plan

The trained POD-RBF interpolation network has been tested and validated by performing the fast algebraic interpolation to obtain the pressure distribution on the PV system surface and comparing them to actual grid-converged fully-turbulent 3D CFD solutions at the specified values of the design variables (wind speed and angle). The solar power industry, engineering design firms, as well as society as a whole, could realize significant savings with the availability of a real-time in-situ wind-load calculator that can prove essential for plug-and-play installation of PV systems. The success of this project will have a direct impact on the installation process of PV systems by allowing readily-available off-the shelf components and modules to be assembled and mounted without the need for extensive engineering analysis behind the scene. Additionally, this technology allows for automated parametric design optimization in order to arrive at the best fit for a set of given operating conditions. All these tasks are currently prohibitive due to massive computational resources, effort, and time required to address large-scale CFD analysis problems, all made possible by a simple but robust technology that can yield massive savings for the solar power industry.

The technology combines on this effort is very versatile since it includes the use of CFD and a multifaceted interpolation approach. It is then worthy to mention that new technologies and research can be derived from this work. The reduction of system using the interpolation approach to extract dominant features can be vastly used on design applications in various disciplines such as image processing, signal analysis, data compression, process identification and control in chemical engineering, oceanography, vibration analysis, coherent structures in fluids, control of fluids, electrical power grids, and wind engineering to name a few.

The plan for Phase 2 is to build upon the expertise acquired in Phase I to develop, implement, test, and market the final Wind-Load Calculator as a product that can accurately and efficiently render the wind loads on PV system given a set of design parameters and can simultaneously provide recommendations as to the best-suitable installation configuration for the desired objectives and required constrains. To this end, tasks to be performed during Phase 2 of this project include but are not limited to:

- Extend parameterization to include all possible design variables such as array configuration, orientation, supports, elevation, topography, etc.
- Produce solid models within the design space of all parameters.
- Perform CFD analysis of the augmented set of models for all design variable ranges.
- Build augmented POD basis and interpolation network.
- Optimize POD interpolation.
- Program POD interpolation network on the cloud to access from mobile devices.
- Develop web-based application (calculator) to perform wind-load analysis and optimization.
- Test and deploy application on a variety of architectures.
- Develop life-cycle support system for the application.
- Pursue distribution network for licensing of the application.

6. Market interest, communications with partners, media attention and coverage

As part of presenting the Wind-Load Calculator to the solar market/community, we have created a flyer with the description of our product. The flyer can be seen below in Figure 32.

Figure 32. Wind-Load Calculator Flyer.

7. Publications/Presentations/Travel

- Solar Power International was held in Chicago, IL October 21-24, 2013. Stephen Barkaszi from FSEC attended the conference to network and to discuss the project with stakeholders and obtain feedback from potential user groups.
- A full paper was submitted and accepted for publication and presentation at the 8th International Conference on Inverse Problems in Engineering (ICIPE) (http://icipe2014.org/index.php/icipe2014/ICIPE2014). This conference will be held in Krakow, Poland, May 12-15, 2014. A representative from the team will attend the conference to present the results from the research carried out on this project. In addition, as a result of this conference publication there will be an invitation to submit a full paper to appear on a special issue of the Journal of Inverse Problems in Science and Engineering (Taylor and Francis).
- A full paper was submitted and accepted for publication and presentation at the 10th International Conference on Heat Transfer, Fluid Mechanics, and Thermodynamics (HEFAT) co-sponsored by the International Centre for Heat and Mass Transfer (ICHMT) (http://edas.info/web/hefat2014/home.html). This conference will be held in Orlando, Florida, July 14-16, 2014. A representative from the team will attend the conference to present the results from the research carried out on this project. An extended version of this paper that includes the experimental verification presented in this report will be submitted for consideration to the ASME Journal of Solar Energy Engineering.

8. References

[1] G. M. Feldman D.; Barbose, R.; Darghouth, N.; James, T.; Weaver, S.; Goodrich, A.; Wiser. R., "Photovoltaic System Pricing Trends: Historical, Recent, and Near-Term Projections," National Renewable Energy Laboratory, Golden, CO2013.

[2] D. S. Kristen Ardani, Robert Margolis, Jesse Morris, Carolyn Davidson, Sarah Truitt, and Roy Torbert, "Non-Hardware (“Soft”) Cost-Reduction Roadmap for Residential and Small Commercial Solar Photovoltaics, 2013-2020," National Renewable Energy Laboratory, Golden, CO. August 2013.

[3] Erwin, J., Bitsuamlak, G., Chowdhury, A., Barkaszi, S., and Gamble, S. (2012) Full Scale and Wind Tunnel Testing of a Photovoltaic Panel Mounted on Residential Roofs. Advances in Hurricane Engineering: pp. 471-482. doi: 10.1061/9780784412626.041.

[4] Significant Changes to the Wind Load Provisions of ASCE 7-10, T. Eric Stafford, www.asce.org.

[5] Minimum Design Loads for Buildings and Other Structures. Reston, VA: American Society of Civil Engineers, 2010.

9. Appendix

9.1. POD Snapshot Fortran Code

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CCC CCC
CCC WINDLOAD CALCULATOR CCC
CCC CCC
CCC PURPOSE: CCC
CCC GENERATE POD SNAPSHOTS FROM CFD FILES CCC
CCC CCC
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C C-----------------------------------------------------------------------
C
INTEGER IOS
REAL*8 X(8000),Y(8000),Z(8000),P(8000)
REAL*8 U(8000,91)
CHARACTER*80 FILENAME
C C-----------------------------------------------------------------------
CC
COUNTERS
C
IMAX=117
JMAX=60
NMAX=IMAX*JMAX
C C-----------------------------------------------------------------------
CC
LOOP THROUGH WIND ANGLES AND SPEEDS
C
WRITE (*,*) 'GENERATING SNAPSHOTS FROM CFD FILES'
WRITE (*,*)
C
DO IS=1,7
ISPEED=80+20*(IS-1)
N1=(ISPEED)/100
N2=(ISPEED-100*N1)/10
N3=(ISPEED-100*N1-10*N2)
C
DO IA=1,13
IANGLE=30*(IA-1)
N4=(IANGLE)/100
N5=(IANGLE-100*N4)/10
N6=(IANGLE-100*N4-10*N5)
C
FILENAME='Data/'//
& 'Top_'//CHAR(N1+48)//CHAR(N2+48)//CHAR(N3+48)//'mph_'
& //CHAR(N4+48)//CHAR(N5+48)//CHAR(N6+48)//
& 'degrees.csv'
C
ISIA=13*(IS-1)+IA
C C-----------------------------------------------------------------------
CC
READ FILE
C
```

```
OPEN (21,FILE=FILENAME,STATUS='OLD',IOSTAT=IOS)
C
XMIN= 1.D+020
XMAX=-1.D+020
YMIN= 1.D+020
YMAX=-1.D+020
ZMIN= 1.D+020
ZMAX=-1.D+020
C
IF (IOS.EQ.0) THEN
DO N=1,NMAX
READ (21,*) P(N),PS,CP,TX,TY,TZ,T1,T2,T3,X(N),Y(N),Z(N)
IF (X(N).GT.XMAX) XMAX=X(N)
IF (X(N).LT.XMIN) XMIN=X(N)
IF (Y(N).GT.YMAX) YMAX=Y(N)
IF (Y(N).LT.YMIN) YMIN=Y(N)
IF (Z(N).GT.ZMAX) ZMAX=Z(N)
IF (Z(N).LT.ZMIN) ZMIN=Z(N)
END DO
END IF
C
CLOSE (21)
C C--------------------------------------------------------------------
CC
SET LIMITS
C
DX=(XMAX-XMIN)/DBLE(JMAX-1)
DY=(YMAX-YMIN)/DBLE(JMAX-1)
DZ=(ZMAX-ZMIN)/DBLE(IMAX-1)
C C--------------------------------------------------------------------
CC
ORDER SNAPSHOT
C
DO I=1,IMAX
DO J=1,JMAX
IJ=JMAX*(I-1)+J
IF ((IANGLE.GT.105).AND.(IANGLE.LT.285)) THEN
X1=XMAX-DBLE(J-1)*DX
ELSE
X1=XMIN+DBLE(J-1)*DX
END IF
Y1=YMIN+DBLE(J-1)*DY
IF ((IANGLE.LT.105).OR.
& ((IANGLE.GT.195).AND.(IANGLE.LT.255))) THEN
Z1=ZMIN+DBLE(I-1)*DZ
ELSE
Z1=ZMAX-DBLE(I-1)*DZ
END IF
RMIN=1.D+020
NN=1
DO N=1,NMAX
RR=(X(N)-X1)*(X(N)-X1)+(Y(N)-Y1)*(Y(N)-Y1)+(Z(N)-Z1)*(Z(N)-Z1)
IF (RR.LT.RMIN) THEN
RMIN=RR
NN=N
END IF
```

```
END DO
U(IJ,ISIA)=144.D+000*P(NN)
END DO
END DO
C C-------------------------------------------------------------------
C
WRITE (*,*)
& 'SPEED: '//CHAR(N1+48)//CHAR(N2+48)//CHAR(N3+48)//'mph, '//
& 'ANGLE: '//CHAR(N4+48)//CHAR(N5+48)//CHAR(N6+48)//'... Done'
END DO
END DO
C C-------------------------------------------------------------------
CC
WRITE SNAPSHOTS
C
OPEN (22,FILE='POD_Snapshots.dat')
DO N=1,NMAX
WRITE (22,'(91(E12.6,2X))') (U(N,M),M=1,91)
END DO
CLOSE (22)
C C-------------------------------------------------------------------
CC C-------------------------------------------------------------------
CC
GENERATE VERIFICATION DATA
C
WRITE (*,*)
WRITE (*,*) 'GENERATING VERIFICATION DATA FROM CFD FILES'
WRITE (*,*)
C
IS=1
ISPEED=90
N1=(ISPEED)/100
N2=(ISPEED-100*N1)/10
N3=(ISPEED-100*N1-10*N2)
C
DO IA=1,2
IANGLE=180*(IA-1)
N4=(IANGLE)/100
N5=(IANGLE-100*N4)/10
N6=(IANGLE-100*N4-10*N5)
C
FILENAME='Data/'//
& 'Ver_'//CHAR(N1+48)//CHAR(N2+48)//CHAR(N3+48)//'mph_'
& //CHAR(N4+48)//CHAR(N5+48)//CHAR(N6+48)//
& 'degrees.csv'
C
ISIA=2*(IS-1)+IA
C C-------------------------------------------------------------------
CC READ FILE
C
OPEN (21,FILE=FILENAME,STATUS='OLD',IOSTAT=IOS)
C
XMIN= 1.D+020
XMAX=-1.D+020
YMIN= 1.D+020
YMAX=-1.D+020
```

```
ZMIN= 1.D+020
ZMAX=-1.D+020
C
IF (IOS.EQ.0) THEN
DO N=1,NMAX
READ (21,*) P(N),PS,CP,TX,TY,TZ,T1,T2,T3,X(N),Y(N),Z(N)
IF (X(N).GT.XMAX) XMAX=X(N)
IF (X(N).LT.XMIN) XMIN=X(N)
IF (Y(N).GT.YMAX) YMAX=Y(N)
IF (Y(N).LT.YMIN) YMIN=Y(N)
IF (Z(N).GT.ZMAX) ZMAX=Z(N)
IF (Z(N).LT.ZMIN) ZMIN=Z(N)
END DO
END IF
C
CLOSE (21)
C C--------------------------------------------------------------------
CC
SET LIMITS
C
DX=(XMAX-XMIN)/DBLE(JMAX-1)
DY=(YMAX-YMIN)/DBLE(JMAX-1)
DZ=(ZMAX-ZMIN)/DBLE(IMAX-1)
C C--------------------------------------------------------------------
CC
ORDER SNAPSHOT
C
DO I=1,IMAX
DO J=1,JMAX
IJ=JMAX*(I-1)+J
IF ((IANGLE.GT.105).AND.(IANGLE.LT.285)) THEN
X1=XMAX-DBLE(J-1)*DX
ELSE
X1=XMIN+DBLE(J-1)*DX
END IF
Y1=YMIN+DBLE(J-1)*DY
IF ((IANGLE.LT.105).OR.
& ((IANGLE.GT.195).AND.(IANGLE.LT.255))) THEN
Z1=ZMIN+DBLE(I-1)*DZ
ELSE
Z1=ZMAX-DBLE(I-1)*DZ
END IF
RMIN=1.D+020
NN=1
DO N=1,NMAX
RR=(X(N)-X1)*(X(N)-X1)+(Y(N)-Y1)*(Y(N)-Y1)+(Z(N)-Z1)*(Z(N)-Z1)
IF (RR.LT.RMIN) THEN
RMIN=RR
NN=N
END IF
END DO
U(IJ,ISIA)=144.D+000*P(NN)
END DO
END DO
C C--------------------------------------------------------------------
C
```

```
      WRITE (*,*)
     & 'SPEED: '//CHAR(N1+48)//CHAR(N2+48)//CHAR(N3+48)//'mph, '//
     & 'ANGLE: '//CHAR(N4+48)//CHAR(N5+48)//CHAR(N6+48)//'... Done'
      END DO
C C-------------------------------------------------------------------------
CC
      WRITE SNAPSHOTS
C
      OPEN (22,FILE='POD_Verification.dat')
      DO N=1,NMAX
      WRITE (22,'(2(E12.6,2X))') (U(N,M),M=1,2)
      END DO
      CLOSE (22)
C C-------------------------------------------------------------------------
C
      END
```

9.2. POD-RBF Interpolation MathCad Code

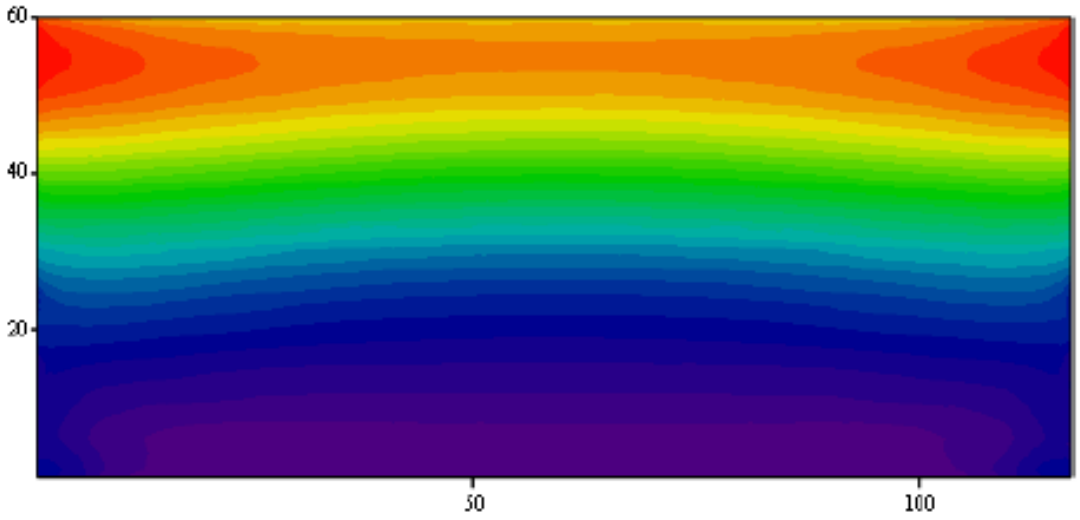
Read Snapshots

$U := READPRN("POD\_Snapshots.dat")$ $N := rows(U) = 7020$ $M := rows(U^T) = 91$

$I := 117$ $J := 60$

$i := 1..I$ $j := 1..J$ $is := 1$ $ia := 1$ $m := 13 \cdot (is - 1) + ia$

$P_{i,j} := U_{J \cdot (i-1)+j,\, m}$ $\min(P) = -14.0314$ $\max(P) = -5.43536$



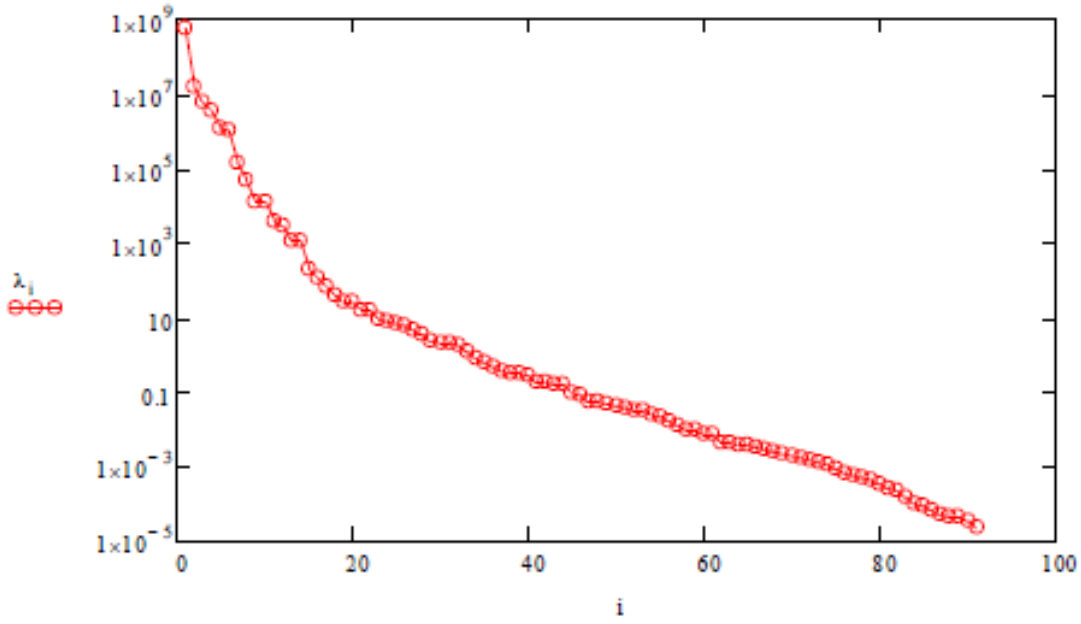
P

Perform POD

$C := 0 \qquad \lambda := 0 \qquad V := 0 \qquad \Phi := 0 \qquad W := 0$

$C := U^{T} \cdot U \qquad \lambda := \text{eigenvals}\ (C) \qquad \lambda := \text{reverse}\ (\text{sort}\ (\lambda))$

$i := 1 .. M$



Truncate POD Basis at K

$K := 12 \qquad j := 1 .. K \qquad V^{\langle j \rangle} := \text{eigenvec}\ (C, \lambda_j) \qquad V_{i,j} := \frac{V_{i,j}}{\sqrt{\lambda_j}}$

$\Phi := U \cdot V \qquad W := \Phi \cdot \Phi^{T} \cdot U$

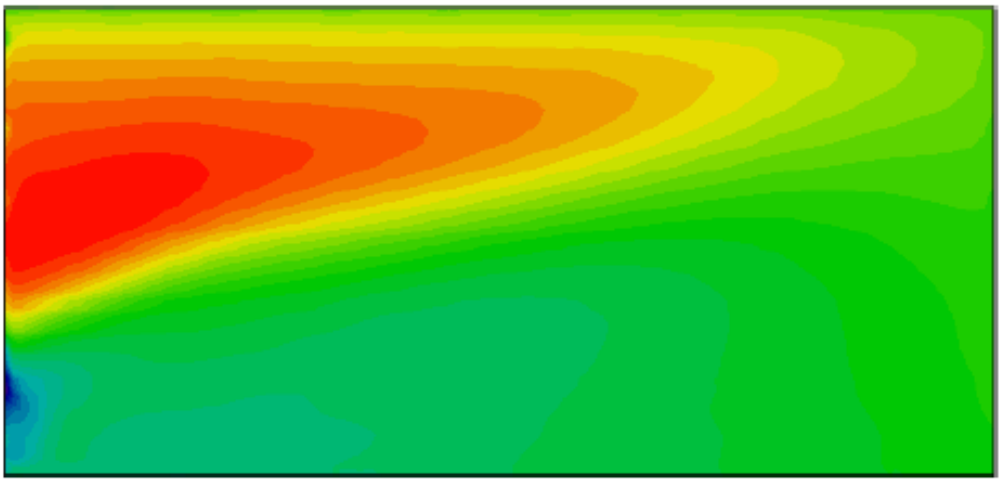
Compare reduced-order model W to original model U at is-speed and ia-angle:

$i := 1..I \qquad j := 1..J \qquad is := 4 \qquad ia := 2 \qquad m := 13\cdot(is-1)+ia$

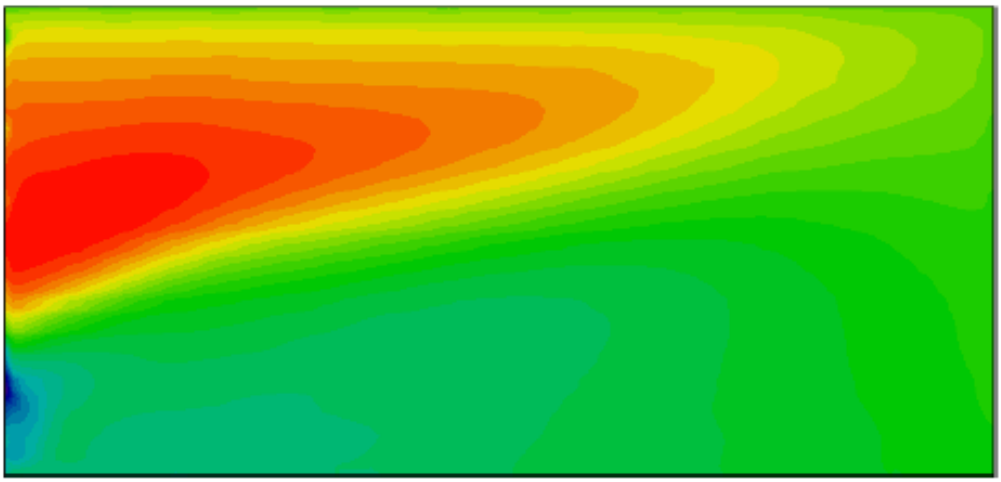
$P_{i,j} := U_{J\cdot(i-1)+j,m} \qquad \min(P) = -76.1764 \qquad \max(P) = -1.61912$

$Pw_{i,j} := W_{J\cdot(i-1)+j,m} \qquad \min(Pw) = -76.09527 \qquad \max(Pw) = -1.65806$

$$RMS := \frac{1}{\max(P)-\min(P)}\cdot\sqrt{\frac{1}{I\cdot J}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(P_{i,j}-Pw_{i,j}\right)^2} = 0.025604\cdot\%$$



P



Pw

Build POD-RBF Network:

Amplitude Matrix:

$A := \Phi^T \cdot U$ rows (A) = 12 rows (A^T) = 91

Inverse Multiquadric RBF: $c := 0.25 \cdot \sqrt{(200 - 80)^2 + 360^2} = 94.86833$

$$f(S, \alpha, S_i, \alpha_i) := \left[\frac{(S - S_i)^2 + (\alpha - \alpha_i)^2}{c^2} + 1 \right]^{\frac{-1}{2}}$$

$Ns := 7$ $i := 1..Ns$

$Na := 13$ $j := 1..Na$

$S_{Na \cdot (i-1)+j} := 80 + 20 \cdot (i - 1)$

$\alpha_{Na \cdot (i-1)+j} := 30 \cdot (j - 1)$

RBF Interpolation Matrix:

$Np := Ns \cdot Na$ $i := 1..Np$ $j := 1..Np$ $F_{i,j} := f(S_i, \alpha_i, S_j, \alpha_j)$ rows (F) = 91 rows (F^T) = 91

cond1 (F) = 5.24708×10^7

Expansion Coefficiens:

$B := A \cdot F^{-1}$ $C := \Phi \cdot B$ rows (C) = 7020 rows (C^T) = 91

Final Interpolation:

$$p(V, a, i) := \sum_{j=1}^{Np} \left(C_{i,j} \cdot f(V, a, S_j, \alpha_j) \right)$$

Write POD Coefficients and interpolation parameters to files:

WRITEPRN ("POD_RBF_Coefficients.dat") := C

WRITEPRN ("POD_RBF_Speeds.dat") := S

WRITEPRN ("POD_RBF_Angles.dat") := α

RBF Interpolation Test at is-speed and ia-angle:

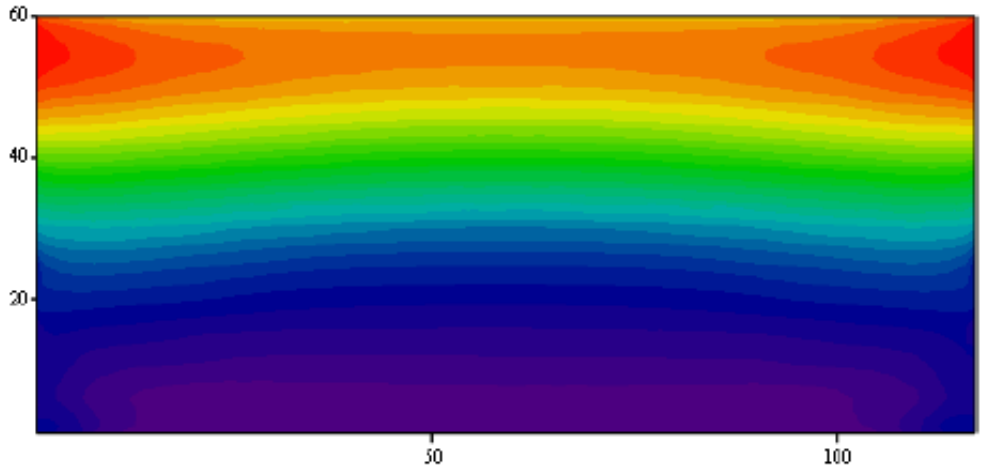
$is := 7$ $ia := 1$ $m := 13\cdot(is - 1) + ia$

$V := 80 + 20\cdot(is - 1)$ $a := 30\cdot(ia - 1)$

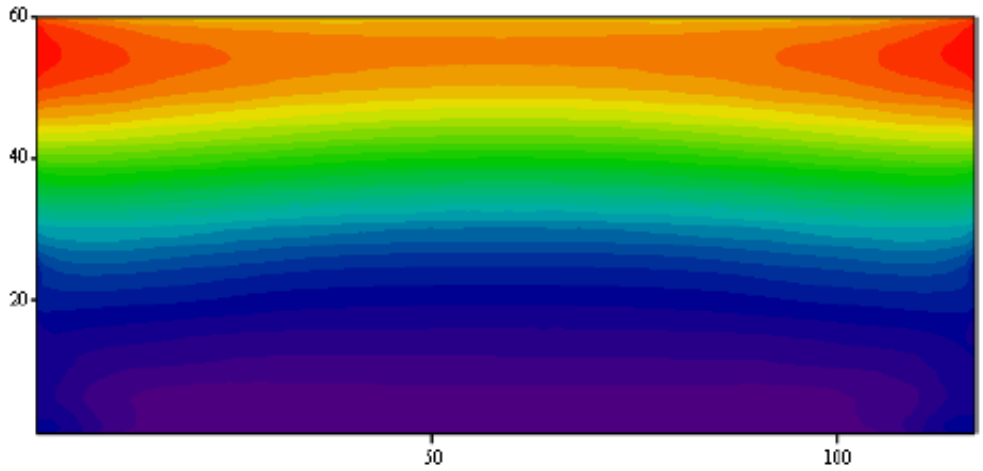
$i := 1..I$ $j := 1..J$

$P_{i,j} := U_{J\cdot(i-1)+j,m}$ $\min(P) = -87.656$ $\max(P) = -34.3602$

$P_{rbf}{}_{i,j} := p[V, a, J\cdot(i - 1) + j]$ $\min(P_{rbf}) = -87.65419$ $\max(P_{rbf}) = -34.18862$



P

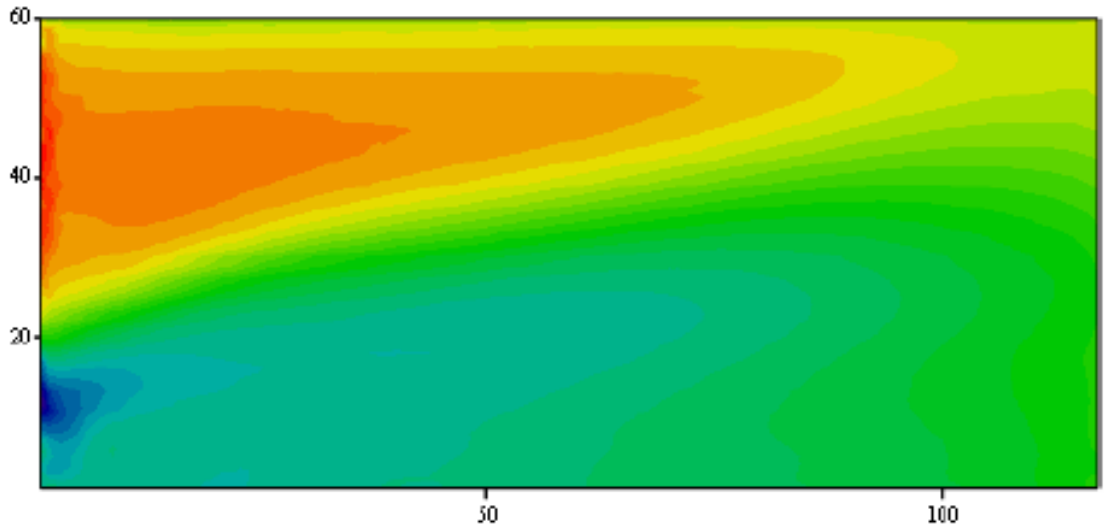


P_{rbf}

POD-RBF Snapshot at arbitrary Speed V and Angle a:

$V := 135$ $a := 13$

$P_{rbf_{i,j}} := p[V, a, J\cdot(i-1)+j]$ $\min(P_{rbf}) = -71.81569$ $\max(P_{rbf}) = -5.01103$



P_{rbf}

Average Pressure (psf): $P_m := \text{mean}(P_{rbf})$ $P_m = -30.67993$

Uplift (lbf): $F_u := P_m\cdot\left[\left(9\cdot\frac{39}{12}\right)\cdot\left(3\cdot\frac{60}{12}\right)\right]$ $F_u = -13460.81795$

POD-RBF Verification using intermediate CFD data (90mph, 0 degrees):

$U := \text{READPRN}(\text{"POD\_Verification.dat"})$ $N := \text{rows}(U) = 7020$ $M := \text{rows}(U^T) = 2$

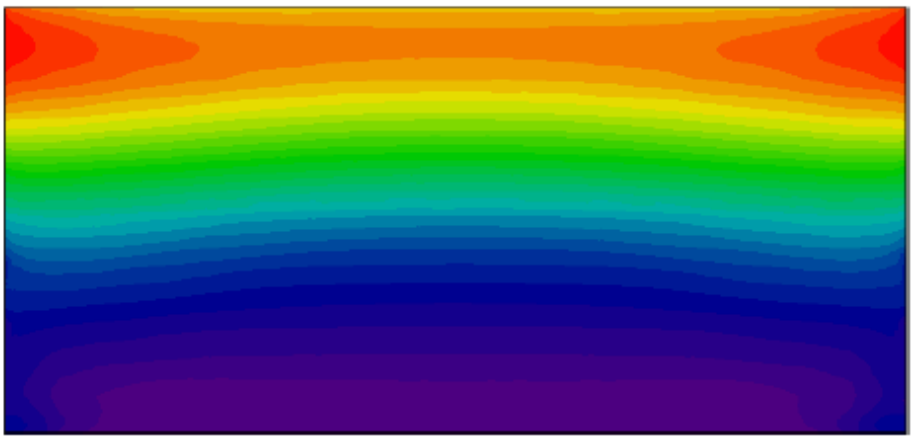
$I := 117$ $J := 60$

$i := 1..I$ $j := 1..J$

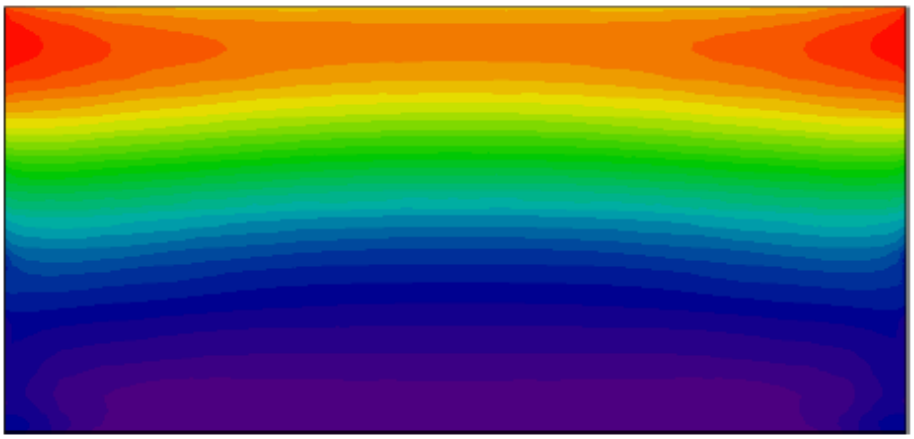
$m := 1$ $P_{CFD_{i,j}} := U_{J\cdot(i-1)+j,m}$ $\text{mean}(P_{CFD}) = -13.15707$

$V := 90$ $a := 0$ $P_{POD_{i,j}} := p[V, a, J\cdot(i-1)+j]$ $\text{mean}(P_{POD}) = -13.12537$

$$RMS := \frac{1}{\max(P_{CFD}) - \min(P_{CFD})} \cdot \sqrt{\frac{1}{I\cdot J} \sum_{i=1}^{I} \sum_{j=1}^{J} \left(P_{CFD_{i,j}} - P_{POD_{i,j}}\right)^2} = 0.339198 \cdot \%$$



P_{CFD}



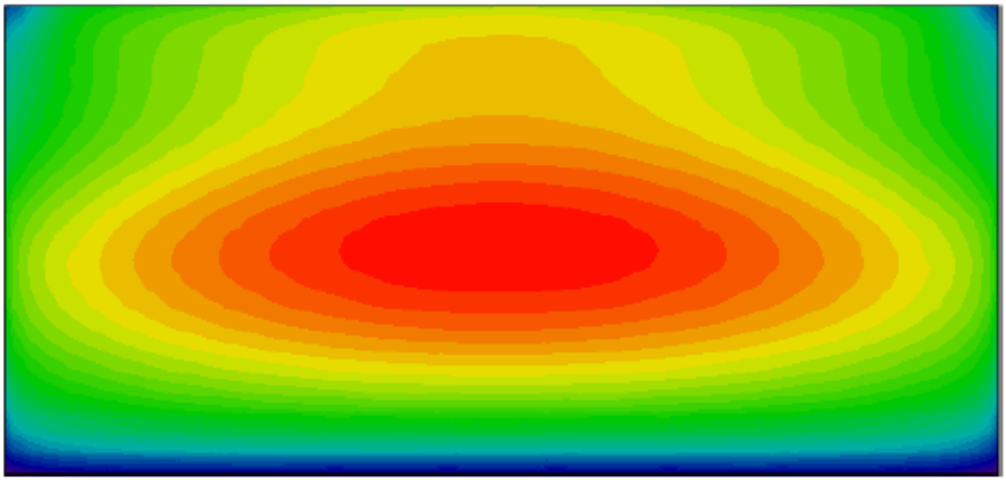
P_{POD}

POD-RBF Verification using intermediate CFD data (90mph, 180 degrees):

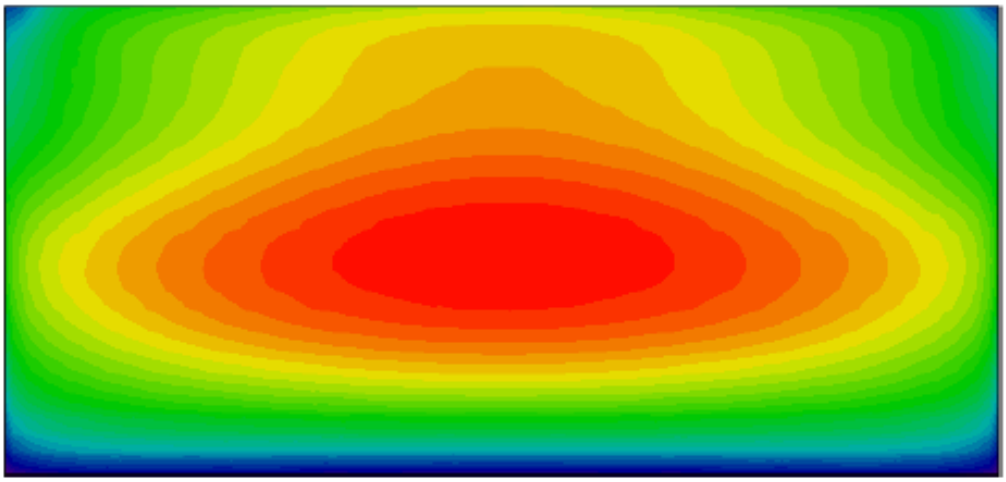
$m := 2$ $\quad P_{CFD_{i,j}} := U_{J\cdot(i-1)+j,m}$ $\quad mean\left(P_{CFD}\right) = -8.14446$

$V := 90$ $\quad a := 180$ $\quad P_{POD_{i,j}} := p[V, a, J\cdot(i-1)+j]$ $\quad mean\left(P_{POD}\right) = -8.14439$

$$RMS := \frac{1}{\max\left(P_{CFD}\right) - \min\left(P_{CFD}\right)} \cdot \sqrt{\frac{1}{I\cdot J}\sum_{i=1}^{I}\sum_{j=1}^{J}\left(P_{CFD_{i,j}} - P_{POD_{i,j}}\right)^2} = 0.986829\cdot\%$$



P_{CFD}



P_{POD}

9.3. POD-RBF Implementation C# Code

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WindLoadCalculator
{
	public class WindLoad
	{
		public double[] V = {80,80,80,80,80,80,80,80,80,80,80,80,80
		,100,100,100,100,100,100,100,100,100,100,100,100,100
		,120,120,120,120,120,120,120,120,120,120,120,120,120
		,140,140,140,140,140,140,140,140,140,140,140,140,140
		,160,160,160,160,160,160,160,160,160,160,160,160,160
		,180,180,180,180,180,180,180,180,180,180,180,180,180
		,200,200,200,200,200,200,200,200,200,200,200,200,200 };

		public double[] A = {0, 30, 60, 90,120,150,180,210,240,270,300,330,360,
		0, 30, 60, 90,120,150,180,210,240,270,300,330,360,
		0, 30, 60, 90,120,150,180,210,240,270,300,330,360,
		0, 30, 60, 90,120,150,180,210,240,270,300,330,360,
		0, 30, 60, 90,120,150,180,210,240,270,300,330,360,
		0, 30, 60, 90,120,150,180,210,240,270,300,330,360 };

		public double[,] C = new double[7020, 91];
		public double[] Pressure = new double[7020];

		public WindLoad()
		{
			ReadC();
		}

		public void ReadC()
		{
			StreamReader reader1 = File.OpenText(@"C:\Users\reaglin\Documents\Visual
			Studio2012\Projects\WindLoadCalculator\WindLoadCalculator\POD_RBF_Coefficients_csv
			.txt");
			string str = reader1.ReadToEnd();
			reader1.Close();
			reader1.Dispose();

			string[] s = str.Split(new char[] { ',' });
			for (int i = 0; i<7020; i++)
			{
				for (int j = 0; j < 91; j++)
				{
					C[i,j] = Convert.ToDouble(s[i*91 + j]);
				}
			}
		}

		public void CalculatePressure(double v, double a)
		{
			double c = 0.25 * Math.Sqrt(Math.Pow(200 - 80, 2) + Math.Pow(360, 2));
			for (int i = 0; i < 7020; i++)
			{
```

```
                                        Pressure[i] = 0;
                                        for (int j=0; j<91; j++)
                                        {
                                                Pressure[i] = Pressure[i] + C[i,j]/Math.Sqrt((Math.Pow(v-
                                                V[j],2)+Math.Pow(a- A[j],2))/Math.Pow(c,2) +1);
                                        }
                                }
                        }
                }
        }
```

9.4. GUI JavaScript Code

```
.block label { display: inline-block; width: 240px; text-align: right; }
.block input {width: 60px; text-align: left }
.progress-label {
position: relative;
left: 30%;
top: 4px;
font-weight: bold;
text-shadow: 1px 1px 0 #fff;
}
.CalculateButton {
	-moz-box-shadow:inset 0px 1px 0px 0px #f9eca0;
	-webkit-box-shadow:inset 0px 1px 0px 0px #f9eca0;
	box-shadow:inset 0px 1px 0px 0px #f9eca0;
	background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #f0c911), color-stop(1, #f2ab1e) );
	background:-moz-linear-gradient( center top, #f0c911 5%, #f2ab1e 100% );

	background-color:#f0c911;
	-webkit-border-top-left-radius:28px;
	-moz-border-radius-topleft:28px;
	border-top-left-radius:28px;
	-webkit-border-top-right-radius:0px;
	-moz-border-radius-topright:0px;
	border-top-right-radius:0px;
	-webkit-border-bottom-right-radius:28px;
	-moz-border-radius-bottomright:28px;
	border-bottom-right-radius:28px;
	-webkit-border-bottom-left-radius:0px;
	-moz-border-radius-bottomleft:0px;
	border-bottom-left-radius:0px;
	text-indent:0px;
	border:1px solid #e65f44;
	display:inline-block;
	color:#c92200;
	font-family:Arial;
	font-size:17px;
	font-weight:bold;
	font-style:normal;
	height:65px;
	line-height:65px;
width:195px;
```

```
		text-decoration:none;
		text-align:center;
		text-shadow:1px 1px 0px #ded17c;
}

.CalculateButton:hover {
		background:-webkit-gradient( linear, left top, left bottom, color-stop(0.05, #f2ab1e), color-stop(1, #f0c911) );
		background:-moz-linear-gradient( center top, #f2ab1e 5%, #f0c911 100% );
		background-color:#f2ab1e;
}
		.CalculateButton:active {
			position:relative;
			top:1px;
}

#progressbar .ui-progressbar-value {
background-color: #ccc;
}

<h1>Wind Load Calculator</h1>
<table>
		<tr>
			<td valign="center">
<div id="slider-range-velocity" style="float:left; height: 160px"></div>
&amp;nbsp;&amp;nbsp;<input type="text" id="velocity" style="border:0; color:#f6931f; font-weight:bold; width: 120px; font-size:
50px;float:right; " />
		</td>
		<td>
			<input type="text" class="knob" id="angle" style="border:0; color:#f6931f; font-weight:bold; font-
		size:60px" data-cursor="true" data-fgcolor="#f6931f" /><br/><br/>
		</td>
		<td>
<div id="heatmapArea" style="width:702px;padding:0;height:360px;cursor:pointer;position:relative;">

<canvas id="canvas1" width="702" height="360"></canvas><br/>
</div>
		</td>
</tr>
<tr>
		<td>
		<label for="amount">Wind Velocity (mph)</label>
		</td>
		<td>
			<label for="amount">Wind Direction (degrees)</label>
		</td>
		<td>Lift (lbf) <div id="lift"></div> </td>
</tr>
<tr>
<td colspan="2"> Move sliders to set wind velocity and wind direction. <br/>
<div id="progressbar">
			<div class="progress-label"></div>
		</div>
```

```
		</td>
		<td>
		<button id="calculate" class="CalculateButton">Calculate</button>
		</td>
		</tr>
</table>

$(document).ready(function () {
		$(".knob").knob({
			min:0,
			max:360
				});
		$("#drawGrid").click(
			function () {
				drawHeatMap();
			});
		$( "#progressbar" ).progressbar({
value: 0
});
		$("#calculate").mousedown(
			function () {
			$( "#progressbar" ).progressbar({
		value: false
});
		$(".progress-label").text( "Calculating" );
});

		$("#calculate").click(
			function () {

			drawHeatMap();
				$( "#progressbar" ).progressbar({
		value: 100
});
		$(".progress-label").text( "Complete!" );
		});
	});
$( "#progressbar" ).progressbar({
value: 0,

});

$( "#slider-range-velocity" ).slider({
		range: "min",
		value: 80,
		min: 80,
		max: 200,
		orientation: "vertical",

slide: function( event, ui ) {
$( "#velocity" ).val( ui.value );
}
});
```

```
$( "#velocity" ).val( $( "#slider-range-velocity" ).slider( "value" ) );

		var trueWidth = 351;
		var trueHeight = 180;
		var margin = 1;
		var nRows = 3;
		var nColumns = 9;


function Point(x,y)
{
		this.x = x;
		this.y = y;
}
var heatmap;
var maxWindLoad;
function drawHeatMap() {
		maxWindLoad = getMaxWindLoad();
		var cGradient =
			{ 0.1: "rgb(0,255,0)", 1.0: "rgb(255,0,0)"};
var hmConfig = {
		element: document.getElementById('heatmapArea'),
		radius: 20,
		opacity: 50,
		gradient: cGradient,
		legend: {
			position: 'br',
		title: 'Wind Load Values'
},
};

			heatmap = h337.create(hmConfig);
		var data = {
max: maxWindLoad,
data: [
{ x: 100, y: 200, count: windLoad(100,200) }

// ...
]
		}
			heatmap.store.setDataSet(data);
			drawGrid();
			heatmapWindLoads();
}


function drawGrid() {
		var canvas = document.getElementById('canvas1');
		var ctx = canvas.getContext('2d');

		/* Generates a grid based on the canvas size */
		var xSize = (canvas.width - 2 * margin) / nColumns;
		var ySize = (canvas.height - 2 * margin) / nRows;

		drawGridLines(ctx, margin, margin, xSize, ySize);
```

```
		drawGridLines(ctx, margin+2, margin-2, xSize, ySize);

}

function drawGridLines(ctx, x0, y0, xSize, ySize) {
		// Draw Y lines
		for (var i = 0; i < nColumns + 1; i++) {
			ctx.lineWidth = 2;
			ctx.beginPath();
			ctx.moveTo(x0 + i * xSize, y0);
			ctx.lineTo(x0 + i * xSize, y0 + nColumns * xSize);
			ctx.stroke();
}
		// Draw X lines
		for (var i = 0; i < nRows + 1; i++) {
			ctx.lineWidth = 2;
			ctx.beginPath()
			ctx.moveTo(x0, y0 + i * ySize);
			ctx.lineTo(x0 + nColumns * ySize, y0 + i * ySize);
			ctx.stroke();
}
}

function cellUpperLeft(rowNum, colNum)
{
		var canvas = document.getElementById('canvas1');

		/* Generates a grid based on the canvas size */
		var xSize = (canvas.width - 2 * margin) / nRows;
		var ySize = (canvas.height - 2 * margin) / nColumns;
		var ulX = margin + rowNum * xSize;
		var ulY = margin + colNum * ySize;

		return new Point(ulX, ulY);
}

function cellLowerRight(rowNum, colNum)
{
		var canvas = document.getElementById('canvas1');
		/* Generates a grid based on the canvas size */
		var xSize = (canvas.width - 2 * margin) / nRows;
		var ySize = (canvas.height - 2 * margin) / nColumns;
		var ulX = margin + (rowNum + 1) * xSize;
		var ulY = margin + (colNum + 1) * ySize;

		return new Point(ulX, ulY);
}


Function calculate()
{
		drawGrid();
		var canvas = document.getElementById('canvas1');
		var context = canvas.getContext('2d');
```

```
		// Min and Max on drawn axis
		// margin is global
		var minX = margin;
		var maxX = canvas.width - 2*margin;
		var minY = margin;
		var maxY = canvas.height - 2*margin;

		var scaleX = trueWidth/(maxX - minX);
		var scaleY = trueHeight/(maxY - minY);

		var maxWindLoad = getMaxWindLoad();
		alert("Max Wind Load Calculated as " + maxWindLoad);

		for (var x = 0; x < nRows ; x++)
		{
			for (var y=0; y<nColumns; y++)
			{
				// now that we have the max wind load
		var upperLeft = cellUpperLeft(x,y);
		var lowerRight = cellLowerRight(x,y);

		// need to transform to actual coordinates
		var ulWindLoad = windLoad(upperLeft.x, upperLeft.y);
		var lrWindLoad = windLoad(lowerRight.x, lowerRight.y);

		var csUL = ulWindLoad/maxWindLoad;
		var csLR = lrWindLoad/maxWindLoad;
		drawGradient(upperLeft.x,upperLeft.y,lowerRight.x,lowerRight.y, csUL, csLR);
			}
		  }
		}

		function heatmapWindLoads()
		{
			var canvas = document.getElementById('canvas1');
			var velocity = Number($('#velocity').val());
			var angle = Number($('#angle').val());

			var scaleX = canvas.width / trueWidth;
			var scaleY = canvas.height / trueHeight;
			var increment = 6;
			for (var x = 0; x < trueWidth; x+=increment)
			{

			// x is the true X, y is the true Y
			for (var y=0; y<trueHeight; y+=increment)
			{
				var wl = windLoad(velocity, angle, x, y);
				// only need to plot every other point
				heatmap.store.addDataPoint(x * scaleX + margin, y * scaleY + margin, wl);
			}
			}
}
function getMaxWindLoad()
{
```

```
		var velocity = Number($('#velocity').val());
		var angle = Number($('#angle').val());

		// trueWidth and trueHeight are global
		var scaleMax = 0;
		for (var x = 0; x < trueWidth; x++)
		{
			// x is the true X, y is the true Y
			for (var y=0; y<trueHeight; y++)
			{
			var wl = windLoad(velocity, angle, x,y);
			if (wl > scaleMax)
			{
				scaleMax = wl;
			}
		}
		}
		return scaleMax;
}
function windLoad(velocity, angle, x, y) {
		// The wind load function will use the true X and true Y
		// This is a dummy function
		/* This is simply a dummy function for the wind load */
		return (velocity*x + angle*y);
}
```