

# 2015 University Turbine Systems Research (UTSR) Industrial Fellowship Program

---

## Turbine Aerodynamics

Prepared For:



General Electric  
300 Garlington Road  
Greenville, SC

&



Southwest Research Institute  
P.O. Drawer 28510  
San Antonio, TX

Prepared By:  
Kathryn Kempema  
University of Wyoming  
Mechanical Engineering

## 1. Introduction

The GE Power & Water Turbine Aerodynamics group works out of Greenville South Carolina. One of the group's main focuses is comparing experimental data with the outputs from predictive Computational Fluid Dynamics (CFD) programs. These comparisons are used to quantify the performance of the CFD programs and identify ways in which the CFD modeling can be improved. This internship consisted of three major projects. The first project was to write a Post Processing CFD code in MATLAB. The second project was to write a data reduction code to support turbine rig validation testing. The final project was to use CFD to explore geometry features to optimize cavity swirlers. The basic schedule of these projects can be seen below in Table 1.

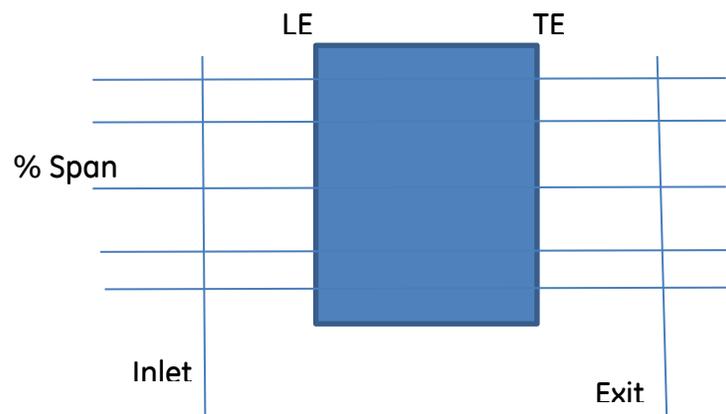
**Table 1.** Summer Work Schedule

	June 1 2015	June 8 2015	June 15 2015	June 22 2015	June 29 2015	July 6 2015	July 13 2015	July 20 2015	July 27 2015	August 3 2015	August 10 2015
	FW23	FW24	FW25	FW26	FW27	FW28	FW29	FW30	FW31	FW32	FW33
Post Processing CFD	Yellow										
Phase 1 Data Reduction	Green										
CFX Training	Red										
CFD Project							Blue				
Report out										Orange	

## 2. Post Processing CFD Tool

GE Turbine Aerodynamics currently uses several tools to post-process CFD results. These tools, however, do not lend themselves easily to customization. A MATLAB program was desired to allow for such customization in support of on-going projects.

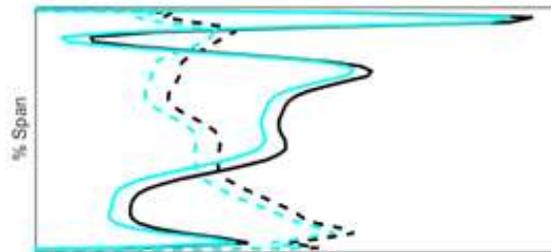
The basic design process starts with the airfoils being designed and run through a CFD program. Then using a script, radial or span average quantities are pulled from the CFD model (See Figure 1) and printed out in '.dat' files. The MATLAB script then collects all the data from the '.dat' files, and creates the post-processing plots.



**Figure 1.** The blue rectangle represents the airfoil. Horizontal lines represent % Spans, and vertical lines represent the Inlet, Exit, Leading Edge (LE) and Trailing Edge (TE) planes.

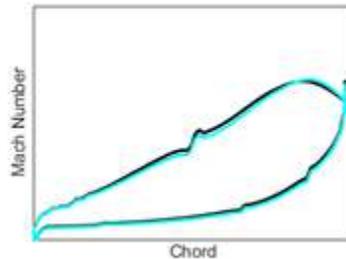
There are two different types of plots the MATLAB script is set up to make. First using the inlet and exit planes, the data with respect to percent span can be shown on the same plot (see

Figure 2) or separate plots (as defined by the user). On any given plots, up to seven design sets can be compared. The script leverages a zoom sub-function such that extreme values, which occasionally appear at the edges, don't eliminate the user's ability to view the central trend.



**Figure 2.** Plot of variable vs. % span at inlet and exit plane

The second type of plot is a plot of the distribution on the airfoil surface of Mach number vs chord (see Figure 3). This plot is made for selected spans defined by the user.



**Figure 3.** An example of airfoil surface Mach number made by the MATLAB script

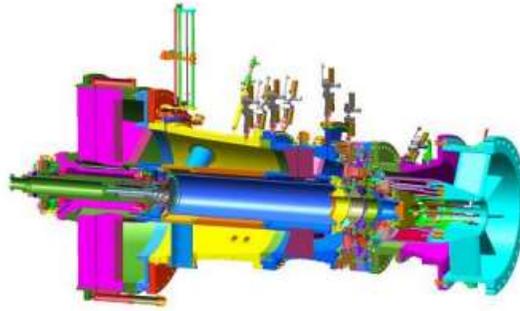
The program is set up to make both styles of plot described above for two, three or four stage turbine designs. For convenience the program is also set up to allow the user to specify color order to ensure they can tell the lines apart on the plots.

The program complements the post-processing tools available to the designer, providing useful flexibility and customization to the plots used during the design process (in one step) while providing a platform to quickly compare multiple designs.

### **3. Data Reduction Code to Support Turbine Rig Testing**

This project is in support of rig tests to be performed in a validation test vehicle which uses multiple sensors to gather temperature and pressure data for a range of operating conditions. The data collected is then analyzed and compared to CFD models to verify their validity.

Due to instrumentation differences between this test and previous ones, the data reduction code needed to be updated, and rewritten from FORTRAN to MATLAB for the test facility.



**Figure 4.** Schematic of validation rig.

A MATLAB template available from a prior test was leveraged and integrated with the older data reduction code. The challenge of the project was to translate the Fortran code to MATLAB and then combine/compare the two codes: this involved a good deal of debugging and figuring out the internal function structures, as well as re-processing known test data to ensure accuracy between the old and new codes.

Currently the 54 page FORTRAN code has been translated into MATLAB and debugged. One interesting aspect of this project was examining the idiosyncrasies of the MATLAB and FORTRAN coding platforms.

## **4. Swirler Design**

### **4.1 Swirler Introduction**

In a gas turbine engine, there is a cavity below the turbine flowpath between the bladed rotor, which is rotating, and the vane row, which is stationary. Air (from a compressor source) is being introduced in the cavity. The purpose of the air is to insure that no hot air from the main flow path gets ingested into the cavity. When the cool air exits the cavity through an overlap discourager and enters the main flow path, local mismatch in merging air stream angle results in mixing losses adversely impacting turbine efficiency.

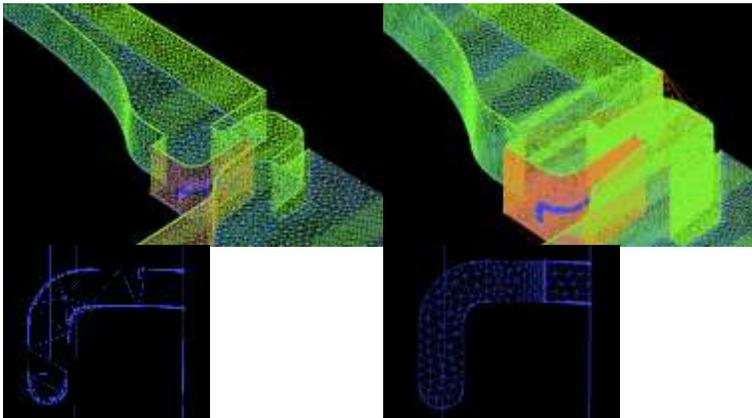
Cavity swirlers can be designed to direct the flow so that mixing and boundary flow losses across the blade decrease. A swirler is an object put in the cavity to direct the flow and maximize the swirl. However, cavity swirlers do cause an increase in the torque needed to turn the blade, and therefore the benefit of reducing the mixing losses need to be balanced against the increase of rotor torque. This project explores through CFD the impact of several design parameters on the swirl.

### **4.2 Background and Model Design**

The geometry for the study is shown below in Figure 5. This model is set up so the user defines a sector of the total 360 degrees that the part takes up and uses rotational symmetry to find the total torque. The geometric model is built in NX 9.0.3 (UG), meshed in ICEM, and tagged in CFX for the boundary conditions.

The CFD analysis and post-processing was done in CFX. Monitor points were used to track convergence of the results at key locations in the CFD domain. For every CFD case, the torque, swirl factor, and swirler efficiency were calculated from the results and compared.

Initially all of the design factors were examined using a coarse mesh (running on local desktop). To verify the coarse mesh's results, select fine mesh cases were also run and compared. Figure 5, illustrates both the coarse mesh (left) and the fine mesh (right).



**Figure 5.** Details of ICEM coarse (left) and fine mesh (right)

### 4.3 Tested Variables & Results

Multiple quantities were varied in order to explore the design space. Due to the proprietary nature of the study, the details are omitted.

- First, the purge mass flow was varied.
- Second, a count study was performed.
- Finally, several swirler geometric parameters were varied.

When all results are condensed into a single plot of cavity swirl vs rotor torque (not shown), the relationship appeared to be linear. The fine mesh results also followed a similar linear trend. Overall these CFD models provided a good preliminary view of the effects of different geometries, counts or purge flow on the flow physics, providing useful trends to the turbine designer.

### 5. Acknowledgements

I'd like to thank all the members of Turbine Aerodynamics Department at GE and my manager Gunnar Siden, for having me on the team. I learned a lot about the process of designing turbine airfoils and appreciate everyone in the group's willingness to explain concepts and help me grow my skills. Thanks to Sylvain Pierre, Scott Holloway and Justin Mussetto for mentoring me though the projects and answering a gigantic number of questions. Additional thanks to the SWRI for funding me to learn more about this interesting topic.