The Use of Advanced Simulation Tools in Capstone Design Projects

Thomas J. Barber
Department of Mechanical Engineering
University of Connecticut
Storrs, CT 06269
United States of America
barbertj@engr.uconn.edu

Abstract: - Capstone design courses are run in many different ways in the academic community. Many universities strive to promote immersion into the real world of engineering through industrially sponsored projects. While this approach offers many immediate benefits to near-graduating seniors, it introduces many unique problems to the academic community. Real-world problems differ significantly from academic courses included in the curriculum of most engineering institutions. Since students typically have little exposure to advanced simulation software tools because of their steep learning curve, new education approaches are needed to accommodate such programs. This paper addresses the experience of several senior design teams at the University of Connecticut in using commercial software to support their design project.

Key-Words: - Capstone design, simulation software

1 Introduction

Senior level capstone design courses are run in many different ways in the academic community. Many institutions have programs based on developing a single product entity, like an SAE or mini-Baja vehicle [1]. Small student teams work on individual vehicle components, but the goal is to complete a single entity for some series of trials at the end of the program. Other institutions do not require the development of a working model; rather their goal is a paper design study, e.g. an airplane or a model manufacturing plant. Some institutions have their students work at developing new experiments to support their academic program, thereby modernizing their thermal-fluids and dynamics laboratory experiments on a regular basis. In contrast to these approaches, a growing number of institutions strive to promote immersion into the real world of engineering through industrially sponsored projects. While this approach offers many immediate benefits to near-graduating seniors, it introduces many unique problems to the undergraduate student body.

Many U.S. engineering schools have moved towards a 128 credit degree requirement. This is driven by concerns of economic competitiveness and an effort to keep the engineering degree to 4 – 4.5 years in length. This results in seniors having little depth beyond the basic engineering [ME] curriculum. Such a narrowly focused education can represent a challenge to capstone design students, where the local sponsoring companies technical and applications are extremely diverse. At the University of Connecticut (UConn), while the aerospace / defense industries are big supporters of the program, local sponsors specialize in computer chip photolithography, elevator design, production of personal care products, and development of surgical products [2]. Although the projects tend to fall into general categories: thermal-fluid problems, applied mechanics problems, manufacturing process or product development, the sophistication and depth of these problems is frequently beyond the students academic training at the onset of the program.

Many projects of a mechanical nature follow the classical design paradigm taught in class. For thermal-fluid, applied mechanics, and material processing driven projects, the goal may not be a mechanical design, but to develop ideas that could lead to either improved design concepts or processes. In industry, this is seen in terms of efforts to reduce that amount of testing as a means to shorten design time and lower development costs. Therefore, industrially defined projects may have either a larger mechanical or a larger analytical / computational element. For these latter projects, students must become rapidly knowledgeable in advanced computational fluid dynamics (CFD) or finite element analysis (FEA) software.

One needs to remember that the typical focus of classroom education is largely based on analytical work. The laboratory component of engineering courses normally involves preset and predefined experiments to verify classroom concepts to expose students to physical numbers. Simple software is being currently used to eliminate some of the more tedious tasks students perform. Java based software
enables students to eliminate tedious table interpolation tasks in Compressible Flow studies and thereby permit assignment of more interesting complex problems. It is much less common for computational software to be integrated into the curriculum of such a course [4-7]. Different schools of thought exist as to the best approach for introducing students to CFD. Students are offered the opportunity to explore the concepts they are learning and develop a feel for the parameters that control fluid motion using Fluent’s FlowLab [8] software, which uses a set of predefined fluid mechanics problems under a single interface.

Use of general or commercial CFD solvers, in general, has been viewed as not appropriate for classroom work due to their high learning curve and how long it takes to gain proficiency in their use. Intermediate and advanced level CFD courses teach modeling and numerical methods using textbooks, computer programming assignments and specialty or commercial software and are open as an elective to undergraduate students. In general, as senior level student start their capstone program, they do not have this academic training. Introductory lectures of accuracy, convergence and grid resolution-distribution are therefore necessary to introduce students to the starting concepts for doing numerical simulations. Some sites have even provided step-by-step procedures for setting up and completing simulations of basic fluid flow problems, e.g. laminar flow in a pipe, turbulent flow in a pipe, etc. There are however several issues that are not as easily handled and represent major hurdles for undergraduate level students, i.e. (1) breaking down a complex problem into a series of solvable problems, (2) the difficulty of going from a CAD representation to a satisfactory grid, (3) navigating through the large number of options in a typical CFD code, (4) learning about complex physics and (5) learning how to use and interpret the results in order to influence a design. Not all of these issues will be explored in the following paper.

A brief comment about our typical student’s contact with CFD software is worth noting. Most have gained some limited CFD experience with simple preset CFD problems using Fluent’s FlowLab software [8], a derivative software package of the generalized CFD Fluent software [9], through use in their Fluid Mechanics courses. The biggest problem the students faced in these exercises was not in the use of the software or of understanding the fluid mechanics principles, rather it was in analyzing and presenting their results in a coherent and meaningful fashion. This paper will explore in greater details each of the types of hurdles faced by the students and then cite examples from recent senior design projects to illustrate how the students successfully and unsuccessfully handled these problems.

2 Student Software Training

The students in the UConn ME senior design program have a typical undergraduate Mechanical Engineering degree background. As they enter the senior year, they have had one or two semesters of Fluid Mechanics and some exposure to Numerical Methods. In order to familiarize the students with CFD concepts, 2-3 abbreviated lectures similar to Bhaskaran’s notes [10] are presented. The topics covered include differencing concepts, gridding, turbulence modeling, accuracy, convergence, etc. The biggest conceptual problems for undergraduate students are turbulence, including issues of y+ and wall functions/wall integration. The students are then encouraged to work through several FlowLab exercises. Computational grids are built in, with the user having the discretion to select either a coarse, medium or fine resolution mesh. While the problems are preset, they connect the lecture material to numerically predicted flow physics.

The CFD software used in all cases at the University of Connecticut was the commercially available software FLUENT [9], while the mesh generation software used was Fluent’s GAMBIT software. Initial training in the use of Fluent was achieved by working through Bhaskaran’s web site [10] of basic flow training test cases. The students were asked to work through a series of basic validation studies comparing predictions to either analysis or data, e.g. laminar and turbulent flow [11] through a straight pipe, jet flow, etc. Next, a selected series of FLUENT training tutorials were recommended. The advantage of these tutorials was that a mesh was provided for each tutorial. Furthermore, each tutorial came with detailed instructions. Finally, students were encouraged to break down their application into a series simpler, but known problems and perform a selected set validation studies on them.

Through all of these exercises, the students learned the basics of executing the code and presenting solution results. While learning how to use Fluent, the students were urged to maintain an on-line FAQ directory to enable other user’s to accelerate their proficiency in the use of FLUENT. This was subdivided into four separate categories: (1) Interface-GUI, (2) CAD to grid, (3) Solver convergence, and (4) Post-processing. This is a
student maintained, chat-room type tool that has been successful over the past year.

3 Student CFD Code Usage Issues

3.1 CAD to Grid

Ever since CFD has been used by industrial engineers to analyze practical problems, grid generation has consumed a major portion of the analysis cycle time. Many experienced users cite their experience in terms of the qualitative time bar shown in Fig. 1. As long as geometries have simple topologies, students have demonstrated their ability to construct structured or tetrahedral grids with little difficulty. When the analysis of more elaborate configurations is needed, most students have difficulty in developing gridding strategies that experienced users have learned. As cited earlier, one advantage of the FlowLab software concept was the use of a pre-built mesh. Stern et al. [12] have tried to further generalize the use of CFD in the classroom environment, with the creation of a CFD interface. While this allows the user greater flexibility on the flow side of the problem, it still utilized the pre-built mesh approach.

![Figure 1: CFD analysis cycle time](image)

Consider for example the case of one student team’s experience, working on a project for Westinghouse Electric Company, a leader in providing services, technology, and equipment for the commercial nuclear power industry asked their student design team to investigate issues with the thrust bearing assembly in their reactor coolant pump, see Fig. 2.

![Figure 2: Reactor coolant pump and thrust bearing](image)

The thrust bearing assembly is regarded as one of the more important parts of the reactor coolant pump. It ensures the successful operation of the reactor coolant pump, and consequently the success of the entire plant. The thrust bearing assembly provides both radial and axial support to the reactor coolant pump primary shaft. Servicing has identified wiped bearings due to a lack of oil lubrication between metal surfaces as a major problem. This phenomenon causes premature wear on the thrust and journal bearing pads and may cause failure of the part. To better understand how this may occur, the team modeled the oil flow throughout the thrust bearing assembly. The students approached the problem by segmenting the highly complex thrust bearing assembly into several subsection problems for analysis. CAD models for each of these subsections were simplified for generation of and computational meshes were generated using GAMBIT. Some of geometries were axisymmetric, some 3-D and some had a combination of stationary and moving walls. A sample of the geometrical complexity is seen in Fig. 3.

![Figure 3: Geometry of middle tank subsection](image)

![Figure 4: Velocity contours in galley way](image)

![Figure 5: Velocity contours in top section](image)

The subsection analyses began with the impeller, which forces the oil from the main housing into the interior through four galley ways. The galley way was chosen first to model, since it’s a simple tube and the mass flow rate was known. As the oil flows down and right towards the pressure outlet, see Fig. 4, the area through which the oil can flow is reduced due to an effective wall that is caused by flow separation in the 90° turn. Another subsection analysis was of the top cover of the thrust bearing assembly. It can be noted that the highest velocity oil in this sections is along the shaft. This can best be seen in the velocity contours in conjunction with the sweep surface shown in Fig. 5.
3.2 Navigating the Number of Code Options

Using the CFD training tutorials have helped student teams to perform flow simulations for a variety of flow problems. The students however have had problems identifying options to use in three areas; (1) which turbulence model and wall treatment to use, (2) which far-field boundary conditions to use in open domains with subsonic or supersonic flows, and (3) what changes in differencing scheme or under-relaxation parameter should be used to stabilize a difficult calculation. Selecting the appropriate option is frequently problem dependent and causes difficulty even for experienced software users.

The students also had difficulty in determining how Fluent post-processes quantities. In the above discussed parachute study, the students needed to determine what reference area was used in the code calculation of the drag coefficient. In practice, the frontal area is used either for forebody or parachute calculations. When the students performed axisymmetric, ¼ revolution or full 3-D calculations varying mesh density, they had difficulty ascertaining what reference area the code was using.

3.3 Interpreting the Results

Once a converged CFD calculation has been successfully obtained, the results are typically displayed using available post-processing tools and either line or contour plots of basic flow properties can be generated. For an engineering design problem, such plots can help one to understand governing flow mechanisms. One student design team was tasked by Hamilton Sundstrand with supporting the design of the cabin air ventilation system for the lunar-bound Orion Crew Exploration Vehicle (CEV). An important role of the air ventilation system is to replicate free convection in a gravity environment of space.

The team considered different possible vent designs and locations, analyzed them in FLUENT. The flow was determined to be laminar and a three-dimensional tetrahedral mesh of 800,000 points was used for all simulations. The key metric for examining the simulations was that a velocity of 15 to 200 fpm must be maintained everywhere in the cabin. Initially, the team studied design performance through the use of velocity contours. Slices where taken of the velocity field were plotted for several different vertical locations throughout the cabin. Thresholding of velocities under the required specification of 15 fpm was employed, and these regions were not plotted but were represented as areas of white. Additionally, regions above the specification of 200 fpm are indicated by the regions plotted as red. All designs had regions where the velocity fell below the critical 15 fpm. Two horizontal slices for one particular design are shown in Fig. 8. The 3.5% slice (off of the CEV floor) clearly shows the injection of fresh air every 60 degrees, while the 43% slice shows what appears to be a stagnant ring region in the middle of the cabin. It was however not intuitively obvious how this particular vent orientation / location produced the observed stagnant flow pattern.

![Figure 6: Velocity contours at 3.5% and 43% off the CEV floor for Vent 1 design](image)

In order to determine the air flow patterns inside of the cabin that a vent configuration produced, velocity vectors were superimposed on top of the velocity contours for a vertical plane through the CEV. Results were obtained for two designs; Vent 1, where 6 vents were evenly distributed circumferentially near the CEV base, and Vent 2, where 4 vents were evenly distributed circumferentially near the CEV base and 2 vents were distributed out of phase more than halfway from the CEV base, see Figs. 7. This analysis gives a different view of what is happening in the cabin and makes it easier to determine the air flow patterns of each model. Additionally, it can be simply seen what is changing from one model to the next. As seen previously, the areas of red correspond to velocities over the 200 fpm specification. From Figure 7a, it was seen that a large recirculation vortex was produced inside the cabin. The core of the vortex corresponds to the regions seen in Fig. 6, where the velocities fell below 15 fpm. Additionally, this vortex are a cause of concern for the integrity of related life support systems, as the air in these locations will have a greater residence time and correspondingly high levels of CO2. Contrastingly, from Fig. 7b, the large recirculation vortex seen with Vent 1 in Fig. 7a was not produced. The lower vents were able to penetrate further into the cabin in Vent 2 than in Vent 1. From this method of analysis, it was predicted that Vent 2 would have almost no low-velocity region in the cabin and furthermore it would produce lower residence times for the air inside of the cabin. Therefore, Vent 2 would appear to be a better design than Vent 1.
The team also explored other ways to view the venting process through a two-dimensional study of a single vent in a closed rectangular domain. Initially, the team displayed only velocity contours, see Fig. 8. From this plot, one sees the jet injection, but as it mixed, the cabin and jet air are undistinguishable. The students then considered particle tracking and mass diffusion displays. In mass diffusion, the injected / new air is considered one species while the cabin / old air is another. One was clearly able to identify the penetration and evacuation of the new air. Considering an inverse view, the students used particle tracking, see Fig. 8, where a single particle was injected into a given location of the flow being studied and its path tracked throughout the flow over the course of time. The color variation displays the transient time history. The tracking of particles, such as carbon dioxide, is important to select a final design configuration. It verifies that the ventilation system is properly circulating the air and not allowing for the pockets of carbon dioxide buildup with the cabin. Such a buildup could be detrimental to the astronauts.

The real challenge however is to avoid using the solution to generate only “Colorized Fluid Dynamic” images. The students were pressed to demonstrate their solutions made sense relative to classical flow theory and also the solution could be used generate quantitative performance output displays. Consider the example of a student team tasked by ASML to design a more sensitive air gage.

ASML develops photolithography machines to etch a prescribed pattern onto a blank silicon wafer computer chip. Air gauges are used to accurately measures distances on the order of 100 microns by sensing the backpressure exerted when a jet of air is streamed through a nozzle perpendicular to a specimen’s surface. The students performed CFD studies to evaluate design concepts and to define and calculate sensitivity from the CFD determined primary flow variables to geometrical modifications that would lead to a more sensitive air gage. The nozzle exit diameter was 1.61mm. The problem they faced however was how to evaluate sensitivity.
students developed reduced order models from the Navier-Stokes equations by assuming that either the distance between the air gage and the wafer surface was small or large. If small, viscous effects were assumed large and a Stokes flow equation was developed. If large, then viscous effects were assumed small and a Bernoulli-based equation was developed. Figure 10 shows a comparison of these reduced order flow models to a CFD solution obtained for a Reynolds number of 200. The distance was kept at 120 microns, while the velocity or flow rate was changed to vary the Reynolds number. When flow rate was reduced producing an effective flow Reynolds number of 50, the CFD solution closely matched the Stokes flow prediction.

After observing that the pressure the stagnation point on the wafer would increase as the nozzle exit to wafer distance was decreased, the students theorized that improved nozzle sensitivity could be correlated to an increased pressure slope with standoff distance. The students then varied geometric features, and obtained an improved air gage design. The students then developed a test rig and performed experiments that validated the new nozzle design’s improved performance.

4 Conclusion

Senior level capstone design courses that promote immersion into the real world of engineering through industrially sponsored projects offer many immediate benefits to near-graduating seniors, but introduces many unique problems to the academic community. Real-world problems differ significantly from academic courses included in the curriculum of most engineering institutions. Furthermore, students typically have little exposure to advanced simulation software tools such as CFD because of their steep learning curve.

The department developed a training approach combining a concentrated series of classroom lectures with immersion into CFD software use. The student effectively utilized available web-based sources to develop a primitive CFD expertise by solving simple unit problems that were part of their classical fluids classroom studies. When the students ventured into real-world problems, they needed to specify the number of grid points, the grid spacing and grid distribution for each zonal block and for each boundary. At this point, the students needed the support of experienced users, either from local industry or from software houses such as Fluent.

The problem of modeling and interpretation was more easily handled by the faculty. Their expertise in fluid dynamics and interpretation of data, either computational or experimental was more than adequate to advise and direct the student teams. Specific code option issues, that were particular to the software used, still required assistance from the software developers.

References: