AC 2007-2323: STRATEGIES FOR THE INTEGRATION OF COMPUTER-BASED SIMULATION TECHNOLOGY INTO THE ENGINEERING CURRICULUM

Rajesh Bhaskaran, Cornell University

Rajesh Bhaskaran is the Director of Swanson Engineering Simulation Program in the Sibley School of Mechanical and Aerospace Engineering at Cornell University. He is leading efforts in the Sibley School to integrate contemporary simulation technologies into the mechanical and aerospace engineering curriculum. As part of this initiative, he is developing web-based curriculum material for integrating the ANSYS and FLUENT packages into solid and fluid mechanics courses. He holds a Ph.D. in Aerospace Engineering from Iowa State University.

Strategies for the Integration of Computer-Based Simulation Technology into the Engineering Curriculum

Abstract

While Computer-Aided Engineering (CAE) technology has revolutionized engineering analysis, design and research, its penetration into the undergraduate mechanical engineering curriculum has been limited. As a result, undergraduate students do not acquire a solid foundation in CAE technology that they can build upon during the course of their careers. Our pedagogical approach for integrating CAE software into courses has three key elements. First, the CAE experience revolves around a series of case studies in which students use CAE software to simulate canonical problems with known analytical solutions, approximations or experimental data. These case studies are a platform both to provide a guided introduction to the appropriate use of CAE technology as well as to reinforce basic theoretical and physical concepts traditionally covered in lectures. Second, web-based tutorials are used to teach students the mechanics of using the software interface. This ensures that instructor time is devoted to explaining underlying concepts rather than to teaching the intricacies of the software interface. Thus, our emphasis is on imparting concepts rather than raw software skills. Third, the necessary numerical concepts are introduced "just-in-time" in a focused manner in order to meet tight time constraints. Verification and validation of results are emphasized throughout. The above approach has been implemented in a fluid dynamics course using the FLUENT package and in two solid mechanics courses using the ANSYS package. "Teaching modules" based on selected case-studies are being developed for the FLUENT and ANSYS packages. Each teaching module consists of three components: (i) a web-based tutorial that takes students through the steps involved in solving the case study problem (with nominal parameters) using the relevant CAE package; (ii) notes describing related theoretical and numerical concepts that can be handed out to students; (iii) problem set with solutions.

1 Introduction

Within the last fifteen years, computer-based simulation has become an integral part of design, analysis and research in engineering. The increasingly widespread use of Computer Aided Engineering (CAE) has been driven by the dramatic reduction in the cost of computing hardware and the maturing of off-the-shelf, commercial software packages. Commercial CAE packages such as ANSYS, ALGOR, FLUENT, Pro/Engineer, and STAR-CD are now routinely used to simulate engineering systems in a wide range of industries. Despite the pervasive use of CAE technology in industry and research, its use in the undergraduate curriculum has been limited. As a result, undergraduate students do not acquire a solid foundation in CAE technology that they can build upon during the course of their careers. Furthermore, the potential for CAE tools to enhance the learning experience is not realized. For instance, these tools can be used as virtual-lab environments for hands-on, visual learning. They also enable the instructor to make strong connections between theory and practice.

The conundrum facing instructors is how to balance traditional theory and numerical methods with the use of sophisticated CAE software. When computer-based methods are taught at the undergraduate level, the focus tends to be on numerical theory with topics such as discretization schemes, element formulation and inversion algorithms discussed in detail. Students in these courses typically develop computer codes to solve a few simple, mostly linear problems. For instance, this approach is used in the popular introductory CFD textbooks by Anderson[1] and Tannehill et al.[2]. It prepares students well for developing their own computer codes. While this is quite appropriate for Ph.D. level students who are likely to be developing their own computer applications, the vast majority of undergraduate students, on becoming practicing engineers, are going to be using off-the-shelf software. They need to know how to assess and validate results from the software, what elements to choose for finite-element analysis (FEA), what approaches to use to overcome convergence problems, how to generate meshes that are appropriate for the problem, what the limitations of popular turbulence models are, etc. They have less of a need to know the details of the various discretization schemes and inversion algorithms, just as a chef doesn't need to know how to grow her favorite vegetable to whip up a great dish with it.

There are two aspects to learning to use a commercial CAE package. First, students need to develop the necessary *skills* to use the software interface to set up and solve a variety of engineering problems. Second, they need to understand the underlying *concepts* in order to apply the software correctly so as to obtain validated results. The training offered by software companies usually stresses the learning of software skills with the underlying concepts touched upon lightly, if at all. A user skilled in the intricacies of the software who has a poor appreciation of the associated concepts is likely to generate grossly incorrect results. This is especially so since there are many different sources of error in a CAE solution: insufficient grid resolution, incorrect boundary conditions, numerical instabilities, inappropriate application of turbulence models, and so on. Our emphasis in this effort is to impart to students underlying *concepts* rather than software *skills*. We seek to teach students how to use general-purpose FEA and CFD software to generate validated results to a range of problems rather than the intricacies of using the software interface. Students will be able to apply these concepts to simulate engineering systems irrespective of the specific software they are using.

We have developed a pedagogical approach that addresses the challenge of balancing traditional theory and numerical methods with the use of sophisticated CAE software. This approach has three key elements. First, the CAE experience revolves around a series of case studies in applying CAE software to solve canonical problems with analytical solutions, approximations or experimental data. This enables a modular approach where individual modules can be "plugged" into courses as deemed appropriate by the instructor. Second, web-based tutorials are used to teach students the mechanics of using the software interface. This ensures that instructor time is devoted to explaining underlying concepts rather than to teaching the intricacies of the interface. Third, the necessary numerical concepts are introduced "just-in-time" in a very focused manner in order to meet tight time constraints. Verification and validation of results are emphasized throughout.

2 Case Study Approach

In the case studies, students use CAE software to solve canonical problems with analytical solutions, approximations or experimental data. This is analogous to validation undertaken during code development where results from the program are continually benchmarked against standard problems in the literature. The use of case studies enables a modular approach: Individual case studies can be incorporated into courses without having to retool the course from the ground up. There is a strong connection between the simulation procedure/results and fundamental physical and numerical principles. We take advantage of this to make strong connections between theory and simulation as shown in the sample case study below. Thus, we are able to reinforce fundamental concepts through the hands-on, visual environment provided by the CAE software while teaching the appropriate use of the software.

Sample Case Study: Compressible Flow in a Nozzle

To illustrate the case study approach, consider the high-speed flow through an axisymmetric converging-diverging nozzle. A preliminary implementation of this case study has been carried out in *MAE 423/523 Intermediate Fluid Dynamics,* an elective second course in fluid dynamics. The software used is FLUENT, a leading package for modeling fluid flow in a variety of industrial applications. We first present the classical inviscid, quasi-1D theory in a lecture format. This theory predicts the operating regime based on the ratio of the incoming stagnation pressure to the exit pressure. For the isentropic case, the theory gives the Mach number, pressure, and temperature variation in the flow direction.

Discussion of the simplified theory is followed up with simulations using FLUENT in a computer lab session. While the simulation, as the 1D theory, assumes inviscid flow, the geometry modeled is 2D axisymmetric. We first perform a simulation for a pressure ratio at which the theory predicts isentropic flow. The corresponding static pressure contours obtained from FLUENT are shown in Figure 1. This plot provides students with an idea of the effect of two-dimensionality on the flow. In the 1D analysis, the pressure contours would all be vertical straight lines. The comparison of the Mach number variation between theory and simulation is shown in Figure 2. At each streamwise location x, the Mach number from 1D theory lies between the centerline and wall values obtained from the simulation. Thus, students can see that the theory provides a good estimate of the average Mach number over a cross-section for the isentropic case.

The isentropic simulation is followed up with a non-isentropic calculation at a much lower pressure ratio. We first ask students to predict the operating regime using theory. Subsequently, the FLUENT calculation for this pressure ratio is performed. The Mach number contours from FLUENT are shown in Figure 3. There is a shock in the diverging section which agrees with what is expected from the theory. Thus, the contour plots in FLUENT provide students with a visual representation of the operating regimes discussed in the theory, reinforcing this topic. The steep gradient across the shock is used to motivate the concept of grid adaption. The adapted grid with increased resolution in the vicinity of the shock is shown in Figure 4.

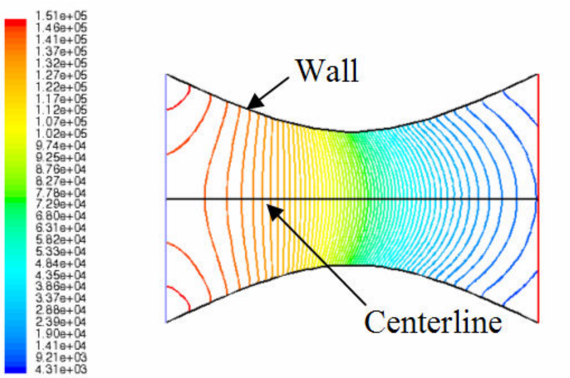

Figure 1: Static pressure contours for non-isentropic flow in a nozzle.

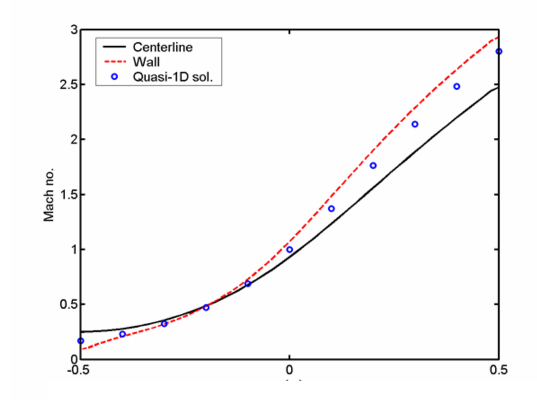

Figure 2: Mach number variation along the nozzle.

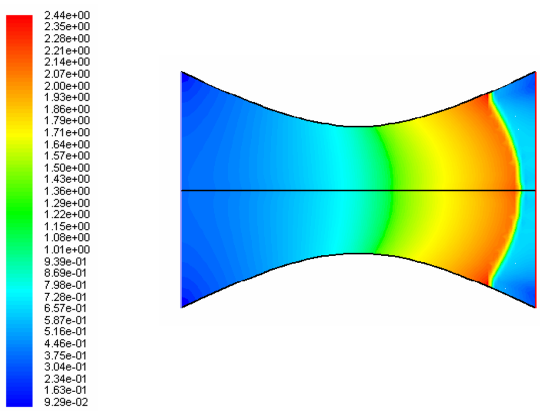

Figure 3: Mach number contours for non-isentropic flow in a nozzle.

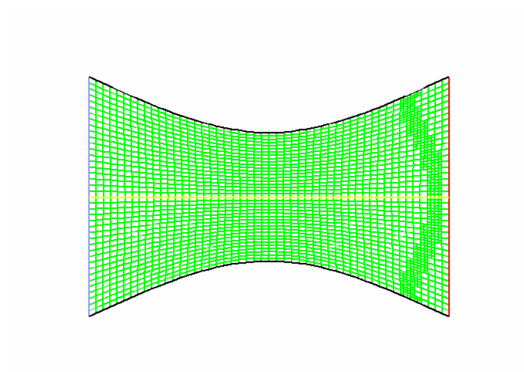

Figure 4: Adapted grid for the non-isentropic case.

In the theory as well as simulations, it was assumed that the flow is inviscid. Students can investigate the validity of this assumption using FLUENT by solving the viscous Navier-Stokes equations rather than the inviscid Euler equations. Momentum conservation for the viscous case is

$$\rho\,(\boldsymbol{V} \bullet \nabla)\,\boldsymbol{V} = -\nabla p + \nabla \bullet \overline{\overline{\tau}}$$

Here ρ is the density, $\boldsymbol{V}$ velocity and p pressure The last term represents the viscous stresses and is set to zero in the inviscid simulations above. In order to investigate viscous effects, this term can be turned on using the menu shown in Figure 5. Since this is a high Reynolds number flow, the flow is turbulent and a turbulence model such as k-epsilon is selected to model the viscous stresses. One can use the detailed field information provided by CFD to demonstrate that the viscous boundary layer near the wall is very thin and attached. Thus, students see that viscous effects are confined to a very thin region near the wall for a high Reynolds number flow with a favorable pressure gradient; the inviscid assumption provides accurate results in these cases. In this fashion, FLUENT is used as a virtual lab environment to investigate the effects of viscosity.

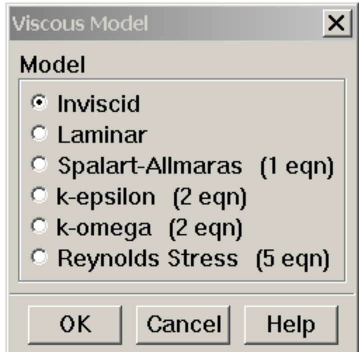

Figure 5: Menu to turn on/off viscous effects.

This sample case study shows how the use of CFD software can be used to reinforce fundamental concepts through a visual, hands-on learning process. It also shows how the software can be used to investigate the effect of specific terms in the governing equations. In the process, students learn the appropriate use of a powerful analysis tool.

3 Web-Based Tutorials

In order to perform the computational case studies, students need to be taught the mechanics of using the CAE software interface. This is most effectively and efficiently done through web-based tutorials. We have developed tutorials to introduce students to FEA technology through ANSYS and CFD technology through FLUENT. ANSYS is a popular general purpose finite element modeling package for numerically solving a wide variety of mechanical problems. We use it for structural mechanics applications; the topics covered in the ANSYS tutorials include truss, plate, curved beam and semi-monocoque shell[3]. The FLUENT tutorials cover pipe, nozzle and airfoil flows[4]. These tutorials are designed to be used in the following mode:

- The user fires up the web browser and CAE software interface side-by-side.
- She reads instructions from the browser and implements them in the CAE software.

A screenshot from the curved beam tutorial in Figure 6 shows the arrangement of the ANSYS and browser windows.

The advantages of using the web to augment instruction are multi-fold:

1. It reduces face-to-face time required for teaching the mechanics of using the software GUI. Instructors don't need to devote scarce classroom time to teach students how to use the software interface to solve problems of interest in the course. Classroom time can be used for more value-added activities such as reinforcing underlying physical and numerical concepts, clearing up misconceptions, etc.

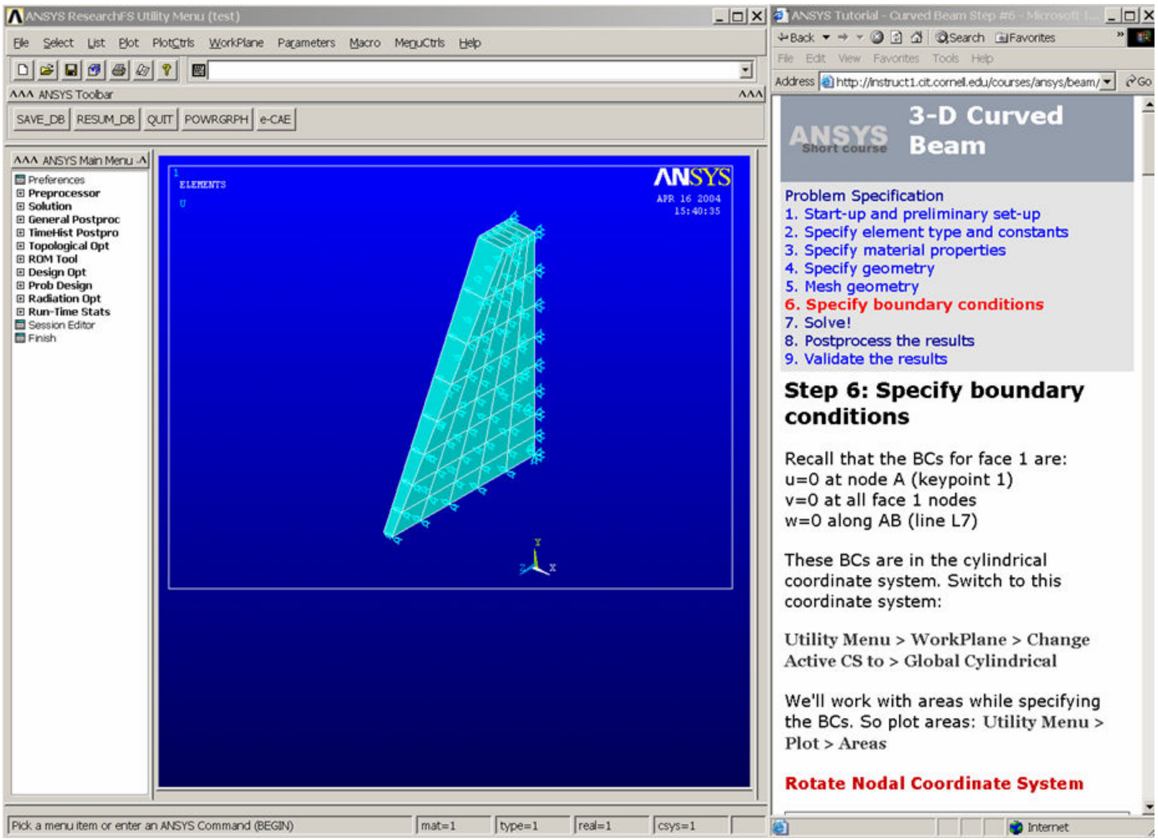

Figure 6: Arrangement of ANSYS and browser windows for the curved beam tutorial.

2. The GUI is learned more effectively through self-paced, hands-on use which allows the students to develop a feel for the interface outside of class.
3. In an era where the search for information often starts with an internet search engine such as Google, the web is a very effective dissemination mechanism which puts a broad audience at one's fingertips. When the search term "CFD tutorials" is entered in Google, our FLUENT tutorials site[4] is listed second among almost 100,000 entries. The high Google rank demonstrates that there are many users in the larger CFD community who are seeking out this material on the internet.
4. The tutorials facilitate asynchronous learning. For instance, some students might be motivated to learn about CAE technology for use in a project or co-op assignment even though they have not received formal classroom instruction in it. They can use the tutorials to learn the basics with some guidance from a faculty member.

Pedagogical Features of the Tutorials

The tutorials developed by the author focus on teaching important underlying concepts while helping students develop a better physical feel for basic phenomena. (In contrast, the prepackaged tutorials offered by the software vendors are focused towards showcasing the features of the software package.) As students follow the steps in a tutorial and click away with the mouse, they are apt to lose track of the big picture. After completing the tutorial, they are left with the feeling that they have followed a recipe without understanding how it needs to be modified for a different but related application. To alleviate this problem, each tutorial has been broken down into a set of steps. Since the high-level solution procedure is the same for similar applications, the ANSYS and FLUENT tutorials each consist of the same set of steps, for the most part. Figure 7 shows the list of steps for an ANSYS tutorial. By encountering the same set of steps in different tutorials, students develop a close familiarity with it and are better equipped to apply this solution procedure to new problems. The list of steps appears at the top of each page of the tutorial with the current step highlighted in a different color. This enables students to keep track of their progress through the solution process, providing a structure to the learning experience.

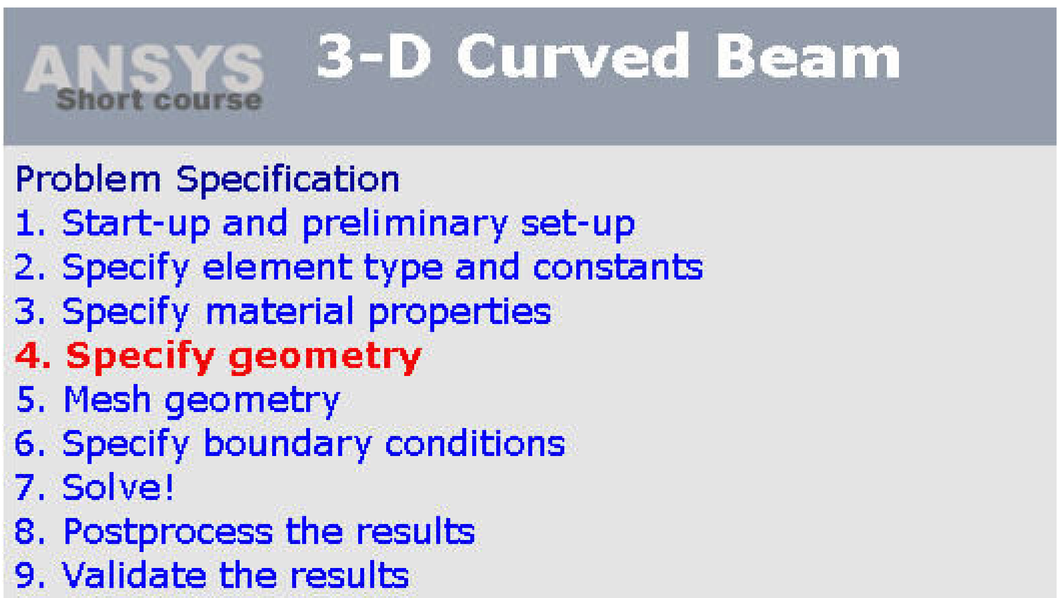

Figure 7: The set of steps in an ANSYS tutorial.

In the tutorials, we provide the rationale for the actions taken in the CAE software rather than just provide a “recipe” of clicks and text entries. The emphasis is on the understanding of the solution procedure, and the analysis and justification of results. The rationale for the actions and related discussion items are highlighted by enclosing them in a box. Students are shown how they can keep checking their work as they build the model and how to take corrective action for common mistakes. Throughout the tutorials, we continually point students to where they can find relevant information in the online documentation. This promotes the spirit of independent learning from the get go and equips students for making the best use of the extensive online documentation included with most CAE packages.

One of the challenges in using CAE technology in the classroom is to teach students to regard the results skeptically and discourage the blind acceptance of results that the computer puts out. This is analogous to the pitfalls of the "formula mentality" where, given a homework problem, students seek a formula in the textbook into which they can plug in the numbers given and get a result. The results could be grossly incorrect in both cases but with CAE technology, the erroneous result would come in multi-dimensional plots with fancy graphics. In order to stress the need for validation, we include a separate validation step in each tutorial as shown in Figure 7. Simple sanity checks are performed first. For example, in structural mechanics problems, we investigate the following questions:

- Does the deformed shape look reasonable and agree with the applied boundary conditions?
- Do the reactions at the supports balance the applied forces for static equilibrium?

Then, we check if the mesh resolution is adequate by comparing results on different meshes. Another important avenue for validating results is making comparisons with theory. For instance, in the plate-with-a-hole tutorial in ANSYS, we check if the stress concentration factor approaches the analytical value for a small hole as the size of the hole is reduced. In the compressible nozzle tutorial in FLUENT, we compare the 2D computational results with the corresponding result from quasi-1D theory as shown in Figure 2.

Leveraging the Tutorials for Experiential Learning

Each tutorial is followed up with in-class sessions to discuss relevant concepts and address student questions. Tweaking the original problem and studying how the solution changes is a very effective way to clarify and reinforce concepts. So we go through hands-on exercises in class which involve modifying the original problem. For instance, the plate-with-a-hole tutorial uses 4-node quad (PLANE42) elements. Students are asked to re-solve the problem using 8-node quad elements (PLANE82) starting from their ANSYS solution for the 4-node quad elements. (They bring their tutorial results to the computer classroom on a USB memory stick.) They are first asked to think about which of the nine tutorial steps they would have to change. This motivates them to think about what is being done in each step. An important issue that is addressed is whether the boundary condition specification step (step 6 in Figure 7) needs to be redone. This is an opportune moment to discuss the difference between applying the loads to the geometry or to the mesh. Since we do the former in the tutorial, there is no need to reapply the loads in this case. This is confirmed by plotting the load symbols in the graphics window and by listing the loads, after the model is remeshed. In the process, students also learn about the ways in which they can test whether they have set up the problem correctly in ANSYS. We compare the element solution for the von Mises stress for the two cases to show that the higher-order element gives a smoother solution and discuss why this is so. This also indicates whether the mesh resolution was adequate in the 4-node quad solution.

The above example illustrates how the web-based tutorials can be leveraged to bring experiential learning into the classroom. Wallace & Weiner[5] provide evidence that using experiential learning exercises in the classroom can be more effective pedagogically than a traditional lecture-based presentation of the material. Hands-on exercises provide extra motivation for students to participate in learning in the classroom. These sessions can also be used to demonstrate how easy it is to get the wrong answer, driving home the importance of validation. The combination of CAE and web technologies enables a new and more effective way of teaching.

4 Strategy for Introducing Numerical Concepts

To perform the case studies, students need to understand basic numerical concepts such as discretization, grid refinement, iterative convergence, first and second-order schemes, the need for turbulence models, etc. While these issues are discussed in detail in numerical methods courses, there is not enough time to do so in courses of a more general scope. In our *Intermediate Fluid Dynamics* course, the necessary background is provided through a focused introduction to the numerical solution procedure before students perform the computational case studies. In order to keep the discussion focused and accessible, we adopt the following pedagogical strategy:

1. We illustrate each step in CFD solution process on a simple 1D model equation on a small grid as shown in Figure 8. For this grid, it is easy to write down the discretized system and invert it manually, making the solution process transparent to the audience.
2. In each step, we relate model problem concepts to the general CFD solution process. These model problem concepts are later invoked as necessary in the context of the case studies.

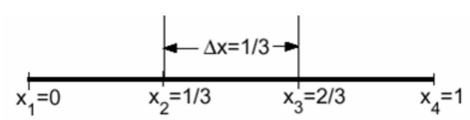

Figure 8: Grid used to illustrate the CFD solution process.

As homework, students implement numerical solutions for linear and non-linear 1D model problems in MATLAB and in the process, explore fundamental numerical issues. Our experience suggests that such an approach makes fundamental concepts more concrete in students' minds than a general verbal and graphical description. Due to time constraints, numerical concepts discussed in the introduction are the minimum necessary to perform and understand the case studies. Topics are revisited later in greater detail as time permits. This approach is similar to the "just-in-time" teaching philosophy in a project-centered course (Schmidt & Beaman, 2003). The idea is to teach a particular concept as students encounter it while performing a project. This provides context and also additional motivation to learn the concept since it is necessary to complete the project.

5 Evaluation Results

A student survey was conducted to evaluate the effectiveness of the ANSYS tutorials in teaching the basics of FEA technology and to get feedback for improving them. Ninety-six survey responses were received. Figure 9 summarizes the survey results on the navigational features and formatting of the tutorials. From the figure, it can be seen that a

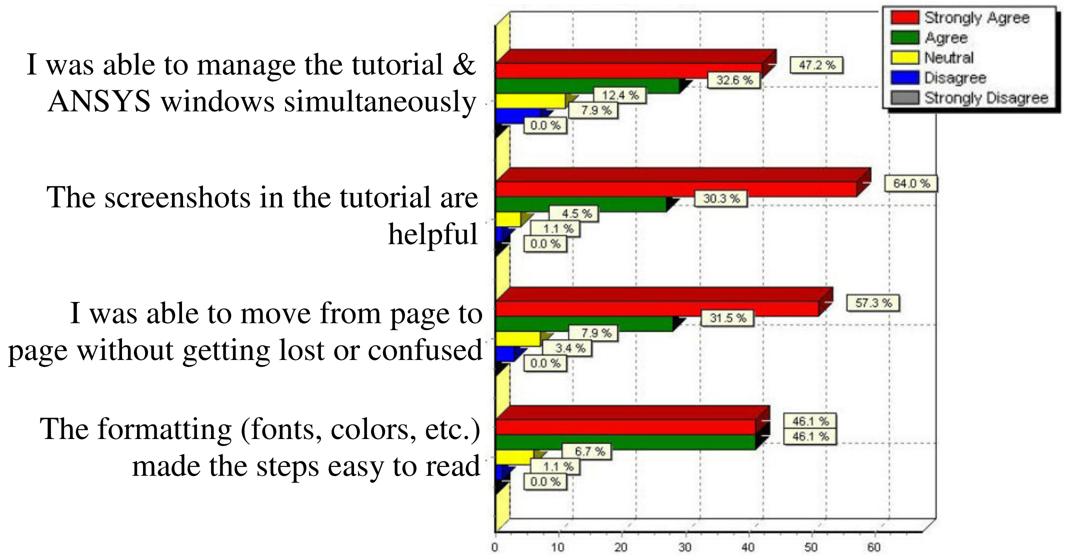

Figure 9: Evaluation results on the navigational features and functionality of the tutorials.

large majority of the students picked *Strongly Agree* or *Agree* to the questions in this section of the survey. From this, it is safe to conclude that the tutorials are user-friendly. Responses to questions on the pedagogical effectiveness of the tutorials are summarized in Figure 10. Most the responses fall in the *Strongly Agree* or *Agree* categories. Only a small minority were in the *Disagree* or *Strongly Disagree* categories, though a larger proportion than on the navigational and formatting features in Figure 9. This is not unexpected since getting the students to understand the rationale behind the steps or preparing them adequately for doing similar problems is more challenging that making the tutorials user-friendly.

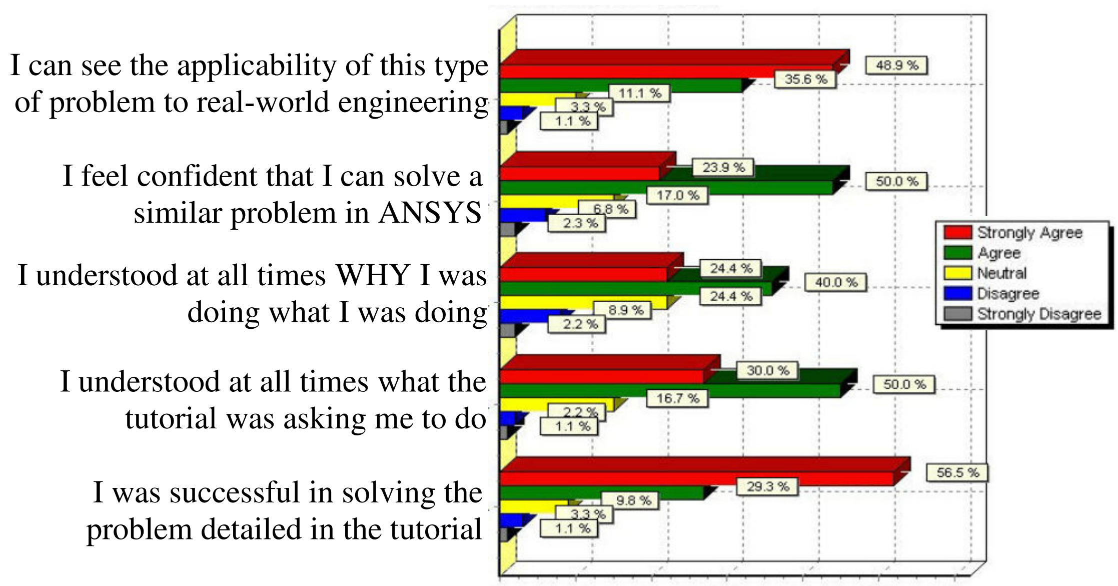

Figure 10: Evaluation results on the pedagogical effectiveness of the tutorials.

6 Conclusion

We have developed a pedagogical approach for integrating modern CAE tools into the undergraduate engineering curriculum. This approach helps the educator balance traditional theory and numerical methods with the use of sophisticated CAE software. The emphasis of our approach is on the understanding of the solution procedure as well as the analysis and validation of results. A key element of our approach is that the CAE experience is built around case studies of selected, usually canonical, problems. The relevant theory for these problems is already covered in the traditional curriculum. This enables us to use the case studies to make strong connections between theory and simulation, with the student gaining insight into how the two methods complement each other. By using the CAE software as a virtual lab environment within which students learn about physical processes through hands-on, visual exploration, the educational experience is enhanced and enriched. The use of case studies enables a modular approach where individual case studies can be integrated into existing courses without having to revamp these courses.

Another key element of our pedagogical approach involves the use of web-based tutorials to teach the mechanics of using the software interface i.e. "skills". This enables students to learn the software interface through self-paced, hands-on use. Classroom time can then be devoted to improving the understanding of "concepts" such as sources of error, validation of the simulation results, limits of the technology, etc. The use of these case studies in general undergraduate courses is facilitated by the "just-in-time" introduction of numerical concepts, a third important element of our approach. Numerical topics covered are the minimum necessary to understand the case studies. The "just-in-time" strategy is aided by illustrating the numerical solution procedure on a simple, if possible 1D, model problem and making extensions to the general case from the model problem. By anchoring this necessarily shortened discussion of numerics within the context of a model problem, important numerical concepts are made more concrete in students' minds. The objective is not to turn out experts in numerical methods but to graduate engineers with a good basic understanding of CAE technology which they can build upon as practicing engineers. The hands-on, visual medium of the CAE software simultaneously becomes a platform through which the educator can make abstract concepts more concrete.

Our pedagogical approach has been implemented for two selected software packages, ANSYS for structural mechanics and FLUENT for fluid mechanics. It can be easily extended to other packages and CAE areas. In order to help instructors readily deploy relevant case studies in their courses, we are in the process of packaging the case studies into "teaching modules". Each teaching module consists of three components: (i) a web-based tutorial that takes students through the steps involved in solving the case study problem (with nominal parameters) using the relevant CAE package; (ii) notes describing related theoretical and numerical concepts that can be handed out to students; (iii) problem set with solutions. Since these modules are designed to be integrated into existing courses, they will help make the CAE experience part-and-parcel of the overall coursework, rather than it being ghettoized in specialized courses. Keeping in mind that

CAE technology has revolutionized engineering practice and research, a gleam steals into one's eyes: an educational revolution, one module at a time.

Bibilography

1. J. D. Anderson, *Computational Fluid Dynamics: The Basics with Applications* (Mc-Graw Hill, New York, 1995).
2. J. C. Tannehill, D.A. Anderson and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer* (Taylor & Francis, Philadelphia, 1997).
3. R. Bhaskaran, ``ANSYS Short Course", *http://instruct1.cit.cornell.edu/courses/ansys*. Chapters: Two-dimensional static truss; Plate with a hole; Three-dimensional curved beam; Semi-monocoque shell.
4. R. Bhaskaran, ``FLUENT Short Course" *http://instruct1.cit.cornell.edu/courses/fluent*. Chapters: Laminar pipe flow; Turbulent pipe flow;Compressible nozzle flow; Airfoil flow.
5. D. R. Wallace and S. T. Weiner, "How might classroom time be used given WWW-based lectures", *Journal of Engineering Education*, 1998, p. 237-248.