43rd AIAA Aerospace Sciences Meeting and Exhibit
10 - 13 January 2005, Reno, Nevada

AIAA 2005-1072

Computational Aerodynamics Goes To School: A Course in CFD for Undergraduate Students

Russell M. Cummings[*] and Scott A. Morton[†]
United States Air Force Academy, USAF Academy, CO, 80840

As aerodynamics education has evolved over the past decades, a slow transition from important analytic methods to increasingly powerful computational methods has taken place. While a basic understanding of theoretical aerodynamics should always be included in coursework, the realities of modern design practices make the usefulness of the traditional approach less and less practical. A new undergraduate course in computational aerodynamics has been developed that attempts to give students experience with the modern computational tools of aerodynamics, primarily from an applications perspective. While introducing students to the important computational topics of accuracy and stability, the course stresses the practical tools that computational aerodynamics requires: importance of understanding the physical problem, developing a good grid, checking results for convergence and accuracy, and computing unsteady, turbulent flows. A number of "lessons learned" have resulted from teaching the course, including the importance of providing appropriate background instruction in computer systems and command languages, providing tutorials for obtaining skill in grid generation and flow visualization, and giving students individual attention for learning the "gray" skills of computational aerodynamics. Provided appropriate attention and support is available, we believe that undergraduate students can be taught computational aerodynamics at a level that will make them intelligent users of modern computational tools.

I. Introduction

THE teaching of computational fluid dynamics (CFD) has long been known to be an important, although advanced, discipline. Steger and Hafez wrote in 1992 that, "On the graduate level CFD is often treated as a separate course. On the undergraduate level it may instead be taught as part of a fluids or a numerical analysis course."[1] In other words, undergraduates can perform CFD projects as part of a fluids course (usually fairly "canned" projects), but they probably will not be able to learn enough about CFD to perform meaningful predictions and become intelligent users of CFD. This belief has probably colored the perception of most aerodynamic professors and forced them to avoid doing too much CFD in their undergraduate courses.

The difficulty has always been that CFD requires some expertise of at least four knowledge areas: flow physics (fluid dynamics and aerodynamics), numerical methods, computers (codes and systems), and validation (experimental and theoretical). These requirements have necessarily seen CFD being taught primarily at the graduate level, often in a sequence of courses designed to concentrate on the first two knowledge areas, usually emphasizing numerical methods. In fact, some people say, with regret, that CFD is usually spelled "Cfd" (with the main emphasis on the numerical methods and computers) when it should be spelled "cFD" (with the main emphasis on the fluid dynamics), which calls attention to the over-emphasis within the CFD community on algorithms, and the lack of emphasis on the resulting aerodynamics.[2] The combination of high-level knowledge and the high-level computers that were often required for practical flow solutions has limited practical CFD education. As time went on, of course, Moore's Law helped take care of the computer capability issue, but few people have realized the paradigm shift that has taken place with PC clusters—many CFD calculations are now affordable and available to students, which means CFD has come of age, whether we like it or not.

Most universities still hold to the observation from 1992, "While universities are teaching the elements of CFD as a discipline, they are also beginning to use CFD solutions and post-processing software to teach fluid dynamics and aerodynamics."[1] While it is absolutely true that CFD makes a good teaching demonstrator (visualizing vortices,

---

[*] Professor, Department of Aeronautics, AIAA Associate Fellow.
[†] Professor, Department of Aeronautics, AIAA Associate Fellow.

showing the flow over an airfoil, etc.), we believe that it is time for CFD to make its way, fully fledged, into the undergraduate curriculum. Certainly, other than performing basic simulations of one-dimensional problems, few undergraduates have been able to participate in the full depth of CFD, especially for turbulent, three-dimensional flows. In addition to this, many educators have realized that undergraduate aerodynamics education must be reformed, with one of the major changes taking place in CFD. While many of us enjoy teaching the classics of aerodynamic theory (such as conformal transformations of flow over a circular cylinder into airfoil flow), the reality has become that few, if any, of our students will every use those concepts in their careers, either in graduate school or industry. We strongly endorse the perspective of David Darmofal and Earll Murman of MIT:

> *"Within aerodynamics, the need for re-engineering the traditional curriculum is critical. Industry, government, and (to some extent) academia has seen a significant shift away from engineering science and highly specialized research-oriented personnel toward product development and systems-thinking personnel. While technical expertise in aerodynamics is required, it plays a less critical role in the design of aircraft than in previous generations. In addition to these influences, aerodynamics has been revolutionized by the development and maturation of computational methods. These factors cast significant doubt that a traditional aerodynamics curriculum with its largely theoretical approach remains the most effective education for the next generation of aerospace engineers. We believe that change is in order."*[3]

We agree completely, and believe that CFD needs to be brought into the undergraduate classroom as soon as possible. The real problem remains the various logistical challenges associated with teaching CFD. In addition, faculty will have to develop the pedagogical processes for learning CFD effectively, but how should that be done?

"The only way to learn CFD is to do it." Those are the sentiments of Prof. Robert MacCormack of Stanford University, spoken during a short course in CFD that he taught several years ago. While his statement may sound overly simple, the reality of the matter is that his observation is very accurate. CFD, like so much of engineering, requires more than book knowledge—real learning comes from coding, applying, comparing, and improving. While there is little need these days for legions of graduate students writing individual codes, the appropriate application of CFD to aerodynamic problems (essentially serving as a numerical wind tunnel) can lead to great advances. It is at this point where education has dropped the ball. There are two basic types of CFD education currently available to students—all or (nearly) nothing—there must be an in-between position. But what will the students of the future need to know about CFD?[4-7] Certainly, we do not want to set loose a generation of CFD amateurs who will prove the wisdom of the phrase, "garbage in, garbage out," but we also don't need to have all students learn the intricacies of hundreds of algorithms. What is the appropriate common ground for CFD in education? Perhaps the answer to the question comes from answering another question, "what do students need to know about CFD for their careers?" It was an attempt to answer this question that led us to create an undergraduate course in CFD, or as we call it, Computational Aerodynamics.

II. Fluids/Aero Courses at the U.S. Air Force Academy

Based on the perception that students would gain more from a computational aerodynamic curriculum than the traditional theoretical curriculum, a reorganization of the fluid/aero courses at the Academy was undertaken. In prior years, students took a sequence of four courses that impacted their knowledge in fluids/aero, with an additional "core" course taken by all cadets in Fundamentals of Aeronautics (AERO 315). With the background knowledge from AERO 315, the students were taught about basic thermodynamics and energy systems (including jet engines) in AERO 310 (another "core" course taken by all cadets). Students who major in aeronautics then specialized within the fluids/aero discipline by taken an additional three courses: AERO 341 (fluid dynamics), 342 (low speed aerodynamics), and 442 (high speed aerodynamics). Additionally, a number of senior-level technical electives were available to the cadets, including AERO 447, an elective in CFD that had been taught a number of times in recent years. The lower portion of Fig. 1 shows the curriculum that existed previously.

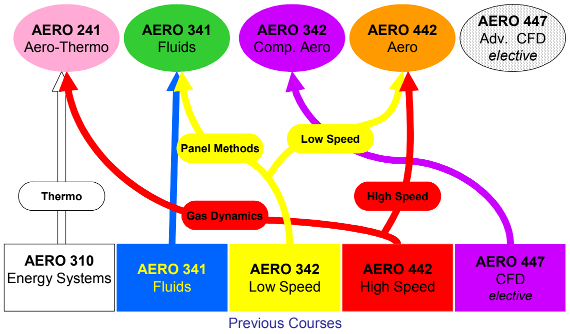


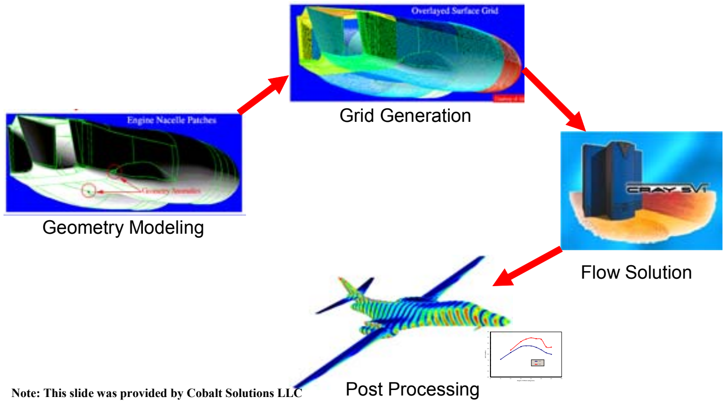
Figure 1. Fluid/Aero Curriculum Changes.

Two years ago the Academy undertook a major reform of the courses within the "core", both to reduce the number of units required to graduate, and to improve the overall educational experience of the students. Of the many changes that took place, AERO 310 (Energy Systems) was dropped from the "core", while AERO 315 (Fundamentals of Aeronautics) was retained. Since all cadets had been required to take AERO 310, the department had to create a course to replace the thermodynamics content that was being lost, so AERO 241 (Aero-Thermodynamics) was instituted. This allowed for a great deal of rethinking and reorganizing of the fluids/aero curriculum, including the possibility of adding a computational aerodynamics component to the course sequence.

In order to accommodate the large amount of material that would be required to teach a course in computational aerodynamics, the following substantial rearrangement of course topics was undertaken (see Fig. 1, where the upper courses refer to the revised curriculum): move gas dynamics component from AERO 442 to the new AERO 241 course (thus combining thermodynamics and gas dynamics into a single course); move potential flow and basic panel methods concepts from the former AERO 342 to AERO 341 (thus leaving room in AERO 342 for more advanced panel methods, such as vortex lattice methods); move low speed aerodynamics from AERO 342 to AERO 442 (making 442 a course in advanced aerodynamics); move CFD topics previously taught in AERO 447 into the new AERO 342. While this represented a great deal of logistical difficulties (especially as the new and old courses were being phased in/phased out simultaneously, the transition is now complete and all new courses have been taught as of Fall 2004. The rearrangement of material, and the addition of so much new material in AERO 342, has led to a modern view of aerodynamics for our students, combining theoretical, experimental, and now computational approaches.

III. The Computational Aerodynamics Course

While we have only taught Computational Aerodynamics a total of three times (first as an elective, then as an experimental course to a select group of students, and finally to the entire junior class), we have learned a great deal about the structure of the course. In addition, a variety of "lessons learned" have become apparent to us. We attempted to pattern the course after the Four Step CFD process (see Fig. 2), where the students would learn each of the steps (to one degree or another): geometry modeling, grid generation, flow solution, and post processing. The topics of the course are presented below, followed by a description of the software used, the projects created for the course, and a listing of the lessons learned.



Note: This slide was provided by Cobalt Solutions LLC

Figure 2. The Four Step CFD Process.

A. Course Topics

As was mentioned previously, CFD requires a broad background in aerodynamics, numerical methods, and computer usage. The course topics were largely presented to the students "just in time" as projects and assignments required a given knowledge set. Course topics include:

- intro to CFD
- computer system overview
- review of vector algebra
- governing equations of fluid motion
- finite differencing/finite volume/finite element approaches
- shock capturing
- truncation error
- stability, consistency, convergence, and CFL number
- classification of PDEs
- algorithm types (explicit, implicit, match to PDE types)
- steady and unsteady flow
- time integration

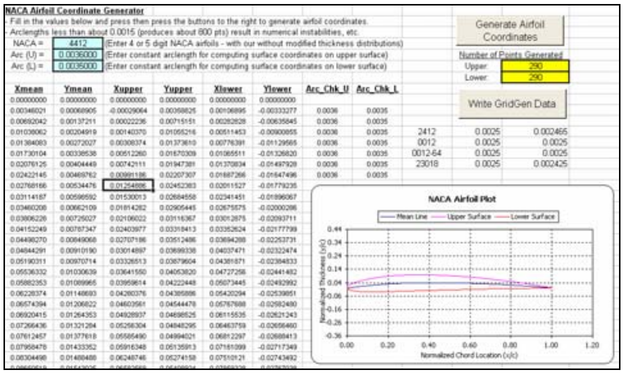
- boundary conditions (farfield, solid surfaces, matching PDE types)
- grid generation (structured, unstructured, Cartesian, overset, topology, transformed coordinates)
- the solution process (initial conditions, convergence, stability/robustness)
- grid independence
- time accuracy
- turbulence models (RANS, LES, DES), model types (eddy viscosity, stress transport)
- potential flow
- panel methods
- vortex lattice methods

While in most cases these topics could only be covered at a basic level, we believe that the goal of the course was actually well served by our inability to spend a great deal of time on each topic. Specifically, instead of trying to teach the students the details of numerical algorithms (for example), we made sure they understood the different types of algorithms, and how various algorithms should be used for various types of flow simulation. A similar example is found in turbulence modeling—we did not teach the students the details of any turbulence model, but attempted to convey the purposes and limitations of various types of models.

B. Software and Computer Resources Used

One of the difficult aspects of teaching computational aerodynamics to anyone is the high learning curve required for the various computer systems and software applications. While we definitely felt the pressure of teaching the students about a great deal of software, the maturing of CFD has helped a great deal. Whereas a few years ago most CFD software was "homegrown" and often cumbersome, complicated, and not robust, a large body of commercial software is now available. A great deal of time and energy was spent evaluating software for CFD research here at the Academy, and we used our experience to help determine which software applications to use in the computational aerodynamics course.

The primary software used for the CFD projects were: GridGen for grid generation, Cobalt as the flow solver, and FieldView for post-processing. In addition, a number of in-house programs and spreadsheets were created for the students to use at different times. For example, a spreadsheet was made that could create geometry inputs for GridGen for any NACA 4- or 5-digit airfoil (see Fig. 3). The program allows for hundreds of equally spaced points to be determined to help make the surface geometry as smooth and accurate as possible. Finally, a project was included on panel methods/vortex lattice methods, and a PC-based program was created that contains the VorLax code in an easy-to-use format. Other freeware/shareware was used as needed for the various computational projects. All of these programs would have been difficult, if not impossible, for a large number of students to use until fairly recently.


NACA Airfoil Coordinate Generator
- Fill in the values below and press then press the buttons to the right to generate airfoil coordinates.
- Arclengths less than about 0.0015 (produces about 800 pts) result in numerical instabilities, etc.
NACA = 4412 (Enter 4 or 5 digit NACA airfoils - with our without modified thickness distributions)
Arc (U) = 0.0036000 (Enter constant arclength for computing surface coordinates on upper surface)
Arc (L) = 0.0035000 (Enter constant arclength for computing surface coordinates on lower surface)
Number of Points Generated
Upper 290
Lower 290

2412	0.0025	0.002465
0012	0.0025	0.0025
0012-64	0.0025	0.0025
23018	0.0025	0.002425

Figure 3. NACA Airfoil Geometry Spreadsheet.

In addition to the various software applications required, we also needed to supply the students with a relatively fast and easy-to-use computer system. We already had a fairly good research Linux parallel computer cluster ("Blackbird" Linux parallel computer with 120 processor Linux cluster, 60 processors of 1GHz/32Gb RAM and 60 processors of 2.4GHz/2Gb RAM). What we did not want was for the either the researchers or the students to "get in each other's way" by competing for computer space. To solve this problem, we used the replacement parts from "Blackbird" to create a second Linux cluster, "Strato", which is currently 40 processors of 1 GHz each, with a total of 12Gb RAM (see Fig. 4). This allows the students to do what students often do . . . mess things up! While we do not encourage the abuse of our computer systems, we knew that from time to time someone would find a way to overload the queuing system, or fill up the memory, or find some unknown new way to overwhelm the system. Having all of the students on this isolated system took much of the pain out of letting them loose on such a system, and allowed the researchers to feel comfortable that they would not have to stop work while the students used the computational resources.

C. Computational Aerodynamics Projects

We spent a great deal of time sorting through a variety of projects that we wanted to give to the students. We all have various projects that we believe are “essential” to learning CFD, and we wanted to avoid overwhelming the students with learning about everything we learned in graduate school—again, we had to remind ourselves of the goal of the course—to educate intelligent users of CFD. So we fashioned projects that would educate and inform, rather than require a great deal of software development.

Figure 4. The Student Computer Cluster “Strato” Along with the Mass Storage System.

The projects were broken down between computational aerodynamic projects and tutorial projects, with the main emphasis being on the computational aerodynamic projects. Tutorials were provided to help the students learn to function in the pre- and post-processing CFD environment with as little pain as possible. Included were tutorials in GridGen, FieldView, Tecplot, MatLab’s Power Spectrum Density capablity (for analyzing unsteady results) and a digitizing program (to digitize airfoil data). As the students were becoming familiar with the software applications, we began projects that would eventually use these applications for performing more advanced CFD experiments. These projects included:

- finite differencing and order of accuracy
- wave equation analysis
- heat equation analysis
- NACA airfoil numerical simulation at various pre-stall (steady flow) angles of attack
- NACA airfoil prediction at a post-stall (unsteady flow) angle of attack
- predicting airplane stability with a vortex lattice program

The first project showed the students the power and limitations of finite differencing. An analytic function, $u = -\sin(x) + \sin^2(2x)\cos(x)$ (see Fig. 5), with many local minima was used so that analytic derivatives could be evaluated and compared with results from various finite difference formulations.[9] The derivatives are evaluated with succeedingly smaller step sizes so that the order of accuracy of the formulations can be determined (1st order accurate vs. 2nd order accurate vs. 4th order accurate). This type of project can be done on a spreadsheet or in Matlab. This level of understanding is essential for being a good user of CFD, but we did not go into great detail about finite difference formulations other than to show how they are derived using Taylors series and comparing how well they work.

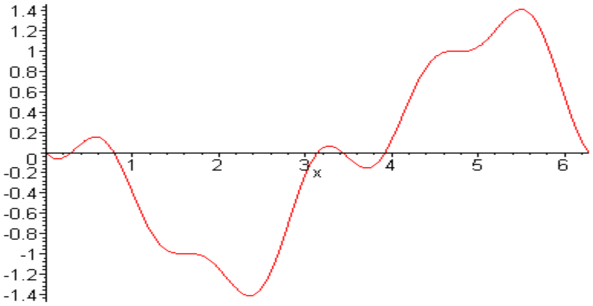


Figure 5. Finite Difference Project Analytic Function.

The next two exercises are fairly common for CFD courses, where the model equations for the various partial differential equation (PDE) types (wave equation for hyperbolic, heat equation for parabolic, and Laplace’s equation for elliptic) are simulated using finite-differences. The students use various explicit methods to solve the wave equation, and also use an implicit method to solve the heat equation. Through these exercises they are learning the connections between PDE types, initial/boundary conditions, algorithms, and accuracy of results.

The main exercise for the term deals with creating grids for, and analyzing the flowfield of an airfoil at various angles of attack. The students, having already learned to use a commercial grid generator, perform a grid sensitivity study for the airfoil prior to running cases at various steady flow conditions (up to but not including stall). They begin by creating the surface geometry for the airfoil using the spreadsheet shown in Fig. 3, and then determining grid spacing for laminar and turbulent flow at various Reynolds numbers using the following relationships:

$$\Delta_{lam} = L\frac{1.3016\ y^{+}_{ave}}{Re_L^{0.75}} \qquad \Delta_{turb} = L\frac{(13.1463 y^{+}_{ave})^{0.875}}{Re_L^{0.90}}$$

where they also learn about the importance of y^+ and the sub-layers within a boundary layer. The grid sensitivity study can be very time consuming, but by the time it is completed the students are quite proficient at making grids and analyzing results. A resulting unstructured grid around an airfoil is shown in Fig. 6, including the improved grid density in the airfoil wake.

Once the grid sensitivity study is completed, results are compared with available experimental data and conclusions are drawn about the ability of CFD to represent the flow over the airfoil (see Fig. 7 for a common comparison). A follow-on to the airfoil project is to run the airfoil at an unsteady flow condition (a post stall angle of attack). In order to perform this prediction the students need to learn about the time scales of the flow unsteadiness they are predicting, which requires them to run the solution at a variety of time steps and them to analyze the results using power spectrum density analysis. Unsteadiness that was predicted by students included vortex shedding at laminar Reynolds numbers, as shown in Fig. 8.

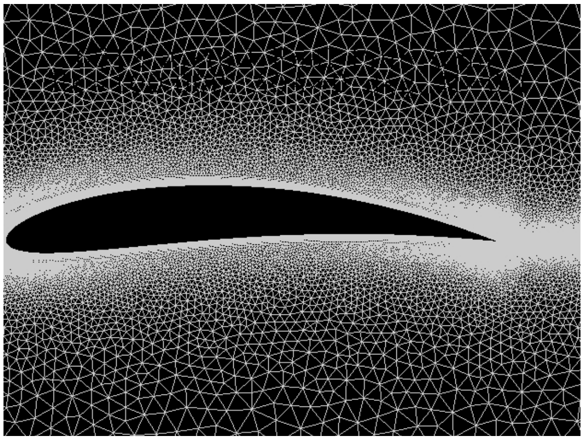


Figure 6. Unstructured grid generated about an NACA airfoil.

The final project of the semester was to show them that computational aerodynamics not only includes CFD but also various other methods, including panel methods and vortex-lattice methods. The students had already been exposed to panel methods in their fluid dynamics course, but we wanted to let them see the power of the vortex-lattice method for evaluating and entire aircraft configuration quickly. A PC version of the Vorlax program was created for the course and used by the students to evaluate. The students evaluate a T-38 at flight conditions and compare their results to available known performance data for the aircraft.[8] We also introduce the students to the concept of semi-empirical methods and discuss the importance of those methods in engineering analysis and design.

In every case, the projects were meant to convey not only the technical knowledge of the subject, but also the practical, hands-on guidelines for performing good CFD simulations, including grid sensitivity studies, time-step studies, laminar vs. turbulent results, unsteady flowfield studies, and comparisons with experimental data and determination of code validity.[11] The overall goal was to allow the students to develop critical thinking skills when it comes to the modern use of CFD.[12]

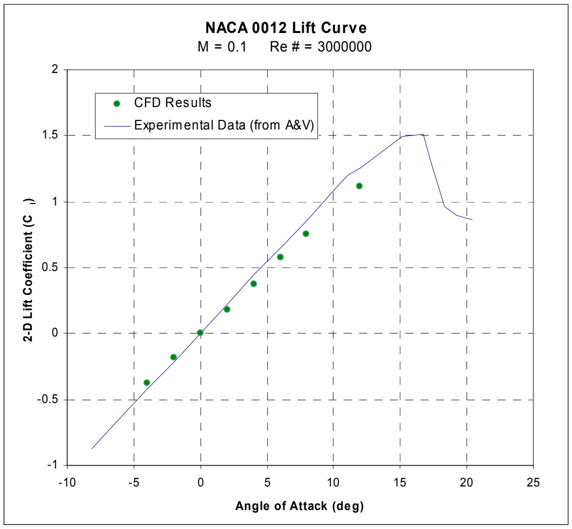


Figure 7. Comparison of CFD predictions with experimental data[10] for a NACA 0012 airfoil.

D. Aerodynamic Concepts Taught To Students

In spite of the fact that the new computational aerodynamics course seemed to concentrate on the computational aspects of CFD, a great deal of aerodynamic concepts were still taught to the students while performing their projects. Not only were aerodynamic concepts will taught, but a number of classic aerodynamic theories were also addressed during lecturing for various projects, especially for panel method and vortex-lattice projects.

Specifically, students learned the following aerodynamic concepts while performing their projects:

- Boundary conditions and their relationship to flow type (viscous or inviscid)

- Boundary layer thickness, growth, and velocity profiles
- Importance of understanding boundary-layer theory while doing CFD (sub-layer types and thicknesses, pressure gradients, etc.)
- Stagnation points and stagnation streamlines
- Flow separation and reattachment
- Laminar separation bubbles
- Airfoil/wing stall
- Airfoil pressure gradients as a function of angle of attack
- Airfoil surface and off-surface pressures, circulation, and the resulting lift and drag variations with angle of attack
- The relationship between pressure gradients and flow separation
- Pressure and skin friction drag
- Unsteady vortex shedding
- The impact of wing-tip vortices
- Compressibility effects at subsonic Mach numbers

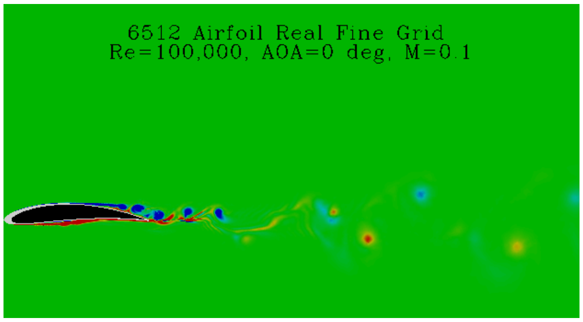


Figure 8. Prediction of laminar flow over NACA 6512 airfoil and resulting vortex shedding.

In addition, the following aerodynamic theories and concepts were taught to the students while they were performing their projects:

- NACA airfoil designations and data
- Potential flow theory
- Kutta-Joukowski theorem
- Thin airfoil theory
- Lifting-line theory

So, while many faculty members might believe they were giving up a great deal by teaching computational aerodynamics to their students, we found that many of the same concepts were still covered, just in different (and project-based) ways.

E. Lessons Learned

A variety of "lessons learned" have come out of the first offerings of our computational aerodynamics course. The course changed during the course of the semesters it was taught, and it will continue change next year. Certainly, we would have liked a portion of the course to be taught in activity/laboratory mode, since a great deal of time was spent learning software, computer systems, operating systems, etc. We have recently learned that our request to teach in a 2 contact hour mode has been approved, and we look forward to seeing how the course is improved because of this change.

Today's students do not (contrary to popular belief) know very much about computers, so we were very surprised at how difficult learning basic Unix commands was for them. Since they have spent their entire computer-using days using PC-based windows software, they had a great deal of difficulty "seeing" a directory structure and navigating directory trees in Unix. Us "old timers" did not understand this at first, since we grew up with punch cards, line editors, and command-based operations! Someday the students will get their wish (that all programs run on PCs), but for now, the breadth of skills required to perform their work was quite high.

There was a great deal of frustration among the students with the logistical overhead required to perform their work (using PCs, a Linux cluster, various commercial software, transferring files, etc.). We have spent the time since last year's course offering to eliminate as many of these difficulties as possible, and early trials with students have recently shown that our improvements are succeeding.

One of the student's major complaint was that they did not have a textbook for the course. We agree with them, since we spent many hours looking at the various CFD books and did not believe that any of them served our purposes very well (learning to be a good user of CFD, rather than a good creator of CFD codes). We are considering taking our notes and materials and writing a textbook specifically for undergraduates and engineers in industry to help them learn how to intelligently use CFD in their courses and work. We have already seen the benefits of the course as our students progress to their senior year, since they now have skills and capabilities that can be used directly in their aircraft design work—a valuable skill indeed! And as these students have gone on to their senior-year course in aerodynamics, we have not seen any appreciable difference in their knowledge of aerodynamics, since many of the most important lessons learned while doing their CFD projects have also served them well in understanding aerodynamics.

IV. Conclusions

A Computational Aerodynamics course was created at the U.S. Air Force Academy to teach all junior-level cadets the importance of modeling & simulation in aircraft design and aerodynamics. Details about the course, including topics covered, software used, projects performed, and lessons learned, have been presented and discussed in the this paper. Students were exposed to a wide variety of knowledge about computational aerodynamics, including how to solve the governing equations of fluid dynamics numerically, how to determine accuracy of results, how to insure grid independent results, and how to analyze unsteady flow problems. In all cases, comparisons with available experimental data were undertaken so that the students could learn the limitations of CFD, as well as importance of understanding experimental results. Vortex-lattice programs were also learned in order to see the importance of panel methods (and even semi-empirical methods) in performing engineering analysis and design work. Future improvements to the course have been proposed, many based on comments from the students who took the course last year. As the course evolves and improves, we believe that the students will fulfill our goal of becoming intelligent users of computational aerodynamics tools, and will be better prepared to meet the technical challenges of the future of aircraft design. Finally, we do not see this course as replacing traditional graduate-level CFD coursework, rather we see it as an important precursor to that work. Graduate students will still need to develop algorithms, write codes, generate turbulence models, and discover new ways to perform CFD simulations. We just believe that undergraduates can understand those concepts without becoming experts in the field.

Acknowledgments

The authors would like to thank the Aeronautics cadets at the U.S. Air Force Academy who have taken the new course in computational aerodynamics. Their feedback was crucial to discovering what was possible and what was beyond the scope of undergraduates who were first learning about computational aerodynamics. We also want to thank the other members of the faculty and staff who helped to make the course a success: James Forsythe (now at Cobalt Solutions, LLC), Douglas Blake (now at AFRL), Barrett McCann, Robert VanDyken, Steven Senator, and Dave McDaniel. Finally, we want to thank Raymond R. Cosner of The Boeing Company who encouraged us to create the course and find an improved way to educate students about aerodynamics.

References

[1] Steger, J.L. and Hafez, M.M., "CFD Goes to School; The University's Role," *Aerospace America*, Vol. 30, No. 1, Jan. 1992, pp. 38-42.

[2] Schiff, L.B., private communication.

[3] Darmofal, D., Murman, E., "Re-Engineering Aerodynamics Education," AIAA Paper 2001-0870, Jan. 2001.

[4] Luckring, J.M., Hemsch, M.J., and Morrison, J.H., "Uncertainty in Computational Aerodynamics," AIAA Paper 2003-0409, Jan. 2003.

[5] Mason, W.H., "Teaching Aerodynamics in the IT Age," AIAA Paper 2002-2725, June 2002.

[6] Drela, M., "Assorted Views on Teaching of Aerodynamics," AIAA Paper 98-2792, June 1998.

[7] Holst, T.L., "Computational Fluid Dynamics Uses in Fluid Dynamics/Aerodynamics Education," NASA TM-108834, 1995.

[8] Brandt, S.A., Stiles, R.J., Bertin, J.J., and Whitford, R., *Introduction to Aeronautics: A Design Perspective*, Reston, VA: AIAA, 1997.

[9] Pulliam, T.H., private communication.

[10] Abbott, I.H. and von Doenhoff, E., *Theory of Wing Sections*, New York: Dover Publications, Inc., 1959.

[11] Cosner, R.R., Oberkampf, W.L., Rahaim, C.P., and Shih, T.I.P., "AIAA Committee on Standards for Computational Fluid Dynamics—Status and Plans," AIAA Paper 2004-0654, Jan. 2004.

[12] Nield, B.N., "Overview of the Boeing 777 High Lift Aerodynamic Design," *Aeronautical Journal*, Vol. 99, No. 989, 1995, pp. 361-371.