INCORPORATING CFD INTO THE UNDERGRADUATE MECHANICAL ENGINEERING PROGRAMME AT THE UNIVERSITY OF MANITOBA

Scott J. Ormiston
SJ_Ormiston@UManitoba.ca
Associate Professor, Dept. of Mechanical and Industrial Engineering
University of Manitoba, Winnipeg, Manitoba, Canada

ABSTRACT

This presentation discusses three aspects of CFD in undergraduate engineering education. First, general ideas are presented about issues affecting a CFD course in an undergraduate mechanical engineering curriculum. This discussion is centred around defining the background of the students, the CFD knowledge required by a graduate mechanical engineer, and the possible career paths of graduating students. The key concepts that should be part of CFD-related undergraduate material, the style of the course, and the role of university facilities are also discussed.

Second, the pros and cons of different methods for delivery of CFD material are presented. It is proposed that the optimal method to use depends on the background of the students and the course objectives.

Third, specific experiences at the University of Manitoba are related. The evolution of a mechanical engineering fourth year course that covers CFD-related material is described. The details of the course history are described in the context of the mechanical engineering programme.

INTRODUCTION

Recently there has been a significant change in users of computational fluid dynamics (CFD) software. A shift has taken place from a user community with terminal degrees at the doctoral level to one with an increasing proportion of terminal degrees at the bachelors level. This shift raises questions about how members of the CFD community obtain the necessary knowledge to work effectively. If bachelors degree graduates are to be CFD practitioners, then how much understanding of CFD basics should they have when they start a job and how will they obtain that? Is on-the-job training sufficient? Are professional short courses or continuing education courses available and adequate? How much of the CFD education should be shifted to an undergraduate programme?

This article expresses the viewpoint of the author on how CFD-related material may be incorporated into a mechanical engineering undergraduate programme. Both general ideas and specific experience are discussed. The views herein are based on the author's experience for ten years as a faculty member in the Department of Mechanical and Industrial Engineering at the University of Manitoba, which has offered an introductory CFD course at the undergraduate level since 1989.

At the University of Manitoba, the Department of Mechanical and Industrial Engineering has at present 20 faculty members, 6 of whom teach courses in the area of thermofluids (thermal sciences and fluid mechanics). Co-operative education streams are available in the mechanical and the industrial programmes; an Aerospace engineering option is offered in the mechanical engineering programme. Each year, approximately 55 students graduate from mechanical engineering and about 15 from industrial engineering.

This article focuses on three topics related to undergraduate CFD education. First, a general discussion of the issues related to designing an undergraduate mechanical engineering CFD courses are discussed. Second, the advantages and disadvantages of different methods of evaluating students are presented. Third, the specific experience at the University of Manitoba is described in detail.

ISSUES

There are many general questions that can be posed in the process of determining how to deliver CFD-related material in an undergraduate programme. The key assumption here is that there will be a single, primary course in which most of the CFD-related material will be delivered. The issues can be described by looking at the course as if it were a computer code: with input, output, a process to perform, and an environment in which it operates.

Input

The input to the course is the background knowledge and skills of the students entering the course. The areas

where the student should have pre-requisite knowledge are:

- mathematics (calculus, differential equations)
- numerical methods (theory)
- computer solution of an engineering problem (by programming in a high level language, using analysis software, or using a spreadsheet)
- fluid mechanics and heat transfer (conduction, convection, fluid flow governing equations)

Output

The output for the course is the knowledge required by the graduate engineer. This requirement is difficult to determine because the career paths of all graduates cannot be predicted. It can be estimated that some graduates will go on to do detailed research in CFD, some will become users of commercial CFD codes in industry, and others will have, at most, only discussions with colleagues or clients that do CFD analysis. In any case, it could be argued that CFD is now ubiquitous and some minimal knowledge of CFD is needed for practising engineers[1]. It is also expected that those who practice CFD will receive more detailed training on the job.

Table 1 is offered as a rough classification of CFD course topics. The minimal knowledge of the graduate engineer is proposed to be all of the topics in the "Basic" category and a few selected topics in the "Intermediate" category.

Environment

The environment in which the CFD course is to be delivered can have a major impact on the course because of the potential dependence of the material on computing facilities and software. The amount of departmental, faculty, and campus-wide support in terms of software licences, installation and maintenance plus computing resources must be carefully assessed.

Another aspect of the environment of the course is whether it is a core course or an elective. The elective environment typically has fewer and more motivated students. In the elective case, expectations about the ability of the students to do advanced material may be higher.

Process

The way the course is structured and the material is delivered is dependent on the other issues. The student background and required output affect the level of the material and to what extent programming assignments, projects, case studies, or commercial software may be used. The resources available may also limit choices in how the process is implemented. Further discussion of possible techniques used in a CFD course appears in the next section.

TECHNIQUES

There are many possible ways to teach CFD to undergraduate students. Previous discussion indicated that all the issues in designing an undergraduate CFD course are inter-dependent. There will be some techniques that could be used in many environments and others that will be unique to a particular institution. The optimal choice of techniques will depend on the institution and the criteria used to decide what is optimal. Those criteria my be related to how well the CFD course prepared the graduate for future work.

It has been assumed that course material is lectured, that the evaluation is some combination of tests (term and/or final exam) and assignments (problems and/or a project). It has also been assumed that the assignments implement the techniques. Assignments can be individual or group efforts and either theoretical or practical. In the practical work, a student performs some computations and sees the implementation of the theory discussed in lectures. The computations in the practical work can be made using software that the student develops via programming, or by using simple or complex existing software. Table 2 lists some possible evaluation methods for an undergraduate CFD course along with advantages and disadvantages of each approach.

UNIVERSITY OF MANITOBA EXPERIENCE

Background

In 1989 a decision was made to introduce CFD into the undergraduate mechanical engineering curriculum at the University of Manitoba. Prior to 1989, the existing CFD-related material was brief coverage of finite difference methods for conduction in a topics in a heat transfer elective and for a boundary layer flow in a fluid mechanics elective. The development of the new course: 025.482 Computational Methods for Thermofluids, is described below. From the beginning the focus of the course material has been on a finite volume method.

[1] It is to be hoped that this point will be made clearer through other presentations and discussion in this session of the conference.

The First Three Years (1989,1990,1991)

An introductory course in CFD was developed and first taught by Dr. D.W. Ruth[2]. In the first two years, the course was taught as a topics in heat transfer elective. The students had been taught Fortran 77 (Watfor77) in first year in a Computer Science course and had used it for an assignment in a second year mechanics course. They had studied matrix algebra, calculus and differential equations in standard applied mathematics courses up to third year.

In the first two years, students were given Fortran codes for steady and unsteady 1D conduction calculations and part of a code for 2D incompressible flow. A project involved individuals developing their own code to solve for steady flow between parallel plates. In the first year, motivated students were attracted to the elective and the project was a success. In the second year as an elective, however, much of the remainder of the graduating class took the course and the project proved to be quite difficult for them.

In the third year, the course became core for mechanical engineering students and was called 025.482 Computational Methods for Thermofluids. The codes for 1D conduction were still given to the class, but because of the experience of the previous year, Professor Ruth changed the project to an open-ended (design) analysis of 2D steady conduction.

In all three years, typeset hand-out lecture notes were provided by Professor Ruth. The well known book by Patankar [1] was used as an optional reference. Evaluation consisted of 6 assignments, a project, and four term tests tests. Enrolments in the three years were in the range of 15 to 40 students.

Evolution of the Course (1992 to present)

The author assumed responsibility for teaching 025.482 in the fall of 1992. In the years 1992 to 1996, there was a typical enrolment of 65 students; after 1996, enrolment dropped to roughly 40 because of the start in 1997 of the Aerospace Engineering option in which students do not take 025.482.

Initially, the overall structure was the same as it had been, except that the lecture notes were only on the chalk board in class, Patankar [1] was a required text book, and the assignments and the project were modified slightly. Fortran codes (slightly different) were still given out for 1D conduction, but codes were also provided for 1D advection-diffusion and 2D convection. Running the 2D convection code to solve the lid-driven cavity flow was used as one of 6 assignments. The project was done in groups of 2 or 3 and involved modifying a 2D steady conduction code to perform an open-ended analysis in a design project. All computing work was done on Unix and there were 4 term tests and no final exam.

Changes were made by the author to 025.482 over the years since 1992. The four key factors addressed in the changes were:

1. No suitable course text book. While Patankar is an excellent book and still a valuable reference, its advanced-level end-of-chapter exercises and shortage of worked examples make it more suitable for a graduate course. It was found that only the basic finite volume method material was suitable for the undergraduate course and that some students were unwilling to pay the high purchase price to use only that portion of the text book.

   There are other, more recently published CFD books that have been considered [2, 3, 4, 5]. Anderson [2] and Tannehill *et al.* [3] contain significant extra advanced and aerospace-related material that would not be used. The book by Ferziger and Perić [4] covers a finite volume method well but is more suitable for a graduate course. Versteeg and Malalsekera [5] is more suitable for undergraduate level but does not have end of chapter problems, has detailed coverage of turbulence modelling that would not be used, and the basic finite volume material is not very different from the existing lecture notes. For these reasons, the author typeset his lecture notes and they are sold at low cost in the University of Manitoba Bookstore [6].

2. Lack of basic numerical methods background. Because the University of Manitoba mechanical engineering students did not have a prior numerical methods course, they were lacking an understanding of a finite difference method and had no practical experience using the computer to solve sets of algebraic equations. A partial remedy to this problem was to add review of matrix equations, Taylor Series, and errors along with an excerpt from Chapra and Canale [7] as a required reference. The required reference was subsequently changed to an excerpt from Jaluria [8]. In addition, the heat transfer course text book [9] is now used as a reference for coverage of finite difference material.

   A numerical methods course was introduced into the mechanical engineering curriculum (in a change to a common numerical methods course for all en-

[2]Background information provided by Dr. D.W. Ruth is gratefully acknowledged.

gineering disciplines) in 1998. It is taught at the second year level by the Applied Mathematics department and provides limited experience in the solution of engineering problems. Students still lack practical experience in numerical methods.

3. Above average work load. It was recognized that the amount of time and effort required in the course was well above average. Therefore, in subsequent years evaluation was modified to 4 smaller assignments, two term tests, a project, and a final exam.

4. Change in students' programming abilities. Initially (in 1992 and 1993), mechanical engineering students' first course in programming was in Fortran 77 and there were assignments or project requiring programming in second and third year (a computer-aided engineering course with finite element method material). More recently, the programming requirements have been removed from second and third year. In addition, the first year programming course taught by Computer Science changed to C++ in 1998 and then to Java in 2000. All these changes have created an overall reduction in programming ability of the students entering the course. In addition, the Fortran versions of code provided for assignments and projects can no longer be read by a significant portion of the class. This poses no problem for assignment codes (which could be just executables), but required the temporary (in 2000) elimination of both a small programming assignment and the requirement of code modifications in the project.

Plans for the Future

Some ideas that are being considered in the future are listed below.

1. Develop a simple programming assignment in multiple languages (and possibly mix C/C++ with existing Fortran 77).
2. Move to greater use of spreadsheet and MATLAB for assignments.
3. Put a greater focus on validation, checks of solutions, and physical interpretation of results.
4. Put less emphasis on fluid flow solution details.
5. Move some of the course material to an elective.

It is hoped that it will be possible to work with Computer Science department to move toward a first programming course covering more fundamentals of C/C++ with engineering applications and with the Applied Mathematics department to use programming and MATLAB assignments in the second year numerical methods course. Otherwise, background material implementation of items 1 and 2 above must be included in 025.482.

CLOSING REMARKS

Defining the student background, the required CFD knowledge of the graduate, the resources available and the methods to be used were identified as the key issues in establishing an undergraduate CFD course. Methods to be used were presented in the context of evaluation of the students. The advantages and disadvantages of assignments covering theory, involving programming, using simple or complex software, and analysing a system in a project or case study were presented briefly. At the University of Manitoba, a fourth year elective introductory CFD course was started in 1989. The course has been in the mechanical engineering core since 1991. The key factors affecting the evolution of the course were the choice of text book, the course work load, and the students' background in numerical methods and computer programming.

REFERENCES

[1] S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere, Washington, 1980.

[2] J. D. Anderson, *Computational Fluid Dynamics The Basics with Applications*, McGraw-Hill Inc., New York, 1995.

[3] J. C. Tannehill, D. A. Anderson, and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, 2nd ed., McGraw-Hill Inc., New York, 1997.

[4] J. H. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 1996.

[5] H. K. Versteeg and W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Longman, England, 1995.

[6] S. J. Ormiston, 25.482 Computational Thermofluids Supplementary Course Notes V3.0, sold in the University of Manitoba Bookstore, Department of Mechanical and Industrial Engineering, University of Manitoba, Sep. 1999, 154 pages.

[7] S. C. Chapra and R. P. Canale, *Numerical Methods for Engineers*, 2nd ed., McGraw-Hill, Inc., New York, 1988.

[8] Y. Jaluria, *Computer Methods for Engineering*, Allyn and Bacon, Inc., Boston, 1988.

[9] F. P. Incropera and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 4th ed., John Wiley and Sons, New York, 1996.

Table 1: CFD topics

Basic	Intermediate	Advanced
• basic discretization • stability • convergence • TDMA • grid independence • validation • boundary conditions • common sense checks • balances • relaxation • error calculation	• aspect ratio effects • advection-diffusion • 2D solvers • source term linearization • sets of equations • upwind schemes	• multigrid • 3D discretization • 3D solvers • code development • grid generation • higher order upwind • turbulence modelling • coupled equation solvers • pressure-velocity coupling • error analysis

Table 2: CFD Evaluation Methods

Method	Advantages	Disadvantages
Theory assignment	• general purpose • few resources needed	• do not see implementation
Programming assignment	• students get appreciation for implementation effort • develop/practice debugging skills • connect theory and practice • get source code to use later	• programming ability required (tutorial, T.A. support needed) • suitable computing environment needed (compiler/editor/debugger) • may need to support multiple languages
Simple software use in an assignment (e.g. Spreadsheet or MATLAB)	• commonly accessible software • students should already know how to use the software • on-line documentation • high probability that software will be used by many students after graduation	• less flexible than a high level language
Complex software use in an assignment (in-house or commercial CFD code)	• more sophisticated analysis possible • students don't need to write any code	• long learning process • students don't know what code is doing • in-house software may not be well-documented • commercial software licence and support needed • low probability that many students will use the software after graduation
Group project (design project or case study)	• more realistic problem can be studied • can be a more complete and open-ended analysis • team work experience	• significant time needed to learn project details • significant time needed to learn to apply software to the project • level of difficulty may be too high for some students • unequal sharing of work load by group members • commercial software licence and support needed