INTEGRATED APPROACH OF COMPUTING AND COMPUTER APPLICATIONS AT AN EARLY STAGE OF UNDERGRADUATE ENGINEERING PROGRAMS

Muthanna Al-Dahhan

Associate Professor, Associate Director of the Chemical Reaction Engineering Laboratory
Chemical Engineering Department, Washington University

St. Louis, MO 63130, USA
muthanna@che.wustl.edu

ABSTRACT

Recently, the role of computing in solving engineering problems, design, data analysis, communication, accessing information, etc. increased dramatically. Hence, this has led to a great demand for computer applications skills in the curricula and practice of various engineering disciplines. As a result, courses in engineering programs have become more computer-oriented at all levels of the curriculum. This raises the following questions: 1) What undergraduate engineering students should know regarding computers and computing? And 2) How best should undergraduate engineering students acquire the computing skills, systematically and effectively, at an early stage of their academic program?

At Washington University-Chemical Engineering Department, we have developed and taught an effective early-years (first or second year) computing course that provides in a systematic manner, the needed computer and computer applications skills for solving engineering and design problems. The course, "Introduction to Computing and Computer Applications" is based on the premise that the engineering students should be provided with an integrated, strong and early introduction to computing techniques and packages. The materials are presented in a form that allows students to follow the logic involved and to understand the relationship between the computer, programming using C/C++ language, numerical computing, computing software packages, such as Matlab, Mathematica, Excel, IMSL, and practical skills for solving design and engineering problems.

In this paper, this course and its approach and materials will be discussed and presented.

Keywords: *undergraduate engineering students, computing course, engineering education*

الملخص

في الو قت الحا ضرأصبح د و ر ألحا سو ب ألا لي أ سا سيا في حل ألمسا ئل ألهند سية ، ألتصا ميم ، تحليل ألمعلو ما ت ، ألا تصا لا ت، الخ . ها ذا ألتطور يتطلب إد خال تطبيقات ألحا سو ب ألا لي في مناهج ألدراسه ألهندسيه وتهيئه ألطلبه لذالك في ألمراحل ألاولى للدراسه. في جامعه واشنطن في سانت لويس قمت في صياغه و تدريس درسا فعالا للصف ألاول و ألثاني لبرامج ألدراسه ألهندسيه عنوانه "مقدمه إلى ألحسابات ألهندسيه و تطبيقات ألحا سو ب ألا لي". يعتمد ألدرس على تدريس ماده متكامله في ما يخص لغه ألحا سو ب ألا لي ، ألطرق ألعدديه و تحليلاتيها ، ألبرامج ألتجاريه وأستخداماتها في حل ألمسا ئل و ألتصاميم ألهندسية. في هاذا ألعرض سوف نشرح محتويات هدا ألدرس.

1. INTRODUCTION

Today the number and types of computer and network applications are vast to count with the current advances made in the development of fast and efficient computers and networks. Hence, the role of computing in solving engineering problems, design, data analysis, communication, accessing information, text processing, graphics, etc. has increased dramatically. As a result, courses in chemical engineering and other engineering disciplines have become more computer-oriented at all levels of the curriculum.

The surveys conducted by CACHE corporation curriculum committee showed that the chemical engineering graduates overwhelmingly considered computing to be an integral part of the undergraduate program (Davis, et al., 1995). This raises the question as to what undergraduate chemical engineering students should know regarding computers and computation. It is evident that the student must (Al-Dahhan, 1995):

- Have access to computing facilities and Internet.
- Be efficient in programming in at least one high-level language (e.g. C/C++ or Fortran).
- Be capable of implementing numerical computing techniques on the computer.
- Be exposed to the use of available software packages for computing, data analysis, design, graphics, etc.
- Be able to use the computer for technical calculations, problem solving, data processing, process design and simulation, visualization, etc.
- Be ready to efficiently use computing support facilities such as operating systems, editors, searching the worldwide web, etc.

Another difficult question that should be addressed is: How best should undergraduate chemical engineering students acquire those skills, systematically and effectively, at an early stage of their academic program.

Unfortunately, most of the current practice is still teaching programming in a selected language (e.g. C/C++, Fortran, etc.) in the early years of the program. Although, this helps students acquire some of the above-mentioned knowledge, it does not provide them with the essential knowledge and skills of how to solve engineering problems, conduct data analysis and design computing, etc. The needed skills mentioned earlier are often introduced by more than one course at different levels of the curriculum. For example, numerical methods and analysis are usually offered at the senior or graduate level. Clearly far too late for the students to use the material in their engineering courses. This does not work very well, since if the students encounter a computational problem in their homework or projects without having the knowledge of how to solve it using computer, it detracts them from the engineering and scientific principles taught in that course. In addition, the students struggle with the computational tools needed to solve the problem at hand and end up losing the perspective of both the scientific/engineering concepts and the computational method.

Hence, it is much more beneficial to the students to have a single required course early in the curriculum that focuses on the skills mentioned previously. Development and teaching such a course is not easy, but are essential for a productive engineering program. At Washington University – Chemical Engineering Department, we have developed and taught such a course that provides in a systematic manner the required computing skills. The course entitled "Introduction to Computing and Computer Applications" is based on the premise that chemical engineering students, in particular, and engineering students in general, should be provided with an integrated, strong and early introduction to computing techniques and packages for solving engineering problems.

2. COURSE STRUCTURE AND CONTENTS

The course have been offered each semester every year since 1991 and required from all engineering students including chemical engineering students. It is assumed that students have had little experience with the computer and its applications in programming, computing, data analysis, communications, etc. Since 1991 the course has been improved and its contents have been modified to fulfill the needs of the students, to implement the comments received during the formal evaluation process conducted at the end of each semester and to utilize the recent advances and changes made in computing software and programming languages. For example, Fortran language was first covered. In 1999, C/C++ language has been used based on the students' request and endorsement of the faculty. Also only Matlab has been used since 1997 instead of both Mathematica and Matlab. Table 1 shows the recent outline of the course structure and contents. The course is structured to achieve the following goals:

- To acquaint the students with the computer and to familiarize them with computing support facilities (such as the operating system, editor, compiler etc.)
- To make the students efficient in programming using C/C++ language.
- To introduce the students to numerical methods and computing and their effective implementation on the computer using a high level language (e.g. C/C++ or other desired language for engineering computations).

Top

- To use the commercially available and widely used computing software and library packages such as Matlab, spreadsheets, IMSL and NAG.
- To practice the acquired knowledge by solving design and engineering problems via homework, exams and project problems.

Although such course or similar course have been offered in other schools, the unique feature of this course is that the material is presented to allow the students to follow the logic involved and to understand the relationship between the computer, programming, numerical computing, computing software packages and practical skills for solving engineering and design problems as shown in Table 1. The course contents are organized into three subjects that include: 1) programming in C/C++, 2) numerical computing and 3) computing software packages such as Matlab, Mathematica, IMSL and Excel, which are used for computing, data analysis and processing and graphics. Each week, some elements of C/C++ programming, numerical computing techniques and computing by commercial software are given. A topic of programming in C/C++ is introduced first followed by the numerical techniques that will be implemented on solving engineering problems using the preceding covered topic in C/C++ programming. In other words, we let students practice programming by implementing numerical computing. For instance, when repetition (loop) structures are introduce, numerical integration is presented where a loop structure is implemented; when array processing is introduced in C/C++, matrices and numerical techniques for solving a set of linear equations are discussed. Students are then introduced to the commercial computing packages to solve the same problem using the covered numerical techniques. This sequence helps students appreciate the flexibility of programming as well as the ease, capabilities and power of computation using software packages for solving engineering problems. In addition, such approach provides students with an efficient way to learn both the numerical techniques and programming as also confirmed by Chapra and Canale (1988). Engineering and science problems are practiced through illustrated examples, workshops, homework assignments and design projects. Although, the course seems to be overloaded, the structure and material presented (Table 1), the provided text-note and the teaching style have allowed most of the students to complete the course without be overloaded. It is noteworthy that, usually engineering students do not need to take any computer or computing course after this course.

3. TEXTBOOK

Numerical computing is vital for solving engineering problems as in practice the solution of most engineering and scientific problems requires the application of numerical computing techniques. In order to match the level of the class, an effective computing text was developed in the form of course note (Al-Dahhan, 2000). It provides an integrated knowledge of C/C++/Fortran programming, numerical methods and the usage of the available computing software packages. The text-note is produced and sold to the students in two volumes by the chemical engineering department office. C/C++ programming sections are presented to follow the approach of the materials listed in Table 1. Many examples with their solutions are provided to enrich the knowledge of students in programming and in implementing various features and flexibilities of C/C++ language. The numerical techniques are presented in a simple manner and easy to understand. Each technique is covered by: 1) a short introduction describing the mathematical basis for the technique, 2) steps of the algorithm needed for the implementation, 3) illustrated examples by hand calculator, following the algorithm steps for the technique, 4) translation of the algorithm steps into a programming flowchart that can be readily implemented in a computer program, 5) programming recommendations and comments. The commercial software packages for computing such as Matlab, Mathematica, Excel and IMSL are summarized in a simple manner with examples that demonstrate their applications for the computing techniques covered in the course.

4. HOMEWORK ASSIGNMENTS

Homework assignments are vital in helping the students absorb the subject properly and in recognizing the importance of such skills to their engineering program and their career. They also help students develop hands-on experience in solving problems by computer. Thus, real-world engineering and design problems are used as the homework assignments. Since the course is not a course in modeling, the statement of the problem is stated clearly and the necessary derivation is discussed. The students recognize that the final derived models or equations cannot be solved analytically or by hand and computer and computing skills must be implemented. Exhibit 2 shows an example of one of the design oriented problems given to the class.

5. COMPUTER LABORATORY WORKSHOP SESSIONS

The workshop sessions are organized, either on weekly basis or at least every other week, in the computer laboratory and are overseen by the instructor and the teaching assistants. The objectives of these sessions (1 – 1.5 hour long) are to supplement lectures, to enhance the needed skills of the students for computing and computer applications, to practice developing different programming structures and debugging them and to demonstrate the applications of the computing packages in solving engineering problems. The assignments are prepared to achieve these objectives.

6. DESIGN PROJECTS

ABET 2000 criteria (EC2000) for accreditation of the chemical engineering programs require implementation of design across the curriculum. Hence, in addition to the design-oriented homework assignments, design-oriented projects are implemented into the course. The students are asked to formulated design-oriented problems from other engineering and science courses, derive the equations needed and implement the learning skills to perform the design calculations.

Top

7. REMARKS

At the end of each semester, evaluation is usually conducted by asking each student to fill an evaluation form designed to properly evaluate the contents of the course and the instructor's teaching effort. The main objectives of such evaluation process are to realize the students concern, if any, to assess whether the goals set for the course have been met or not and to improve the course material and teaching process. Overall, the student's impression of this course is always very favorable as confirmed by their comments and evaluation provided in the evaluation forms, by personal contacts later during their third and fourth years of the program. The benefits of the course include: 1) producing a favorable impact on the chemical engineering courses that use computer-based calculations as a tool, 2) making students efficient in programming of useful techniques, 3) providing a focused and early introduction to numerical computing, 4) using efficiently the available computing software and library packages, 5) developing skills and knowledge required to solve practical engineering, design and scientific problems efficiently.

8. REFERENCES

1. Al-Dahhan, M.H., 1995, "Computing in the Undergraduate ChE Curriculum an Integrated Approach," *Chemical Engineering Education*, pp. 198, Volume: Summer issue.
2. A-Dahhan, M.H., 2001, "Introduction to Computing," Chemical Engineering Department, Washington University, St. Louis, MO.
3. Chapra, S.C. and Canale, R.P., 1988, *Numerical Methods for Engineers*, McGraw-Hill, Inc. 2nd ed..
4. Engineering Criteria (CEC) 2000, 1998. *Accreditation Board for Engineering and Technology, Inc*., 111 Market Place, Suite 1050, Baltimore, MD, USA.

Table 1: Course Structure

WEEK	SUBJECTS
1&2	Introduction to computer architectural components and their functions
	Running computer programs: editing, compiling/linking, executing
	Procedure for solving problems using computer
	Introduction to numerical computing and engineering problem-solving
	Introduction to computing software: Matlab, Mathematica and Spreadsheets
3	Programming: C/C++ syntax / Arithmetic computations
	Numerical computing: The Taylor series
	Matlab & Spreadsheet: Building up calculations and data processing

Top

4	Programming: Simple input and output Numerical computing: Numerical differentiation Matlab & Spreadsheet: Algebraic calculations & Functions; Symbolic computations
5	Programming: Control structure – If structures Numerical computing: Root of equations and Conversion criteria Matlab & Spreadsheet: Root equations computing Design Calculations
6	Programming: Control structure – loop structures Numerical computing: Numerical integration Matlab & Spreadsheet: Computing of integrals Design Calculations
7	Programming: Data files and additional input/output features Numerical computing: Review of basic matrix terminology and operations Matlab & Spreadsheet: Handling data files and matrices computing Design Calculations
8	Programming: Array processing Numerical computing: Systems of linear algebraic equations Matlab & Spreadsheet: Matrix and array computations, solving sets of linear equations Design Calculations
9	Programming: Array processing, continue Numerical computing: IMSL subroutines library package Matlab & Spreadsheet: Functions and more computations Design Calculations
10	Programming: Subprograms – Functions Numerical computing: Systems of nonlinear algebraic equations / IMSL Matlab & Spreadsheet: Solving nonlinear equations
11	Programming: Subprograms – Functions, continue Numerical computing: Curve fitting / IMSL Matlab & Spreadsheet: Curve fitting / statistical analysis, graphics computing Design Calculations
12	Programming: Additional programming features Numerical computing: Interpolation / IMSL Matlab & Spreadsheet: Fitting and Interpolation, graphics
13	Programming: Review Numerical computing: Differential equations / IMSL Matlab & Spreadsheet: Differential equations, graphics
14	Review

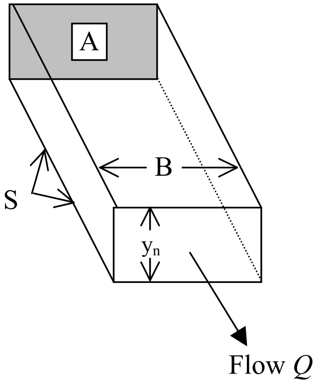
Top

Exhibit 1: An Example of the Homework Assignment

Design of a Channel for Water Distribution

Statement of the Problem

Design an open channel with a rectangular cross section area (A) for water distribution as shown in the figure below:



Under uniform flow conditions, the flow is calculated by the following relationship, based on Manning's equation

$$Q = \frac{y_n B}{n}\left(\frac{y_n\ B}{B + 2y_n}\right)^{2/3} S^{1/2} \tag{1}$$

where n is a roughness coefficient. This equation is used by fluid and water resources engineers to design open channel by estimating the normal depth (y_n). If (y_n) value is less than the critical depth, the flow is subcritical. The critical depth is calculated by:

$$y_n = \left[\frac{Q}{B^2\ g}\right]^{1/3} \tag{2}$$

where g is the acceleration due to gravity $(980\ cm/s^2)$.

Assignment:

Design an open channel to distribute water by calculating the required depth for the following values. If the calculated depth is larger than the critical one, select a volumetric flow rate that ensures subcritical flow in the channel.

$$Q = 14.15\ m^3/s,\ B = 4.572\ m,\ n = 0.017,\ \text{and}\ s = 0.0015 \tag{3}$$

Substituting these values in eq (1), yields a nonlinear design equation to calculate y_n as follows:

$$14.15 = \frac{y_n\ 4.572}{0.017}\left(\frac{y_n\ 4.572}{4.572 + 2y_n}\right)^{2/3}(0.0015)^{1/2} \tag{4}$$

Perform the design computation according to the following:

a. 1. Develop a complete program using Regula-Falsi method to calculate the fluid depth y_n for the equation above.
   2. Determine whether the flow is sub- or super-critical.
   3. Use input statement to provide your input values.
   4. Print the output as follows:

```
Design Calculation for Water Distribution Channel using Regula-Falsi Method

                    Iteration no.     Fluid depth
                    ----------------------------
                          1            xx.xxxxx
                          2            xx.xxxxx
                          3            xx.xxxxx
                          '                '
                          '                '
                          '                '

                                   until converge.
```

b. Use Matlab to perform the design calculation in (a) above using Matlab function(s).

c. If the flow is supercritical, design a channel in which the flow should be subcritical. Use the developed program and Matlab to perform such design calculations.

d. Convert the developed C++ program into a Matlab program in a script file. Read the inputs from a file and print the outputs to a Matlab file.

Top