M. Shacham, N. Brauner and M. B. Cutlip, "From Numerical Problem Solving to Model Based Experimentation – Incorporating Computer Based Tools of Multiscale Modeling into the ChE Curriculum", pp. 32 - 35 in F. Scura, M. Liberti, G. Barbieri, and E. Drioli (Eds), 5th Chemical Engineering Conference for Collaborative Research In Eastern Mediterranean Countries (EMCC5), May 24-29, 2008, Cetraro (CS) – Italy

From Numerical Problem Solving to Model Based Experimentation – Incorporating Computer Based Tools of Multiscale Modeling into the ChE Curriculum

Mordechai Shacham[1], Neima Brauner[2], Michael B. Cutlip[3]
[1]Dept. Chem. Eng., Ben-Gurion University, Beer-Sheva, Israel
[2]Faculty of Engineering, Tel-Aviv University Tel Aviv, Israel
[3]Dept. Chem. Eng., University of Connecticut, Storrs, CT 06269, USA

1. Introduction

Many of the challenges facing chemical engineering departments regarding the use of computers in undergraduate education were recently reviewed by Edgar[1]These challenges come about because of the substantial growth in the number of multiple-purpose and dedicated software packages used in the chemical industry, education, and research. At the same time, there remain substantial practical and educational benefits to teaching computer programming using languages such as Visual Basic, C or C++. There is obviously not enough room in the undergraduate curriculum to include all the courses required to teach computer programming and all the state of the art software packages currently used in chemical engineering. Thus, it is essential to construct a general framework that enables sufficient coverage of computer programming and the use of multiple purpose and dedicated software packages.

The paper is organized as follows. In sections 2 and 3 the current computing needs in the academia and the chemical industry are reviewed. In section 4 a new introductory course for modeling and computation is introduced and finally, the proposed framework for incorporating the tools for multi-scale modeling into the ChE curriculum is presented.

2. Computing in the academia

Seader[2] reviewed the education and training needs of chemical engineers related to the use of computers almost 30 years ago. Since then, this field has expanded considerably. Now it includes the study of computer languages, problem solving using numerical and statistical methods, process simulators, computational fluid dynamics (CFD), virtual laboratory experiments, process and product design and molecular modeling (Edgar[1]). A more detailed description of some of these issues follows.

2.1 Study of computer languages.

The study of computer languages such as Fortran has been included in the ChE curriculum since the nineteen sixties. In the early days of computing, studying computer languages was essential to enable numerical solution of engineering problems. However, upon the introduction of the mathematical software packages (e.g., spreadsheets, Mathcad, POLYMATH) in the mid '80s, their practical importance has somewhat diminished. This trend is also reflected in the small percentage of the practicing engineers who use programming languages and numerical libraries in their work as was found in a recent survey (Edgar[1]). Consequently, there is an ongoing debate whether it is still justified to teach programming languages and how many student credit hours are allocated to this subject area. Programming languages are often taught by computer scientists not by engineers, and this is usually before the students encounter any engineering problems that are complex enough to require programming. This may lead to low motivation among engineering students to study programming.

2.2 Numerical problem solving and visualization

With the introduction of the user-friendly mathematical software packages, numerical solution techniques have gradually replaced analytical and graphical solution techniques in engineering education and practice. A textbook demonstrating the use of POLYMATH for numerical solution of problems in various, required, chemical engineering courses was published by Cutlip and Shacham[3]Currently there are many textbooks which rely on one or more mathematical software packages to numerically solve the presented problems. (See Edgar[1] for a list of such textbooks.).

Most of the problems that are included in the textbooks mentioned in the previous section can be characterized as *Single-Model, Single-Algorithm (***SMSA***)* A typical examples of **SMSA** type problems include the following is the steady-state operation of a tubular reactor where the model consists of a system of ordinary differential equations and explicit algebraic equations. One numerical integration algorithm (such as the 4th order Runge-Kutta) can be used to solve this model.

SMSA types of problems can be solved efficiently by several software packages, as was shown by Shacham and Cutlip[4]. However, even in undergraduate education, there is often a need to solve more complex problems that can be characterized as: *Multiple-Model, Single-Algorithm* (**MMSA**)*, Single-Model, Multiple-Algorithm* (**SMMA**) and *Multiple-Model, Multiple Algorithm* (**MMMA**) problems.

A typical example of a **MMSA** problem is the cyclic operation of a semi-batch bioreactor (Cutlip and Shacham[5]). The three modes of operation of the bioreactor (initialization, processing, and harvesting) are represented by different models comprised of simultaneous ordinary differential equations and explicit algebraic equations. All models can be solved by one numerical integration algorithm (such as the 4^{th} order Runge-Kutta). An example of a **SMMA** problem is the problem of parameter estimation in dynamic systems. In this case there is a model comprising of ordinary differential equations and explicit algebraic equations, with parameters that should be fitted to experimental data using nonlinear regression techniques. One solution option is to solve this system by integrating the differential equations with specified parameter values in an internal loop, and then minimizing the sum of squares of the difference between the calculated and the experimental values using an optimization algorithm in an outer loop. A typical example of an **MMMA** problem is the optimization of the semi-batch bioreactor, described earlier, with respect to some of its operational parameters.

The use of visualization, based on graphical solution techniques (such as the McCabe-Thiele diagram) for pedagogical purposes, has seen renewed application recently (Joo and Choudhary[6]. The creation of the diagrams needed for visualization can also be characterized as a complex problem that cannot be easily tackled with some of the mathematical software packages.

2.2 Large scale simulation

The most commonly used large-scale simulation software packages in undergraduate education include process simulators (Dahm et al.[7]), computational fluid dynamic (CFD) packages (Edgar[1]), virtual laboratory experiments (Wiesner and Lan[8]), and molecular modeling related programs (Edgar[1]). There are many potential applications for large-scale simulation programs that cannot be carried out by the general purpose mathematical software packages. Such applications include visualization of flow fields using CFD software, investigation of cause-effect relationships among operational parameters of a particular process, and the simulation of virtual laboratory experiments. Thus students are able to experience complex systems that may be difficult to attain through direct contact with the equipment itself.

3. Computing in industry

Surveys concerning the use of computer based tools in the industry were carried out recently by the CACHE Corporation (Edgar[1]) and by Cameron and Ingram[9]. The CACHE survey found that practically all the engineers and scientist in the industry (98 %) use spreadsheet programs (the most popular being Excel). Spreadsheet programs are used mainly for data analysis (88%), numerical analysis (47%) material balances (25%) and economic studies (24%). The survey indicated a considerable level of use of database management systems (65%). The level of use of other software tools among the general population of industrial practitioners is much lower, and most of it probably represents their use by the modelers group.

Cameron and Ingram[9] list the tools used by a particular group of the industrial users: "the modelers group" according to the extent of their use, as follows: Excel, flow sheeting packages, MATLAB, direct coding, CFD, hybrid modeling, and simulation with optimization programs. Additional tools such as molecular simulation, expert systems, and programs for risk analysis are used to a lower extent.

4. An introductory course for modeling and computation for chemical engineers

A review of the state of the art of computing in the academia and in the industry has demonstrated that incorporating the most necessary computing tools into the undergraduate curriculum represents a major challenge. In order to meet this challenge, it is necessary to provide the students the ability to solve problems of various levels of complexity in a single course.

One possible approach utilizes the software packages POLYMATH, Excel and MATLAB in such a course (Shacham[10], Cutlip and Shacham[5]). POLYMATH is an easy to learn and user-friendly problem-solving tool which can be employed in most undergraduate and graduate courses for solving **SMSA** problems and carrying out various types of regressions with statistical analysis. Excel is included in the introductory computing course because of its widespread use in the industry and suitability for carrying out parametric studies. MATLAB can be used as a means to learn and apply programming, carry out symbolic manipulations, solve various types of **MMMA** problems, and provide 2D and 3D visualizations.

A new feature included in the POLYMATH package enables the automated export of POLYMATH models to Excel and MATLAB. This capability can significantly shorten the learning curve for these programs. After defining and checking a particular model with POLYMATH, it can be exported to Excel as a complete worksheet, or to MATLAB as a function m-file. The exported models facilitate the introduction and use of the other software tools and help to remove the "un-forgiveness" barrier which prevents many students from attaining programming proficiency. Advanced programming and capabilities are provided with MATLAB.

A new introductory computing course replaced the FORTRAN programming course at the Ben-Gurion University in 2003. The course is either given to freshman chemical engineering students who have already taken an introductory Material and Energy Balance course, or the two courses are given concurrently. Realistic problems, which are simple enough to be understood at the early stage of the ChE studies, are extensively utilized. Many of the examples that are being used in the introductory course are presented in Chapters 2, 4 and 5 of the Cutlip and Shacham[5] textbook.

5. The proposed framework for incorporating multiscale modeling in the undergraduate curriculum

The main ingredient of the proposed framework is a basic computational course, which replaces the traditional computer programming course. Further enhancement of the knowledge acquired in the introductory course can be achieved by using the packages in other modeling and computational-oriented courses. These include courses in numerical methods, optimization, process simulation, dynamics and control, and advanced math. The software packages POLYMATH, Excel and MATLAB can be used throughout the curriculum for solving problems of various complexities (**SMSA**, **MMSA**, etc) and for correlation of data via multiple linear, polynomial, and nonlinear regressions.

In the first chemical engineering course (Material and Energy Balances), it is desirable to introduce physical property databases (such as NIST and DIPPR) in order to encourage the use of reliable data sources, considerations of the units associated with the various properties, and awareness of the experimental errors associated with their values. Process simulation programs (such as HYSIS or Aspen) can be used for mini-projects as recommended by Dahm et al.[7].

Additional software packages (such as commercial dynamic and steady-state process simulation, optimization, design, CFD, and molecular simulation) as well as instructor-prepared demonstration programs can be introduced into the various courses of the ChE.. In these courses, the objectives are to use the programs for numerical, model-based and virtual experimentation, analysis of cause-effect relationships in complex systems, and visualization of challenging concepts. The packages can be introduced to the students in a time-efficient and effective way while simultaneously enabling a better understanding of the specific course material.

References

[1] T. F. Edgar, Chemical Engineering Education, 40 (2006) 231-238
[2] J. D. Seader, Computers and Chemical Engineering 13(1989) 377
[3] M. B. Cutlip, M. Shacham, Problem solving in chemical engineering with numerical methods Prentice-Hall , Upper Saddle River, New-Jersey, 1999
[4] M. Shacham, M.B. Cutlip, Computers in Education Journal, IX (1999) 9-15
[5] M. B. Cutlip, M. Shacham, Problem solving in chemical and biochemical engineering with POLYMATH, Excel and MATLAB. 2nd Ed., Prentice-Hall, Upper Saddle River, New-Jersey, 2008
[6] Y. L. Joo, D. Choudhary, Chemical Engineering Education 40 (2006)., 313-322.
[7] K.D. Dahm, R. P. Hesketh, M. J. Savelski, Chemical Engineering Education, 36(2002) 192
[8] T. F. Wiesner, W. Lan, Journal of Engineering Education, 93(2004) 195.
[9] I. T. Cameron, G. D. Ingham, Computers and Chemical Engineering, 32 (2008) 420-438.
[10] M. Shacham, Computer Applications in Engineering Education, 13(2005) 137.