CHAPTER 1

Problem Solving with Mathematical Software Packages

1.1 Efficient Problem Solving—The Objective of This Book

As an engineering student or professional, you are almost always involved in numerical problem solving on a personal computer. The objective of this book is to enable you to solve numerical problems that you may encounter in your student or professional career in a most effective and efficient manner. The tools that are typically used for engineering or technical problem solving are mathematical software packages that execute on personal computers on the desktop. In order to solve your problems most efficiently and accurately, you must be able to select the appropriate software package for the particular problem at hand. Then you must also be proficient in using your selected software tool.

In order to help you achieve these objectives, this book provides a wide variety of problems from different areas of chemical, biochemical, and related engineering and scientific disciplines. For some of these problems, the complete solution process is demonstrated. For some problems, partial solutions or hints for the solution are provided. Other problems are left as exercises for you to solve.

Most of the chapters of the book are organized by chemical and biochemical engineering subject areas. The various chapters contain between five and twenty-eight problems that represent many of the problem types that require a computer solution in a particular subject area. All problems presented in the book have the same general format for your convenience. The concise problem topic is first followed by a listing of the engineering or scientific concepts demonstrated by the problem. Then the numerical methods utilized in the solution are indicated just before the detailed problem statement is presented. Typically a particular problem presents all of the detailed equations that are necessary for solution, including the appropriate units in a variety of systems, with Système International d'Unités (SI) being the most commonly used. Physical properties are either given directly in the problem or in the appendices. Complete and partial solutions are provided to many of the problems. These solutions will help you learn to formulate and then to solve the unsolved problems in the book as well as the problems that you will face in your student and/or professional career.

Three widely used mathematical software packages are used in this book for solving the various problems: POLYMATH,* Excel,† and MATLAB.‡ Each of these packages has specific advantages that make it the most appropriate for solving a particular problem. In some cases, a combined use of several packages is most desirable. These mathematical software packages that solve the problems utilize what are called "numerical methods." This book presents the fundamental and practical approaches to setting up problems that can then be solved by mathematical software that utilizes numerical methods. It also gives much practical information for problem solving. The details of the numerical methods are beyond the scope of this book, and reference can be made to textbook by Constantinides and Mostoufi.[1] More advanced and extensive treatment of numerical methods can be found in the book of Press et al.[2]

The first step in solving a problem using mathematical software is to prepare a mathematical model of the problem. It is assumed that you have already learned (or you will learn) how to prepare the model of a problem from in particular subject area (such as thermodynamics, fluid mechanics, or biochemical engineering). A general approach advocated in this book is to start with a very simple model and then to make the model more complex as necessary to describe the problem. Engineering and scientific fundamentals are important in model building. The first step in the solution process is to characterize the problem according to the type of the mathematical model that is formulated: a system of algebraic equations or a system of ordinary differential equations, for example. When the problem is characterized in these terms, the software package that efficiently solves this type of problems can be utilized. Most of the later part of this chapter is devoted to learning how to characterize a problem in such mathematical terms.

In order to put the use of mathematical software packages for problem solving into proper perspective, it is important and interesting to review the history in which manual problem solving has been replaced by numerical problem solving.

1.2 From Manual Problem Solving to Use of Mathematical Software

The problem solving tools on the desktop that were used by engineers prior to the introduction of the handheld calculators (i.e., before 1970) are shown in Figure 1–1. Most calculations were carried out using the slide rule. This required carrying out each arithmetic operation separately and writing down the results of such operations. The highest precision of such calculations was to three decimal digits at most. If a calculational error was detected, then all the slide rule and arithmetic calculations had to be repeated from the point where the error occurred. The results of the calculations were typically typed, and hand-drawn

* POLYMATH is a product of Polymath Software (http://www.polymath-software.com).
† Excel is a trademark of Microsoft Corporation (http://www.microsoft.com).
‡ MATLAB is trademark of The Math Works, Inc. (http://www.mathworks.com).

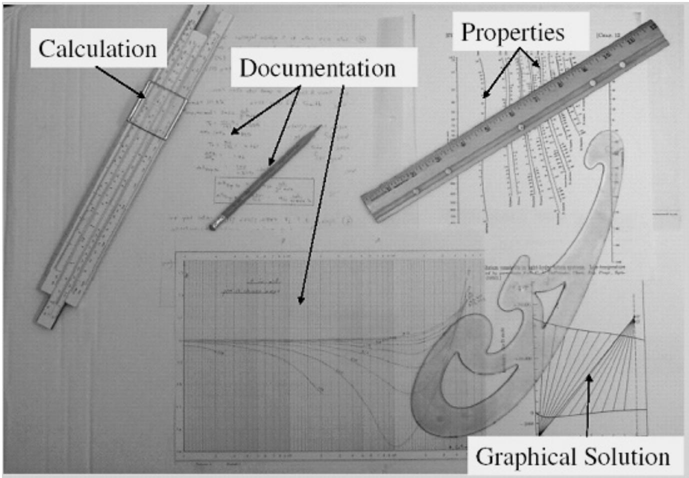

Figure 1–1 The Engineer's Problem Solving Tools Prior to 1970

graphs were often prepared. Temperature- and/or composition-dependent thermodynamic and physical properties that were needed for problem solving were represented by graphs and nomographs. The values were read from a straight line passed by a ruler between two points. The highest precision of the values obtained using this technique was only two decimal digits. All in all, "manual" problem solving was a tedious, time-consuming, and error-prone process.

During the slide rule era, several techniques were developed that enabled solving realistic problems using the tools that were available at that time. Analytical (closed form) solutions to the problems were preferred over numerical solutions. However, in most cases, it was difficult or even impossible to find analytical solutions. In such cases, considerable effort was invested to manipulate the model equations of the problem to bring them into a solvable form. Often model simplifications were employed by neglecting terms of the equations which were considered less important. "Short-cut" solution techniques for some types of problems were also developed where a complex problem was replaced by a simple one that could be solved. Graphical solution techniques, such as the McCabe-Thiele and Ponchon-Savarit methods for distillation column design, were widely used.

After digital computers became available in the early 1960's, it became apparent that computers could be used for solving complex engineering problems. One of the first textbooks that addressed the subject of numerical solution of problems in chemical engineering was that by Lapidus.[3] The textbook by Carnahan, Luther and Wilkes[4] on numerical methods and the textbook by Henley and Rosen[5] on material and energy balances contain many example problems for numerical solution and associated mainframe computer programs (written in the FORTRAN programming language). Solution of an engineering problem using digital computers in this era included the following stages: (1) derive the model equations for the problem at hand, (2) find the appropriate numerical method (algorithm) to solve the model, (3) write and debug a computer language program (typically FORTRAN) to solve the problem using the selected algorithm, (4) validate the results and prepare documentation.

Problem solving using numerical methods with the early digital computers was a very tedious and time-consuming process. It required expertise in numerical methods and programming in order to carry out the 2nd and 3rd stages of the problem-solving process. Thus the computer use was justified for solving only large-scale problems from the 1960's through the mid 1980's.

Mathematical software packages started to appear in the 1980's after the introduction of the Apple and IBM personal computers. POLYMATH version 1.0, the software package which is extensively used in this book, was first published in 1984 for the IBM personal computer.

Introduction of mathematical software packages on mainframe and now personal computers has considerably changed the approach to problem solving. Figure 1–2 shows a flow diagram of the problem-solving process using such a package. The user is responsible for the preparation of the mathematical model (a complete set of equations) of the problem. In many cases the user will also need to provide data or correlations of physical properties of the compounds involved. The complete model and data set must be fed into the mathematical software package. It is also the user's responsibility to categorize the problem type. The problem category will determine the type of numerical algorithm to be used for the solution. This issue will be discussed in detail in the next section.

The mathematical software package will then solve the problem using the selected numerical technique. The results obtained together with the model definition can serve as partial or complete documentation of the problem and its solution.

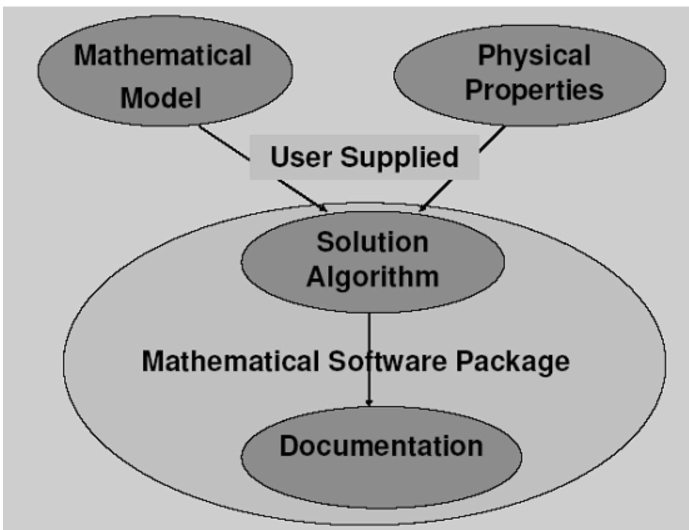

Figure 1–2 Problem Solving with Mathematical Software Packages

1.3 Categorizing Problems According to the Solution Technique Used

Mathematical software packages contain various tools for problem solving. In order to match the tool to the problem in hand, you should be able to categorize the problem according to the numerical method that should be used for its solution. The discussion in this section details the various categories for which representative examples are included in the book. Note that the study of the following categories (a) through (e) is highly recommended prior to using Chapters 7 through 14 of this book that are associated with particular subject areas. Categories (f) through (n) are advanced topics that should be reviewed prior to advanced problem solving.

(a) Consecutive Calculations

These calculations do not require the use of a special numerical technique. The model equations can be written one after another. On the left-hand side a variable name appears (the output variable), and the right-hand side contains a constant or an expression that may include constants and previously defined variables. Such equations are usually called "explicit" equations. A typical example for such a problem is the calculation of the pressure using the van der Waals equation of state.

$$
\begin{aligned}
R &= 0.08206 \\
T_c &= 304.2 \\
P_c &= 72.9 \\
T &= 350 \\
V &= 0.6 \\
a &= (24/64)((R^2 T_c^2)/P_c) \\
b &= (RT_c)/(8P_c) \\
P &= (RT)/(V-b) - a/V^2
\end{aligned}
\tag{1-1}
$$

The various aspects associated with the solution of this type of problem are described in detail in Problems 4.1 and 5.1 (Molar Volume and Compressibility from Redlich-Kwong Equation). In those completely solved problems, the advantages of the different software packages (POLYMATH, Excel, and MATLAB) in the various stages of the solution process are also demonstrated. To gain the most benefit from using this book, you should proceed now to study Problems 4.1 and 5.1 and then return to this point.

(b) System of Linear Algebraic Equations

A system of linear algebraic equations can be represented by the equation:

$$\mathbf{Ax} = \mathbf{b} \tag{1-2}$$

where **A** is an $n \times n$ matrix of coefficients, **x** is an $n \times 1$ vector of unknowns and **b** an $n \times 1$ vector of constants. Note that the number of equations is equal to the number of the unknowns. A detailed description of the various aspects of the solution of systems of linear equations is provided in Problem 2.4 (Steady-State Material Balances on a Separation Train).

(c) One Nonlinear (Implicit) Algebraic Equation

A single nonlinear equation can be written in the form

$$f(x) = 0 \tag{1-3}$$

where f is a function and x is the unknown. Additional explicit equations, such as those shown in Section (a), may also be included. Solved problems associated with the solution of one nonlinear equation are presented in Problems 2.1 (Molar Volume and Compressibility Factor from Van Der Waals Equation), 2.9 (Gas Volume Calculations using Various Equations of State), 2.10 (Bubble Point Calculation for an Ideal Binary Mixture), and 2.13 (Adiabatic Flame Temperature in Combustion). These problems should be reviewed before proceeding further. The use of the various software packages for solving single nonlinear equations is demonstrated in solved Problems 4.2 and 5.2 (Calculation of the Flow Rate in a Pipeline). Please study those solved problems as well.

(d) Multiple Linear and Polynomial Regressions

Given a set of data of measured (or observed) values of a dependent variable: y_i versus n independent variables x_{1i}, x_{2i}, ... x_{ni}, multiple linear regression attempts to find the "best" values of the parameters a_0, a_1, ...a_n for the equation

$$\hat{y}_i = a_0 + a_1 x_{1,i} + a_2 x_{2,i} + \ldots + a_n x_{n,i} \tag{1-4}$$

where $\hat{y}_i$ is the calculated value of the dependent variable at point i. The "best" parameters have values that minimize the squares of the errors

$$S = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{1-5}$$

where N is the number of available data points.

In polynomial regression, there is only one independent variable x, and Equation (1-4) becomes

$$\hat{y}_i = a_0 + a_1 x_i + a_2 x_i^2 + \ldots + a_n x_i^n \tag{1-6}$$

Multiple linear and polynomial regressions using POLYMATH are demonstrated in detail in solved Problems 3.3 (Correlation of Thermodynamic and Physical Properties of n-Propane) and 3.5 (Heat Transfer Correlations from Dimensional Analysis). The use of Excel and MATLAB for the same purpose is demonstrated respectively in Problems 4.4 and 5.4 (Correlation of the Physical Properties of Ethane). These examples should be studied before proceeding further.

(e) Systems of First-Order Ordinary Differential Equations (ODEs) – Initial Value Problems

A system of n simultaneous first-order ordinary differential equations can be written in the following (canonical) form

$$\begin{aligned}
\frac{dy_1}{dx} &= f_1(y_1, y_2, \ldots y_n, x) \\
\frac{dy_2}{dx} &= f_2(y_1, y_2, \ldots y_n, x) \\
&\vdots \\
\frac{dy_n}{dx} &= f_n(y_1, y_2, \ldots y_n, x)
\end{aligned} \tag{1-7}$$

where x is the independent variable and $y_1, y_2, \ldots y_n$ are dependent variables. To obtain a unique solution of n simultaneous first-order ODEs, it is necessary to specify n values of the dependent variables (or their derivatives) at specific values of the independent variable. If those values are specified at a common point, say x_0,

$$\begin{aligned}
y_1(x_0) &= y_{1,0} \\
y_2(x_0) &= y_{2,0} \\
&\vdots \\
y_n(x_0) &= y_{n,0}
\end{aligned} \tag{1-8}$$

then the problem is categorized as an initial value problem.

The solution of systems of first-order ODE initial value problems is demonstrated in Problems 2.14 (Unsteady-state Mixing in a Tank) and 2.16 (Heat Exchange in a Series of Tanks) where POLYMATH is used to obtain the solution. The use of Excel and MATLAB for systems of first-order ODEs is demonstrated respectively in Problems 4.3 and 5.3 (Adiabatic Operation of a Tubular Reactor for Cracking of Acetone).

(f) System of Nonlinear Algebraic Equations (NLEs)

A system of nonlinear algebraic equations is defined by

$$\mathbf{f}(\mathbf{x}) = \mathbf{0} \tag{1-9}$$

where $\mathbf{f}$ is an n vector of functions, and $\mathbf{x}$ is an n vector of unknowns. Note that the number of equations is equal to the number of the unknowns. Solved problems in the category of NLEs are Problems 8.11 (Flow Distribution in a Pipeline Network) and 6.6 (Expediting the Solution of Systems of Nonlinear Algebraic Equations). More advanced treatment of systems of nonlinear equations (obtained when solving a constrained minimization problem), is demonstrated along with the use of various software packages in Problems 4.5 and 5.5 (Complex Chemical Equilibrium by Gibbs Energy Minimization).

(g) Higher Order ODEs

Consider the n-th order ordinary differential equation

$$\frac{d^n z}{dx^n} = G\left(z, \frac{dz}{dx}, \frac{d^2 z}{dx^2}, \ldots \frac{d^{n-1} z}{dx^{n-1}}, x\right) \tag{1-10}$$

This equation can be transformed by a series of substitution to a system of n first-order equations (Equation (1-7)). Such a transformation is demonstrated in Problems 6.5 (Shooting Method for Solving Two-Point Boundary Value Problems) and 8.18 (Boundary Layer Flow of a Newtonian Fluid on a Flat Plate).

(h) Systems of First-Order ODEs—Boundary Value Problems

ODEs with boundary conditions specified at two (or more) points of the independent variable are classified as boundary value problems. Examples of such problems and demonstration of the solution techniques used can be found in Problems 6.4 (Iterative Solution of ODE Boundary Value Problem) and 6.5 (Shooting Method for Solving Two-Point Boundary Value Problems).

(i) Stiff Systems of First-Order ODEs

Systems of ODEs where the dependent variables change on various time (independent variable) scales which differ by many orders of magnitude are called "Stiff" systems. The characterization of stiff systems and the special techniques that are used for solving such systems are described in detail in Problem 6.2 (Solution of Stiff Ordinary Differential Equations).

(j) Differential-Algebraic System of Equations (DAEs)

The system defined by the equations:

$$\begin{aligned} \frac{d\mathbf{y}}{dx} &= \mathbf{f}(\mathbf{y}, \mathbf{z}, x) \\ \mathbf{g}(\mathbf{y}, \mathbf{z}) &= 0 \end{aligned} \tag{1-11}$$

with the initial conditions $\mathbf{y}(\mathbf{x}_0) = \mathbf{y}_0$ is called a system of differential-algebraic equations. Demonstration of one particular technique for solving DAEs can be found in Problem 6.7 (Solving Differential Algebraic Equations – DAEs).

(k) Partial Differential Equations (PDEs)

Partial differential equations where there are several independent variables have a typical general form:

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \tag{1-12}$$

A problem involving PDEs requires specification of initial values and boundary conditions. The use of the “Method of Lines” for solving PDEs is demonstrated in Problems 6.8 (Method of Lines for Partial Differential Equations) and 9.14 (Unsteady-State Conduction in Two Dimensions).

(l) Nonlinear Regression

In nonlinear regression, a nonlinear function g

$$\hat{y}_i = g(a_0, a_1 \ldots a_n, x_{1,i}, x_{2,i}, \ldots x_{n,i}) \tag{1-13}$$

is used to model the data by finding the values of the parameters a_0, a_1 … a_n that minimize the squares of the errors shown in Equation (1-5). Detailed description of the nonlinear regression problem and the method of solution using POLYMATH can be found in Problem 3.1 (Estimation of Antoine Equation Parameters using Nonlinear Regression). The use of Excel and MATLAB for nonlinear regression is demonstrated in respective Problems 4.4 and 5.4 (Correlation of the Physical Properties of Ethane).

(m) Parameter Estimation in Dynamic Systems

This problem is similar to the nonlinear regression problem except that there is no closed form expression for $\hat{y}_i$, but the squares of the errors function to be minimized must be calculated by solving the system of first order ODEs

$$\frac{d\hat{\mathbf{y}}}{dt} = \mathbf{f}\left(a_0, a_1 \ldots a_n, x_1, x_2, \ldots x_n, t\right) \tag{1-14}$$

Problem 6.9 (Estimating Model Parameters Involving ODEs using Fermentation Data) describes in detail a parameter identifications problem and demonstrates its solution using POLYMATH and MATLAB.

(n) Nonlinear Programming (Optimization) with Equity Constraints

The nonlinear programming problem with equity constrains is defined by:

$$\begin{aligned} &\textit{Minimize} \quad f(\mathbf{x}) \\ &\textit{Subject to} \quad \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned} \tag{1-15}$$

where f is a function, $\mathbf{x}$ is an n-vector of variables and $\mathbf{h}$ is an m-vector ($m<n$) of functions.

Problems 4.5 and 5.5 (Complex Chemical Equilibrium by Gibbs Energy Minimization) demonstrate several techniques for solving nonlinear optimization problems with POLYMATH, Excel, and MATLAB.

1.4 EFFECTIVE USE OF THIS BOOK

Readers who wish to begin solving realistic problems with mathematical software packages are recommended to initially study the items listed in categories (a) through (e) in the previous section. In addition, different subject areas will require completion of some of the advanced topics listed in categories (f) through (n) of the previous section. Table 1–1 shows the advanced topics associated with the various subjects covered in Chapters 7 to 14 of this book. For example, you will be able to achieve effective solutions for the problems associated with Chapter 8 (Fluid Mechanics) if you initially study the categories (f) System of Nonlinear Algebraic Equations (NLEs), (g) Higher Order ODEs, (h) Systems of First-Order ODEs—Boundary Value Problems, and (k) Partial Differential Equations (PDEs). It is recommended that the advanced topics listed in Table 1–1 be studied before you begin to work on a chapter related to a particular subject area.

Table 1–1 Advanced Topic Prerequisites for Chapters 7 through 14

Chapter No.	Subject Area	Advanced Topics Required
7	Thermodynamics	(f)
8	Fluid Mechanics	(f), (g), (h), and (k)
9	Heat Transfer	(g), (h), and (k)
10	Mass Transfer	(g), (h), and (k)
11	Chemical Reaction Engineering	(f), (g), (h), (l), and (m)
12	Phase Equilibria and Distillation	(f) and (j)
13	Process Dynamics and Control	(f), (g), (h), and (i)
14	Biochemical Engineering	(f), (g), (h), (i), (l), and (n)

Several of the advanced topics are typically considered in the advanced subject areas of "Numerical Methods," "Advanced Mathematics," and "Optimization." Using the problems pertinent to a particular topic in these advanced subject areas can be very beneficial in the learning process. The problems associated with the various topics are shown in Table 1–2, where solved or partially solved problem numbers are shown in bold numerals

Some problems may require special solution techniques. Solved problems that demonstrate some special techniques are listed in Table 1-3. If a problem matches one or more of the categories in this table, then an examination of the similar solved problem can be very helpful.

Table 1–2 List of Problems Associated with Advanced Topics[a]

	Topic	Pertinent Problem Numbers*
(f)	System of Nonlinear Algebraic Equations (NLEs)	**4.5**, **5.5**, 6.6, 7.13, 7.14, 8.9, **8.10**, 8.12, 8.13, 10.2,**12.1**, 12.2, **12.3**, 12.4, 12.5, **12.8**, 12.9, 14.8, 14.11, 14.15
(g)	Higher Order ODEs	**6.5**. **8.16**, 8.18, **9.2**, **9.5**, 10.11, **13.1**, 13.2, **13.5**, **13.7**, **13.12**, **14.5**
(h)	Systems of First-Order ODEs—Boundary Value Problems	**6.4**,**6.5**,**8.1**,8.2,8.3,8.4,8.18, 9.1, 9.2, 9.6, 9.7, **10.1**, **10.3**, **10.5**, 10.6,10.7, **10.8**, 10.9, 10.10, 10.12, **14.5**
(i)	Stiff Systems of First-Order ODEs	**6.1**, 6.2
(j)	Differential-Algebraic System of Equations (DAEs)	**6.7**, **12.10**, 12.11
(k)	Partial Differential Equations (PDEs)	**6.8**, 8.17, 9.12, 9.13, **9.14**, **10.13**, **10.14**, **10.15**
(l)	Nonlinear Regression	**3.1**, **3.2**, **3.3**, 3.4, **4.4**, **5.4**, 11.7, **13.4**, 13.8, **14.2**, 14.7, 14.8, 14.13, 14.14, 14.15
(m)	Parameter Estimation in Dynamic Systems	**6.9**
(n)	Nonlinear Programming (Optimization)	**4.5**, **5.5**, **6.9**, 14.4, 14.6, 14.11, **14.16**

[a]Solved and partially solved problems are indicated in bold.

Table 1–3 List of Problems Associated with Special Problem Solving Techniques

Differential and Algebraic Equations	**Pertinent Problem Number(s)**
Plotting Solution Trajectory for an Algebraic Equation Using the ODE Solver	7.1, 7.5
Using the l'Hôpital's Rule for Undefined Functions at the Beginning or End Point of Integration Interval	7.11, 7.12
Using "If" Statement to Avoid Division by Zero	8.1
Switching Variables On and Off during Integration	8.16
Retaining a Value when a Condition Is Satisfied	8.16
Generation of Error Function	8.17
Functions Undefined at the Initial Point	9.2, 9.5
Using "If" Statement to Match Different Equations to the Same Variable	9.2, 10.4, 8.6
Ill-Conditioned Systems	6.3
Conversion of a System of Nonlinear Equations into a Single Equation	6.3
Selection of Initial Estimates for Nonlinear Equations	6.6
Modification of Strongly Nonlinear Equations for Easier Solution	6.6
Conversion of a Nonlinear Algebraic Equation to a Differential Equation	7.1, 7.5
Data Modeling and Analysis	
Using Residual Plot for Data Analysis	2.5, 3.1, 3.3, 3.5, 3.8, 3.14
Using Confidence Intervals for Checking Significance of Parameters	2.5, 3.1, 3.3, 3.8
Transformation of Nonlinear Models for Linear Representations	2.5, 3.3, 3.5, 3.8
Checking Linear Dependency among Independent Variables	3.11

1.5 SOFTWARE USAGE WITH THIS BOOK

The problems presented within this text can be solved with a variety of mathematical software packages. The problems along with the appendices provide all the necessary equations and parameters for obtaining problem solutions. However, POLYMATH is extensively utilized to demonstrate problem solutions throughout the book because it is extremely easy to use, and because the equations are entered in basically the same mathematical form as they are written.

Recent enhancements to POLYMATH have enabled the content of this book to be expanded to more directly support problem solving with Excel and MATLAB. This is due to the ability of POLYMATH to export a problem solution to a working spreadsheet in Excel or the necessary m-file of the problem for MATLAB.

The export process to Excel results in a complete working spreadsheet with the same notation and logic as utilized in the POLYMATH problem solution code. In addition, all the intrinsic functions (such as log, exp, sin, etc.) are automatically converted. POLYMATH also provides an Excel Add-In which allows the solution of systems of ordinary differential equations within Excel. This is separate software which operates in Excel and enables the same solution algorithms available in POLYMATH to be utilized within Excel. Complete details and an introduction to problem solving in Excel are given in Chapter 4.

For MATLAB, the m-file for a particular POLYMATH program solution is automatically generated with translation of the program logic and the needed intrinsic functions. Also, the generated statements in the m-file are automatically ordered as required by MATLAB. This use of MATLAB and the use of the generation of m-files from POLYMATH are discussed in Chapter 5.

Within this book, the problems are discussed with reference to important and necessary equations. The solutions are presented using the POLYMATH coding as this provides a clear representation of the problem formulation along with the mathematical equations and logic necessary for problem solution. The Excel and MATLAB solutions can easily be achieved through the use of the POLYMATH program via the automated export capability. All of the completely worked or partially worked problems in the book are available in files for all three packages – POLYMATH, Excel, and MATLAB. Access to these files is provided from the book web site as discussed in the next section. You can obtain an inexpensive educational version of the latest POLYMATH software from the book's web site, **www.problemsolvingbook.com**.

Thus you have a choice of the software with which you wish to generate problem solutions while using this book. POLYMATH is highly recommended as it is used throughout the book and is widely accepted as the most convenient mathematical software for a student or a novice to learn and use. You are encouraged to learn to solve problems with POLYMATH, Excel, and MATLAB as this book will highlight and utilize some of the particular capabilities that each package enables.

1.6 WEB-BASED RESOURCES FOR THIS BOOK

A special web site is dedicated to the continuing support of this book

www.problemsolvingbook.com

and this is identified throughout the book with the icon on the left. This special web site provides readers with the following materials:

1. Solution files for the worked and partially worked problems for POLYMATH, Excel, and MATLAB
2. Access to the latest Educational Version of POLYMATH at a special reduced price that is available only to book owners for educational use
3. MATLAB templates for general problem solving
4. Additional problems as they become available
5. Corrections to the printed book
6. Special resources for students
7. Special resources for instructors who are using this book
   Note: Instructors should obtain more details from one of the authors via e-mail from their home institution to

michael.cutlip@uconn.edu

shacham@bgu.ac.il

GENERAL REFERENCES

1. Constantinides, A., and Mostoufi, N., *Numerical Methods for Chemical Engineers with MATLAB Applications*, 1st ed., Upper Saddle River, NJ: Prentice Hall, 1999.
2. Press, W. H., Teukolsky, S, A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C++: The Art of Scientific Computing*, 2nd ed., Cambridge, England: Cambridge University Press, 2002.
3. Lapidus, L., *Digital Computation for Chemical Engineers*, New York: McGraw-Hill, 1962.
4. Carnahan, B., Luther, H. A. and Wilkes, J. O., *Applied Numerical Methods*, New York: Wiley, 1969.
5. Henley, E. J., and E. M. Rosen, *Material and Energy Balance Computation*, New York: Wiley, 1969.