Prologue

This book introduces chemical engineers to numerical methods and develops their problem-solving skills using the electronic spreadsheet software Excel with the programming language VBA. Scientists and engineers from other disciplines will find the coverage of computational methods applicable to a variety of problems not specific to chemical engineering. Note the top three engineering work activities: design, computer applications, and management (Burton 1998). According to a CACHE survey of chemical engineering graduates (Edgar 2003):

- 100% use a computer every day.
- 98% use Excel or comparable spreadsheet programs (88% for data analysis; 47% for numerical analysis, 25% for material and energy balances, 24% for economic studies).
- 83% said computing enhanced their problem solving skills and wanted more exposure to computational methods.
- 78% indicated that a programming language should be required at the undergraduate level.
- 75% characterized their work as “technical” with their highest priorities being research and development, plant/process support, and process design and analysis.
- 73% responded that their employers expected them to be competent in a programming language (38% were required to write a computer program.)

Excel has become the de facto standard computational tool for engineering or scientific calculations across many fields. Our engineering colleagues use Excel for a variety of engineering and management tasks. The programming language Visual Basic for Applications (VBA, the Basic programming language in MS Office applications) further enhances Excel’s capabilities. We recommend this combination of tools to begin developing the computing skills valued by the practicing chemical engineers who responded to the surveys. Consequently, this book has four primary goals to help students:

1. Learn practical numerical methods for solving engineering problems,
2. Become proficient using Excel for engineering problem solving,
3. Gain skills for enhancing the capabilities of Excel with VBA user-defined functions and macros, and
4. Develop good habits of documenting solutions for archiving and collaboration.

“For the things we have to learn before we can do them, we learn by doing them.” – Aristotle

Learning numerical methods for engineering is a bit like the paradox of which came first, the chicken or the egg. A strong background in engineering science and design is important to apply many numerical techniques; yet many basic problems in modern engineering courses require the use of numerical methods to arrive at a working solution. This book intends to resolve this issue by introducing essential numerical techniques required in a typical chemical engineering curriculum, accompanied by a variety of examples from the discipline.

“The student needs to develop an understanding, however partial and imperfect, by descriptions rather than definitions, by typical examples rather than grandiose theorems.” – Gian-Carlo Rota

Complicated theories of numerical analysis were intentionally sidestepped in favor of concentrating on basic derivations and applications of numerical techniques … risking the “howl of the Boeotians.”¹ In the author’s experience, students begin using these practical computing skills right away in their technical courses and research. Frequently, students returning to campus after completing an engineering internship report that their ability to use Excel spreadsheets and VBA macros for modeling, data collection, and analysis was their most practical skill transferred from college to the workplace.²

“Never start teaching or research in a new field of applied mathematics from general concepts, statements, theories, and theorems. Consider some instructive examples and the general theory will come and be cast naturally.” – G.I Barenblatt

The coverage of topics in this book assumes readers have completed, or are concurrently enrolled in, mathematics courses covering topics of calculus, and differential equations. The examples assume some experience with introductory physics and chemistry. Lower division students need not fully understand the science or engineering behind these examples in order to learn the application of the numerical technique presented. However, they should be able to recognize the different mathematical formulations of the problems, e.g., an integral or differential equation that requires integration limits, initial conditions, or boundary conditions. We draw a few examples from transport phenomena and engineering unit operations. A few examples involve concepts from chemical reaction engineering, a course typically taken late in the curriculum, with concepts that use simple principles of

¹ Gauss (1777-1855) in an 1829 letter to a confident, “… for I fear the howl of the Boeotians if I speak my opinion aloud.”
² At the University of Minnesota Duluth, we teach this course to our students in their sophomore and junior years.
dynamic material and energy balances. If needed, the appendix to Chapter 1 summarizes basic reaction engineering principles used throughout the book.

Excel serves as an ideal computing environment for teaching and learning fundamental concepts of numerical methods. As previously noted, Excel comes equipped with VBA, a versatile, object-oriented programming language used to implement the algorithms presented throughout the text and enhances Excel as an engineering tool. Most VBA programs in this book use double precision calculations (note that, by default, VBA uses single precision). We may modify the macros and example files for our particular needs. Take care to use appropriate VBA programming syntax when modifying these programs for personal use. The appendices to Chapter 14 contain tables listing the macros that accompany the textbook. Although several “canned” software applications are commercially available for solving problems numerically3, we must avoid treating them like a “black box” for the simple reason that:

“You don’t want a fifty-dollar hair cut on a fifty-cent head.” — Garrison Keillor

We use commercial numerical software applications as computational tools, not necessarily teaching tools. They rarely illustrate the numerical algorithms through logically programmed steps required by this teaching text (Kuku and Karamani 2011). The combination of Excel and VBA forms a powerful platform for learning and implementing numerical methods (Coronell and Hariri 2008). The associated software help files are indispensable. Excel and VBA help files include examples of the syntax, and some information about the mathematics and statistics. We encourage students to learn to use the various versions of commercial computational software only after they have mastered the underlying fundamentals, and have a feel for when different methods work or fail. Otherwise, we risk overreliance on the software tools and miss an opportunity to develop our critical thinking skills. Without an understanding of the underlying numerical methods, we may become frustrated when the software does not work in the way we anticipate.

“If the only tool you have is a hammer, every problem begins to look like a nail.” — Abraham Maslow

Furthermore, process simulators incorporate several of the basic numerical methods presented in this book, as well as advanced techniques not covered here.4 Often, the user is required to supply initial conditions or guesses to “jumpstart” the solution. The users need to be familiar with the different solution methods in order to make the proper specifications and coax a solution from the software. Perhaps even more important is having the background to find alternative routes to a solution when a canned program fails.

By design, the arrangement of the coverage of topics follows a specific order that builds on previous topics using a spiral-learning model where the learner revisits a topic at increasing levels of breadth and depth. This book has five principal parts:

I. Introduction to numerical methods and tools for problem solving
   a. Mathematical modeling
   b. Excel for Engineers
   c. VBA for enhancing Excel with custom programming
II. Numerical methods for solving algebraic equations
   a. Linear equations
   b. Taylor series function approximations transition from linear to nonlinear problems
   c. Nonlinear equations
   d. Optimization
III. Numerical methods for working with data
   a. Uncertainty analysis
   b. Regression and data modeling for interpretation and functionalizing noisy experimental results
   c. Interpolation of smooth data for functionalizing experimental results
IV. Numerical methods for engineering mathematics: Calculus and Differential Equations
   a. Integral equations or numerical quadrature
   b. Initial-value problems involving first order differential equations.
   c. Initial-Boundary-value problems involving ordinary and partial differential equations
V. Summary review of numerical methods and software tools

We first introduce modeling concepts to reinforce the need for numerical methods, as well as establish the pattern for problem solving used for many of the example problems. Introductory chapters present practical features of Excel and VBA for engineering problem solving. We then review methods of solving systems of linear equations followed by a transition chapter on the Taylor series, which serves as the foundation for numerical methods presented in the remainder of the book. Derivative approximations derived from Taylor series function approximations are useful for accelerating the search for roots to nonlinear functions. The same procedures carry over into optimization and uncertainty analysis. We positioned uncertainty analysis as the centerpiece of the text to stress the importance of critical thinking about the reliability of our numerical results. Least-squares regression naturally follows optimization as a special application. We use regression techniques to model, smooth, and interpolate data. A separate chapter covers interpolation of smooth data. The last few chapters of the book present numerical solutions to integrals (which are based on interpolating functions) and differential equations that require methods previously introduced for solving systems of equations, derivative approximations, and solutions to nonlinear functions, regression, and interpolation. The final chapter includes appendices with summary tables of useful Excel and VBA functions for implementing numerical methods.

---

3 e.g., Mathcad, Mathematica, Maple, Matlab, Polymath, etc.
4 e.g., UNISIM, Aspen Plus, ChemCAD, HYSYS, Pro II, etc.
With our modern access to vast amounts of information on the internet, encyclopedic coverage of numerical analysis is not necessary in a textbook such as this. Rather, we introduce practical tools and methods to develop the basic concepts in numerical techniques (such as requirements for initial guesses to start iterative solutions), Taylor series analysis, derivative approximations, Monte Carlo techniques, etc. We also include useful tools that students and practicing engineers can use immediately in class or on the job. The book includes several practical, yet powerful VBA macros and functions and provides a foundation of essential numerical concepts that are common to advanced numerical methods.

"Let no one be deluded that knowledge of the path can substitute for putting one foot in front of the other." - M.C. Richards

We strongly recommend working through the examples to get the most out of the book. The examples in the textbook use the computer spreadsheet application Microsoft Windows Excel® version 2007 or later and Visual Basic for Applications (VBA). The Excel example files and VBA function and macro files are available from the author’s web site: www.d.umn.edu/~rda

What’s New in the Third Edition?

Much of the text was revised for clarity and conciseness. Several additional numerical methods and associated macros with examples were added in order to further develop the versatility of the PNM3Suite and expand the coverage of Excel with VBA as a tool for solving scientific and engineering problems. These include:

- Several updates to current macros to minimize user error and improve computational efficiency.
- Color formatting of cells used by some macros to indicate un converged solutions.
- Introduction to the Microsoft Equation object for placing images of equations on the worksheet.
- Added descriptions to user-defined functions and sub procedures for selection from menus.
- Simple bubble sorting macros available in the PNM3Suite.
- Macros to create \( y \) vs \( x \) scatter plots of data following the recommended graphing guidelines.
- User-defined functions that evaluate worksheet formulas now accept named cells in formulas or variables.
- Example of VBA driven animation of function plots.
- Macro for solving linear systems of equations by Gauss elimination with maximum column pivoting and one error improvement step.
- Pivoting and scaling for Gauss Elimination and Crout’s method of \( LU \) decomposition to ensure non-zero terms on the diagonal of a coefficient matrix for linear equations.
- User-form for selecting either Crout reduction of Gaussian elimination to solve a linear system of equations.
- Macros for calculating eigenvalues and eigenvectors of a real, square matrix for linear homogeneous systems by the power, Jacobi, and interpolation methods.
- User-defined array function for solving a linear system of equations arranged in matrix format, which also checks for consistent arrays and the determinant using Excel worksheet functions.
- Macro for solving implicit equations by fixed-point or Gauss-Seidel iteration with relaxation.
- Accurate finite difference approximations of higher order derivatives with Richardson’s extrapolation using Ridders’ algorithm.

"If I have seen further it is only by standing on the shoulders of Giants." – Isaac Newton
• Bisection macro for root finding in a single equation.
• User-defined function for finding the root of a formula on an Excel worksheet by the regula falsi, secant, and quasi-Newton methods.
• User-defined function for finding a root to a nonlinear equation by rational Padé interpolation.
• Additional options for Gauss-Newton, Levenberg-Marquardt, and Powell’s methods on the user-form ROOTS_UsrFrm for finding roots to systems of nonlinear algebraic equations.
• Firefly Algorithm and macros for multivariable, multimodal optimization.
• Example of the Law of Propagation of Uncertainty for a function involving a product/quotient.
• Switched the order of independent and dependent ranges in arguments for user-defined functions to maintain consistency with Excel worksheet functions, such as TREND.
• Macros for uncertainty and regression add comments to output cells to define terms.
• Sub procedure to implement nonlinear rational least-squares regression.
• User-defined function for Lagrange polynomial interpolation.
• User-defined function with the Bulirsch-Stoer algorithm for rational interpolation.
• Hermite interpolating polynomials using information about the first derivative at the data points.
• Inverse polynomial interpolation.
• User-defined functions for two-dimensional (bivariate) interpolation by linear, Stineman, cubic spline, and constrained cubic spline methods.
• User-defined functions for integrating data sets by the composite trapezoidal rule and Simpson’s rule with Richard-son’s extrapolation where possible.
• Simple macros for demonstrating Gauss-Legendre and Gauss-Kronrod 10 and 15 point single integrals.
• Adaptive quadrature with Simpson’s 1/3 rule employing Richardson’s error estimates for interval size control.
• Introduction to quasi-linearization for solving finite-difference equations in two-point boundary-value problems.
• Optional arguments for the limits of integration in the user-defined array function for orthogonal collocation.
• Application of the method of orthogonal collocation to parabolic partial differential equations.
• User-form calling macros end with _UsrFrm to distinguish them from sub procedures with similar methods.
• User-defined function for expanding the capability of unit conversions in an Excel worksheet.
• User-defined function that returns a value for the ideal gas constant in various unit systems.

Acknowledgements
I am indebted to my long-time friends, colleagues, and mentors: Professor Owen Hanna at UCSB for his creative genius, Professor Keith Lodge at UMD for his helpful reviews and suggestions, and Ronald Visness, retired scientist with the Minnesota Department of Natural Resources, who inspired me to take my first steps on this project. Each has provided much support, encouragement, enthusiasm, as well as helpful feedback along the way. I am also thankful to my students for their insightful comments and suggestions. I am most grateful to my eternal companion Elsie and our children for their support of this project.

I dedicate this book to my mother the nurse and father the aerospace engineer, who started a typical conversation with the following exchange:

Juanita: “It’s not rocket science!”
Sterling: “Well, actually …”

Duluth, Minnesota: 1 January 2015
Richard A. Davis

About the Author
Richard Davis is a Jean G. Blehert Distinguished Professor and Head of the Department of Chemical Engineering at the University of Minnesota Duluth. He received his B.S. degree in Chemical Engineering from Brigham Young University and Ph.D. degree in Chemical Engineering from the University of California Santa Barbara. He has over two decades experience teaching a variety of courses including computational methods, unit operations of momentum, heat and mass transfer, chemical reactor design, engineering economics, bioprocess engineering, green engineering, and separations. His current teaching and research interests include process modeling and simulation applied to mineral processing, energy conversion, pollution control, chemical process safety, and environmental management. He publishes regularly in both technical and educational journals.

Professor Davis serves as the Executive Secretary for the National Chemical Engineering Honor Society Omega Chi Epsilon, and is active in AIChE and ASEE. He is the academic adviser to the local engineering student chapters of Tau Beta Pi, Omega Chi Epsilon, and the Society for Mining, Metallurgy, and Exploration. He is the recipient of the University’s Outstanding Adviser and Jean G. Blehert Exceptional Teaching awards. His students have twice recognized him for his teaching and service to the department and profession with the Beta Zeta Chapter of Omega Chi Epsilon awards.