Numerical Methods with Chemical Engineering Applications

Designed primarily for undergraduates, but also graduates and practitioners, this textbook integrates numerical methods and programming with applications from chemical engineering. Combining mathematical rigor with an informal writing style, it thoroughly introduces the theory underlying numerical methods, its translation into MATLAB programs, and its use for solving realistic problems. Specific topics covered include accuracy, convergence and numerical stability, as well as stiffness and ill-conditioning. MATLAB codes are developed from scratch, and their implementation is explained in detail, all while assuming limited programming knowledge. All scripts employed are downloadable, and built-in MATLAB functions are discussed and contextualized. Numerous examples and homework problems – from simple questions to extended case studies – accompany the text, allowing students to develop a deep appreciation for the range of real chemical engineering problems that can be solved using numerical methods. This is the ideal resource for a single-semester course on numerical methods, as well as other chemical engineering courses taught over multiple semesters.

Kevin D. Dorfman is Professor of Chemical Engineering and Materials Science at the University of Minnesota. He is the recipient of the Colburn Award of the AIChE, the DARPA Young Investigator Award, a Packard Fellowship, the NSF CAREER Award, and both the Dreyfus New Faculty and Teacher-Scholar Awards. He has co-authored over 100 papers.

Prodromos Daoutidis is Professor and Executive Officer in the Department of Chemical Engineering and Materials Science at the University of Minnesota. He is the recipient of the NSF CAREER Award, the PSE Model Based Innovation Prize, the Ted Peterson Award of the AIChE, the McKnight Land Grant Professorship, the Ray D. Johnson/Mayon Plastics Professorship, and the Shell Chair. He has been a Humphrey Institute Policy Fellow. He has co-authored two research monographs and over 220 papers.
Numerical Methods with Chemical Engineering Applications

KEVIN D. DORFMAN AND PRODROMOS DAOUTIDIS
Contents

List of Illustrations xi
List of Tables xvi
List of Programs xvii
Preface xix
Acknowledgments xxiv

1 Mathematical Modeling and Structured Programming 1
1.1 Elements of a Mathematical Model 1
1.2 Some Analytically Solvable Chemical Engineering Models 4
  1.2.1 Vapor–Liquid Equilibrium 4
  1.2.2 Batch Reactor with a Single Reactant 6
  1.2.3 Batch Reactor with Three Species 8
  1.2.4 Continuous Stirred Tank Reactor 12
  1.2.5 Reaction–Diffusion in a Slab 13
  1.2.6 Unsteady Diffusion 15
1.3 Computer Math 20
  1.3.1 Three Key Operations 20
  1.3.2 Words and Round-Off Error 22
1.4 Structured Programming 23
  1.4.1 Learning to Program 24
  1.4.2 The for Loop 25
  1.4.3 The while Loop 28
  1.4.4 If-else Statements 30
  1.4.5 Modular Programming 33
1.5 Further Reading 35
Problems 35
Computer Problems 41

2 Linear Algebraic Equations 45
2.1 Definition of a Linear System of Algebraic Equations 45
2.2 Does a Solution Exist? 47
2.3 Conditioning 48
  2.3.1 Vector Spaces 49
Contents

2.3.2 Vector Norms 51
2.3.3 Matrix Norms 52
2.3.4 Condition Number 53
2.4 Cramer’s Rule 56
2.5 Gauss Elimination 58
  2.5.1 Algorithm 60
  2.5.2 Computational Cost 64
2.6 Pivoting 66
  2.6.1 A Simple Example 66
  2.6.2 Algorithm 68
2.7 Case Study: Multi-component Flash 72
  2.7.1 Problem Statement 72
  2.7.2 Solution 73
2.8 Calculation of the Determinant by Gauss Elimination 78
2.9 Banded Matrices 80
2.10 Simultaneous Gauss Elimination 82
2.11 LU Decomposition 83
  2.11.1 Doolittle’s Method 84
  2.11.2 Implementation 89
2.12 Case Study: Linear Process Flow Sheet 91
  2.12.1 Problem Statement 91
  2.12.2 Solution 92
2.13 Iterative Methods 95
  2.13.1 Jacobi’s Method 96
  2.13.2 Gauss–Seidel Method 100
  2.13.3 Successive Relaxation Method 103
  2.13.4 Convergence 105
2.14 Application: Polynomial Regression 107
2.15 Implementation in MATLAB 112
2.16 Further Reading 114
Problems 116
Computer Problems 132

3 Nonlinear Algebraic Equations 136
3.1 Introduction 136
  3.1.1 Superposition 137
  3.1.2 Differences Between Linear and Nonlinear Equations 138
3.2 Solution Approaches 140
  3.2.1 Bracketing Methods 141
  3.2.2 Fixed-Point (Open) Methods 143
3.3 Fixed-Point Methods for Single Nonlinear Algebraic Equations 145
  3.3.1 Picard’s Method 146
  3.3.2 Newton’s Method 148
3.4 Convergence of General Fixed-Point Iteration 152
3.4.1 Will We Find a Solution? 153
3.4.2 Rate of Convergence 156

3.5 Case Study: Eigenvalue Calculation in Unsteady Diffusion 158
3.5.1 Problem Statement 158
3.5.2 Solution 159

3.6 Case Study: Extension of a Tethered Polymer by a Force 160
3.6.1 Problem Statement 160
3.6.2 Solution 162

3.7 Systems of Nonlinear Algebraic Equations 164
3.7.1 Picard’s Method 165
3.7.2 Newton–Raphson 167

3.8 Case Study: Nonlinear Flash 177
3.8.1 Problem Statement 177
3.8.2 Solution 177
3.8.3 A Variation on the Problem 179

3.9 Continuation Methods 180
3.9.1 Zero-Order Continuation 181
3.9.2 First-Order Continuation 181

3.10 Case Study: Liquid–Liquid Phase Diagram from the van Laar Model 182
3.10.1 Problem Statement 182
3.10.2 Solution 183

3.11 Implementation in MATLAB 188

3.12 Further Reading 190
Problems 191
Computer Problems 195

4 Initial Value Problems 206
4.1 Introduction 206

4.2 First-Order ODE IVPs 208
4.2.1 Explicit (Forward) Euler 209
4.2.2 Implicit (Backward) Euler 214
4.2.3 Predictor–Corrector 220
4.2.4 Runge–Kutta Methods 221

4.3 Numerical Stability 228
4.3.1 Linear IVPs 229
4.3.2 Nonlinear IVPs 232

4.4 Case Study: CSTR with a Fluctuating Inlet 232
4.4.1 Problem Statement 232
4.4.2 Solution 233

4.5 Case Study: Liquid Level Dynamics and Control 236
4.5.1 Problem Statement 236
4.5.2 Solution 238
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.6</td>
<td>Systems of ODE IVPs</td>
<td>242</td>
</tr>
<tr>
<td>4.6.1</td>
<td>Explicit Methods</td>
<td>243</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Implicit Euler</td>
<td>245</td>
</tr>
<tr>
<td>4.7</td>
<td>Higher-Order ODE IVPs</td>
<td>249</td>
</tr>
<tr>
<td>4.8</td>
<td>Numerical Stability for Systems of ODEs</td>
<td>252</td>
</tr>
<tr>
<td>4.8.1</td>
<td>Analysis by Matrix Diagonalization</td>
<td>252</td>
</tr>
<tr>
<td>4.8.2</td>
<td>Stiff Systems</td>
<td>254</td>
</tr>
<tr>
<td>4.9</td>
<td>Case Study: Non-Isothermal PFR</td>
<td>256</td>
</tr>
<tr>
<td>4.9.1</td>
<td>Problem Statement</td>
<td>256</td>
</tr>
<tr>
<td>4.9.2</td>
<td>Solution</td>
<td>258</td>
</tr>
<tr>
<td>4.10</td>
<td>Implementation in MATLAB</td>
<td>261</td>
</tr>
<tr>
<td>4.11</td>
<td>Further Reading</td>
<td>262</td>
</tr>
<tr>
<td></td>
<td>Problems</td>
<td>263</td>
</tr>
<tr>
<td></td>
<td>Computer Problems</td>
<td>270</td>
</tr>
</tbody>
</table>

### 5 Dynamical Systems

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>275</td>
</tr>
<tr>
<td>5.2</td>
<td>Equilibrium Points and Their Stability</td>
<td>276</td>
</tr>
<tr>
<td>5.3</td>
<td>Dynamical Systems on the Plane</td>
<td>277</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Linear Systems on the Plane – Classification of Steady States</td>
<td>278</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Nonlinear Systems on the Plane</td>
<td>282</td>
</tr>
<tr>
<td>5.4</td>
<td>Nonlinear Phenomena</td>
<td>287</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Periodic Solutions</td>
<td>288</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Limit Cycles</td>
<td>289</td>
</tr>
<tr>
<td>5.5</td>
<td>Bifurcations</td>
<td>291</td>
</tr>
<tr>
<td>5.5.1</td>
<td>One-Dimensional Systems</td>
<td>291</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Two-Dimensional Systems</td>
<td>295</td>
</tr>
<tr>
<td>5.6</td>
<td>Deterministic Chaos</td>
<td>297</td>
</tr>
<tr>
<td>5.7</td>
<td>Case Study: Non-Isothermal CSTR</td>
<td>299</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Problem Statement</td>
<td>300</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Solution</td>
<td>302</td>
</tr>
<tr>
<td>5.8</td>
<td>Implementation in MATLAB</td>
<td>312</td>
</tr>
<tr>
<td>5.9</td>
<td>Further Reading</td>
<td>312</td>
</tr>
<tr>
<td></td>
<td>Problems</td>
<td>313</td>
</tr>
<tr>
<td></td>
<td>Computer Problems</td>
<td>317</td>
</tr>
</tbody>
</table>

### 6 Boundary Value Problems

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>ODE Boundary Value Problems</td>
<td>328</td>
</tr>
<tr>
<td>6.2</td>
<td>Finite Difference Approximations</td>
<td>330</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Discretization in Space</td>
<td>330</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Derivation of Finite Difference Equations</td>
<td>330</td>
</tr>
<tr>
<td>6.3</td>
<td>Solution Procedure</td>
<td>333</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Dirichlet Boundary Conditions</td>
<td>334</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Neumann and Robin Boundary Conditions</td>
<td>341</td>
</tr>
</tbody>
</table>
### 6.3.3 Position-Dependent Terms
344
### 6.3.4 Mesh Refinement
348
### 6.4 Case Study: Reaction–Diffusion in a Packed Bed
352
#### 6.4.1 Problem Statement
352
#### 6.4.2 Solution
353
### 6.5 Coupled BVPs
355
#### 6.5.1 Approach #1: Appending the Unknown Vectors
356
#### 6.5.2 Approach #2: Interlacing the Variables
357
### 6.6 Case Study: Coupled Heat and Mass Transfer
358
#### 6.6.1 Problem Statement
358
#### 6.6.2 Solution
359
### 6.7 Implementation in MATLAB
366
### 6.8 Further Reading
366
Problems
366
Computer Problems
368

### 7 Partial Differential Equations
374
#### 7.1 Introduction
374
#### 7.2 Solution of IBVPs – Method of Lines
375
##### 7.2.1 Solution Method
376
##### 7.2.2 Numerical Stability
379
#### 7.3 Case Study: Unsteady Diffusion Through a Membrane
386
##### 7.3.1 Problem Statement
386
##### 7.3.2 Solution
388
#### 7.4 Two-Dimensional BVPs
392
#### 7.5 Case Study: 2D Reaction–Diffusion
401
##### 7.5.1 Problem Statement
401
##### 7.5.2 Solution
402
#### 7.6 Implementation in MATLAB
408
#### 7.7 Further Reading
409
Problems
409
Computer Problems
413

### 8 Interpolation and Integration
421
#### 8.1 Introduction
421
#### 8.2 Polynomial Interpolation
422
##### 8.2.1 Linear Interpolation
422
##### 8.2.2 Quadratic Interpolation
423
##### 8.2.3 Newton’s Divided Differences and Polynomial Interpolation
425
##### 8.2.4 Equally Spaced Data
431
##### 8.2.5 Error of Newton’s Interpolating Polynomials
433
#### 8.3 Newton–Coates Integration
434
##### 8.3.1 Trapezoidal Rule (n = 1)
435
##### 8.3.2 Simpson’s 1/3 Rule (n = 2)
436
## Contents

8.3.3 Higher-Order Formulas

8.4 Applications of Trapezoidal Rule

8.4.1 Multiple Trapezoidal Rule

8.4.2 Richardson Extrapolation

8.4.3 Romberg Integration

8.5 Gauss Quadrature

8.6 Case Study: Concentrated Differential Distillation

8.6.1 Problem Statement

8.6.2 Solution

8.7 Implementation in MATLAB

8.8 Further Reading

Problems

Computer Problems

Appendix A MATLAB “Tutorial”

A.1 Data Structures

A.2 Matrix and Vector Manipulation

A.3 Order of Operations

A.4 Displaying Data on the Screen

A.5 Logical Flow

A.6 Loops

A.7 Plotting

A.8 Functions

Appendix B Determinant of a Matrix

Index
Illustrations

1.1 Idea behind a mathematical model 1
1.2 Concentration profile for Example 1.6 11
1.3 Isothermal continuous stirred tank reactor 12
1.4 Connection between numerical methods and problems 22
1.5 Illustration of 214.321 157 68 represented as a 16 bit word using base 10 23
1.6 Flowchart illustrating the logic of a while loop 28
1.7 Flowchart illustrating the logic of (a) an if statement, (b) an if-else statement, and (c) an if-elseif statement 30
2.1 Comparison between (a) a simple mass balance problem and (b) a more realistic mass balance problem 46
2.2 Plots of the three possibilities for a system of linear equations 48
2.3 Interpretation of the solution of a linear system as the linear combination of basis vectors 50
2.4 Geometrical comparison of (a) an ill-conditioned system and (b) a well-conditioned system of linear algebraic equations 51
2.5 Schematic illustration of multi-component flash 72
2.6 Temperature window for three-component equilibrium of xylene isomers for an atmospheric flash of a feed of 10% o-xylene to 20% liquid 77
2.7 Illustration of the scaling of the time for Gauss elimination using a banded solver 82
2.8 Process flow sheet consisting of a reactor and separator with recycle 91
2.9 Convergence of Jacobi’s method for Eq. (2.13.7) with initial guess (0,0) 101
2.10 Comparison of the convergence of Jacobi’s method and Gauss–Seidel for Eq. (2.13.7) with initial guess (0,0) 103
2.11 Rate of convergence for Eq. (2.13.7) with different relaxation parameters $\omega$ for the initial guess (0,0) 105
2.12 Rate of convergence for Eq. (2.13.29) using Jacobi’s method for the initial guess (0,0) 107
2.13 Difference between (a) interpolation and (b) regression 108
2.14 Process schematic for Problem 2.80 132
3.1 Illustration of the real roots for Eqs. (3.1.16), (3.1.19), and (3.1.20) 139
3.2 Illustration of using a bracketing method for Eq. (3.1.16) 141
3.3 Example of a piecewise continuous function where there are an infinite number of roots 143
3.4 Evolution of the fixed-point method in Eq. (3.2.14) and Eq. (3.2.15) for $x^{(0)} = 1$ 145
List of Illustrations

3.5 Result of up to ten iterations of Picard’s method for Eq. (3.3.2) for the initial guesses $x^{(0)} = -0.1, 0.6, 1.99,$ and 2.01 148
3.6 Convergence of Newton’s method for Eq. (3.2.13) using an initial guess $x^{(0)} = 1$ 150
3.7 Result of Newton’s method for Eq. (3.3.2) for the initial guesses $x^{(0)} = -0.1, 0.6, 1.99,$ and 2.01 151
3.8 Geometric interpretation of Newton’s method 152
3.9 Result of Newton’s method for Eq. (3.3.2) for the initial guesses $x^{(0)} = 1.55$ and 1.51 152
3.10 Illustration of the mean value theorem of calculus 153
3.11 Demonstration of quadratic convergence for Newton’s method using the solutions from Example 3.7 158
3.12 Illustration of the force–extension of a polymer chain 161
3.13 Force extension equation (3.6.7) for DNA with $F = 30$ pN 163
3.14 Examples of making the first step in Newton’s method for (a) a small value of $X/L$ and (b) a value of $X/L \approx 1$ for Fig. 3.13 164
3.15 Convergence of Picard’s method for Eqs. (3.7.6) and (3.7.7) 167
3.16 Lack of convergence of Picard’s method for Eqs. (3.7.6) and (3.7.7) when the equation order is reversed 168
3.17 Evolution of the solution to a system of nonlinear equations towards the root using Newton–Raphson 170
3.18 Trajectory of $x_1^{(k)}$ and $x_2^{(k)}$ as a function of iteration number for the solution of Eqs. (3.7.6) and (3.7.7) 173
3.19 Demonstration of quadratic convergence for the solution of Eqs. (3.7.6) and (3.7.7) by Newton–Raphson 175
3.20 Illustration of (a) first-order continuation and (b) the failure of continuation methods at a turning point 182
3.21 Liquid–liquid phase diagram for a van Laar model with an upper-consolute temperature of 110°C and composition $x_1 = 0.4$ 188
4.1 Illustration of the basic idea behind solving an ordinary differential equation 208
4.2 Graphical illustration of explicit (forward) Euler 210
4.3 Concentration profiles for a batch reactor with second-order kinetics (Eq. (4.2.17)) as a function of time for different values of the step size $h$ from explicit Euler 214
4.4 Schematic illustration of the implicit Euler method 215
4.5 Implicit Euler solution to Eq. (4.2.32) with initial condition $y(0) = 1$ for different step sizes $h$ 218
4.6 Implicit Euler solution to Eq. (4.2.39) for different step sizes $h$ 219
4.7 Predictor–corrector solution to Eq. (4.2.53) for different step sizes $h$ 222
4.8 RK4 solution to Eq. (4.2.91) for different step sizes $h$ 227
4.9 The first four steps of the explicit Euler integration of Eq. (4.3.1) with a step size $h = 4$ 228
4.10 Schematic illustration of a continuous stirred tank reactor with a fluctuating inlet concentration given by Eq. (4.4.2) 233
4.11 Dimensionless concentration as a function of time for different frequencies $\omega$ of the forcing function 235
List of Illustrations

4.12 Dimensionless concentration as a function of time for different values of the strength of the oscillation, $\epsilon$ 235
4.13 Illustration of the numerical stability of the solution for the oscillating CSTR using RK4 with different step sizes, $h$ 236
4.14 Comparison of the linear model in Eq. (4.5.7) and the nonlinear model in Eq. (4.5.3) for a sudden decrease of 10% (a) and 85% (b) in inlet flow rate 240
4.15 Dynamics of the liquid height using the proportional controller in Eq. (4.5.9) for a sudden decrease of 85% in inlet flow rate for $K_c = 10$, 20, and 30 ft²/min 241
4.16 Solution to Eqs. (4.6.43) and (4.6.44) with the initial conditions $y_1(0) = 1$ and $y_2(0) = 2$ using implicit Euler 249
4.17 Illustration of Eqs. (4.8.28) and (4.8.29) for (a) short times, where $\lambda_2$ contributes and (b) long times, where this fast transient has decayed 255
4.18 Schematic illustration of a plug-flow reactor 257
4.19 Evolution of the temperature and concentration in the non-isothermal PFR 259
4.20 Number of tube radii required to reach a conversion of 90% in the non-isothermal PFR 260
4.21 Example of an unstable numerical solution for the non-isothermal PFR using RK4 260
5.1 (a) Schematic illustration of the trajectories $y_1(t)$ and $y_2(t)$. (b) Phase plane representation of $y_2(t)$ as a function of $y_1(t)$ 278
5.2 Illustration of (a) a stable node, (b) an unstable node, and (c) a saddle point 279
5.3 Illustration of (a) a stable focus, (b) an unstable focus, and (c) a center 281
5.4 Illustration of the case of repeated eigenvalues for (a) two linearly independent eigenvectors and (b) one independent eigenvector 282
5.5 Phase plane for Example 5.1 284
5.6 Solution to Example 5.2 leading to (a) a stable focus, (b) a center, and (c) an unstable focus 287
5.7 Phase plane for the predator–prey equations for $a=b=c=d=1$ 289
5.8 Phase plane for Eqs. (5.4.12) and (5.4.13) 290
5.9 Bifurcation diagram for Eq. (5.5.3), which exhibits a pitchfork bifurcation 292
5.10 Bifurcation diagram for Eq. (5.5.4), which exhibits a subcritical pitchfork bifurcation 293
5.11 Bifurcation diagram for Eq. (5.5.5), which exhibits a trans-critical (exchange of stability) bifurcation 294
5.12 Bifurcation diagram for Eq. (5.5.6), which exhibits a hysteresis bifurcation 295
5.13 Hopf bifurcation for the system of Eqs. (5.5.7) and (5.5.8) 296
5.14 Integration of the Lorenz equations for an initial condition $x = y = z = 10$ out to a time of $t_f = 30$ 298
5.15 Integration of the Lorenz equations to $t=15$, where $x=4.0049$, $y=5.9239$, $z=16.8624$ (dashed lines). The data are rounded to two decimal places and then integrated out to a time of $t_f = 30$ 298
5.16 Phase portrait for the Lorenz equations 299
5.17 Schematic illustration of a non-isothermal CSTR 300
5.18 Upper and lower branches of steady states for the non-isothermal CSTR with the parameters in Eq. (5.7.20) 305
List of Illustrations

5.19 Complete set of steady states for the non-isothermal CSTR with the parameters in Eq. (5.7.20) 307
5.20 Complete set of steady states for the non-isothermal CSTR with the parameters in Eq. (5.7.20) 309
5.21 Phase plane for the non-isothermal CSTR with the parameters in Eq. (5.7.20) with $\tau = 1$ 310
5.22 Limit cycle for the non-isothermal CSTR 311
5.23 Phase plane for Problem 5.17 316
5.24 Schematic of a CSTR bioreactor for Problem 5.27 320
6.1 Discretization of spatial domain into $n$ equidistant nodes 330
6.2 Comparison of the exact solution and the numerical solution of the reaction–diffusion equation with $n = 11$ nodes 337
6.3 Comparison of the exact solution and the numerical solution of the reaction–diffusion equation as a function of the number of nodes 337
6.4 Solution to Eq. (6.3.15) using finite differences with 51 nodes 341
6.5 Introduction of a fictitious node for Neumann and Robin boundary conditions 342
6.6 Solution to Eq. (6.3.15) with the Neumann boundary condition (6.3.29) on the left boundary 344
6.7 Schematic illustration of the heating problem in Example 6.5 345
6.8 Approach for handling the case where the heater start/end does not align with the location of the nodes 346
6.9 Temperature profile for Example 6.5 for (a) different heating rates at $\epsilon = 0.5$ and (b) different heating areas at $q_0 = 1000$ W/m$^3$ 347
6.10 Schematic illustration of mesh refinement from $n = 3$ to $n = 5$ to $n = 9$. Each time, the new grid has an additional node between each of the old grid points 348
6.11 Result of mesh refinement for the reaction–diffusion equation with a Thiele modulus of unity 351
6.12 Schematic illustration of a packed bed with a spatially dependent reaction rate 352
6.13 Concentration profile in a packed bed with discrete reaction sites as a function of the Thiele modulus 355
6.14 Discretization of space for a system of two coupled boundary value problems 356
6.15 Non-isothermal reaction–diffusion between an empty reservoir on the left and a full reservoir on the right 359
6.16 Impact of the activation energy on the concentration and temperature profile 364
6.17 A problem when $\phi = 0$ for an endothermic reaction 365
7.1 Schematic illustration of the method of lines solution to an IBVP 376
7.2 Integration of Eq. (7.2.1) by explicit Euler with (a) $h = 0.001$ and (b) $h = 0.0065$ 382
7.3 Solution to Eq. (7.2.31) using implicit Euler 386
7.4 Diffusion through a membrane of thickness $L$ connecting two infinite reservoirs at concentrations $c_i$ and $c_0$ 386
7.5 Comparison of the explicit Euler solution and the exact solution in Eq. (7.3.10) for two different times $\tau = 0.001$ and $\tau = 0.06$ 389
List of Illustrations

7.6 Comparison of the explicit Euler solution and the exact solution in Eq. (7.3.10) as a function of the number of terms in the Fourier series 390
7.7 Diffusion through a membrane into a finite tank 392
7.8 Schematic illustration of the problem of heat transfer in a square domain 392
7.9 Discretization of the PDE at node $i,j$ 393
7.10 Conversion from local indices to a global index 395
7.11 Discretization with the location of the central node and examples of the fictitious nodes on the boundaries 397
7.12 Solution to the heat equation with a source at the center 401
7.13 Solution to the heat equation with two sources, one at full intensity and the second at half intensity 401
7.14 Comparison of the concentration profile for the (a) linear and (b) nonlinear reaction 404
7.15 Concentration at the center of the domain as a function of Thiele modulus 408
7.16 Boundary conditions for Problem 7.12 412
7.17 Geometry for Problem 7.23 418
7.18 Grid for Problem 7.23 419
8.1 Schematic illustration of using an interpolating polynomial for numerical integration 422
8.2 Schematic illustration of linear interpolation 423
8.3 Schematic illustration of quadratic interpolation 423
8.4 Comparison of a cubic interpolation to $\tan(x)$ at the points $x_0 = 0$, $x_1 = \pi/3$, $x_2 = \pi/6$, and $x_3 = \pi/4$ with the exact result 430
8.5 Utility of multiple trapezoidal rule to handle functions with significant curvature 440
8.6 Error in the multiple trapezoidal rule integration of Eq. (8.4.8) as a function of the number of trapezoids 441
8.7 Schematic illustration of the Romberg integration algorithm 444
8.8 Schematic illustration of the computational advantage of increasing the number of trapezoids by a factor of 2 rather than some arbitrary number 445
8.9 Uniformly spaced interpolation points may lead to overestimates or underestimates of the area under the curve 447
8.10 Concentrated differential distillation 452
A.1 Example of MATLAB plotting a single function 478
A.2 Example of MATLAB plotting two functions 479
A.3 Example of a hierarchy of functions for the case study in Section 3.10 481
# Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Antoine coefficients for isomers of xylene</td>
<td>73</td>
</tr>
<tr>
<td>3.1</td>
<td>Antoine coefficients for ethylbenzene and toluene</td>
<td>177</td>
</tr>
<tr>
<td>5.1</td>
<td>Conditions for phase plane plots in Problem 5.29</td>
<td>326</td>
</tr>
<tr>
<td>8.1</td>
<td>Values of the integral in Eq. (8.3.22) using Newton–Coates formulas of order $n$</td>
<td>439</td>
</tr>
<tr>
<td>8.2</td>
<td>Gauss points and weight for three-point, four-point, and five-point Gauss quadrature</td>
<td>449</td>
</tr>
<tr>
<td>8.3</td>
<td>Values of the integral in Eq. (8.3.22) using Gauss quadrature formulas of order $n$</td>
<td>451</td>
</tr>
<tr>
<td>8.4</td>
<td>Vapor–liquid equilibrium data for the mole fraction of toluene</td>
<td>452</td>
</tr>
<tr>
<td>8.5</td>
<td>Tabulated data for Example 8.12</td>
<td>456</td>
</tr>
<tr>
<td>8.6</td>
<td>Rate data for Problem 8.12</td>
<td>460</td>
</tr>
<tr>
<td>8.7</td>
<td>Concentration data of effluent in Problem 8.13</td>
<td>461</td>
</tr>
<tr>
<td>8.8</td>
<td>Concentration data for breakthrough curve in Problem 8.14</td>
<td>462</td>
</tr>
<tr>
<td>A.1</td>
<td>Logical operators in MATLAB</td>
<td>475</td>
</tr>
</tbody>
</table>
# Programs

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Naive Gauss elimination</td>
<td>64</td>
</tr>
<tr>
<td>2.2</td>
<td>Gauss elimination with pivoting</td>
<td>70</td>
</tr>
<tr>
<td>2.3</td>
<td>Determinant from Gauss elimination</td>
<td>79</td>
</tr>
<tr>
<td>2.4</td>
<td>Banded Gauss elimination</td>
<td>81</td>
</tr>
<tr>
<td>2.5</td>
<td>LU decomposition</td>
<td>89</td>
</tr>
<tr>
<td>2.6</td>
<td>Jacobi’s method</td>
<td>97</td>
</tr>
<tr>
<td>2.7</td>
<td>Jacobi’s method with matrices</td>
<td>99</td>
</tr>
<tr>
<td>2.8</td>
<td>Gauss–Seidel method</td>
<td>102</td>
</tr>
<tr>
<td>2.9</td>
<td>Successive relaxation</td>
<td>104</td>
</tr>
<tr>
<td>2.10</td>
<td>Linear regression</td>
<td>110</td>
</tr>
<tr>
<td>3.1</td>
<td>Bracketing</td>
<td>141</td>
</tr>
<tr>
<td>3.2</td>
<td>Picard’s method</td>
<td>146</td>
</tr>
<tr>
<td>3.3</td>
<td>Newton’s method</td>
<td>149</td>
</tr>
<tr>
<td>3.4</td>
<td>Picard’s method for a system of equations</td>
<td>165</td>
</tr>
<tr>
<td>3.5</td>
<td>Newton–Raphson</td>
<td>171</td>
</tr>
<tr>
<td>4.1</td>
<td>Explicit Euler</td>
<td>210</td>
</tr>
<tr>
<td>4.2</td>
<td>Implicit Euler</td>
<td>215</td>
</tr>
<tr>
<td>4.3</td>
<td>Predictor–corrector</td>
<td>220</td>
</tr>
<tr>
<td>4.4</td>
<td>Fourth-order Runge–Kutta</td>
<td>225</td>
</tr>
<tr>
<td>4.5</td>
<td>Implicit Euler for a system of equations</td>
<td>246</td>
</tr>
<tr>
<td>8.1</td>
<td>Polynomial interpolation</td>
<td>427</td>
</tr>
<tr>
<td>8.2</td>
<td>Generate interpolant</td>
<td>428</td>
</tr>
<tr>
<td>8.3</td>
<td>Interpolation with equally spaced data</td>
<td>432</td>
</tr>
<tr>
<td>8.4</td>
<td>Newton–Coates integration</td>
<td>437</td>
</tr>
<tr>
<td>8.5</td>
<td>Multiple trapezoidal rule</td>
<td>441</td>
</tr>
<tr>
<td>8.6</td>
<td>Richardson extrapolation</td>
<td>443</td>
</tr>
<tr>
<td>8.7</td>
<td>Romberg integration</td>
<td>445</td>
</tr>
<tr>
<td>8.8</td>
<td>Gauss quadrature</td>
<td>450</td>
</tr>
</tbody>
</table>
Preface

Why write yet another book on numerical methods? Having taught this course for a combined 10 years, we have continuously struggled to find a single textbook that works for chemical engineering students. There are outstanding books on the mathematical underpinnings of numerical methods, but the level of sophistication (theory–lemma) is often a turnoff for chemical engineering students and overkill at the introductory stage. There are also excellent generic books on numerical methods that include some engineering applications. We used one such book as a reference for the course, since many students do better with supplemental reading. The challenge with general books is that they do not incorporate sufficient chemical engineering examples, which leads to students wondering exactly how the textbook (which is often expensive) connects to the class. Moreover, with the explosion of online resources, it is quite easy to find explanations of standard numerical methods with the click of a mouse. (Whether those explanations are good or even correct is another issue entirely!) At the other end of the spectrum are chemical engineering textbooks solely devoted to applications with little discussion of the underlying numerical methods. These books provide a wealth of example problems, but they are not suitable for teaching students how to actually solve the problems.

Our goal in writing this textbook is to take a “Goldilocks” approach: not too rigorous, not too applied! At the same time, we want to firmly embed our discussion of numerical methods in the context of chemical engineering applications. The material presented in the book is based on the content of a course in numerical methods developed in our department more than 20 years ago. The book is intended for an undergraduate course in numerical methods, primarily for chemical engineering students. At Minnesota, this is a core course taught in the spring semester of the junior year. The students are taking this course concurrently with (i) Reaction Kinetics and Reactor Design and (ii) Mass Transfer and Separations. This location in the curriculum offers many opportunities to motivate the methods covered in the book and use them to solve realistic chemical engineering problems. For example, we have timed the numerical methods course and the reactors course so that, when we cover systems of ordinary differential equations in numerical methods, the reactors course covers systems of chemical reactions. The students have already taken Transport Phenomena and Thermodynamics in the fall, so these subjects are also covered in the book through examples of problems that could not be solved by hand in the preceding courses.

While we believe this to be the ideal location for the course, we provide sufficient background information for the different problems so that the students can solve them
without necessarily having a previous course in the subject. As a result, the book can certainly be used early in the curriculum, for example in conjunction with Mass and Energy Balances, especially if the goal is to enhance the use of numerical methods in that class and to build a foundation in numerical methods that can be used later in the curriculum. The book would also fit well at the end of the chemical engineering curriculum, when the students have all of the scientific and engineering background but cannot yet solve “real” problems that lack closed-form solutions.

An emerging challenge that we are facing in our curriculum is a lack of programming experience. This is ironic, as it is hard to imagine that students have become any less reliant on computation in their education or their daily life. The problem is that our university (and others elsewhere) no longer requires entering engineering students to take an introductory programming course. Even when such a course is required, it rarely focuses on the aspects that are required for scientific computing. The logic for this evolution in the chemical engineering curriculum is two-fold. The first is commendable, namely creating a manageable course load for students so that they can graduate in a timely manner. Because of the recent expansion of the chemical engineering curriculum in areas such as biology, something needs to be removed to make space. The second is less defensible, namely that students will use packages such as MATLAB or Aspen to solve chemical engineering problems, so they no longer need to be taught basic programming. At a philosophical level, this argument is very troubling – it essentially argues that engineers do not need to understand how their tools work. At a more practical level, although programming environments (such as MATLAB) are prevalent nowadays, they are often used without a solid understanding of the underlying numerical methods. This “black box” approach can easily lead to erroneous analysis and interpretation of results. Knowledge of the underlying numerical methods is essential in order to be able to choose from a suite of options usually available, and use them correctly and efficiently. Numerical methods go hand-in-hand with programming. Moreover, with the discipline moving towards more fundamental, science-oriented research, the students’ background in mathematics is often lacking, and so is their ability to appreciate the power of modern mathematics to provide solutions to cutting-edge problems across science and engineering. Our overall goal in this book is to address this problem with a single-semester, focused course that integrates basic programming skills and numerical methods to solve chemical engineering problems.

**Organization of the Book**

The book starts with a review of mathematical problems arising in chemical engineering, highlighting the (restrictive) assumptions which make them amenable to analytical solutions, and thus motivating the need for numerical methods. The problems covered in the rest of the text are (in order): linear algebraic equation systems, nonlinear algebraic equation systems, ordinary differential equation systems – initial value problems, ordinary differential equations – boundary value problems, classes of partial differential equations (initial-boundary and two-dimensional boundary value problems), and...
numerical integration. As such, the level of difficulty gradually increases, and the solution methods naturally build on those developed earlier. This arrangement of the material deviates from the standard approach in numerical methods in several substantial respects. For example, we have split up the material on numerical differentiation and integration, which often appear together early in a numerical methods course. The numerical differentiation appears in the context of boundary value problems, where it becomes clear why it is necessary, and the numerical integration is deferred to the end of the book, as its importance in chemical engineering is subordinate to solving nonlinear algebraic equations and differential equations.

For each class of problems, we start with a thorough derivation of the numerical approximation methods covered. We discuss in some length topics such as accuracy, convergence, and numerical stability properties of the derived methods. Topics such as ill-conditioning and stiffness are also covered as they are essential to be able to detect limitations inherent to the nature of the underlying physical/engineering problem we are trying to solve. We devote a separate chapter to dynamical systems, where we also provide an introduction to nonlinear dynamical phenomena.

We show how each numerical method can be programmed within MATLAB. We begin by assuming very limited programming experience, introducing elementary programming concepts in the first chapter. We then build on these ideas as we develop the necessary MATLAB codes from scratch, explaining in a step-by-step fashion their structure, function, and implementation. We then use (and modify) the codes that we built to solve chemical engineering problems. Importantly, this is not a book on using canned routines in MATLAB. Rather, we take advantage of MATLAB’s easy debugging and interfacing with plot commands to make the course much simpler than would be the case if the students used a structured programming language like C. Indeed, the only canned numerical method that we use in the book is MATLAB’s linear solver (the slash operator), and only after we have covered elementary methods for solving linear algebraic systems. We believe this to be a key feature of the book in that we essentially use the MATLAB environment to teach programming. Recognizing that students should also know how to use the canned routines as well, the end of each chapter provides a brief outline of MATLAB’s built-in functions, but only after the students have learned the basics. All of the scripts appearing in the book are available as a downloadable resource.

In each chapter, in addition to short illustrative examples, we include detailed case studies that draw from core chemical engineering subjects (mass balances, thermodynamics, transport, kinetics, reactor design). We use these case studies to illustrate the value of numerical methods in solving realistic chemical engineering problems, but also to provide new perspective on problems that students have already seen (e.g. using continuation to generate phase diagrams). These examples are spaced out in such a way so that their frequency increases as the book proceeds. We have found this to be a useful way to organize a course using this book; as the material becomes more difficult, we slow down the pace of new numerical concepts and reinforce their applicability in chemical engineering.
Finally, numerous problems are included at the end of each chapter that address theoretical questions as well as application of the methods to chemical engineering problems. These problems have a wide range of difficulty and depth, ranging from simple short answer problems that test basic concepts to extended questions that we usually assign over several weeks for students to address in teams. A complete solution manual for instructors, including all of the codes required to solve the problems, is also available.
“Dorfman and Daoutidis’s new book is a welcome addition to the undergraduate chemical engineering textbook literature. It provides a very attractive combination of programming, applied mathematics and chemical engineering applications, and is written in an accessible style. The incorporation of example MATLAB codes will be very helpful to students.”

Michael D. Graham, University of Wisconsin-Madison
Acknowledgments

While we are the ones who put the pen to paper (or, more appropriate, fingers to keyboard) to write this book, its content is the result of several decades of development in the chemical engineering curriculum. The numerical methods course as part of the core chemical engineering curriculum was born out of the vision of H. Ted Davis more than 20 years ago. Numerous colleagues have contributed to the course over the years. We thank all of them, especially Bob Tranquillo, Jeff Derby, and Satish Kumar who have taught the course several times and helped shape its content. We are particularly indebted to Jeff Derby’s contributions to early versions of several topics in the book, most notably in linear algebra and the discussion of the Lorenz attractor. We would also like to thank Andre Mkhoyan, Yiannis Kaznessis, and Matt Neurock for their contributions in recent teachings of the course as recitation instructors in the Minnesota team-teaching system. We have benefitted from the help of numerous excellent teaching assistants and undergraduates in preparing the problems and figures in this book. In particular, we thank Doug Tree, Sarit Dutta, Shawn Dodds, and Patrick Smadbeck for some of the most challenging problems in the book, and Yanal Oueis, Nate Lawdenski, Cody Palbicki, Albert Wu, Steven Cordes, Scott King, and Sarit Dutta for providing problems and figures, as well as finding many (but likely not all) typos in the book. While we have had ample assistance in preparing this text, we take responsibility for the accuracy of the final product.

This book would not have been finished without the support from the Department of Chemical Engineering and Materials Science under the leadership of Frank Bates and Dan Frisbie, who provided both the encouragement to undertake this project and financial resources to support many of the students acknowledged above. Likewise, we would have been stopped at the gates without the patience and loving support of our families.