Numerical Methods in Engineering with Python 3

This book is an introduction to numerical methods for students in engineering. It covers the usual topics found in an engineering course: solution of equations, interpolation and data fitting, solution of differential equations, eigenvalue problems, and optimization. The algorithms are implemented in Python 3, a high-level programming language that rivals MATLAB® in readability and ease of use. All methods include programs showing how the computer code is utilized in the solution of problems.

The book is based on *Numerical Methods in Engineering with Python*, which used Python 2. Apart from the migration from Python 2 to Python 3, the major change in this new text is the introduction of the Python plotting package *Matplotlib*.

Jaan Kiusalaas is a Professor Emeritus in the Department of Engineering Science and Mechanics at Pennsylvania State University. He has taught computer methods, including finite element and boundary element methods, for more than 30 years. He is also the co-author or author of four books – *Engineering Mechanics: Statics*; *Engineering Mechanics: Dynamics; Mechanics of Materials*; *Numerical Methods in Engineering with MATLAB* (2nd edition); and two previous editions of *Numerical Methods in Engineering with Python*.

NUMERICAL METHODS IN ENGINEERING WITH PYTHON 3

Jaan Kiusalaas
The Pennsylvania State University

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town,
Singapore, São Paulo, Delhi, Mexico City

Cambridge University Press
32 Avenue of the Americas, New York, NY 10013-2473, USA

www.cambridge.org
Information on this title: www.cambridge.org/9781107033856

First published 2013

Printed in the United States of America

A catalog record for this publication is available from the British Library.

Library of Congress Cataloging in Publication data

Kiusalaas, Jaan.
Numerical methods in engineering with Python 3 / Jaan Kiusalaas.
pages cm
Includes bibliographical references and index.
ISBN 978-1-107-03385-6
1. Engineering mathematics – Data processing. 2. Python (Computer program language) I. Title.
TA345.K58 2013
620.00285′5133–dc23 2012036775

ISBN 978-1-107-03385-6 Hardback

Additional resources for this publication at www.cambridge.org/kiusalaaspython.

Cambridge University Press
978-1-107-03385-6 - Numerical Methods in Engineering with Python 3
Jaan Kiusalaas
Frontmatter
More information

Contents

Preface ix

1 Introduction to Python 1
1.1 General Information 1
1.2 Core Python 4
1.3 Functions and Modules 16
1.4 Mathematics Modules 18
1.5 `numpy` Module 20
1.6 Plotting with `matplotlib.pyplot` 25
1.7 Scoping of Variables 28
1.8 Writing and Running Programs 29

2 Systems of Linear Algebraic Equations 31
2.1 Introduction 31
2.2 Gauss Elimination Method 37
2.3 LU Decomposition Methods 44
Problem Set 2.1 55
2.4 Symmetric and Banded Coefficient Matrices 59
2.5 Pivoting 69
Problem Set 2.2 78
*2.6 Matrix Inversion 84
*2.7 Iterative Methods 87
Problem Set 2.3 98
2.8 Other Methods 102

3 Interpolation and Curve Fitting 104
3.1 Introduction 104
3.2 Polynomial Interpolation 105
3.3 Interpolation with Cubic Spline 120
Problem Set 3.1 126
3.4 Least-Squares Fit 129
Problem Set 3.2 141

4 Roots of Equations 145
4.1 Introduction 145
4.2 Incremental Search Method 146

4.3 Method of Bisection ... 148
4.4 Methods Based on Linear Interpolation ... 151
4.5 Newton-Raphson Method ... 156
4.6 Systems of Equations ... 161
Problem Set 4.1 ... 166
*4.7 Zeros of Polynomials ... 173
Problem Set 4.2 ... 180
4.8 Other Methods ... 182

5 Numerical Differentiation ... 183
5.1 Introduction ... 183
5.2 Finite Difference Approximations ... 183
5.3 Richardson Extrapolation ... 188
5.4 Derivatives by Interpolation ... 191
Problem Set 5.1 ... 195

6 Numerical Integration ... 199
6.1 Introduction ... 199
6.2 Newton-Cotes Formulas ... 200
6.3 Romberg Integration ... 207
Problem Set 6.1 ... 212
6.4 Gaussian Integration ... 216
Problem Set 6.2 ... 230
*6.5 Multiple Integrals ... 232
Problem Set 6.3 ... 243

7 Initial Value Problems ... 246
7.1 Introduction ... 246
7.2 Euler's Method ... 247
7.3 Runge-Kutta Methods ... 252
Problem Set 7.1 ... 263
7.4 Stability and Stiffness ... 268
7.5 Adaptive Runge-Kutta Method ... 271
7.6 Bulirsch-Stoer Method ... 280
Problem Set 7.2 ... 287
7.7 Other Methods ... 292

8 Two-Point Boundary Value Problems ... 293
8.1 Introduction ... 293
8.2 Shooting Method ... 294
Problem Set 8.1 ... 304
8.3 Finite Difference Method ... 307
Problem Set 8.2 ... 317

9 Symmetric Matrix Eigenvalue Problems ... 321
9.1 Introduction ... 321
9.2 Jacobi Method ... 324
9.3 Power and Inverse Power Methods ... 336
Problem Set 9.1 ... 345
9.4 Householder Reduction to Tridiagonal Form ... 351

9.5 Eigenvalues of Symmetric Tridiagonal Matrices 359
Problem Set 9.2 368
9.6 Other Methods 373

10 Introduction to Optimization 374
10.1 Introduction 374
10.2 Minimization Along a Line 376
10.3 Powell's Method 382
10.4 Downhill Simplex Method 392
Problem Set 10.1 399

Appendices 407
A1 Taylor Series 407
A2 Matrix Algebra 410

List of Program Modules (by Chapter) 417

Index 421

Cambridge University Press
978-1-107-03385-6 - Numerical Methods in Engineering with Python 3
Jaan Kiusalaas
Frontmatter
More information

Preface

This book is targeted toward engineers and engineering students of advanced standing (juniors, seniors, and graduate students). Familiarity with a computer language is required; knowledge of engineering mechanics (statics, dynamics, and mechanics of materials) is useful, but not essential.

The primary purpose of the text is to teach numerical methods. It is not a primer on Python programming. We introduce just enough Python to implement the numerical algorithms. That leaves the vast majority of the language unexplored.

Most engineers are not programmers, but problem solvers. They want to know what methods can be applied to a given problem, what their strengths and pitfalls are, and how to implement them. Engineers are not expected to write computer code for basic tasks from scratch; they are more likely to use functions and subroutines that have been already written and tested. Thus, programming by engineers is largely confined to assembling existing bits of code into a coherent package that solves the problem at hand.

The "bit" of code is usually a function that implements a specific task. For the user the details of the code are unimportant. What matters are the interface (what goes in and what comes out) and an understanding of the method on which the algorithm is based. Because no numerical algorithm is infallible, the importance of understanding the underlying method cannot be overemphasized; it is, in fact, the rationale behind learning numerical methods.

This book attempts to conform to the views outlined earlier. Each numerical method is explained in detail and its shortcomings are pointed out. The examples that follow individual topics fall into two categories: hand computations that illustrate the inner workings of the method, and small programs that show how the computer code is utilized in solving a problem. Problems that require programming are marked with ■.

The material consists of the usual topics covered in an engineering course on numerical methods: solution of equations, interpolation and data fitting, numerical differentiation and integration, solution of ordinary differential equations, and eigenvalue problems. The choice of methods within each topic is tilted toward relevance to engineering problems. For example, there is an extensive discussion of symmetric, sparsely populated coefficient matrices in the solution of simultaneous equations.

In the same vein, the solution of eigenvalue problems concentrates on methods that efficiently extract specific eigenvalues from banded matrices.

An important criterion used in the selection of methods was clarity. Algorithms requiring overly complex bookkeeping were rejected regardless of their efficiency and robustness. This decision, which was taken with great reluctance, is in keeping with the intent to avoid emphasis on programming.

The selection of algorithms was also influenced by current practice. This disqualified several well-known historical methods that have been overtaken by more recent developments. For example, the secant method for finding roots of equations was omitted as having no advantages over Ridder's method. For the same reason, the multistep methods used to solve differential equations (e.g., Milne and Adams methods) were left out in favor of the adaptive Runge-Kutta and Bulirsch-Stoer methods.

Notably absent is a chapter on partial differential equations. It was felt that this topic is best treated by finite element or boundary element methods, which are outside the scope of this book. The finite difference model, which is commonly introduced in numerical methods texts, is just too impractical in handling multidimensional boundary value problems.

As usual, the book contains more material than can be covered in a three-credit course. The topics that can be skipped without loss of continuity are tagged with an asterisk (*).

What Is New in This Edition

This book succeeds *Numerical Methods in Engineering with Python*, which was based on Python 2. As the title implies, the new edition migrates to Python 3. Because the two versions are not entirely compatible, almost all computer routines required some code changes.

We also took the opportunity to make a few changes in the material covered:

- An introduction to the Python plotting package `matplotlib.pyplot` was added to Chapter 1. This package is used in numerous example problems, making the book more graphics oriented than before.
- The function `plotPoly`, which plots data points and the corresponding polynomial interpolant, was added to Chapter 3. This program provides a convenient means of evaluating the fit of the interpolant.
- At the suggestion of reviewers, the Taylor series method of solving initial value problems in Chapter 7 was dropped. It was replaced by Euler's method.
- The Jacobi method for solving eigenvalue problems in Chapter 9 now uses the threshold method in choosing the matrix elements marked for elimination. This change increases the speed of the algorithm.
- The adaptive Runge-Kutta method in Chapter 7 was recoded, and the Cash-Karp coefficients replaced with the Dormand-Prince coefficients. The result is a more efficient algorithm with tighter error control.

- Twenty-one new problems were introduced, most of them replacing old problems.
- Some example problems in Chapters 4 and 7 were rearranged or replaced with new problems. The result of these changes is better coordination of examples with the text.

The programs listed in the book were tested with Python 3.2 under Windows 7. The source codes are available at *www.cambridge.org/kiusalaaspython*.