

Also available → a diskette version with FORTRAN
and C languages source codes

Numerical Methods

All Computer programs in
FORTRAN 77/90

C Language conversion of
programs in an appendix

E Balagurusamy

Numerical Methods

ABOUT THE AUTHOR

E Balagurusamy holds an ME (Hons) degree in Electrical Engineering and a Ph D in Systems Engineering. He has been a teacher, trainer, and consultant in the area of computing for the past two decades. He has also served as a consultant to the United Nations Development Organisation (UNIDO) for a number of years. Currently, he is the Director of PSG Institute of Management, Coimbatore, a premier center for management sciences in the country.

A prolific writer, Prof. Balagurusamy has published a large number of research papers in international journals of repute like IEEE Transactions on Reliability, International Journal of Mathematics, Science and Education and International Journal of Systems Science. He has also authored several books including *Reliability Engineering* (TMH) and *Computer-Oriented Statistical and Numerical Methods*. His best selling books on computing include *Programming in ANSI C*, *Object-Oriented Programming with C++*, *Programming with Java*, and *FORTRAN for Beginners* (all by TMH).

A recipient of numerous honours and awards, he has been listed in the Directory of Who's Who of Intellectuals and in the Directory of Distinguished Leaders in Education.

Numerical Methods

E Balagurusamy

Director

PSG Institute of Management

Coimbatore

2004



Tata McGraw-Hill Publishing Company Limited

NEW DELHI

McGraw-Hill Offices

New Delhi New York St Louis San Francisco Auckland Bogotá Caracas
Lisbon London Madrid Mexico City Milan Montreal
San Juan Singapore Sydney Tokyo Toronto

Tata McGraw-Hill



A Division of The McGraw-Hill Companies

Information contained in this work has been obtained by Tata McGraw-Hill Publishing Company Limited, from sources believed to be reliable. However, neither Tata McGraw-Hill nor its authors guarantee the accuracy or completeness of any information published herein, and neither Tata McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that Tata McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

© 1999, Tata McGraw-Hill Publishing Company Limited

Fourth reprint 2000

RODCRRYCRZZDC

Reprint-2004

No part of this publication can be reproduced in any form or by any means without the prior written permission of the publishers

This edition can be exported from India only by the publishers,
Tata McGraw-Hill Publishing Company Limited

ISBN 0-07-463311-2

Published by Tata McGraw-Hill Publishing Company Limited,
7 West Patel Nagar, New Delhi 110 008, typeset in New Century at
The Composers, 20/5 Old Market, West Patel Nagar, New Delhi 110 008,
and printed at Rashtriya Printers, Naveen Shahdara, Delhi 110 032

Cover: Rashtriya

Preface

Application of computer-oriented *numerical methods* has become an integral part of the life of all the modern engineers and scientists. The advent of powerful small computers and workstations has tremendously increased the speed, power and flexibility of numerical computing. Recognizing the importance of computers and numerical computing, all the universities now offer courses in both computing and numerical methods in their engineering curriculum. They are also trying to integrate computer-oriented numerical analysis into other courses such as mathematics and computer science. It is believed that students who can understand, enjoy, and successfully apply the methods of numerical computing to solve problems make better logicians and thereby better engineers and scientists.

The contents of this book are derived from a study of curricula offered by various universities across the country. The book covers both the introductory topics and the more advanced topics such as eigenvalue problems and partial differential equations. The primary goal is to provide students with a sound introduction of numerical methods as well as to make the learning a pleasurable experience. Logical arrangement of topics, clarity of presentation, and illustration through examples aid the student to become more and more adept in applying the methods.

There are a number of books in this field. Why another book? This book is uniquely different from many other books in a number of ways. The salient features are:

- Presents a detailed account of *process* and *characteristics* of numerical computing
- Discusses the *concept of computing* using modern computers.
- Places emphasis on the nature of *computer arithmetic* which is different from real arithmetic in a number of ways.
- The concept, cause and consequence of *errors* in the application of numerical computing have been highlighted.

- An overview of FORTRAN 77/90 has been given which would serve as an introduction to the beginners and as a reference to those who have already learned this language.
- Mathematical derivation of each method is given to build the reader's understanding of numerical analysis.
- Wherever possible, *algorithms* of computing are given in pseudocode using boxes.
- A variety of *solved examples* that would sharpen the skills in both the theory and application of numerical computing are provided.
- *Computer programs* for almost all numerical methods discussed have been presented in both FORTRAN 77 and ANSI C languages.
- A separate diskette version of the book is available; the diskette contains the source codes of programs in FORTRAN 77 and C languages.
- Programs are designed using the modern *modular and structured programming* concepts. These modules can be used as building blocks in other programs.
- *Error analysis* of each method is presented.
- Each chapter begins with a statement of *need and scope* giving a preview of what is coming later.
- Each chapter ends with a *summary* and a set of *key terms* that remind the reader what has been covered and discussed.
- *Review questions* provide an opportunity to the reader to test his understanding of the concepts.
- Numerous end-of-chapter *practice exercises* are an important supplement to the text. They would not only improve the understanding of algorithms but also enhance the application skills.
- *Programming projects* give students ample opportunities to practice their skills of scientific programming.

I have exerted a conscious effort to make the book *student-oriented* and *student-friendly*. I hope the students would find the book not only interesting but also useful.

I wish to acknowledge the many helpful suggestions of reviewers which have certainly improved both the content and quality of the material. It is the persuasion and encouragement of Dr N Subrahmanyam and Vibha Mahajan of Tata McGraw-Hill that has made the publication of this book possible. Thanks are due to K Balakrishnan and J R Pratibha whose excellent word processing skills made the preparation of the book in its present form much easier and possible.

The book is full of equations and formulas that contain a large number of variables, subscripts, superscripts, etc. I shall be grateful, if the readers could communicate to me any errors they discover.

I dedicate this book to my wife Sushila who is the inspiration in all my work.

Contents

Preface

v

1

- 1. Introduction to Numerical Computing**
 - 1.1 Introduction 1
 - 1.2 Numeric Data 2
 - 1.3 Analog Computing 2
 - 1.4 Digital Computing 3
 - 1.5 Process of Numerical Computing 3
 - 1.6 Characteristics of Numerical Computing 8
 - Accuracy* 8
 - Rate of convergence* 8
 - Numerical stability* 8
 - Efficiency* 9
 - 1.7 Computational Environment 9
 - 1.8 New Trends in Numerical Computing 9
 - 1.9 Mathematical Background 10
 - 1.10 Summary 10
 - Key terms* 10
 - Review questions* 10

- 2. Introduction to Computers and Computing Concepts** 12
 - 2.1 Introduction 12
 - 2.2 Evolution of Numerical Computing and Computers 14
 - Modern computers* 15
 - Fifth generation computers* 16
 - 2.3 Types of Computers 17
 - Principles of operation* 17
 - Applications* 18
 - Size and capability* 18
 - 2.4 Computing Concepts 20

2.5	Computer Organisation	21
	<i>Input devices</i>	21
	<i>Processing units</i>	22
	<i>Output devices</i>	22
	<i>External storage devices</i>	23
2.6	Driving the Computer: The Software	24
	<i>Operating system</i>	24
	<i>Utility programs</i>	25
	<i>Language processors</i>	26
	<i>Application programs</i>	26
2.7	Programming Languages	27
	<i>Machine language</i>	27
	<i>Assembly language</i>	27
	<i>Procedure-oriented language (POL)</i>	28
	<i>Common high-level languages</i>	29
2.8	Interactive Computing	30
2.9	Problem Solving and Algorithms	30
2.10	Flow Charting	32
2.11	Structuring the Logic	33
2.12	Using the Computer	35
2.13	Summary	36
	<i>Key terms</i>	36
	<i>Review questions</i>	37
3.	Computer Codes and Arithmetic	40
3.1	Introduction	40
3.2	Decimal System	41
3.3	Binary System	42
3.4	Hexadecimal System	42
3.5	Octal System	44
3.6	Conversion of Numbers	45
	<i>Non-decimal system to decimal system</i>	45
	<i>Decimal system to non-decimal system</i>	46
	<i>Octal and hexadecimal conversion</i>	48
3.7	Representation of Numbers	49
	<i>Integer representation</i>	50
	<i>Floating-point representation</i>	51
3.8	Computer Arithmetic	52
	<i>Integer arithmetic</i>	52
	<i>Floating-point arithmetic</i>	53
3.9	Errors in Arithmetic	55
3.10	Laws of Arithmetic	57
3.11	Summary	58
	<i>Key terms</i>	58
	<i>Review questions</i>	59
	<i>Review exercises</i>	60

4. Approximations and Errors in Computing

- 4.1 Introduction 61
- 4.2 Significant Digits 62
- 4.3 Inherent Errors 64
 - Data errors* 64
 - Conversion errors* 64
- 4.4 Numerical Errors 65
 - Roundoff errors* 65
 - Chopping* 65
 - Symmetric roundoff* 66
 - Truncation errors* 67
- 4.5 Modelling Errors 69
- 4.6 Blunders 69
- 4.7 Absolute and Relative Errors 70
- 4.8 Machine Epsilon 71
- 4.9 Error Propagation 73
 - Addition and subtraction* 73
 - Multiplication* 73
 - Division* 74
 - Sequence of computations* 76
 - Addition of a chain of numbers* 77
 - Polynomial functions* 79
- 4.10 Conditioning and Stability 80
- 4.11 Convergence of Iterative Processes 86
- 4.12 Error Estimation 87
- 4.13 Minimising the Total Error 88
- 4.14 Pitfalls and Precautions 88
- 4.15 Summary 89
 - Key terms* 90
 - Review questions* 90
 - Review exercises* 91

5. FORTRAN 77 Overview

- 5.1 Need and Scope 93
- 5.2 A Sample Program 93
- 5.3 FORTRAN Constants 97
- 5.4 FORTRAN Variables 98
- 5.5 Subscripted Variables 99
- 5.6 Input/Output Statements 100
- 5.7 Computations 101
 - Mixed-mode expressions* 102
- 5.8 Control of Execution 103
 - Block if-else structure* 103
 - Relational expressions* 104
 - Logical expressions* 105
 - Do-while structure* 106

5.9	Subprograms	109
	<i>Function subprograms</i>	109
	<i>Subroutine subprogram</i>	110
5.10	Intrinsic Functions	112
5.11	Debugging, Testing and Documentation	113
5.12	Summary	114
	<i>Key terms</i>	114
	<i>Review questions</i>	115
	<i>Review exercises</i>	116

✓ 6. Roots of Nonlinear Equations

121

6.1	Introduction	121
	<i>Algebraic equations</i>	122
	<i>Polynomial equations</i>	122
	<i>Transcendental equations</i>	122
6.2	Methods of Solution	123
6.3	Iterative Methods	124
6.4	Starting and Stopping an Iterative Process	125
	<i>Starting the process</i>	125
	<i>Stopping criterion</i>	127
6.5	Evaluation of Polynomials	127
	<i>Program POLY</i>	129
6.6	Bisection Method	131
	<i>Convergence of bisection method</i>	134
	<i>Program BISECT</i>	134
6.7	False Position Method	138
	<i>False position formula</i>	138
	<i>False position algorithm</i>	139
	<i>Convergence of false position method</i>	140
	<i>Program FALSE</i>	141
6.8	Newton-Raphson Method	145
	<i>Newton-Raphson algorithm</i>	146
	<i>Convergence of Newton-Raphson method</i>	147
	<i>Program NEWTON</i>	148
	<i>Limitations of Newton-Raphson method</i>	151
6.9	Secant Method	151
	<i>Secant algorithm</i>	153
	<i>Convergence of secant method</i>	155
	<i>Program SECANT</i>	157
6.10	Fixed-Point Method	160
	<i>Convergence of fixed point iteration</i>	163
	<i>Program FLXEDP</i>	164
6.11	Determining all Possible Roots	166
6.12	Systems of Nonlinear Equations	168
	<i>Fixed point method</i>	168
	<i>Newton-Raphson method</i>	170
6.13	Roots of Polynomials	173
	<i>Multiple roots</i>	174

- Deflation and synthetic division* 175
- Complex roots* 176
- Purification of roots* 177
- 6.14 Multiple Roots by Newton's Method 178
 - Program MULTIR* 178
- 6.15 Complex Roots by Bairstow Method 183
 - Program COMPR* 188
- 6.16 Muller's Method 195
 - Complex roots* 199
 - Multiple roots* 199
 - Program MULLER* 199
- 6.17 Summary 199
 - Key terms* 200
 - Review questions* 200
 - Review exercises* 202
 - Programming projects* 205

✓ Direct Solution of Linear Equations

206

- 7.1 Need and Scope 206
- 7.2 Existence of Solution 207
 - Unique solution* 208
 - No solution* 208
 - No unique solution* 209
 - Ill-conditioned systems* 209
- 7.3 Solution by Elimination 209
- 7.4 Basic Gauss Elimination Method 212
 - Computational effort* 215
 - Program LEG1* 216
- 7.5 Gauss Elimination with Pivoting 219
 - Program LEG2* 222
- 7.6 Gauss-Jordan Method 228
 - Computational effort* 229
- 7.7 Triangular Factorization Methods 230
 - Dolittle algorithm* 231
 - Program DOLIT* 235
 - Crout algorithm* 240
 - Cholesky method* 241
- 7.8 Round-off Errors and Refinement 242
- 7.9 Ill-Conditioned Systems 244
- 7.10 Matrix Inversion Method 245
 - Computing matrix inverse* 246
 - Condition number* 246
- 7.11 Summary 247
 - Key terms* 247
 - Review questions* 248
 - Review exercises* 249
 - Programming projects* 251

8. Iterative Solution of Linear Equations	252
8.1 Need and Scope	252
8.2 Jacobi Iteration Method	252
<i>Program JACIT</i>	255
8.3 Gauss-Seidel Method	259
<i>Algorithm</i>	261
<i>Program GASIT</i>	262
8.4 Method of Relaxation	266
8.5 Convergence of Iteration Methods	267
<i>Condition for convergence</i>	267
<i>Rate of convergence</i>	268
8.6 Summary	272
<i>Key terms</i>	272
<i>Review questions</i>	272
<i>Review exercises</i>	273
<i>Programming projects</i>	274
9. Curve Fitting: Interpolation	275
9.1 Introduction	275
9.2 Polynomial Forms	277
9.3 Linear Interpolation	279
9.4 Lagrange Interpolation Polynomial	281
<i>Program LAGRAN</i>	284
9.5 Newton Interpolation Polynomial	286
9.6 Divided Difference Table	290
<i>Program NEWINT</i>	292
9.7 Interpolation with Equidistant Points	294
<i>Forward difference table</i>	296
<i>Backward difference table</i>	298
9.8 Spline Interpolation	299
<i>Cubic splines</i>	302
<i>Algorithm</i>	307
<i>Program SPLINE</i>	308
<i>Equidistant knots</i>	312
9.9 Chebyshev Interpolation Polynomial	314
<i>Chebyshev points</i>	314
<i>Chebyshev polynomials</i>	315
9.10 Summary	315
<i>Key terms</i>	316
<i>Review questions</i>	317
<i>Review exercises</i>	317
<i>Programming projects</i>	321
10. Curve Fitting: Regression	323
10.1 Introduction	323
10.2 Fitting Linear Equations	324
<i>Least squares regression</i>	325
<i>Algorithm</i>	327
<i>Program LINREG</i>	327

- 10.3 Fitting Transcendental Equations 329
- 10.4 Fitting a Polynomial Function 333
 - Algorithm for polynomial fit* 336
 - Program POLREG* 337
- 10.5 Multiple Linear Regression 341
- 10.6 Ill-Conditioning in Least-Squares Methods 342
- 10.7 Summary 343
 - Key terms* 343
 - Review questions* 343
 - Review exercises* 344
 - Programming projects* 346

11 Numerical Differentiation

347

- 11.1 Need and Scope 347
- 11.2 Differentiating Continuous Functions 348
 - Forward difference quotient* 348
 - Central difference quotient* 349
 - Error analysis* 351
 - Higher-order derivatives* 354
- 11.3 Differentiating Tabulated Functions 355
 - Error analysis* 357
 - Higher-order derivatives* 358
- 11.4 Difference Tables 360
- 11.5 Richardson Extrapolation 362
- 11.6 Summary 364
 - Key terms* 365
 - Review questions* 365
 - Review exercises* 366
 - Programming projects* 368

12 Numerical Integration

369

- 12.1 Need and Scope 369
- 12.2 Newton-Cotes Methods 371
- 12.3 Trapezoidal Rule 372
 - Error analysis* 373
 - Composite trapezoidal rule* 374
 - Program TRAPE1* 377
- 12.4 Simpson's 1/3 Rule 379
 - Error analysis* 380
 - Composite Simpson's 1/3 rule* 382
 - Program SIMS1* 383
- 12.5 Simpson's 3/8 Rule 385
- 12.6 Higher Order Rules 387
- 12.7 Romberg Integration 388
 - Program ROMBRG* 392
- 12.8 Gaussian Integration 394
 - Changing limits of integration* 397
 - Higher-order Gaussian formulae* 399

- 12.9 Summary 401
- Key terms* 401
- Review questions* 402
- Review exercises* 403
- Programming projects* 406

13. Numerical Solution of Ordinary Differential Equations

408

- 13.1 Need and Scope 408
 - Number of independent variables* 410
 - Order of equations* 410
 - Degree of equations* 410
 - Linear and nonlinear equations* 411
 - General and particular solutions* 411
 - Initial value problems* 412
 - One-step and multistep methods* 412
 - Scope* 412
- 13.2 Taylor Series Method 413
 - Improving accuracy* 414
 - Picard's method* 417
- 13.3 Euler's Method 419
 - Accuracy of Euler's method* 421
 - Program EULER* 423
- 13.4 Heun's Method 425
 - Error analysis* 428
 - Program HEUN* 429
- 13.5 Polygon Method 432
 - Program POLYGN* 434
- 13.6 Runge-Kutta Methods 436
 - Determination of weights* 437
 - Fourth-order Runge-Kutta methods* 439
 - Program RUNGE4* 440
- 13.7 Accuracy of One-step Methods 442
- 13.8 Multistep Methods 444
 - Milne-Simpson method* 445
 - Program MILSIM* 447
 - Adams-Bashforth-Moulton method* 450
- 13.9 Accuracy of Multistep Methods 451
 - Milne-Simpson method* 451
 - Adams method* 452
 - Modifiers* 452
- 13.10 Systems of Differential Equations 454
- 13.11 Higher-Order Equations 456
- 13.12 Summary 457
 - Key terms* 458
 - Review questions* 458
 - Review exercises* 460
 - Programming projects* 463

14. Boundary-value and Eigenvalue Problems

- 14.1 Need and Scope 464
- 14.2 Shooting Method 465
- 14.3 Finite Difference Method 469
- 14.4 Solving Eigenvalue Problems 472
- 14.5 Polynomial Method 473
 - The Fadeev-Leverrier method* 474
 - Evaluating the eigenvalues* 475
 - Determining the eigenvectors* 476
 - Computing algorithm* 477
- 14.6 Power Method 477
- 14.7 Summary 479
 - Key terms* 480
 - Review questions* 480
 - Review exercises* 481
 - Programming projects* 482

15. Solution of Partial Differential Equations

483

- 15.1 Need and Scope 483
- 15.2 Deriving Difference Equations 484
- 15.3 Elliptic Equations 486
 - Laplace's equation* 486
 - Liebmann's iterative method* 489
 - Poisson's equation* 490
- 15.4 Parabolic Equations 492
 - Solution of heat equation* 493
 - Bender-Schmidt method* 494
 - The Crank-Nicholson method* 495
- 15.5 Hyperbolic Equations 497
 - Solution of hyperbolic equations* 498
 - Starting values* 499
- 15.6 Summary 500
 - Key terms* 501
 - Review questions* 501
 - Review exercises* 502
 - Programming projects* 504

Appendix A: Solution of Linear Systems by Matrix Methods

505

- A.1 Overview of Matrices 505
- A.2 Solution of Linear Systems by Determinants 509
- A.3 Solution of Linear Systems by Matrix Inversion 510
- A.4 Gauss-Jordan Matrix Inversion 513

Appendix B: Solution of Polynomials by Graeffe's Root

516

Squaring Method

Appendix C: Difference Operators and Central Difference Interpolation Formulae	520
C.1 Introduction	520
C.2 Finite Differences	520
C.3 Difference Operators	522
C.4 Relations between the Operators	523
C.5 Central Difference Interpolation Formulae	525
Appendix D: C Programs	531
D.1 Introduction	531
D.2 FORTRAN to C Conversion	533
Appendix E: Bibliography	600
E.1 Numerical Computing	600
E.2 Programming	601
Index	602