

1. Record Nr.	UNINA9910830712603321
Autore	Yang Won Y
Titolo	Applied Numerical Methods Using MATLAB
Pubbl/distr/stampa	Newark : , : John Wiley & Sons, Incorporated, , 2020 ©2020
ISBN	1-5231-5509-4 1-119-62682-X 1-119-62687-0 1-119-62671-4
Edizione	[2nd ed.]
Descrizione fisica	1 online resource (653 pages)
Altri autori (Persone)	CaoWenwu KimJaekwon ParkKyung W ParkHo-Hyun JoungJingon RoJong-Suk LeeHan L HongCheol-Ho ImTaeho
Disciplina	518
Soggetti	Numerical analysis - Data processing
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di contenuto	Cover -- Title Page -- Copyright -- Contents -- Preface -- Acknowledgments -- About the Companion Website -- Chapter 1 MATLAB Usage and Computational Errors -- 1.1 Basic Operations of MATLAB -- 1.1.1 Input/Output of Data from MATLAB Command Window -- 1.1.2 Input/Output of Data Through Files -- 1.1.3 Input/Output of Data Using Keyboard -- 1.1.4 TwoDimensional (2D) Graphic Input/Output -- 1.1.5 Three Dimensional (3D) Graphic Output -- 1.1.6 Mathematical Functions -- 1.1.7 Operations on Vectors and Matrices -- 1.1.8 Random Number Generators -- 1.1.9 Flow Control -- 1.2 Computer Errors vs. Human Mistakes -- 1.2.1 IEEE 64bit Floating Point Number Representation -- 1.2.2 Various Kinds of Computing

Errors -- 1.2.3 Absolute/Relative Computing Errors -- 1.2.4 Error Propagation -- 1.2.5 Tips for Avoiding Large Errors -- 1.3 Toward Good Program -- 1.3.1 Nested Computing for Computational Efficiency -- 1.3.2 Vector Operation vs. Loop Iteration -- 1.3.3 Iterative Routine vs. Recursive Routine -- 1.3.4 To Avoid Runtime Error -- 1.3.5 Parameter Sharing via GLOBAL Variables -- 1.3.6 Parameter Passing Through VARARGIN -- 1.3.7 Adaptive Input Argument List -- Chapter 2 System of Linear Equations -- 2.1 Solution for a System of Linear Equations -- 2.1.1 The Nonsingular Case ( $M = N$ ) -- 2.1.2 The Underdetermined Case ( $M < N$ ): Minimumnorm Solution -- 2.1.3 The Overdetermined Case ( $M > N$ ): Leastsquares Error Solution -- 2.1.4 Recursive LeastSquares Estimation (RLSE) -- 2.2 Solving a System of Linear Equations -- 2.2.1 Gauss(ian) Elimination -- 2.2.2 Partial Pivoting -- 2.2.3 GaussJordan Elimination -- 2.3 Inverse Matrix -- 2.4 Decomposition (Factorization) -- 2.4.1 LU Decomposition (Factorization) - Triangularization -- 2.4.2 Other Decomposition (Factorization) - Cholesky, QR and SVD -- 2.5 Iterative Methods to Solve Equations.

2.5.1 Jacobi Iteration -- 2.5.2 GaussSeidel Iteration -- 2.5.3 The Convergence of Jacobi and GaussSeidel Iterations -- Chapter 3 Interpolation and Curve Fitting -- 3.1 Interpolation by Lagrange Polynomial -- 3.2 Interpolation by Newton Polynomial -- 3.3 Approximation by Chebyshev Polynomial -- 3.4 Pade Approximation by Rational Function -- 3.5 Interpolation by Cubic Spline -- 3.6 Hermite Interpolating Polynomial -- 3.7 TwoDimensional Interpolation -- 3.8 Curve Fitting -- 3.8.1 StraightLine Fit - A Polynomial Function of Degree 1 -- 3.8.2 Polynomial Curve Fit - A Polynomial Function of Higher Degree -- 3.8.3 Exponential Curve Fit and Other Functions -- 3.9 Fourier Transform -- 3.9.1 FFT vs. DFT -- 3.9.2 Physical Meaning of DFT -- 3.9.3 Interpolation by Using DFS -- Chapter 4 Nonlinear Equations -- 4.1 Iterative Method toward Fixed Point -- 4.2 Bisection Method -- 4.3 False Position or Regula Falsi Method -- 4.4 Newton(Raphson) Method -- 4.5 Secant Method -- 4.6 Newton Method for a System of Nonlinear Equations -- 4.7 Bairstow's Method for a Polynomial Equation -- 4.8 Symbolic Solution for Equations -- 4.9 Real World Problems -- Chapter 5 Numerical Differentiation/Integration -- 5.1 Difference Approximation for the First Derivative -- 5.2 Approximation Error of the First Derivative -- 5.3 Difference Approximation for Second and Higher Derivative -- 5.4 Interpolating Polynomial and Numerical Differential -- 5.5 Numerical Integration and Quadrature -- 5.6 Trapezoidal Method and Simpson Method -- 5.7 Recursive Rule and Romberg Integration -- 5.8 Adaptive Quadrature -- 5.9 Gauss Quadrature -- 5.9.1 GaussLegendre Integration -- 5.9.2 GaussHermite Integration -- 5.9.3 GaussLaguerre Integration -- 5.9.4 GaussChebyshev Integration -- 5.10 Double Integral -- 5.11 Integration Involving PWL Function -- Chapter 6 Ordinary Differential Equations.

6.1 Euler's Method -- 6.2 Heun's Method - Trapezoidal Method -- 6.3 RungeKutta Method -- 6.4 PredictorCorrector Method -- 6.4.1 AdamsBashforthMoulton Method -- 6.4.2 Hamming Method -- 6.4.3 Comparison of Methods -- 6.5 Vector Differential Equations -- 6.5.1 State Equation -- 6.5.2 Discretization of LTI State Equation -- 6.5.3 Highorder Differential Equation to State Equation -- 6.5.4 Stiff Equation -- 6.6 Boundary Value Problem (BVP) -- 6.6.1 Shooting Method -- 6.6.2 Finite Difference Method -- Chapter 7 Optimization -- 7.1 Unconstrained Optimization -- 7.1.1 Golden Search Method -- 7.1.2 Quadratic Approximation Method -- 7.1.3 NelderMead Method -- 7.1.4 Steepest Descent Method -- 7.1.5 Newton Method -- 7.1.6

Conjugate Gradient Method -- 7.1.7 Simulated Annealing -- 7.1.8 Genetic Algorithm -- 7.2 Constrained Optimization -- 7.2.1 Lagrange Multiplier Method -- 7.2.2 Penalty Function Method -- 7.3 MATLAB BuiltIn Functions for Optimization -- 7.3.1 Unconstrained Optimization -- 7.3.2 Constrained Optimization -- 7.3.3 Linear Programming (LP) -- 7.3.4 Mixed Integer Linear Programming (MILP) -- 7.4 Neural Network[K1] -- 7.5 Adaptive Filter[Y3] -- 7.6 Recursive Least Square Estimation (RLSE)[Y3] -- Chapter 8 Matrices and Eigenvalues -- 8.1 Eigenvalues and Eigenvectors -- 8.2 Similarity Transformation and Diagonalization -- 8.3 Power Method -- 8.3.1 Scaled Power Method -- 8.3.2 Inverse Power Method -- 8.3.3 Shifted Inverse Power Method -- 8.4 Jacobi Method -- 8.5 GramSchmidt Orthonormalization and QR Decomposition -- 8.6 Physical Meaning of Eigenvalues/Eigenvectors -- 8.7 Differential Equations with Eigenvectors -- 8.8 DoA Estimation with Eigenvectors[Y-3] -- Chapter 9 Partial Differential Equations -- 9.1 Elliptic PDE -- 9.2 Parabolic PDE -- 9.2.1 The Explicit Forward Euler Method -- 9.2.2 The Implicit Backward Euler Method. 9.2.3 The CrankNicholson Method -- 9.2.4 Using the MATLAB function 'pdepe()' -- 9.2.5 TwoDimensional Parabolic PDEs -- 9.3 Hyperbolic PDES -- 9.3.1 The Explicit Central Difference Method -- 9.3.2 Two Dimensional Hyperbolic PDEs -- 9.4 Finite Element Method (FEM) for Solving PDE -- 9.5 GUI of MATLAB for Solving PDES - PDEtool -- 9.5.1 Basic PDEs Solvable by PDEtool -- 9.5.2 The Usage of PDEtool -- 9.5.3 Examples of Using PDEtool to Solve PDEs -- Appendix A Mean Value Theorem -- Appendix B Matrix Operations/Properties -- B.1 Addition and Subtraction -- B.2 Multiplication -- B.3 Determinant -- B.4 Eigenvalues and Eigenvectors of a Matrix1 -- B.5 Inverse Matrix -- B.6 Symmetric/Hermitian Matrix -- B.7 Orthogonal/Unitary Matrix -- B.8 Permutation Matrix -- B.9 Rank -- B.10 Row Space and Null Space -- B.11 Row Echelon Form -- B.12 Positive Definiteness -- B.13 Scalar (Dot) Product and Vector (Cross) Product -- B.14 Matrix Inversion Lemma -- Appendix C Differentiation W.R.T. A Vector -- Appendix D Laplace Transform -- Appendix E Fourier Transform -- Appendix F Useful Formulas -- Appendix G Symbolic Computation -- G.1 How to Declare Symbolic Variables and Handle Symbolic Expressions -- G.2 Calculus -- G.2.1 Symbolic Summation -- G.2.2 Limits -- G.2.3 Differentiation -- G.2.4 Integration -- G.2.5 Taylor Series Expansion -- G.3 Linear Algebra -- G.4 Solving Algebraic Equations -- G.5 Solving Differential Equations -- Appendix H Sparse Matrices -- Appendix I MATLAB -- References -- Index -- Index for MATLAB Functions -- Index for Tables -- EULA.

---

## Sommario/riassunto

"This book makes use of MATLAB software to teach the fundamental concepts using the software to solve practical engineering and/or science problems. The programs are presented in a complete form so that readers can run them instantly with no programming skill, allowing them to focus on understanding the mathematical manipulation process and making interpretations of the results. The book targets students who do not like and/or do not have time to derive and prove mathematical results, helping them develop their problem-solving capability without being involved in details about the MATLAB codes. It also targets students who want to delve into details, helping them understand underlying algorithms and equations as deeply as they want"--

---