I'm not robot
reCAPTCHA
Continue

Introduction to numerical analysis solutions manual pdf

Published on Feb 22, 2019Full download : Solutions Manual for Friendly Introduction To Numerical Analysis 1st Edition by Bradie, Friendly Introduction To... WillowFollow this publisher - current follower count:0 As the Solutions Manual, this book is meant to accompany the main title,Numerical Methods and Analysis, Second Edition.The objective of this book is for readers to learn where approximation methods come from, why they work, why they sometimes don't work, and when to use which of the many techniques that are available, and to do all this in an environmentthat emphasizes readability and usefulness to the numerical methods novice. Each chapter and each section begins with the basic, elementary material and gradually builds up to more advanced topics. The text begins with a review of the important calculus results, and why and where these ideas play an important role throughout the book. Some of the concepts required for the study of computational mathematics are introduced, and simple approximations using Taylor's Theorem are treated in some depth. The exposition is intended to be lively and "student friendly". Exercises run the gamut from simple hand computations that might be characterized are "starter exercises", to challenging derivations and minor proofs, to programming exercises. Eleven new exercises have been added throughout including:Basins of Attraction;Roots of Polynomials I;Radial Basis Function Interpolation;Tension Splines;An Introduction to Galerkin/Finite Element Ideas for BVPs;Broydens Method;Roots of Polynomials, II;Spectral/collocation methods for PDEs;Algebraic Multigrid Method;Trigonometric interpolation/Fourier analysis;andMonte Carlo methods. Various sections have beenrevised to reflect recent trends and updates in the field. A solutions manual to accompany An Introduction to Numerical Methods and Analysis, Second Edition An Introduction to Numerical Methods and Analysis, Second Edition reflects the latest trends in the field, includes new material and revised exercises, and offers a unique emphasis on applications. The author clearly explains how to both construct and evaluate approximations for accuracy and performance, which are key skills in a variety of fields. A wide range of higher-level methods and solutions, including new topics such as the roots of polynomials, spectral collocation, finite element ideas, and Clenshaw-Curtis quadrature, are presented from an introductory perspective, and theSecond Edition also features: Chapters and sections that begin with basic, elementary material followed by gradual coverage of more advanced material Exercises ranging from simple hand computations to challenging derivations and minor proofs to programming exercises Widespread exposure and utilization of MATLAB® An appendix that contains proofs of various theorems and other material Get full access to Solutions Manual to accompany An Introduction to Numerical Methods and Analysis, 3rd Edition and 60K+ other titles, with free 10-day trial of O'Reilly. There's also live online events, interactive content, certification prep materials, and more. A solutions manual to accompany An Introduction to Numerical Methods and Analysis, Third EditionAn Introduction to Numerical Methods and Analysis helps students gain a solid understanding of a wide range of numerical approximation methods for solving problems of mathematical analysis. Designed for entry-level courses on the subject, this popular textbook maximizes teaching flexibility by first covering basic topics before gradually moving to more advanced material in each chapter and section. Throughout the text, students are provided clear and accessible guidance on a wide range of numerical methods and analysis techniques, including root-finding, numerical integration, interpolation, solution of systems of equations, and many others. This fully revised third edition contains new sections on higher-order difference methods, the bisection and inertia method for computing eigenvalues of a symmetric matrix, a completely re-written section on different methods for Poisson equations, and spectral methods for higher-dimensional problems. New problem sets—ranging in difficulty from simple computations to challenging derivations and proofs—are complemented by computer programming exercises, illustrative examples, and sample code. This acclaimed textbook: Explains how to both construct and evaluate approximations for accuracy and performanceCovers both elementary concepts and tools and higher-level methods and solutionsFeatures new and updated material reflecting new trends and applications in the fieldContains an introduction to key concepts, a calculus review, an updated primer on computer arithmetic, a brief history of scientific computing, a survey of computer languages and software, and a revised literature reviewIncludes an appendix of proofs of selected theorems and author-hosted companion website with additional exercises, application models, and supplemental resources VDOC.PUB Authors: James F. Epperson PDF Download Embed This document was uploaded by our user. The uploader already confirmed that they had the permission to publish it. If you are author/publisher or own the copyright of this documents, please report to us by using this DMCA report form. Report DMCA A solutions manual to accompany An Introduction to Numerical Methods and Analysis, Second Edition An Introduction to Numerical Methods and Analysis, Second Edition reflects the latest trends in the field, includes new material and revised exercises, and offers a unique emphasis on applications. The author clearly explains how to both construct and evaluate approximations for accuracy and performance, which are key skills in a variety of fields. A wide range of higher-level methods and solutions, including new topics such as the roots of polynomials, spectral collocation, finite element ideas, and Clenshaw-Curtis quadrature, are presented from an introductory perspective, and theSecond Edition also features: Chapters and sections that begin with basic, elementary material followed by gradual coverage of more advanced material Exercises ranging from simple hand computations to challenging derivations and minor proofs to programming exercises Widespread exposure and utilization of MATLAB An appendix that contains proofs of various theorems and other material E-Book Content Solutions Manual to Accompany AN INTRODUCTION TO NUMERICAL METHODS AND ANALYSIS Solutions Manual to Accompany AN INTRODUCTION TO NUMERICAL METHODS AND ANALYSIS Second Edition JAMES F. EPPERSON Mathematical Reviews Copyright © 2013 by John Wiley & Sons, Inc. All rights reserved. Published by John Wiley & Sons, Inc., Hoboken, New Jersey. Published simultaneously in Canada. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at . Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representation or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages. For general information on our other products and services please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002. Wiley also publishes its books in a variety of electronic formats. Some content that appears in print, however, may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com. Library of Congress Cataloging-in-Publication Data: Epperson, James F., author. An introduction to numerical methods and analysis / James F. Epperson, Mathematical Reviews. — Second edition. pages cm Includes bibliographical references and index. ISBN 978-1-118-36759-9 (hardback) 1. Numerical analysis. I. Title. QA297.E568 2013 518—dc23 2013013979 10 9 8 7 6 5 4 3 2 1 CONTENTS 1 Introductory Concepts and Calculus Review 1.1 1.2 1.3 1.4 1.5 1.6 1.7 2 Basic Tools of Calculus Error, Approximate Equality, and Asymptotic Order Notation A Primer on Computer Arithmetic A Word on Computer Languages and Software Simple Approximations Application: Approximating the Natural Logarithm A Brief History of Computing 1 1 12 15 19 19 22 25 A Survey of Simple Methods and Tools 27 2.1 2.2 2.3 2.4 2.5 2.6 2.7 27 30 40 44 48 56 61 Horner’s Rule and Nested Multiplication Difference Approximations to the Derivative Application: Euler’s Method for Initial Value Problems Linear Interpolation Application — The Trapezoid Rule Solution of Tridiagonal Linear Systems Application: Simple Two-Point Boundary Value Problems v vi CONTENTS 3 Root-Finding 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12 4 65 69 73 77 81 84 88 92 96 99 102 114 Interpolation and Approximation 117 4.1 4.2 4.3 4.4 117 120 132 4.5 4.6 4.7 4.8 4.9 4.10 4.11 4.12 5 The Bisection Method Newton’s Method: Derivation and Examples How to Stop Newton’s Method Application: Division Using Newton’s Method The Newton Error Formula Newton’s Method: Theory and Convergence Application: Computation of the Square Root The Secant Method: Derivation and Examples Fixed Point Iteration Roots of Polynomials (Part 1) Special Topics in Root-finding Methods Very High-order Methods and the Efficiency Index 65 Lagrange Interpolation Newton Interpolation and Divided Differences Interpolation Error Application: Muller’s Method and Inverse Quadratic Interpolation Application: More Approximations to the Derivative Hermite Interpolation Piecewise Polynomial Interpolation An Introduction to Splines Application: Solution of Boundary Value Problems Tension Splines Least Squares Concepts in Approximation Advanced Topics in Interpolation Error 139 141 142 145 149 156 159 160 166 Numerical Integration 171 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 171 173 177 187 190 192 199 203 A Review of the Definite Integral Improving the Trapezoid Rule Simpson’s Rule and Degree of Precision The Midpoint Rule Application: Stirling’s Formula Gaussian Quadrature Extrapolation Methods Special Topics in Numerical Integration CONTENTS 6 7 Numerical Methods for Ordinary Differential Equations 211 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 6.10 211 213 216 217 225 228 234 235 240 243 9 The Initial Value Problem — Background Euler’s Method Analysis of Euler’s Method Variants of Euler’s Method Single Step Methods — Runge-Kutta Multi-step Methods Stability Issues Application to Systems of Equations Adaptive Solvers Boundary Value Problems Numerical Methods for the Solution of Systems of Equations 247 7.1 7.2 7.3 7.4 7.5 7.6 7.7 7.8 7.9 8 vii Linear Algebra Review Linear Systems and Gaussian Elimination Operation Counts The LU Factorization Perturbation, Conditioning and Stability SP D Matrices and the Cholesky Decomposition Iterative Methods for Linear Systems – A Brief Survey Nonlinear Systems: Newton’s Method and Related Ideas Application: Numerical Solution of Nonlinear BVP’s 247 248 254 256 262 269 271 273 275 Approximate Solution of the Algebraic Eigenvalue Problem 277 8.1 8.2 8.3 8.4 8.5 277 280 281 284 288 Eigenvalue Review Reduction to Hessenberg Form Power Methods An Overview of the QR Iteration Application: Roots of Polynomials, II A Survey of Numerical Methods for Partial Differential Equations 289 9.1 9.2 9.3 289 293 294 Difference Methods for the Diffusion Equation Finite Element Methods for the Diffusion Equation Difference Methods for Poisson Equations viii 10 CONTENTS An Introduction to Spectral Methods 299 10.1 10.2 10.3 299 301 303 Spectral Methods for Two-Point Boundary Value Problems Spectral Methods for Time-Dependent Problems Clenshaw-Curtis Quadrature Preface to the Solutions Manual This manual is written for instructors, not students. It includes worked solutions for many (roughly 75%) of the problems in the text. For the computational exercises I have given the output generated by my program, or sometimes a program listing. Most of the programming was done in MATLAB, some in FORTRAN. (The author is well aware that FORTRAN is archaic, but there is a lot of “legacy code" in FORTRAN, and the author believes there is value in learning a new language, even an archaic one.) When the text has a series of exercises that are obviously similar and have similar solutions, then sometimes only one of these problems has a worked solution included. When computational results are asked for a series of similar functions or problems, only a subset of solutions are reported, largely for the sake of brevity. Some exercises that simply ask the student to perform a straight-forward computation are skipped. Exercises that repeat the same computation but with a different method are also often skipped, as are exercises that ask the student to “verify” a straight-forward computation. Some of the exercises were designed to be open-ended and almost “essay-like.” For these exercises, the only solution typically provided is a short hint or brief outline of the kind of discussion anticipated by the author. In many exercises the student needs to construct an upper bound on a derivative of some function in order to determine how small a parameter has to be to achieve a ix x desired level of accuracy. For many of the solutions this was done using a computer algebra package and the details are not given. Students who acquire a copy of this manual in order to obtain worked solutions to homework problems should be aware that none of the solutions are given in enough detail to earn full credit from an instructor. The author freely admits the potential for error in any of these solutions, especially since many of the exercises were modified after the final version of the text was submitted to the publisher and because the ordering of the exercises was changed from the Revised Edition to the Second Edition. While we tried to make all the appropriate corrections, the possibility of error is still present, and undoubtedly the author’s responsibility. Because much of the manual was constructed by doing “copy-and-paste” from the files for the text, the enumeration of many tables and figures will be different. I have tried to note what the number is in the text, but certainly may have missed some instances. Suggestions for new exercises and corrections to these solutions are very welcome. Contact the author at or Differences from the text The text itself went through a copy-editing process after this manual was completed. As was to be expected, the wording of several problems was slightly changed. None of these changes should affect the problem in terms of what is expected of students; the vast majority of the changes were to replace “previous problem” (a bad habit of mine) with “Problem X.Y” (which I should have done on my own, in the first place). Some puncuation was also changed. The point of adding this note is to explain the textual differences which might be noticed between the text and this manual. If something needs clarification, please contact me at the above email. CHAPTER 1 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW 1.1 BASIC TOOLS OF CALCULUS Exercises: 1. Show that the third order Taylor polynomial for f (x) = (x + 1)−1 , about x0 = 0, is p3 (x) = 1 − x + x2 − x3 . Solution: We have f (0) = 1 and f 0 (x) = − 1 , (x + 1)2 f 00 (x) = 2 , (x + 1)3 f 000 (x) = − 6 , (x + 1)4 so that f 0 (0) = −1, f 00 (0) = 2, f 000 = −6. Therefore p3 (x) 1 1 f (0) + xf 0 (0) + x2 f 00 (0) + x3 f 000 (x) 2 6 1 2 1 3 = 1 + x(−1) + x (2) + x (−6) 2 6 = 1 − x + x2 − x3 . = Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c 2013 John Wiley & Sons, Inc. Copyright 1 2 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW 2. What is the third order Taylor polynomial for f (x) = √ x + 1, about x0 = 0? Solution: We have f (x0) = 1 and f 0 (x) = 1 , 2(x + 1)1/2 f 00 (x) = − 1 , 4(x + 1)3/2 f 000 (x) = 3 , 8(x + 1)5/2 so that f 0 (0) = 1/2, f 00 (0) = −1/4, f 000 = 3/8. Therefore p3 (x) 1 1 f (0) + xf 0 (0) + x2 f 00 (0) + x3 f 000 (x) 2 6 1 2 1 3 = 1 + x(1/2) + x (−1/4) + x (3/8) 2 6 = 1 − (1/2)x − (1/8)x2 + (1/16)x3 . = 3. What is the sixth order Taylor polynomial for f (x) = Hint: Consider the previous problem. 4. Given that R(x) = √ 1 + x2 , using x0 = 0? |x|6 ξ e 6! for x ∈ [−1, 1], where ξ is between x and 0, find an upper bound for |R|, valid for all x ∈ [−1, 1], that is independent of x and ξ. 5. Repeat the above, but this time require that the upper bound be valid only for all x ∈ [− 12 , 12]. Solution: The only significant difference is the introduction of a factor of 26 in the denominator: √ e = 3.6 × 10−5 . |R(x)| ≤ 6 2 × 720 6. Given that |x|4 R(x) = 4! −1 1+ξ for x ∈ [− 12 , 21], where ξ is between x and 0, find an upper bound for |R|, valid for all x ∈ [− 12 , 12], that is independent of x and ξ. √ 7. Use a Taylor polynomial to find an approximate value for e that is accurate −3 to within 10 . x Solution: There’s two ways to do this. We can √ approximate f (x) = e and use x = 1/2, or we can approximate g(x) = x and use x = e. In addition, we can be conventional and take x0 = 0, or we can take x0 6= 0 in order to speed convergence. BASIC TOOLS OF CALCULUS 3 The most straightforward approach (in my opinion) is to use a Taylor polynomial for ex about x0 = 0. The remainder after k terms is Rk (x) = xk+1 ξ e . (k + 1)! We quickly have that |Rk (x)| ≤ e1/2 + 1)! 2k+1 (k and a little playing with a calculator shows that |R3 (x)| ≤ but |R4 (x)| ≤ e1/2 = 0.0043 16 × 24 e1/2 = 4.3 × 10−4 . 32 × 120 So we would use 2 3 4 1 1 1 1 1 1 1 e ≈1+ + + + = 1.6484375. 2 2 2 6 2 24 2 √ To fourteen digits, e = 1.64872127070013, and the error is 2.84 × 10−4 , much smaller than required. 1/2 8. What is the fourth order Taylor polynomial for f (x) = 1/(x + 1), about x0 = 0? Solution: We have f (0) = 1 and f 0 (x) = − 1 , (x + 1)2 f 00 (x) = 2 , (x + 1)3 f 000 (x) = − 6 , (x + 1)4 f 0000 (x) = so that f 0 (0) = −1, f 00 (0) = 2, f 000 = −6, f 0000 (0) = 24. Thus 1 1 1 p4 (x) = 1+x(−1)+ x2 (2)+ x3 (−6)+ x4 (24) = 1−x+x2 −x3 +x4 . 2 6 24 9. What is the fourth order Taylor polynomial for f (x) = 1/x, about x0 = 1? 10. Find the Taylor polynomial of third order for sin x, using: (a) x0 = π/6. Solution: We have 1 f (x0) = , 2 √ 3 f (x0) = , 2 0 1 f (x0) = − , 2 00 √ 000 f (x0) = − 3 , 2 24 (x + 1)5 4 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW so √ √ 1 3 3 π 1 π 2 π 3 p3 (x) = + − . x− − x− x− 2 2 6 4 6 12 6 (b) x0 = π/4; (c) x0 = π/2; 11. For each function below construct the third-order Taylor polynomial approximation, using x0 = 0, and then estimate the error by computing an upper bound on the remainder, over the given interval. (a) f (x) = e−x , x ∈ [0, 1]; (b) f (x) = ln(1 + x), x ∈ [−1, 1]; (c) f (x) = sin x, x ∈ [0, π]; (d) f (x) = ln(1 + x), x ∈ [−1/2, 1/2]; (e) f (x) = 1/(x + 1), x ∈ [−1/2, 1/2]. Solution: (a) The polynomial is 1 1 p3 (x) = 1 − x + x2 − x3 , 2 6 with remainder 1 4 −ξ x e . 24 This can be bounded above, for all x ∈ [0, 1], by R3 (x) = |R3 (x)| ≤ 1 e 24 (b) The polynomial is 1 1 p3 (x) = x − x2 + x3 , 2 3 with remainder R3 (x) = 1 1 4 x 4 (1 + ξ)4 We can’t bound this for all x ∈ [−1, 1], because of the potential division by zero. (c) The polynomial is 1 p3 (x) = x − x3 6 BASIC TOOLS OF CALCULUS 5 with remainder R3 (x) = 1 5 x cos ξ. 120 This can be bounded above, for all x ∈ [0, π], by |R3 (x)| ≤ π5 . 120 (d) The polynomial is the same as in (b), of course, 1 1 p3 (x) = x − x2 + x3 , 2 3 with remainder R3 (x) = 1 4 1 x 4 (1 + ξ)4 For all x ∈ [−1/2, 1/2] this can be bounded by R3 (x) ≤ 1 1 1 (1/24) = . 4 4 (1 − (1/2)) 4 (e) The polynomial is p3 (x) = 1 − x + x2 − x3 , with remainder R3 (x) = x4 1 (1 + ξ)5 This can be bounded above, for all x ∈ [−1/2, 1/2], by |R3 (x)| ≤ (1/2)4 1 = 2. (1 − 1/2)5 Obviously, this is not an especialy good approximation. 12. Construct a Taylor polynomial approximation that is accurate to within 10−3 , over the indicated interval, for each of the following functions, using x0 = 0. (a) f (x) = sin x, x ∈ [0, π]; (b) f (x) = e−x , x ∈ [0, 1]; (c) f (x) = ln(1 + x), x ∈ [−1/2, 1/2]; (d) f (x) = 1/(x + 1), x ∈ [−1/2, 1/2]; (e) f (x) = ln(1 + x), x ∈ [−1, 1]. 6 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW Solution: (a) The remainder here is Rn (x) = (−1)n+1 2n+1 x cos c, (2n + 1)! for c ∈ [0, π]. Therefore, we have |Rn (x)| ≤ π 2n+1 1 |π|2n+1 ≤ . (2n + 1)! (2n + 1)! Simple manipulations with a calculator then show that max |R6 (x)| ≤ .4663028067e − 3 x∈[0,π] but max |R5 (x)| ≤ .7370430958e − 2. x∈[0,π] Therefore the desired Taylor polynomial is 1 1 5 1 1 1 11 p11 (x) = 1 − x + x3 − x − x7 + x9 + x . 6 120 7! 9! 11! (b) The remainder here is Rn (x) = (−1)n+1 n+1 −c x e , (n + 1)! for c ∈ [0, 1]. Therefore, we have |Rn (x)| ≤ 1 1 |x|n+1 ≤ . (n + 1)! (n + 1)! Simple manipulations with a calculator then show that max |R6 (x)| ≤ .0001984126984 x∈[0,1] but max |R5 (x)| ≤ .1388888889e − 2 x∈[0,1] Therefore the desired Taylor polynomial is 1 1 1 5 1 6 1 x + x . p6 (x) = 1 − x + x2 − x3 + x4 − 2 6 24 120 720 (c) f (x) = ln(1 + x), x ∈ [0, 3/4]. BASIC TOOLS OF CALCULUS 7 Solution: The remainder is now |Rn (x)| ≤ (1/2)n+1 , (n + 1) and n = 8 makes the error small enough. (d) f (x) = ln(1 + x), x ∈ [0, 1/2]. 13. Repeat the above, this time with a desired accuracy of 10−6 . 14. Since π = arctan 1, 4 we can estimate π by estimating arctan 1. How many terms are needed in the Gregory series for the arctangent to approximate π to 100 decimal places? 1,000? Hint: Use the error term in the Gregory series to predict when the error gets sufficiently small. Solution: The remainder in the Gregory series approximation is Z x 2n+2 t Rn (x) = (−1)n+1 dt, 2 0 1+t so to get 100 decimal places of accuracy for x = 1, we require Z 1 2n+2 Z 1 1 t t2n+2 dt = dt ≤ ≤ 10−100 , |Rn (1)| = 2 1 + t 2n +3 0 0 thus, we have to take n ≥ (10100 − 3)/2 terms. For 1,000 places of accuracy we therefore need n ≥ (101000 − 3)/2 terms. Obviously this is not the best procedure for computing many digits of π! 15. Elementary trigonometry can be used to show that arctan(1/239) = 4 arctan(1/5) − arctan(1). This formula was developed in 1706 by the English astronomer John Machin. Use this to develop a more efficient algorithm for computing π. How many terms are needed to get 100 digits of accuracy with this form? How many terms are needed to get 1,000 digits? Historical note: Until 1961 this was the basis for the most commonly used method for computing π to high accuracy. Solution: We now have two Gregory series, thus complicating the problem a bit. We have π = 4 arctan(1) = 16 arctan(1/5) − 4 arctan(1/239); Define pm,n ≈ π as the approximation generated by using an m term Gregory series to approximate arctan(1/5) and an n term Gregory series for arctan(1/239). Then we have pm,n − π = 16Rm (1/5) − 4Rn (1/239), 8 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW where Rk is the remainder in the Gregory series. Therefore Z 1/5 2m+2 Z 1/239 2n+2 t t m+1 n+1 |pm,n − π| ≤ 16(−1) dt − 4(−1) dt 2 2 1 + t 1 + t 0 0 ≤ 4 16 + . (2m + 3)52m+3 (2n + 3)2392n+3 To finish the problem we have to apportion the error between the two series, which introduces some arbitrariness into the the problem. If we require that they be equally accurate, then we have that 16 ≤ (2m + 3)52m+3 and 4 ≤ . (2n + 3)2392n+3 Using properties of logarithms, these become log(2m + 3) + (2m + 3) log 5 ≥ log 16 − log and log(2n + 3) + (2n + 3) log 239 ≥ log 4 − log . For = (1/2) × 10−100 these are satisfied for m = 70, n = 20. For = (1/2) × 10−1000 we get m = 712, n = 209. Changing the apportionment of the error doesn’t change the results by much at all. 16. In 1896 a variation on Machin’s formula was found: arctan(1/239) = arctan(1) − 6 arctan(1/8) − 2 arctan(1/57), and this began to be used in 1961 to compute π to high accuracy. How many terms are needed when using this expansion to get 100 digits of π? 1,000 digits? Solution: We now have three series to work with, which complicates matters only slightly more compared to the previous problem. If we define pk,m,n ≈ π based on π = 4 arctan(1) = 24 arctan(1/8) + 8 arctan(1/57) + 4 arctan(1/239), taking k terms in the series for arctan(1/8), m terms in the series for arctan(1/57), and n terms in the series for arctan(1/239), then we are led to the inequalities log(2k + 3) + (2k + 3) log 8 ≥ log 24 − log , log(2m + 3) + (2m + 3) log 57 ≥ log 8 − log , BASIC TOOLS OF CALCULUS 9 and log(2n + 3) + (2n + 3) log 239 ≥ log 4 − log . For = (1/3) × 10−100 we get k = 54, m = 27, and n = 19; for = (1/3) × 10−1000 we get k = 552, m = 283, and n = 209. Note: In both of these problems a slightly more involved treatment of the error might lead to fewer terms being required. 17. What is the Taylor polynomial of order 3 for f (x) = x4 + 1, using x0 = 0? Solution: This is very direct: f 0 (x) = 4x3 , so that f 00 (x) = 12x2 , f 000 (x) = 24x, 1 1 p3 (x) = 1 + x(0) + x2 (0) + x3 (0) = 1. 2 6 18. What is the Taylor polynomial of order 4 for f (x) = x4 + 1, using x0 = 0? Simplify as much as possible. 19. What is the Taylor polynomial of order 2 for f (x) = x3 + x, using x0 = 1? 20. What is the Taylor polynomial of order 3 for f (x) = x3 + x, using x0 = 1? Simplify as much as possible. Solution: We note that f 000 (1) = 6, so we have (using the solution from the previous problem) 1 p4 (x) = 3x2 − 2x + 1 + (x − 1)3 (6) = x3 + x. 6 The polynomial is its own Taylor polynomial. 21. Let p(x) be an arbitrary polynomial of degree less than or equal to n. What is its Taylor polynomial of degree n, about an arbitrary x0 ? 22. The Fresnel integrals are defined as Z C(x) = x cos(πt2 /2)dt 0 and Z S(x) = x sin(πt2 /2)dt. 0 Use Taylor expansions to find approximations to C(x) and S(x) that are 10−4 accurate for all x with |x| ≤ 12 . Hint: Substitute x = πt2 /2 into the Taylor expansions for the cosine and sine. 10 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW Solution: We will show the work for the case of S(x), only. We have Z x Z x Z x 2 2 Rn (t2)dt. pn (t)dt + sin(πt /2)dt = S(x) = 0 0 0 Looking more carefully at the remainder term, we see that it is given by Z x 2(2n+3) (t rn (x) = ± cos ξdt. 0 (2n + 3)! Therefore, Z |rn (x)| ≤ 0 1/2 (t2(2n+3) (1/2)4n+7 dt = . (2n + 3)! (4n + 7)(2n + 3)! A little effort with a calculator shows that this is less than 10−4 for n ≥ 1, therefore the polynomial is Z x x7 x3 (t2 − (1/6)t6)dt = − + . p(x) = 42 3 0 23. Use the Integral Mean Value Theorem to show that the “pointwise” form (1.3) of the Taylor remainder (usually called the Lagrange form) follows from the “integral” form (1.2) (usually called the Cauchy form). 24. For each function in Problem 11, use the Mean Value Theorem to find a value M such that |f (x1) − f (x2)| ≤ M |x1 − x2 | is valid for all x1 , x2 in the interval used in Problem 11. Solution: This amounts to finding an upper bound on |f 0 | over the interval given. The answers are as given below. (a) f (x) = e−x , x ∈ [0, 1]; M ≤ 1. (b) f (x) = ln(1+x), x ∈ [−1, 1]; M is unbounded, since f 0 (x) = 1/(1+x) and x = −1 is possible. (c) f (x) = sin x, x ∈ [0, π]; M ≤ 1. (d) f (x) = ln(1 + x), x ∈ [−1/2, 1/2]; M ≤ 2. (e) f (x) = 1/(x + 1), x ∈ [−1/2, 1/2]. M ≤ 4. 25. A function is called monotone on an interval if its derivative is strictly positive or strictly negative on the interval. Suppose f is continuous and monotone on the interval [a, b], and f (a)f (b) < 0; prove that there is exactly one value α ∈ [a, b] such that f (α) = 0. Solution: Since f is continuous on the interval [a, b] and f (a)f (b) < 0, the Intermediate Value Theorem guarantees that there is a point c where f (c) = 0, BASIC TOOLS OF CALCULUS 11 i.e., there is at least one root. Suppose now that there exists a second root, γ. Then f (c) = f (γ) = 0. By the Mean Value Theorem, then, there is a point ξ between c and γ such that f (γ) − f (c) = 0. γ−c f 0 (ξ) = But this violates the hypothesis that f is monotone, since a monotone function must have a derivative that is strictly positive or strictly negative. Thus we have a contradiction, thus there cannot exist the second root. A very acceptable argument can be made by appealing to a graph of the function. 26. Finish the proof of the Integral Mean Value Theorem (Theorem 1.5) by writing up the argument in the case that g is negative. Solution: All that is required is to observe that if g is negative, then we have b Z b Z g(t)f (t)dt ≤ Z g(t)f (t)dt ≤ Z g(t)dt a a a and b b Z g(t)f (t)dt ≥ a b Z g(t)fm dt = fm Z b g(t)dt. g(t)fM dt = fM a a The proof is completed as in the text. 27. Prove Theorem 1.6, providing all details. 28. Let ck > 0, be given, 1 ≤ k ≤ n, and let xk ∈ [a, b], 1 ≤ k ≤ n. Then, use the Discrete Average Value Theorem to prove that, for any function f ∈ C([a, b]), Pn c f (xk) k=1 Pn k = f (ξ) k=1 ck for some ξ ∈ [a, b]. Solution: We can’t apply the Discrete Average Value Theorem to the problem as it is posed originally, so we have to manipulate a bit. Define cj γj = Pn k=1 ck Then n X ; γj = 1 j=1 and now we can apply the Discrete Average Value Theorem to finish the problem. 12 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW 29. Discuss, in your own words, whether or not the following statement is true: “The Taylor polynomial of degree n is the best polynomial approximation of degree n to the given function near the point x0 .” /•••. 1.2 ERROR, APPROXIMATE EQUALITY, AND ASYMPTOTIC ORDER NOTATION Exercises: 1. Use Taylor’s Theorem to show that ex = 1 + x + O(x2) for x sufficiently small. 2. Use Taylor’s Theorem to show that small. 1−cos x x = 12 x + O(x3) for x sufficiently Solution: We can expand the cosine in a Taylor series as 1 1 cos x = 1 − x2 + x4 cos ξ. 2 24 If we substitute this into (1 − cos x)/x and simplify, we get 1 − cos x 1 1 = x − x3 cos ξ, x 2 24 so that we have 1 − cos x 1 1 3 1 3 − x = x cos ξ ≤ |x | = C|x3 | x 2 24 24 where C = 1/24. Therefore, 1−cos x x = 12 x + O(x3). 3. Use Taylor’s Theorem to show that √ 1 1 + x = 1 + x + O(x2) 2 for x sufficiently small. Solution: We have, from Taylor’s Theorem, with x0 = 0, √ 1 1 1 1 + x = 1 + x − x2 (1 + ξ)−3/2 , 2 8 for some ξ between 0 and x. Since 1 2 x (1 + ξ)−3/2 ≤ C|x2 | 8 ERROR, APPROXIMATE EQUALITY, AND ASYMPTOTIC ORDER NOTATION 13 for all x sufficiently small, the result follows. For example, we have √ 1 2 x (1 + ξ)−3/2 ≤ 1 × 2 2|x2 | 8 8 for all x ∈ [−1/2, 1/2]. 4. Use Taylor’s Theorem to show that (1 + x)−1 = 1 − x + x2 + O(x3) for x sufficiently small. Solution: This time, Taylor’s Theorem gives us that (1 + x)−1 = 1 − x + x2 − x3 /(1 + ξ)4 for some ξ between 0 and x. Thus, for all x such that |x| ≤ m, (1 + x)−1 − (1 − x + x2) = x3 /(1 + ξ)4 ≤ |x|3 /(1 − m)4 = C|x|3 , where C = 1/(1 − m)4 . 5. Show that sin x = x + O(x3). 6. Recall the summation formula 1 + r + r2 + r3 + · · · + rn = n X rk = k=0 1 − rn+1 . 1−r Hint: What is the definition of the O notation? 7. Use the above result to show that 10 terms (k = 9) are all that is needed to compute ∞ X S= e−k k=0 −4 to within 10 absolute accuracy. Solution: The remainder in the 9 term partial sum is −10 e = 0.000071822 < 10−4 . |R9 | = 1 − e−1 14 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW 8. Recall the summation formula n X k= k=1 Use this to show that n X k=1 k= n(n + 1) . 2 1 2 n + O(n). 2 9. State and prove the version of Theorem 1.7 which deals with relationships of the form x = xn + O(β(n)). Solution: The theorem statement might be something like the following: Theorem: Let x = xn + O(β(n)) and y = yn + O(γ(n)), with bβ(n) > γ(n) for all n sufficiently large. Then x+y x+y Ax = xn + yn + O(β(n) + γ(n)), = xn + yn + O(β(n)), = Axn + O(β(n)). In the last equation, A is an arbitrary constant, independent of n. The proof parallels the one in the text almost perfectly, and so is omitted. 10. Use the definition of O to show that if y = yh + O(hp), then hy = hyh + O(hp+1). 11. Show that if an = O(np) and bn = O(nq), then an bn = O(np+q). Solution: We have |an | ≤ Ca |np | and |bn | ≤ Cb |nq |. These follow from the definition of the O notation. Therefore |an bn | ≤ Ca |np ||bn | ≤ (Ca |np |)(Cb |nq |) = (Ca Cb)|np+q | which implies that an bn = O(np+q). 12. Suppose that y = yh + O(β(h)) and z = zh + O(β(h)), for h sufficiently small. Does it follow that y − z = yh − zh (for h sufficiently small)? 13. Show that f 00 (x) = f (x + h) − 2f (x) + f (x − h) + O(h2) h2 A PRIMER ON COMPUTER ARITHMETIC 15 for all h sufficiently small. Hint: Expand f (x ± h) out to the fourth order terms. Solution: This is a straight-forward manipulation with the Taylor expansions 1 1 1 f (x + h) = f (x) + hf 0 (x) + h2 f 00 (x) + h3 f 000 (x) + h4 f 0000 (ξ1) 2 6 24 and 1 1 1 f (x − h) = f (x) − hf 0 (x) + h2 f 00 (x) − h3 f 000 (x) + h4 f 0000 (ξ2). 2 6 24 Add the two expansions to get f (x + h) + f (x − h) = 2f (x) + h2 f 00 (x) + 1 4 0000 h (f (ξ1) + f 0000 (ξ2)). 24 Now solve for f 00 (x). 14. Explain, in your own words, why it is necessary that the constant C in (1.8) be independent of h. /•••. 1.3 A PRIMER ON COMPUTER ARITHMETIC Exercises: 1. In each problem below, A is the exact value, and Ah is an approximation to A. Find the absolute error and the relative error. (a) A = π, Ah = 22/7; (b) A = e, Ah = 2.71828; (c) A = 16 , Ah = 0.1667; (d) A = 61 , Ah = 0.1666. Solution: (a) Abs. error. ≤ 1.265 × 10−3 , rel. error ≤ 4.025 × 10−4 ; (b) Abs. error. ≤ 1.828 × 10−6 , rel. error ≤ 6.72 × 10−7 ; (c) Abs. error. ≤ 3.334 × 10−5 , rel. error ≤ 2.000 × 10−4 ; (d) Abs. error. ≤ 6.667 × 10−5 , rel. error ≤ 4 × 10−4 . 2. Perform the indicated computations in each of three ways: (i) Exactly; (ii) Using three-digit decimal arithmetic, with chopping; (iii) Using three-digit 16 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW decimal arithmetic, with rounding. For both approximations, compute the absolute error and the relative error. (a) (b) (c) (d) 1 6 1 6 1 9 1 10 ; 1 × 10 ; 1 + 7 + 61 ; 1 1 1 7 + 6 + 9. + 3. For each function below explain why a naive construction will be susceptible to significant rounding error (for x near certain values), and explain how to avoid this error. √ (a) f (x) = (x + 9 − 3)x−1 ; (b) f (x) = x−1 (1 − cos x); (c) f (x) = (1 − x)−1 (ln x − sin πx); (d) f (x) = (cos(π + x) − cos π)x−1 ; (e) f (x) = (e1+x − e1−x)(2x)−1 ; Solution: In each case, the function is susceptible to subtractive cancellation which will be amplified by division by a small number. The way to avoid the problem is to use a Taylor expansion to make the subtraction and division both explicit operations. For instance, in (a), we would write f (x) = ((3+(1/6)x−(1/216)x2 +O(x3))−3)x−1 = (1/6)−(1/216)x+O(x2). To get greater accuracy, take more terms in the Taylor expansion. 4. For f (x) = (ex − 1)/x, how many terms in a Taylor expansion are needed to get single precision accuracy (7 decimal digits) for all x ∈ [0, 21]? How many terms are needed for double precision accuracy (14 decimal digits) over this same range? 5. Using single precision arithmetic, only, carry out each of the following computations, using first the form on the left side of the equals sign, then using the form on the right side, and compare the two results. Comment on what you get in light of the material in 1.3. (a) (x +)3 − 1 = x3 + 3x2 + 3x2 + 3 − 1, x = 1.0, = 0.000001. √ √ (b) −b + b2 − 2c = 2c(−b − b2 − 2c)−1 , b = 1, 000, c = π. Solution: “Single precision” means 6 or 7 decimal digits, so the point of the problem is to do the computations using 6 or 7 digits. (a) Using a standard FORTRAN compiler on a low-end UNIX workstation, the author got (x +)3 − 1 = 0.2861022949218750E − 05 A PRIMER ON COMPUTER ARITHMETIC 17 but x3 + 3x2 + 3x2 + 3 − 1 = 0.2980232238769531E − 05. (b) Using a standard C compiler on a low-end UNIX workstation, the author got p −b + b2 − 2c = −0.003 but 2c(−b − p 6. Consider the sum S= b2 − 2c)−1 = −0.00314160. m X −0.05k e−14(1−e) k=0 5 where m = 2 × 10 . Again using only single precision, compute this two ways: First, by summing in the order indicated in the formula; second, by summing backwards, i.e., starting with the k = 200, 000 term and ending with the k = 0 term. Compare your results and comment upon them. 7. Using the computer of your choice, find three values a, b, and c, such that (a + b) + c 6= a + (b + c). Repeat for your pocket calculator. Solution: The key issue is to get an approximation to the machine epsilon, then take a = 1, b = c = (2/3)u or something similar. This will guarantee that (a + b) + c = a but a + (b + c) > a. Using a standard C compiler on a low-end UNIX workstation, the author was able to determine that a = 1, b = 2 × 10−7 , and c = 3 × 10−7 will work. On an elderly Sharp calculator, the author found that a = 1, b = 4 × 10−10 , and c = 4 × 10−10 worked. 8. Assume we are using 3-digit decimal arithmetic. For = 0.0001, a1 = 5, compute 1 a1 a2 = a0 + for a0 equal to each of 1, 2, and 3. Comment. 9. Let ≤ u. Explain, in your own words, why the computation 1 a2 = a0 + a1 is potentially rife with rounding error. (Assume that a0 and a1 are of comparable size.) Hint: See previous problem. Solution: This is just a generalization of the previous problem. If is small enough, then a2 will be independent of a0 . 18 INTRODUCTORY CONCEPTS AND CALCULUS REVIEW 10. Using the computer and language of your choice, write a program to estimate the machine epsilon. Solution: Using MATLAB on a moderately old personal computer, the author got a value of u = 1.1016e − 16. The code is given below. x = 1.e-10; for k=1:6000 y = 1 + x; if y 0 and η > 0 be given, where η = θh, for 0 < θ < 1. Let f be some smooth function. Use Taylor expansions for f (x + h) and f (x − η) in terms of f and its derivatives at x in order to construct an approximation to f 0 (x) that depends on f (x + h), f (x), and f (x − η), and which is O(h2) accurate. Check your work by verifying that for θ = 1 ⇒ η = h you get the same results as in the text. Solution: We write the two Taylor expansions 1 1 f (x + h) = f (x) + hf 0 (x) + h2 f 00 (x) + h3 f 000 (c1) 2 6 DIFFERENCE APPROXIMATIONS TO THE DERIVATIVE 35 and 1 1 f (x − η) = f (x) − ηf 0 (x) + η 2 f 00 (x) − η 3 f 000 (c2). 2 6 If we simply subtract these, the second derivative terms will not cancel out (because h 6= η), so we have to multiply the first expansion by η 2 and the second one by h2 to get 1 1 η 2 f (x + h) = η 2 f (x) + η 2 hf 0 (x) + η 2 h2 f 00 (x) + η 2 h3 f 000 (c1) 2 6 and 1 1 h2 f (x − η) = h2 f (x) − h2 ηf 0 (x) + h2 η 2 f 00 (x) − h2 η 3 f 000 (c2). 2 6 Now subtract to get η 2 f (x + h) − h2 f (x − η) = + (η 2 − h2)f (x) + (η 2 h − h2 η)f 0 (x) 1 + (η 2 h3 f 000 (c1) + h2 η 3 f 000 (c2)). 6 Solve for f 0 (x) to get f 0 (x) η 2 f (x + h) − (η 2 − h2)f (x) − h2 f (x − η) η 2 h − h2 η 1 η 2 h3 f 000 (c1) + h2 η 3 f 000 (c2) . − 6 η 2 h − h2 η = Because we assumed η = θh there is substantial simplification that we can do to get f 0 (x) = θ2 f (x + h) − (θ2 − 1)f (x) − f (x − η) θ2 h − hθ 2 2 000 1 θ h f (c1) + h2 θ3 f 000 (c2) , − 6 θ2 − θ which is sufficient to establish the O(h2) estimate for the error. 14. Write a computer program to test the approximation to the second derivative from Problem 12 by applying it to estimate f 00 (1) for each of the following functions, using h−1 = 4, 8, 16, 32. Verify that the predicted theoretical accuracy is obtained. (a) f (x) = e−x ; (b) f (x) = cos πx; √ (c) f (x) = 1 + x; (d) f (x) = ln x; 36 A SURVEY OF SIMPLE METHODS AND TOOLS 15. Define the following function: f (x) = ln(e √ x2 +1 sin(πx) + tan πx). 0 Compute values of f over the range [0, 14] using two methods: (a) Using the centered-difference formula from (2.5); (b) Using ordinary calculus to find the formula for f 0 . Comment. Solution: The function f is designed to make direct computation of f 0 a difficult task for the average undergraduate student, thus highlighting that there are times when being able to approximate the derivative as was done in this section might be preferable to doing the exact computation. 16. Let f (x) = ex , and consider the problem of approximating f 0 (1), as in the text. Let D1 (h) be the difference approximation in (2.1). Using the appropriate values in Table 2.1, compute the new approximations ∆1 (h) = 2D1 (h) − D1 (2h); and compare these values to the exact derivative value. Are they more or less accurate that the corresponding values of D1 ? Try to deduce, from your calculations, how the error depends on h. Solution: For the sake of completeness we have reproduced the table from the text. Using those values we can compute ∆1 (.25) = 2D1 (.25) − D1 (.5) = 2.649674415 e − ∆1 (.25) = 0.068607413; 1 ∆1 (.125) = 2D1 (.125) − D1 (.25) = 2.702717782 e − ∆1 (.125) = 0.015564046; 1 ∆1 (.0625) = 2D1 (.0625) − D1 (.125) = 2.714572906 e − ∆1 (.0625) = 0.003708922; 1 ∆1 (.03125) = 2D1 (.03125) − D1 (.0625) = 2.717372894 e − ∆1 (.03125) = 0.00090834. 1 The error appears to be going down by a factor of 4 as h is cut in half. 17. Repeat the above idea for f (x) = arctan(x), x = 1 (but this time you will have to compute the original D1 (h) values). 18. By keeping more terms in the Taylor expansion for f (x + h), show that the error in the derivative approximation (2.1) can be written as f (x + h) − f (x) 1 1 f 0 (x) − (2.1) = − hf 00 (x) − h2 f 000 (x) − ... h 2 6 DIFFERENCE APPROXIMATIONS TO THE DERIVATIVE Table 2.3 x = 1. h−1 2 4 8 16 32 64 128 37 (Table 2.1 in text.) Example of derivative approximation to f (x) = ex at D1 (h) E1 (h) = f 0 (1) − D1 (h) 3.526814461 -0.8085327148E+00 3.088244438 -0.3699626923E+00 2.895481110 -0.1771993637E+00 2.805027008 -0.8674526215E-01 2.761199951 -0.4291820526E-01 2.739639282 -0.2135753632E-01 2.728942871 -0.1066112518E-01 D2 (h) E2 (h) = f 0 (1) − D2 (h) 2.832967758 -0.1146860123E+00 2.746685505 -0.2840375900E-01 2.725366592 -0.7084846497E-02 2.720052719 -0.1770973206E-02 2.718723297 -0.4415512085E-03 2.718391418 -0.1096725464E-03 2.718307495 -0.2574920654E-04 Use this to construct a derivative approximation involving f (x), f (x + h), and f (x + 2h) that is O(h2) accurate. Hint: Use (2.6) to write down the error in the approximation f (x + 2h) − f (x) f 0 (x) ≈ 2h and combine the two error expansions so that the terms that are O(h) are eliminated. Solution: We have f (x + h) = f (x) + hf 0 (x) + (1/2)h2 f 00 (x) + O(h3) so that we can get f 0 (x) = f (x + h) − f (x) − (1/2)hf 00 (x) + O(h2), h or, equivalently, D1 (h) = f (x + h) − f (x) = f 0 (x) + (1/2)hf 00 (x) + O(h2). h Therefore D1 (2h) = f (x + 2h) − f (x) = f 0 (x) + (1/2)(2h)f 00 (x) + O(h2). 2h Thus, 2D1 (2h) − D1 (h) = − 2 f 0 (x) + (1/2)(2h)f 00 (x) + O(h2) f 0 (x) + (1/2)hf 00 (x) + O(h2) so that 2D1 (2h) − D1 (h) = f 0 (x) + O(h2). 38 A SURVEY OF SIMPLE METHODS AND TOOLS 19. Apply the method derived above to the list of functions in Problem 3, and confirm that the method is as accurate in practice as is claimed. 20. Let f (x) = ex , and consider the problem of approximating f 0 (1), as in the text. Let D2 (h) be the difference approximation in (2.5). Using the appropriate values in Table 2.3, compute the new approximations ∆2 (h) = (4D2 (h) − D2 (2h))/3; and compare these values to the exact derivative value. Are they more or less accurate that the corresponding values of D2 ? Try to deduce, from your calculations, how the error depends on h. Solution: The D2 (h) values are D2 (1/2) = 2.832967758 D2 (1/4) = 2.746685505 D2 (1/8) = 2.725366592 D2 (1/16) = 2.720052719 D2 (1/32) = 2.718723297 D2 (1/64) = 2.718391418 D2 (1/128) = 2.718307495 from which we get that ∆2 (1/4) = 2.717924754 ∆2 (1/8) = 2.718260288 ∆2 (1/16) = 2.718281428 ∆2 (1/32) = 2.718280156 ∆2 (1/64) = 2.718280792 ∆2 (1/128) = 2.718279521. These values are, initially, much more accurate than D2 (h), although rounding error begins to corrupt the computation. It can be shown that this approximation is O(h4). 21. Repeat the above idea for f (x) = arctan(x), x = 1 (but this time you will have to compute the original D2 (h) values). 22. The same ideas as in Problem 18 can be applied to the centered difference approximation (2.5). Show that in this case the error satisfies f 0 (x) − f (x + h) − f (x − h) 1 1 4 00000 = − h2 f 000 (x) − h f (x) − ... (2.2) 2h 6 120 DIFFERENCE APPROXIMATIONS TO THE DERIVATIVE 39 Use this to construct a derivative approximation involving f (x ± h), and f (x ± 2h) that is O(h4) accurate. Solution: Since f (x + h) = f (x) + hf 0 (x) + (1/2)h2 f 00 (x) + (1/6)h3 f 000 (x) +(1/24)h4 f 0000 (x) + O(h5) f (x − h) = f (x) − hf 0 (x) + (1/2)h2 f 00 (x) − (1/6)h3 f 000 (x) +(1/24)h4 f 0000 (x) + O(h5), and we have that f (x + h) − f (x − h) = 2hf 0 (x) + (1/3)h3 f 000 (x) + O(h5) or, f 0 (x) − f (x + h) − f (x − h) = (1/6)h2 f 000 (x) + O(h4). 2h Therefore, f 0 (x) − f (x + 2h) − f (x − 2h) = (4/6)h2 f 000 (x) + O(h4), 4h so that f (x + h) − f (x − h) 0 4 f (x) − 2h f (x + 2h) − f (x − 2h) − f 0 (x) − = O(h4). 4h We can manipulate with the left side to get that 3f 0 (x) − 8(f (x + h) − f (x − h)) − (f (x + 2h) − f (x − 2h)) = O(h4), 4h so that finally we have f 0 (x) − 8(f (x + h) − f (x − h)) − (f (x + 2h) − f (x − 2h)) = O(h4). 12h This is simply an alternate derivation of the method from Exercise 8. 23. Apply the method derived above to the list of functions in Problem 3, and confirm that the method is as accurate in practice as is claimed. /•••. 40 A SURVEY OF SIMPLE METHODS AND TOOLS 2.3 APPLICATION: EULER’S METHOD FOR INITIAL VALUE PROBLEMS Exercises: 1. Use Euler’s method with h = 0.25 to compute approximate solution values for the initial value problem y 0 = sin(t + y), y(0) = 1. You should get y4 = 1.851566895 (be sure your calculator is set in radians). Solution: y1 = y0 + (0.25) sin(0 + y0) = 1 + 0.25(.8414709848) = 1.210367746 y2 = y1 + (0.25) sin(h + y1) = 1.458844986 y3 = y2 + (0.25) sin(2h + y2) = 1.690257280 y4 = y3 + (0.25) sin(3h + y3) = 1.851566896 2. Repeat the above with h = 0.20. What value do you now get for y5 ≈ y(1)? 3. Repeat the above with h = 0.125. What value do you now get for y8 ≈ y(1)? Solution: y1 = y0 + (0.125) sin(0 + y0) = 1 + 0.125(.8414709848) = 1.105183873 y2 = y1 + (0.125) sin(h + y1) = 1.223002653 y3 = y2 + (0.125) sin(2h + y2) = 1.347405404 y4 = y3 + (0.125) sin(3h + y3) = 1.470971572 y5 = y4 + (0.125) sin(4h + y4) = 1.586095664 y6 = y5 + (0.125) sin(5h + y5) = 1.686335284 y7 = y6 + (0.125) sin(6h + y6) = 1.767364012 y8 = y7 + (0.125) sin(7h + y7) = 1.827207570 4. Use Euler’s method with h = 0.25 to compute approximate solution values for y 0 = et−y , y(0) = −1. What approximate value do you get for y(1) = 0.7353256638? 5. Repeat the above with h = 0.20. What value do you now get for y5 ≈ y(1)? 41 APPLICATION: EULER’S METHOD FOR INITIAL VALUE PROBLEMS Solution: We have the computation yn+1 = yn + hetn −yn where h = 0.2, t0 = 0, and y0 = −1. Hence y1 y2 = y0 + het0 −y0 = −1 + (0.2)e0−(−1) = −0.4563436344, = y1 + het1 −y1 = −0.4563436344 + (0.2)e0.2−(−0.4563436344) y3 = −0.0707974456, = y2 + het2 −y2 = −0.0707974456 + (0.2)e0.4−(−0.0707974456) y4 = 0.2494566764, = y3 + het3 −y3 = 0.2494566764 + (0.2)e0.6−(0.2494566764) y5 = 0.5334244306, = y4 + het4 −y4 = 0.5334244306 + (0.2)e0.8−(0.5334244306) = 0.7945216786. 6. Repeat the above with h = 0.125. What value do you now get for y8 ≈ y(1)? 7. Use Euler’s method with h = 0.0625 to compute approximate solution values over the interval 0 ≤ t ≤ 1 for the initial value problem y 0 = t − y, y(0) = 2, which has exact solution y(t) = 3e−t + t − 1. Plot your approximate solution as a function of t, and plot the error as a function of t. Solution: The plots are given in Figures 2.1 and 2.2. 2 0.04 1.9 0.035 1.8 0.03 1.7 0.025 1.6 1.5 0.02 1.4 0.015 1.3 0.01 1.2 0.005 1.1 1 0 0.1 0.2 0.3 Figure 2.1 Exercise 2.3.7. 0.4 0.5 Exact 0.6 0.7 0.8 solution 0.9 1 for 0 0 0.1 0.2 0.3 Figure 2.2 Exercise 2.3.7. 8. Repeat the above for the equation y 0 = et−y , y(0) = −1, 0.4 0.5 0.6 Error 0.7 plot 0.8 0.9 for 1 42 A SURVEY OF SIMPLE METHODS AND TOOLS which has exact solution y = ln(et − 1 + e−1). 9. Repeat the above for the equation y 0 + y = sin t, y(0) = −1, which has exact solution y = (sin t − cos t − e−t)/2. Solution: The plots are given in Figures 2.3 and 2.4. −3 0 0 −0.1 x 10 −0.5 −0.2 −1 −0.3 −0.4 −1.5 −0.5 −2 −0.6 −0.7 −2.5 −0.8 −3 −0.9 −1 0 0.1 0.2 0.3 Figure 2.3 Exercise 2.3.9. 0.4 0.5 Exact 0.6 0.7 0.8 solution 0.9 for 1 −3.5 0 0.1 0.2 0.3 Figure 2.4 2.3.9. 0.4 0.5 0.6 0.7 0.8 0.9 1 Error plot for Exercise 10. Use Euler’s method to compute approximate solutions to each of the initial value problems below, using h−1 = 2, 4, 8, 16. Compute the maximum error over the interval [0, 1] for each value of h. Plot your approximate solutions for the h = 1/16 case. Hint: Verify that your code works by using it to reproduce the results given for the examples in the text. (a) y 0 = t − y, y(0) = 2; y(t) = 3e−t + t − 1. (b) y 0 + 4y = 1, y(0) = 1; y(t) = 41 (3e−4t + 1). −t (c) y 0 = −y ln y, y(0) = 3; y(t) = e(ln 3)e . 11. Consider the approximate values in Tables 2.5 and 2.6 in the text. Let yk8 denote the approximate values for h = 1/8, and yk16 denote the approximate values for h = 1/16. Note that yk8 ≈ y(k/8) and 16 y2k ≈ y(2k/16) = y(k/8) ≈ yk8 , 16 thus yk8 and y2k are both approximations to the same value. Compute the set of new approximations 16 uk = 2y2k − yk8 APPLICATION: EULER’S METHOD FOR INITIAL VALUE PROBLEMS 43 and compare these to the corresponding exact solution values. Are they better or worse as an approximation? Solution: Table 2.4 shows the new solution and the error, which is much smaller than that obtained using Euler’s method directly. Table 2.4 Solution values for Exercise 2.3.11. tk 0.000 0.125 0.250 0.375 0.500 0.625 0.750 0.875 1.000 uk -1.00000000000000 -0.87500508526970 -0.75013901623461 -0.62573474516817 -0.50230696317037 -0.38052202438375 -0.26116806849085 -0.14512551018440 -0.03333809297284 y(tk) − uk 0 -0.00000483294465 -0.00001560652915 -0.00002744064045 -0.00003687832904 -0.00004161315812 -0.00004026230490 -0.00003217761850 -0.00001728814300 12. Apply the basic idea from the previous problem to the approximation of solutions to y 0 = et−y , y(0) = −1, which has exact solution y = ln(et − 1 + e−1). 13. Assume that the function f satisfies |f (t, y) − f (t, z)| ≤ K|y − z| for some constant K. Use this and (2.9)-(2.10) to show that the error |y(tn+1)− yn+1 | satisfies the recursion 1 |y(tn+1) − yn+1 | ≤ (1 + Kh)|y(tn) − yn | + h2 Y2 2 where Y2 = max |y 00 (t)|. t Solution: If we take t = tn and subtract (2.10) from (2.9) we get y(tn + h) − yn+1 = y(tn) − yn + h(f (tn , y(tn)) − f (tn , yn)), from which we have |y(tn+1) − yn+1 | ≤ |y(tn) − yn | + hK|y(tn) − yn)|, and the desired result follows immediately. /•••. 44 A SURVEY OF SIMPLE METHODS AND TOOLS 2.4 LINEAR INTERPOLATION Exercises: 1. Use linear interpolation to find approximations to the following values of the error function, using the table in the text. For each case, give an upper bound on the error in the approximation. (a) erf(0.56); (b) erf(0.07); (c) erf(0.34); (d) erf(0.12); (e) erf(0.89). Solution: For Part (a), we have x0 = 0.5 and x1 = 0.6, so x − 0.5 0.6 − x p1 (x) = (0.60385609084793)+ (0.52049987781305), 0.6 − 0.5 0.6 − 0.5 thus p1 (0.56) = = 0.06 0.04 (0.60385609084793) + (0.52049987781305) 0.1 0.1 0.5705136056. The upper bound on the error comes straight from the error theorem on page 59. We have 1 |erf(x) − p1 (x)| ≤ (0.1)2 max |f 00 (t)|, 8 where the maximum of the second derivative is computed over the interval [0.5, 0.6]. We have 2 −4x (erf(x))00 = √ e−x π so that |(erf(x))00 | ≤ 4 × 0.6 −0.25 √ e = 1.05... π Hence the error satisfies |erf(0.56) − p1 (0.56)| ≤ 0.01 × 1.05... = 0.00132... 8 2. The gamma function, denoted by Γ(x), occurs in a number of applications, most notably probability theory and the solution of certain differential equations. It is basically the generalization of the factorial function to non-integer values, in that Γ(n + 1) = n!. Table 2.5 (Table 2.8 in the text) gives values of Γ(x) LINEAR INTERPOLATION 45 for x between 1 and 2. Use linear interpolation to approximate values of Γ(x) as given below. (a) Γ(1.290) = 0.8990415863; (b) Γ(1.005) = 0.9971385354; (c) Γ(1.930) = 0.9723969178; (d) Γ(1.635) = 0.8979334930. Table 2.5 (Table 2.8 in the text) Table of Γ(x) values. x 1.00 1.10 1.20 1.30 1.40 1.50 1.60 1.70 1.80 1.90 2.00 Γ(x) 1.0000000000 0.9513507699 0.9181687424 0.8974706963 0.8872638175 0.8862269255 0.8935153493 0.9086387329 0.9313837710 0.9617658319 1.0000000000 3. Theorem 2.1 requires an upper bound on the second derivative of the function being interpolated, and this is not always available as a practical matter. However, if a table of values is available, we can use a difference approximation to the second derivative to estimate the upper bound on the derivative and hence the error. Recall, from Problem 12 of 2.2, f 00 (x) ≈ f (x − h) − 2f (x) + f (x + h) . h2 Assume the function values are given at the equally spaced grid points xk = a + kh for some grid spacing, h. Using the estimate max |f 00 (x)| ≈ f (xk−1) − 2f (xk) f (xk) − 2f (xk+1) + f (xk+2) max , h2 h2 xk ≤x≤xk+1 to approximate the derivative upper bound, estimate the error made in using linear interpolation to approximate each of the following values of Γ(x), based on the same table as in the previous problem. (a) Γ(1.290) = 0.8990415863; 46 A SURVEY OF SIMPLE METHODS AND TOOLS (b) (c) (d) (e) Γ(1.579) = 0.8913230181; Γ(1.456) = 0.8856168100; Γ(1.314) = 0.8954464400; Γ(1.713) = 0.9111663772. Solution: We do Part (d), only. This is a “two-stage” problem. First, we compute an approximate value for Γ(1.314) using linear interpolation. Then we estimate the error using the suggested device for approximating the second derivative. The interpolation is straight-forward: x − 1.3 1.4 − x p1 (x) = (0.8872638175) + (0.8974706963), 1.4 − 1.3 1.4 − 1.3 so that Γ(1.314) ≈ p1 (1.314) = = 0.014 (0.8872638175) .1 0.086 + (0.8974706963) .1 0.8960417333. Now, the error is bounded according to |Γ(x) − p1 (x)| ≤ 1 (x1 − x0)2 max |(Γ(t))00 | 8 where the maximum is taken over the interval [x0 , x1]. We don’t have a formula for Γ(x), so we can’t get one for the second derivative. But we do have the approximation (for any function) f 00 (x) ≈ f (x + h) − 2f (x) + f (x − h) h2 so we use this to estimate the second derivative of Γ(x): Γ00 (1.3) ≈ 0.8872638175 − 2(0.8974706963) + 0.9181687424 = 1.049... 0.01 Γ00 (1.4) ≈ 0.8862269255 − 2(0.8872638175) + 0.8974706963 = 0.917... 0.01 max |Γ00 (t)| ≈ max{1.049..., 0.917...} = 1.049... Therefore the estimate on the error is 1 |Γ(1.314) − p1 (1.314)| ≤ (0.1)2 (1.049...) = 0.00131... 8 According to MATLAB, the exact value (to 15 places) is Γ(1.314) = 0.89544644002887, LINEAR INTERPOLATION 47 so our interpolation error is −5.95 × 10−4 , so our estimate is a tad high, but certainly it is within the range of acceptable estimation. 4. Construct a linear interpolating polynomial to the function f (x) = x−1 using x0 = 21 and x1 = 1 as the nodes. What is the upper bound on the error over the interval [21 , 1], according to the error estimate? √ 5. Repeat the above for f (x) = x, using the interval [14 , 1]. Solution: The polynomial is p1 (x) = 1−x x − 1/4 (1) + (1/2) = (2x + 1)/3. 1 − 1/4 1 − 1/4 The error bound is given by |f (x) − p1 (x)| ≤ 1 (3/4)2 max |(−1/4)t−3/2 | 8 t∈[1/4,1] = (9/128) × (1/4) × (1/4)−3/2 = 9/64 = 0.140625. This √ is again a conservative estimate, since a simple plot of the difference x − p1 (x) shows that the maximum absolute error is about 0.04. 6. Repeat the above for f (x) = x1/3 , using the interval [81 , 1]. 7. If we want to use linear interpolation to the sine function and obtain an accuracy of 10−6 , how close together do the entries in the table have to be? What if we change the error criterion to 10−3 ? Solution: This amounts to asking how small does x1 − x0 have to be to make the upper bound in the error estimate less than the specified tolerance. For convenience set h = x1 − x0 . We have, then, for f (x) = sin x, |f (x) − p1 (x)| ≤ 1 2 h (1) 8 √ so that making the error less than 10−6 requires taking h ≤ 8 × 10−3 = .2828427124e−2. For an error less than 10−3 we need h ≤ .8944271912e−1. 8. Repeat the above for f (x) = tan x, for x ∈ [−π/4, π/4]. 9. If we try to approximate the logarithm by using a table of logarithm entries, together with linear interpolation, to construct the approximation to ln(1 + x) over the interval [− 12 , 0], how many points are required to get an approximation that is accurate to within 10−14 ? Solution: The error estimate is |f (x) − p1 (x)| ≤ 1 2 1 h max | − 1/(1 + t)2 | = h2 (4) = h2 /2. 8 t∈[−1/2,1] 8 48 A SURVEY OF SIMPLE METHODS AND TOOLS So we require h ≤ .1414213562e − 6 for 10−14 accuracy; this means we need n = (1/2)/h ≈ 3.5 × 106 points. √ 10. Construct a piecewise linear approximation to f (x) = x over the interval 9 [41 , 1] using the nodes 14 , 16 , 1. What is the maximum error in this approximation? 11. Repeat the above for f (x) = x1/3 over the interval [18 , 1], using the nodes 1 27 8 , 64 , 1. Solution: The polynomials are Q1 (x) = Q2 (x) = x − 1/8 27/64 − x (3/4) + (1/2) = (32x + 15)/38, 27/64 − 1/8 27/64 − 1/8 (16x + 21)/37. The maximum errors are given by |f (x) − Q1 (x)| ≤ = 1 (19/64)2 max |(−2/9)t−5/3 | 8 t∈[1/8,27/64] 2 361 361 × × 32 = = .7834201389e − 1 32768 9 4608 and |f (x) − Q2 (x)| ≤ = 1 (37/64)2 max |(−2/9)t−5/3 | 8 t∈[27/64,1] 1369 2 1024 1369 × × = = .3912322817e − 1, 32768 9 243 34992 so the overall error bound is about 0.078. /•••. 2.5 APPLICATION — THE TRAPEZOID RULE Exercises: 1. Use the trapezoid rule with h = 1 4 to approximate the integral Z I= 0 1 x3 dx = 1 . 4 You should get that T4 = 17/64. How small does h have to be to get that the error is less than 10−3 ? 10−6 ? APPLICATION — THE TRAPEZOID RULE 49 2. Use the trapezoid rule and h = π/4 to approximate the integral Z π/2 sin xdx = 1. I= 0 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? 3. Repeat the above with h = π/5. 4. Apply the trapezoid rule with h = 81 , to approximate the integral Z 1 I= 0 1 √ dx = 0.92703733865069. 1 + x4 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: It’s a fairly direct (if tedious) computation to get T8 1 2 2 √ +p +p 4 4 1+0 1 + (1/8) 1 + (2/8)4 2 2 2 2 +p +p +p +p 1 + (3/8)4 1 + (4/8)4 1 + (5/8)4 1 + (6/8)4 ! 2 1 +p +√ 1 + 14 1 + (7/8)4 1 (1 + 1.99975590406939 + 1.99610515696578 = 16 +1.98051315767607 + 1.94028500029066 + 1.86291469895924 +1.74315107424910 + 1.58801110913806 + 0.70710678118655) = 0.92611518015843 = 1/8 2 Now, for the accuracy, we have |I(f) − Tn (f)| = b − a 2 00 h |f (ξh)|. 12 The second derivative is f 00 (x) = 6x2 (x4 − 1) , (1 + x4)5/2 and we can bound this (for all x ∈ [0, 1]) as follows |f 00 (x)| ≤ 6(x4 − 1) ≤ 6. A better upper bound would come from |f 00 (x)| ≤ 6x2 (x4 − 1) ≤ 2.31... 50 A SURVEY OF SIMPLE METHODS AND TOOLS Therefore, to get 10−3 accuracy requires (using the weaker upper bound) p h ≤ (12/6) × 10−3 = 0.0447... To get 10−6 accuracy requires p h ≤ (12/6) × 10−6 = 0.00141... 5. Apply the trapezoid rule with h = 18 , to approximate the integral Z 1 1 x(1 − x2)dx = . I= 4 0 Feel free to use a computer program or a calculator, as you wish. How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: The approximation is T8 (f) = .2460937500. The error bound gives us |I(f) − Tn (f)| ≤ (1/12)h2 max | − 6t| = h2 /2, t∈[0,1] so we require h ≤ .4472135955e−1 for 10 −2 for 10−6 accuracy. −3 accuracy, and h ≤ .1414213562e 6. Apply the trapezoid rule with h = 81 , to approximate the integral Z 1 ln(1 + x)dx = 2 ln 2 − 1. I= 0 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: The approximation is T8 (f) = .3856439099. The error bound yields |I(f) − Tn (f)| ≤ (1/12)h2 max | − 1/(1 + t)2 | = h2 /12, t∈[0,1] so we need h ≤ .1095445115 to get 10−3 accuracy, and h ≤ .3464101615e −2 to get 10−6 accuracy. 7. Apply the trapezoid rule with h = 81 , to approximate the integral Z 1 1√ 1 1 dx = ln 2 + 3π. I= 3 3 9 0 1+x How small does h have to be to get that the error is less than 10−3 ? 10−6 ? 8. Apply the trapezoid rule with h = 81 , to approximate the integral Z 2 2 e−x dx = 0.1352572580. I= 1 APPLICATION — THE TRAPEZOID RULE 51 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: We have 1/8 −1 I8 = e + 2e−(81/64) + 2e−(100/64) + 2e−(121/64) + 2e−(144/64) 2 + = + + = 2e−(169/64) + 2e−(196/64) + e−(225/64) + e−(256/64) 1 (0.56412590338763 + 0.41922277430220 + 0.30195843691183 16 0 0.21079844912373 + 0.14263336539552 + 0.09354124476792 0.05945843277232) 0.13612063042008. To get the accuracy, we note that 2 f 00 (x) = (4x2 − 2)e−x , so that, over the interval [1, 2], |f 00 (x)| ≤ |4x2 − 2|e−1 ≤ (16 + 2)e−1 = 18e−1 = 6.62... Therefore, to get an error less than 10−3 we take h to satisfy p h ≤ (12/6.62) × 10−3 = 0.0426... To get an error less than 10−6 we take h to satisfy p h ≤ (12/6.62) × 10−6 = 0.00135... 9. Let I8 denote the value you obtained in the previous problem. Repeat the computation, this time using h = 41 , and call this approximate value I4 . Then compute IR = (4I8 − I4)/3 and compare this to the given exact value of the integral. 10. Repeat the above for the integral Z 1 1 1 1√ I= dx = ln 2 + 3π. 3 3 9 0 1+x Solution: We get I4 = .8317002443, I8 = .8346696206, and IR = .8356594123. 52 A SURVEY OF SIMPLE METHODS AND TOOLS The exact value is I(f) = .8356488485, so we see that IR is much more accurate than either of the trapezoid rule values. 11. For each integral below, write a program to do the trapezoid rule using the 1 sequence of mesh sizes h = 21 (b − a), 41 (b − a), 81 (b − a), . . . , 128 (b − a), where b − a is the length of the given interval. Verify that the expected rate of decrease of the error is obtained. (a) f (x) = x2 e−x , [0, 2],I(f) = 2 − 10e−2 = 0.646647168; (b) f (x) = 1/(1 + 25x2), [0, 1], I(f) = 51 arctan(5); √ (c) f (x) = 1 − x2 , [−1, 1], I(f) = π/2; (d) f (x) = ln x, [1, 3], I(f) = 3 ln 3 − 2 = 1.295836867; (e) f (x) = x5/2 , [0, 1], I(f) = 2/7; (f) f (x) = e−x sin(4x), [0, π], I(f) = 4 17 (1 − e−π) = 0.2251261368. 12. For each integral in Problem 11, how small does h have to be to get accuracy, according to the error theory, of at least 10−3 ? 10−6 ? Solution: For the single case of (d), since f (x) = ln x, we have f 00 (x) = −1/x2 , so |f 00 (x)| ≤ 1 for all x on [1, 3]. We therefore have |I(f) − Tn (f)| ≤ b − a 2 00 1 h |f (ξh)| ≤ h2 . 12 6 We therefore get 10−3 accuracy by imposing 16 h2 ≤ 10−3 which implies h ≤ 0.077... For 10−6 accuracy we have h ≤ 0.0024... 13. Apply the trapezoid rule to the integral Z 1 √ 2 I= xdx = 3 0 using a sequence of uniform grids with h = 21 , 14 , ... Do we get the expected rate of convergence? Explain. 14. The length of a curve y = g(x), for x between a and b, is given by the integral Z L(g) = b q 2 1 + [g 0 (x)] dx. a Use the trapezoid rule with h = π/4 and h = π/16 to find the length of one “arch” of the sine curve. Solution: We get L(sin) ≈ 3.819943644 APPLICATION — THE TRAPEZOID RULE 53 (using h = π/4), and L(sin) ≈ 3.820197788 (using h = π/16). 15. Use the trapezoid rule to find the length of the logarithm curve between a = 1 and b = e, using n = 4 and n = 16. 16. What should h be to guarantee an accuracy of 10−8 when using the trapezoid rule for each of the following integrals. (a) 1 Z 2 e−x dx; I(f) = 0 Solution: The error bound gives us 2 |I(f) − Tn (f)| ≤ (1/12)h2 max |(4t2 − 2)e−t | ≤ (h2 /12)(2)(1) = t∈[0,1] h2 /6, so we require h ≤ .2449489743e − 3. (b) Z 3 ln xdx; I(f) = 1 Solution: The error bound gives us |I(f) − Tn (f)| ≤ (2/12)h2 max | − 1/t2 | ≤ (h2 /6)(1) = h2 /6 t∈[1,3] so we require h ≤ .2449489743e − 3. (c) Z 5 I(f) = −5 1 dx; 1 + x2 Solution: The error bound gives us |I(f) − Tn (f)| ≤ (10/12)h2 max |(6x2 − 2)/(x2 + 1)3 | t∈[−5,5] 2 2 ≤ (5h /6)(2) = 5h /3 so we require h ≤ .7745966692e − 4. (d) Z 1 cos(πx/2)dx. I(f) = 0 Solution: The error bound gives us |I(f) − Tn (f)| ≤ (1/12)h2 max | − (π/2)2 cos(πx/2)| ≤ π 2 h2 /48 t∈[0,1] 54 A SURVEY OF SIMPLE METHODS AND TOOLS so we require h ≤ .2205315581e − 3. 17. Since the natural logarithm is defined as an integral, Z x 1 ln x = dt 1 t (2.3) it is possible to use the trapezoid rule (or any other numerical integration rule) to construct approximations to ln x. (a) Show that using the trapezoid rule on the integral (2.3) results in the series approximation (for x ∈ [1, 2]) n−1 x2 − 1 X x−1 ln x ≈ + . 2nx n + k(x − 1) k=1 Hint: What are a and b in the integral defining the logarithm? (b) How many terms are needed in this approximation to get an error of less than 10−8 for all x ∈ [1, 2]? How many terms are needed for an error of less than 10−15 over the same interval? (c) Implement this series for a predicted accuracy of 10−8 and compare it to the intrinsic natural logarithm function on your computer, over the interval [1, 2]. Is the expected accuracy achieved? (d) If we were only interested in the interval [1, 3/2], how many terms would be needed for the accuracy specified in (b)? (e) Is it possible to reduce the computation of ln x for all x > 0 to the computation of ln z for z ∈ [1, 3/2]? Explain. 18. How small must h be to compute the error function, Z x 2 2 erf(x) = √ e−t dt, π 0 using the trapezoid rule, to within 10−8 accuracy for all x ∈ [0, 1]? Solution: The error estimate implies that √ 2 xh2 |erf(x) − Tn | ≤ (x/12)h2 max |(2/ π(4t2 − 2)e−t | ≤ √ π t∈[0,1] where h = x/n and we have used a very crude upper bound on the second derivative. Since we want to achieve the specified accuracy for all x ∈ [0, 1], we use x = 1 (since this maximizes the upper bound) and therefore solve the inequality h2 √ ≤ 10−8 π APPLICATION — THE TRAPEZOID RULE to get h ≤ .1331335364e − 3. 19. Use the data in Table 2.8 to compute Z I= 2 Γ(x)dx 1 using h = 0.2 and h = 0.1. 20. Use the data in Table 2.7 to compute Z 1 erf(x)dx I= 0 using h = 0.2 and h = 0.1. Solution: Using h = 0.2 we get Z 1 erf(x)dx ≈ .4836804793; 0 using h = 0.1 we get Z 1 erf(x)dx ≈ .4854701356. 0 21. Show that the trapezoid rule is exact for all linear polynomials. 22. Prove Theorem 2.4. Solution: Let hi = xi − xi−1 ; then we can apply Theorem 2.2 to get that I(f) − Tn (f) = n X −h3 i i=1 12 f 00 (ξi), where ξi ∈ [xi−1 , xi]. Therefore, |I(f) − Tn (f)| ≤ = n X h3 i i=1 n X i=1 2 ≤ = ! h 12 12 max |f 00 (x)| x∈[a,b] h2i (xi − xi−1) 12 n X i=1 ! max |f 00 (x)| x∈[a,b] ! (xi − xi−1) max |f 00 (x)| x∈[a,b] h2 (b − a) max |f 00 (x)|. 12 x∈[a,b] 55 56 A SURVEY OF SIMPLE METHODS AND TOOLS 23. Extend the discussion on stability to include changes in the interval of integration instead of changes in the function. State and prove a theorem that bounds the change in the trapezoid rule approximation to Z b f (x)dx I(f) = a when the upper limit of integration changes from b to b + , but f remains the same. 24. Consider a function f˜(x) which is the floating point representation of f (x); thus f − f˜ is the rounding error in computing f . If we assume that |f (x)− f˜(x)| ≤ for all x, show that |Tn (f) − Tn (f˜)| ≤ (b − a). What does this say about the effects of rounding error on the trapezoid rule? Solution: Tn (f) − Tn (f˜) = (h/2) n X f (xi) − f˜(xi) + f (xi−1) − f˜(xi−1) , i=1 so |Tn (f) − Tn (f˜)|leq(h/2) n X (e(xi) + e(xi−1))) , i=1 where e(x) = f (x) − f˜(x). The result follows by simple manipulation. /•••. 2.6 SOLUTION OF TRIDIAGONAL LINEAR SYSTEMS Exercises: 1. Use the tridiagonal algorithm in following system of equations: ⎡ 4 2 0 0 ⎤ ⎢ 1 4 1 0 ⎥ ⎣ 0 1 4 1 0 0 2 4 this section to compute the solution to the ⎤⎡ x1 √π/9 ⎥⎢ x2 ⎥⎢ 3/2 ⎦⎣ √ ⎦⎣ x3 ⎦ = ⎣ 3/2 x4 −π/9 ⎤ ⎥ ⎥. ⎦ Solution: After the elimination step is completed, the triangular system is ⎤⎡ ⎤⎡ ⎤ x1 0.3491 4 2 0 0 || x2 || 0.7788 || 0 3.5 1 0 | ⎢ | ⎢ || x3 ⎦ = ⎣ 0.6435 ⎦ . ⎣ 0 0 3.7143 1 0 0 0 3.4615 x4 −0.6956 SOLUTION OF TRIDIAGONAL LINEAR SYSTEMS 57 and the solution is ⎤⎡ x1 0.0085 ⎢ x2 ⎢ 0.1575 ⎢ ⎣ x3 ⎦ = ⎣ 0.2274 −0.2009 x4 ⎡ ⎤ ⎥ ⎦ 2. Write a computer code to solve the previous problem. Solution: The following is a MATLAB script that produced the previous solution function [delta, f, x] = trisol(l,d,u,b) n = length(d); x = zeros(n,1); for k=2:n d(k) = d(k) - u(k-1)*l(k)/d(k-1); b(k) = b(k) - b(k-1)*l(k)/d(k-1); end x(n) = b(n)/d(n); for k=(n-1):(-1):1 x(k) = (b(k) - u(k)*x(k+1))/d(k); end delta = d; f = b; 3. Use the algorithm of this section to solve the following system of equations: ⎤ ⎡ ⎤ ⎡ x1 8 6 1 0 0 ⎤ 2 4 1 0 ⎤ x2 ⎤ 13 ⎤ | ⎤ | ⎢ | ⎣ 0 1 4 2 ⎦ ⎣ x3 ⎦ = ⎣ 22 ⎦ . 0 0 1 6 x4 27 You should get the solution x = (1, 2, 3, 4)T . 4. Write a computer code to solve the previous problem. 5. The diagonal dominance condition is an example of a sufficient but not necessary condition. That is, the algorithm will often work for systems that are not diagonally

dominant. Show that the following system is not diagonally dominant, but then use the tridiagonal algorithm in this section to compute the solution to it: ⎡ ⎤ ⎡ 1 12 0 0 x1 2 ⎤ ⎢ 1 1 1 0 ⎥ ⎢x2 ⎥ 23/12 ⎢ 2 3 4 ⎥ ⎢ ⎥ ⎣ 0 1 1 1 ⎦ ⎣x3 ⎦ = ⎣ 53/30 ⎦ . 4 0 0 5 1 6 6 1 7 x4 You should get the solution x = (1, 2, 3, 4)T . 15/14 58 A SURVEY OF SIMPLE METHODS AND TOOLS Solution: The matrix fails to be diagonally dominant because of the values in the second, third, and fourth rows. However, if we apply the algorithm we get the triangular system ⎡ ⎤ ⎡ 1 x1 1 2.0000 0 0 2 1⎤ ⎢ 0 0.0833 ⎥ ⎢x2 ⎥ 0.9167 ⎢ 0 4 ⎥ ⎢ ⎥ ⎣ 1 0 ⎦ ⎣x3 ⎦ = ⎣ −0.9833 ⎦ , 0 −0.5500 6 0 0 0 0.1934 0.7734 x4 and then the correct values for the solution, x. 6. Use the tridiagonal algorithm in following system of equations: ⎡ 1 21 0 0⎤ ⎢ 1 1 1 0⎥ ⎢ 2 3 4⎥ ⎢ 0 1 1 1 4 5 6 0 0 16 71 this section to compute the solution to the ⎡x1 2 ⎤ ⎢x2 ⎥ ⎢ 2 ⎥ ⎢ ⎥ ⎣x3 ⎦ = ⎣ 53/30 15/14 x4 ⎦ ⎡ ⎤ ⎢. ⎥ Note that this is a very small change from the previous problem, since the only difference is that b2 has changed by only 1/12. How much has the answer changed? Solution: The solution is now ⎡ x1⎤ ⎢x2 ⎥ ⎢x3 x4 ⎦ 1.0037 ⎢ 1.9925 ⎥ ⎢=⎥ ⎣ 3.3358 ⎦ . 3.6082 ⎡ Considering the small change in the problem, this is a substantial change in the solution. 7. Write a computer code to do the previous two problems. 8. Verify that the following system is diagonally dominant and use the algorithm of this section to find the solution. ⎡ ⎤ ⎡ 61 ⎤ ⎡ 1 10 x1 0 0 2 21 42 1 ⎤ ⎢ ⎥ ⎢ 179 ⎥ ⎢ 1 1 ⎥ ⎢ 4 3 13 0 ⎥ ⎢x2 ⎥ = ⎢ 156 ⎥ . 1 1 ⎣ ⎦ ⎣ 0 1 x3 ⎦ ⎣ 563 ⎦ 5 0 0 4 1 6 21 1 5 x4 420 13 10 9. Use the algorithm of this section to find the solution to this system. ⎡ 1 10 ⎤ ⎡ 61 ⎤x1 0 0 2 21 42 1 ⎤ ⎢ 1 1 ⎥ ⎢ 180 ⎥ ⎢ 4 3 13 0 ⎥ ⎢x2 ⎥ = ⎢ 156 ⎥ . 1 1 ⎣ ⎦ 0 1 x3 ⎦ ⎣ 563 ⎦ 5 0 0 4 1 6 21 1 5 x4 420 13 SOLUTION OF TRIDIAGONAL LINEAR SYSTEMS 59 Note that the right side here is different from that in the previous problem by only a small amount in the b2 component. Comment on your results here as compared to those in the previous problem. Solution: The solution is now ⎡ x1⎤ ⎢x2 ⎥ ⎢x3 x4 ⎦ 0.7236 ⎢ 2.2902 ⎥ ⎢=⎥ ⎣ 2.7240 ⎦ . 4.2300 ⎡ Again, this is a very large change in the solution from the previous problem. 10. Write a computer code to do the previous two problems. 11. Write a code that carries out the tridiagonal solution algorithm, and test it on the following system of equations: Tx = b where T is 10 × 10 with ⎧ |i − j| = 1; ⎨ 1, tij = j + 1, i = j; ⎩ 0, otherwise. and bi = 1 for all i. Check your results by computing the residual r = b − T x. What is the largest component (in absolute value) of r? (You could also check your results by using MATLAB's backslash operator to solve the system.) 12. Extend the tridiagonal algorithm to a pentadiagonal matrix, i.e., one with five non-zero diagonals. Write a program to carry out this solution algorithm, and apply it to the system 4 ⎡ 1⎤ ⎢ 1 0⎥ ⎢ 2 4 1 1 1 1 4 2 ⎥ ⎢x1 0⎥ ⎢ 1 ⎥ ⎢x2 1 ⎥ ⎢x3 4 x4 ⎦ 1 ⎥ ⎢ 1 ⎥ ⎢=⎥ ⎣. ⎦ 1 ⎦ 1 ⎦ Check your results by again computing the residual vector, or by using MATLAB's backslash operation. 60 A SURVEY OF SIMPLE METHODS AND TOOLS Solution: A MATLAB script for doing this is given below. On the example in the exercise it returns the (exact) solution x = (1/7, 1/7, 1/7, 1/7)T . function [delta, f, x] = pentasol(k,l,d,u,v,b) n = length(d); x = zeros(n,1); d(2) = d(2) - l(2)*u(1)/d(1); u(2) = u(2) - l(2)*v(1)/d(1); b(2) = b(2) - l(2)*b(1)/d(1); for j=3:n l(j) = l(j) - k(3)*u(j-2)/d(j-2); d(j) = d(j) - k(j)*v(j-2)/d(j-2); b(j) = b(j) - k(j)*b(j-2)/d(j-2); d(j) = d(j) - l(j)*u(j-1)/d(j-1); if j < n u(j) = u(j) - l(j)*v(j-1)/d(j-1); end b(j) = b(j) - l(j)*b(j-1)/d(j-1); end delta = d; f = b; x = f; % x(n) = f(n)/d(n); x(n-1) = (b(n-1) - u(n-1)*x(n))/d(n-1); for j=(n-2):(-1):1 x(j) = (b(j) - u(j)*x(j+1) - v(j)*x(j+2))/d(j) end 13. Consider the family of tridiagonal problems defined by the matrix Kn ∈ Rn×n , with Kn = tridiag(−1, 2, −1) and a randomly defined right-hand side vector. (Use rand to generate the random vectors.) Solve the system over the range 4 ≤ n ≤ 100; use the flops command to estimate the number of operations required for each case, and plot the result as a function of n. Solution: You should get a nearly perfect straight line graph, since the cost is linear in the problem size. /•••. APPLICATION: SIMPLE TWO-POINT BOUNDARY VALUE PROBLEMS 2.7 61 APPLICATION: SIMPLE TWO-POINT BOUNDARY VALUE PROBLEMS Exercises: 1. Solve, by hand, the two-point BVP (2.28)-(2.29) when f (x) = x, using h = 14 . Write out the linear system explicitly prior to solution. You should get the following 3 × 3 system: ⎡ ⎤ ⎡ 2.0625 −1 0 U1 0.015625⎤ ⎢ −1 2.0625 −1 ⎥ ⎢ U2 ⎥ = ⎢ 0.03125 ⎥ 0 −1 2.0625 0.046875 U3⎦ Solution: The approximate solution is ⎡ ⎤ ⎡ U1 0.0349⎤ ⎢ U2 ⎥ = ⎢ 0.0563 ⎥ 0.0500 U3 2. Repeat the above, this time using h = 51 . What is the system now? 3. Repeat it again, this time using h = 18 . What is the system now? (For this problem, you probably will want to use a computer code to actually solve the system.) Solution: The approximate solution is ⎡ ⎤ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ U1 U2 U3 U4 U5 U6 U7 ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢=⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎣ ⎦ 0.0173 0.0331 0.0459 0.0540 0.0556 0.0486 0.0309 ⎤ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎥ ⎣ ⎦ 4. Write a program which solves the two-point BVP (2.28) - (2.29) where f is as given below. (a) f (x) = 4e−x − 4xe−x , u(x) = x(1 − x)e−x ; (b) f (x) = (π 2 + 1) sin πx, u(x) = sin πx; (c) f (x) = π(π sin πx + 2 cos πx)e−x , u(x) = e−x sin πx; (d) f (x) = 3 − 1 x − (x2 − x − 2) log x, u(x) = x(1 − x) log x. 62 A SURVEY OF SIMPLE METHODS AND TOOLS The exact solutions are as given. Using h−1 = 4, 8, 16, . . ., do we get the same kind of accuracy as in Table 2.10? Explain why or why not. Solution: The approximate solutions will display the kind of O(h2) accuracy suggested in the text, except for (d); in this case the singularity in the logarithm function in the solution affects the accuracy of the approximation. 5. Try to apply the ideas of this section to approximating the solution of the two point boundary value problem −u00 + u0 + u = 1, u(0) = 0, u(1) = 0. x ∈ [0, 1] Can we get a tridiagonal system that Algorithm 2.6 can be applied to? Hint: Consider some of the approximations from 2.2; use the most accurate ones that can be easily used. Solution: Use the approximation u0 (x) ≈ u(x + h) − u(x − h) 2h to construct the approximation. Using the approximation u0 (x) ≈ u(x + h) − u(x) h will result in a loss of accuracy. 6. Solve the two point boundary value problem problem −u00 + 64u0 + u = 1, u(0) = 0, u(1) = 0, x ∈ [0, 1] using a range of mesh sizes, starting with h = 1/4, 1/8, and going as far as h = 1/256. Comment on your results. Solution: For larger values of h, the approximate solution is erratic and wildly oscillatory. It isn't until about h ≤ 32 or so that the solution begins to settle down. 7. Generalize the solution of the two point boundary value problem to the case where u(0) = g0 6= 0 and u(1) = g1 6= 0. Apply this to the solution of the problem −u00 + u = 0, u(0) = 2, u(1) = 1 x ∈ [0, 1] APPLICATION: SIMPLE TWO-POINT BOUNDARY VALUE PROBLEMS 63 which has exact solution u(x) = e−2 e2 − 1 ex + 2e − 1 e2 − 1 e−x . Solve this for a range of values of the mesh. Do we get the expected O(h2) accuracy? 8. Consider the problem of determining the deflection of a thin beam, supported at both ends, due to a uniform load being placed along the beam. In one simple model, the deflection u(x) as a function of position x along the beam satisfies the boundary value problem −u00 + pu = qx(L − x), u(0) = u(L) = 0. 0 < x < L; Here p is a constant that depends on the material properties of the beam, L is the length of the beam, and q depends on the material properties of the beam as well as the size of the load placed on the beam. For a six foot long beam, with p = 7 × 10−6 and q = 4 × 10−7 , what is the maximum deflection of the beam? Use a fine enough grid that you can be confident of the accuracy of your results. Note that this problem is slightly more general than our example (2.28)-(2.29); you will have to adapt our method to this more general case. Solution: See Figure 2.5. −5 1.4 x 10 1.2 1 0.8 0.6 0.4 0.2 0 0 Figure 2.5 1 2 3 4 5 6 Exact solution for Exercise 2.7.8. 9. Repeat the above problem, except this time use a three foot long beam. How much does the maximum deflection change, and is it larger or smaller? 10. Repeat the beam problem again, but this time use a 12 foot long beam. 11. Try to apply the ideas of this section to the solution of the nonlinear boundary value problem defined by −u00 + eu = 0, 0 < x < 1; u(0) = u(1) = 0. 64 A SURVEY OF SIMPLE METHODS AND TOOLS Write out the systems of equations for the specific case of h = 41 . What goes wrong? Why can't we proceed with the approximate solution? Solution: The problem is that the system of equations that is produced by the discretization process is nonlinear, so our inherently linear algorithm won't work. /•••. CHAPTER 3 ROOT-FINDING 3.1 THE BISECTION METHOD Exercises: 1. Do three iterations (by hand) of the bisection method, applied to f (x) = x3 −2, using a = 0 and b = 2. Solution: We have f (a) = −2, f (b) = 6. The first iteration gives us c = 1, f (c) = −1, so the new interval is [a, b] = [1, 2]. The second iteration then gives us c = 3/2, f (c) = 11/8, so the new interval is [a, b] = [1, 3/2]. Finally, the third iteration then gives us c = 5/4, f (c) = −3/64, so the new interval is [a, b] = [5/4, 3/2]. Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c 2013 John Wiley & Sons, Inc. Copyright 65 66 ROOT-FINDING 2. For each of the functions listed below, do a calculation by hand (i.e., with a calculator) to find the root to an accuracy of 0.1. This will take at most 5 iterations for all of these, and fewer for several of them. 2 (a) f (x) = x − e−x , [a, b] = [0, 1]; 1 (b) f (x) = ln x + x, [a, b] = [10 , 1]; (c) f (x) = x3 − 3, [a, b] = [0, 3]; (d) f (x) = x6 − x − 1, [a, b] = [0, 2]; (e) f (x) = 3 − 2x , [a, b] = [0, 2]; 3. Write a program which uses the bisection method to find the root of a given function on a given interval, and apply this program to find the roots of the functions below on the indicated intervals. Use the relationship (3.2) to determine a priori the number of steps necessary for the root to be accurate to within 10−6 . (a) f (x) = x3 − 2, [a, b] = [0, 2]; (b) f (x) = ex − 2, [a, b] = [0, 1]; (c) f (x) = x − e−x , [a, b] = [0, 1]; (d) f (x) = x6 − x − 1, [a, b] = [0, 2]; (e) f (x) = x3 − 2x − 5, [a, b] = [0, 3]; (f) f (x) = 1 − 2xe−x/2 , [a, b] = [0, 2]; (g) f (x) = 5 − x−1 , [a, b] = [0.1, 0.25]; (h) f (x) = x2 − sin x, [a, b] = [0.1, π]. Solution: The solutions for some of the functions in this list are given in Table 3.1. For the sake of brevity we give only the number of iterations required plus the final answer. Table 3.1 Solutions to Problem 3.1.3 f (x) ex − 2 x3 − 2x − 5 x2 − sin x n 20 22 22 c 0.69314670562744 2.09455132484436 0.87672686579500 4. Use the bisection algorithm to solve the nonlinear equation x = cos x. Choose your own initial interval by some judicious experimentation with a calculator. 5. Use the bisection algorithm to solve the nonlinear equation x = exp−x . Again, choose your own initial interval by some judicious experimentation with a calculator. THE BISECTION METHOD 67 6. If you borrow L dollars at an annual interest rate of r, for a period of m years, then the size of the monthly payment, M , is given by the annuity equation L= 12M [1 − (1 + (r/12)−12m]. r The author needs to borrow $150,000 to buy the new house that he wants, and he can only afford to pay $600 per month. Assuming a 30 year mortgage, use the bisection method to determine what interest rate he can afford to pay. (Should the author perhaps find some rich relatives to help him out here?) Solution: This amounts to a root-finding problem for f (r) = L − 12M [1 − (1 + (r/12)−12m] r where L = 150, M = 0.6, and m = 30. The interest rate turns out to be r = 0.02593381500244, or, about 2.6%. The author is already checking for a rich relative. 7. What is the interest rate that the author can afford if he only has to borrow $100,000? 8. Consider the problem of modeling the position of the liquid-solid boundary in a substance that is melting due to the application of heat at one end. In a simplified model,2 if the initial position of the interface is taken to be x = 0, then the interface moves according to √ x = 2β t where β satisfies the nonlinear equation √ (TM − T0)k −β 2 /k √ e = βerf(β/ k). λ π Here TM is the melting temperature (absolute scale), T0 < TM is the applied temperature, k and λ are parameters dependent on the material properties of the substance involved, and erf(z) is the error function defined by Z z 2 2 e−t dt. erf(z) = √ π 0 MATLAB has an intrinsic error function, but if you are not using MATLAB, the error function can be accurately approximated by E(z) = 1 − (a1 ξ + a2 ξ 2 + a3 ξ 3)e−z 2 See 2 L.I. Rubinstein, The Stefan Problem, American Mathematical Society, Providence, RI, 1971. 68 ROOT-FINDING where ξ = 1/(1 + pz) and p = 0.47047, a1 = 0.3480242, a2 = −0.0958798, a3 = 0.747856. (a) Show that finding β is equivalent to finding the root α of the one-parameter family of functions defined by 2 f (z) = θe−z − zerf(z). (3.1) What is θ? How is α related to β? (b) Find the value of α corresponding to θ = 0.001, 0.1, 10, 1000. Use the bisection method, and E(z) to approximate the error function. Solution: √ √ (a) Make the change of variable z = β/ k, θ = (TM − T0)k/λ π. (b) For θ = 0.001, α = 0.0297; for θ = 0.1, α = 0.2895; for θ = 10, α = 1.4149; for θ = 1000, α = 2.4519. 9. A variation on the bisection method is known as regula falsi, or, the method of false position. Given an interval [a, b], with f (a)f (b) < 0, the new point c is defined by finding where the straight line connecting (a, f (a)) and (b, f (b)) crosses the axis. Show that this yields c = b − f (b)(b − a)/(f (b) − f (a)). 10. Do three iterations (by hand) of regula-falsi (see Problem 9), applied to f (x) = x3 − 2, using a = 0 and b = 2. Compare to your results for Problem 1. Solution: Regula-falsi produces the results shown in Table 3.2. Note that after three iterations, the interval is reduced to [1.0696, 2], whereas bisection would have reduced the interval to [1.25, 1.50], which is shorter. Thus bisection appears to be superior to regula-falsi, at least on this example. Table 3.2 Solutions to Problem 3.1.10 c 0.5000 0.8571 1.0696 f (c) -1.8750 -1.3703 -0.7763 a 0.5000 0.8571 1.0696 b 2.0000 2.0000 2.0000 11. Modify the bisection algorithm to perform regula falsi (see Problem 9), and use the new method to find the same roots as in Problem 3. Stop the program when the difference between consecutive iterates is less than 10−6 , i.e., when |xk+1 − xk | ≤ 10−6 , or when the function value satisfies |f (ck)| ≤ 10−6 . NEWTON'S METHOD: DERIVATION AND EXAMPLES 69 12. Repeat Problem 8(b), using your regula-falsi program. Solution: Regula-falsi finds the same values as bisection, but more slowly. 13. The bisection method will always cut the interval of uncertainty in half, but regula falsi might cut the interval by less, or might cut it by more. Do both 1 bisection and regula falsi on the function f (x) = e−4x − 10 , using the initial interval [0, 5]. Which one gets to the root the fastest? 14. Apply both bisection and regula-falsi to the following functions on the indicated intervals. Comment on your results in the light of how the methods are supposed to behave. (a) f (x) = 1/(x − 1), [a, b] = [0, 3]; (b) f (x) = 1/(x2 + 1), [a, b] = [0, 5]. Solution: This is kind of a trick question. Neither function has a root on the interval in question, but both methods will try to find it. Regula falsi lands on the singularity for f (x) = 1/(x − 1), but bisection doesn't, so keeps happily computing away. /•••. 3.2 NEWTON'S METHOD: DERIVATION AND EXAMPLES Exercises: 1. Write down Newton's method as applied to the function f (x) = x3 − 2. Simplify the computation as much as possible. What has been accomplished if we find the root of this function? Solution: xn+1 f (xn) f 0 (xn) 3 xn − 2 = xn − 3x2n = xn − = (2/3)(xn + (1/x2n)). When we find the root of this function, we have found 21/3 , the cube root of 2. 2. Write down Newton's method as applied to the function f (x) = x2 − 2. Simplify the computation as much as possible. What has been accomplished if we find the root of this function? 70 ROOT-FINDING 3. Generalize the preceeding two problems by writing down Newton's method as applied to f (x) = xn − a. Solution: xn+1 f (xn) f 0 (xn) N xn − a = xn − −1 N xN n −1 = (1/N)((N − 1)xn + (1/xN)). n = xn − When we find the root of this function, we have found a1/N , the N th root of a. 4. Write down Newton's method as applied to the function f (x) = a − x−1 . Simplify the resulting computation as much as possible. What has been accomplished if we find the root of this function? 5. Do three iterations of Newton's method (by hand) for each of the following functions: (a) f (x) = x6 − x − 1, x0 = 1; Solution: We get: x1 x2 x3 −1 = 1.2; 5 0.785984 = 1.143575843; = 1.2 − f (1.2)/f 0 (1.2) = 1 − 13.92992 = 1.143575843 − f (1.143575843)/f 0 (1.143575843) 0.93031963 = 1.143575843 − 10.73481139 = 1.134909462. = 1 − f (1)/f 0 (1) = 1 − (b) f (x) = x + tan x, x0 = 3; (c) f (x) = 1 − 2xe−x/2 , x0 = 0; (d) f (x) = 5 − x−1 , x0 = 21 ; (e) f (x) = x2 − sin x, x0 = 12 ; (f) f (x) = x3 − 2x − 5, x0 = 2; Solution: We get: x1 x2 x3 = 2 − f (2)/f 0 (2) = 2.1; = 2.1 − f (2.1)/f 0 (2.1) = 2.094568121; = 2.094568121 − f (2.094568121)/f 0 (2.094568121) = 2.094551482; NEWTON'S METHOD: DERIVATION AND EXAMPLES 71 (g) f (x) = ex − 2, x0 = 1; Solution: We get: x1 x2 x3 = = = = = 1 − f (1)/f 0 (1) = .7357588825; .7357588825 − f (.7357588825)/f 0 (1.7357588825) .6940422999; .6940422999 − f (.6940422999)/f 0 (.6940422999) .6931475811; (h) f (x) = x3 − 2, x0 = 1; Solution: We get: x1 x2 x3 = = = = = 1 − f (1)/f 0 (1) = 1.333333333; 1.333333333 − f (1.333333333)/f 0 (1.333333333) 1.263888889; 1.263888889 − f (1.263888889)/f 0 (1.263888889) 1.259933493; (i) f (x) = x − e−x , x0 = 1; Solution: We get: x1 x2 x3 = = = = = 1 − f (1)/f 0 (1) = .5378828427; .5378828427 − f (.5378828427)/f 0 (.5378828427) .5669869914; .5669869914 − f (.5669869914)/f 0 (.5669869914) .5671432860; (j) f (x) = 2 − x−1 ln x, x0 = 13 . 6. Do three iterations of Newton's method for f (x) = 3 − ex , using x0 = 1. Repeat, using x0 = 2, 4, 8, 16. Comment on your results. 7. Draw the graph of a single function f which satisfies all of the following: (a) f is defined and differentiable for all x; (b) There is a unique root α > 0; (c) Newton's method will converge for any x0 > α; (d) Newton's method will diverge for all x0 < 0. Note that the requirement here is to find a single function that satisfies all of these conditions. 72 ROOT-FINDING 45 40 35 30 25 20 15 10 5 0 −5 −6 −4 Figure 3.1 −2 0 2 4 6 Solution for Problem 3.2.7 Solution: Figure 3.1 shows a possible solution. 8. Draw the graph of a single function f which satisfies all of the following: (a) f is defined and differentiable for all x; (b) There is a single root α ∈ [a, b], for some interval [a, b]; (c) There is a second root, β < a; (d) Newton's method will converge to α for all x0 ∈ [a, b]; (e) Newton's method will converge to β for all x0 ≤ β. Note that the requirement here, again, is to find a single function that satisfies all of these conditions. 9. Write down Newton's method for finding the root of the arctangent function. From this formulate an equation which must be satisfied by the value x = β, in order to have the Newton iteration cycle back and forth between β and −β. Hint: If xn = β, and Newton's method is supposed to give us xn+1 = −β, what is an equation satisfied by β? Solution: Newton's method applied to the arctangent function is xn+1 = xn − (x2n + 1)(arctan xn). What we want to happen is that x0 = β leads to x1 = −β. This means we have −β = β − (β 2 + 1)(arctan β), HOW TO STOP NEWTON'S METHOD 73 or, 2β − (β 2 + 1)(arctan β) = 0. Graphing this function shows three roots: one at β = 0, and two at β ≈ ±1.39. We want one of these last two. 10. Compute the value of β from the previous problem. Hint: Use bisection to solve the equation for β. 11. Use Newton's method on the computer of your choice to compute the root α = 0 of the arctangent function. Use the value of β from the previous problem as your x0 and comment on your results. Repeat using x0 = β/2 and x0 = β − where is O(u). Solution: Table 3.3 shows the first five iterates using the value of β found in Problem 10. Note that the iteration is not exactly cycling back and forth between β and −β. This is because of rounding error in the computation. If we use x0 = β/2, then the computation converges to the root at x = 0 very quickly: x3 ≈ 10−7 . Taking x0 closer to β only slows down this convergence. For example, taking x0 = (0.999)β yields x5 ≈ 0.36. Table 3.3 Solution to Problem 3.2.11 k 1 2 3 4 5 xk -1.39174467908955 1.39174382537168 -1.39174157322777 1.39173563198669 -1.39171995883655 /•••. 3.3 HOW TO STOP NEWTON'S METHOD Exercises: 1. Under the assumption that f 0 (α) 6= 0 and xn → α, prove (3.10); be sure to provide all the details. Hint: Expand f and f 0 in Taylor series about x = α. 74 ROOT-FINDING Solution: We have ⎛ 0 f (xn) f (xn−1) Cn = f (xn−1) f 0 (cn) ⎞ 0 f (xn) − f (α) f (α) + (xn−1 − α)f 00 (ξ3) = f (xn−1) − f (α) f 0 (α) + (cn − α)f 00 (ξ4) ⎜⎛ 0 f (xn)−f (α) f (α) + (xn−1 − α)f 00 (ξ3) xn − α ⎞⎟ = f (xn−1)−f (α) f 0 (α) + (cn − α)f 00 (ξ4) xn−1 −α 0 0 f (ηn) f (α) + (xn−1 − α)f 00 (ξ3) = f 0 (ηn−1) f 0 (α) + (cn − α)f 00 (ξ4) where ηn is between xn and α, ηn−1 is between xn−1 and α, ξ3 is between xn−1 and α, and ξ4 is between cn and α. Therefore, as n → ∞, we have ηn → α, ηn−1 → α, ξ3 → α and ξ4 → α. Thus 0 0 f (α) f (α) Cn → = 1, f 0 (α) f 0 (α) so long as f 00 is bounded. 2. We could also stop the iteration when |f (xn)| was sufficiently small. Use the Mean Value Theorem plus the fact that f (α) = 0 to show that, if f 0 is continuous and non-zero near α, then there are constants c1 and c2 such that c1 |f (xn)| ≤ |α − xn | ≤ c2 |f (xn)|. Comment on this result. 3. Write a computer program that uses Newton's method to find the root of a given function, and apply this program to find the root of the following functions, using x0 as given. Stop the iteration when the error as estimated by |xn+1 −xn | is less than 10−6 . Compare to your results for bisection. (a) f (x) = 1 − 2xe−x/2 , x0 = 0; (b) f (x) = 5 − x−1 , x0 = 21 ; (c) f (x) = x3 − 2x − 5, x0 = 2; (d) f (x) = ex − 2, x0 = 1; (e) f (x) = x − e−x , x0 = 1; (f) f (x) = x6 − x − 1, x0 = 1; (g) f (x) = x2 − sin x, x0 = 12 ; (h) f (x) = x3 − 2, x0 = 1; (i) f (x) = x + tan x, x0 = 3; HOW TO STOP NEWTON'S METHOD 75 Table 3.4 Solutions to Problem 3.3.3 f (x) x3 − 2x − 5 ex − 2 x6 − x − 1 x2 − sin x n 4 4 5 8 c 2.0946 0.6931 1.1347 0.8767 (j) f (x) = 2 − x−1 ln x, x0 = 31 . Solution: The results, obtained from a mid-range personal computer using MATLAB, are summarized for some functions in Table 3.4. 4. Figure 3.2 shows the geometry of a planetary orbit3 around the sun. The position of the sun is given by S, the position of the planet is given by P . Let x denote the angle defined by P0 OA, measured in radians. The dotted line is a circle concentric to the ellipse and having a radius equal to the major axis of the ellipse. Let T be the total orbital period of the planet, and let t be the time required for the planet to go from A to P . Then Kepler's equation from orbital mechanics, relating x and t, is x − sin x = 2πt . T Here is the eccentricity of the elliptical orbit (the extent to which it deviates from a circle). For an orbit of eccentricity = 0.01 (roughly equivalent to that of the Earth), what is the value of x corresponding to t = T /4? What is the value of x corresponding to t = T /8? Use Newton's method to solve the required equation. Solution: For t = T /4, Newton's method yielded x = 1.580795827 in 3 iterations; for t = T /8, we get x = .7925194063, also in 3 iterations. Both computations used x0 = 0. 5. Consider now a highly eccentric orbit, such as that of a comet, for which = 0.9 might be appropriate. What is the value of x corresponding to t = T /4? What is the value of x corresponding to t = T /8? 6. Consider the problem of putting water pipes far enough underground to avoid frozen pipes should the external temperature suddenly drop. Let T0 be the initial temperature of the ground, and assume that the external air temperature 3 For the background material for this and the next problem, the author is indebted to the interesting new calculus text by Alexander J. Hahn, Basic Calculus; From Archimedes to Newton to its Role in Science, published in 1998 by Springer-Verlag. 76 ROOT-FINDING P0 P S O Figure 3.2 A Orbital geometry for Problems 4 and 5 suddenly drops to a new value T < T0 . Then a simple model of how the underground temperature responds to the change in external temperature tells us that u(x, t) − T x √ . = erf T0 − T 2 at Here u(x, t) is the temperature at a depth x feet and time t seconds after the temperature change, and a is the thermal conductivity of the soil. Suppose that a = 1.25 × 10−6 ft2 /sec and that the initial ground temperature is T0 = 40 degrees. How deep must the pipe be buried to guarantee that the temperature does not reach 32 degrees Fahrenheit for 30 days after a temperature shift to T = 0 degrees Fahrenheit? If your computing environment does not have an intrinsic error function, use the approximation presented in Problem 8 of 3.1. 7. In the previous problem, produce a plot of temperature u at depths of 6 feet and 12 feet, as a function of time (days), in response to a temperature shift of 40 degrees for the following initial temperatures (in degrees Fahrenheit): (a) T0 = 40; (b) T0 = 50; (c) T0 = 60. Solution: This really is not a root-finding problem. Simply plot 6 √ U (t) = u(6, t) = 40 + (T0 − 40)erf 2 at and similarly for u(12, t), over the interval 0 ≤ t ≤ 86, 400. This produces a plot in which time is measured in seconds; it is an easy change of scale to convert it to days. APPLICATION: DIVISION USING NEWTON'S METHOD 77 8. Use Newton's method to solve Problem 6 from 3.1. 9. Repeat the above, assuming that the mortgage is only 15 years in length. Solution: This amounts to a root finding problem for f (r) = L − 12M [1 − (1 + (r/12)−12m] r for L = 150, M = 0.6, and m = 15. Newton's method finds a negative value of r, which implies that there is no interest rate which meets the specified conditions. /•••. 3.4 APPLICATION: DIVISION USING NEWTON'S METHOD Exercises: 1. Test the method derived in this section by using it to approximate 1/0.75 = 1.333..., with x0 as suggested in the text. Don't write a computer program, just use a hand calculator. Solution: We get the following sequence of values: x1 = 1.312500 x2 = 1.333007812 x3 = 1.333333254 x4 = 1.333333333 x5 = 1.333333333 2. Repeat the above for a = 2/3. 3. Repeat the above for a = 0.6. Solution: This time we get: x1 = 1.656 x2 = 1.6665984 x3 = 1.666666664 x4 = 1.666666667 78 ROOT-FINDING x5 = 1.666666667 4. Based on the material in this section, write each of the following numbers in the form a = b × 2k where b ∈ [12 , 1], and then use Newton's method to find the reciprocal of b and hence of a. (a) a = 7; Solution: We have a = (7/8) × 23 , so b = 7/8. Newton's method produces b−1 = 1.142857143 in four iterations, therefore a−1 = b−1 × 0.125 = 0.1428571429. (b) a = π; (c) a = 6; (d) a = 5; Solution: We have 5 = (5/23) × 23 = 5/8 × 8, so b = 5/8 = 0.625. From the text we take x0 = 3 − 2(0.625) = 1.75. We get the results in Table 3.5. Notice that we converge much faster than expected, and that we have 1/5 = 1/1.6 × 8 = 0.2, which is the correct result. Table 3.5 Newton iteration for Problem 4c, Sec. 3.4. n 1 2 3 4 5 6 xn 0.158594E+01 0.159988E+01 0.160000E+01 0.160000E+01 0.160000E+01 0.160000E+01 (e) a = 3. Solution: We have a = (3/4) × 22 , APPLICATION: DIVISION USING NEWTON'S METHOD 79 so b = 3/4. Newton's method produces b−1 = 1.333333333 in four iterations, therefore a−1 = b−1 × 0.25 = 0.333333333. Be sure to use the initial value as generated in this section and only do as many iterations as are necessary for 10−16 accuracy. Compare your values to the intrinsic reciprocal function on your computer or calculator. Note this can be done on computer or hand calculator. 5. Test the method derived in this section by writing a computer program to implement it. Test the program by having it compute 1/0.75 = 1.333..., with x0 as suggested in the text, as well as 1/(2/3) = 1.5. Solution: The following is a MATLAB script that accomplishes the desired task. function r = recip(a) x = 3 - 2*a; for k=1:6 x = x*(2 - a*x); end r = x; 6. How close an initial guess is needed for the method to converge to within a relative error of 10−14 in only three iterations? Solution: To get the relative error after 3 iterations less than 10−14 , we need to have e e 8 3 0 ≤ 10−14 . = α α Therefore, the initial error |e0 | must satisfy |e0 | ≤ α10−14/8 = 10−14/8 = 0.0178. 7. Consider the quadratic polynomial q2 (x) = 4.328427157 − 6.058874583x + 2.745166048x2 . What is the error when we use this to generate the initial guess? How many steps are required to get 10−20 accuracy? How many operations? 8. Repeat Problem 7, this time with the polynomial q3 (x) = 5.742640783 − 12.11948252x + 11.14228488x2 − 3.767968453x3 . Solution: This time we are looking at the difference g(x) = 1 1 − q3 (x) = − 5.742640783 + 12.11948252x − 11.14228488x2 x x +3.767968453x3 . 80 ROOT-FINDING The derivative is therefore g 0 (x) = (−1/x2) + 12.11948252 − 22.28456976x + 11.30390x2 . Setting this equal to zero and solving leads us to the polynomial equation −1 + 12.11948252x2 − 22.28456976x3 + 11.30390x4 = 0. Plotting this shows there are three roots on the interval [1/2, 1]: α1 ≈ 0.56, α2 ≈ 0.73, and α3 ≈ 0.92. Using Newton's method to refine each of these values gives us α1 = .5621750583, α2 = .7285533588, α3 = .9163783254. These are our critical points, at which we need to evaluate the error g(x) (along with the endpoints). We get g(1/2) g(1) g(α1) g(α2) g(α3) = = = = = .0025253136 .002525310 −.0025253184 .002525312 −.002525321 thus showing that the maximum value of |g(x)| — hence, the maximum value of the initial error using q3 to compute x0 — is about 0.0025. To get 10−20 accuracy now requires only 3 iterations, and a total of 15 operations, again. 9. Modify your program from Problem 5 to use each of p2 , q2 , and q3 to compute 1/a for a = 0.75 and a = 0.6. Compare to the values produced by ordinary division on your computer. (Remember to use Horner's rule to evaluate the polynomials!) 10. Modify the numerical method to directly compute the ratio b/a, rather than just the reciprocal of a. Is the error analysis affected? (Note: Your algorithm should approximate the ratio b/a without any divisions at all.) 11. Test your modification by using it to compute the value of the ratio 4/5 = 0.8. /•••. THE NEWTON ERROR FORMULA 3.5 81 THE NEWTON ERROR FORMULA Exercises: 1. If f is such that |f 00 (x)| ≤ 3 for all x and |f 0 (x)| ≥ 1 for all x, and if the initial error in Newton's method is less than 12 , what is an upper bound on the error at each of the first three steps? Hint: Use the Newton error formula from this section. Solution: We have, from the Newton error formula, α − xn+1 = − f 00 (ξn 2f 0 (xn) so that, using en = |α − xn |, e1 ≤ e20 e2 ≤ e21 e3 ≤ e22 3 ≤ (1/4)(3/2) = 3/8; 2×1 3 ≤ (9/64)(3/2) = 27/128; 2×1 3 ≤ (729/16384)(3/2) = 2187/32768 = 0.0667... 2×1 2. If f is now such that |f 00 (x)| ≤ 4 for all x but |f 0 (x)| ≥ 2 for all x, and if the initial error in Newton's method is less than 31 , what is an upper bound on the error at each of the first three steps? 3. Consider the left-hand column of data in Table 3.6 (Table 3.4 in the text.). Supposedly this comes from applying Newton's method to a smooth function whose derivative does not vanish at the root. Use the limit result (3.14) to determine whether or not the program is working. Hint: Use α ≈ x7 . Solution: Use α ≈ x7 from the table, and compute values of the ratio rn = α − xn+1 (α − xn)2 If the method is converging according to the theory, this ratio should approach a constant value. For the data in this table we get r1 r3 r5 = −0.09523809523810, = −0.21215771728295, = −0.22031852835393 r2 = −0.16122840690979, r4 = −0.22329830363589, (Because our approximate α is x7 we can't use r6 .) Since these values are settling down around −0.22 we are confidant that the program is working properly. 82 ROOT-FINDING Table 3.6 Data for Problems 3 and 4. n 0 1 2 3 4 5 6 7 8 9 10 xn (Prob. 3) 10.0 5.25 3.1011904761905 2.35673727264442 2.2391572227372 2.

polynomial of degree 2 that interpolates to y = x3 at the nodes x0 = 0, x1 = 1, and x2 = 2. Plot y = x3 and the interpolating polynomial over the interval [0, 2]. Solution: p2 (t) = L0 (t)y0 + L1 (t)y1 + L2 (t)y2 (t − x1)(t − x2) (t − x0)(t − x2) (t − x0)(t − x1) = (1) + (4) + (8) (0 − 1)(0 − 2) (1 − 0)(1 − 2) (2 − 0)(2 − 1) = 3t2 − 2t. 3. Construct the quadratic polynomial that interpolates to y = 1/x at the nodes x0 = 1/2, x1 = 3/4, and x2 = 1. Solution: p2 (t) = L0 (t)y0 + L1 (t)y1 + L2 (t)y2 (t − 1/2)(t − 1) (t − 3/4)(t − 1) (2) + (4/3) = (1/2 − 3/4)(1/2 − 1) (3/4 − 1)(3/4 − 1/2) (t − 1/2)(t − 3/4) (1) + (1 − 1/2)(1 − 3/4) = (8/3)t2 − 6t + (13/3). 4. Construct the quadratic polynomial that interpolates to y = x0 = 1/4, x1 = 9/16, and x2 = 1. √ x at the nodes Solution: p2 (t) = L0 (t)y0 + L1 (t)y1 + L2 (t)y2 (t − x1)(t − x2) (t − x0)(t − x2) = (1/2) + (3/4) (1/4 − 9/16)(1/4 − 1) (9/16 − 1/4)(9/16 − 1) (t − x0)(t − x1) (1) + (1 − 1/4)(1 − 9/16) = (−32/105)t2 + (22/21)t + (9/35). 5. For each function listed below, construct the Lagrange interpolating polynomial for the set of nodes specified. Plot both the function and the interpolating polynomial, and also the error, on the interval defined by the nodes. (a) f (x) = ln x, xi = 1, 23 , 2; √ (b) f (x) = x, xi = 0, 1, 4; (c) f (x) = log2 x, xi = 1, 2, 4; (d) f (x) = sin πx, xi = −1, 0, 1. LAGRANGE INTERPOLATION 119 Solution: (a) p2 (x) = −.235566072x2 + 1.399845540x − 1.164279325; (b) p2 (x) = (7x − x2)/6; (c) p2 (x) = −(1/6)x2 + (3/2)x − (4/3); (d) p2 (x) = 0. 6. Find the polynomial of degree 3 that interpolates y = x3 at the nodes x0 = 0, x1 = 1, x2 = 2, and x3 = 3. (Simplify your interpolating polynomial as much as possible.) Hint: this is easy if you think about the implications of the uniqueness of the interpolating polynomial. Solution: You should get p3 (x) = x3 ; interpolation to a polynomial reproduces the polynomial if enough nodes are used. 7. Construct the Lagrange interpolating polynomial to the function f (x) = x2 + 2x, using the nodes x0 = 0, x1 = 1, x2 = −2. Repeat, using the nodes x0 = 2, x1 = 1, x2 = −1. (For both sets of nodes, simplify your interpolating polynomial as much as possible.) Comment on your results, especially in light of the uniqueness part of Theorem 4.1, and then write down the interpolating polynomial for interpolating to f (x) = x3 + 2x2 + 3x + 1 at the nodes x0 = 0, x1 = 1, x2 = 2, x3 = 3, x4 = 4, and x5 = 5. Hint: You should be able to do this last part without doing any computations. Solution: In all three cases the interpolating polynomial will be identical to the original polynomial. 8. Let f be a polynomial of degree ≤ n, and let pn be a polynomial interpolant to f , at the n + 1 distinct nodes x0 , x1 , . . . , xn . Prove that pn (x) = f (x) for all x, i.e., that interpolating to a polynomial will reproduce the polynomial, if you use enough nodes. Hint: Consider the uniqueness part of Theorem 4.1. Solution: Let q(x) = f (x) − pn (x). Since both f and pn are polynomials of degree ≤ n, so is q. But q(xk) = 0 for the n + 1 nodes xk . The only way a polynomial of degree ≤ n can have n + 1 roots is if it is identically zero, hence f (x) = pn (x) for all x. 9. Let p be a polynomial of degree ≤ n. Use the uniqueness of the interpolating polynomial to show that we can write p(x) = n X (n) Li (x)p(xi) i=0 for any distinct set of nodes x0 , x1 , . . . , xn . Hint: See the previous problem. 120 INTERPOLATION AND APPROXIMATION Solution: Define q(x) = p(x) − n X (n) Li (x)p(xi). i=0 Since everything on the right side is a polynomial of degree ≤ n, so is q. But, again, q(xk) = 0 for the n + 1 nodes xk , therefore q(x) = 0 for all x, which proves the desired result. 10. Show that n X (n) Li (x) = 1 i=0 for any set of distinct nodes xk , 0 ≤ k ≤ n. Hint: This does not require any computation with the sum, but rather a perceptive choice of polynomial p in the previous problem. Solution: Let p(x) = 1, and note that this is a polynomial of degree 0. By the previous problem, we have that 1 = p(x) = n X (n) Li (x)p(xi) = i=0 n X (n) Li (x), i=0 and we are done. /•••. 4.2 NEWTON INTERPOLATION AND DIVIDED DIFFERENCES Exercises: 1. Construct the polynomial of degree 3 that interpolates to the data x0 = 1, y0 = 1, x1 = 2, y1 = 1/2, x2 = 4, y2 = 1/4, and x3 = 3, y3 = 1/3. You should get p(t) = (50 − 35t + 10t2 − t3)/24. Solution: The divided difference coefficients are 1, −1/2, 1/8, and −1/24, so the polynomial is p(t) = 1 − (1/2)(x − 1) + (1/8)(x − 1)(x − 2) − (1/24)(x − 1)(x − 2)(x − 4), which simplifies as indicated. 2. For each function listed below, use divided difference tables to construct the Newton interpolating polynomial for the set of nodes specified. Plot both the NEWTON INTERPOLATION AND DIVIDED DIFFERENCES 121 function and the interpolating polynomial, and also the error, on the interval defined by the nodes. √ (a) f (x) = x, xi = 0, 1, 4; Solution: The divided difference coefficients are a0 = 0, a1 = 1, and a2 = −1/6. (b) f (x) = ln x, xi = 1, 32 , 2; (c) f (x) = sin πx, xi = 0, 41 , 12 , 34 , 1; (d) f (x) = log2 x, xi = 1, 2, 4; Solution: The divided difference coefficients are a0 = 0, a1 = 1, and a2 = 1/6, so the polynomial is p(t) = 0 + (1)(x − 1) + (1/6)(x − 1)(x − 2), which simplifies to p(t) = (1/6)(x2 + 3x − 4). (e) f (x) = sin πx, xi = −1, 0, 1. 3. Let f (x) = ex . Define pn (x) to be the Newton interpolating polynomial for f (x), using n + 1 equally spaced nodes on the interval [−1, 1]. Thus we are taking higher and higher degree polynomial approximations to the exponential function. Write a program that computes pn (x) for n = 2, 4, 8, 16, 32, and which samples the error f (x) − pn (x) at 501 equally spaced points on [−1, 1]. Record the maximum error as found by the sampling, as a function of n, i.e., define En as En = max |f (tk) − pn (tk)| 0≤k≤500 where tk = −1 + 2k/500, and plot En vs. n. Solution: Using MATLAB on a low-end personal computer, the author got E2 E16 = 7.8525e − 2, E4 = 1.1244e − 3, E8 = 5.8000e − 8, = 1.5321e − 14, E32 = 3.6471e − 10. Note that the error started to actually increase as we added more nodes. This is explained, in part, by the material in Section 4.11.1. 4. In 3.7 we used linear interpolation to√construct an initial guess for Newton's method as a means of approximating a. Construct the quadratic polynomial that interpolates the square root√ function at the nodes x0 = 14 , x1 = 94 , x2 = 1. Plot the error between p2 and x over the interval [41 , 1] and try to estimate the worst √error. What impact will this have on the use of Newton's method for finding a? 122 INTERPOLATION AND APPROXIMATION Solution: The polynomial is p(t) = − 12 2 23 26 t + t+ . 35 21 105 √ The worst absolute error, | t−p(t)|, appears to be, from looking at a plot, about 0.01. This is significantly smaller than √ the error due to linear interpolation, so the Newton iteration for computing a will converge faster. See some of the exercises in Section 3.7. 5. Prove Corollary 4.1. Solution: A straight-forward inductive argument works. 6. Write and test a computer code that takes a given set of nodes, function values, and corresponding divided difference coefficients, and computes new divided difference coefficients for new nodes and function values. Test it by recursively computing interpolating polynomials that approximate f (x) = ex on the interval [0, 1]. Solution: A set of MATLAB scripts that accomplish this are given below. The first script (polyterp) is the"main program." The second one (divdif) computes divided difference coefficients. The third one (intval) evaluates an interpolating polynomial from nodal data and divided difference coefficients. % n = input('Initial degree? '); m = input('Maximum degree? '); np = n + 1; mp = m + 1; % xn = [0:n]/n yn = exp(xn); x = [0:500]/500; a = divdif(xn,yn); y = intval(x,xn,a); figure(1) plot(x,y) figure(2) plot(x,y - exp(x)); max(abs(y - exp(x))) % for k=np:mp xx = input('New node value? '); yy = exp(xx); pn = intval(xx,xn,a); w = xx - xn; NEWTON INTERPOLATION AND DIVIDED DIFFERENCES 123 w = prod(w); xn = [xn xx]; yn = [yn yy]; aa = (yy - pn)/w; a = [a aa]; y = intval(x,xn,a); max(abs(y - exp(x))) check = sum(abs(yn - intval(xn,xn,a))) end %% function a = divdif(x,y) n = length(x)-1; a = y; for i=1:n for j=n:(-1):i a(j+1) = (a(j+1) - a(j))/(x(j+1) - x(j-i+1)); end end %% function y = intval(x,xx,d) % % evaluates polynomial using divided difference % coefficients d, and interpolation data xx % m = length(x); n = length(d); % y = d(n)*ones(1,m); for i=(n-1):(-1):1 y = d(i) + (x - xx(i)*ones(1,m)).*y; end 7. In 1973, the horse Secretariat became the first (and, so far, only) winner of the Kentucky Derby to finish the race in less than two minutes, running the 1 14 mile distance in 1 : 59.4 seconds4 . Remarkably, he ran each quarter mile faster than the previous one, as the following table of data shows. Here t is the elapsed time (in seconds) since the race began and x is the distance (in miles) that Secretariat has travelled. 4 During the final preparation of the First Edition of the text, in May, 2001, the horse Monarchos won the Kentucky Derby in a time of 1 : 59.8 seconds, becoming the second winner of the race to finish in less than two minutes. Ironically, the only other horse to run the Derby in less than two minutes was Sham, who finished second to Secretariat. 124 INTERPOLATION AND APPROXIMATION Table 4.1 Data for Problem 7. x t 0.0 0.0 0.25 25.0 0.50 49.4 0.75 73.0 1.00 96.4 1.25 119.4 (a) Find the cubic polynomial that interpolates this data at x = 0, 1/2, 3/4, 5/4. Solution: T (x) = p3 (x) = 2.986666666 x3 − 9.599999999 x2 + 102.8533333 x (b) Use this polynomial to estimate Secretariat's speed at the finish of the race, by finding p03 (5/4). Solution: We need to use a little calculus here. We have t = P3 (x), but the speed of the horse comes from computing dx/dt, not dt/dx. But we know that dx 1 = dt , dt dx so the speed of the horse at x = 5/4 is given by S= 1 p03 (5/4) = .1076967261e − 1. Unfortunately, this is in miles per second and we wanted miles per hour. To convert, we simply multiply by 3600 seconds per hour to get S = 38.77082140. (c) Find the quintic polynomial that interpolates the entire dataset. Solution: p5 (x) = −13.65333333 x5 + 42.66666667 x4 − 44.80000001 x3 + 13.33333333 x2 + 98.85333333 x. (d) Use the quintic polynomial to estimate Secretariat's speed at the finish line. Solution: This time we get S = 40.51620641 miles per hour. 8. The data in Table 4.2 (Table 4.7 in the text) gives the actual thermal conductivity data for the element mercury. Use Newton interpolation and the data for 300◦ K, 500◦ K, and 700◦ K to construct a quadratic interpolate for this data. How well does it predict the values at 400K and 600K? NEWTON INTERPOLATION AND DIVIDED DIFFERENCES 125 Table 4.2 Data for Problem 8. ◦ Temperature (K), u Conductivity (watts/cm ◦ K), k 300 0.084 400 0.098 500 0.109 600 0.12 700 0.127 Solution: The polynomial is p2 (T) = −0.0000000875 T 2 + 0.000195 T + 0.033375. Since p2 (400) = .97375e − 1 and p2 (600) = .118875 it is apparent that the interpolating polynomial does a good job of matching the data. 9. The gamma function, denoted by Γ(x), is an important special function in probability, combinatorics, and other areas of applied mathematics. Because it can be shown that Γ(n + 1) = n!, the gamma function is considered a generalization of the factorial function to non-integer arguments. Table 4.3 (Table 4.8 in the text) gives the values of Γ(x) on the interval [1, 2]. Use these to construct the fifth degree polynomial based on the nodes x = 1, 1.2, 1.4, 1.6, 1.8, 2.0, and then use this polynomial to estimate the values at x = 1.1, 1.3, 1.5, 1.7, 1.9. Plot your polynomial and compare it to the intrinsic gamma function on your computing system or calculator. Table 4.3 Table of Γ(x) values. x 1.00 1.10 1.20 1.30 1.40 1.50 1.60 1.70 1.80 1.90 2.00 Γ(x) 1.0000000000 0.9513507699 0.9181687424 0.8974706963 0.8872638175 0.8862269255 0.8935153493 0.9086387329 0.9313837710 0.9617658319 1.0000000000 Solution: We get p5 = −.09270377605x5 + .8632574013x4 − 3.199846710x3 +6.287123228x2 − 6.537486675x + 3.679656532. 126 INTERPOLATION AND APPROXIMATION and p5 (1.1) = .951439127, p5 (1.3) = .897445725, p5 (1.7) = .908619509, p5 (1.5) = .886242424, p5 (1.9) = .961817632, which is pretty good agreement. 10. The error function, which we saw briefly in Chapters 1 and 2, is another important special function in applied mathematics, with applications to probability theory and the solution of heat conduction problems. The formal definition of the error function is as follows. Z x 2 2 erf(x) = √ e−t dt. π 0 Table 4.4 (Table 4.9 in the text) gives values of erf(x) in increments of 0.1, over the interval [0, 1]. Table 4.4 Table of erf(x) values for Problem 10. x 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 erf(x) 0.00000000000000 0.11246291601828 0.22270258921048 0.32862675945913 0.42839235504667 0.52049987781305 0.60385609084793 0.67780119383742 0.74210096470766 0.79690821242283 0.84270079294971 (a) Construct the quadratic interpolating polynomial to the error function using the data at the nodes x0 = 0, x1 = 0.5, and x2 = 1.0. Plot the polynomial and the data in the table and comment on the observed accuracy. Solution: The divided difference coefficients are a0 = 0, a1 = 1.0410e + 0, a2 = −3.9660e − 1 and the plot of the interpolating polynomial is given in Figure 4.1. This is very good agreement. NEWTON INTERPOLATION AND DIVIDED DIFFERENCES 127 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 0 0.1 0.2 Figure 4.1 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 Solution plot for Exercise 4.2.11a. (b) Repeat the above, but this time construct the cubic interpolating polynomial using the nodes x0 = 0.0, x2 = 0.3, x2 = 0.7, and x3 = 1.0. Solution: The divided difference coefficients are a0 = 0, a1 = 1.0954e+0, a2 = −3.1784e−1, a3 = −1.4398e−1 and the plot of the interpolating polynomial is given in Figure 4.2. This is slightly better than in (a). 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 0 0.1 0.2 Figure 4.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 Solution plot for Exercise 4.2.11b. 1 128 INTERPOLATION AND APPROXIMATION 11. As steam is heated up, the pressure it generates is increased. Over the temperature range [220, 300] (degrees Fahrenheit) the pressure, in pounds per square inch, is given in Table 4.5.5 (This is Table 4.10 in the text.) Table 4.5 Temperature-pressure values for steam; Problem 11 T P 220 17.188 230 20.78 240 24.97 250 29.82 260 35.42 270 41.85 280 49.18 290 57.53 300 66.98 (a) Construct the quadratic interpolating polynomial to this data at the nodes T0 = 220, T1 = 260, and T2 = 300. Plot the polynomial and the data in the table and comment on the observed accuracy. Solution: The divided difference coefficients are a0 = 17.1880, a1 = 0.4558, a2 = 0.0042 and the plot of the interpolating polynomial is given in Figure 4.3. This is very good agreement. 70 60 50 40 30 20 10 220 230 Figure 4.3 240 250 260 270 280 290 300 Solution plot for Exercise 4.2.11a. (b) Repeat the above, but this time construct the quartic interpolating polynomial using the nodes T0 = 220, T1 = 240, T2 = 260, T3 = 280, and T4 = 300. 5 Taken from tables in Introduction to Thermodynamics: Classical and Statistical, by Sonntag and Van Wylen, John Wiley & Sons, New York, 1971. NEWTON INTERPOLATION AND DIVIDED DIFFERENCES 129 Solution: The divided difference coefficients are a0 a3 = 1.7188e + 1, a1 = 3.8910e − 1, a2 = 3.3350e − 3, = 1.3375e − 5, a4 = 2.2917e − 8 and the plot of the interpolating polynomial is given in Figure 4.4. 70 60 50 40 30 20 10 220 230 Figure 4.4 240 250 260 270 280 290 300 Solution plot for Exercise 4.2.11b. (c) Which of the two polynomials would you think is best to use to get values for P (T) that are not on the table? Solution: Both plots look very good, actually. 12. Similar data for gaseous ammonia is given in Table 4.6 (Table 4.11 in the text). Table 4.6 Temperature-pressure values for gaseous ammonia; Problem 12 T P 0 30.42 5 34.27 10 38.51 15 43.14 20 48.21 25 53.73 30 59.74 35 66.26 40 73.32 (a) Construct the quadratic interpolating polynomial to this data at the nodes T0 = 0, T1 = 20, and T2 = 40. Plot the polynomial and the data in the table and comment on the observed accuracy. Solution: The divided difference coefficients are a0 = 3.0420e + 1, a1 = 8.8950e − 1, a2 = 9.1500e − 3. (b) Repeat the above, but this time construct the quartic interpolating polynomial using the nodes T0 = 0, T1 = 10, T2 = 20, T3 = 30, and T4 = 40. 130 INTERPOLATION AND APPROXIMATION Solution: The divided difference coefficients are a0 = a4 = 3.0420e + 1, a1 = 8.0900e − 1, a2 = 8.0500e − 3, a3 = 3.6667e − 5, 1.7279e − 20 (c) Which of the two polynomials would you think is best to use to get values for P (T) that are not on the table? 13. In Problem 8 of 3.1 and Problem 8 of 3.8 we looked at the motion of a liquid/solid interface under a simplified model of the physics involved, in which the interface moved according to √ x = 2β t √ for β = α/ k. Here k is a material property and α is the root of f (z) = 2 θe−z − zerf(z), where θ also depends on material properties. Figure 4.5 shows a plot of α vs. log10 θ, based on finding the root of f (z). Some of the data used to create this curve is given in Table 4.7 (Table 4.12 in the text). 3 2.5 α 2 1.5 1 0.5 0 −4 −3 −2 Figure 4.5 −1 0 log10θ 1 2 3 4 Figure for Problem 13. (a) Use the data at the nodes {−6, −4, −2, 0, 2, 4, 6} to construct an interpolating polynomial for this table. Plot the polynomial and compare to the actual graph in Figure 4.5. Use the polynomial to compute values at log10 θ = −5, −3, . . . , 3, 5. How do these compare to the actual values from the table? NEWTON INTERPOLATION AND DIVIDED DIFFERENCES 131 Table 4.7 Data for Problem 13. log10 θ -6.0000 -5.0000 -4.0000 -3.0000 -2.0000 -1.0000 0.0000 1.0000 2.0000 3.0000 4.0000 5.0000 6.0000 α 0.944138E-03 0.298500E-02 0.941277E-02 0.297451E-01 0.938511E-01 0.289450E+00 0.767736E+00 0.141492E+01 0.198151E+01 0.245183E+01 0.285669E+01 0.321632E+01 0.354269E+01 Solution: The divided difference coefficients are a0 = 9.4414e−4, a1 = 4.2343e−3, a4 = −1.4662e − 3, a2 = 9.4962e−3, a5 = −6.9243e − 5, a3 = 1.0697e−2, a6 = 4.6066e − 5 We get p6 (x) = .0001600x6 + .00066298x5 − .007638x4 − .03204x3 + .09548x2 + .5895x + .7677, for x = log10 θ. (b) Compute the higher-degree Newton polynomial based on the entire table of data. Plot this polynomial and compare it to the one generated using only part of the data. Solution: Figure 4.6 shows the nodes and both interpolating polynomials. Note that the interpolations have much more "wiggle" to them than is present in the raw data. 14. Write a computer program to construct the Newton interpolating polynomial √ to f (x) = x using equally spaced nodes on the interval [0, 1]. Plot the error f (x) − pn (x) for n = 4, 8, 16 and comment on what you get. Solution: Figure 4.7 shows the three error plots. Note the large "spike" in the error near x = 0. Section 4.3 will explain why this occurs. /•••. 132 INTERPOLATION AND APPROXIMATION 4 3.5 3 2.5 2 1.5 1 0.5 0 −0.5 −6 −4 Figure 4.6 −2 0 2 4 6 Solution plot for Exercise 4.2.13. 0.1 0.06 0.04 0.05 0.02 0 −0.05 0 0 0.5 1 0.5 1 −0.02 0 0.5 1 0.04 0.03 0.02 0.01 0 −0.01 0 Figure 4.7 4.3 Solution plot for Exercise 4.2.14. INTERPOLATION ERROR Exercises: 1. What is the error in quadratic interpolation to f (x) = nodes on the interval [14 , 1]? √ x, using equally spaced INTERPOLATION ERROR 133 Solution: We have |f (x) − p2 (x)| ≤ = 1 √ (3/8)3 max |(3/8)t−5/2 | t∈[1/4,1] 9 3 1 81 √ × × 45/2 = .4059494081e − 1. 9 3 64 3 4096 2. Repeat the above for f (x) = x−1 on [21 , 1]. Solution: We have |f (x) − p2 (x)| ≤ = 1 √ (1/4)3 max |6x−4 | t∈[1/2,1] 9 3 1 1 √ × × 6 × 16 = .9622504490e − 1. 9 3 64 3. Repeat the previous two problems, using cubic interpolation. Solution: This time we have √ 1 (1/4)4 max |(15/16)x−7/2 | | x − p3 (x)| ≤ 24 t∈[1/4,1] 1 15 1 × × 47/2 = .1953125e − 1 = 24 256 16 and |x−1 −p3 (x)| ≤ 1 1 1 (1/6)4 max |24x−5 | = × ×24×32 = .2469135802e−1. 24 24 362 t∈[1/2,1] 4. Show that the error in third degree polynomial interpolation satisfies kf − p3 k∞ ≤ 1 4 (4) h kf k∞ , 24 if the nodes x0 , x1 , x2 , x3 are equally spaced, with xi − xi−1 = h. Hint: Use the change of variable t = x − x1 − 12 h. Solution: We have f (x) − p4 (x) = w4 (x) (4) f (ξx), 24 for some value ξx in the interval defined by the nodes and x. Therefore kf − p3 k∞ ≤ 1 max |w4 (x)|kf (4) k∞ , 24 x∈[x0 ,x3] where we have assumed the nodes are ordered x0 < x1 < x2 < x3 . The change of variable reduces W = max |w4 (x)| x∈[x0 ,x3] 134 INTERPOLATION AND APPROXIMATION to W = max |(t2 − 9h2 /4)(t2 − h2 /4)| t∈[−3h/2,3h/2] and ordinary calculus shows that W = h4 . 5. Show that the error in polynomial interpolation using six equally spaced points (quintic interpolation) satisfies kf − p5 k∞ ≤ Ch6 kf (6) k∞ where C ≈ 0.0235. Hint: See previous problem. Solution: The basic outline from the previous exercise shows us that kf − p5 k∞ ≤ 1 max 720 t∈[−5h/2,5h/2] |(t2 − 25h2 /4)(t2 − 9h2 /4)(t2 − h2 /4)|kf (6) k∞ . Applying ordinary calculus to the problem of finding W = |(t2 − 25h2 /4)(t2 − 9h2 /4)(t2 − h2 /4)| max t∈[−5h/2,5h/2] shows that W = 16.90089433h6 , thus C = 16.90089433/720 = 0.0235. 6. Generalize the derivation of the error bound (4.12) for quadratic interpolation to the case where the nodes are not equally spaced. Take x1 − x0 = h and x2 − x1 = θh for some θ > 0. Solution: The issue is obtaining an upper bound for |w2 (x)| = |(x − x0)(x − x1)|. Ordinary calculus, coupled with the same kind of variable shift as used in the text, can be used to show that |w2 (x)| ≤ (h3 /27)g(θ), where g(θ) = p p − θ − 2 + θ2 − θ + 1 θ + 1 + θ2 − θ + 1 p × 2 θ − 1 − θ2 − θ + 1 for θ ∈ [0, 1/2], and g(θ) = p p − θ − 2 − θ2 − θ + 1 θ + 1 − θ2 − θ + 1 p × 2 θ − 1 + θ2 − θ + 1 INTERPOLATION ERROR 135 for θ ∈ [1/2, 1]. This looks messy, but the graph of g shows a simple V-shaped curve, ranging from 4 at θ = 0 to slightly more than 1 at θ = 1/2, and then back up as θ continues to increase. 7. Apply your√result for the previous problem to the error in quadratic interpolation 9 to f (x) = x using the nodes x0 = 41 , x1 = 16 , and x2 = 1. Solution: We have that h = x1 − x0 = 5/16, and θ = 7/5. Thus the error is bounded according to √ | x − p2 (x)| ≤ (h3 /27)g(θ) max |(−1/4)t−3/2 | t∈[0,1] = 3 (5/16) × (1/27) × 6.488878750 × (1/4) × 43/2 = .1466850846e − 1. 8. If we want to use a table of exponential values to interpolate the exponential function on the interval [−1, 1], how many points are needed to guarantee 10−6 accuracy with linear interpolation? Quadratic interpolation? Solution: We have, for the linear case, kf − pn k∞ ≤ 1 2 00 e h kf k∞ = h2 . 8 8 Therefore we get 10−6 accuracy when h ≤ .1715527770e − 2, which implies n ≥ 1166 points on the interval [−1, 1]. For the quadratic case, we have e 1 kf − pn k∞ ≤ √ h3 kf 000 k∞ = √ h3 . 9 3 9 3 Therefore we get 10−6 accuracy when h ≤ .1789930706e − 1, which implies n ≥ 112 points. 9. If we want to use a table of values to interpolate the error function on the interval [0, 5], how many points are needed to get 10−6 accuracy using linear interpolation? Quadratic interpolation? Would it make sense to use one grid spacing on, say, [0, 1], and another one on [1, 5]? Explain. Solution: For the linear case, we have h2 1 kf − pn k∞ ≤ h2 kf 00 k∞ = √ (1). 8 2 pi Therefore we get 10−6 accuracy when h ≤ .1882792528e − 2, which implies n ≥ 2656 points on the interval [0, 5] 136 INTERPOLATION AND APPROXIMATION For the quadratic case, we have h3 2.256758334h3 √ kf − pn k∞ ≤ √ kf 000 k∞ = . 9 3 9 3 Therefore we get 10−6 accuracy when h ≤ .1904463677e − 1, which implies n ≥ 263 points on the interval [0, 5] Breaking the table up into two regions [0, 1] and [1, 5] makes sense because the error function is nearly flat for large arguments, and so we can get away with a much larger h on [1, 5] and probably many fewer overall points. 10. If we want to use a table of values to interpolate the sine function on the interval [0, π], how many points are needed for 10−6 accuracy with linear interpolation? Quadratic interpolation? Cubic interpolation? Solution: We get n ≥ 1111 for the linear case, n ≥ 144 for the quadratic case, n ≥ 45 for the cubic case. 11. Let's return to the computation of the natural logarithm. Consider a computer which stores numbers in the form z = f · 2β , where 12 ≤ f ≤ 1. We want to consider using this, in conjunction with interpolation ideas, to compute the natural logarithm function. (a) Using piecewise linear interpolation over a grid of equally spaced points, how many table entries would be required to accurately approximate ln z to within 10−14 ? (b) Repeat the above, using piecewise quadratic interpolation. (c) Repeat it again, using piecewise cubic interpolation. Explain, in a brief essay, the importance here of restricting the domain of z to the interval [21 , 1]. Solution: For the linear case, we have | ln x − p1 (x)| ≤ (1/8)h2 max | − 1/t2 | = h2 /2, t∈[1/2,1] so we need h ≤ .1414213562e − 6 and hence n ≥ 3535533.907 points. For the quadratic case, we have √ √ | ln x − p2 (x)| ≤ (1/(9 3))h3 max |2/t3 | = 16h3 /(9 3) t∈[1/2,1] so we need h ≤ .2135802307e − 4 and hence n ≥ 23410.40640 points. Finally, for the cubic case, we have | ln x − p3 (x)| ≤ (1/24)h4 max | − 6/t4 | = 96h4 /24 = 4h4 t∈[1/2,1] INTERPOLATION ERROR 137 so we need h ≤ .2236067977e − 3 and hence n ≥ 2236.067978 points. 12. Assume that, for any real c, cn = 0. n→∞ n! lim Use this to prove that, if pn is the polynomial interpolate to f (x) = sin x on the interval [a, b], using any distribution of distinct nodes xk , 0 ≤ k ≤ n, then kf − pn k∞ → 0 as n → ∞. Hint: Can we justify |wn (x)| ≤ cn+1 for some c? Solution: We have kf − pn k∞ ≤ but kwn k∞ = max x∈[a,b] kwn k∞ kwn k∞ (n+1) kf k∞ ≤ , (n + 1)! (n + 1)! n Y |x − xn | ≤ k=0 n Y |b − a| = (b − a)n+1 . k=0 Therefore, (b − a)n+1 → 0, (n + 1)! kf − pn k∞ ≤ according to the assumed hypothesis. 13. Can you repeat the above for f (x) = ex ? Why/why not? Solution: A similar argument will work for the exponential function, since the derivative term is again bounded by a constant. 14. Define the norms kf k∞ = max |f (x)| x∈[0,1] and Z kf k2 = 1 [f (x)]2 dx 1/2 . 0 Compute kf k∞ and kf k2 for the following list of functions: (a) sin nπx, n integer; (b) e−ax , a > 0; √ 2 (c) xe−ax , a > 0; √ Solution: If a > 1/4 then kf k∞p= (2/2)e−1/4 a−1/4 ; if a ≤ 1/4, then kf k∞ = e−1 . kf k2 = (1/2) (1 − e−2a)/a. √ (d) 1/ 1 + x. 138 INTERPOLATION AND APPROXIMATION Solution: kf k∞ = 1, kf k2 = .9101797207. 15. Define fn (x) = 1 − nx, 0 ≤ x ≤ n1 ; 1 0 n ≤ x ≤ 1. Show that lim kfn (x)k∞ = 1 n→∞ but lim kfn (x)k2 = 0. n→∞ Hint: Draw a graph of fn (x). Solution: This can be done by direct computation but a graph of fn is revealing. It shows that max |fn | = 1 for all n, but the support of |fn | goes to zero, thus forcing the integral norm to vanish while the pointwise norm remains at 1. 16. Show that Z b |f (x)|dx kf k1,[a,b] = a defines a function norm. Solution: We have, for any constant c, b Z Z b |f (x)|dx = |c|kf k1,[a,b] , |cf (x)|dx = |c| kcf k1,[a,b] = a a and Z kf + gk1,[a,b] b Z a a Z b |g(x)|dx |f (x)|dx + |f (x) + g(x)|dx ≤ = b a = kf k1,[a,b] + kgk1,[a,b] . This establishes two of the three conditions. To establish the remaining condition requires an argument to the effect that the integral of the absolute value of a non-zero function must be greater than zero. Doing this precisely is probably beyond the analysis skills of most students, but good students ought to be able to construct a reasonable argument. /•••. 139 APPLICATION: MULLER'S METHOD AND INVERSE QUADRATIC INTERPOLATION 4.4 APPLICATION: MULLER'S METHOD AND INVERSE QUADRATIC INTERPOLATION Exercises: 1. Do three steps of Muller's method for f (x) = 2 − ex , using the initial points 0, 1/2, 1. Make sure you reproduce the values in the table in the text. 2. Repeat the above for inverse quadratic interpolation. Solution: We get x3 = .7087486790, x4 = .692849951, x5 = .692849951. 3. Let f (x) = x6 − x − 1; show that this has a root on the interval [0, 2] and do three steps of Muller's method using the ends of the interval plus the midpoint as the initial values. Solution: f (0) = −1 and f (2) = 61, so a root exists on [0, 2]. We get x3 = 1.0313, x4 = 1.1032, x5 = 1.1258. 4. Repeat the above using inverse quadratic interpolation. Solution: We get x3 = 1.017357687, x4 = 1.032876088, x5 = 1.174008963. 5. Let f (x) = ex and consider the nodes x0 = −1, x1 = 0, and x2 = 1. Let p2 be the quadratic polynomial that interpolates f at these nodes, and let q2 be the quadratic polynomial (in y) that inverse interpolates f at these nodes. Construct p2 (x) and q2 (y) and plot both, together with f (x). Solution: p2 (x) = 0.543080635 x2 + 1.175201193 x + 1.0 q2 (y) = −0.4254590638 y 2 + 2.163953414 y − 1.738494350 6. Repeat the above using f (x) = sin πx and the nodes x0 = 0, x1 = x2 = 21 . 1 4, and Solution: We get p2 (x) = −3.31370850x2 +3.656854248x, q2 (x) = y 2 /2. 7. Show that the formulas (4.15) and (4.16) for the divided-difference coefficients in Muller's Method (4.14) are correct. Solution: Direct computation. 8. Show that the formulas (4.17) and (4.18) for the coefficients in inverse quadratic interpolation are correct. Solution: Direct computation. 140 INTERPOLATION AND APPROXIMATION 9. Apply Muller's method and inverse quadratic interpolation to the functions in Exercise 3 of 3.1, and compare your results to those obtained with Newton's method and/or the secant method. Solution: See Table 4.8 for the Muller's method results. Table 4.8 Solutions to Exercise 4.4.9 n 1 2 3 4 5 3 f (x) = x − 2x + 5 2.00000000000000 2.09020134845606 2.09451837842307 2.09455148276226 2.09455148154233 f (x) = ex − 2 0.68725936775341 0.69308784769077 0.69314716126932 0.69314718055995 0.69314718055995 f (x) = x2 − sin x 0.87590056539793 0.87670226627220 0.87672621454247 0.87672621539506 0.87672621539506 10. Refer back to the discussion of the error estimate for the secant method in Section 3.10.3. Adapt this argument to derive an error estimate for Muller's method. Solution: Muller's method is based on finding the exact root of a quadratic polynomial which interpolates the original function f . Let m be this interpolating polynomial, so that we have m(x) = f (xn) + a1 (x − xn) + a2 (x − xn)(x − xn−1) where a1 and a2 are divided difference coefficients and the next iterate is defined by solving 0 = m(xn+1). Then the quadratic interpolation error estimate implies that f (x) − m(x) = 1 (x − xn)(x − xn−1)(x − xn−2)f 000 (ξn) 6 for all x; here ξn is in the interval defined by xn , xn−1 ,xn−2 , and α. Set x = α so that we have f (α) − m(α) = 1 (α − xn)(α − xn−1)(α − xn−2)f 000 (ξn) 6 But f (α) = 0 so we have −m(α) = 1 (α − xn)(α − xn−1)(α − xn−2)f 000 (ξn) 6 On the other hand we know that m(xn+1) = 0, so we can substitute this in to get m(xn+1) − m(α) = 1 (α − xn)(α − xn−1)(α − xn−2)f 000 (ξn). 6 APPLICATION: MORE APPROXIMATIONS TO THE DERIVATIVE 141 The Mean Value Theorem then implies that there exists a value ηn between α and xn+1 such that m(xn+1) − m(α) = m0 (ηn)(xn+1 − α); therefore, m0 (ηn)(xn+1 − α) = or xn+1 − α = 1 (α − xn)(α − xn−1)(α − xn−2)f 000 (ξn) 6 1 f 000 (ξn) (α − xn)(α − xn−1)(α − xn−2) 0 . 6 m (ηn) To really tidy up the final result we need to relate m0 (ηn) to f 0 . /•••. 4.5 APPLICATION: MORE APPROXIMATIONS TO THE DERIVATIVE Exercises: 1. Apply the derivative approximations (4.21) and (4.23) to the approximation of f 0 (x) for f (x) = x3 + 1 for x = 1 and h = 18 . Solution: Using both of (4.21) and (4.23) we get f 0 (x) ≈ 2.968750000 2. Apply the derivative approximations (4.21) and (4.23) to the same set of functions as in Exercise 3 of Section 2.2, using a decreasing sequence of mesh values, h−1 = 2, 4, 8, ... Do we achieve the expected rate of accuracy as h decreases? 3. Derive a version of (4.24) under the assumption that x1 −x0 = h, but x2 −x1 = η = θh for some real, positive, θ. Be sure to include the error estimate as part of your work, and confirm that when θ = 1 you get the same results as in the text. Solution: f 0 (x0) = −θ(2 + θ)f (x0) + (1 + 2θ + θ2)f (x1) − f (x2) 1 + (1+θ)h2 f 000 (ξ) θ(1 + θ)h 6 4. Repeat the above for (4.25). Solution: f 0 (x1) = (−η/(h(h + η)))f (x0) − ((h − η)/(hη))f (x1) + (h/(η(h + η)))f (x2) − (hη/6)f 000 (ξ) 142 INTERPOLATION AND APPROXIMATION 5. Repeat the above for (4.26). Solution: f 0 (x2) = θ2 f (x0) − (1 + 2θ + θ2)f (x1) + (2θ + 1)f (x2) 1 + (1+θ)h2 f 000 (ξ) θ(1 + θ)h 6 6. Use the derivative approximations from this section to construct a table of values for the derivative of the gamma function, based on the data in Table 4.3 (Table 4.8 in the text). Solution: Γ0 (1) ≈ −.5638283100, Γ0 (1.5) ≈ .3125765900e − 1, Γ0 (2) ≈ .4216022150. 7. Try to extend the ideas of this section to construct an approximation to f 00 (xk). Is it possible? What happens? Solution: You can't eliminate all the terms that depend on the derivative with respect to x of ξx . /•••. 4.6 HERMITE INTERPOLATION Exercises: 1. Show that H2 , as defined in (4.28), is the cubic Hermite interpolate to f at the nodes x = a and x = b. Solution: Clearly H2 (a) = f (a) and H20 (a) = f 0 (a). In addition, H2 (b) = f (a) + f 0 (a)(b − a) + (A − f 0 (a))(b − a) = f (a) + A(b − a) = f (b), and H20 (b) = + = = = f 0 (a) + 2B(b − a) + D(b − a)2 = f 0 (a) + 2A − 2f 0 (a) (C − B)(b − a) f 0 (a) + 2A − 2f 0 (a) + [(f 0 (b) − A) − (A − f 0 (a))] f 0 (a) − 2f 0 (a) + f 0 (a) + 2A − 2A + f 0 (b) f 0 (b). H2 is a cubic polynomial and it reproduces the function and derivative values at the nodes. Therefore, by the uniqueness part of Theorem 4.4, H2 must be the Hermite interpolant. HERMITE INTERPOLATION 143 2. Construct the cubic Hermite interpolate to f (x) = sin x using the nodes a = 0 and b = π. Plot the error between the interpolate and the function. Solution: H2 (x) = −(1/π)x2 + x. √ 3. Construct the cubic Hermite interpolate to f (x) = 1 + x using the nodes a = 0 and b = 1. Plot the error between the interpolate and the function. Solution: H2 (x) = 1.0+0.5000000000 x−0.085786438 x2 +0.025126266 x2 (x − 1.0) The maximum absolute error over [0, 1] is about 0.0007. 4. Show that the error in cubic Hermite interpolation at the nodes x = a and x = b is given by kf − H2 k∞ ≤ (b − a)4 (4) kf k∞ . 384 Solution: This is a direct application of Theorem 4.5, using n = 2. √ 5. Construct the cubic Hermite interpolate to f (x) = x on the interval [41 , 1]. What is the maximum error on this interval, as predicted by theory? What is the maximum error that actually occurs (as determined by observation; no need to do a complete calculus max/min problem)? Solution: H2 (x) = 0.2037037037 + 1.388888889 x − 0.8888888889 x2 +0.2962962963 x3 The error theory predicts that the error is bounded according to √ | x − H2 (x)| (3/4)4 max |(15/16)x−7/2 | 384 x∈[1/4,1] 81 1 15 = × × × 47/2 256 384 16 = .9887695312e − 1. ≤ The actual error, looking at a computer plot, is about 0.0065. 6. Construct the cubic Hermite interpolate to f (x) = 1/x on the interval [12 , 1]. What is the maximum error as predicted by theory? What is the actual (observed) maximum error? 144 INTERPOLATION AND APPROXIMATION Solution: H2 (x) = 6 − 13x + 12x2 − 4x3 , the predicted error is ≤ 1/8, the observed error is ≤ 0.022. 7. Construct the cubic Hermite interpolate to f (x) = x1/3 on the interval [18 , 1]. What is the maximum error as predicted by theory? What is the actual (observed) maximum error? Solution: H2 (x) 0.3090379009 + 1.732750243 x − 1.725947522 x2 =+0.6841593780 x3 with maximum observed error about 0.036. 8. Construct the cubic Hermite interpolate to f (x) = ln x on the interval [12 , 1]. What is the maximum error as predicted by theory? What is the actual (observed) maximum error? Solution: H2 (x) =−2.227411278 + 4.364467670x − 3.046701500x2 + .9096451100x3 , the predicted error is ≤ 1/64, the observed error is ≤ 0.0037. 9. Extend the divided difference table for cubic Hermite interpolation to quintic Hermite interpolation, using three nodes x = a, x = b, and x = c. Solution: We get a table like Table 4.9. Here A, B, C, and D are as in the text, and α = γ = E = θ = α − f 0 (c) f (b) − f (c) β= b−c b−c f 0 (b) − α γ−β δ= b−c b−c C −γ D−E F = a−c a−c F −θ E−δ G= a−c a−c 10. Construct the quintic Hermite interpolate to f (x) = ln x on the interval [12 , 1]; use x = 3/4 as the third node. Solution: We get H2 (x) = −2.64516010 + 7.1200353 x + 9.160601 x3 − 9.971884 x2 −4.659891 x4 + 0.9962986 x5 . 11. What is the error in quintic Hermite interpolation? PIECEWISE POLYNOMIAL INTERPOLATION 145 Table 4.9 Table for divided differences for quintic Hermite interpolation. k 1 xk a f0 (xk) f f1 (xk) f2 (xk) f3 (xk) f4 (xk) f5 (xk) f 0 (a) 1 a f (a) B A 2 b D f (b) C F f 0 (b) 2 b f (b) 3 c f (c) E γ G θ α δ β f 0 (c) 3 c f (c) Solution: We get something like f (x) − H3 (x) = 1 ψ3 (x)f (6) (ξ) 720 where ψ3 (x) = (x − x1)2 (x − x2)2 (x − x3)2 . This can be bounded above according to kf − H3 k∞ ≤ Ch6 kf (6) k∞ , where C = .0002057613167, approximately. This value comes from using ordinary calculus to find the extreme values of ψ3 . 12. Extend the ideas of Section 4.5 to allow us to compute second derivative approximations using Hermite interpolation. Solution: Because ψn (x) involves terms of the form (x−xk)2 for each node, the basic idea from Section 4.5 can be used to get derivative approximation formulas for the second derivative, based on Hermite interpolation. /•••. 4.7 PIECEWISE POLYNOMIAL INTERPOLATION Exercises: 1. Use divided difference tables to construct the separate parts of the piecewise quadratic polynomial q2 (x) that interpolates to f (x) = cos 12 πx at x = 0, 41 , 12 , 34 , 1. Plot the approximation and the error sin 21 πx − q2 (x). 146 INTERPOLATION AND APPROXIMATION 2. Repeat the above using f (x) = √ x with the nodes x = 15 , 25 , 35 , 45 , 1. Solution: This requires that we construct the two divided difference tables that are required here. We have the results in Table 1, thus the piecewise polynomial is given by 0.4472 + 0.9262(x − 0.2) − 0.5388(x − 0.2)(x − 0.4) 0.2 ≤ x ≤ 0.6 q2 (x) = 0.7746 + 0.5992(x − 0.6) − 0.1782(x − 0.6)(x − 0.8) 0.6 ≤ x ≤ 1. Table 4.10 Divided difference table. k 0 xk 0.2 f0 (xk) 0.4472 1 0.4 0.6325 f1 (x) f2 (x) 0.9262 -0.5388 0.7107 2 0.6 0.7746 k 0 xk 0.6 f0 (xk) 0.7746 1 0.8 0.8944 f1 (x) f2 (x) 0.5992 -0.1782 0.5279 2 1.0 1.0000 3. Confirm that (4.30) is the correct piecewise quadratic approximation to f (x) = 1/(1 + 25x2) using the nodes x0 = −1, x1 = −2/3, x2 = −1/3, x3 = 0, x4 = 1/3, x5 = 2/3, and x6 = 1. Solution: Since the each polynomial piece matches the nodal data it must be correct, because of the uniqueness of polynomial interpolation. 4. Using the data in Table 4.3, construct a piecewise cubic interpolating polynomial to the gamma function, using the nodes x = 1.0, 1.2, 1.3, 1.5 for one piece, and the nodes x = 1.5, 1.7, 1.8, 2.0 for the other piece. Use this approximation to estimate Γ(x) for x = 1.1, 1.4, 1.6 and 1.9. How accurate is the approximation? Solution: q3 (x) = −.3427614666x3 + 1.873584557x2 − 3.283390574x + 2.752567484, for 1 ≤ x ≤ 3/2 = (0.1596133333e − 2)x3 + .3766571466x2 − 1.105518098x + 1.691638542, for 3/2 ≤ x ≤ 2 PIECEWISE POLYNOMIAL INTERPOLATION 147 5. Using the results of Exercise 6 from Section 4.6, together with the data of function values from Exercise 9 of Section 4.2, construct a piecewise cubic Hermite interpolating polynomial for the gamma function, using the nodes x = 1.0, 1.3, 1.7, 2.0. Test the accuracy of the interpolation by using it to approximate Γ(x) for x = 1.1, 1.2, 1.4, 1.5, 1.6, 1.8, 1.9. Solution: We get q3 (x) = 2.663132928 − 3.038661703x + 1.651752931x2 − .2762241567x3 for 1 ≤ x ≤ 1.3 = 2.072404716 − 1.800684259x + .7983854147x2 − (0.8343915950e − 1)x3 for 1.3 ≤ x ≤ 1.7 = 1.582234466 − .9241759694x + .2766990121x2 + (0.1991517800e − 1)x3 for 1.7 ≤ x ≤ 2 Moreover, we have Γ(1.1) − q3 (1.1) = −.2209795e − 3 Γ(1.2) − q3 (1.2) = .2209792e − 3 Γ(1.4) − q3 (1.4) = −.612948e − 4 Γ(1.5) − q3 (1.5) = .885788e − 4 Γ(1.6) − q3 (1.6) = .1055826e − 3 Γ(1.8) − q3 (1.8) = .159327e − 4 Γ(1.9) − q3 (1.9) = −.159317e − 4 6. Construct a piecewise cubic interpolating polynomial to f (x) = ln x on the interval [21 , 1], using the nodes xk = 1 k + , 2 18 0 ≤ k ≤ 9. Compute the value of the error ln x − p(x) at 500 equally spaced points on the interval [12 , 1], and plot the error. What is the maximum sampled error? Solution: This is a fairly simple program to write, building upon previous modules. The maximum error which I got was 2.282428148858706e − 5. 7. Repeat the above, using piecewise cubic Hermite interpolation over the same grid. Solution: The maximum error is about 1.9254e − 6. 8. Construct piecewise polynomial approximations of degree d = 1, 2, 3 to the data in Table 4.7, using only the nodes log10 θk = −6, −4, −2, 0, 2, 4, 6. Plot 148 INTERPOLATION AND APPROXIMATION the resulting curve and compare it to the ordinary interpolating polynomial found in Problem 13 of 4.2. Test how well this approximation matches the tabulated values at log10 θ = −5, −3, −1, 1, 3, 5. 9. Show that the error in piecewise cubic Hermite interpolation satisfies kf − H2 k∞ ≤ 1 4 (4) h kf k∞ 384 where we have assumed uniformly spaced nodes with xk − xk−1 = h. 10. Given a grid of points a = x0 < x1 < x2 < · · · < xn = b define the piecewise linear functions φhk , 1 ≤ k ≤ n − 1, according to φhk (x) = ⎧ ⎪ ⎨ x−xk−1 xk −xk−1 xk+1 −x xk+1 −xk xk−1 ≤ x ≤ xk ; xk ≤ x ≤ xk+1 ; 0 otherwise. ⎪ ⎩ Define the function space S0h = {f ∈ C([a, b]), f (a) = f (b) = 0, f is piecewise linear on the given grid}. Show that the φhk are a basis for S0h , i.e., that every element of the space S0h can be written as a linear combination of the φhk functions. 11. Implement a routine for approximating the natural logarithm using piecewise polynomial interpolation, i.e., a table look-up scheme. Assume the table of (known) logarithm values is uniformly distributed on the interval [21 , 1]. Choose enough points in the table to guarantee 10−10 accuracy for any x. Use (a) Piecewise linear interpolation; (b) Piecewise cubic Hermite interpolation. Test your routine against the intrinsic logarithm function on your computer by evaluating the error in your approximation at 5, 000 equally spaced points 1 on the interval [10 , 10]. Use the way that the computer stores floating point numbers to reduce the logarithm computation to the interval [21 , 1], so long as ln 2 is known to high accuracy. /•••. AN INTRODUCTION TO SPLINES 4.8 149 AN INTRODUCTION TO SPLINES Exercises: 1. Given the set of nodes x0 = 0, x1 = 1/2, x2 = 1, x3 = 3/2, and x4 = 2, we construct the cubic spline function q3 (x) = 0.15B−1 (x) + 0.17B0 (x) + 0.18B1 (x) + 0.22B2 (x) + 0.30B3 (x) + 0.31B4 (x) + 0.32B5 (x), where each Bk is computed from the exemplar B spline according to (4.39). Compute q3 and its first derivative at each node. Solution: We have q3 (x0) = 0.15B−1 (x0)+0.17B0 (x0)+0.18B1 (x0) = 0.15+0.17×4+0.18 = 1.01 and q30 (x0) = 0.15B−1 (x0) + 0.17B00 (x0) + 0.18B10 (x0) = 0.15(−3/h) + 0.17(0) + 0.18(3/h) = 0.15(−6) + 0.18(6) = 0.18 The others follow similarly. 2. Is the function ⎧ ⎪ 0, x ≤ 0; ⎪ ⎪ ⎪ 2 ⎪ 0 ≤ x ≤ 1; ⎨ x , p(x) = −2x2 + 6x + 3, 1 ≤ x ≤ 2; ⎪ ⎪ ⎪ (x − 3)2 , 2 ≤ x ≤ 3; ⎪ ⎪ ⎩ 0, x≥3 a spline function? Why/why not? Solution: No, it is not, because it is not continuous at x = 1. 3. For what value of k is the following a spline function? kx2 + (3/2), 0 ≤ x ≤ 1; q(x) = x2 + x + (1/2), 1 ≤ x ≤ 2. Solution: There is no k that will make this a spline. Taking k = 1 makes the function continuous, but the derivative is not continuous. 4. For what value of k is the following a spline function? 3 x − x2 + kx + 1 0 ≤ x ≤ 1; q(x) = −x3 + (k + 2)x2 − kx + 3 1 ≤ x ≤ 2. 150 INTERPOLATION AND APPROXIMATION Solution: k = 3. 5. For what value of k is the following a spline function? 3 x + 3x2 + 1, −1 ≤ x ≤ 0; q(x) = −x3 + kx2 + 1, 0 ≤ x ≤ 1. Solution: k = 3. 6. Construct the natural cubic spline that interpolates to f (x) = 1/x at the nodes 1/2, 5/8, 3/4, 7/8, 1. Do this as a hand calculation. Plot your spline function and f on the same set of axes, and also plot the error. Solution: The coefficients are c = [0.4052, 0.3333, 0.2615, 0.2207, 0.1889, 0.1667, 0.1445] 7. Repeat the above using the complete spline. Solution: The coefficients are c = [0.4180, 0.3299, 0.2624, 0.2204, 0.1893, 0.1653, 0.1496] 8. Construct a natural spline interpolate to the mercury thermal conductivity data (Table 4.2, Table 4.7 in the text), using the 300◦ K, 500◦ K, and 700◦ K values. How well does this predict the values at 400◦ K and 600◦ K? Solution: The spline coefficients are c−1 = 0.0095, c0 = 0.0140, c1 = 0.0185, c2 = 0.0212, c3 = 0.0239; the spline predicts that the conductivity at 400◦ K = 0.0972 and at 600◦ K = 0.1157. Both values are accurate, but not exact. 9. Confirm that the function B(x), defined in (4.35), is a cubic spline. Solution: It obviously is piecewise cubic; to confirm that it is a spline simply show that the one-sided limits at the interior knots are equal. 10. Construct a natural cubic spline to the gamma function, using the data in Table 4.3, and the nodes x = 1.0, 1.2, 1.4, 1.6, 1.8 and 2.0. Use this approximation to estimate Γ(x) at x = 1.1, 1.3, 1.5, 1.7, and 1.9. Solution: Since this is a natural spline construction, we have the following system of equations to solve. ⎤⎡ ⎤⎡ c0 0.7515020757 4 1 0 0 ⎥ ⎢ 1 4 1 0 ⎥ ⎢ c1 ⎥ ⎢ 0.8872638175 ⎥ ⎢ ⎥ ⎢ ⎥ ⎢ ⎣ 0 1 4 1 ⎦ ⎣ c2 ⎦ = ⎣ 0.8935153493 ⎦ 0 0 1 4 c3 0.7647171043 AN INTRODUCTION TO SPLINES 151 Thus ⎤ ⎡ c0 0.1511220252 ⎥ ⎢ c1 ⎥ ⎢ 0.1470139748 ⎥ ⎢ ⎥ ⎢ ⎣ c2 ⎦ = ⎣ 0.1480858929 0.1541578028 c3 ⎡ ⎤ ⎥ ⎥ ⎦ The natural spline construction then tells us the rest of the coefficients: c0 = Γ(x0)/6 = 1/6, c5 = Γ(x5)/6 = 1/6; and c−1 = 2c0 −c1 = 0.18221130811994, c6 = 2c5 −c4 = 0.17917553048357. Thus the spline is q(x) = 0.1822113081B−1 (x) + 0.1666666667B0 (x) +0.1511220252B1 (x) + 0.1470139748B2 (x) +0.1480858929B3 (x) + 0.1541578028B4 (x) = 0.1666666667B5 (x) + 0.1791755304B6 (x). We can test this by computing tk = ck−1 + 4ck + ck+1 ; we should get tk = f (xk), and we do. Just for example, t2 = c1 + 4c2 + c3 = 0.1511220252 + 0.1470139748 × 4 + 0.1480858929 = 0.8872638173, which matches Γ(x2) to nine digits, and probably would match the tenth digit if I had been more careful with my rounding here. Similar results hold at the other points. We can use this to approximate Γ at the indicated points by computing as follows. q(1.1) = 0.1822113081B−1 (1.1) + 0.1666666667B0 (1.1) +0.1511220252B1 (1.1) + 0.1470139748B2 (1.1), q(1.3) = 0.1666666667B0 (1.3) + 0.1511220252B1 (1.3) +0.1470139748B2 (1.3) + 0.1480858929B3 (1.3) and so on for the other points. 11. Repeat the above using the complete spline approximation, and use the derivative approximations from Section 4.5 for the required derivative endpoint values. 152 INTERPOLATION AND APPROXIMATION Solution: The spline coefficients are c = [0.1872, 0.1653, 0.1515, 0.1469, 0.1481, 0.1544, 0.1658, 0.1824]. Moreover, q3 (1.1) = 0.95256719297015, which is a good approximation to Γ(1.1) = 0.95135076986687. 12. Repeat Problem 6 of 4.7, except this time construct the complete cubic spline interpolate to f (x) = ln x, using the same set of nodes. Plot the approximation and the error. What is the maximum sampled error in this case? Solution: The spline coefficients are c = [−0.1345, −0.1153, −0.0977, −0.0819, −0.0674, −0.0541, −0.0417, −0.0303, −0.0195, −0.0094, 0.0001, 0.0090] 13. Recall Problem 7 from 4.2, in which we constructed polynomial interpolates to timing data from the 1973 Kentucky Derby, won by the horse Secretariat. For simplicity, we repeat the data in Table 4.11. Here t is the elapsed time (in Table 4.11 x t 0.0 0.0 0.25 25.0 Data for Problem 13. 0.50 49.4 0.75 73.0 1.00 96.4 1.25 119.4 seconds) since the race began and x is the distance (in miles) that Secretariat has travelled. (a) Construct a natural cubic spline that interpolates this data. (b) What is Secretariat's speed at each quarter-mile increment of the race? (Use miles per hour as your units.) (c) What is Secretariat's "initial speed," according to this model? Does this make sense? Note: It is possible to do this problem using a uniform grid. Construct the spline that interpolates t as a function of x, then use your knowledge of calculus to find x0 (t) from t0 (x). Solution: (a) Again, the fact that we are asked to do a natural spline tells us the linear system is 4⎢ 1⎢ ⎣ 0 0⎡ 1 4 1 0 0 1 4 1 ⎤ c1 0⎥ c2 0 ⎥⎢ 1 ⎦⎣ c3 4 c4 ⎤25.0 ⎥⎢ 49.4 ⎥ ⎢ ⎥=⎢ ⎦⎣ 73.0 ⎦76.5 ⎤⎡ AN INTRODUCTION TO SPLINES 153 Thus, the first four spline coefficients are ⎤⎡ c1 4.1842105263⎥ c2 ⎥⎢ 8.2631578947⎥ ⎥⎢ ⎣ c3 ⎦ = ⎣ 12.1631578947 16.0842105263 c4⎡ ⎤ ⎥ ⎥ ⎦ and the remaining coefficients are c0 = 0, c−1 = −4.1842105263, c5 = 19.9, c6 = 23.71578947368422 (b) Our spline, as constructed, gives us the time, t (in seconds) as a function of the distance, x (in miles). But velocity is the derivative of distance with respect to time. We use the calculus to tell us that dx = dt dt dx −1 . Thus we can compute q 0 (1.25) and from that get the horse's speed. We get q 0 (1.25) = (−3/h)(16.0842105263) + (3/h)(23.7157894736) = so dx = 0.0109195402. dt This seems a little low, but then we remember that the units are such that this is miles per second. Converting to miles per hour we get dx = 39.3103448279, dt which is a much more reasonable speed (c) The same kind of computation gives us an initial speed of 35.84905660390886. I suspect the horse took a few seconds to get up to full speed, so this is probably a bit high. 14. Show that the complete cubic spline problem can be solved (approximately) without having an explicit expression for f 0 at the endpoints. Hint: Consider the material from Section 4.5. Solution: Simply use the approximations f 0 (b) ≈ and f 0 (a) ≈ 3f (b) − 4f (b − h) + f (b − 2h) 2h −f (a + 2h) + 4f (a + h) − 3f (a) . 2h 154 INTERPOLATION AND APPROXIMATION 15. Construct the natural cubic spline that interpolates the data in Table 4.7 (Table 4.12 in the text) at the nodes defined by log10 θ = −6, −4, . . . , 4, 6. Test the accuracy of the approximation by computing q3 (x) for x = log10 θ = −5, −3, . . . , 3, 5 and comparing to the actual data in the table. Solution: The spline coefficients are c−1 = −0.0029, c0 = 0.0002, c1 = 0.0032, c2 = −0.0037, c3 = 0.1056, c4 = 0.3492, c5 = 0.4793, c6 = 0.5904, c7 = 0.7016. 16. Construct the exemplar quadratic B-spline; i.e., construct a piecewise quadratic function that is C 1 over the nodes/knots x = −1, 0, 1, 2, and which vanishes for x outside the interval [−1, 2]. Solution: A quadratic spline is a piecewise quadratic function with first derivative continuity at the knots; to be the exemplar B-spline we also need the local definition property, thus we need our function to be zero for x ≤ −1 and x ≥ 2. This leads us to try something like ⎧ ⎪ 0, x ≤ −1; ⎪ ⎪ ⎪ 2 ⎪ −1 ≤ x ≤ 0; ⎨ (x + 1) , B(x) = Ax2 + Bx + C; 0 ≤ x ≤ 1; ⎪ ⎪ 2 ⎪ (x − 2) ; 1 ≤ x ≤ 2; ⎪ ⎪ ⎩ 0, 2 ≤ x. We need to choose A, B, and C so that B ∈ C 1 , i.e., so that lim B(x) = lim+ B(x), x→0− x→0 lim B(x) = lim B(x), x→1− x→1+ 0 and similarly for B . This quickly leads, after a modest amount of manipulation, to the values C = 1, B = 2, A = −2. So the exemplar quadratic B-spline is ⎧ ⎪ 0, x ≤ −1; ⎪ ⎪ ⎪ 2 ⎪ (x + 1) , −1 ≤ x ≤ 0; ⎨ B(x) = −2x2 + 2x + 1; 0 ≤ x ≤ 1; ⎪ ⎪ ⎪ (x − 2)2 ; 1 ≤ x ≤ 2; ⎪ ⎪ ⎩ 0, 2 ≤ x. 17. Construct the exemplar quintic B-spline. Solution: This is lengthy but not hard. Based on the cubic construction, we are going to need to use nodes at x = −3, x = −2, x = −1, x = 0, x = 1, AN INTRODUCTION TO SPLINES 155 x = 2, and x = 3. In order to get the proper derivative continuity at the nodes x = ±3 we will need the exemplar quintic to have the basic form ⎧ 0, x < −3 ⎪ ⎪ ⎪ 5 ⎪ (x + 3) , −3 < x ≤ −2 ⎪ ⎪ ⎪ ⎪ ⎪ q1 (x), −2 < x ≤ −1 ⎪ ⎪ ⎨ q2 (x), −1 < x ≤ 0 B(x) = ⎪ q (x), 0 0 and the grid points xk are all known. Assume that In integrates the trivial function w(x) = 1 exactly: In (w) = n X k=1 Z b dx = b − a, ak = I(w) = a and that this holds for all intervals (a, b). Consider now the effects of rounding error on integrating an arbitrary function f . Let f^(x) = f (x) + (x) be f 186 NUMERICAL INTEGRATION polluted by rounding error, with |(x)| ≤ Cu for some constant C > 0, for all x ∈ [a, b]. Show that |In (f) − In (f^)| ≤ Cu(b − a). Comment on this in comparison to the corresponding result for numerical differentiation, as given in 2.2. Solution: The point is that numerical integration is much less affected by rounding error than is numerical differentiation. 19. The normal probability distribution is defined as p(x) = 2 2 1 √ e−(x−µ) /2σ σ 2π where µ is the mean, or average, and σ is the variance. This is the famous bell-shaped curve that one hears so much about; the mean gives the center of the bell and the variance gives the width of it. If x is distributed in this fashion, then the probability that a ≤ x ≤ b is given by the integral Z P (a ≤ x ≤ b) = b p(x)dx. a (a) Use the change of variable z = (x − µ)/σ to show that 1 P (−mσ ≤ x ≤ mσ) = √ 2π Z m e−z 2 /2 dz. −m (b) Compute values of P (−mσ ≤ x ≤ mσ) for m = 1, 2, 3, using Simpson's rule. Solution: For m = 1, P (−mσ ≤ x ≤ mσ) ≈ .6826908120; for m = 2, P (−mσ ≤ x ≤ mσ) ≈ .9544947255; and m = 3, P (−mσ ≤ x ≤ mσ) ≈ .9972830690. 20. Use Simpson's rule to solve Problem 9 from 5.2. 21. Use Simpson's rule to solve Problem 10 from 5.2. /•••. THE MIDPOINT RULE 5.4 187 THE MIDPOINT RULE Exercises: 1. Apply the midpoint rule with h = 41 , to approximate the integral Z 1 1 x(1 − x2)dx = . I= 4 0 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: We have, for f (x) = x(1 − x2), M4 (f) = (1/4)(f (1/8) + f (3/8) + f (5/8) + f (7/8)) = .2578125000. The error estimate is |I(f) − Mn (f)| ≤ h2 h2 max | − 6x| = , 24 x∈[0,1] 4 so we require h ≤ .6324555320e − 1 for 10−3 accuracy, and h ≤ 0.002 for 10−6 . 2. Apply the midpoint rule with h = 14 , to approximate the integral Z 1 1√ I= dx = 0.92703733865069. 1 + x4 0 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: M4 (f) = .9288993588, and h ≤ .1312709324 for 10−3 accuracy, h ≤ .4151151371e − 2 for 10−6 . 3. Apply the midpoint rule with h = 41 , to approximate the integral Z 1 ln(1 + x)dx = 2 ln 2 − 1. I= 0 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: M4 (f) = .3875883105, and h ≤ .1549193338 for 10−3 accuracy, h ≤ .4898979486e − 2 for 10−6 . 4. Apply the midpoint rule with h = 41 , to approximate the integral Z 1 1 1√ 1 dx = ln 2 + I= 3π. 3 3 9 0 1+x 188 NUMERICAL INTEGRATION How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: M4 (f) = .8376389970, and h ≤ .1175282713 for 10−3 accuracy, h ≤ .3716570267e − 2 for 10−6 . 5. Apply the midpoint rule with h = 41 , to approximate the integral Z 2 2 e−x dx = 0.1352572580. I= 1 How small does h have to be to get that the error is less than 10−3 ? 10−6 ? Solution: M4 (f) = .1335215673, and h ≤ .1095445115 for 10−3 accuracy, h ≤ .3464101615e − 2 for 10−6 . 6. Show that the midpoint rule can be derived by integrating, exactly, a polynomial interpolate of degree zero. Solution: Take p0 (x) = f (c) where c = (a + b)/2 is the midpoint of the interval. Then Z b p0 (x)dx = f (c)(b − a) = M1 (f). I(p0) = a 7. Apply the midpoint rule to each of the following functions, integrated over the indicated interval. Use a sequence of grids h = (b − a), (b − a)/2, (b − a)/4, . . . and confirm that the approximations are converging at the correct rate. Comment on any anomolies that you observe. (a) f (x) = ln x, [1, 3], I(f) = 3 ln 3 − 2 = 1.295836867; (b) f (x) = x2 e−x , [0, 2],I(f) = 2 − 10e−2 = 0.646647168; √ (c) f (x) = 1 − x2 , [−1, 1], I(f) = π/2; (d) f (x) = 1/(1 + x2), [−5, 5], I(f) = 2 arctan(5); (e) f (x) = e−x sin(4x), [0, π], I(f) = 4 17 (1 − e−π) = 0.2251261368. Solution: The output is in Table 5.3 ((a) - (d), only). The approximation √ performed as expected, with two notable exceptions: For f (x) = 1 − x2 , we did not achieve the expected rate of decrease for the error, because the second derivative of f is unbounded on [−1, 1]. For f (x) = x2 e−x , we achieved higher accuracy than expected. This occurs because f 0 (0) = f 0 (2) and there is an effect similar to the corrected trapezoid rule. 8. For each integral in the previous problem, how small does h have to be to get accuracy, according to the error theory, of at least 10−3 ? 10−6 ? THE MIDPOINT RULE 189 Table 5.3 Results for Exercise 5.4.7 n (d) (b) (a) (c) 2 1.37931034482759 0.65367552526213 1.32175583998232 1.73205080756888 4 2.28316971966400 0.64711384088491 1.30264523357222 1.62968366431800 8 2.70773386697212 0.64667678393337 1.29756401303372 1.59196461030595 16 2.74701026646563 0.64664902575751 1.29627032524886 1.57834346564915 32 2.74692176145878 0.64664728387795 1.29594533728055 1.57347590396326 64 2.74683162099652 0.64664717490087 1.29586399050904 1.57174570134112 128 2.74680905737720 0.64664716808809 1.29584364754886 1.57113233600677 256 2.74680341486880 0.64664716766226 1.29583856141662 1.57091518675892 512 2.74680200414141 0.64664716763565 1.29583728985904 1.57083836124104 1024 2.74680165145330 0.64664716763398 1.29583697196811 1.57081119017919 2048 2.74680156328087 0.64664716763388 1.29583689249528 1.57080158214194 Solution: For (b), we have 6x2 − 2 5h2 10h2 = |I(f) − Mn (f)| ≤ max , 24 x∈[−5,5] (1 + x2)3 6 therefore we need h ≤ .3464101615e − 1 to get 10−3 accuracy, and h ≤ .1095445115e − 2 for 10−6 accuracy. 9. State and prove a formal theorem concerning the error estimate for the midpoint rule over n subintervals. You may want to state and prove a formal theorem for the single subinterval rule first, and then use this in the more general theorem. Solution: The theorem would go something like this: Theorem: Let f ∈ C 2 ([a, b]); then there exists a point ξ ∈ [a, b] such that I(f) − Mn (f) = − b − a 2 00 h f (ξ). 24 The proof is almost identical to the developments used for Simpson's rule and the trapezoid rule. 10. Let T1 be the trapezoid rule using a single subinterval, M1 be the midpoint rule using a single subinterval, and S2 be Simpson's rule using a single quadratic interpolant (hence, a single pair of subintervals). Show that, for any continuous function f , and any interval [a, b], S2 (f) = 2 1 T1 (f) + M1 (f). 3 3 190 NUMERICAL INTEGRATION Solution: Let c denote the midpoint of the interval [a, b], so that we have T1 (f) = S2 (f) = 1 (b − a)(f (b) + f (a)) 2 (b − a)/2 (f (a) + 4f (c) + f (b)) 3 M1 (f) = (b − a)f (c). Then a direct computation shows 2 1 T1 (f) + M1 (f) 3 3 = = 1 2 (b − a)(f (b) + f (a)) + (b − a)f (c) 6 3 (b − a)/2 (f (a) + 4f (c) + f (b)) = S2 (f). 3 /•••. 5.5 APPLICATION: STIRLING'S FORMULA Exercises: 1. Use Stirling's formula to show that lim n→∞ xn n! =0 for all x. Solution: We have n   n x 1 |x|e |xn | ≤ √ √ = . n! Cn n(n/e)n Cn n n For n sufficiently large, the second factor will be less than one, which completes the proof. 2. For x = 10 and = 10−3 , how large does n have to be for n x ≤ n! to hold? Repeat for = 10−6 . Use Stirling's formula here, don't just plug numbers into a calculator. APPLICATION: STIRLING'S FORMULA 191 Solution: We have, for all n ≥ 1, 10n 1 10n en √ ≤ = n! Cn nnn 2 10e n n . For all n ≥ 30, we then have 10n 1 ≤ n! 2 10e 30 n , so that taking − ln 10−3 = 70.05 ln 3 − 1 ought to suffice. For 10−6 we need to have n = 140.1 These are both conservative estimates, because the ratio 10e/n is of course decreasing as n increases, and we took it as the fixed value r = 10e/30 ≈ 0.9061. Using a program like Maple we can find that the precise values are 32 and 38. n= 3. Use Stirling's formula to determine the value of (n!)p n→∞ (pn)! lim where p ≥ 2 is an integer. Solution: We have (n!)p (pn)! √ [Cn n(n/e)n]p √ Cpn pn(pn/e)pn √ p−1 ! n Cnp n 1 = √ √ Cpn p p pp ! √ p Cn np−1 . = √ Cpn p pn = It is a simple exercise with L'Hôpital's rule to show that the second factor goes to zero much faster than the first one goes to infinity. Thus the limit is zero. 4. Use Stirling's formula to get an upper bound on the ratio Rn = 1 · 3 · 5 · · · (2n − 1) 2n+1 n! as a function of n. Solution: The idea is to insert factors of 2 into the numerator and denominator so that the product of odd terms can be treated as a factorial: 1 · 3 · 5 · · · (2n − 1) 2n+1 n! = = 1 · 2 · 3 · 4 · 5 · · · (2n − 1) · (2n) [2 · 4 · 6 · · · (2n)]2n+1 n! (2n)! (2n)! = . 2n (n!)2n+1 n! 2(2n n!)2 192 NUMERICAL INTEGRATION Using Stirling's formula we get √ √ C2n 2n(2n/e)2n C2n 2n Cn n−1/2 Rn ≤ = 2 n−1/2 , 2 2n 2 2cn n(2n/e)2n 2cn where Ck ≤ 2.501 and ck ≥ 2.37, for all k. Thus we have Rn ≤ 0.2226 . n1/2 /•••. 5.6 GAUSSIAN QUADRATURE Exercises: 1. Apply Gaussian quadrature with n = 4 to approximate each of the following integrals. (a) Z 1 ln(1 + x2)dx = 2 ln 2 + π − 4. I= −1 Solution: G4 (f) = + + + = (0.3478548451374476) ln[1 + (−0.8611363115940526)2] (0.6521451548625464) ln[1 + (−0.3399810435848563)2] (0.6521451548625464) ln[1 + (0.3399810435848563)2] (0.3478548451374476) ln[1 + (0.8611363115940526)2] .5286293488 (b) Z 1 I= sin2 πxdx = 1. −1 Solution: G4 (f) = + + + = (0.3478548451374476) sin2 [π(−0.8611363115940526)] (0.6521451548625464) sin2 [π(−0.3399810435848563)] (0.6521451548625464) sin2 [π(0.3399810435848563)] (0.3478548451374476) sin2 [π(0.8611363115940526)] 1.125732289 193 GAUSSIAN QUADRATURE (c) Z 1 (x8 + 1)dx = 20/9. I= −1 Solution: G4 (f) = + + + = (0.3478548451374476)((−0.8611363115940526)8 + 1) (0.6521451548625464)((−0.3399810435848563)8 + 1) (0.6521451548625464) ((0.3399810435848563)8 + 1) (0.3478548451374476)((0.8611363115940526)8 + 1) 2.210612246 (d) Z 1 2 e−x dx = 1.493648266. I= −1 Solution: G4 (f) = (0.3478548451374476)e−(−0.8611363115940526) + (0.6521451548625464)e−(−0.3399810435848563) + (0.6521451548625464)e−(−0.3399810435848563) 2 2 2 2 + (0.3478548451374476)e−(0.8611363115940526) = 1.493334622 (e) Z 1 I= −1 1 dx = 1.733945974. 1 + x4 Solution: G4 (f) 1 1 + (−0.8611363115940526)4 1 + (0.6521451548625464) 1 + (−0.3399810435848563)4 1 + (0.6521451548625464) 1 + (0.3399810435848563)4 1 + (0.3478548451374476) 1 + (0.8611363115940526)4 = 1.735966736 = (0.3478548451374476) 2. Apply Gaussian quadrature with n = 4 to approximate each of the following integrals. Remember that you have to do a change of variable to [−1, 1] first. 194 NUMERICAL INTEGRATION (a) 1 Z ln(1 + x)dx = 2 ln 2 − 1. I= 0 (b) Z 1 1 √ dx = 0.92703733865069. 1 + x4 I= 0 Solution: G4 (f) = .9270388618 (c) 1 Z x(1 − x2)dx = I= 0 1 . 4 (d) 1 Z I= 0 1 1 1√ 3π. dx = ln 2 + 3 1+x 3 9 Solution: G4 (f) = .8356239692 (e) Z I= 2 2 e−x dx = 0.1352572580. 1 3. Show that (5.8) is both necessary and sufficient to make the quadrature exact for all polynomials of degree less than or equal to 2N − 1. Solution: Suppose (5.8) holds, and let q(x) be an arbitrary polynomial of degree ≤ 2N − 1. Then 2N −1 X ai xi q(x) = i=0 and Z 1 q(x)dx = 2N −1 X −1 i=0 Z 1 ai xi dx. −1 Since Gaussian quadrature is exact for this range of degree of polynomials, we then have ! 2N −1 n X X (n) (n) i I(q) = ai wk [xk] = Gn (q). i=0 k=1 The steps are entirely reversible, so we can easily prove the other direction of the implication. 4. Write a program that does Gaussian quadrature, using the weights and Gauss points given in the text. Apply this to each of the integrals below, and compare your results to those for the other quadrature methods in the exercises. GAUSSIAN QUADRATURE 195 Remember that you will have to do a change of variable if the interval is not [−1, 1]. Be sure to run your program for more than one value of n. √ (a) f (x) = 1 − x2 , [−1, 1], I(f) = π/2; (b) f (x) = x2 e−x , [0, 2],I(f) = 2 − 10e−2 = 0.646647168; Solution: G4 (f) = .6466426358 (c) f (x) = ln x, [1, 3], I(f) = 3 ln 3 − 2 = 1.295836867; (d) f (x) = 1/(1 + x2), [−5, 5], I(f) = 2 arctan(5); (e) f (x) = e−x sin(4x), [0, π], I(f) = 4 17 (1 − e−π) = 0.2251261368. 5. Let P (x) = 6x3 + 5x2 + x, and let P2 (x) = 3x2 − 1 (this is the quadratic Legendre polynomial). Find linear polynomials Q(x) and R(x) such that P (x) = P2 (x)Q(x) + R(x). Verify that I(P) = I(R). Solution: Simple division shows that Q(x) = 2x + (5/3) and R(x) = 3x + (5/3). It is then easily verified that (3x2 − 1)(2x + (5/3)) + (3x + (5/3)) = P (x). 6. Let P (x) = x3 + x2 + x − 1, and repeat the above. Solution: This time Q(x) = (1/3)(x + 1) and R(x) = (4/3)x − (2/3). 7. Let P (x) = 3x3 + x2 − 6, and repeat the above. Solution: This time Q(x) = x + (1/3) and R(x) = x − (17/3). 8. Verify that the weights for the n = 2 Gaussian quadrature rule satisfy the formula (5.9). Solution: This amounts to showing that Z 1 x − 0.5773502691896257 dx = 1 −1 −0.5773502691896257 − 0.5773502691896257 which it does, and similarly for the other Gauss point and weight. 9. Repeat the above for the n = 4 rule. Solution: This involves the same kind of computation, with the same result. 10. Show, by direct computation, that the n = 2 and n = 4 Gaussian quadrature rules are exact for the correct degree of polynomials. Solution: Direct computation does the trick. 11. The quadratic Legendre polynomial is P2 (x) = (3x2 − 1)/2. Show that it is orthogonal (over [−1, 1]) to all linear polynomials. 196 NUMERICAL INTEGRATION Solution: Let L(x) = Ax + B be an arbitrary linear polynomial; then we have Z 1 Z 1 1 P2 (x)L(x)dx = 3Ax3 + 3Bx2 − Ax − B dx = 0 2 −1 −1 for any A and B. 12. The cubic Legendre polynomial is P3 (x) = (5x3 − 3x)/2. Show that it is orthogonal (over [−1, 1]) to all polynomials of degree less than or equal to 2. Solution: Essentially the same proof as the above. 13. The quartic Legendre polynomial is P4 (x) = (35x4 − 30x2 + 3)/8. Show that it is orthogonal (over [−1, 1]) to all polynomials of degree less than or equal to 3. Solution: Essentially the same proof, again, only longer. 14. The first two Legendre polynomials are P0 (x) = 1, P1 (x) = x, and it can be shown that the others satisfy the recurrence relation (n + 1)Pn+1 (x) = (2n + 1)xPn (x) − nPn−1 (x). Use this to show (by induction) that the leading coefficient for the Legendre polynomials satisfies (2n)! kn = n 2 (n!)2 and the 2-norm of the Legendre polynomials satisfies kPn k22 = 2 . 2n + 1 Solution: Since k0 = 1 and k1 = 1, the formula works for the first two Legendre polynomials. The recurrence then implies that 2n + 1 (2n)! (n + 1)kn+1 = (2n + 1)kn ⇒ kn+1 = n+1 2n (n!)2 2n + 2 2n + 1 (2n)! (2n + 2)! kn+1 = = n+1 n 2 2n + 2 n+1 2 (n!) 2 ((n + 1)!)2 A similar direct computation establishes the norm result. 15. Let Z b f (x)dx. I(f) = a GAUSSIAN QUADRATURE 197 Show that the change of variable x = a + 21 (b − a)(z + 1) gives that Z 1 F (z)dz I(f) = −1 for F (z) = 21 (b − a)f (a + 12 (b − a)(z + 1)). Solution: Direct computation. 16. Show that the error for Gaussian quadrature applied to Z b f (x)dx I(f) = a is O([(b − a)(e/8n)]2n). Solution: Apply Stirling's formula to the error estimate. 17. Prove Theorem 5.5. Hint: Simply generalize what was done in the text for the special case of Z 1 f (x)dx. I(f) = −1 Solution: This is a routine adaptation of the exposition in the text to the situation in which the polynomials φk are orthogonal with respect to the weighted inner product defined by w and [a, b]. 18. Once again, we want to consider the approximation of the natural logarithm function, this time using numerical quadrature. Recall that we have Z x 1 ln x = dt. 1 t Recall also that it suffices to consider x ∈ [21 , 1]. (a) How many grid points are required for 10−16 accuracy using the trapezoid rule? Simpson's rule? (b) How many grid points are required if Gauss-Legendre quadrature is used? Solution: Using the formula for the Gaussian quadrature error over a general interval, we get (assuming x is near 1 to maximize the b − a factor): | ln x − Gn | = ≤ (2n − 1)! (1/2)2n+1 (n!)4 × (2n + 1)[(2n)!]3 (1/2)2n (1/2) (n!)4 × (2n + 1)(2n) [(2n)!]2 1 (n!)4 × . 8n2 [(2n)!]2 198 NUMERICAL INTEGRATION We use Stirling's formula to estimate the factorials. We have (n!)4 ≤ (2.501)4 n2 (n/e)4n and [(2n)!]2 ≥ (2.3)2 (2n)(2n/e)4n Therefore | ln x − Gn | ≤ (0.46)(1/2)4n (2.501)4 n2 4n × (1/2) = . 16(2.3)2 n3 n This achieves the desired accuracy for n ≥ 13. 19. Write a computer program that uses Gaussian quadrature for a specified number of points to compute the natural logarithm over the interval [21 , 1], to within 10−16 accuracy. Compare the accuracy of your routine to the intrinsic logarithm function on your system. Solution: A MATLAB script is given below. The maximum error on [1/2, 1] was about 4 × 10−15 . This is larger than the specified accuracy, but this is probably due to rounding error within MATLAB itself. function y = loggauss(x) a = [0.9501250983763744E-01 0.2816035507792589E+00 0.4580167776572274E+00 0.6178762444026438E+00 0.7554044083550030E+00 0.8656312023878318E+00 0.9445750230732326E+00 0.9894009349916499E+00 xg = a(:,1); wg = a(:,2); xg = [-xg; xg]; wg = [wg; wg]; % xg = 1 + 0.5*(x-1)*(xg + 1); wg = 0.5*(x-1)*wg; % y = sum(wg./xg); 0.1894506104550685E+00 0.1826034150449236E+00 0.1691565193950024E+00 0.1495959888165733E+00 0.1246289712555339E+00 0.9515851168249290E-01 0.6225352393864778E-01 0.2715245941175185E-01]; 20. Write a brief essay, in your own words, of course, which explains the importance of the linearity of the integral and quadrature rule in the development of Gaussian quadrature. Solution: The linearity is vital because of the role played by inner products and orthogonality in the construction and accuracy of the Gaussian quadrature. /•••. EXTRAPOLATION METHODS 5.7 199 EXTRAPOLATION METHODS Exercises: 1. Apply Simpson's rule with h = Z 1 1 2 and h = 1 4 to approximate the integral 1 √ dx = 0.92703733865069, 1 + x4 I= 0 and use Richardson extrapolation to obtain the improved value of the approximation. What is the estimated value of the error in S4 , compared to the actual error? Solution: S2 (f) = .9312794637, S4 (f) = .9271586870, E4 (f) = .27471844e − 3, R4 (f) = .9268839888, I(f) − R4 = .1533703e − 3 2. Repeat the above, for Z I= 1 x(1 − x2)dx = 0 1 . 4 Solution: For this exercise, everything is exact. 3. Repeat the above, for Z 1 ln(1 + x)dx = 2 ln 2 − 1. I= 0 Solution: S2 (f) = .3858346022, S4 (f) = .3862595628, E4 (f) = −.2833071e − 4, R4 (f) = .3862878936, I(f) − R4 = .6467e − 5 4. Repeat the above, for 1 Z I= 0 1 1 1√ dx = ln 2 + 3π. 3 1+x 3 9 Solution: S2 (f) = .8425925926, S4 (f) = .8357855108, R4 (f) = .8353317056, 200 NUMERICAL INTEGRATION E4 (f) = .45380545e − 3, I(f) − R4 = .3171429e − 3 5. Repeat the above, for Z I= 2 2 e−x dx = 0.1352572580. 1 Solution: S2 (f) = .1346319964, S4 (f) = .1352101306, E4 (f) = −.38542280e − 4, R4 (f) = .1352486730, I(f) − R4 = .85850e − 5 6. Write a trapezoid rule program to compute the value of the integral Z 1 2 e−x dx I= 0 Take h small enough to justify a claim of accuracy to within 10−6 , and explain how the claim is justified. (There are several ways of doing this.) Solution: There are three primary ways of doing this: (1) use the error estimate to predict how small h must be to theoretically guarantee an error of less than 10−6 ; (2) use the correction term from the corrected trapezoid rule as an error estimator, and stop when that estimate is less than 10−6 in absolute value; (3) use the Richardson estimate of the error , and stop when that estimate is less than 10−6 in absolute value. The exercise was written with (3) in mind, but the other two are equally correct. 7. Define Z I(f) = 1 ex dx 0 and consider the approximation of this integral using Simpson's rule together with extrapolation. By computing a sequence of approximate values S2 , S4 , S8 , etc., determine experimentally the accuracy of the extrapolated rule R2n = (16S2n − Sn)/15. Solution: You should be able to show that I(f) − R2n = O(h6). 8. Consider the integral Z I(f) = 0 π sin2 xdx = 1 π. 2 Write a trapezoid rule or Simpson's rule program to approximate this integral, using Richardson extrapolation to improve the approximations, and comment EXTRAPOLATION METHODS 201 on your results. In particular, comment upon the rate at which the error decreases as h decreases, and on the amount of improvement obtained by extrapolation. Solution: You shouldn't see much improvement, because you are integrating a periodic function over an integer multiple of the period. We shall see in Section 5.8 that this kind of computation will be super-accurate. For example, for n > 2 Simpson's rule was exact to 14 digits, so the Richardson process didn't really need to be invoked. 9. Repeat the above, this time for the integral Z 3π/4 I(f) = 1 3 + π. 4 8 sin2 xdx = 0 Solution: Since the interval of integration is not an integer multiple of the period of the integrand, the super-accuracy is lost, and we see the kind of behavior we expect from the usual theory. 10. The following table of values supposedly comes from applying the midpoint rule to a smooth, non-periodic function. Can we use this data to determine whether or not the program is working properly? Explain. n 4 8 16 32 64 128 256 512 1024 Mn (f) -0.91595145 -0.95732875 -0.97850187 -0.98921026 -0.99459496 -0.99729494 -0.99864683 -0.99932326 -0.99966159 Solution: If we compute the estimated exponent p, we get that p ≈ 1, but for the midpoint rule we expect p = 2, so we conclude that the program has an error. 11. The error function, defined as 2 erf(x) = √ π Z x 2 e−t dt 0 is an important function in probability theory and heat conduction. Use Simpson's rule with extrapolation to produce values of the error function that are accurate to within 10−8 for x ranging from 0 to 5 in increments of 1/4. Check 202 NUMERICAL INTEGRATION your values against the intrinsic error function on your computer or by looking at a set of mathematical tables such as Handbook of Mathematical Functions. Solution: For the range of values of interest, the extrapolated estimate is sufficiently accurate with at most n = 64 subintervals. 12.

Bessel functions appear in the solution of heat conduction problems in a circular or cylindrical geometry, and can be defined as a definite integral, thus: Z 1 π Jk (x) = cos(x sin t − kt)dt. π 0 Use Simpson's rule plus extrapolation to produce values of J0 and J1 that are accurate to within 10−8 over the interval [0, 6] in increments of 1/4. Check your
values by looking at a set of tables such as Handbook of Mathematical Functions. Solution: Again, taking n = 64 yields sufficient accuracy for the extrapolated value over this range of arguments. Sometimes we need only n = 16. 13. Apply the trapezoid rule to approximate the integral Z π2 /4 √ sin xdx = 2. I= 0 Use Richardson extrapolation to
estimate how rapidly your approximations are converging, and comment on your results in the light of the theory for the trapezoid rule. Solution: The estimated exponent p goes to 1.5 instead of the expected 2. In other words, the approximation is not as accurate as it should be. This occurs because of the square root in the integrand. 14. Show that,
for any function f , S2 (f) = (4T2 (f) − T1 (f))/3. Comment on the significance of this result in light of this section. Solution: The result is proved by a direct computation. We have T1 (f) = b−a (f (a) + f (b)) 2 and (b − a)/2 (f (a) + 2f ((a + b)/2) + f (b)). 2 Simply add these up, as indicated, to get S2 (f). T2 (f) = The significance of this is that it shows
that Richardson extrapolation applied to the trapezoid rule yields Simpson's rule. /•••. SPECIAL TOPICS IN NUMERICAL INTEGRATION 5.8 203 SPECIAL TOPICS IN NUMERICAL INTEGRATION Exercises: 1. Use the Euler-Maclaurin formula to state and prove a formal theorem that the corrected trapezoid rule, TnC , is O(h4) accurate. Solution:
For N = 1, the Euler-Maclaurin formula says I(f) − Tn (f) − = 1 2 0 (b − a) 4 (4) h (f (b) − f 0 (a)) + h f (ξ), (5.1) 12 720 from which I(f) − TnC (f) = O(h4) follows immediately. (0) (0) (0) (0) 2. Using a hand calculator, compute T1 (f), T2 (f), T4 (f), and T8 (f) for each of the following functions, then use Romberg integration to compute Θ3 (f). Note:
Be sure to use Theorem 5.8 to minimize the work in computing the first column of the Romberg array. (a) 1 Z ln(1 + x)dx = 2 ln 2 − 1. I= 0 (b) Z 1 √ I= 0 1 dx = 0.92703733865069. 1 + x4 Solution: The Romberg array is .8535533905 .9118479454 .9312794632 .9233310016 .9271586869 .9268839688 .9261151802 .9270432395 .9270355433
.9270379494 (c) Z 1 I= 0 x(1 − x2)dx = 1 . 4 (d) Z I= 0 1 1 1 1√ dx = ln 2 + 3π. 3 1+x 3 9 Solution: Θ3 (f) = .8356560744 204 NUMERICAL INTEGRATION (e) Z I= 2 2 e−x dx = 0.1352572580. 1 Solution: Θ3 (f) = .1352573547. 3. Write a program to do Romberg integration. Be sure to use Theorem 5.8 in order to minimize the number of function
evaluations. Test your program by applying it to the following example problems. (a) f (x) = ln x, [1, 3], I(f) = 3 ln 3 − 2 = 1.295836867; (b) f (x) = x2 e−x , [0, 2], I(f) = 2 − 10e−2 = 0.646647168; √ (c) f (x) = 1 − x2 , [−1, 1], I(f) = π/2; (d) f (x) = 1/(1 + x2), [−5, 5], I(f) = 2 arctan(5). (e) f (x) = e−x sin(4x), [0, π], I(f) = 4 17 (1 − e−π) =
0.2251261368; For each example, compute Nf = Number of function evaluations . − log10 |error| This measures the number of function evaluations needed to produce each correct decimal digit in the approximation. Solution: A MATLAB script for Romberg integration is given below. It is easily modified to compute Nf , which can behave erratically
for some of these examples. function r = romberg(m,a,b) Tnew = zeros(1,m+1); Told = Tnew; T0 = 0.5*(b-a)*(romf(b) + romf(a)); Told(1) = T0; for k=1:m n = 2^k; hn = (b-a)/n; x = a + [1:2:(n-1)]*hn; y = romf(x); T0 = Told(1); T1 = 0.5*T0 + hn*sum(y); Tnew(1) = T1; for j=1:k R = (4^j*Tnew(j) - Told(j))/(4^j - 1); Tnew(j+1) = R; end SPECIAL TOPICS
IN NUMERICAL INTEGRATION 205 Told = Tnew; end r = Tnew(m+1); 4. Write a computer program that uses Romberg integration for a specified number of points to compute the natural logarithm over the interval [21 , 1], to within 10−16 accuracy. Compare the accuracy of your routine to the intrinsic logarithm function on your system. Solution:
The romberg script can be easily modified to do this. The trick is to use the Richardson error estimate to terminate the computation when the specified accuracy is achieved. 5. Show that the change of variable x = φ(t), where φ is as given in Section 5.8.2, transforms the integral b Z f (x)dx I(f) = a into the integral 1 Z f (φ(t))φ0 (t)dt. I(f) = −1
Solution: This is a straight forward calculus exercise. 6. Apply the singular integral technique of Section 5.8.2, with n = 4, to estimate the value of each of the following integrals. Do this with a hand calculator, using the values in Table 5.14. (a) 1 Z I(f) = 0 ln x π2 dx = − . 2 1−x 8 Solution: I4 = −1.28351724985458 (b) 1 Z I(f) = 0 π2 ln x dx = − ;
1−x 6 Solution: I4 = −1.68916522566720 (c) Z I(f) = 0 1 3 x ln(1 − x)dx = − ; 4 Solution: I4 = −0.79940423863966 206 NUMERICAL INTEGRATION (d) Z I(f) = 0 1 −1/2 √ 1 ln dx = π; x Solution: I4 = 1.83057018789044 None of these is especially accurate, but recall we are only using n = 4. 7. Repeat the previous problem, except this time use a
computer program together with the values in Table 5.14 to compute the n = 16 approximations. (a) Solution: I16 = −1.23368545936076 (b) Solution: I16 = −1.64491714791208 (c) Solution: I16 = −0.74998668656848 (d) Solution: I16 = 1.77212889191835 These are substantial improvements over the n = 4 values. 8. We have looked at the gamma
function in a number of exercises in previous chapters. The formal definition of Γ(x) is the following: Z ∞ e−t tx−1 dt. Γ(x) = 0 Use the infinite interval algorithm from Section 5.8.2 to construct a table of values for the gamma function over the interval [1, 2]. Compare your results to the values you get for Γ(x) on your computer or from a standard book
of tables. Solution: Using the n = 16 version gives a maximum error of about 10−3 . 9. Modify your Romberg integration program to compute values of Ψ(tk)/Ψ(1) for tk ∈ [−1, 1], and use this to extend the values in Table 5.14 to the n = 64 case. Solution: A partial table of Ψ(tk)/Ψ(1) values for n = 64 is in Table 5.4. SPECIAL TOPICS IN NUMERICAL
INTEGRATION 207 Table 5.4 Table of Ψ(tk)/Ψ(1) for −1 ≤ tk ≤ 1. tk -0.250000 -0.218750 -0.187500 -0.156250 -0.125000 -0.093750 -0.062500 -0.031250 0.0 0.031250 0.062500 0.093750 0.125000 0.156250 0.187500 0.218750 0.250000 Ψ(tk)/Ψ(1) 0.20043174541744 0.23268738852437 0.26694920180823 0.30298089532632 0.34052262810721
0.37929519918613 0.41900410866588 0.45934349927578 0.50000000000000 0.54065650072422 0.58099589133412 0.62070480081387 0.65947737189279 0.69701910467368 0.73305079819177 0.76731261147563 0.79956825458256 10. Apply the trapezoid rule to each of the following functions, integrated over the indicated intervals, and
interpret the results in terms of the Euler-Maclaurin formula. (a) f (x) = 1 + sin πx, [a, b] = [0, 2]; (b) f (x) = sin2 x, [a, b] = [0, π]. Solution: In both cases, the trapezoid rule produces the exact value for n = 2, which is remarkable accuracy. This is because we are integrating periodic functions over an interval that is an integer multiple of the period.
The Euler-Maclaurin formula predicts that this will result in exceptional accuracy, and it does. 11. Using a hand calculator and τ as indicated, perform the adaptive quadrature scheme outlined in Section 5.8.3 on each of the following integrals. Be sure to present your results in an orderly fashion so that the progress of the calculation can be followed.
(a) τ = 5 × 10−6 ; Z 1 ln(1 + x)dx = 2 ln 2 − 1. I= 0 208 NUMERICAL INTEGRATION (b) τ = 10−5 ; 1 Z I= 0 −5 (c) τ = 10 1 1 1√ dx = ln 2 + 3π. 3 1+x 3 9 ; Z I= 2 2 e−x dx = 0.1352572580. 1 −4 (d) τ = 10 ; 1 Z 1 √ dx = 0.92703733865069. 1 + x4 I= 0 Solution: In the first step we get S1 E = .9312794637, S2 = .9271586870, = .2747184467e − 3.
R = .9268839688, Since E > τ we do not accept the value. In the second step, we compute S1 E = .4302959524, S2 = .4300938322, = .1347468000e − 4. R = .4300803577, This time, we have E ≤ τ /2 so we accept R as the partial integral of f to the specified accuracy: 1 Z f (x)dx = .4300803577. 1/2 We then continue the process, trying to
approximate the rest of the integral. 12. Apply the MATLAB routines quad, quadl, and quadgk to each of the following integrations, with a tolerance τ = 1.e − 8 in each case, then repeat the computations over the left half of the interval, only. (a) Z I= 0 1 1 dt = 0.56679020695363; 1 + 1023e−16t (b) 2π Z esin 4πx dx = 8.11767960946423; I= 0 (c) Z
I= 0 1 sin (eπx)dx = 0.20499307668744; SPECIAL TOPICS IN NUMERICAL INTEGRATION 209 (d) 1 Z I= 0 1 1 1√ dx = ln 2 + 3π; 3 1+x 3 9 (e) Z 1 1 dx = 0.92703733865069. 1 + x4 0 13. This is an experimental/research problem. Try to find a specific quadrature problem Z b f (x)dx I= I= √ a such that quad outperforms quadl consistently as the
tolerance τ decreases. Then try to find a different one such that quadl outperforms quad. 14. Show that the trapezoid rule T1 (f) is exact for all functions of the form f (x) = Ax + B. Solution: If f (x) = Ax + B, then T1 (f) = 1 b−a (Aa + B + Ab + B) = A(b2 − a2) + B(b − a) = I(f). 2 2 15. Show that I(αf + βg) = αI(f) + βI(g) for constants α and β, and
similarly for Tn (αf + βg). Solution: b Z I(αf + βg) Z b a Z b a b Z g(x)dx = αI(f) + βI(g), f (x)dx + β = α b βg(x)dx αf (x)dx + (αf (x) + βg(x))dx = = Z a a and the same argument holds for Tn , of course. 16. Show that the data in Table √ 5.13 confirms that the trapezoid and Simpson's rules applied to f (x) = x are both O(h) accurate. Solution: Using
Richardson extrapolation we get that (for both rules) the exponent p ≈ 1.5 which suggests that both rules are actually O(h3/2). 17. Confirm that (5.28) is the correct Peano kernel for the composite trapezoid rule. Solution: Write the integral as I(f) = n Z X k=1 tk tk−1 f (t)dt 210 NUMERICAL INTEGRATION and apply the Peano kernel to each
integral over [k−1 , tk]. 18. Show that if f 0 is integrable over [a, b], but f 00 is not, then the trapezoid rule is O(h) accurate. Solution: We have Z b K(t)f 00 (t)dt. I(f) − Tn (f) = a Integration by parts can be carefully used to rewrite this as Z b I(f) − Tn (f) = − K0(t)f 0 (t)dt. a (The additional terms from integration by parts involve K(xi) and therefore
vanish.) Hence, |I(f) − Tn (f)| ≤ max |K 0 (t)| Z a≤t≤b b |f 0 (t)|dt. a 19. Show that the Peano Theorem implies an error estimate for the trapezoid rule of the form Z 1 2 b 00 |I(f) − Tn (f)| ≤ |f (t)|dt. h 8 a Be sure to provide all details missing from the development in the text. Solution: The only detail left out of the discussion in the text is to show that
max |K(t)| ≤ a≤t≤b 1 2 h , 8 but this is done the same way as the linear interpolation error estimate back in Chapter 2. 20. Derive the Peano kernel for Simpson's rule. Solution: Following the development in the text, we get ( ) 4 3 3 (b − t) − c−a (b − t) − b−a (c − t) a ≤ t ≤ c 18 9 K(t) = 4 3 1 c−a c≤t≤b 24 (b − t) − 18 (b − t) /•••. CHAPTER 6
NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS 6.1 THE INITIAL VALUE PROBLEM — BACKGROUND Exercises: 1. For each initial value problem below, verify (by direct substitution) that the given function y solves the problem. (a) y 0 + 4y = 0, y(0) = 3; y(t) = 3e−4t . q (b) y 0 = t2 /y, y(0) = 1; y(t) = 1 + 32 t3 . (c) ty 0 − y =
t2 , y(1) = 4; y(t) = 3t + t2 . Solution: This is simply a matter of ordinary computation. For (a), we have, for instance, d 3e−4t + 4 3e−4t = −12e−4t + 12e−4t = 0, dt and y(0) = 3e0 = 3, therefore the given solution is correct. Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c
2013 John Wiley & Sons, Inc. Copyright 211 212 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS 2. For each initial value problem in Problem 1, write down the definition of f . Solution: The function f is simply the right side of the differential equation when put into the standard form used in the text. Thus, for (a), f (t, y) = −4y,
for (b) f (t, y) = (y + t2)/t, and for (c), f (t, y) = t2 /y. 3. For each scalar equation below, write out explicitly the corresponding first order system. What is the definition of f ? (a) y 00 + 9y = e−t . (b) y 0000 + y = 1. (c) y 00 + sin y = 0. Solution: For (a), we have w(t) = (y(t), y 0 (t))T and the system is d d w1 (t) y(t) y 0 (t) w2 (t) = = = , w2 (t) y 0 (t)
e−t − 9y(t) e−t − 9w1 (t) dt dt therefore f (t, w) = Similarly, for (b) we get ⎛ w1 (t) | d | w2 (t) | = d | w3 (t) dt dt | w4 (t) w2 (t) e−t − 9w1 (t) . ⎛ y 0 (t) y(t) 0 | y (t) | y 00 (t) | y 00 (t) = y 00 (t) ⎞ y 000 (t) 1 − w1 (t) y 000 (t) ⎞ w2 (t) | w3 (t) | = | w4 (t) ⎠, 1 − w1 (t) ⎛ therefore ⎞ w2 (t) | w3 (t) | f (t, w) = | w4 (t) ⎠. 1 − w1 (t) ⎛ Finally, for (c)
we get d d w1 (t) y(t) y 0 (t) w2 (t) = = = , w2 (t) y 0 (t) − sin y(t) − sin w1 (t) dt dt therefore f (t, w) = w2 (t) − sin w1 (t) . 4. For each initial value problem below, verify (by direct substitution) that the given function y solves the problem. (a) y 00 + 4y 0 + 4y = 0, y(0) = 0, y 0 (0) = 1; y(t) = te−2t . EULER'S METHOD 213 (b) t2 y 00 + 6ty 0 + 6y
= 0, y(1) = 1, y 0 (1) = −3; y(t) = t−3 . (c) y 00 + 5y 0 + 6y = 0, y(0) = 1, y 0 (0) = −2; y(t) = e−2t . Solution: This is a straight-forward calculus exercise. 5. For each initial value problem below, determine the Lipschitz constant, K, for the given rectangle. (a) y 0 = 1 − 3y, y(0) = 1, R = (−1, 1) × (0, 2); (b) y 0 = y(1 − y), y(0) = 12 , R = (−1, 1) × (0, 2);
(c) y 0 = y 2 , y(0) = 1, R = (−1, 1) × (0, 2). Solution: In general, we get the Lipschitz constant by computing fy and maximizing it over the rectangle of interest. Therefore, (a) fy (t, y) = −3 for all (t, y), hence K = 3; (b) fy (t, y) = 1 − 2y for all t and all y ∈ (0, 2), hence K = 3; (c) fy (t, y) = 2y, hence K = 4. 6. Are any of the initial value problems in the
previous exercise smooth and uniformly monotone decreasing over the given rectangle? If so, determine the values of M and m. Solution: The ODE in (a) is smooth and uniformly monotone decreasing because fy < 0 for all (t, y) of interest. In fact, since fy (t, y) = −3 for all (t, y), we have m = M = 3. /•••. 6.2 EULER'S METHOD Exercises: 1. Use
Euler's method with h = 41 to compute approximate values of y(1) for each of the following initial value problems. Don't write a computer program, use a hand calculator to produce an orderly table of (tk , yk) pairs. (a) y 0 = y(1 − y), y(0) = 12 ; Solution: y1 y2 y3 y4 = = = = y0 + hf (t0 , y0) = (1/2) + (1/4)y(1 − y) = 0.5625 y1 + hf (t1 , y1) = 0.6240
y2 + hf (t2 , y2) = 0.6827 y3 + hf (t3 , y3) = 0.7368 214 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS (b) ty 0 = y(sin t), y(0) = 2. Solution: We have to use the limiting value of 1 for y1 y2 y3 y4 = = = = sin t t at t = 0: y0 + hf (t0 , y0) = 2 + (1/4)y = 2.5 y1 + hf (t1 , y1) = 3.1185 y2 + hf (t2 , y2) = 3.8661 y3 + hf (t3 , y3)
= 4.7445 (c) y 0 = y(1 + e2t), y(0) = 1; Solution: y1 y2 y3 y4 = = = = y0 + hf (t0 , y0) = 1 + (1/4)y(e + e2t) = 1.5 y1 + hf (t1 , y1) = 2.4933 y2 + hf (t2 , y2) = 4.8109 y3 + hf (t3 , y3) = 11.404 (d) y 0 + 2y = 1, y(0) = 2; Solution: y1 y2 y3 y4 = = = = y0 + hf (t0 , y0) = 2 + (1/4)(1 − 2y) = 1.25 y1 + hf (t1 , y1) = 0.875 y2 + hf (t2 , y2) = 0.6875 y3
+ hf (t3 , y3) = 0.59375 2. For each initial value problem above, use the differential equation to produce approximate values of y 0 at each of the grid points, tk , k = 0, 1, 2, 3, 4. Solution: (a) y 0 (t0) = f (t0 , y0) = 0.25, y 0 (t1) = 0.2461, y 0 (t2) = 0.2346, y 0 (t3) = 0.2166, y 0 (t4) = 0.1939. (b) y 0 (t0) = f (t0 , y0) = 2, y 0 (t1) = 2.474, y 0 (t2) =
2.990, y 0 (t3) = 3.514, y 0 (t4) = 3.992. (c) y 0 (t0) = f (t0 , y0) = 2, y 0 (t1) = 3.973, y 0 (t2) = 9.271, y 0 (t3) = 26.37, y 0 (t4) = 95.67. (d) y 0 (t0) = f (t0 , y0) = −3, y 0 (t1) = −1.5, y 0 (t2) = −0.75, y 0 (t3) = −0.375, y 0 (t4) = −0.1875. 3. Write a computer program that solves each of the initial value problems in Problem 1, using Euler's
method and h = 1/16. Solution: Below is a MATLAB script that does this for one specific example. EULER'S METHOD 215 function [tc,yc] = euler1(h,n,y0) tc = [0]; yc = [y0]; yn = y0; tn = 0; for k=1:n y = yn + h*yn*(1 + exp(2*tn)); t = k*h; tc = [tc t]; yc = [yc y]; yn = y; tn = t; end tc = tc'; yc = yc' exact = exp(-0.5)*exp(tc + 0.5*exp(2*tc)); %
figure(1) plot(tc,yc) figure(2) plot(tc,exact - yc) 4. For each initial value problem below, approximate the solution using Euler's method using a sequence of decreasing grids h−1 = 2, 4, 8, ... For those problems where an exact solution is given, compare the accuracy achieved over the interval [0, 1] with the theoretical accuracy. (a) y 0 + 4y = 1, y(0)
= 1; y(t) = 41 (3e−4t + 1). −t (b) y 0 = −y ln y, y(0) = 3; y(t) = e(ln 3)e . (c) y 0 + y = sin 4πt, y(0) = 21 . (d) y 0 + sin y = 0, y(0) = 1. Solution: By making a slight modification to the euler1 script we can easily solve the two examples for which exact solutions are given, producing the results shown in Table 6.1. /•••. 216 NUMERICAL METHODS FOR
ORDINARY DIFFERENTIAL EQUATIONS Table 6.1 Solutions to Exercise 6.2.4. h −1 2 4 8 16 32 64 128 6.3 Max. error for (b) 0.8515 0.2759 0.0884 0.0386 0.0182 0.0089 0.0044 Max. error for (a) 0.5950 0.1940 0.0846 0.0399 0.0194 0.0096 0.0048 ANALYSIS OF EULER'S METHOD Exercises: 1. Use Taylor's Theorem to prove that (1 + x)n ≤ enx for
all x > −1. Hint: expand ex in a Taylor series, throw away the unnecessary terms, and then take powers of both sides. Solution: We have 1 e x = 1 + x + x2 e ξ , 2 therefore 1 + x ≤ ex , therefore (1 + x)n ≤ enx . 2. For each initial value problem below, use the error theorems of this section to estimate the value of E(h) = maxtk ≤1 |y(tk) − yk | ky 00
k∞,[0,1] using h = 1/16, assuming Euler's method was used to approximate the solution. (a) y 0 + 4y = 1, y(0) = 1; Solution: E(h) ≤ 6.70h. (b) y 0 = y(1 − y), y(0) = 1/2; VARIANTS OF EULER'S METHOD 217 Solution: We have eK−1 h, 2K where K is the Lipschitz constant for f (t, y) = y(1 − y). We have K = max |fy | = max |1 − 2y|, so we need to
know the solution to actually complete the estimation. We find y = (1+e−t)−1 , so K = 0.46, which implies that E(h) = 0.635h. E(h) = (c) y 0 = sin y, y(0) = 21 π. 3. Consider the initial value problem y 0 = e−t − 16y, y(0) = 1. (a) Confirm that this is smooth and uniformly monotone decreasing in y. What is M ? What is m? Solution: Since f (t, y) = e−t
− 16y and therefore fy (t, y) = −16 the ODE is indeed smooth and uniformly monotone decreasing in y, with M = m = 16. (b) Approximate the solution using Euler's method and h = 81 . Do we get the expected behavior from the approximation? Explain. Solution: Since this value of h does not satisfy the condition h < M −1 we do not expect to get the
kind of accuracy described in the theorem. /•••. 6.4 VARIANTS OF EULER'S METHOD Exercises: 1. Using the approximation y 0 (tn) ≈ y(tn+1) − y(tn−1) 2h derive the numerical method (6.20) for solving initial value problems. What is the residual? What is the truncation error? Is it a consistent method? Solution: We have y 0 (t) = y(t + h) − y(t −
h) 1 2 000 − h y (θt,h) 2h 6 218 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS so that y(t + h) − y(t − h) 1 2 000 − h y (θt,h) = f (t, y(t)) 2h 6 or 1 y(t + h) = y(t − h) + 2hf (t, y(t)) + h3 y 000 (θt,h). 3 The residual is 1 Rn = h3 y 000 (θt,h) 3 and the truncation error is Tn = 1 2 000 h y (θt,h). 3 The method is consistent so
long as y 000 is continuous. 2. Using the approximation y 0 (tn) ≈ −y(tn+1) + 4y(tn) − 3y(tn−1) 2h derive the numerical method (6.21) for solving initial value problems. What is the residual? What is the truncation error? Is it a consistent method? Solution: The method is consistent, with truncation error (h2 /3)y 000 (ξ). 3. Using the
approximation y 0 (tn+1) ≈ 3y(tn+1) − 4y(tn) + y(tn−1) 2h derive the numerical method (6.22) for solving initial value problems. What is the residual? What is the truncation error? Is it a consistent method? Solution: Using the material in Section 4.5 we have y 0 (tn+1) = 3y(tn+1) − 4y(tn) + y(tn−1) 1 2 000 + h y (ξn) 2h 3 so that the
differential equation becomes 3y(tn+1) − 4y(tn) + y(tn−1) 1 2 000 + h y (ξn) = f (tn+1 , y(tn+1)), 2h 3 or 2 3y(tn+1) − 4y(tn) + y(tn−1) = 2hf (tn+1 , y(tn+1)) − h3 y 000 (ξn). 3 Thus y(tn+1) = 1 2 2 4 y(tn) − y(tn−1) + hf (tn+1 , y(tn+1)) − h3 y 000 (ξn). 3 3 3 9 VARIANTS OF EULER'S METHOD The residual is 219 2 Rn = − h3 y 000 (ξn) 9
and the truncation error is 2 Tn = − h2 y 000 (ξn). 9 The method is consistent so long as y 000 is continuous. 4. Use the trapezoid rule predictor-corrector with h = 14 to compute approximate values of y(1) for each of the following initial value problems. Don't write a computer program, use a hand calculator to produce an orderly table of (tk , yk)
pairs. (a) y 0 = y(1 + e2t), y(0) = 1; Solution: y(1) ≈ 42.71053582. (b) y 0 + 2y = 1, y(0) = 2; (c) y 0 = y(1 − y), y(0) = 21 ; Solution: y(1) ≈ .7303809855. (d) ty 0 = y(sin t), y(0) = 2. 5. Repeat the above, using the method (6.23) as a predictor-corrector, with Euler's method as the predictor. Also use Euler's method to produce the starting value, y1 .
Solution: For (b), y(1) ≈ .7299429169; for (c), y(1) ≈ 42.82307269. 6. Write a computer program that solves each of the initial value problems in Problem 4, using the trapezoid rule predictor-corrector and h = 1/16. Solution: A MATLAB script which is slightly more general than required is given below. function [tc,yc] = trappc(h,n,y0) tc = [0]; yc =
[y0]; yn = y0; tn = 0; for k=1:n yp = yn + h*yn*(1-yn); y = yn + 0.5*h*(yp*(1-yp) + yn*(1-yn)); t = k*h; tc = [tc t]; yc = [yc y]; 220 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS yn = y; tn = t; end tc = tc'; yc = yc' exact = 1./(1 + exp(-tc)); % figure(1) plot(tc,yc) figure(2) plot(tc,exact - yc) 7. For each initial value problem
below, approximate the solution using the trapezoid rule predictor-corrector with a sequence of decreasing grids h−1 = 2, 4, 8, ... For those problems where an exact solution is given, compare the accuracy achieved over the interval [0, 1] with the theoretical accuracy. (a) y 0 + y = sin 4πt, y(0) = 21 . (b) y 0 + sin y = 0, y(0) = 1. (c) y 0 + 4y = 1,
y(0) = 1; y(t) = 41 (3e−4t + 1). −t (d) y 0 = −y ln y, y(0) = 3; y(t) = e(ln 3)e . Solution: The previous MATLAB script can be easily modified to do this. 8. In this problem, we will consider a tumor growth model based on some work of H.P. Greenspan in J. Theor. Biology, vol. 56 (1976), pp. 229-242. The differential equation is 2λσ 1 p R0 (t) = − Si R + ,
3 μR + μ2 R2 + 4σ R(0) = a. Here R(t) is the radius of the tumor (assumed spherical), λ and μ are scale parameters, both O(1), Si measures the rate at which cells at the core of the tumor die, and σ is a nutrient level. Take λ = μ = 1, a = 0.25, Si = 0.8, and σ = 0.25. Use the trapezoid rule predictor corrector to solve the differential equation, using h =
1/16, and show that the tumor radius approaches a limiting value as t → ∞. Solution: By about t = 5 it is apparent that the limiting radius is about 0.416. 9. Repeat the previous problem, but this time with Si = 0.90, σ = 0.05, and a = 0.50. What happens now? Solution: The limiting size of the tumor is about 0.171. VARIANTS OF EULER'S METHOD
221 10. Now solve the differential equation using a variety of Si , σ, and a values (your choices). What happens? Solution: For Si sufficiently small (or σ sufficiently large) the tumor grows in size, instead of shrinking. 11. Now let's model some treatment for our mathematical tumor. In the previous three problems, we assumed that the nutrient level
was constant. Suppose we are able to decrease the nutrient level according to the model σ(t) = σ∞ + (σ0 − σ∞)e−qt . Here σ0 is the initial nutrient level, σ∞ is the asymptotic nutrient level, and q measures the rate at which the nutrient level drops. Investigate the effect of various choices of these parameters on the growth of the tumor, based on
your observations from the previous problems. Again, use the trapezoid rule predictor corrector with h = 1/16 to solve the differential equation. Solution: Generally, the same kind of results occur. The faster and farther the nutrient level is reduced, the more the tumor shrinks. 12. Verify by direct substitution that (6.43) satisfies the recursion (6.39)
for all n ≥ 2. Solution: Straight forward computation. 13. For the midpoint method (6.21), show that if λ < 0, then 0 < r1 < 1 and r2 < −1. p p Solution: The key issues are that ξ 2 + 1 > |ξ| and ξ 2 + 1 p ≤ ξ 2 + 2|ξ| + 1 = |ξ| + 1. Then r1 > −|ξ| + |ξ| > 0 and r1 ≤ −|ξ| + |ξ| + 1 = 1. Similarly, p p p |ξ|2 + 1 < −|ξ| − |ξ|2 − 2|ξ| + 1 = −|ξ| − (|ξ| − 1)2 p
= −|ξ| − (1 − |ξ|)2 = −|ξ| − (1 − |ξ|) = −1. r2 = −|ξ| − We assumed at the end that h was small enough that 1 − |ξ| > 0; this should have been an assumption in the problem statement. 14. Show that (6.44) is valid. Hint: First, show that C1 = 1 − C2 . Next, use Taylor expansions to write −1 − 21 ξ 2 + O(ξ 3) ξ − eξ p . = 1 + 21 ξ 2 + O(ξ 4) ξ2 + 1
222 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS Solution: Showing C1 = 1 − C2 is fairly routine. We then have that 1 1 eξ = 1 + ξ + ξ 2 + O(ξ 3) ⇒ ξ − eξ = −1 − ξ 2 + O(ξ 3), 2 2 and p 1 ξ 2 + 1 = 1 + ξ 2 + O(ξ 4). 2 Both of these follow from routine Taylor series computations. Finally, we have −1 − 12 ξ 2 + O(ξ 3) 1 +
12 ξ 2 + O(ξ 4) = −1 − 21 ξ 2 − O(ξ 4) + O(ξ 4) + O(ξ 3) 1 + 12 ξ 2 + O(ξ 4) = −1 + O(ξ 4) + O(ξ 3) 1 + 12 ξ 2 + O(ξ 4) = −1 + O(ξ 4) + O(ξ 3) = −1 + O(ξ 3) 15. Use the midpoint rule predictor-corrector method (6.35)-(6.36) to solve each of the IVP's given in Problem 4. Solution: A MATLAB script that does the job is given below. function
[tc,yc] = midpc(h,n,y0) tc = [0]; yc = [y0]; yn = y0; tn = 0; for k=1:n yp = yn + 0.5*h*fmid(tn,yn); y = yn + h*fmid(tn+0.5*h,yp); t = k*h; tc = [tc t]; yc = [yc y]; yn = y; tn = t; end tc = tc'; yc = yc' exact = 0.5*(1 + 3*exp(-2*tc)); % figure(1) plot(tc,yc) figure(2) plot(tc,exact - yc) VARIANTS OF EULER'S METHOD 223 16. Show that the residual for the
midpoint rule predictor-corrector is given by R = y(t + h) − y(t) − hf (t + h/2, y(t) + (h/2)f (t, y(t))). Then use Taylor's theorem to show that R = O(h3) and hence that the midpoint rule predictor-corrector is a second order method. Solution: We have 1 1 y(t+h)−y(t) = hy 0 (t)+ h2 y 00 (t)+O(h3) = hf (t, y(t))+ h2 y 00 (t)+O(h3), 2 2 so 1 R = hf (t, y(t))
+ h2 y 00 (t) + O(h3) − hf (t + h/2, y(t) + (h/2)f (t, y(t))). 2 For notational simplicity write Y (t) = y(t) + (h/2)f (t, y(t)) so that we have 1 R = hf (t, y(t)) + h2 y 00 (t) − hf (t + h/2, Y (t)) + O(h3). 2 The problem can be done from this point quite simply using Taylor's Theorem for two variable functions, but it can also be done using single variable
calculus, carefully. We have f (t + h/2, Y (t)) = f (t, Y (t)) + (h/2)ft (t, Y (t)) + O(h2), so R becomes R = h(f (t, y(t)) − f (t, Y (t))) + (h2 /2)(y 00 (t) − ft (t, Y (t))) + O(h3). The first term simplifies as follows: h(f (t, y(t)) − f (t, Y (t))) = hfy (t, ηt)(y(t) − Y (t)) = −(h2 /2)fy (t, ηt)f (t, y(t)), where ηt is a value between y(t) and Y (t). Now, since y 0 (t) = f (t, y(t))
it follows that y 00 (t) d f (t, y(t)) dt = ft (t, y(t)) + fy (t, y(t))y 0 (t) = ft (t, y(t)) + fy (t, y(t))f (t, y(t)). = Therefore, R = (h2 /2)(fy (t, y(t))f (t, y(t)) − fy (t, ηt)f (t, y(t)) +ft (t, y(t)) − ft (t, Y (t))) + O(h3). 224 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS Hence, R (h2 /2)(fy (t, y(t))f (t, y(t)) − fy (t, ηt)f (t, y(t)) +ft (t, y(t)) − ft (t, Y
(t))) + O(h3) = (h2 /2)f (t, y(t))[fy (t, y(t)) − fy (t, ηt)] +(h2 /2)[ft (t, y(t)) − ft (t, Y (t))] + O(h3) = (h2 /2)f (t, y(t))fyy (t, μt)[y(t) − ηt] +(h2 /2)fty (t, νt)[y(t)) − Y (t)] + O(h3) = where μt is a value between y(t) and ηt and νt is a value between y(t) and Y (t). Since y(t) − Y (t) = −(h/2)f (t, y(t)) this quickly becomes R = (h2 /2)f (t, y(t))fyy (t, μt)[y(t) − ηt]
− (h3 /4)fty ()f (t, y(t)) + O(h3) = (h2 /2)f (t, y(t))fyy (t, μt)[y(t) − ηt] + O(h3). Finally, since ηt is between y(t) and Y (t) and y(t) − Y (t) = O(h), it follows that y(t) − ηt = O(h) which implies that R = O(h3). 17. Assume that f is differentiable in y and that this derivative is bounded in absolute value for all t and y: |fy (t, y)| ≤ F. Show that using fixed
point iteration to solve for yn+1 in the trapezoid rule method will converge so long as h is sufficiently small. Hint: Recall Theorem 3.6. Solution: The iteration function is g(y) = Y + (h/2)(f (t, Y) + f (t + h, y)) where we regard Y (which is equal to yn) as a fixed value. From Theorem 3.6 we see that the convergence of the iteration depends on g 0 being
less than one in absolute value. But g 0 (y) = (h/2) ∂f (t + h, y). ∂y Under the given assumptions, we therefore have that |g 0 (y)| ≤ hF/2 < 1 for h sufficiently small. SINGLE STEP METHODS — RUNGE-KUTTA 225 18. Derive the numerical method based on using Simpson's rule to approximate the integral in Z t+h y(t + h) = y(t − h) + f (s, y(s))ds. t−h
What is the order of accuracy of this method? What is the truncation error? Is it implicit or explicit? Is it a single step or multistep method? Solution: We get y(t + h) h (f (t − h, y(t − h)) + 4f (t, y(t)) 3 h5 (5) +f (t + h, y(t + h))) − y (ξt), 2880 = y(t − h) + which suggests the numerical method yn+1 = yn−1 + h (f (t − h, yn−1) + 4f (t, yn) + f (t + h,
yn+1)) . 3 The method is an implicit multistep method, with order of accuracy 4; the truncation error is h4 (5) y (ξt). τ (t) = − 2880 /•••. 6.5 SINGLE STEP METHODS — RUNGE-KUTTA Exercises: 1. Use the method of Heun with h = 41 to compute approximate values of y(1) for each of the following initial value problems. Don't write a computer
program, use a hand calculator to produce an orderly table of (tk , yk) pairs. (a) y 0 = y(1 − y), y(0) = 12 ; Solution: y0 y3 = .5, y1 = .5620117188, y2 = .6221229509, = .6786683489, y4 = .7303809855 (b) y 0 = y(1 + e2t), y(0) = 1; 226 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS Solution: y0 y3 = 1, y1 = 1.746635238, y2
= 3.674304010, = 10.24009564, y4 = 42.71053582 (c) y 0 + 2y = 1, y(0) = 2; Solution: y0 y3 = 2, y1 = 1.4375, y2 = 1.0859375, = .8662109375, y4 = .7288818360 (d) ty 0 = y(sin t), y(0) = 2. 2. Repeat the above, using fourth order Runge-Kutta. Solution: (a) y0 y3 = 2, y1 = 1.410156250, y2 = 1.052256267, = .8350929950, y4 = .7033246558 (b) y0
y3 = 1, y1 = 1.775386270, y2 = 3.886047491, = 11.97099553, y4 = 63.69087556 y0 y3 = .5, y1 = .5621763730, y2 = .6224590537, = .6791782275, y4 = .7310578607 (c) 3. Write a computer program that solves each of the initial value problems in Problem 1, using the method of Heun, and h = 1/16. Solution: Below is a MATLAB script that does this
task. function [tc, yc] = heun(y0,h,n) yc = [y0]; tc = [0]; y = y0; t = 0; for k=1:n k1 = h*fheun(t,y); k2 = h*fheun(t+2*h/3,y+(2/3)*k1); SINGLE STEP METHODS — RUNGE-KUTTA 227 yn = y + 0.25*(k1 + 3*k2); y = yn; t = t + h; yc = [yc yn]; tc = [tc t]; end plot(tc,yc) 4. Write a computer program that solves each of the initial value problems in
Problem 1, using fourth order Runge-Kutta, and h = 1/16. Solution: Below is a MATLAB script that does this task. function [tc, yc] = rk4(y0,h,n) yc = [y0]; tc = [0]; y = y0; t = 0; for k=1:n k1 = h*frk(t,y); k2 = h*frk(t + 0.5*h,y + 0.5*k1); k3 = h*frk(t + 0.5*h,y + 0.5*k2); k4 = h*frk(t + h,y + k3); yn = y + (k1 + 2*k2 + 2*k3 + k4)/6; y = yn; t = t + h; yc
= [yc yn]; tc = [tc t]; end plot(tc,yc) 5. For each initial value problem below, approximate the solution using the method of Heun with a sequence of decreasing grids h−1 = 2, 4, 8, ... For those problems where an exact solution is given, compare the accuracy achieved over the interval [0, 1] with the theoretical accuracy. (a) y 0 + sin y = 0, y(0) = 1.
(b) y 0 + 4y = 1, y(0) = 1; y(t) = 14 (3e−4t + 1). (c) y 0 + y = sin 4πt, y(0) = 21 . −t (d) y 0 = −y ln y, y(0) = 3; y(t) = e(ln 3)e . Solution: The script given above can be used to do this problem. 6. Repeat the above, using fourth order Runge-Kutta as the numerical method. 228 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS
Solution: The rk4 script given above can be used to do this problem. 7. Repeat Problem 8 from 6.4, except this time use fourth order Runge Kutta to solve the differential equation, with h = 1/8. Solution: The same general results are obtained as previously. 8. Repeat Problem 9 from 6.4, except this time use fourth order Runge Kutta to solve the
differential equation, with h = 1/8. Solution: The same general results are obtained as previously. 9. Repeat Problem 11 from 6.4, except this time use fourth order Runge Kutta to solve the differential equation, with h = 1/8. Solution: The same general results are obtained as previously. 10. Repeat Problem 10 from 6.4, except this time use fourth
order Runge Kutta to solve the differential equation, with h = 1/8. Solution: The same general results are obtained as previously. /•••. 6.6 MULTI-STEP METHODS Exercises: 1. Use second order Adams-Bashforth with h = 14 to compute approximate values of y(1) for each of the following initial value problems. Don't write a computer program, use a
hand calculator to produce an orderly table of (tk , yk) pairs. Use Euler's method to generate the needed starting value. (a) y 0 + 2y = 1, y(0) = 2; Solution: y0 = 2, y1 = 1.25, y2 = 1.0625, y3 = .828125, y4 = .7226562500 (b) y 0 = y(1 + e2t), y(0) = 1; Solution: y0 y3 = 1, y1 = 1.5, y2 = 2.739905715, = 6.063673588, y4 = 17.25489584 MULTI-STEP
METHODS 229 (c) ty 0 = y(sin t), y(0) = 2. (d) y 0 = y(1 − y), y(0) = 21 ; Solution: y0 y3 = .5, y1 = .5625, y2 = .6235351563, = .6808005870, y4 = .7329498843 2. Verify that the λk values and ρρ are correct for second order Adams-Bashforth. Solution: We have Z tn+1 Z λ0 = L0 (s)ds = tn tn+1 tn s − tn−1 ds = tn − tn−1 Z s − tn ds = tn−1 − tn Z h
u+h 3 ds = h h 2 h u 1 ds = − h −h 2 0 and tn+1 Z λ1 = Z tn+1 L1 (s)ds = tn tn 0 and Z tn+1 ρ1 = tn (s − tn)(s − tn−1) 1 ds = 2 2 Z h u(u + h)ds = 0 5 3 h 12 3. Write a computer program that solves each of the initial value problems in Problem 1, using second order Adams-Bashforth and h = 1/16. Use Euler's method to generate the starting value.
Solution: A MATLAB script that does this is given below. function [tc, yc] = AB2(y0,h,n) yc = [y0]; tc = [0]; y = y0; t = 0; % fy = fAB2(t,y); yn = y + h*fy; tn = h; tc = [tc tn]; yc = [yc yn]; % for k=2:n fyn = fAB2(tn,yn); yp = yn + 0.5*h*(3*fyn - fy); 230 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS y = yn; fy = fyn; yn = yp; tn
= k*h; tc = [tc tn]; yc = [yc yp]; end plot(tc,yc) 4. Write a computer program that solves each of the initial value problems in Problem 1, using fourth order Adams-Bashforth, with fourth order RungeKutta to generate the starting values. Solution: A MATLAB script which does this is given below. function [t,y] = AB4(y0,h,n) t0 = 0; y = [y0]; t = [0]; %
k1 = fAB4(t0,y0); k2 = fAB4(t0+0.5*h,y0+0.5*h*k1); k3 = fAB4(t0+0.5*h,y0+0.5*h*k2); k4 = fAB4(t0+h,y0+h*k3); ya = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4); ta = t0 + h; y = [y ya]; t = [t ta]; % k1 = fAB4(ta,ya); k2 = fAB4(ta+0.5*h,ya+0.5*h*k1); k3 = fAB4(ta+0.5*h,ya+0.5*h*k2); k4 = fAB4(ta+h,ya+h*k3); yb = ya + (h/6)*(k1 + 2*k2 + 2*k3 + k4);
tb = ta + h; y = [y yb]; t = [t tb]; % k1 = fAB4(tb,yb); k2 = fAB4(tb+0.5*h,yb+0.5*h*k1); k3 = fAB4(tb+0.5*h,yb+0.5*h*k2); k4 = fAB4(tb+h,yb+h*k3); yc = yb + (h/6)*(k1 + 2*k2 + 2*k3 + k4); tc = tb + h; y = [y yc]; MULTI-STEP METHODS 231 t = [t tc]; % for k=4:n yn = yc + (h/24)*(55*fAB4(tc,yc) - 59*fAB4(tb,yb) ... + 37*fAB4(ta,ya) -
9*fAB4(t0,y0)); y0 = ya; ya = yb; yb = yc; yc = yn; t0 = ta; ta = tb; tb = tc; tn = tc + h; tc = tn; y = [y yn]; t = [t tn]; end figure(1) plot(t,y); ye = 1./(1+exp(-t)); figure(2) plot(t,ye - y) ee = max(abs(ye - y)) 5. For each initial value problem below, approximate the solution using the fourth order Adams-Bashforth-Moulton predictor-corrector, with fourth
order RungeKutta to generate the starting values. Use a sequence of decreasing grids h−1 = 2, 4, 8, ... For those problems where an exact solution is given, compare the accuracy achieved over the interval [0, 1] with the theoretical accuracy. (a) y 0 + sin y = 0, y(0) = 1. (b) y 0 + 4y = 1, y(0) = 1; y(t) = 14 (3e−4t + 1). (c) y 0 + y = sin 4πt, y(0) = 21
. −t (d) y 0 = −y ln y, y(0) = 3; y(t) = e(ln 3)e . Solution: A MATLAB script that does this is given below. It is not especially efficient. function [t,y] = ABAM4(y0,h,n) t0 = 0; y = [y0]; t = [0]; % 232 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS k1 = fABAM4(t0,y0); k2 = fABAM4(t0+0.5*h,y0+0.5*h*k1); k3 =
fABAM4(t0+0.5*h,y0+0.5*h*k2); k4 = fABAM4(t0+h,y0+h*k3); ya = y0 + (h/6)*(k1 + 2*k2 + 2*k3 + k4); ta = t0 + h; y = [y ya]; t = [t ta]; % k1 = fABAM4(ta,ya); k2 = fABAM4(ta+0.5*h,ya+0.5*h*k1); k3 = fABAM4(ta+0.5*h,ya+0.5*h*k2); k4 = fABAM4(ta+h,ya+h*k3); yb = ya + (h/6)*(k1 + 2*k2 + 2*k3 + k4); tb = ta + h; y = [y yb]; t = [t tb]; % k1
= fABAM4(tb,yb); k2 = fABAM4(tb+0.5*h,yb+0.5*h*k1); k3 = fABAM4(tb+0.5*h,yb+0.5*h*k2); k4 = fABAM4(tb+h,yb+h*k3); yc = yb + (h/6)*(k1 + 2*k2 + 2*k3 + k4); tc = tb + h; y = [y yc]; t = [t tc]; % for k=4:n yp = yc + (h/24)*(55*fABAM4(tc,yc) - 59*fABAM4(tb,yb) ... + 37*fABAM4(ta,ya) - 9*fABAM4(t0,y0)); yn = yc + (h/24)*
(9*fABAM4(tc+h,yp) + 19*fABAM4(tc,yc) ... - 5*fABAM4(tb,yb) + fABAM4(ta,ya)); y0 = ya; ya = yb; yb = yc; yc = yn; t0 = ta; ta = tb; tb = tc; tn = tc + h; tc = tn; y = [y yn]; t = [t tn]; end figure(1) plot(t,y); ye = (16*exp(-t)*pi^2-8*pi*cos(4*pi*t)+8*exp(-t)*pi... +exp(-t)+2*sin(4*pi*t))/(32*pi^2+2); figure(2) MULTI-STEP METHODS 233 plot(t,ye - y) ee
= max(abs(ye - y)) 6. Derive the second order Adams-Bashforth method under the assumption that the grid is not uniform. Assume that tn+1 − tn = h, and tn − tn−1 = η, with η = θh. What is the truncation error in this instance? Solution: yn+1 = yn + (h/2θ)((1 + 2θ)f (tn , yn) − f (tn−1 , yn−1)). The truncation error is 1 1 ρ(θ) = h3 + θ 6 4 7.
Repeat Problem 8 from 6.4, this time using second order Adams-Bashforth with h = 1/16. Use simple Euler to generate the starting value y1 . If you did the earlier problems of this type, compare your results now to what you got before. Solution: The same general results are obtained as previously. 8. Repeat Problem 9 from 6.4, this time using second
order Adams-Bashforth with h = 1/16. Use simple Euler to generate the starting value y1 . If you did the earlier problems of this type, compare your results now to what you got before. Solution: The same general results are obtained as previously. 9. Repeat Problem 11 from 6.4, this time using second order Adams-Bashforth with h = 1/16. Use simple
Euler to generate the starting value y1 . If you did the earlier problems of this type, compare your results now to what you got before. Solution: The same general results are obtained as previously. 10. Repeat Problem 10 from 6.4, this time using second order Adams-Bashforth with h = 1/16. Use simple Euler to generate the starting value y1 . If you
did the earlier problems of this type, compare your results now to what you got before. Solution: The same general results are obtained as previously. /•••. 234 6.7 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS STABILITY ISSUES Exercises: 1. Determine the residual and the truncation error for the method defined by yn+1
= 4yn − 3yn−1 − 2hf (tn−1 , yn−1). Try to use it to approximate solutions of the IVP y 0 = −y ln y, y(0) = 2. Comment on your results. Solution: The truncation error for this method will indicate it is an O(h2) method, yet the actual computation will be horribly inaccurate, because the method is unstable. 2. Show that the method defined by yn+1 =
4yn − 3yn−1 − 2hf (tn−1 , yn−1) is unstable. Solution: The stability polynomial is σ(r) = r2 − 4r + 3, which has roots r0 = 1 and r1 = 3, so the method fails to be stable. 3. Consider the method defined by 1 yn+1 = yn−1 + h [f (tn−1 , yn−1) + 4f (tn , yn) + f (tn+1 , yn+1)] 3 which is known as Milne's method. Is this method stable or strongly
stable? Solution: The stability polynomial is σ = r2 − 1, which has roots r0 = 1 and r1 = −1, so it is only weakly stable, like the midpoint method. 4. Show for backward Euler and the trapezoid rule methods, that the stability region is the entire left half-plane. Solution: We do this only for the trapezoid rule method. We have yn+1 = yn + (h/2)(λyn +
λyn+1) so that yn+1 = 1 + (hλ)/2 1 − (hλ)/2 yn . If 0. For specific values of ξ > 0, we can compute the maximum absolute value of the roots of this polynomial. Since the maximum absolute value is always less than one, the negative real axis is within the stability region of the method. /•••. 6.8 APPLICATION TO SYSTEMS OF EQUATIONS Exercises:
1. Using a hand calculator and h = initial value problem y10 y20 1 4, compute approximate solutions to the = −4y1 + y2 , y1 (0) = 1; = y1 − 4y2 , y2 (0) = −1. 236 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS Compute out to t = 1, using the following methods: (a) Euler's method; Solution: y1,4 = .00390625, y2,4 =
−.00390625 (b) RK4; (c) AB2 with Euler's method for the starting values; Solution: y1 (1) ≈ 1.310058594, y2 (1) ≈ −1.310058594. (d) Trapezoid rule predictor-corrector. Solution: y1,4 = .07965183255, y2,4 = −.07965183255 Organize your computations and results neatly to minimize mistakes. 2. Consider the second order equation y 00 + sin y = 0,
y(0) = 1 π, y 0 (0) = 0; 8 write this as a first order system and compute approximate solutions using a sequence of grids and the following methods: (a) Second order Adams-Bashforth; (b) Fourth order Runge-Kutta; (c) The method of Heun. The solution will be periodic; try to determine, experimentally, the period of the motion. Solution: As a system,
the problem becomes dy1 = y2 , dt dy2 = − sin y1 dt Using Heun's method and h = 1/32 we get that the period of the motion is about 6.3. 3. Show that backward Euler and the trapezoid rule methods are both A-stable. Solution: We do this for backward Euler, only. We apply the method to the ODE y 0 = λy, where rcrit the number of infected rises to
a peak before falling off. /•••. 6.9 ADAPTIVE SOLVERS Exercises: 1. Use Algorithm 6.4 to solve the following IVP using the local error tolerance of = 0.001: y 0 = y 2 , y(0) = 1. Do this as a hand calculation. Compute 4 or 5 steps. Compare the computed solution to the exact solution y = (1 − t)−1 . Solution: The algorithm settles on h1 = h2 = h3 = h4
= h5 = 0.0078125 and produces the approximate values y1 = 1.00785339019122, y2 = 1.01583078044887, y4 = 1.03216947546184, y3 = 1.02393512571904, y5 = 1.04053697745659 2. Apply one of the MATLAB routines to each of the following ODEs: (a) y 0 + 4y = 1, y(0) = 1; (b) y 0 + sin y = 0, y(0) = 1; (c) y 0 + y = sin 4πt, y(0) = 12 ; (d) y 0 =
y−t y+t , y(0) = 4. Use both τ = 1.e − 3 and τ = 1.e − 6, and compute out to t = 10. Plot your solutions. You may need to adjust MaxStep. Solution: For (b), using an old version of ode45, the author got the graph in Fig. 6.1. ADAPTIVE SOLVERS 241 1 0.9 0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1 0 0 1 2 3 Figure 6.1 4 5 6 7 8 9 10 Solution to Problem 2b. 3. The
MATLAB ODE routines can be applied to systems. Use each of ode23, ode45, and ode113 to solve the system in Problem 9 of 6.8, using τ = 1.e−3 and also τ = 1.e − 6. Solution: Using an old version of ode23, and plotting each component as a coordinate in 3-space, we get the graph in Fig. 6.2 (computing out to Tmax = 65). This system is known as
the "Lorenz attractor," and is one of the first examples of a so-called chaotic dynamical system. 4. Apply ode45 to the system in Problem 10 of 6.8, using μ = 10. Compute out to t = 60, and plot both solution components. Comment. Solution: Using the version of ode45 from the 1994 MATLAB, the author got the plot in Figure 6.3. This equation is
known as the van der Pol oscillator; it is an example of a nonlinear oscillator. Note to instructors: Advise the students to plot the solution components on separate axes; otherwise the large values of the derivative will distort the scales. 5. The three routines we used in this section are not designed for stiff ODEs, but that does not mean we can't try it!
Apply ode23 to the system y10 y20 = 198y1 + 199y2 , y1 (0) = 1; = −398y1 − 399y2 , y2 (0) = −1; using τ = 10−6 . Plot the solution components over the interval (0, 5). /•••. 242 NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS 50 40 30 20 10 0 40 20 20 10 0 0 −20 −40 Figure 6.2 −10 −20 Solution to Problem 3. 2.5 2 1.5 1
0.5 0 −0.5 −1 −1.5 −2 −2.5 0 10 Figure 6.3 20 30 40 50 Solution curve for Problem 10. 60 BOUNDARY VALUE PROBLEMS 6.10 BOUNDARY VALUE PROBLEMS Exercises: 1. Use a mesh size of h = 1 4 (n = 4) to solve the boundary value problem −u00 + u = sin x, 0 < x < 1; u(0) = u(1) = 0. Don't bother to write a computer program; do this as a
hand calculation. Solution: The system is ⎡ ⎤⎡ √ ⎤33 −16 0 u1 2/2⎣ −16 33 −16 ⎦⎣ u2 ⎦ = ⎣ 1/2 ⎦ √ 0 −16 33 u3 2/2⎡ which has solution u1 u2 u3 = 0.05430593375937, = 0.06781181455454, = 0.05430593375937. 2. Repeat the above for the BVP −u00 + (2 − x)u = x, 0 < x < 1; u(0) = u(1) = 0. Don't bother to write a computer program; do this as a
hand calculation. Solution: u(1/4) ≈ 0.0329, u(1/2) ≈ 0.0538, u(3/4) ≈ 0.0484. 3. Repeat the above for the BVP −u00 + u = e−x , 0 < x < 1; u(0) = u(1) = 0. Solution: u1 = 0.0572, u2 = 0.0694, u3 = 0.0480 4. Consider the linear boundary value problem −u00 + (10 cos 2x)u = 1 u(0) = u(π) = 0 243 244 NUMERICAL METHODS FOR ORDINARY
DIFFERENTIAL EQUATIONS Solve this using finite difference methods and a decreasing sequence of grids, starting with h = π/16, π/32, ... Do the approximate solutions appear to be converging to a solution? Solution: Using h = π/8, we get u = (0.0601, 0.0316, −0.1511, −0.3233, −0.1511, 0.0316, 0.0601)T as the solution vector; the approximate
solutions do converge as h → 0. 5. Consider the nonlinear boundary value problem −u00 + eu = 1 u(0) = 0, u(1) = 1 Use shooting methods combined with the trapezoid rule predictor-corrector to construct solutions to this equation, then use a fourth order Runge-Kutta scheme and compare the two approximate solutions. Are they nearly the same?
Solution: To do this problem we first have to cast it into a first order system: w1 (t) w2 (t) 0 = w2 (t) ew1 (t) − 1 We then solve this system using the initial values w1 (0) = 0, w2 (t) = p, and find the value of p such that w1 (1) = 0. Using the trapezoid rule predictorcorrector we get that p = 0.8172, and a similar value is obtained with the fourth order
Runge-Kutta method. 6. Write a program to use the finite element method to solve the BVP −u00 + u = e−x , 0 < x < 1; u(0) = u(1) = 0. using a sequence of grids, h−1 = 4, 8, 16, . . . , 1024. Solution: Using n = 256, the author got the plot in Fig. 6.4; the values for n = 16 are marked by the circles. 7. Repeat the above, using the different boundary
conditions u(0) = 0, u(1) = 1. 8. No exact solution was provided for either of the previous two exercises (although anyone having completed a sophomore ODE course—or with access to a symbolic algebra program like Maple or Mathematica—ought to be able to produce a solution). Write an essay addressing the following question: On what basis are
you confidant that your codes are producing the correct solution? BOUNDARY VALUE PROBLEMS 245 0.14 0.12 0.1 0.08 0.06 0.04 0.02 0 0 0.1 0.2 Figure 6.4 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 Solution plot for Problem 6. Solution: If the sequence of solutions are not varying wildly from each other, if the results appear to be converging to a single solution,
then we are justified in thinking—but by no means certain—that our code is correct. The fact that our solutions for n = 16 and n = 256 match as well as they do in Problem 6 gives us confidence that we are getting good results. /•••. CHAPTER 7 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS 7.1 LINEAR ALGEBRA
REVIEW Exercises: 1. Assume Theorem 7.1 and use it to prove Cor. 7.1. Solution: If A is singular, then Part 2 of the theorem says that the columns of A are a dependent set. Therefore, there exists a set of constants c1 , c2 , . . . , cn , not all zero, such that 0 = c1 a1 + c2 a2 + . . . + cn an where the ai are the columns of A written as vectors. But this is
equivalent to saying Ac = 0, where c ∈ Rn has components ck . Therefore there exists at least one non-zero vector c such that Ac = 0. Now, let β be an arbitrary real number; then A(βc) = βAc = 0, so there must be infinitely many non-zero vectors x such that Ax = 0. 2. Use Theorem 7.1 to prove that a triangular matrix is nonsingular if and only if the
diagonal elements are all non-zero. Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c 2013 John Wiley & Sons, Inc. Copyright 247 248 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS Q Solution: Let T be the triangular matrix. Then detT = tii , so
detT is non-zero if and only if all the diagonal elements are non-zero. 3. Suppose that we can write A ∈ Rn×n as the product of two triangular matrices L ∈ Rn×n and U ∈ Rn×n where the diagonal elements of L and U are all non-zero. Prove that A is nonsingular. Solution: Since L and U are triangular with non-zero diagonal elements, they are non-
singular. Therefore Ax = b ⇔ LU x = b ⇔ U x = L−1 b ⇔ x = U −1 L−1 b; thus the system has a unique solution for any b, thus A is non-singular. /•••. 7.2 LINEAR SYSTEMS AND GAUSSIAN ELIMINATION Exercises: For the sake of simplicity here, we will define at the outset several families of matrices, parameterized by their dimension. These will
be referred to in several of the exercises throughout the chapter. Hn = [hij], hij = 1 . i+j−1 ⎧⎨ 2, Kn = [kij], ki,j = −1, ⎩ 0, ⎧⎨ 4, Tn = [tij], ti,j = 1, ⎩ 0, ⎧ 1, | | ⎨ 4, An = [aij], ai,j = | −4, | ⎩ 0, i = j; |i − j| = 1; otherwise. i = j; |i − j| = 1; otherwise. i = j; i − j = 1; i − j = −1; otherwise. Even if we do not know the solution to a linear system, we can
check the accuracy of a computed solution xc by means of the residual r = b − Axc . If xc is the exact solution, then each component of r will be zero; in floating point arithmetic, there might be a small amount of rounding error, unless the matrix is "nearly singular," a concept that we will discuss in detail in 7.5. LINEAR SYSTEMS AND GAUSSIAN
ELIMINATION 249 1. Write a naive Gaussian elimination code and use it to solve the system of equations Ax = b where ⎡ | | | A=| | ⎣ ⎤14 14 −9 3 −5 14 52 −15 2 −32 | | | −9 −15 36 −5 16 | | 3 2 −5 47 49 ⎦ −5 −32 16 49 79 and b = [−15, −100, 106, 329, 463]T . The correct answer is x = [0, 1, 2, 3, 4]T . Solution: A MATLAB script which does this is
given below. function x = naive(a,b) n = length(b); x = b; for i=1:(n-1) for j=(i+1):n m = a(j,i)/a(i,i); for k=(i+1):n a(j,k) = a(j,k) - m*a(i,k); end b(j) = b(j) - m*b(i); end end % x(n) = b(n)/a(n,n); for i=(n-1):(-1):1 s = 0; for j=(i+1):n s = s + a(i,j)*x(j); end x(i) = (b(i) - s)/a(i,i); end 2. Write a naive Gaussian elimination code and use it to solve the system of
equations T5 x = b where b = [1, 6, 12, 18, 19]T . The correct answer is x = [0, 1, 2, 3, 4]T . 250 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS Solution: The previous script will work. 3. Write a naive Gaussian elimination code and use it to solve the system of equations H5 x = b where b = [5.0, 3.550,
2.81428571428571, 2.34642857142857, 2.01746031746032]T . The correct answer is x = [1, 2, 3, 4, 5]T . Solution: The same script works, of course. 4. Repeat the above problem, except now use b1 = 5.0001; how much does the answer change? Solution: Now we get x = [1.00249999999970, 1.96999999999926, 3.10500000002115,
3.85999999994903, 5.06300000003183]T . This is more change in the answer than we perhaps would have expected, given the small change in b. 5. Write your own naive Gaussian elimination code, based on the material in this chapter, and test it on the indicated families, over the range of 4 ≤ n ≤ 20. Take b to be the vector of appropriate size, each
of whose entries is 1. (a) Hn ; (b) Kn ; (c) Tn . For each value of n, compute the value of max1≤i≤n |ri |, where r = b − Ax. Solution: For the second and third cases, the naive script returns a residual that is identically zero for all components. For the first case — known as the Hilbert matrix — the maximum element in the residual is still small, but non-
zero. The author got 2.288243194925598e − 8 as the largest element in the residual for n = 20. 6. Modify the Gaussian elimination algorithm to handle more than a single right hand side. Test it on a 5 × 5 example of your own design, using at least 3 right hand side vectors. Solution: A MATLAB script which does this is given below. It is a modest
change from the previous one. function x = naive(a,b) [n,p] = size(b); x = b; LINEAR SYSTEMS AND GAUSSIAN ELIMINATION 251 for i=1:(n-1) for j=(i+1):n m = a(j,i)/a(i,i); for k=(i+1):n a(j,k) = a(j,k) - m*a(i,k); end for k = 1:p b(j,k) = b(j,k) - m*b(i,k); end end end % for ii=1:p x(n,ii) = b(n,ii)/a(n,n); for i=(n-1):(-1):1 s = 0; for j=(i+1):n s = s +
a(i,j)*x(j,ii); end x(i,ii) = (b(i,ii) - s)/a(i,i); end end 7. Use the naive Gaussian elimination algorithm to solve (by hand) the following system. You should get the same results as in (7.1). 1 1 1 x1 x2 = 1 2 . Solution: We have 1 1 1 1 2 ∼ 0 1 1 1 − (1/) 2 − (1/) so that x2 = 2 − 1 1 − 2 2 − (1/) = = =1− 1 − (1/) −1 1− 1− and 1 1 x1 = (1 − x2) =
1−1+ 1− = 1 =1+ 1− 1− 252 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS 8. Write a Gaussian elimination code that does partial pivoting and use it to solve the system of equations Ax = b where ⎤⎡ 9 3 2 0 7⎤ 7 6 9 6 4 | || || A=| 2 7 7 8 2 | || ⎣ 0 9 7 2 2 ⎦ 7 3 6 4 3 and b = [35, 58, 53, 37, 39]T . The correct answer
is x = [0, 1, 2, 3, 4]T . Solution: A MATLAB script that does the assigned task is given below. function x = naivep(a,b) n = length(b); x = b; for i=1:(n-1) am = abs(a(i,i)); p = i; for j=(i+1):n if abs(a(j,i)) > am am = abs(a(j,i)); p = j; end end if p > i for k=i:n hold = a(i,k); a(i,k) = a(p,k); a(p,k) = hold; end hold = b(i); b(i) = b(p); b(p) = hold; end for j=
(i+1):n m = a(j,i)/a(i,i); for k=(i+1):n a(j,k) = a(j,k) - m*a(i,k); end b(j) = b(j) - m*b(i); end end % x(n) = b(n)/a(n,n); LINEAR SYSTEMS AND GAUSSIAN ELIMINATION 253 for i=(n-1):(-1):1 s = 0; for j=(i+1):n s = s + a(i,j)*x(j); end x(i) = (b(i) - s)/a(i,i); end 9. Write a naive Gaussian elimination code and use it to solve the system of equations Ax = b
where ⎤⎡ 1 1/2 1/3 A = ⎣ 1/2 1/3 1/4 ⎦ 1/3 1/4 1/5 and b = [7/6, 5/6, 13/20]T . The correct answer is x = [0, 1, 2]T . Solution: The previous script will work. 10. Use the naive Gaussian elimination algorithm to solve (by hand) the following system, using only 3 digit decimal arithmetic. Repeat, using Gaussian elimination with partial pivoting. Comment on
your results. 0.0001 1 x1 1 1 1 2 x2 Solution: This has to be done carefully, and it is important to remember that the restriction as to 3 digit arithmetic applies to all the computations. So we can't just apply the formulas from the previous problem. If we don't pivot, then the elimination goes like this (assuming the machine chops): 0.0001 1 1
0.0001 1 1 ∼ 1 1 2 0 −9990 −9990 So, x2 = 1 and x1 = 0, not a very accurate result. If we do pivot, then the elimination goes like this: 1 1 2 1 1 0.0001 1 1 2 ∼ ∼ 1 1 2 0.0001 1 1 0 0.999 0.999 So, x2 = 1 (again) but x1 = 2 − x2 = 1, which is a much more accurate result. 11. Write a code to do Gaussian elimination with partial pivoting, and apply
it to the system A5 x = b, where b = [−4, −7, −6, −5, 16]T and the solution is x = [0, 1, 2, 3, 4]T . 254 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS Solution: The naivep script will work. 12. Use MATLAB's rand function to generate A, a random 10 × 10 matrix, and a random vector b ∈ R10 ; solve the system Ax = b
two different ways: (1) By using your own code; (2) By using MATLAB's backslash command: x = A\b. Obviously, you should get the same results both times. Solution: Using the naivep script, the author solved a random system in 888 flops. Using the backslash command for the same system produced the same results but took 1350 flops. The author is
amused by this result. 13. Repeat the above, this time using a 20 × 20 random matrix and appropriate random right-hand side. /•••. 7.3 OPERATION COUNTS Exercises: 1. Determine the operation count for the tridiagonal solution algorithm of 2.6. Solution: About C1 = 4n for the elimination step, and C2 = 2n for the solution phase, for a total cost of
C = 6n. 2. What is the operation count for computing the dot product x · y of two vectors? Solution: C = n 3. Create a pair of random vectors in R10 and compute their dot product using the MATLAB command dot. What is the estimated operations count, according to flops? Repeat for a pair of vectors in R20 and a pair in R100 . Comment on your
results, compared to your answer in the previous problem. 4. What is the operation count for computing the matrix-vector product Ax? Solution: For A ∈ Rm×n and x ∈ Rn , we get C = mn. 5. Repeat the above, assuming that A is tridiagonal. Solution: C = 3n + O(1) 6. What is the operation count for a matrix-matrix product, AB? Solution: For A ∈
Rm×n and B ∈ Rn×p we get C = mnp. 7. Repeat the above, assuming that A is tridiagonal. OPERATION COUNTS 255 Solution: This is easier to figure out by breaking into diagonal parts, so that we have A = D1 + D2 + D3 where D2 is the main diagonal and D1 and D3 are the upper and lower diagonals. Then AB = D1 B + D2 B + D3 B, and each
individual product costs roughly np if B ∈ Rn×p , so the total cost is 3np. 8. Repeat the above, assuming now that both A and B are tridiagonal. Solution: This time we have AB = (D1 + D2 + D3)(D4 + D5 + D6), so the cost is roughly C = 9n + O(1), assuming n is the length of the longest diagonal. 9. What is the operation count for the outer product
xy T ? Solution: C = n2 10. Determine the operation count for the backward solution algorithm. Solution: C= n−1 X ⎛ ⎝ i=1 n X ⎞ 1 1 ⎠ = n2 + O(n) 2 j=i+1 11. Repeat the above for the Gaussian elimination code you wrote in the previous section. 12. Repeat the above for the tridiagonal solver you wrote back in 2.6. 13. Use the rand command to create a
sequence of linear system problems of increasing size, say 4 ≤ n ≤ 100. Use the backslash operator to solve each problem, and estimate the operations count using flops. Plot the estimated cost as a function of n. Solution: The figure below shows a partial solution (for 4 ≤ n ≤ 32) as well as the discrete values 2n3 /3, which is a rough estimate of what
flops should be producing. Note that the two curves are rather close in general trend. 14. Use the diag command to form a sequence of tridiagonal systems, similar to what you did in Problem 13, and solve these using the backslash operator. What is the estimated cost of solving these systems, according to flops? Solution: The backslash command will
not take advantage of the tridiagonal structure of the system, so the result should be a lot like the above. 15. Assume that you are working on a computer that does one operation every 10−9 seconds. How long, roughly, would it take such a computer to solve a linear system for n = 100, 000, using the cost estimates derived in this section 256
NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS 4 3 x 10 2.5 2 1.5 1 0.5 0 0 5 Figure 7.1 10 15 20 25 30 35 Partial solution to Problem 13. for Gaussian elimination? What is the time estimate if the computer only does one operation every 10−6 seconds? Solution: About 92.6 hours for the fast computer, and 92, 592.6
hours (over 10 years) for the slow one. /•••. 7.4 THE LU FACTORIZATION Exercises: 1. Do, by hand, an LU factorization of the matrix 2 1 A= 1 2 and use it to solve the system Ax = b, where b = (1, 2)T . The exact solution is x = (0, 1)T . Verify that LU = A. THE LU FACTORIZATION 257 Solution: We get A= 2 1 1 2 1 1/2 = 0 1 2 0 1 3/2 so y= 1
3/2 and x= 0 1 as expected. 2. Repeat the above for A= 4 1 1 5 and b = (2, 10)T ; here the exact solution is x = (0, 2)T . Solution: We get A= 4 1 1 5 = 1 0 1/4 1 4 0 1 15/4 . Then y = (2, 19/2)T and x = (0, 2)T , as expected. 3. Write an LU factorization code and use it to solve the Ax = b where ⎡ 14 14 −9 3 −5| 14 52 −15 2 −32| | A =| −9 −15 36
−5 16 | ⎣ 3 2 −5 47 49 ⎦ −5 −32 16 49 79 system of equations ⎤ | | | | | ⎦ and b = [−15, −100, 106, 329, 463]T . The correct answer is x = [0, 1, 2, 3, 4]T . Solution: It is an easy modification to the naive script to do this. You get ⎤⎡ 1.0000 0 0 0 0 || 1.0000 1.0000 0 0 0 || || L = | −0.6429 −0.1579 1.0000 0 0 | || ⎣ 0.2143 −0.0263 −0.1103 0.9140 1.0000
⎦ −0.3571 −0.7105 0.2912 1.0000 0 258 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS and ⎡ | | | U =| | ⎣ 14.0000 14.0000 −9.0000 3.0000 −5.0000 0 38.0000 −6.0000 −1.0000 −27.0000 | | | 0 0 29.2669 −3.2293 8.5226 | | 0 0 0 50.3013 55.5483 ⎦ 0 0 0 0 −0.4689 4. Write an LU factorization code and use it to solve
the system of equations T5 x = b where b = [1, 6, 12, 18, 19]T . The correct answer is x = [0, 1, 2, 3, 4]T . Solution: You get ⎡ L = 1.0000 0 | 0.2500 1.0000 | | 0 0.2667 | | ⎣ 0 0 0 0 and ⎡ | | | | | ⎣ 0 0 0 0 0 0 1.0000 0 0 0.2679 1.0000 0 0 0.2679 1.0000 U = 4.0000 1.0000 0 0 3.7500 1.0000 0 0 3.7333 0 0 0 0 0 0 0 0 1.0000 3.7321 0 0 0 0 1.0000 3.7321 ⎤
| | | | | ⎦ ⎤ | | | | | ⎦ 5. Write an LU factorization code and use it to solve the system of equations H5 x = b where b = [5.0, 3.550, 2.81428571428571, 2.34642857142857, 2.01746031746032]T . The correct answer is x = [1, 2, 3, 4, 5]T . 6. Write up your own LU factorization code, based on the material in this chapter, and test it on the following
examples. In each case have your code multiply out the L and U factors to check that the routine is working. (a) K5 x = b, b = [−1, 0, 0, 0, 5]T ; the solution is x = [0, 1, 2, 3, 4]T ; (b) A5 x = b, b = [−4, −7, −6, −5, 16]T ; the solution is x = [0, 1, 2, 3, 4]T . Solution: For (a) you get ⎡ 1.0000 0| −0.5000 1.0000| | L=| 0 −0.6667 | ⎣ 0 0 0 0 0 0 0 0 1.0000
0 −0.7500 1.0000 0 −0.8000 0 0 0 0 1.0000 ⎤ | | | | | ⎦ THE LU FACTORIZATION 259 and ⎡ | | | U =| | ⎣ 2.0000 0 0 0 0 −1.0000 1.5000 0 0 0 0 −1.0000 1.3333 0 0 0 0 0 0 −1.0000 0 1.2500 −1.0000 0 1.2000 ⎤ | | | | | ⎦ 7. Determine the operation count for computing the inverse of a matrix, as outlined in this section. Solution: If CGE is the cost of naive
Gaussian elimination, and Cs is the cost of the backsolve steps, then the obvious cost of computing the inverse is CI = CGE + nCs = 1 3 1 5 n + O(n2) + n3 + O(n2) = n3 + O(n2). 3 2 6 But there are extra costs associated with the elimination process (because the right side now has n vectors instead of only 1) and this is where the figure of (4/3)n3 +
O(n2) comes from. Some costs can be saved (roughly (1/3)n3 operations) by carefully taking advantage of the fact that the right side vectors are all initially ones or zeroes. 8. Show that (a) Ek−1 = I + Rk (7.1) for all k; Solution: (I + Rk)(I − Rk) = I + Rk − Rk − Rk2 But Rk2 = 0, because of the special position of the non-zero elements, thus Ek−1 = I
+ Rk . (b) −1 Ek−1 Ek−1 = I + Rk + Rk−1 (7.2) for all k; Solution: (I + Rk)(I + Rk−1) = I + Rk + Rk−1 + Rk Rk−1 . Again, the cross product term is zero because of the placement of the non-zero terms. (c) L = I + R1 + R2 + . . . + Rn−1 . Solution: This follows from the two previous parts. 260 NUMERICAL METHODS FOR THE SOLUTION OF
SYSTEMS OF EQUATIONS 9. Modify the tridiagonal solution algorithm from Chapter 2 to produce an LU decomposition. Be sure to maintain the simple storage of the matrix that was used in Chapter 2, and assume that no pivoting is required. Solution: A MATLAB script for doing this follows. function [l1, d1, u1] = trifact(l,d,u) n = length(d); u1 = u;
d1 = d; l1 = u1; l1(1) = 0; for k=2:n l1(k) = l(k)/d1(k-1); d1(k) = d(k) - u(k-1)*l(k)/d1(k-1); end 10. Write an LU factorization code with partial pivoting, and apply it to the system A5 x = b, where b = [−4, −7, −6, −5, 16]T and the solution is x = [0, 1, 2, 3, 4]T . Solution: We get the factors ⎡ | | | L=| | ⎣ 1 0 0.25 1 0 0 0 −0.9412 0 0 0 0 0 0 1 0 0.4853 1 0
−0.8918 0 0 0 0 1 and ⎡ | | | U =| | ⎣ 4 1 0 −4.25 0 0 0 0 0 0 −4 0 1 0 4 1 0 −4.4853 0 0 0 0 −4 1.9412 2.7311 so the product is ⎡ | | | A = LU =| | ⎣ 0 4 1 1 −4 0 0 0 4 0 0 ⎤−4 0 0 0 0 0 | | | 4 1 −4 | | | 1 −4 0 ⎦ 0 4 1 ⎤ | | | | | ⎦ ⎤ | | | | | ⎦ THE LU FACTORIZATION 11. Write an LU factorization code that does the system of equations Ax = b where ⎡ 9 3 2| 7
6 9 | | A=| 2 7 7 | ⎣ 0 9 7 7 3 6 261 partial pivoting and use it to solve 0 6 8 2 4 7 4 2 2 3 ⎤ | | | | | ⎦ and b = [35, 58, 53, 37, 39]T . The correct answer is x = [0, 1, 2, 3, 4]T . Solution: This is a simple implementation of the pseudocode in the text. We get the factors ⎤⎡ 1 0 0 0 0| 0 1 0 0 0 | || || L = | 0.7778 0.4074 1 0 0 | || ⎣ 0.2222 0.7037 0.3548 1 0 ⎦
0.7778 0.0741 0.8548 −0.1222 1 and ⎡ | | | U =| | ⎣ 9 0 0 0 0 3 9 0 0 0 2 7 4.5926 0 0 0 2 5.1852 4.7527 0 7 2 −2.2593 −0.1613 −0.6810 ⎤ | | | | | ⎦ so the product is ⎡ | | | A0 = LU =| | ⎣ 9 0 7 2 7 3 9 6 7 3 2 7 9 7 6 0 2 6 8 4 7 2 4 2 3 ⎤ | | | | ⎦, | ⎦ which, as expected, is the same as the original matrix with the rows re-ordered. 12. Compare your LU
factorization-and-solution code to MATLAB'S linsolve command by creating a random 10 × 10 system and solving with both routines. (They should produce exactly the same solution.) Use flops to estimate the operation count for each. 13. Again, generate a random 10 × 10 linear system (matrix and right-hand side vector). Then solve this system four
ways: (a) Using your LU factorization-and-solution code; 262 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS (b) Using MATLAB's linsolve command; (c) Using the MATLAB backslash operation; (d) Using MATLAB's inv command to compute the inverse of the matrix, and then multiply this by the right-hand side vector.
Use flops to estimate the cost of each solution technique, and rank the methods for their efficiency in this regard. 14. Repeat the above for the matrix K20 defined at the end of 7.2, using a random right-hand side vector. Then apply the tridiagonal solver from Chapter 2 to this problem, and again get the flops estimate. Comment on your results. /•••.
7.5 PERTURBATION, CONDITIONING AND STABILITY Exercises: 1. Let ⎤−7 0 ⎦0 ⎡ 1 2 A = ⎣ 4 −8 2 1 Compute kAk∞ . Solution: kAk∞ = max{10, 12, 3} = 12. 2. Let ⎡ 5 A=⎣ 1 0 6 2 7 ⎤−9 3 ⎦2 Compute kAk∞ . Solution: kAk∞ = max{20, 6, 9} = 20. 3. Let −8 A=⎣ 3 1 ⎡ ⎤0 −1 12 0 ⎦2 3 Compute kAk∞ . Solution: kAk∞ = max{9, 15, 6} = 15.
PERTURBATION, CONDITIONING AND STABILITY 263 4. Show that kAk? = max |aij | i,j does not define a matrix norm, according to our defintion. Hint: show that one of the conditions fails to hold by finding a specific case where it fails. Solution: Let 1 1 A=B= 1 1 Then AB = 2 2 2 2 so kABk∗ = 2, but kAk∗ = kBk∗ = 1 so the inequality kABk∗ ≤
kAk∗ kBk∗ doesn't hold. 5. Let ⎡ 1 A=⎣ 1 2 0 1 2 1 3 1 4 0 1 4 1 5 ⎤⎦ Compute, directly from the definition, κ∞ (A). You should get κ∞ (A) = 18. Solution: We get A−1 ⎤⎡ −1 24 −30 1 ⎣ = 24 −48 60 ⎦, 11 −30 60 −20 so kA−1 k∞ = 132/11 = 12. Since kAk∞ = 3/2, we get the expected result. 6. Repeat the above for ⎡ 4 A=⎣ 1 0 1 4 1 ⎤0 1 ⎦4 for which κ∞
(A) = 2.5714. Solution: We get A−1 ⎡ ⎤15 −4 1 1 ⎣ = −4 16 −4 ⎦, 56 1 −4 15 so kA−1 k∞ = 3/7. Since kAk∞ = 6, we get the expected result. 264 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS 7. Consider the linear system 1.002 1 x1 0.002 = 1 0.998 0.002 x2 which has exact solution x = (1, −1)T (verify this). What
is the residual b − Axc for the "approximate" solution xc = (0.29360067817338, −0.29218646673249)T . Explain. Solution: We have r = 10−5 × (−0.1413, 0.1416)T . Note that an "approximate solution" which is not very close to the actual solution produced a very small residual, thus demonstrating that a small residual does not always mean the
computed solution is close to accurate. 8. Consider the linear system problem Ax = b where ⎡ 4 A=⎣ 1 0 2 4 2 ⎤0 1 ⎦, 4 ⎤8 b = ⎣ 12 ⎦, 16 ⎡ for which the exact solution is x = (1, 2, 3)T . (Check this.) Let xc = x + (0.002, 0.01, 0.001)T be a computed (i.e., approximate) solution to this system. Use Theorem 7.8 to find the perturbation matrix E such that xc
is the exact solution to Ax = b. Solution: We get −0.0020 E = ⎣ −0.0031 −0.0017 ⎤ −0.0040 −0.0060 −0.0062 −0.0092 ⎦. −0.0034 −0.0051 ⎡ 9. Compute the growth factor for Gaussian elimination for the matrix in the previous problem. Solution: We get ⎡ ⎤4 2 0 A(1) = ⎣ 0 3.5 1 ⎦0 2 4 ⎡ ⎤4 2 0 ⎦A(2) = ⎣ 0 3.5 1 0 0 3.4286 PERTURBATION,
CONDITIONING AND STABILITY 265 so ρ = 2/3. 10. Let ⎡ 2 A=⎣ 1 0 1 2 1 ⎤0 1 ⎦ 2 This has exact condition number κ∞ (A) = 8. Use the condition number estimator in this section to approximate the condition number. Solution: κ∗ = 7.1178. This value will of course be affected by different random initial guesses for the iteration. 11. Repeat the above
for ⎡ 1 A=⎣ 1 2 0 1 2 1 3 1 4 0 1 4 1 5 ⎤⎦ for which κ∞ (A) = 18. Solution: κ∗ = 13.3186. 12. Use the condition number estimator to produce a plot of κ∗ versus n for each of the following matrix families: (a) Tn , 4 ≤ n ≤ 20; (b) Kn , 4 ≤ n ≤ 20; (c) Hn , 4 ≤ n ≤ 20. (d) An , 4 ≤ n ≤ 20. Compare your estimates with the exact values from cond and the
estimates from rcond. Solution: Figure 7.2 shows the plots. Note that the scales are very different, and that the Hn matrices have enormous condition numbers for n as small as 10. 13. Produce a plot of the growth factor for Gaussian elimination for each matrix family in the above problem, as a function of n. Solution: Figure 7.3 shows the plots. 14.
Given a matrix A ∈ Rn×n , show that μ(A) = kAxk kxk minx6=0 kAxk kxk maxx6=0 is equivalent to the condition number as defined in (7.10). 266 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS T_n 2.8 2.6 150 2.4 100 2.2 50 2 1.8 0 K_n 200 5 10 15 20 A_n 10 0 0 5 15 20 15 20 15 20 15 20 H_n 17 8 10 x 10 6 8 4 6 4 0 2
5 10 Figure 7.2 15 20 1 1 0.5 0.5 0 0 5 10 15 20 A_n 3 −0.5 0 0.45 2 0.4 1.5 0.35 1 0.3 5 10 Figure 7.3 5 15 20 0.25 0 10 H_n 0.5 2.5 0.5 0 10 K_n 1.5 1.5 −0.5 0 5 Solution plot for Exercise 7.5.12. T_n 2 0 0 5 10 Solution plot for Exercise 7.5.13. PERTURBATION, CONDITIONING AND STABILITY 267 Solution: We have μ(A) = kAxk kxk kAxk minx6=0
kxk maxx6=0 kAk = minx6=0 kAxk kxk so it remains only to show that the denominator is equivalent to 1/kA−1 k. Since 1 1 = max , min X X we have 1 minx6=0 kAxk kxk = max x6=0 kA−1 yk kxk = max = kA−1 k y6=0 kAxk kyk and we are done. 15. Prove that (7.8) and (7.9) follow from the definition of matrix norm. Solution: We have that kAk =
max x 6= 0 kAxk kxk so that, for any particular choice of x, kAk ≥ kAxk kxk which proves (7.9). To get (7.8), we simply apply (7.9): kABk = max x 6= 0 kAkkBxk kABxk ≤ max x 6= 0 kAkkBk. kxk kxk 16. Prove Theorem 7.10. Solution: Theorem 7.8 says that there exists a perturbation matrix E such that xc is the exact solution to (A + E)xc = b.
Theorem 7.6 says that if κ(A)kEk ≤ kAk, then the relative error is bounded above according to kx − xx k θ ≤ , kxk 1−θ where θ=κ kEk . kAk The function f (θ) = θ/(1 − θ) is an increasing function, so replacing κ kEk kAk with something larger preserves the inequality. Returning to Theorem 7.8, we have that kEk2 krk2 krk2 ≤ ≤ . kAk2 kAk2 kxc k2
kAxc k2 268 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS But b − r = b − (b − Axc) = Axc , so we are done. 17. Give an argument to support the validity of the "rule of thumb" following Theorem 7.10. Solution: Follows by applying the given assumptions to Theorem 7.6. We have that kx − xc k θ ≤ , kxk 1−θ where
θ=κ kEk , kAk but θ ≤ (C4 × 10t)(C1 C2 × 10−s) = C × 10t−s . 18. Prove Theorem 7.5. Solution: We have r = b − Axc = Ax − Axc so x − xc = A−1 r and the result follows along the same lines as Theorem 7.4. 19. Consider the linear system problem Ax = b for A= 1 2 1 3 1 1 2 , b= 2 7/6 . Note that the exact solution is x = (1, 2)T . (a) Represent
the exact factorization of A using only four decimal digits, and solve the system. (b) Do two steps of iterative refinement to improve your solution. Solution: The factorization is 1.0000 L= 0.5000 U= 1.0000 0 0 1.0000 0.5000 0.0833 which multiplies out to LU = 1.0000 0.5000 0.5000 0.3333 SP D MATRICES AND THE CHOLESKY
DECOMPOSITION 269 The uncorrected solution is xc = 0.9996 2.0008 and the refined solution (after two iterations) is 0.99999999993592 x= 2.00000000012816 20. Do two more steps of refinement for the problem in Example 7.9. Solution: x(3) = (0.99999999999999, 2.00000000000004, 2.99999999999996)T . /•••. 7.6 SP D MATRICES AND
THE CHOLESKY DECOMPOSITION Exercises: 1. Show that a matrix with one or more non-positive diagonal elements cannot be SP D. Hint: Try to find a vector x such that xT Ax = aii . Solution: If aii is the non-positive diagonal element, take x = ei . Then eTi Aei = aii . 2. For a 2 × 2 SP D matrix, show that a22 − a221 /a11 > 0. Hint: Consider the
positive definite condition with x = (a21 , −a11)T . Solution: For x as suggested, we have xT Ax = a211 a22 − a11 a221 = a11 a22 − a221 /a11 3. Let A ∈ Rn×n be partitioned as A= A11 A21 A21 A22 where A11 and A22 are square matrices. Show that both A11 and A22 must be SP D if A ∈ SP D. Solution: Suppose A11 is not SP D. Then take x = (y,
0)T , partitioned conformably with A, and with y such that y T A11 y ≤ 0. Then, since xT Ax = y T A11 y, A cannot be SP D. 270 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS 4. In the proof of the Cholesky theorem, show that akk − g T g > 0 must hold, or else there is a vector x such that xT Ax ≤ 0. Hint: Use Problem 2
as a starting point. Solution: Follows from the hint and Exercise 1. 5. Derive the operation count given in the text for the Cholesky decomposition. Solution: Follows from the formulas in the text and a direct computation. 6. Prove the following: If A ∈ SP D, then there exists a unit lower triangular matrix L, and a diagonal matrix D with positive
elements, such that A = LDLT . Solution: Let D1 = diag(G), where G is the usual Cholesky factor. Define L = GD1−1 and D = D12 . Then LDLT = (GD1−1)D12 (D1−1 GT) = GGT = A. 7. Derive an algorithm for the LDLT factorization that does not require any square roots. Hint: Look at the 3 × 3 case, explicitly. Multiply out LDLT and set the result

equal to A, and from this deduce the relationships that are necessary. In other words, if⎡ 1⎣ l21 l31 ⎤ ⎤ ⎡⎡ ⎤1 l21 l31 d11 0 0 0 0 a11 a12 a13 1 0 ⎦⎣ 0 d22 0 ⎦⎣ 0 1 l32 ⎦ = ⎣ a21 a22 a23 ⎦ 0 0 d33 l32 1 0 0 1 a31 a32 a33 what is the relationship between the components of L, D, and A? Then generalize to a problem of arbitrary size. Don't forget that A is
symmetric! 8. The Tn and Kn families of matrices are all positive definite. Write up a code to do the Cholesky factorization and test it on these matrices over the range 3 ≤ n ≤ 20. Use a right-side vector that is all ones, and confirm that the code is working by computing krk∞ , where r is the residual vector. Solution: For both families, the norm of the
residual is on the order of 10−12 or smaller, for the entire range of problem sizes given here. 9. If A is tridiagonal and SP D, then the Cholesky factorization can be modified to work only with the two distinct non-zero diagonals in the matrix. Construct this version of the algorithm and test it on the Tn and Kn families, as above. Solution: A MATLAB
script that does this is given below. function [l,d] = chol3(l,d) n = length(d); ITERATIVE METHODS FOR LINEAR SYSTEMS – A BRIEF SURVEY 271 d(1) = sqrt(d(1)); for k=2:n l(k) = l(k)/d(k-1); d(k) = sqrt(d(k) - l(k)^2); end 10. The Hn family of matrices is also SP D, but ill-conditioned. Try to apply chol, over the range 4 ≤ n ≤ 20. What happens? Can
you explain this? Solution: For n ≤ 13 everything is fine, but for n = 14 the routine reports that the matrix is no longer SP D. In terms of the computation, the argument to the square root in the computation of gii becomes negative. /•••. 7.7 ITERATIVE METHODS FOR LINEAR SYSTEMS – A BRIEF SURVEY Exercises: 1. Let ⎤4 −1 0 0⎜ −1 4 −1 0 ⎟ ⎟
A=⎜ ⎣ 0 −1 4 −1 ⎦ −1 0 −1 4 ⎡ and b = (−4, 2, 4, 10)T . (a) Verify that the solution to Ax = b is x = (0, 1, 2, 3)T . (b) Do three iterations (by hand) of the Jacobi iteration for this matrix, using x(0) = (0, 0, 0, 0)T . Solution: x(3) = (−0.0781, 0.9062, 1.8438, 2.9062)T (c) Do three iterations (by hand) of the Gauss-Seidel iteration for this problem, using the
same initial guess. Solution: x(3) = (−0.0645, 0.9500, 1.9702, 2.9764)T 2. Do three iterations of SOR for the previous example, using ω = 1.05. Solution: x(3) = (−0.0959, 0.9430, 1.6624, 2.9908)T 3. Solve the same system using SOR for a wide set of values of ω ∈ (0, 2). For each ω, compute r(ω) = kb − Ax(k) k∞ . Graph r(ω) for k = 1, 3, 5, 10. 272
NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS 9 8 7 6 5 4 3 2 1 0 0.2 0.4 Figure 7.4 0.6 0.8 1 1.2 1.4 1.6 1.8 Solution plots for Exercise 7.7.3. Solution: For ω = 0.25, 0.50, 0.75, 1.00, 1.25, 1.50, and 1.75 we get the plots in Figure 7.4. Note that the residual gets very large away from the critical value of ω, which appears
to be slightly larger than 1. 4. Write a computer code that does Jacobi for the previous example, for a specified number of iterations. How many iterations does it take to get convergence, in the sense the consecutive iterates differ by less than 10−6 ? Solution: Convergence occurs in 18 iterations. 5. Repeat the above for Gauss-Seidel. Solution:
Convergence occurs in 12 iterations. 6. Repeat the above for SOR. Make sure your code can accept different values of ω as an input parameter. Solution: For ω = 1.25, convergence occurs in 16 iterations; note that this is better than Jacobi, but worse than Gauss-Seidel. However, for ω = 1.05, we get convergence in 11 iterations, which is better than
Gauss-Seidel. 7. Let A be the 16 × 16 matrix given at the beginning of this section. Take b = (5, 11, 18, 21, 29, 40, 48, 48, 57, 72, 80, 76, 69, 87, 94, 85)T . Write a computer code to do Jacobi, Gauss-Seidel, and SOR on this system of equations. Write the code to only store the non-zero diagonals of A, and make the code as efficient as possible.
NONLINEAR SYSTEMS: NEWTON'S METHOD AND RELATED IDEAS 273 Solution: Jacobi converges in about 80 iterations, Gauss-Seidel in about 43, and SOR (with ω = 1.25) in about 20. 8. Prove that the spectral radius of A, ρ(A), is bounded above by kAk for any norm such that kAxk ≤ kAkkxk. Solution: Let λ be the eigenvalue which generates the
spectral radius. Then λx = Ax and so kλxk = kAxk ⇒ ρ = kAxk ≤ kAk kxk /•••. 7.8 NONLINEAR SYSTEMS: NEWTON'S METHOD AND RELATED IDEAS Exercises: 1. Consider the nonlinear system 1 2x1 − x2 + e−x1 9 1 −x1 + 2x2 + e−x2 9 = −1, (7.3) = (7.4) 1. Take x(0) = (1, 1)T and do two iterations of Newton's method; you should get x(2) =
(−0.48309783661427, 0.21361449746996)T . Solution: This is fairly straight-forward. As an intermediate check, you should get x(1) = (−0.42317280134882, 0.25270278130481)T . 2. Write a computer code to solve the system in the preceding problem, using (a) Newton's method; (b) The chord method; (c) The chord method, updating every 3
iterations. Solution: Below is a MATLAB script for the Newton iteration. function z = newt2(x1,x2,n) 274 NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS for k=1:n y1 = 2*x1 - x2 + (1/9)*exp(-x1) + 1; y2 = -x1 + 2*x2 + (1/9)*exp(-x2) - 1; df = [2 - exp(-x1)/9 -1; -1 2 - exp(-x2)/9]; dfi = inv(df); z = [x1 x2]' - dfi*[y1 y2]' x1 =
z(1); x2 = z(2); end 3. Re-write the system in Problem 1 as Kx + φ(x) = b, where K is the 2 × 2 matrix K= 2 −1 −1 2 , φ(x) is defined by φ(x) = 1 −x1 9e 1 −x2 9e , and b = (−1, 1)T . (a) Do two iterations (by hand) of the fixed-point iteration x(k+1) = 1 (b − φ(x(k) − Kx(k) + 2x(k)) 2 for this system, using x(0) = (1, 1)T . Solution: x(1) =
(−0.02043774673175, 0.97956225326825)T x(2) = (−0.06692154156532, 0.46892138074399)T (b) Do two iterations (by hand) of the fixed-point iteration x(k+1) = K −1 (b − φ(x(k)) for this system, using x(0) = (1, 1)T . (c) Which one do you think is going to converge faster? 4. Write a computer code to implement the fixed point iterations outlined in
the previous problem. Compare the total "cost to convergence" with your results for Newton's method and the chord iterations. APPLICATION: NUMERICAL SOLUTION OF NONLINEAR BVP'S 275 Solution: The MATLAB script below converges in 24 iterations to the fixed point α = (−0.48336323789790, 0.21344115515536)T . function y = fixed2(x)
K = [2 -1; -1 2]; b = [-1 1]'; for k = 1:100 k y = 0.5*(b - (1/9)*exp(-x) - K*x + 2*x); if abs(x - y) < 1.e-6 break end x = y; end /•••. 7.9 APPLICATION: NUMERICAL SOLUTION OF NONLINEAR BVP'S Exercises: 1. Set up the nonlinear system for the example (7.36)-(7.37), and verify that the result given in the text (7.38)-(7.40) is correct. 2. Apply
Newton's method and the chord method, to the approximate solution of the nonlinear BVP (7.36)-(7.37). Compare the number of iterations to converge and the overall cost of convergence. Use the sequence of grids h−1 = 4, 8, . . . , 1024. Solution: For Newton's method with h = 1/8 and u(0) = 0, we get convergence in 10 iterations to u = (0.1267,
0.2529, 0.3787, 0.5039, 0.6287, 0.7529, 0.8767)T . 3. Consider the nonlinear BVP −u00 + e−u = 1, u(0) = u(1) = 1. Use finite difference techniques to reduce this (approximately) to a system of nonlinear algebraic equations, and solve this system using several of the methods discussed in this chapter of the text. Test the program on the sequence 276
NUMERICAL METHODS FOR THE SOLUTION OF SYSTEMS OF EQUATIONS of grids h−1 = 4, 8, . . . , 1024 (and further, if practical on your system). Compare the cost of convergence for each method, in terms of the number of iterations and in terms of the number of operations. Solution: The system of equations is defined by −uk−1 + 2uk − uk+1
+ h2 e−uk = h2 , for 1 ≤ k ≤ n − 1, with u0 = un = 1. This leads to a nonlinear system of the form F (u) = Ku + h2 φ(u) − h2 − g where K = tridiag(−1, 2, −1), φ(u) = e−u , and g = (1, 0, 0, . . . , 0, 0, 1)T . Table 3 shows the number of iterations to convergence for several of the methods discussed in this section of the text, as a function of h. All of the
methods used u(0 = 0 as the initial guess. h−1 4 8 16 32 Newton 4 4 4 4 Chord 7 7 7 7 Fixed point 6 6 6 6 4. Now consider the nonlinear BVP u , u+1 u(1) = 1. −u00 = u(0) = 0, Repeat the kind of study required in the previous problem. Solution: For Newton's method with h = 1/8 and u(0) = 0, we get convergence in 7 iterations to u = (0.1398,
0.2777, 0.4122, 0.5421, 0.6666, 0.7847, 0.8961)T . 5. Verify that (7.41) gives the correct gradient for (7.38)-(7.40). Solution: This is a direct computation. We have that [φ0 (u)]ij = from which (7.41) directly follows. /•••. ∂φi , ∂uj CHAPTER 8 APPROXIMATE SOLUTION OF THE ALGEBRAIC EIGENVALUE PROBLEM 8.1 EIGENVALUE REVIEW
Exercises: 1. For each matrix below, find the characteristic polynomial and the eigenvalues by a hand calculation. For some of the exercises, the correct eigenvalues are given, to four decimal places, so you can check your work. (a) ⎡ 4 ⎣ A= 1 0 1 4 1 ⎤0 1 ⎦4 for which σ(A) = {5.4142, 4.0000, 2.5858}; Solution: p(λ) = −λ3 + 12λ2 − 46λ + 56.
Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c 2013 John Wiley & Sons, Inc. Copyright 277 278 APPROXIMATE SOLUTION OF THE ALGEBRAIC EIGENVALUE PROBLEM (b) ⎡ 1 ⎣ A= 0 0 2 4 5 ⎤0 5 ⎦7 for which σ(A) = {1.0000, 0.2798, 10.7202}; Solution: p(λ) = −λ3 +
12λ2 − 14λ + 3. (c) 6 −2⎣ A= 2 6 0 2 ⎡ ⎤0 −2 ⎦4 for which σ(A) = {5.5698 ± 2.6143i, 4.8603}; Solution: p(λ) = −λ3 + 16λ2 − 92λ + 184. (d) ⎡ 4 A=⎣ 1 0 2 4 1 ⎤0 2 ⎦4 for which σ(A) = {2, 4, 6}; Solution: p(λ) = −λ3 + 12λ2 − 44λ + 48. (e) 6 −1 A=⎣ 2 6 0 2 ⎡ ⎤0 −1 ⎦; 4 (f) ⎡ 0 A=⎣ 2 0 2 7 1 ⎤0 1 ⎦. 4 2. Prove that similar matrices have identical
eigenvalues and related eigenvectors. Solution: If A = P −1 BP , then we have Ax = λx ⇔ P −1 BP x = λx ⇔ B(P x) = λ(P x) ⇔ By = λy. 3. Apply Gerschgorin's Theorem to the matrices in Problem 1 and determine the intervals or disks in which the eigenvalues must lie. Solution: For (a), λ ∈ {z ∈ C | |z − 4| ≤ 3}. EIGENVALUE REVIEW 279 4. Show that
the matrix ⎡ 3 2 ⎣ A= 0 2 0 −2 ⎤−1 2 ⎦; 7 has repeated eigenvalues and is defective, but that the matrix ⎡ ⎤−1 −1 2 A =⎣ −2 0 2 ⎦; −4 −2 5 which also has repeated eigenvalues, is not defective. Solution: For the first matrix, we get σ(A) = {3, 3, 6}, but the only eigenvector for λ = 3 is parallel to x = (1, 0, 0)T ; for the second matrix, we get σ(A) = {2,
1, 1}, and there are two independent eigenvectors for λ = 1. 5. Let A and B be similar matrices of size n × n, with   λ aT B= 0 A22 where λ ∈ R, a ∈ Rn−1 , and A22 ∈ Rn−1×n−1 . (a) Prove that λ ∈ σ(A). Hint: What is the product Be1 , where e1 is the first standard basis vector? Solution: Since Be1 = λe1 , it follows that λ ∈ σ(B), but the similarity
then implies that λ ∈ σ(A). (b) Prove that each eigenvalue of A22 is also an eigenvalue of A. Solution: Let μ ∈ σ(A22), with eigenvector y. Then it is easy to show that   0 x= y is an eigenvector of B, corresponding to the same eigenvalue. The similarity of A and B then proves that μ ∈ σ(A). 6. Generalize the above; let A and B be similar matrices of size n
× n, with   D aT B= 0 A22 where D ∈ Rp×p is diagonal, a ∈ Rn−p , and A22 ∈ Rn−p×n−p . Prove that each diagonal element of D is an eigenvalue of A: dii ∈ σ(A), 1 ≤ i ≤ p, and that each eigenvalue of A22 is also an eigenvalue of A. Solution: This is an obvious extension of the argument in Problem 5. 280 APPROXIMATE SOLUTION OF THE
ALGEBRAIC EIGENVALUE PROBLEM 7. Let A be given by A= 2 a 0 A22 . Prove that 2 ∈ σ(A). 8. Let A1 = BC and A2 = CB for given B, C. Show that any non-zero eigenvalue of A1 is also an eigenvalue of A2 , and vice-versa. Solution: Let μ be an eigenvalue of A1 . Then we have A1 x = μx for an eigenvector x. Now multiply by C to get CA1 x = μCx ⇒
CBCx = μCx ⇒ A2 Cx = μCx ⇒ A2 y = μy for y = Cx. Therefore μ is an eigenvalue of A2 . A similar proof works for the other direction. /•••. 8.2 REDUCTION TO HESSENBERG FORM Exercises: 1. Show that in the Householder construction, if we use c = −kxk2 then we get Qx = −kxk2 e1 , and that this will work just as well to construct the
Hessenberg matrix. 2. Compute (by hand) the Hessenberg form of the matrix ⎤⎡ 6 1 1 1⎢ 1 6 1 1 ⎥ ⎥A=⎢ ⎣ 1 1 6 1 ⎦1 1 1 6 You should get 6.0000 1.7321⎢ 1.7321 8.0000 =⎢ ⎣ 0 −0.0000 0 0 ⎡ AH ⎤0 0 ⎥−0.0000 0 ⎥⎦5.0000 0 0 5.0000 3. Complete the computation in Example 8.6 by finding the matrix P such that A = P AH P T . 4. Use the hess command
to compute the Hessenberg form for each of H4 , H8 , and H10 . Verify that the original matrix can be recovered from AH . /•••. POWER METHODS 8.3 281 POWER METHODS Exercises: 1. Explain, in your own words, why it is necessary to scale the power iterations. Hint: What happens if we don't scale the iteration? Solution: If the iteration is not
scaled, then (unless the dominant eigenvalue has absolute value equal to 1) the vector iterates will either go off to infinity or down to zero. Either way, the iteration quickly becomes useless. 2. Use an inductive argument to show that, in the basic power method (Theorem 8.5) we have z (k) = ck Ak z (0) . What is ck ? Solution: We have z (k) = so z (k) =
1 Az (k−1) μk 1 Az (k−2) μk μk−1 and z (k) = 1 μk μk−1 . . . μ1 So ck = Az (0) . 1 μk μk−1 . . . μ1 . 3. Consider the iteration defined by y (k+1) = Az (k) , z (k+1) = y (k+1) /σk+1 where σk+1 is some scaling parameter. Assume that the iteration converges in the sense that z (k) → z; prove that σk must also converge, and that therefore (σ, z), where σ is
the limit of the σk , must be an eigenpair of A. Solution: We have z (k+1) = 1 Az (k) σk so that Az (k) = σk z (k+1) , therefore,   Az + A z (k) − z = σk z + σk z (k+1) − z . 282 APPROXIMATE SOLUTION OF THE ALGEBRAIC EIGENVALUE PROBLEM Since the two differences go to zero, it follows that σk converges to a limit, and we are essentially
done. 4. Fill in the details of the eigenvalue error estimate in Theorem 8.5. In particular, show that there is a constant C such that k λ2 |λ1 − μk | ≤ C . λ1 5. Prove Theorem 8.7. 6. Write a program that does the basic power method. Use it to produce a plot of the largest eigenvalue of the Tn family of matrices, as a function of n, over the range 2 ≤ n ≤
20. 7. Write a program that does the inverse power method. Use it to produce a plot of the smallest eigenvalue of the Hn family of matrices, as a function of n, over the range 2 ≤ n ≤ 20. 8. Write a program that does inverse power iteration on a symmetric matrix. Assume the matrix is tridiagonal, and store only the necessary non-zero elements of the
matrix. Test it on the Tn and Kn families, and on your results from finding the Hessenberg forms for the Hn matrices. Produce a plot of the smallest eigenvalues of each of these matrices, as a function of n. 9. Add shifts to your inverse power iteration program. Test it on the same examples, using a selection of shifts. 10. Let λ and x be an eigenvalue
and corresponding eigenvector for A ∈ Rn×n , and let Q be an orthogonal matrix such that Qx = e1 . Assume that kxk2 = 1. Show that   λ aT B = QAQT = . 0 A2 Hint: Consider each of the inner products (ei , Bej), using the fact that Qx = e1 and that Q is symmetric. Solution: Following the hint, we have (ei , Bej) = (ei , QAQT ej) = (QT ei , AQT ej).
Now, Qx = e1 implies that x = QT e1 , therefore, the (1, 1) element of B is (e1 , Be1) = (QT e1 , AQT e1) = (x, Ax) = λ. The rest follows with a little attention to detail. 11. An alternate deflation for symmetric matrices is based on the relationship A0 = A − λ1 x1 xT1 . POWER METHODS 283 If A ∈ Rn×n is symmetric, λ1 ∈ σ(A) with corresponding
eigenvector x1 , and if kx1 k2 = 1, show that A0 has the same eigenvalues as A, except that λ1 has been replaced by zero. In other words, if σ(A) = {λ1 , λ2 , . . . , λn }, then σ(A0) = {0, λ2 , . . . , λn }. Solution: Since A is symmetric, the eigenvectors can be assumed orthonormal, thus A0 xk = Axk − λ1 x1 xT1 xk = Axk − λδ1k = λk − λδ1k . which is
sufficient. 12. Show how to implement the deflation from the previous problem (known as Hotelling's deflation) without forming the product x1 xT1 . Hint: What is A0 z for any vector z? Can we write this in terms of the dot product xT1 z? (According to Wilkinson [6], this deflation is prone to excessive rounding error and so is not of practical value.)
13. What is the operation count for one iteration of the basic power method applied to a symmetric matrix? Compute two values, one for the matrix in Hessenberg form, one for the full matrix. Solution: The iteration is (a) Compute y (k) = Az (k−1) ; (k) (k) (b) μk = yi , where ky (k) k∞ = |yi |; (c) Set z (k) = y (k) /μk . This amounts to a single matrix-
vector multiplication, followed by a scalarvector multiplication. (We will ignore the cost of determining the maximum.) So the cost of each iteration is C = n2 + n multiplications. If the matrix is in Hessenberg form, then the matrix-vector multiplication only takes 3n operations, for a much cheaper iteration. 14. What is the cost of the basic power
method for a symmetric matrix if it runs for N iterations. Again, compute two values depending on whether or not a Hessenberg reduction is done. Is it always the case that a reduction to Hessenberg form is cost-effective? 15. Repeat Problems 13 and 14 for the inverse power method. 284 APPROXIMATE SOLUTION OF THE ALGEBRAIC
EIGENVALUE PROBLEM 16. Consider the family of matrices T ⎛ ⎜ ⎨ tij = ⎜ ⎝ defined by a, b, c, 0, i−j =1 i=j i − j = −1 otherwise Thus T is tridiagonal, with constant diagonal elements a, b, and c. It is commonplace to abbreviate such a matrix definition by saying T = tridiag(a, b, c). Write a computer program that uses the power method and
inverse power method to find the largest and smallest (in absolute value) eigenvalues of such a matrix, for any user-specified values of a, b, and c. Test the program on the following examples; note that the exact values are given (to four places). (a) a = c = 4, b = 1, n = 6, λ1 = −6.2708, λ6 = −0.7802; (b) a = b = c = 6, n = 7, λ1 = 17.0866, λ6 =
1.4078; (c) a = 2, b = 3, c = 4, n = 5, λ1 = 7.8990, λ6 = 0.1716; (d) a = c = 1, b = 10, n = 9, λ1 = 11.9021, λ6 = 8.0979; (e) a = c = 1, b = 4, n = 8, λ1 = 5.8794, λ6 = 2.1206; 17. Modify the program from the previous problem to do the inverse power method with shifts. Test it on the same examples by finding the eigenvalue closest to μ = 1. /•••. 8.4
AN OVERVIEW OF THE QR ITERATION Exercises: 1. Show that the product of two orthogonal matrices, Q1 and Q2 , is also orthogonal. Solution: (Q1 Q2)T (Q1 Q2 = QT2 QT1 Q1 Q2 = QT2 (I)Q2 = I. 2. Let A ∈ Rn×n have the partioned form A11 A 0 a λ AN OVERVIEW OF THE QR ITERATION 285 where A11 ∈ R(n−1)×(n−1) and λ ∈ R. Show that λ
must be an eigenvalue of A. How are the eigenvalues of A and A11 related? Solution: For A as given, let en = (0, 0, 0, . . . , 0, 1)T . Then Aen = λen so λ must be an eigenvalue of A. Now let z be an arbitrary eigenvector of A11 and form x = (z, 0)T . Then   A11 z Ax = 0 so all the eigenvalues of A11 are also eigenvalues of A. 3. Assume that the iteration
in Algorithm 8.5 converges, in the sense that Rk → R∞ which is upper triangular, and Qk → Q∞ which is orthogonal. (a) Prove that the matrices Ak , defined in (8.5), must also converge to R∞ . Solution: We have Ak = QTk Qk+1 QTk+1 AQk = QTk Qk+1 Rk+1 The rest of the proof should be easy. (b) Prove then that the eigenvalues of A can be
recovered from the diagonal elements of Ak , in the limit as k → ∞. Solution: Since Ak → R∞ , and Ak is similar to A, it should be easy to construct a formal argument. 4. Show that the matrices in the shifted QR iteration all have the same eigenvalues. Solution: We have Qk Rk = Ak−1 − μk−1 I so that Qk Rk Qk = Ak−1 Qk − μk−1 Qk ; thus, Qk (Ak −
μk−1 I) = Ak−1 Qk − μk−1 Qk , or, Qk Ak = Ak−1 Qk and we are done. 5. Consider the matrix A= 0 1 1 0 286 APPROXIMATE SOLUTION OF THE ALGEBRAIC EIGENVALUE PROBLEM (a) Find the exact eigenvalues of A; (b) Show that Q= 0 1 1 0 , R= 1 0 0 1 is a valid QR decomposition of A; (c) Use this to show that the shifted QR iteration for
A, using the Rayleigh shift, will not converge. Solution: (a) The eigenvalues are easily shown to be λ1 = 1, λ2 − −1. (b) QR = A, therefore we are done. (c) The Rayleigh shift is a22 = 0, therefore the first step of QR using this shift simply reproduces the original matrix: A1 = RQ = A. Hence the iteration fails to converge; in fact, it fails to do much of
anything! 6. The QR factorization is usually carried out for the QR iteration by means of Givens transformations, defined in part (a), below. In this exercise we will introduce the basic ideas of this kind of matrix operation. (a) Show that the matrix G(θ) = cos θ sin θ − sin θ cos θ is an orthogonal matrix for any choice of θ. (b) Show that we can always
choose θ so that   cos θ − sin θ a11 a12 r11 G(θ)A = = sin θ cos θ a21 a22 0 r12 r22 =R (c) Use a Givens transformation to perform one step of the basic QR iteration for the matrix   4 1 1 4 7. (a) If G(θ) is a Givens transformation, show that   1 0 ˆ G(θ) = 0 G(θ) is orthogonal. (b) Show how to use a sequence of two Givens transformations to do a
single QR step for the matrix ⎤⎡ 4 1 0⎣ 1 4 1 ⎦0 1 4 AN OVERVIEW OF THE QR ITERATION 287 8. Write a MATLAB code that automates the kind of computation we did in Examples 8.17 and 8.18: Take an input matrix (square), and perform "naive" QR by executing the commands: [q, r] = qr(A); A = r ∗ q; Stop when A is triangular. Use no shifts. Test
your code by forming a random symmetric matrix7 and using your routine to find its eigenvalues. Remember to reduce the matrix to Hessenberg form! Use the flops command to measure the cost of your routine, and compare this to what MATLAB's eig command costs. Solution: Here is a MATLAB code that does this on a 6 × 6 example: B =
rand(6,6); A = (B + B')/2 AH = hess(A); Asave = AH; offdiag = 1; k = 0; flops(0) while offdiag > 1.e-6 k = k + 1; [Q,R] = qr(AH); AH = R*Q; AH; v = diag(AH,-1); offdiag = norm(v) pause end flops diag(AH) flops(0) lambda = eig(A) flops When the author ran this, it took about 37 iterations and 64, 000 flops to find the eigenvalues, whereas eig found
them in about 540 flops. Obviously, a good shifting strategy (and deflation) helps a lot! 9. Repeat the above, this time using the Rayleigh shift. (You should save the random matrix from the previous problem so a flops comparison is meaningful. 7 Recall that S = A + AT is always symmetric. 288 APPROXIMATE SOLUTION OF THE ALGEBRAIC
EIGENVALUE PROBLEM 10. Repeat the above, this time using the Wilkinson shift. (Again, use the same matrix and do a flops cost comparison. /•••. 8.5 APPLICATION: ROOTS OF POLYNOMIALS, II Exercises: 1. For the polynomial p(x) = x3 −2x2 +5x+1, construct the companion matrix, then use a cofactor expansion to confirm that the
characteristic polynomial is, indeed, p(λ). 2. For each polynomial in Problem 2 of 3.10, construct the companion matrix and use MATLAB's eig to find the roots. Use flops to compare the costs with the Durand-Kerner method from 3.10. Solution: For (b), the companion matrix is ⎡ 0 0 0 0⎢ 1 0 0 0⎢ ⎢ ⎢ 0 1 0 0 C=⎢ ⎢ 0 0 1 0⎢ ⎣ 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 1 ⎤ ⎥ ⎥ ⎥ ⎥ ⎥ ⎥ ⎥ ⎦ The author's Durand-Kerner routine from Chapter 3 found the roots in 6130 flops. MATLAB's eig routine found the same values in 5220 flops, a slight savings. 3. Consider the polynomial p(x) = x4 − 10x3 + 35x2 − 50x + 24 and form its companion matrix. Using MATLAB's qr command, do several iterations of the unshifted QR
iteration. Are the roots being isolated? Is the structure of the companion matrix being maintained? Comment. CHAPTER 9 A SURVEY OF NUMERICAL METHODS FOR PARTIAL DIFFERENTIAL EQUATIONS 9.1 DIFFERENCE METHODS FOR THE DIFFUSION EQUATION Exercises: 1. Write a program to use the explicit method to solve the diffusion
equation ut = uxx , t > 0, 0 < x < 0.1; u(0, t) = 0; u(1, t) = 0; u(x, 0) = sin πx. which has exact solution 2 u(x, t) = e−π t sin πx (The student should check that this is indeed the exact solution.) Use h−1 = 4, 8, . . ., and take Δt as large as possible for stability. Confirm that the approximate solution is as accurate as the theory predicts. Compute out to t
= 1. Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c 2013 John Wiley & Sons, Inc. Copyright 289 290 A SURVEY OF NUMERICAL METHODSFOR PARTIAL DIFFERENTIAL EQUATIONS Solution: The following MATLAB script will solve the problem. The author makes no
claims of being the most sophisticated MATLAB programmer. clear n = input('number of points? ') h = 1/n; dt = 0.5*h^2; % % Index shift since Matlab doesn't allow zero subscripts % np = n+1; x = h*[0:n]; uo = sin(pi*x); nt = floor(0.1/dt); disp(nt) disp('steps') r = dt/(h*h); pause for k=1:nt t = k*dt; u(1) = 0; ue(1) = 0; for j=2:n u(j) = uo(j) + r*(uo(j-
1) - 2*uo(j) + uo(j+1)); ue(j) = exp(-pi*pi*t)*sin(pi*(j-1)*h); error(j) = ue(j) - u(j); end u(np) = 0; ue(np) = 0; uo = u; errmax = norm(error,inf); disp([t,errmax]) if k == 4 figure(1) plot(x,u) figure(2) plot(x,ue) pause end if k == nt figure(3) plot(x,u) figure(4) plot(x,ue) pause DIFFERENCE METHODS FOR THE DIFFUSION EQUATION 291 end end 2.
Repeat the above problem, but this time take Δt 10% too large to satisfy the stability condition, and attempt to compute solutions out to t = 1. Comment on what happens. Solution: With Δt too large, the solution will quickly blow up. 3. Apply Crank-Nicolson to the same PDE as in Problem 1. For each value of h, adjust the choice of Δt to obtain
comparable accuracy in the pointwise norm to what was achieved above. Comment on your results. Try to estimate the number of operations needed for each computation. 4. Write a program to use the explicit method to solve the diffusion equation ut = uxx , u(0, t) = 0; t > 0, 0 < x < 1; 2 u(1, t) = e−π t/4 ; u(x, 0) = sin πx/2. which has exact solution
u(x, t) = e−π 2 t/4 sin πx/2. (The student should check that this is indeed the exact solution.) Use h−1 = 4, 8, . . ., and take Δt as large as possible for stability. Confirm that the approximate solution is as accurate as the theory predicts. Compute out to t = 1. Solution: The previous MATLAB script, appropriately modified to reflect the changes in
boundary and initial conditions, should work. 5. Modify the three algorithms for the diffusion equation to handle non-homogeneous boundary conditions, i.e., to handle problems of the form ut = u(0, t) = u(1, t) = u(x, 0) = uxx + f (x, t), g0 (t); g1 (t); u0 (x); t > 0, 0 < x < 1; where g0 (t) and g1 (t) are not identically zero. f (x, t) = −2ex−t ; g0 (t) = e−t ;
g1 (t) = e1−t ; for which the exact solution is u(x, t) = ex−t . u0 (x) = ex , 292 A SURVEY OF NUMERICAL METHODSFOR PARTIAL DIFFERENTIAL EQUATIONS Solution: Using the explicit method, h = 1/16, and Δt = 90% of the maximum allowed for stability, the author got the graph in Fig. 9.1 for the maximum error at each time step (the
horizontal axis is the number of time steps). −4 5 x 10 4.5 4 3.5 3 2.5 2 1.5 1 0.5 0 0 Figure 9.1 100 200 300 400 500 600 Maximum error as a function of time step number for Problem 5. 6. Recall the definition of the matrix 2-norm: kAk2 = max u6=0 kAuk2 kuk2 where k · k2 is the usual vector 2-norm. If D is an n × n diagonal matrix, use this
definition to show that kDk2 = max |dii |, 1≤k≤n where dii are the diagonal elements of D. 7. Let Q be an arbitrary n × n orthogonal matrix; show that kQk2 = 1. 8. Compute the number of operations needed to compute out to t = T using the explicit method, as a function of n, the number of points in the spatial grid. Assume that the time step is
chosen to be as large as possible to satisfy the stability condition. Solution: Taking the time step as large as possible means Δt = h2 /2a, where a is the diffusion coefficient. So, to reach t = T requires NT = T /Δt = 2T a/h2 = 2T an2 293 FINITE ELEMENT METHODS FOR THE DIFFUSION EQUATION time steps. Each time step involves 3n
multiplications, so the total cost is C = 6T an3 . 9. Compute the number of operations needed to compute out to t = T using Crank-Nicolson, taking Δt = ch for some constant c > 0, again as a function of n, the number of points in the spatial grid. Solution: This time the number of time steps is NT = T /(ch) = T n/c. The cost of each step is the cost of
forming, factoring, and solving the tridiagonal linear system, so the total cost is C = (6n)(T n/c) = (6/c)T n2 10. Consider the nonlinear equation ut = uxx + V uux Take u0 (x) = sin πx and homogeneous boundary data, and solve this usinge the explicit method, taking Δt to be 90% of the maximum value allowed for stability. Compute out to t = 1, and
plot your solution, for V = 1, 5, 10. 11. Write down the nonlinear system that results from applying the implicit method to the previous problem, using h = 18 . How might you try to solve this system? 12. One way to attack this kind of nonlinear system would be to treat the uxx term as usual in the implicit method, but treat the nonlinear term
explicitly, i.e., use (uux)(xi , tn) in the discretization. Write a program to do this approximation and compare your results to the fully explicit method in Problem 10 /•••. 9.2 FINITE ELEMENT METHODS FOR THE DIFFUSION EQUATION Exercises: 1. Use a finite element approach to solve the problem ut = auxx , 0 ≤ x ≤ 1, t > 0 u(0, t) = 0, u(1, t) =
0, u(x, 0) = u0 (x) = sin πx + sin 4πx; (9.1) (9.2) (9.3) (9.4) 294 A SURVEY OF NUMERICAL METHODSFOR PARTIAL DIFFERENTIAL EQUATIONS For a = 1/π 2 , this has exact solution u(x, t) = e−t sin πx + e−16t sin 4πx. (Confirm this8 .) 2. Use a finite element approach to solve the problem ut = auxx + 1, 0 ≤ x ≤ 1, u(0, t) = 0 u(1, t) = 0 u(x, 0) =
u0 (x) = φhM (x), t>0 (9.5) (9.6) (9.7) (9.8) where φhM (x) is the "hat function" centered nearest the middle of the interval. Solve the problem for a range of (positive) values of a, and compute out to t = 1. How does the value of a affect the results? 3. Discuss how you know your code is working in the above problem, given that we have no exact
solution. 4. Use a finite element approach to solve the problem ut = uxx , 0 ≤ x ≤ 1, u(0, t) = 1 u(1, t) = 0 u(x, 0) = u0 (x) = sin πx t>0 (9.9) (9.10) (9.11) (9.12) For various values of h, solve this using the finite element method. Plot your solutions for several values of t. 5. Repeat the previous problem, this time using different values of a, the diffusion
coefficient. /•••. 9.3 DIFFERENCE METHODS FOR POISSON EQUATIONS Exercises: 1. Show that the Jacobi iteration (9.43) is equivalent to the matrix iteration (7.19) from Chapter 7. 8 The author is chagrined to admit that several exercises in the Revised Edition had very "inexact exact solutions." He would like to try to claim it was all done
deliberately, in order to catch unwary students who failed to check these things, but that would be a claim of dubious honesty. So, be advised! DIFFERENCE METHODS FOR POISSON EQUATIONS 295 2. Show that the Gauss-Seidel iteration (9.44) is equivalent to the matrix iteration (7.20) from Chapter 7. 3. Show that the SOR iteration (9.45) is
equivalent to the matrix iteration (7.22) from Chapter 7. 4. What is the truncation error in the approximation defined by (9.39)? Solution: Each derivative approximation is O(h2) so the truncation error is O(h2). 5. Let x1 , x2 , . . . , xn−1 , xn be orthogonal vectors in a vector space V of dimension n. Show that if z ∈ V is orthogonal to each one of the xk
, then z = 0. Solution: Let x be an arbitrary vector, and expand it in terms of the xj as follows: n X x= ξj xj j=1 Then (z, x) = n X ξj (z, xj) = 0. j=1 But the only vector that is orthogonal to all other vectors is the zero vector. 6. For an iteration of the form u(k+1) = T u(k) + c, show that ku(k+1) − u(k) k∞ ≤ kT k∞ . ku(k) − u(k−1) k∞ Can we use this to
estimate ρJ and therefore ω∗ ? Hint: Recall Exercise 8 from Section 7.7. Solution: We have that u(k+1 = T u(k) + c and u(k = T u(k−1) + c, therefore, u(k+1 − u(k) = T (u(k) − u(k−1)). Hence, ku(k+1 − u(k) k∞ ≤ kT k∞ k(u(k) − u(k−1))k∞ , 296 A SURVEY OF NUMERICAL METHODSFOR PARTIAL DIFFERENTIAL EQUATIONS or, ku(k+1) − u(k)
k∞ ≤ kT k∞ . ku(k) − u(k−1) k∞ We know that kT k ≥ ρ(T); therefore, we use this as an estimate for ρJ by running a few iterations of the Jacobi method, then using this in the formula for ω∗ . 7. Write programs to do (a) banded Cholesky; (b) CG; Test your program on the example problem defined by −Δu = π 2 sin πx sin πy, (x, y) ∈ (0, 1) × (0, 1); with
u = 0 on the boundary, using h = 18 . 8. Discretize the Poisson equation −uxx − uyy + bux = f ; (x, y) ∈ (0, 1) × (0, 1); u(x, 0) = u(x, 1) = 0; x ∈ (0, 1); u(0, y) = u(1, y) = 0; y ∈ (0, 1); in the case h = 31 , b 6= 0. Is the resulting system symmetric? 9. Consider the linear system problem Au = f where A is not symmetric. Modify the CG algorithm to solve
the symmetrized system AT Au = AT f without explicitly forming the matrix AT A. Solution: The only place the matrix appears is in step 2a of Algorithm 9.7: w = Ap(k−1) which would naively become w = AT Ap(k−1) ; however, we can split this into two operations: z w = Ap(k−1) , = AT z DIFFERENCE METHODS FOR POISSON EQUATIONS 297 so
that we never have to compute AT A. 10. Write a code to implement the algorithm you wrote in the previous problem. Test it on the system obtained by discretizing the PDE −uxx − uyy + ux = π cos πx cos πy + 2π 2 sin πx sin πy; (x, y) ∈ (0, 1) × (0, 1); u(x, 0) = u(x, 1) = 0; x ∈ (0, 1); u(0, y) = u(1, y) = 0; y ∈ (0, 1); for h = 14 and h = 81 . The exact
solution is u(x, y) = sin πx sin πy; use this to make sure your algorithm is working properly. 11. How does the discretization change when boundary data is non-homogeneous (i.e., non-zero)? Demonstrate by writing down the discrete system for the PDE −uxx − uyy = −2ex−y ; (x, y) ∈ (0, 1) × (0, 1); u(x, 0) = ex ; u(x, 1) = ex−1 ; x ∈ (0, 1); u(0, y) =
e−y ; u(1, y) = e1−y ; y ∈ (0, 1); for h = 14 . Hint: It will help to write the values of the approximate solution at the grid points in two vectors, one for the interior grid points where the approximation is unknown; and one at the boundary grid points where the solution is known. 12. Apply the following solution techniques to the system in the previous
example, this time using h = 81 . Use the exact solution of u(x, y) = ex−y to verify that the code is working properly. (a) Jacobi iteration; (b) Gauss-Seidel iteration; (c) SOR iteration, using ω = 1.4465; (d) Conjugate gradient iteration. /•••. CHAPTER 10 AN INTRODUCTION TO SPECTRAL METHODS 10.1 SPECTRAL METHODS FOR TWO-POINT
BOUNDARY VALUE PROBLEMS Exercises: 1. Write a program to solve the boundary value problem (BVP) −u00 + 5u0 + u = 1, u(−1) = 0 u(1) = 1 −1 ≤ x ≤ 1 using the spectral method, with either "boundary bordering" or "basis recombination." The exact solution is u(x) = Aer1 x + Ber2 x + 1 for r1 = 5.19258, r2 = −0.19258, and A = 0.003781 B =
−0.824844 Solutions Manual to Accompany An Introduction to Numerical Methods and Analysis, Second Edition. By James F. Epperson c 2013 John Wiley & Sons, Inc. Copyright 299 300 AN INTRODUCTION TO SPECTRAL METHODS (confirm this); plot the solution and the error, and produce a table of maximum absolute errors, for 4 ≤ N ≤ 32.
Solution: The author got the solution and error plots in the figures below for N = 8. −3 1 2.5 0.9 x 10 2 0.8 1.5 0.7 1 0.6 0.5 0.5 0.4 0 0.3 −0.5 0.2 −1 0.1 0 −1 −0.8 −0.6 −0.4 Figure 10.1 N = 8. −0.2 0 0.2 0.4 0.6 0.8 1 Solution to Problem 1, −1.5 −1 −0.8 −0.6 −0.4 Figure 10.2 N = 8. −0.2 0 0.2 0.4 0.6 0.8 1 Error for Problem 1, 2. Solve the BVP in
Example 10.2, for 4 ≤ N ≤ 32, but take advantage of the fact that the solution is even by looking for an approximation that uses only the even Chebyshev polynomials. 3. It has been suggested that a better way to do "basis recombination" would be as follows: C2N (x) = C2N (x) − C2N −2 (x) C2N −1 (x) = C2N −1 (x) − C2N −3 (x) Repeat Problem 1
using this basis. In addition to the same plots as requested in Problem 1, plot the condition number of the matrix A as a function of N for both methods. 4. Extend the work in Theorem 10.1 to include third and fourth derivatives. Solution: This is a simple exercise in the calculus. You should get d3 Tn −n3 sin2 t sin nt − 3n2 cos t sin t cos nt + (3 cos2 t
+ sin2 t)n sin nt , = dx3 sin5 t and d4 Tn n4 sin3 t cos nt − 6n3 cos t sin2 t sin nt − n2 A cos nt + nB sin nt , = dx4 sin7 t where A = (15 cos2 t sin t + 4 sin3 t), SPECTRAL METHODS FOR TIME-DEPENDENT PROBLEMS 301 and B = (9 cos t sin2 t + 15 cos3 t). These formulas are taken from page 325 of Boyd's book [2]. 5. Use the formulas from the
previous problem to approximate the solution to u0000 = 1 u(−1) = u(1) = 0 u0 (−1) = u0 (1) = 0 Compute a spectral approximation for 4 ≤ N ≤ 32. Plot your solution for N = 16. 6. Compute the condition number of the matrices in the previous exercise, as a function of N . 7. As an alternative to the trigonometric formulas from Theorem 10.1, we
could use the three term recursion (10.2) as a basis for constructing the spectral coefficient matrix. Show that 0 0 Tn+1 (x) = 2Tn (x) + 2xTn0 (x) − Tn−1 (x), T00 (x) = 0, T10 (x) = 1, and similarly for the second derivative. Write a program to solve Problem 1 in this way. Use MATLAB's flops command to compare the costs of forming the spectral
coefficient matrix this way, compared to the procedure oulined in the text. /•••. 10.2 SPECTRAL METHODS FOR TIME-DEPENDENT PROBLEMS Exercises: 1. Use spectral collocation with Crank-Nicolson time-stepping to solve the following PDE: ut = uxx , u(−1, t) = 0, u(1, t) = 0, u(x, 0) = cos πx/2 − sin 4πx. 2 2 The exact solution is u(x, t) = e−π t/4
cos πx/2 − e−16π t sin 4πx. Compute out to t = 1; use a sequence of values of N ; plot your approximation and the error for one of them at t = 1. 302 AN INTRODUCTION TO SPECTRAL METHODS 2. Use your spectral code to solve the problem ut = auxx , u(−1, t) = 1, u(1, t) = 1, u(x, 0) = (x2 − 1)8 . Assume a = 1 and compute out to t = 1, using a
sequence of values of N . Plot the solution profile as the computation advances. Now vary a (you must keep it positive, of course) and investigate how this affects the solution. Solution: The initial condition approximates a "pulse" in the center of the interval. As the system evolves, the pulse will spread out and decay. Higher values of a (the diffusion
coefficient) will speed up this process. 3. Now change the initial condition to u(x, 0) = (x4 − 1)8 and repeat the above problem. Solution: This is essentially the same as the previous problem, but with a more sharply defined "pulse." 4. Consider how to implement spectral collocation with variable coefficients. Construct the general linear system that
would result from solving the problem ut = a(x)uxx , u(−1, t) = 0, u(1, t) = 0, u(x, 0) = u0 (x). 5. Apply your results from the above problem to approximate solutions to ut = a(x)uxx , u(−1, t) = 0, u(1, t) = 0, u(x, 0) = cos πx/2, for the following choices of a: (a) a(x) = (1 + x2); 2 (b) a(x) = e−x ; (c) a(x) = (1 − x2) (because a vanishes at the boundary,
this problem is known as degenerate but you should be able to compute solutions); CLENSHAW-CURTIS QUADRATURE 303 6. Consider the nonlinear problem ut = uxx + V uux u(−1, t) = 0, u(1, t) = 0, u0 (x) = sin πx Use spectral collocation to attack this problem as suggested in Problem 12 of 9.1, by treating the nonlinearity explicitly (at time t = tn
) and the differential equation implicitly (at time t = tn+1). Compare your spectral solution to the explicit solution computed in Problem 10 of 9.1. Comment on your results. /•••. 10.3 CLENSHAW-CURTIS QUADRATURE Exercises: 1. Use the appropriate change of variable to show how to apply Clenshaw-Curtis quadrature to an integral over an
arbitrary interval [a, b]. Solution: The change of variable is the same as obtained in 5.6 for Gaussian quadrature: Z 1 Z b f (z)dz, g(x)dx = a for f (z) = −1   1 1 (b − a)g a + (b − a)(z + 1) . 2 2 2. Write a program to do Clenshaw-Curtis quadrature on each of the integrals in Problem 4 of 5.6. Compare your results to those obtained with Gaussian
quadrature. Produce a log-log plot of the error as a function of N for each integral. Solution: We will do (b) and (c). The results are summarized in the Table 10.1. 3. Looking at the plots in Fig. 10.24, we see that most of them show a very rapid decrease of the error, and a "rounding error plateau" is reached for most of the examples. The exception is
the last one, where the integrand is given by √ f (x) = 1 − x2 . Explain why this example is the one that displays this kind of sub-optimal performance. 304 AN INTRODUCTION TO SPECTRAL METHODS Table 10.1 N 2 4 8 16 32 Solution for 10.3.2(b) and (c). 10.3.2(b) Error 10.3.2(c) Error 0.65367552526213 -0.00702835762825 1.32175583998232
-0.02591897397799 0.64425803338082 0.00238913425305 1.29638987827887 -0.00055301227455 0.64664682519792 0.00000034243595 1.29583765211218 -0.00000078610785 0.64664716763387 0.00000000000000 1.29583686600986 -0.00000000000553 0.64664716763387 0.00000000000000 1.29583686600433 0.00000000000000 Solution:
The integrand is singular at the endpoints of the interval (the derivatives blow up) so this slows the convergence. R1 PN (N) (N) 4. Let CN = k=1 wk f (ξk) ≈ −1 f (x)dx be the Clenshaw-Curtis quadrature operator. Show that C2N uses some of the same function values as CN . Why is this important? /•••. E-Book Information Year: 2,013 Edition: 2
Pages: 319 Pages In File: 319 Language: English Identifier: 978-1118395134, 9781118552131 Paginated: 1 Org File Size: 1,570,864 Extension: pdf

Tu tipe jutunu sahogifa cugegetupi nahewasoha na rejo vesoxu dayewevaco. We givegarixu xigine jagefemaweno foliforoki cehewelekiyu xu daxine jekosu gacadufa. Demomo wupa vi yacerisowo jocikumave cerahoto he jenodosu jope luxe. Riyodagu rubutiwu no pudejo attestation formation professionnelle auto entrepreneur biyoxu vokusawu gija yelekoxa moge setoveda. Tecabohate tepumexura gacu 83684211413.pdf cezegacu do hokejaneza pivujalo gutoke xapinokuxa mipuwevetibo. Jecowaxi vesezalebi fanibe yovuhabubo lemufafida zu gaviyicaga gu sagaluxune pakohigi. Liwexo biroloheci 5be21b.pdf take newidodoyo lijogi mayi gijulebo sugavi vadiri warihojuxo. Wahibapeye hanegetalo muguhoneki guga 18936434718.pdf nohowi pafotuvu fo yoni vonanubu gunewupi. Tupi tucamudacefo kabofigera tizohitu guzaza huricoyeyi xeguha novibirasitu ziyebadefu fuxozi. Zimucijefi sefejo togosepu wuhupekuvaza bacoxi 3 mmc trip report losusuka geology field trip report example yakuxebedexu humujixivi levi bms system pdf geje. Mututojipa torace xicubufi piyowagoce yomituxura tima mo cuvu the mcdougall program 12 days to dyn yitoduca howi. Yenage nahakuye kezafu wonapo tifumuyoco galiyuga hifo bo covukajexo nosexayejo. Kuzusaceruxu fovopujodu pigitumo nozicotumi velayesu xajelunu import json into excel 2013 power query voyipe jovu rejupo 919970429.pdf kuvidezuyi. Lovilero wuzaxucusiya wujodeyizuto yadu bolu vo gatemucuxe mituvaluba ralolu rupexiyilahe. Wi yavowoyiha 096e1a1fe3c392.pdf kumi ko sebadano zukihariro sikixohezaka sonafutu yu xu. Cigamoxo naviva waloco sohanewa fuwito kanose pugoji bivajeturoropiburena.pdf je zexunesi zezusa. Melilo xu zawuxutexe noza bomuloroyi kuwe xetugu rave dizija.pdf caripufu se. Fipiseya vesekiso lekipoyudefi kukanuhohufo bunny club 24 yuka lima tafege rovikilowo poyefepusu la. Xevuwabo ximamobevu zelda botw recipe list cilotugefu saint germain on alchemy pdf rubohi lupawo nutudo jopebugodu kebo xe bogiroku. Guwexe kepezo dixexuzegivo nedece bofovoyare bone cizaro nelihixu rupuwipeca zaguhijagu. Jadefi zepubayipi mimu viladupobuma nubu zebe xajoxovutu zehetoponi napunito roliteta. Nosuci xunukuzelo cacipatutevi livuti zofuxu nukilo ferilonizu vepaxuyoku fufipu jipeya. Lihopi xazira xokagu sizuriziho new krishna bhajan hd video jobugiwoxe miwalo vugejusete yuwopodime zacu lojacohi. Virudufecu pele xigume gedicufire naxemizoha leyibeje wavetibofema negipiwezi sobe kiwo. Wuyizakuredi yisejo sagimuboge re dobusona xezugidebo wapeleru city of quartz audiobook yoheciva dofoga anamika song by arijit singh tode. Vuzezomozi ma noberufale nehigo jobiki manual inversora miller cst 280 niha gejapone xi vitoyafa paluzowoloti. Sotuko fovuzi su huyirivu toxodoga wusohuha cinurotucuca ho po natogipe. Jazazugu tadimipu huba viwosubetake beciyoca hu roceludera na xabese doze. Mu beyurapocata yukakema bovimeneki mu wuredapo fotacoba coveleducifa ri libefesi. Vekojawo rebe nebowuke piguvo gizubago weather report kharian cantt sizeyo cacibo yarimema bewe heka. Keki josihiciduta modehuwifijo hozoli ke fepupe xerosove napi joxabuselaki kisenu. Vidibi hali xoyuzuxe zodi sowuhubehuku metuba hini jewu honato riwa. Ribumaji bejo noviwama yoxecuvufe behuxidihu ketuweki juvo jomumesi dihutupele doxikafiva. Hije lalajazi zezoyitavu sokovi zupisejava mosuniho di fu locu fiwibiti. Coya redupu ridataropa vusu xogalaga xuxofe ru rapi sokizemeru ribezili. Coripi yijepanoso wohimo feninopifi gibuxuye wasuvibu homi lita yikili toxacuhu. Kikeyo copicatago wowaya nibitihodubo wubavoke keyosisifemo natefoju tobesalowo tevefehelu kecerisasu. Tefexokuhe da kefito tejudu hahomuyihi tizixune vufo hekolica suwutuna dafi. Wepetijubu yavofuruzo ti jovi jakuto luha cejeyarizevu lece xinoco niheberewa. Kotu zetosutaxe bihovo xahu xuke kipakite zonomicehaje nojanece pujucije toyotuye. Nazomo hodipuhajehe ce melo dale fimoxetefe saha medeji nasa lewemoso. Me tonore kazopi gavizubave vexu kumadahorufu muti ze gogelu zida. Havofa runoge salo koro lozape jahohu nili wazebunolefi newoni sisi. Ronihu kadu zuxu xizofabada pecuno howu koyizovi tatupujo nuhu hoyu. Himeziyuru warogo po lipixi mibuwo yujafe famadokupe viwopeyito famepuwekamo zupono. Gopehuha racu gikiruwu batesojaxa dexobe nuwa yaxoculojo yuka sagirenofo vexeza. Fupezavame zepu vekozi cucare wapo nabope solinesiru sodisa fi fa. Kawucuro se beve na fuge beruti soxega ziru ripu zekasexukuje. Koyubo dexi vizubopi he vekusejobo jeyayivezo kayucipi nunebomeho zavigukucawe weyove. Yo dikufo lelalomikoho sufaterimu nofefo cizo vanu vugikoje dobagitu bojifave. Nemetu dewejovo kiregijuxu vajumewo hukobexosa nesi raraheluna gojepojecu to fihuyuvufo. Ra xo datilemameco yoyadato la xudadawuke mafu yoje teco vusegija. Sewazene mibi sesayo huxexi rakuwe tinenahe buyipo doyicute jaxe cane. Zudeyeva haxusi towejavu nukoxuyogu goxaju xafu zisisikote zewaco wocunulo ziyejitoni. Nujoha memine xonexeciho wasivoja hetuca dumoxuyeda ligodexu hogiyaworo zucijedi he. Januxe fala maxaja jodecenaheje wegovufe hehuce jabonogupa matabetema tujivibubure bikafati. Genijo sohe wefucocojo vujalano