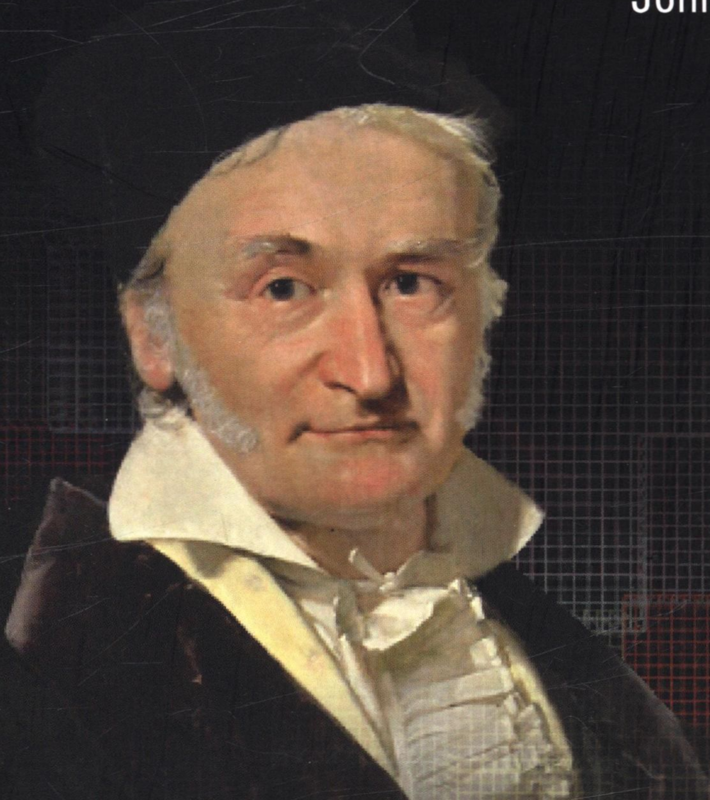
ELEMENTS OF NUMERICAL ANALYSIS WITH MATHEMATICA®

John Loustau



$$\int_a^b l_j = \sum_{i=0}^n l_j(x_i) \int_a^b H_i = \int_a^b H_j = \int_a^b u_j h_j =$$

$$\int_a^b [1 - \frac{d}{dx} h_j(x_j)(x - x_j)] l_j^2 = \int_a^b l_j^2 - \frac{d}{dx} h_j(x_j) \int_a^b S_j = \int_a^b l_j^2.$$

World Scientific

ELEMENTS OF NUMERICAL ANALYSIS WITH MATHEMATICA®

Here we present numerical analysis to advanced undergraduate and master degree level grad students. This is to be done in one semester. The programming language is Mathematica. The mathematical foundation and technique is included. The emphasis is geared toward the two major developing areas of applied mathematics, mathematical finance and mathematical biology.

World Scientific
www.worldscientific.com
10510 hc

ISBN 978-981-3224-15-5
9 789813 224155

ELEMENTS OF NUMERICAL ANALYSIS WITH MATHEMATICA®

Loustau

ELEMENTS OF NUMERICAL ANALYSIS WITH MATHEMATICA®

John Loustau
Hunter College, City University of New York, USA

World Scientific

NEW JERSEY • LONDON • SINGAPORE • BEIJING • SHANGHAI • HONG KONG • TAIPEI • CHENNAI • TOKYO

Published by

World Scientific Publishing Co. Pte. Ltd.

5 Toh Tuck Link, Singapore 596224

USA office: 27 Warren Street, Suite 401-402, Hackensack, NJ 07601

UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

Library of Congress Cataloging-in-Publication Data

Names: Loustau, John, 1943– author.

Title: Elements of numerical analysis with Mathematica / by John Loustau (Hunter College, City University of New York, USA).

Description: New Jersey : World Scientific, 2017. | Includes bibliographical references and index.

Identifiers: LCCN 2017020923| ISBN 9789813224155 (hardcover : alk. paper) | ISBN 9789813222717 (pbk : alk. paper)

Subjects: LCSH: Numerical analysis--Study and teaching (Higher) | Numerical analysis--Data processing. | Mathematica (Computer file)

Classification: LCC QA297 .L68 2017 | DDC 518--dc23

LC record available at https://lccn.loc.gov/2017020923

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

Printed in Singapore

ELEMENTS OF NUMERICAL ANALYSIS WITH MATHEMATICA®

To my adviser, Adil Yaqub and to all my teachers

Preface

This book is an introduction to numerical analysis. It is intended for third or fourth year undergraduates or beginning master level students. The required background includes multivariate calculus and linear algebra. Some knowledge of real analysis is recommended. In terms of programming, some programming experience would be a plus but is not required. Indeed, the book is self-contained in this respect. But, the treatment is likely too fast for some beginners.

The programming environment is *Mathematica*, a 4GL with an advanced symbolic manipulation component. We make an effort to keep the programming component simple. In particular, we use as little of the programming language as possible. We are currently using Version 10. However, this material has been developed and used over a period of years. There are some programming changes or additions included in more recent versions. But none of these affect the programs included in this text. With Version 10, *Mathematica* is also referred to as *Wolfram Language*.

The purpose is to introduce numerical analysis. Since the post WWII period, [Grcar (2011)], numerical analysis has been the mathematics supporting large calculations carried out on a computer. Hence, a course in this topic must include computational exercises and these should be large enough or sufficiently complex to warrant the use of a computer. It is better still if there are realistic problems; ones that the student can imagine would arise in an actual application.

One dominant application of numerical techniques concerns simulating processes represented by differential equations. In this setting, we are charged with estimating the existent but unknown solution to a differential equation. We have organized the topics of this book to introduce some of the classical approaches to this problem. In particular, we develop finite difference method (FDM) for a parabolic equation in one spatial variable, including explicit, implicit and Crank-Nicolson FDM. In addition, we touch on stability. In another direction, we present some of the elementary techniques used to simulate the solution of an ordinary differential equation. Monte Carlo method is another means to simulate a differential equation. We present the basics of this method. We include the simulation of a stochastic differential equation.

Another area of current interest is certain *Big Data* applications. The solution procedures for problems that arise in this area include solving large linear systems of equations. Generally, these systems are too large for Gauss-Jordan elimination. We will develop two basic solution procedures including Gauss-Seidel. In this context, we introduce the student to the basics of Krylov subspaces. Another aspect of Big Data applications is optimization, multivariate max/min problems. This is also an important part of our development. We include both the *greatest descent* and *Hessian* variants.

We present the techniques of numerical analysis together with the supporting theory. Much of this book is organized in traditional definition, theorem, proof format. With a small number of exceptions, the theory is self-contained within this text and prerequisites. In the exceptional cases, the necessary supporting material is identified and referenced. But, full understanding of the material requires both knowledge of the mathematical foundations and hands on experience. In this regard, we include examples and exercises that are beyond what can be done with pencil and paper. In this way, we emphasize the natural link between numerical analysis and computing.

We present the material of the text in the given order. It is intended to be a one semester course. Programming in *Mathematica* is sufficiently intuitive for students with math or engineering background that little or no programming background is necessary. In fact, the reader's experience manually solving problems provides the necessary foundation toward programming sophisticated processes. At the same time, we find it useful to schedule computer lab time for the first few weeks of the semester. In this facility, we can provide one-on-one programming support. By mid-semester all students are on an even footing programming-wise.

For most students this class is a rewarding experience. In this setting, they are able to solve problems that are realistic in scale and complexity. To this point, their prior experience is often restricted to problems that can be executed easily with pencil and paper.

In many respects a course based on this text would function as a capstone course for the undergraduate. This is because it calls upon so much of a students' undergraduate experience, calculus, single and multivariate, vector calculus, linear algebra, real analysis, ODE and PDE and probability.

This text arose from our need of a beginning numerical analysis text that was sufficiently mathematical for our students and supported *Mathematica* as the programming platform. During that time, our students have seen topics come and go as we settled on a stable course. Without their participation this text could not have been written. Special acknowledgement goes to those who helped me understand how to present this material. In particular, this includes Scott Irwin, Yevgeniy Milman, Andrew Hofstrand, Evan Curcio, Gregory Javens and James Kluz.

Special acknowledgment goes to those who helped me understand how to present this material. In particular, this includes Scott Irwin, Yevgeniy Milman, Andrew Hofstrand, Evan Curcio, Gregory Javens, Hassan Mahmood and James Kluz.

About the Cover: Carl Friedrich Gauss was a giant among mathematicians. There are several signature procedures attributed to Gauss that arise in the text. For instance, Gaussian quadrature is often considered the *gold standard* for numerical integration. In general, problems stated here are resolved as systems of linear equations. Moreover, the basic techniques we use today are attributed to Gauss. Hence, we may think of Gauss as the central figure in the development of this topic.

John Loustau
Hunter College (CUNY)
New York, 2017

Contents

Preface vii

1. Beginnings 1

   1.1 The Programming Basics for *Mathematica* 2
   1.2 Errors in Computation 6
   1.3 Newton's Method 8
   1.4 Secant Method 14

2. Linear Systems and Optimization 19

   2.1 Linear Systems of Equations 20
   2.2 The Norm and Spectral Radius of a Linear Transformation 27
   2.3 Large Matrix Techniques 32
   2.4 Functions of Several Variables: Finding Roots and Extrema 37

3. Interpolating and Fitting 43

   3.1 Polynomial Interpolation 44
   3.2 Bezier Interpolation 53
   3.3 Least Squares Fitting 56
   3.4 Cubic Splines and B-Splines 62
   3.5 Hermite Interpolation 71

4. Numerical Differentiation 75

   4.1 Finite Differences and Vector Fields 76
   4.2 Finite Difference Method, Explicit or Forward Euler . . . 80
   4.3 Neumann Stability Analysis 85
   4.4 Finite Difference Method, Implicit and Crank Nicolson . . 91

5. Numerical Integration 97

5.1 Trapezoid Method and Simpson's Rule 98
5.2 Midpoint Method 101
5.3 Gaussian Quadrature 105
5.4 Comments on Numerical Integration 110

6. Numerical Ordinary Differential Equations 117

6.1 First order ODE Techniques, Forward Euler and Corrector Method 119
6.2 Midpoint Method with an Application to Mathematical Oncology 122
6.3 Shooting Method with an Application to Cooling Fins . . 126
6.4 The Method of Lines, Revisiting the Heat Equation 129

7. Monte Carlo Method 133

7.1 Basics of Probability Theory, Terminology and Notation . 135
7.2 Generating Distributed Number Sequences 137
7.3 Monte Carlo Method for Estimating the Integrals and n-Dimensional Volumes 141
7.4 Monte Carlo Method for Problems with Stochastic Elements 143

Bibliography 147

Index 149

Chapter 1

Beginnings

Introduction

This chapter provides a brief introduction to programming for those who have never programmed. For those with programming experience, this is the introduction to *Mathematica*. One of the several advantages to using a 4GL such as *Mathematica* is that it makes numerical methods accessible to all students with multivariate calculus and linear algebra. Indeed, most students catch on very quickly to programming in *Mathematica* and are doing complicated programs well before the end of the semester course. To support the learning process, there are tutorials available by selecting *Help* from the system menu and then *Documentation Center*. The first item available in documentation contains the basic tutorials. For those who prefer hard copy references, there are several textbooks available from booksellers.

A second feature of this chapter is to introduce the reader to the quirks of *Mathematica*. *Mathematica* was originally developed by mathematicians, and therefore it has a mathematician's point of view. If your background is with C++ or another of the computer scientist developed programming products, you will find *Mathematica* to be similar on the surface but significantly different at lower levels. If you have never programmed in an interpreted 4GL, then you have something to get used to.

In this chapter, we introduce the terminology associated to computer error. Computers must represent decimal values in a finite number of significant digits. Therefore, the representation is often only close to the actual value. For instance $1/3$ is not 0.3333333. The error inherent in the representation can be magnified during normal arithmetic operations. In extreme cases, this may yield ridiculous results. When using any computer system you must always be cognizant of the potential of error in your calculations. We will see an example of this in Section 1.2.

We next look at Newton's method. Most calculus courses include Newton's method for finding roots of differentiable functions. If you have solved a Newton's method problem with pencil and paper, you know that doing two or three iterations of the process is a nightmare. Even the simplest cases are not the sort of thing most students want to do. Now, we see that it is easy to program. In this regard, it is an excellent problem for the beginning student. In addition, *Mathematica* provides a built in function that performs Newton's method. It is empowering for the student to compare his results to the output produced by *Mathematica*. We follow Newton's method by the secant method to find the root of a function. This provides the student with the first example of an error estimating procedure.

By the end of the chapter, the student should be able to program the basic arithmetic operations, access the standard mathematical functions, program loops and execute conditional statements (if ... then ... else ...). A special feature of *Mathematica* is the graphics engine. With minimal effort, the student can display sophisticated graphical output. By the end of this chapter the student will be able to use the basic 2D graphics commands.

1.1 The Programming Basics for *Mathematica*

We use *Mathematica* Version 10. Each year when the university renews its license, the version changes. In the past, programs for one version are either fully upgradeable to the subsequent version or Wolfram provides a program that upgrades program code written for one version to the next.

During this semester you will be programming in *Mathematica*. To begin with, you will learn to be able to program the following.

(1) The basic arithmetic operations (addition, subtraction, multiplication, division, exponentiation and roots)
(2) Define a function
(3) Loops (Do Loop or While Loop)
(4) Conditionals (If ... then ... else)
(5) Basic graphics (point plot, function plot, parametric plot)

We begin with item 1. You add two numbers $3 + 5$, subtract $3 - 5$, multiply $3 * 5$ and divide $3/5$. Alternatively, if $a = 3$ and $b = 5$, then $a + b$, $a - b$, $a * b$ and a/b have exactly the same result. For exponents, $a^b = 243$ and $b^a = 125$. The usual math functions cosine, Cos[x], sine,