



An Introduction to

Numerical Algorithms

Derek O'Connor
University College Dublin

Typeset by the author in plain T_EX using Pallotino & Computer Modern fonts

NUMERICAL ALGORITHMS

Derek O'Connor

University College, Dublin

October 8, 2004

PREFACE

Computers are useless. They can only give you answers.

Pablo Picasso (1881-1973)

Ask yourself the question : Would I fly an airplane that uses software written by me?

Anon.

This is a very basic introduction to the construction, analysis, and testing of numerical algorithms on digital computers.

There are three theoretical ideas that are important in the construction of these algorithms :

- Approximation (Taylor's Theorem, etc.)
- Iteration (Successive Approximation)
- Transformation (Matrix Factorizations, etc.)

The important technical ideas are :

- Finite Space (Finite precision arithmetic)
- Finite Time (Speed of the algorithm)
- Convergence Testing (When to stop an algorithm)
- Error Trapping (No algorithm is perfect)

The technical ideas could be crudely summarized as : how to compute with floating point arithmetic.

No matter how good the construction and analysis of an algorithm is, it must be implemented in some programming language and then tested on a real computer.

Students need to experiment with algorithms and numbers using MATLAB and MAPLE . Many of the great algorithmic nuggets have been found through numerical experiments, later polished by theory.

If, having studied these notes and carried out the exercises and assignments :

- you feel ignorant of numerical algorithms, then one purpose of these notes has been achieved.
- you feel you could not earn a living writing numerical algorithms, then another purpose of these notes has been achieved.
- you feel you could earn a living using numerical algorithms, then yet another purpose of these notes has been achieved.

CONTENTS

CHAPTER 1. INTRODUCTION

- 1.1 Introduction
- 1.2 Standard Numerical Problems
- 1.3 General Numerical Algorithms
- 1.4 Some Typical Problem Areas
- 1.5 Numerical Software
- 1.6 Historical Notes

CHAPTER 2. COMPUTER NUMBERS & ARITHMETIC ■

- 2.1 Introduction
- 2.2 Mathematical Number Systems
- 2.3 Counting & Number Representation
- 2.4 Computer Number Systems
- 2.5 Floating Point Numbers
- 2.6 IEEE Arithmetic
- 2.7 Perturbation Analysis
- 2.8 Notes

CHAPTER 3. MATHEMATICAL REVIEW

- 3.1 Introduction
- 3.2 Sets and Boolean Algebra
- 3.3 Sequences, Series, and Limits
- 3.4 Functions
- 3.5 Calculus
 - 3.5.1 Derivatives
 - 3.5.2 Integrals
 - 3.5.3 Mean Value Theorems
- 3.6 Linear Spaces
- 3.7 Normed Linear Spaces
 - 3.7.1 Inner Product Spaces
 - 3.7.2 Sequences in Normed Linear Spaces
 - 3.7.3 Mappings
 - 3.7.4 Contraction Mappings
 - 3.7.5 Successive Approximations

CHAPTER 4. NON-LINEAR EQUATIONS

- 4.1 Introduction
- 4.2 Order of Convergence
- 4.3 Algorithms for Non-Linear Equation
 - 4.3.1 Converting to Fixed-Point Form
 - 4.3.2 The Bisection Algorithm
 - 4.3.3 The Newton Algorithm
 - 4.3.4 The Secant Algorithm
 - 4.3.5 Higher Order Algorithms
 - 4.3.6 Comparison of Algorithms
 - 4.3.7 Poly-Algorithms
- 4.4 Stopping Rules
- 4.5 Implementation Notes
- 4.6 The Zeros of Polynomials
 - 4.6.1 Algorithms for Polynomials
- 4.7 Notes

CHAPTER 5. NUMERICAL LINEAR ALGEBRA

- 5.1 Introduction
- 5.2 Mathematical Basis
 - 5.2.1 Matrix Algebra
 - 5.2.2 Matrix Analysis
 - 5.2.3 Matrix Factorizations
 - 5.2.4 Basic Linear Algebra Algorithms
- 5.3 Direct Methods
 - 5.3.1 Gaussian Elimination
 - 5.3.2 Pivoting
 - 5.3.3 LUP Decomposition
 - 5.3.4 Inverses and Determinants
 - 5.3.5 Special Matrices
 - 5.3.6 Iterative Refinement
 - 5.3.7 Iterative Methods
- 5.4 Sparse Matrices

APPENDIX I. AN ALGORITHMIC LANGUAGE

APPENDIX II. A SYNOPSIS OF MATLAB

APPENDIX II. ANALYSIS OF ALGORITHMS