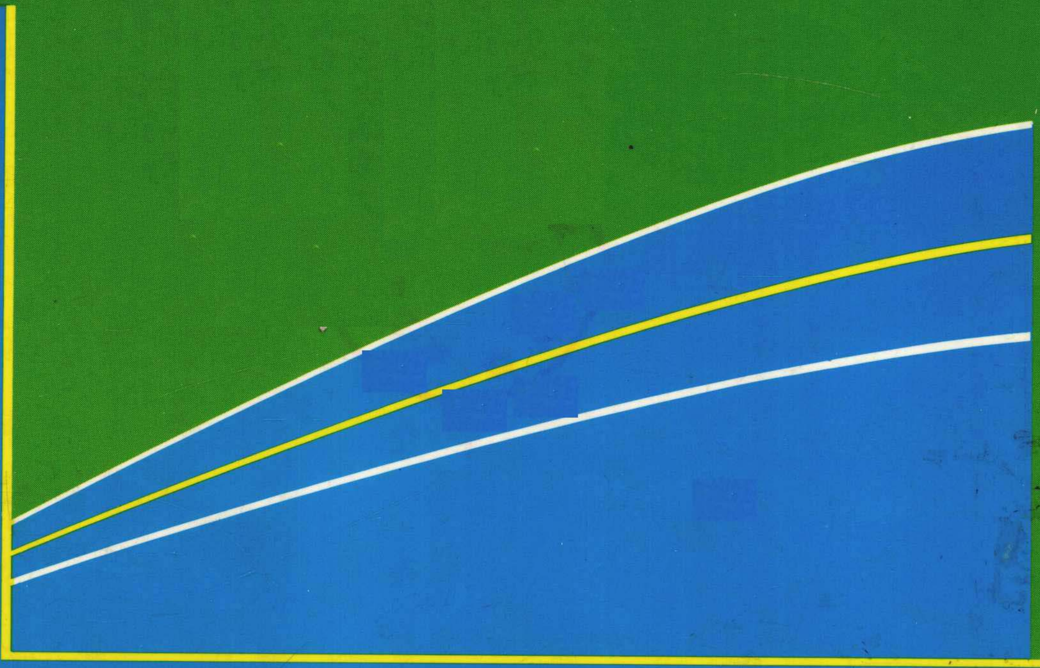
NUMERICAL ANALYSIS

An Introduction



Lars Eldén · Linde Wittmeyer-Koch

NUMERICAL ANALYSIS

An Introduction

Lars Eldén
Linde Wittmeyer-Koch

Department of Mathematics
Linköping University
Linköping, Sweden

ACADEMIC PRESS, I

Harcourt Brace Jovanovich, Publish

Boston San Diego New York
London Sydney Tokyo Toronto

This book is printed on acid-free paper. ♾

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24-28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data
Eldén, Lars, date
[Numerisk analys. English]
Numerical analysis : an introduction / Lars Eldén, Linde Wittmeyer
-Koch.
p. cm.
Rev. translation of: Numerisk analys.
Includes bibliographical references.
ISBN 0-12-236430-9 (alk. paper)
1. Numerical analysis. I. Wittmeyer-Koch, Linde, 1939-
II. Title.
QA297.E4213 1990 89-71379
519.4—dc20 CIP

Printed in the United States of America

90 91 92 93 9 8 7 6 5 4 3 2 1

Preface

This book is intended as a textbook for an introductory course in numerical analysis. The level is suitable for the advanced undergraduate. We believe that the book is accessible to a rather wide audience, since most of the mathematics prerequisite should be covered in first-year calculus and algebra courses.

The aim of this book is to give an introduction to some of the basic ideas in numerical analysis. We have chosen to cover a comparatively wide range of material, including classical algorithms for the solution of nonlinear equations and for linear systems, methods for interpolation, integration and approximation. We have also tried to give an introduction to some areas, which we think are important in the applications that a science or engineering student will meet. These areas include computer arithmetic (floating point) and standard functions, splines and finite elements. In a few areas (approximation and linear systems of equations) we have chosen to give a somewhat more comprehensive presentation.

In our opinion a broad introductory course should emphasize the understanding of methods and algorithms. In order to really grasp what is involved in numerical computations, the student must first perform computations using a simple calculator. We supply a number of exercises intended for this. To further help the student in the learning process we recommend the use of high level programming systems (like Matlab or Mathematica) for more realistic computer assignments than we have given in this book.

This book is a revision of a corresponding book earlier published in Swedish. We are grateful to many of our colleagues at the Department of Mathematics, Linköping University, for suggesting several improvements, and for correcting errors. In particular we would like to mention Tommy Elfving, Jan Eriksson, and Ulla Ouchterlony. Ingegerd Skoglund has constructed most of the TEX macros we have used.

Linköping, January 1990

Lars Eldén
Linde Wittmeyer-Koch

Contents

1 Introduction

1.1 Mathematical Models and Numerical Approximations

Mathematical models are basic tools in scientific problem solving. Typically, some fundamental natural laws are used to derive one or several equations, which model the process that is being studied. Through the mathematical treatment of the equations, answers can be found to questions that are posed in connection with the problem area. This is illustrated schematically in Figure 1.1.1.

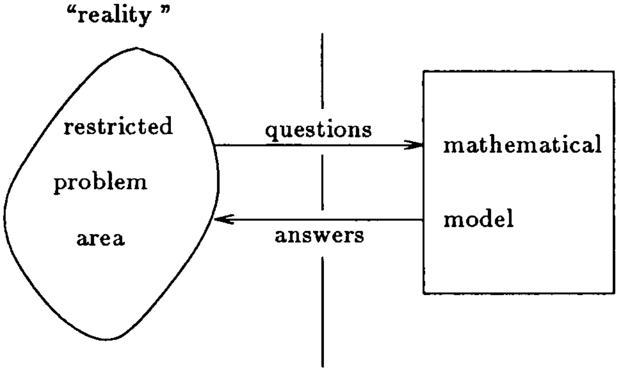


Figure 1.1.1

It is important to remember that in many cases the problem area described by a mathematical model is very narrow. Further, there are often simplifications in the assumptions. Therefore, the mathematical model is not an exact description of reality, and the answers that it produces must be checked and compared with experimental results.

We shall illustrate the concept of a mathematical model using an example from structural mechanics. It is not our aim to give a description that is comprehensive from the point of view of mechanics. Nor will the mathematical aspects be treated in detail. Rather, we shall use a relatively simple example to show that mathematical analysis is not enough to answer the questions that are relevant from the viewpoint of an application.

Consider a beam of length 1, rigidly built-in at both ends.

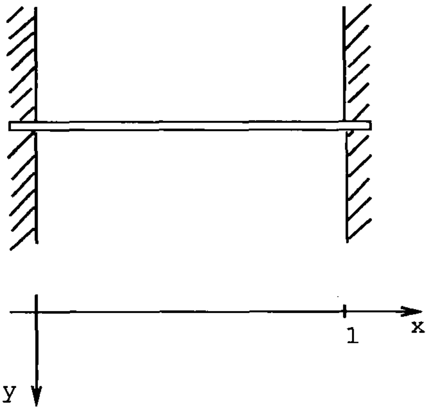


Figure 1.1.2

The following question is interesting: How much will the beam deflect under a certain load?

It is also important to know bending moments, shear forces, etc. Information about these can be found as byproducts when the question about the deflection is answered.

We introduce a coordinate system according to the figure, and let $y(x)$ denote the deflection of the beam under a certain load. If the load is $q(x)$, not necessarily constant over the length of the beam, then (under certain assumptions) $y(x)$ satisfies the differential equation

$$\frac{d^2}{dx^2}\big(EI(x)\,\frac{d^2y}{dx^2}\big) = q(x), \qquad 0 < x < 1, \tag{1.1.3a}$$

with boundary conditions

$$y(0) = y'(0) = y(1) = y'(1) = 0; \tag{1.1.3b}$$

E is a material constant (Young's modulus of elasticity) and $I(x)$ is the moment of inertia of the cross-section of the beam. Note how the boundary conditions describe the fact that the beam is rigidly built-in at both ends.

(1.1.3a) is called the beam equation and (1.1.3) is an example of a **boundary value problem** for a fourth-order differential equation. If we can solve (1.1.3), i.e., determine the function $y(x)$ that satisfies (1.1.3), then we have answered the question about the deflection of the beam.

In some cases, it is easy to solve the boundary value problem analytically. Assume, e.g., that the coefficients and the load are constant

$$I(x) = I_0, \quad q(x) = q_0,$$

and put

$$Q = \frac{q_0}{EI_0}.$$

We immediately see that

$$y(x) = \frac{Q}{24}x^4 + Ax^3 + Bx^2 + Cx + D,$$

satisfies the differential equation (1.1.3a) for arbitrary A, B, C and D. From the boundary conditions we can determine A, B, C and D and get

$$y(x) = \frac{Q}{24}(x^4 - 2x^3 + x^2) = \frac{q_0}{24EI_0}(x^4 - 2x^3 + x^2). \qquad (1.1.4)$$

With the analytical solution (1.1.4), we can determine the deflection of the beam for any value of x and for any value of the load q_0.

The problem is slightly more complicated if the load $q(x)$ is not constant, but it can still be solved analytically, if we can determine successive primitive functions of primitive functions of q.

The problem becomes considerably more complicated if we assume that the beam rests on an elastic foundation.

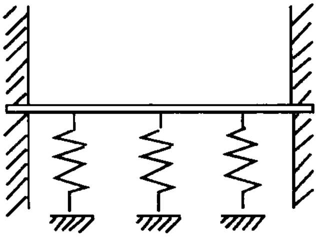


Figure 1.1.5

The deflection of the beam now satisfies the equation

$$\frac{d^2}{dx^2}\big(EI(x)\,\frac{d^2y}{dx^2}\big) + k(x)y = q(x), \tag{1.1.6}$$

with boundary values (1.1.3b). $k(x)$ is the foundation spring function and is assumed to be a continuous function of x.

Using mathematical analysis, one can prove theorems about the existence and uniqueness of solutions of (1.1.6), but in general no explicit solution can be found. An explicit solution is a formula, where for each set of values of the relevant parameters and for each x we can easily compute the corresponding function value $y(x)$.

Thus, we have a mathematical model which is a good description of the problem under study, but it does not give us a direct answer to our question. Now there are essentially two alternatives: either we can make simplifications in the model (e.g., assume that the coefficients are constant) so that an analytical solution can be determined, or we can introduce **numerical approximations** in the equation. In many cases, the first alternative is inadequate, as the model may become so bad that the answers are no longer reliable. Errors are introduced in the second alternative also, but here the answers are more reliable since it is possible to estimate how the approximations affect the accuracy of the solution. The two alternatives are illustrated in Figure 1.1.7.

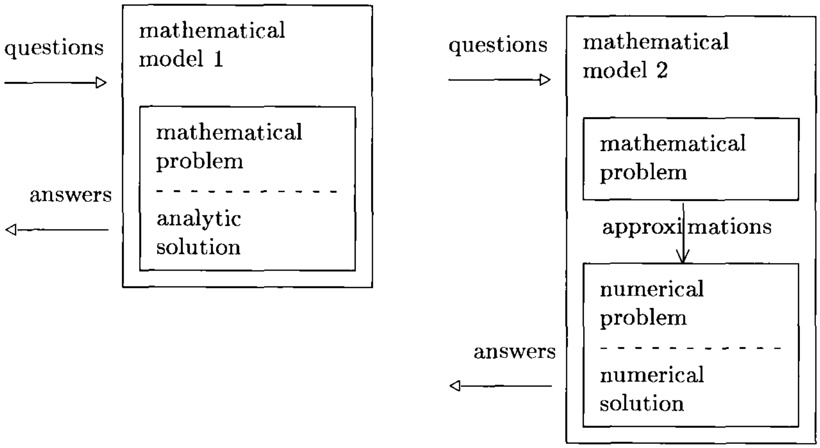


Figure 1.1.7

We will now sketch how numerical approximations can be made in the differential equation (1.1.6). For a detailed description, see Chapter 10.

For many purposes it is sufficient to compute approximations of the solution at a finite number of points $(x_i)_{i=1}^n$ in the interval $[0, 1]$. We approximate the derivatives in the differential equation (1.1.6) and the boundary conditions (1.1.3b) by difference quotients, e.g.,

$$y''(x_i) \approx \frac{y(x_{i+1}) - 2y(x_i) + y(x_{i-1})}{h^2};$$

here we have assumed that the partitioning of the interval is equidistant, i.e., $x_{i+1} - x_i = h$. In this way, the differential equation (1.1.6) is replaced by a system of linear equations

$$Ay = q; \tag{1.1.8}$$

the components of the vector y are approximations of the function values

$$y_i \approx y(x_i), \qquad i = 1, 2, \ldots, n;$$

the elements of the matrix A depend on the coefficients in (1.1.6) (the material properties), and the right hand side q is a vector of values of the function $q(x)$ (the load).

The linear system of equations (1.1.8) is called a **discretization** of the differential equation (1.1.6) with boundary conditions (1.1.3b). The system of equations can be solved using Gaussian elimination.

Thus, we have replaced the mathematical problem of solving a differential equation by the numerical problem of solving a linear system of equations. We will now make the notion of a numerical problem somewhat more precise:

A **numerical problem** is a clear and unambiguous description of the functional connection between **input data**, i.e., the "independent variables" of the problem, and **output data**, i.e., the desired results. Input data and output data consist of a *finite number* of real quantities.

It is obvious that the problem of solving (1.1.8) can be considered as a numerical problem. Input data are the coefficients of the matrix A and the right hand side q; output data are the components of the vector y.

We stated that (1.1.8) can be solved using Gaussian elimination. That is an example of an algorithm.

An **algorithm** for a numerical problem is a complete description of a finite number of well-defined operations, through which each permissible input data vector is transformed into an output data vector.

By operations, here we mean arithmetic and logical operations and previously defined algorithms. An algorithm can be described loosely or in great detail. A comprehensive description is obtained when an algorithm is formulated using a programming language.

The objective of **numerical analysis** is to construct and analyze numerical methods and algorithms for the solution of problems in science and technology. In connection with the above discussion, we give here a few examples of interesting questions in numerical analysis:

- How large is the discretization error when the boundary value problem is approximated by (1.1.8), i.e., how large are the errors $y_i - y(x_i)$?
- How long does it take to solve (1.1.8) using a certain computer?
- How do the rounding errors of the computer arithmetic influence the accuracy of the solution?

Exercises

1. Derive (1.1.4) and sketch the curve for some value of Q. What is the maximal deflection of the beam?

2. Find the chapter on beams in a textbook in structural mechanics and write the beam equation using the notation of that book.

3. What are the boundary conditions for (1.1.3a) if the beam is freely supported at both ends. Determine an analytical solution for this case (assume constant coefficients).

References

The definitions of a numerical problem and algorithm are taken from

G. Dahlquist and Å. Björck, *Numerical Methods*, Prentice–Hall, Englewood Cliffs, New Jersey, 1974.

This textbook is recommended for a more extensive course in numerical analysis.

2 Error Analysis and Computer Arithmetic

In Chapter 1, we illustrated how approximations are introduced in the solution of mathematical problems that cannot be solved exactly. One of the most important tasks in **numerical analysis** is to estimate the accuracy of the result of a numerical computation. In this chapter, we discuss different sources of error that affect the computed result and we derive methods for error estimation. In particular, we describe some properties of computer arithmetic. A standard for floating point arithmetic was adopted by IEEE in 1985. We give a brief presentation of the standard and its implementation in a floating point processor.

2.1 Sources of Error

There are essentially three types of errors that affect the result of a numerical computation:

1. **Errors in the input data** are often inevitable. The input data can be the result of measurements with a limited accuracy, or real numbers, which must be represented with a fixed number of digits.
2. **Rounding errors** arise when computations are performed using a fixed number of digits in the operands.
3. **Truncation errors** arise when "an infinite process is replaced by a finite one", e.g., when an infinite series is approximated by a partial sum, or a function is approximated by a polynomial.

Truncation errors will be discussed in connection with the different numerical methods. In this chapter, we study the other two sources of error.

The different types of errors are illustrated in Figure 2.1.1, which relates to the discussion in Chapter 1.

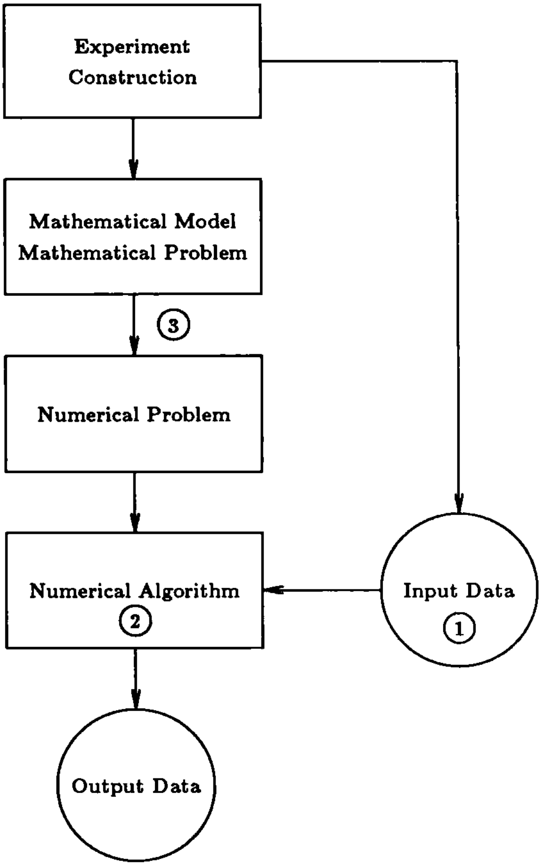


Figure 2.1.1 Sources of error.

The following notation will be used:

R_X error in the result coming from input data errors,
R_{XF} error in the result coming from errors in the function values used,
R_B rounding error,
R_T truncation error.

R_{XF} is a special case of R_X.