NUMERICAL ANALYSIS

K.D. Anderson
J.S.C. Prentice
C.M. Villet

Last updated: 21 October 2015

Disclaimer: Links are followed at own risk when viewing this document electronically.

Contents

Chapter 1

INTRODUCTION

1.1 Numerical methods and analysis

When solving a mathematical problem, such as determining a definite integral or solving a differential equation, we attempt to do so *analytically*—we determine an expression for the indefinite integral and then substitute the limits, or we apply an appropriate technique to find an expression that relates the dependent and independent variables of the DE. The precise definition of an analytic expression or solution is one that can be expressed in terms of a bounded number of certain elementary functions: constants (including complex numbers), one variable x, elementary operations of arithmetic $(+ - \times \div)$, n-th roots, exponents (which includes trigonometric functions and inverse trigonometric functions) and logarithms. However, we are often confronted with mathematically posed problems that simply cannot be solved analytically, such as the transcendental equation

$$\sin x - 0.625x = 0 \tag{1.1}$$

or the non-linear differential equation

$$\frac{dN}{dt} = aN - k(t)N^{1.7} \tag{1.2}$$

both of which we shall revisit in later chapters. Problems which cannot be solved analytically are generally nonlinear in nature, which is clearly the case with these two equations.

The only hope we have of dealing with such problems is by finding a *numerical solution*. In equation (1.1) this would be a numerical value for x that satisfies the equation; in equation (1.2) the numerical solution is a set of numbers that represents the true solution over the relevant interval of integration. *Numerical methods* are the mathematical tools we use to find such numerical solutions.

1.1.1 Numerical analysis.

As the term suggests, *numerical analysis* is the mathematical study of numerical methods and, in the broader sense, the analysis of the field of numerical methods as a whole. Numerical analysis addresses the following:

(a) The derivation of numerical methods from fundamental mathematical ideas.

(b) Investigating the properties of numerical methods, such as accuracy and stability.

Numerical methods tend to be approximation techniques, i.e. they yield approximate solutions rather than exact solutions. Through appropriate analysis of the method, we may understand the nature of the *approximation error* and we will probably be able to successfully implement the method subject to a desired level of accuracy. If, through appropriate analysis of Consequently, the analysis of numerical methods is an extremely important part of the field.

1.1.2 Types of methods.

In these notes, we study the following methods:

(a) Nonlinear algebraic equations in one variable - chapter 3 - we investigate the bisection method, linear interpolation, Newton's method, and fixed-point iteration.

(b) Systems of linear equations - chapter 4 - we investigate the Jacobi method that can be used to find numerical solutions to large systems of linear equations.

(c) Approximation of functions and data sets - chapter 5 - we investigate polynomial and Lagrange interpolation, least-squares polynomial fitting, and Chebyshev series.

(d) Numerical differentiation - chapter 6 - Approximations to derivatives of various order using Taylor series.

(e) Numerical integration (quadrature) - chapter 7 - we study the Trapezium and Simpson methods, based on Lagrange interpolation, used to approximate definite integrals.

(f) Initial-value problems - chapter 8 - Euler's method, modified Euler method, Runge-Kutta methods of orders two and four, used to approximate initial-value problems arising from ordinary differential equations.

We will devote our efforts to the derivation and error analysis of most of these methods, and we will demonstrate the methods by means of numerical examples. Consequently, to a large extent, these notes are theoretical in nature.

1.2 Number representations and round-off errors

Numbers play an important part of numerical analysis (and mathematics in general with a whole field dedicated to their study). We normally think of numbers as constant-valued entities and use the numerals $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, and combinations thereof, to represent them, e.g. 2, 42, 5345, etcetera. A k-digit number is represented by using k numerals. It should be clear the k must be a natural number, and we refer to the number as a finite digit number. It is also quite possible for a number to be a ∞-digit number, we simply refer to such a number as an infinitely digit number. The natural numbers, integers, and rational numbers may all be represented as finite digit numbers while the irrational and real numbers are mostly infinite digit numbers. For example, 1 is a 1-digit number, 12 is a 2-digit number and 87543 is a 5-digit number.

1.2.1 Decimal number representation

In the study of numerical analysis, we might want to make use of a finite precision device, e.g. a calculator or computer, to perform computations. Finite precision devices tend to have limited memory and might only be able to store and represent finite digit numbers quite accurately. Due to this memory constraint, there is a very specific way of representing numbers and using these numbers in computations within a finite precision environment.

Consider the mathematical constant π, which we know to be an irrational number and may be represented as the fraction $\frac{22}{7}$. A finite precision device cannot interpret the meaning of the symbol π and we rather make use of $\frac{22}{7}$ in computations.

We know that π can also be written as

$$3.14159265358979323846264338327950288419716939937510\ldots$$

We call this representation of the number the *decimal representation* or *decimal form* of the number. As another example, we know that we can represent $\frac{1}{2}$ as 0.5. Most real numbers are represented this way.

We define the *normalised floating-point form* of a number to be

$$\pm\, 0.d_1d_2\ldots d_k \times 10^n, \tag{1.3}$$

where d_i, called a *decimal digit*, is an integer-valued number with $1 \leq d_1 \leq 9$ and $0 \leq d_i \leq 9$, for each $i = 2, 3, \ldots, k$. We call any number represented by (1.3) a *k-digit decimal number*. Thus, π in normalised decimal floating-point form is

$$0.3141592653589793238462643383279502884197169399375 10\ldots$$

and the normalised floating-point form of $\frac{1}{8}$, i.e. 0.125, which is a 3-digit decimal number.

Consider a positive real number y with the normalised floating-point form

$$y = 0.d_1d_2\ldots d_kd_{k+1}d_{k+2}\ldots \times 10^n.$$

The finite floating-point form of y, which we denote $fl(y)$, is obtained by terminating y after k decimal digits. This termination is done in one of two ways. The first way, called *truncation* or *chopping*, is done by simply getting rid of the digits $d_{k+1}d_{k+2}\ldots$, which produces

$$fl(y) = 0.d_1d_2\ldots d_k \times 10^n.$$

The second way, called *rounding*, is done by adding $5\times 10^{n-(k+1)}$ to y and then chopping the result to obtain the floating-point form

$$fl(y) = 0.\delta_1\delta_2\ldots\delta_k \times 10^n.$$

When rounding, if $d_{k+1} \geq 5$, we add 1 to d_k to obtain $fl(y)$, this is called *rounding up*; if $d_{k+1} < 5$, we simply chop off all the digits after the first k digits, this is called *rounding down*. It should be clear that rounding down yields $\delta_i = d_i$ for each $i = 1, 2, 3, \ldots, k$, but this is not necessarily the case when rounding up.

1.2.2 Error

Round-off error refers to errors in representing numbers in a finite precision environment. For example, we know that

$$\pi = 3.14159265358979323846264338327950288419716939937510...$$

but in most desktop computers we find that

$$\pi = 3.14159265358979.$$

In other words, due to memory constraints as mentioned earlier, the value of π is rounded off to 14 decimal places. In a finite precision device this round-off process is applied to all numerical values, so that round-off error is present most of the time (an obvious exception is the integers, which do not have a fractional part). The digits lost in this rounding-off process constitute the round-off error. By and large, we do not expect round-off error to significantly compromise calculations performed on a computer (round-off error is typically of the order of 10^{-15}), although sometimes round-off errors may be amplified in the course of a computation. We generally find that the approximation errors mentioned previously are far more significant than round-off error, and so we restrict our study of error to approximation errors rather than round-off errors in these notes. Indeed, it is appropriate to regard approximation errors as a consequence of the mathematical nature of the numerical method itself, whereas round-off errors may be seen as a technological limitation—the price we pay for using finite-precision devices.

Chapter 2

SERIES EXPANSIONS

2.1 Introduction

Sequences and series play an important role in numerical analysis. For example, Newton's method (see §3.4) generates a sequence of approximations to the root of a non-linear equation.

We review the theory of sequences and series and describe a useful power series that may be used to approximate functions about a given point.

2.2 Sequences

A *sequence of real numbers* is a function from the natural numbers $\mathbb{N} = \{1, 2, \dots\}$ onto the real numbers $\mathbb{R}$, i.e.

$$\begin{aligned} 1 &\mapsto x_1 \\ 2 &\mapsto x_2 \\ 3 &\mapsto x_3 \\ 4 &\mapsto x_4 \\ &\vdots \\ n &\mapsto x_n \\ &\vdots \end{aligned}$$

Thus $f(n) = x_n$. The numbers $x_1, x_2, x_3, \dots, x_n, \dots$, in the range of the function, are called the *elements* or *terms* of the sequence. A sequence is called *infinite* if it has an infinite amount of terms, otherwise it is called *finite*.

Example 2.1. Consider the following

$$\begin{aligned} 1 &\mapsto \sqrt{1} \\ 2 &\mapsto \sqrt{2} \\ 3 &\mapsto \sqrt{3} \\ 4 &\mapsto \sqrt{4} \\ &\vdots \end{aligned}$$

This is indeed a sequence, because the numbers on the left are in $\mathbb{N}$ and the numbers on the right are in some subset of $\mathbb{R}$. We note that the sequence is generated by the function $f(n) = \sqrt{n}$.

A sequence is denoted by its elements $x_1, x_2, x_3, \ldots$, or using the shorter notation $(x_n)_{n=1}^{\infty}$, where $n \in \mathbb{N}$. This latter notation is simplified to (x_n) if clear from context that we are dealing with an infinite or a finite sequence. Parentheses are deliberately used to emphasize the importance of ordering in a sequence. If we consider the set $\{1, 2, 3, 4\}$ and rewrite it as $\{4, 3, 2, 1\}$, then it is still considered to be the same set. However, when we consider the sequence $(1, 2, 3, 4)$ and rewrite it as $(4, 3, 2, 1)$, then the two sequences are considered to be two different. *Why are these two sequences considered to be different from each other?*

Most often a sequence is defined by giving a formula for its nth term x_n. For example, consider the sequence of reciprocals of the odd numbers

$$\left(1, \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \ldots\right).$$

This sequence may be written as

$$\left(\frac{1}{2n+1} : n \in \mathbb{N}\right)$$

or more simply

$$x_n = \frac{1}{2n+1},$$

where it is understood that $n \in \mathbb{N}$. Another way of defining a sequence is to specify the value for x_1 and giving a formula for x_{n+1} in terms of x_n, the sequence is then said to be defined *recursively* or *inductively*.

Example 2.2. Consider the Fibonacci sequence $F = (f_n)$ given by

$$1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ \ldots$$

If we specify $f_1 = 1$ and $f_2 = 1$, then we may give the recursion formula for every other term in the sequence as $f_{n+1} = f_n + f_{n-1}$, where $n \geq 2$.

The limit of a sequence (x_n) is the real number ℓ such that

$$\lim_{n\to\infty} x_n = \ell \tag{2.1}$$

A sequence (x_n) is called *convergent* if its limit ℓ exists; if this limit does not exist, then the sequence is called *divergent*. It should be noted that the limit of a sequence is unique. Given two sequences (x_n) and (y_n) such that $\lim_{n\to\infty} x_n = \ell_1$ and $\lim_{n\to\infty} y_n = \ell_2$, then

$$\begin{aligned}
\lim_{n\to\infty} (x_n \pm y_n) &= \ell_1 \pm \ell_2 \\
\lim_{n\to\infty} (cx_n) &= c\ell_1, \quad c \in \mathbb{R} \\
\lim_{n\to\infty} (x_n y_n) &= \ell_1 \ell_2 \\
\lim_{n\to\infty} \left(\frac{x_n}{y_n}\right) &= \frac{\ell_1}{\ell_2}, \quad y_n \neq 0 \text{ and } \ell_2 \neq 0 \\
\lim_{n\to\infty} (x_n)^{(y_n)} &= x^y, \quad x > 0 \text{ and } x_n > 0
\end{aligned}$$

There are numerous other useful limit theorems and the reader is referred to consult an analysis text.

Example 2.3. Consider the sequence $\left(\frac{1}{n}\right)$. If $n \to \infty$ then $\frac{1}{n} \to 0$ and thus

$$\lim_{n\to\infty} \left(\frac{1}{n}\right) = 0.$$

Hence 0 is the limit of the sequence.

A sequence (x_n) is *bounded* if there exists numbers $a, b \in \mathbb{R}$ such that $a < x_n < b$ for all $n \in \mathbb{N}$. The number a is called the lower bound of the sequence and the number b is called the upper bound of the sequence.

Example 2.4. Consider the sequence of even, positive integers $(2, 4, 6, 8, \dots)$. It should not be difficult to see that the number 1 is a lower bound of this sequence. Next we consider the sequence of reciprocals of the even, positive integers $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}, \dots\right)$. In this case, the number 1 is the upper bound of the sequence.

A sequence (x_n) is called *increasing* if

$$x_1 \leq x_2 \leq x_3 \leq \cdots \leq x_n \leq x_m \leq \cdots$$

for all $n < m$ and $n, m \in \mathbb{N}$. Similarly, the sequence (x_n) is called *decreasing* if

$$x_1 \geq x_2 \geq x_3 \geq \cdots \geq x_n \geq x_m \geq \cdots$$

for all $n < m$ and $n, m \in \mathbb{N}$. A sequence (x_n) is called *monotone* if it is increasing or decreasing. It can be shown that a monotonic sequence is convergent if and only if it is bounded.

Example 2.5. A ball, with diameter ϵ, is dropped from a height h_0. Each time it drops h metres, it rebounds rh metres. Determine how many times the ball bounces before it stops moving.

Solution: We note that if the initial height was h_0, then after the first bounce the ball will reach a height of $h_1 = rh_0$, the height after the second bounce would be $h_2 = rh_1$, and after n bounces the height would be $h_n = rh_{n-1}$. But it follows that

$$h_n = rh_{n-1} = r\big(rh_{n-2}\big) = r^2 h_{n-2} = \cdots$$

and we conclude that $h_n = r^n h_0$. Clearly, the bounces of the ball form a recursive sequence.

The ball stops bouncing when $|h_n| \leq \epsilon$. Since $h_n = r^n h_0$, it follows that

$$\begin{aligned} r^n h_0 &\leq \epsilon \\ r^n &\leq \frac{\epsilon}{h_0} \\ n\ln(r) &\leq \ln\left(\frac{\epsilon}{h_0}\right) \\ \therefore \quad n &\geq \frac{\ln(\epsilon) - \ln(h_0)}{\ln(r)} \end{aligned}$$

If $h_0 = 8$m, $r = 0.7$ and $\epsilon = 0.07$m, then we find $n \geq 13.2858$ and we conclude that it would take 14 bounces before the balls stops.

2.3 Series

Informally, a series is the sum of the terms of a sequence. If we let (x_n) be a sequence, then sum of the first k terms of this sequence, i.e.

$$s_k = x_1 + x_2 + \cdots + x_k = \sum_{i=1}^{k} x_i, \quad k \leq n$$

is called the *kth partial sum* of the sequence. Note that the partial sums form a sequence by themselves, i.e.

$$\begin{aligned}
s_1 &= x_1 \\
s_2 &= s_1 + x_2 \quad (= x_1 + x_2) \\
s_3 &= s_2 + x_3 \quad (= x_1 + x_2 + x_3) \\
&\vdots \\
s_n &= s_{n-1} + x_n \quad (= x_1 + x_2 + x_3 + \cdots + x_{n-1} + x_n) \\
&\vdots
\end{aligned}$$

This pair of sequences $\big((x_n), (s_n)\big)$ is called the *series* generated by the sequence (x_n). The numbers x_i, $i = 1, 2, \ldots, n$, in the partial sums are called the *terms* of the sequence. A series is called *infinite* if it has an infinite amount of terms, otherwise it is called *finite* if it has a finite amount of terms.

Instead of writing $\big((x_n), (s_n)\big)$ every time to denote a series generated by (x_n), it is convention to use the notation

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + x_3 + \cdots + x_n$$

if dealing with a finite series; for an infinite series one would similarly have

$$\sum_{i=1}^{\infty} x_i = x_1 + x_2 + x_3 + \cdots .$$

The symbol Σ is the Greek letter sigma, and one often refers to the notation above as "sigma-notation". When clear from context whether we are dealing with a finite or infinite series, the sigma-notation is shortened to $\sum_i x_i$.

Let $\sum_{i=1}^{\infty} x_i$ be a series. If the sequence (s_n) of partial sums of this series converges to the limit s, then the series is called *convergent* and the limit s is called the *sum* of the series. The sum is denoted

$$s = \sum_{i=1}^{\infty} x_i$$

If this limit does not exist, then the series is said to be *divergent*.

> **Example 2.6.** Consider the series $\sum_{i=1}^{\infty} c$, where $c \in \mathbb{R}$ is a constant. Clearly this infinite series does not have a sum and thus is divergent. However, if we were to consider the finite series $\sum_{i=1}^{n} c$, then it should be clear that this series does have a sum and is convergent. The sum is none other than $s = nc$.

Two very important questions arise when studying series.

1. Does the series converge or diverge?
2. What is the sum of the series if it is convergent?

Example 2.7. Given the infinite series

$$\sum_{i=0}^{\infty} ar^i = a + ar + ar^2 + ar^3 + \cdots + ar^i + \cdots, \tag{2.2}$$

where $a \in \mathbb{R}$ is a non-zero constant.

Consider the nth partial sum $s_n = a + ar + ar^2 + \cdots + ar^n$. If we multiply s_n by r and subtract the result from s_n, then we obtain

$$\begin{aligned} s_n - rs_n &= (a + ar + ar^2 + \cdots + ar^n) - r(a + ar + ar^2 + \cdots + ar^n) \\ &= a + ar + ar^2 + \cdots + ar^n - ar - ar^2 - \cdots - ar^{n+1} \\ &= a - ar^{n+1} \\ s_n(1 - r) &= a(1 - r^{n+1}) \end{aligned}$$

and therefore

$$s_n = \frac{a(1 - r^{n+1})}{1 - r}.$$

If $0 < |r| < 1$, then the term $r^{n+1} \to 0$ as $n \to \infty$ and we obtain the sum of the geometric series (2.2) as

$$s = \frac{a}{1 - r}. \tag{2.3}$$

The series (2.2) is called a *geometric series*.

A simpler way of determining whether a series converges or diverges is to make use of the following theorem:

Theorem 2.1 (Ratio Test)**.** The series $\sum_{n=0}^{\infty} a_n$ converges if

$$\lim_{n\to\infty} \left| \frac{a_n}{a_{n-1}} \right| < 1$$

and diverges if

$$\lim_{n\to\infty} \left| \frac{a_n}{a_{n-1}} \right| > 1.$$

If this limit is equal to 1, then we cannot conclude anything about the convergence or divergence of the series.

Example 2.8. Let us reconsider the geometric series (2.2). By the ratio test, we now have

$$\left| \frac{x_n}{x_{n-1}} \right| = \left| \frac{ar^n}{ar^{n-1}} \right| = |r|$$

and thus we require $|r| < 1$ for convergence. This concurs with our earlier analysis in example 2.7.

2.4 Taylor series expansions

From analysis it is known that a continuous function may be approximated by finite or infinite series, and these approximations are normally done by power series expansions. A *power series* is

an infinite series of the form

$$\sum_{i=0}^{\infty} a_i x^i = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n + \cdots \tag{2.4}$$

Example 2.7 is a special case of a power series. In the following sections we shall discuss a very useful power series expansion—the *Taylor series expansion* of a function—which is a powerful analytical and numerical tool in applied mathematics and is used extensively in later chapters.

2.4.1 Taylor's Theorem (1715)

We consider a function f that has continuous derivatives up to $(n+1)$th order on an interval $[a, b]$. From the fundamental theorem of calculus we have

$$f(b) = f(a) + \int_a^b f'(x)\,dx.$$

Repeated integration by parts yields

$$\begin{aligned}
f(b) &= f(a) + (x-b)\big[f'(x)\big]_a^b + \int_a^b (x-b) f''(x)\,\mathrm{d}x \\
&= f(a) + (b-a)f'(a) + \left[\frac{(b-x)^2}{2} f''(x)\right]_a^b + \int_a^b \frac{(b-x)^2}{2} f'''(x)\,\mathrm{d}x \\
&= f(a) + (b-a)f'(a) + \frac{(b-a)^2}{2!} f''(a) \\
&\quad + \left[-\frac{(b-x)^3}{2\times 3} f'''(x)\right]_a^b + \int_a^b \frac{(b-x)^3}{2\times 3} f^{(4)}(x)\,\mathrm{d}x \\
&= f(a) + (b-a)f'(a) + \frac{(b-a)^2}{2!} f''(a) \\
&\quad + \frac{(b-a)^3}{3!} f'''(a) + \int_a^b \frac{(b-x)^3}{3!} f^{(4)}(x)\,\mathrm{d}x
\end{aligned}$$

After n steps we have Taylor's Theorem:

$$\begin{aligned}
f(b) &= f(a) + (b-a)f'(a) + \cdots + \frac{(b-a)^n}{n!} f^{(n)}(a) + R_n \\
R_n &= \int_a^b \frac{(b-x)^n}{n!} f^{(n+1)}(x)\,\mathrm{d}x
\end{aligned}$$

With the substitutions $b \to x, a \to x_0, x \to t$ we obtain the more familiar form

$$\begin{aligned}
f(x) &= f(x_0) + (x-x_0)f'(x_0) + \cdots + \frac{(x-x_0)^n}{n!} f^{(n)}(x_0) + R_n \\
R_n &= \int_{x_0}^x \frac{(x-t)^n}{n!} f^{(n+1)}(t)\,\mathrm{d}t
\end{aligned} \tag{2.5}$$

We refer to R_n as the *residual term.*

2.4.2 Taylor series

Consider a function f for which derivatives of all orders exist at x_0. In other words, we may write (2.5) for arbitrary n. The power series in $(x - x_0)$ on the right-hand side of (2.5) will converge to a finite value if $\lim_{n\to\infty} R_n = 0$. Hence, we may expand $f(x)$ then as an infinite power series:

$$f(x) = \sum_{n=0}^{\infty} \frac{(x-x_0)^n}{n!} f^{(n)}(x_0) \tag{2.6}$$

The convergence, or lack thereof, of the series (2.6) may be investigated by means of estimates of R_n. Cauchy and Lagrange have given estimates for the residual term (see §2.4.3), but we could also make use of theorem 2.1.

Example 2.9. We want to write $f(x) = e^x$ as an infinite power series. Since e^x and all its derivatives exist at $x = 0$, we choose x_0 in (2.6). Then

$$\begin{aligned} f(x) &= e^x & f(0) &= 1 \\ f'(x) &= e^x & f'(0) &= 1 \\ &\vdots & &\vdots \\ f^{(n)}(x) &= e^x & f^{(n)}(0) &= 1 \end{aligned}$$

and so

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

We investigate the convergence of this series using theorem 2.1:

$$e^x = \sum_{n=0}^{\infty} a_n \qquad a_n = \frac{x^n}{n!}$$

Hence

$$\left|\frac{a_n}{a_{n-1}}\right| = \left(\frac{x^n}{n!}\right) \Big/ \left(\frac{x^{n-1}}{(n-1)!}\right) = \frac{x}{n}$$

so that

$$\left|\frac{a_n}{a_{n-1}}\right| = \frac{|x|}{n} \to 0 \text{ as } n \to \infty$$

Thus, this series converges for all values of x.

Example 2.10. We obtain the so-called *binomial expansion* by determining the Taylor series of $f(x) = (1 + x)^p$ for all real values of p. Let $x_0 = 0$. Then

$$\begin{aligned} f(x) &= (1+x)^p & f(0) &= 1 \\ f'(x) &= p(1+x)^{p-1} & f'(0) &= p \\ f''(x) &= p(p-1)(1+x)^{p-2} & f''(0) &= p(p-1) \\ &\vdots & &\vdots \\ f^{(n)} &= p(p-1)\cdots(p-n+1)(1+x)^{p-n} & f^{(n)}(0) &= p(p-1)\cdots(p-n+1) \end{aligned}$$

and so

$$(1+x)^p = \sum_{n=0}^{\infty} \binom{p}{n} x^n \tag{2.7}$$

where

$$\binom{p}{n} = \frac{p\,(p-1)\cdots(p-n+1)}{n!}$$

are called the *binomial coefficients.* We note that $\binom{p}{0} = 1$.

A special case is $p = \frac{1}{2}$, e.g.

$$\begin{aligned} (1+x)^{\frac{1}{2}} &= 1 + \frac{1}{2}x + \frac{\left(\frac{1}{2}\right)\left(-\frac{1}{2}\right)}{2!}x^2 + \cdots \\ &= 1 + \frac{1}{2}x - \frac{1}{8}x^2 + \cdots \end{aligned}$$

We use theorem 2.1 to test the convergence of this series. From (2.7) the series has the form

$$(1+x)^p = \sum_{n=0}^{\infty} a_n$$

with

$$a_n = \frac{p(p-1)\cdots(p-n+1)}{n!}x^n.$$

Hence

$$\frac{a_n}{a_{n-1}} = \frac{p(p-1)\cdots(p-n+2)(p-n+1)x^n}{(1)(2)\cdots(n-1)n} \times \frac{(1)(2)\cdots(n-1)}{p(p-1)\cdots(p-n+2)x^{n-1}}$$

and so

$$\left|\frac{a_n}{a_{n-1}}\right| = \left|\frac{(p-n+1)x}{n}\right| = \left|\frac{p+1}{n} - 1\right| |x| \,.$$

Clearly $\left|\frac{a_n}{a_{n-1}}\right| \to |x|$ as $n \to \infty$, and so this series converges only if $|x| < 1$.

Example 2.11. Find e^x to an accuracy of ϵ. We calculate here the series

$$1 + x + \frac{x^2}{2!} + \cdots + \frac{x^N}{N!},$$

where the Nth term is the first one for which $\left|\frac{x^N}{N!}\right| < \epsilon$. A numerical complication that arises is that $69! \approx 10^{100}$ and that $n!$ cannot be evaluated using a pocket calculator for $n \geq 70$. Instead we make use of the recursion relation (from example 2.9)

$$a_0 = 1 \qquad a_n = \left(\frac{x}{n}\right) a_{n-1} \qquad n > 0$$

We obtain $e^1 = 2.718282$ with 10 terms. Note the required accuracy: $\epsilon = 10^{-6}$, therefore *six* decimal places are shown.

2.4.3 Lagrange estimate of the error term

Lagrange obtained an estimate of the error term R_n in (2.5) which is very useful for analytical purposes. The generalized mean-value theorem for integral calculus allows us to write the residual term as

$$R_n = f^{(n+1)}(\xi_x) \int_{x_0}^{x} \frac{(x-t)^n}{n!} \, \mathrm{d}t$$

where $x_0 < \xi_x < x$. The integral in the residual term is now easily determined to be

$$R_n = f^{(n+1)}(\xi_x) \left[\frac{-1}{n!} \times \frac{(x-t)^{n+1}}{(n+1)}\right]_{x_0}^{x} = \frac{(x-x_0)^{n+1}}{(n+1)!} f^{(n+1)}(\xi_x) \,.$$

The series in (2.5) now becomes

$$\begin{aligned} f(x) = f(x_0) + (x-x_0)f'(x_0) + \cdots \\ + \frac{(x-x_0)^n}{n!}f^{(n)}(x_0) + \frac{(x-x_0)^{n+1}}{(n+1)!}f^{(n+1)}(\xi_x). \end{aligned} \tag{2.8}$$

It is often useful to define $x = x_0 + h$. Then (2.8) becomes

$$\begin{aligned} f(x_0+h) = f(x_0) + hf'(x_0) + \cdots \\ + \frac{h^n}{n!}f^{(n)}(x_0) + \frac{h^{n+1}}{(n+1)!}f^{(n+1)}(\xi_x) \end{aligned} \tag{2.9}$$

where $x_0 < \xi < x_0 + h$.

2.4.4 Expansion of multivariable functions

If a function is dependent on more than one variable, we use *consecutive* Taylor expansions with respect to each of the variables. Since all other variables are held constant while expanding with respect to a particular variable, all derivatives in the expansion are partial. Here, we expand a function of two variables. We consider $f(x_0 + h, y_0 + k)$ and use (2.9) to obtain an expansion with respect to x (holding y constant) and then we expand each term of this expansion with respect to y (holding x constant), i.e.

$$\begin{aligned}
f(x_0+h,y_0+k) &= f(x_0,y_0+k) + h\frac{\partial f}{\partial x}\bigg|_{(x_0,y_0+k)} + \frac{h^2}{2!}\frac{\partial^2 f}{\partial x^2}\bigg|_{(x_0,y_0+k)} + \cdots \\
&= f(x_0,y_0) + k\frac{\partial f}{\partial y}\bigg|_{(x_0,y_0)} + \frac{k^2}{2!}\frac{\partial^2 f}{\partial y^2}\bigg|_{(x_0,y_0)} + \cdots \\
&\quad + \frac{\partial}{\partial y}\left(h\frac{\partial f}{\partial x}\bigg|_{(x_0,y_0+k)} + \frac{h^2}{2!}\frac{\partial^2 f}{\partial x^2}\bigg|_{(x_0,y_0+k)} + \cdots \right) \\
&= f(x_0,y_0) + k\frac{\partial f}{\partial y}\bigg|_{(x_0,y_0)} + \frac{k^2}{2!}\frac{\partial^2 f}{\partial y^2}\bigg|_{(x_0,y_0)} + \cdots \\
&\quad + h\left(\frac{\partial f}{\partial x}\bigg|_{(x_0,y_0)} + k\frac{\partial^2 f}{\partial y \partial x}\bigg|_{(x_0,y_0)} + \cdots \right) \\
&\quad + \frac{h^2}{2!}\left(\frac{\partial^2 f}{\partial x^2}\bigg|_{(x_0,y_0)} + \frac{\partial^3 f}{\partial y \partial x^2}\bigg|_{(x_0,y_0)} \cdots \right) + \cdots \\
&= f(x_0,y_0) + \left(h\frac{\partial f}{\partial x}\bigg|_{(x_0,y_0)} + k\frac{\partial f}{\partial y}\bigg|_{(x_0,y_0)} \right) \\
&\quad + \left(\frac{h^2}{2!}\frac{\partial^2 f}{\partial x^2}\bigg|_{(x_0,y_0)} + hk\frac{\partial^2 f}{\partial y \partial x}\bigg|_{(x_0,y_0)} + \frac{k^2}{2!}\frac{\partial^2 f}{\partial y^2}\bigg|_{(x_0,y_0)} \right) + \cdots
\end{aligned}$$

and in summary

$$f(x_0+h,y_0+k) = \sum_{n=0}^{\infty}\sum_{m=0}^{\infty} \frac{h^n k^m}{n!m!}\frac{\partial^{n+m} f}{\partial x^n \partial y^m}\bigg|_{(x_0,y_0)}.$$

The series for a function of more than two variables is analogous.

Chapter 3

NONLINEAR EQUATIONS

3.1 Introduction

Generally speaking, nonlinear equations and, in particular, so-called *transcendental* equations, cannot be solved *analytically*, and so may *only* be solved numerically. We consider the solution of nonlinear equations in one variable, although we briefly describe a method for a system of two nonlinear equations at the end of this chapter.

3.2 The bisection method

Consider the equation $f(x) = 0$, where f is assumed to be continuous. By the intermediate value theorem, if an interval $[x_1, x_2]$ can be found on which f changes sign, i.e.

$$f(x_1)f(x_2) < 0,$$

then f *has at least one real root* on the interval.

Assume that f has only one root on $[x_1, x_2]$, denote this root as x_0. The interval is now halved by determining

$$x_3 = \frac{x_1 + x_2}{2}.$$

This value x_3 may be regarded as an *approximation* to x_0. The approximation may be improved by *iterating* (repeating) the halving process. At each iteration there are two possibilities:

(a) $f(x_1)f(x_3) < 0$. The root thus lies on $[x_1, x_3]$, and x_2 is replaced by x_3.

(b) $f(x_3)f(x_2) < 0$. The root lies on $[x_3, x_2]$, and x_1 is replaced by x_3.

The halving process is iterated until a specified *accuracy* ϵ *in the function value* is reached, that is

$$|f(x_3)| < \epsilon. \tag{3.1}$$

Convergence. Since any curve on a small enough interval may be approximated by a straight line, we have that $f(x_3)$ will converge to $f(x_0)$, in the vicinity of x_0, at the same rate that

$$\eta = |x_0 - x_3|$$

converges to zero. For the nth iteration we have

$$\eta_n \approx \frac{\eta_{n-1}}{2}.$$

The bisection method is thus said to be *linearly* convergent.

Example 3.1. Solve $f(x) = \sin x - 0.625x = 0$ to an accuracy of 8 decimal places in the function value.

We show the values for the first four iterations in the following table:

iteration	x_1	x_3	x_2	$f(x_1)$	$f(x_3)$	$f(x_2)$
1	1	1.5	2	+0.216	+0.060	−0.341
2	1.5	1.75	2	+0.060	−0.110	−0.341
3	1.5	1.625	1.75	+0.060	−0.171	−0.110
4	1.5	1.5625	1.625	+0.060	+0.023	−0.171

After 25 iterations we find

$$x = 1.59934789 \text{ rad } (91.635884^\circ)$$

3.3 Linear interpolation

Although the bisection method is reliable, it is also slow. This is because very little information about $f(x)$ is used—indeed, only the sign of f is used. In the linear interpolation method we also make use of the *numerical values* of $f(x)$.

Consider the equation $f(x) = 0$. Let (x_1, y_1) and (x_2, y_2) be two points on the curves $y = f(x)$ in the vicinity of the root $x = x_0$. We approximate the curve in this region by a *straight line* through the two points. The zero point x_3 of the straight line may be regarded as an approximation to the root x_0.

The value of x_3 is obtained from the equation for the straight line

$$y - y_1 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x - x_1).$$

The point $(x_3, 0)$ lies on this line so that

$$0 - y_1 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_3 - x_1)$$

and so

$$x_3 = \frac{x_1 y_2 - x_2 y_1}{y_2 - y_1}.$$

The interpolation process is now iterated by finding the straight line through (x_2, y_2) and (x_3, y_3) and hence the approximation x_4, and so on. In general, the $(i-1)$th iteration uses (x_{i-1}, y_{i-1}) and (x_i, y_i) to find x_{i+1}:

$$x_{i+1} = \frac{x_{i-1} y_i - x_i y_{i-1}}{y_i - y_{i-1}}. \tag{3.2}$$

For a given accuracy ϵ this process is repeated until the condition (3.1) is satisfied.

Convergence. Let the error in x after the $(i-1)$th iteration be denoted by η_i. In other words, $x_i = x_0 + \eta_i$. Then from equation (3.2) it follows that

$$x_0 + \eta_{i+1} = \frac{(x_0 + \eta_{i-1}) y_i - (x_0 + \eta_i) y_{i-1}}{y_i - y_{i-1}}$$

and so

$$\eta_{i+1} = \frac{\eta_{i-1} f(x_0 + \eta_i) - \eta_i f(x_0 + \eta_{i-1})}{f(x_0 + \eta_i) - f(x_0 + \eta_{i-1})}.$$

Each function may be expanded in a Taylor series about x_0:

$$\begin{aligned} f(x_0 + \eta_i) &= f(x_0) + \eta_i f'(x_0) + \frac{1}{2}\eta_i^2 f''(x_0) + \cdots \\ &= 0 + \eta_i f'(x_0) + \frac{1}{2}\eta_i^2 f''(x_0) + \cdots \end{aligned}$$

We assume that $f'(x_0) \neq 0$ and $f''(x_0) \neq 0$. Since η_i is small (by assumption) we have, to the *lowest order* in η_i

$$\begin{aligned}\eta_{i+1} &\approx \frac{\eta_{i-1}\left[\eta_i f'(x_0) + \frac{1}{2}\eta_i^2 f''(x_0)\right] - \eta_i\left[\eta_{i-1} f'(x_0) + \frac{1}{2}\eta_{i-1}^2 f''(x_0)\right]}{\eta_i f'(x_0) - \eta_{i-1} f'(x_0)} \\ &= \frac{1}{2}\left(\frac{\eta_i \eta_{i-1}\left(\eta_i - \eta_{i-1}\right) f''(x_0)}{\left(\eta_i - \eta_{i-1}\right) f'(x_0)}\right) \\ &= \left(\frac{f''(x_0)}{2f'(x_0)}\right)\eta_i \eta_{i-1} \\ &\equiv A\eta_i\eta_{i-1}, \quad \text{where } A \text{ is a constant.} \end{aligned} \tag{3.3}$$

We attempt to satisfy this relationship by assuming

$$\eta_i = K\eta_{i-1}^a \quad \Longleftrightarrow \quad \eta_{i+1} = K\eta_i^a$$

where K is some constant, so that, from (3.3), we have

$$\eta_{i+1} \approx A\eta_i\left(\frac{1}{K}\eta_i\right)^{\frac{1}{a}}.$$

But

$$\eta_{i+1} = K\eta_i^a$$

from our earlier assumption, and since the powers of η_i must be the same on both sides of the equation, it follows that

$$1 + \frac{1}{a} = a$$

which gives

$$a^2 - a - 1 = 0$$

which has the root

$$a = \frac{1+\sqrt{5}}{2} \approx 1.618$$

which is the also called the *golden mean number*. Hence

$$\eta_{i+1} \approx K\eta_i^{1.618}.$$

This rate of convergence is termed *superlinear*. (The other root $a \approx -0.62$ corresponds to divergence.)

Example 3.2. Solve $f(x) = \sin x - 0.625x = 0$ accurate to 8 decimal places in the function value.

The values for the first 3 iterations are shown in the following table.

i	x_i	x_{i+1}	y_i	y_{i+1}	x_{i+2}	y_{i+2}
1	1	2	0.2165	−0.3407	1.3885	0.1156
2	2	1.3885	−0.3407	0.1156	1.5434	0.0350
3	1.3885	1.5434	0.1156	0.0350	1.6106	−0.0074

After 6 iterations we obtain $x_0 = 1.59934789$ rad (compare with 25 iterations for the bisection method).

3.4 Newton's method

For this method we assume that f is at least twice differentiable. We now use both the values $f(x)$ and its first derivative to solve $f(x) = 0$. Let x_1 be an initial estimate of the root x_0, and draw a tangent line at the point $(x_1, f(x_1))$ to the curve of f. The x-intercept of the tangent line, denote it x_2, is presumably a better estimate of x_0 than x_1 was. The slope of the tangent line is $f'(x_1)$ and is determined by

$$f'(x_1) = \frac{f(x_2) - f(x_1)}{x_2 - x_1}.$$

However, since x_2 is the x-intercept of the tangent line, we have $f(x_2) = 0$ and thus

$$f'(x_1) = \frac{f(x_1)}{x_1 - x_2}.$$

Solving this last equation for x_2 we obtain

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}.$$

This process can be repeated and we find, in general, after the ith iteration we have

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}. \tag{3.4}$$

Note: The above result could also have been obtained by calculating the error η for an initial estimate x_1 by solving

$$f(x_1 + \eta) = 0.$$

We expand the LHS in a Taylor series about $x = x_1$

$$f(x_1) + \eta f'(x_1) + \frac{\eta^2}{2!} f''(x_1) + \cdots = 0.$$

For small η, $f(x_1) + \eta f'(x_1) \approx 0$ and so

$$\eta \approx -\frac{f(x_1)}{f'(x_1)}.$$

Newton's method is thus equivalent to a first-order Taylor expansion of the function $f(x)$.

Convergence. Let the error in x after the ith iteration be denoted by η_i. In other words, $x_i = x_0 + \eta_i$. From equation (3.4)

$$x_0 + \eta_{i+1} = x_0 + \eta_i - \frac{f(x_0 + \eta_i)}{f'(x_0 + \eta_i)}.$$

A Taylor expansion about $x = x_0$ for both $f(x_0 + \eta_i)$ and $f'(x_0 + \eta_i)$ gives

$$\eta_{i+1} = \eta_i - \frac{f(x_0) + \eta_i f'(x_0) + \frac{1}{2}\eta_i^2 f''(x_0) + \cdots}{f'(x_0) + \eta_i f''(x_0) + \cdots}.$$

Expanding the denominator using

$$\frac{1}{1+x} = 1 - x + x^2 - x^3 + \cdots$$

we find

$$\begin{aligned}
\frac{1}{f'(x_0) + \eta_i f''(x_0) + O(\eta_i^2)} &= \frac{1}{f'(x_0)\left(1 + \eta_i \frac{f''(x_0)}{f'(x_0)} + O(\eta_i^2)\right)} \\
&= \frac{1}{f'(x_0)} \frac{1}{1 + \eta_i \frac{f''(x_0)}{f'(x_0)} + O(\eta_i^2)} \\
&= \frac{1}{f'(x_0)} \left(1 - \eta_i \frac{f''(x_0)}{f'(x_0)} + O(\eta_i^2)\right)
\end{aligned}$$

To lowest order in η we have

$$\begin{aligned}\eta_{i+1} &\approx \eta_i - \frac{1}{f'(x_0)}\left(1 - \eta_i \frac{f''(x_0)}{f'(x_0)}\right)\left(0 + \eta_i f'(x_0) + \frac{\eta_i^2}{2} f''(x_0)\right) \\ &\approx \left(\frac{f''(x_0)}{2f'(x_0)}\right)\eta_i^2. \end{aligned} \tag{3.5}$$

The convergence is said to be *quadratic.*

It is important to note that (3.3) and (3.5) give estimates for the convergence rate *only if convergence actually occurs.* It is possible that both the linear interpolation method and Newton's method may diverge.

Example 3.3. Find the root of $f(x) = \sin x - 0.625x = 0$ correct to 8 decimal places in the function value.

Equation (3.4) is iterated using

$$f(x) = \sin x - 0.625x$$

and

$$f'(x) = \cos x - 0.625$$

with an initial estimate $x = 1.5$. The results of the process are shown in the following table:

i	x_i	$f(x_i)$	$f'(x_i)$	$\Delta x_i = -\frac{f(x_i)}{f'(x_i)}$
1	1.5	0.060	-0.554	0.108
2	1.608	-0.0059	-0.662	-0.00884
3	1.59941	-3.9×10^{-5}	-0.654	-6.0×10^{-5}
4	1.59934789	-2.1×10^{-9}		

After 4 iterations the root $x = 1.59934789$ rad is found.

Calculation hints

(a) Sometimes f' is a cumbersome analytical expression. In such cases f' may be estimated *numerically* by

$$f'(x_i) \approx \frac{f(x_i + h) - f(x_i)}{h}$$

where h is small. One should be aware of complications that may arise from numerical differentiation (see chapter 6 for better approximations. Linear interpolation is indeed Newton's method with simple numerical differentiation).

(b) When $f(x) = 0$ has more than one root, it is sometimes found that both linear interpolation and Newton's method converge to the same root, for any initial estimate. Say x_0 is such a root. Then we may write

$$f(x) = (x - x_0)g(x).$$

Once the root x_0 is found, it may be eliminated from the problem by subsequently solving

$$g(x) = \frac{f(x)}{x - x_0}.$$

3.5 Fixed-point iteration

If an equation can be written in the form

$$x = g(x) \tag{3.6}$$

then a root of the equation may be regarded as a fixed point that is mapped to itself under the map $x = g(x)$. The iteration process

$$x_{i+1} = g(x_i) \tag{3.7}$$

may then be used, *possibly*, to find the root.

Example 3.4. Find the positive root of $x = 2\sin x$ correct to 8 decimal places.

The following table shows 6 iterations of (3.7) with $g(x) = 2\sin x$, using initial value $x = 2$:

i	x_i	$g(x_i)$
1	2	1.82
2	1.82	1.94
3	1.94	1.87
4	1.87	1.91
5	1.91	1.88
6	1.88	1.90

The required accuracy is actually achieved after 39 iterations and we find $x_0 = 1.89549427$.

A transcendental equation of the form (3.6) can often be written in this form in several ways. For example, $e^x = 3x^2$ may be written as

$$x = \ln 3x^2$$

or

$$x = \pm\sqrt{\frac{e^x}{3}}.$$

Convergence. We now obtain an analytical condition for convergence. Consider the error in x after the ith iteration,

$$x_{i+1} - x_0 = g(x_i) - g(x_0)$$

where x_0 is the root of (3.6). From the mean-value theorem of differential calculus we have

$$\frac{g(x_i) - g(x_0)}{x_i - x_0} = g'(\xi_i)$$

where $x_0 < \xi_i < x_i$. We can thus rewrite the ith error as

$$x_{i+1} - x_0 = g'(\xi_i)(x_i - x_0).$$

Iterating this recursion relation i times gives

$$x_{i+1} - x_0 = g'(\xi_i)g(\xi_{i-1}) \cdots g'(\xi_1)(x_1 - x_0).$$

Now let

$$m = \max\left(|g'(\xi_i)|\right).$$

Then we have

$$|x_{i+1} - x_0| \leq m^i \, |x_1 - x_0| \, .$$

For convergence we must have that $|x_{i+1} - x_0| \to 0$ as $i \to \infty$, which will only be true if $m < 1$.

Condition for convergence: For the iteration scheme (3.7) to converge, we require

$$|g'(x)| < 1 \tag{3.8}$$

in the neighbourhood of the root.

Example 3.5. We investigate the convergence for the case in example 3.4.

For $x = g(x) = 2\sin x$ we have $g'(x) = 2\cos x$. Hence, for convergence we require

$$|\cos x| < \frac{1}{2}$$

and so we obtain an interval of convergence

$$1.047 < x < 2.094$$

The root $x_0 = 1.895$ indeed lies on this interval.

Note: Condition (3.8) may be applied to any iterative method that can be written in the form (3.6). Newton's method may be written as $x = g(x)$ with

$$g(x) = x - \frac{f(x)}{f'(x)}$$

and so

$$g'(x) = 1 - \frac{\big(f'(x)\big)^2 - f(x)f''(x)}{\big(f'(x)\big)^2}.$$

Then we must have

$$\left|\frac{f(x)f''(x)}{\big(f'(x)\big)^2}\right| < 1$$

in the neighbourhood of the root.

3.6 Systems of nonlinear equations

We describe Newton's method for two simultaneous nonlinear equations. The extension to systems of more than two equations can be found in most standard texts.

Consider two equations in unknowns x and y:

$$\begin{aligned} f_1(x, y) &= 0 \\ f_2(x, y) &= 0 \end{aligned} \tag{3.9}$$

As for the one-dimensional Newton's method, we try to find the corrections (η_x, η_y) to an initial guess (x_1, y_1) such that (3.9) is satisfied:

$$\begin{aligned} f_1(x_1 + \eta_x, y_1 + \eta_y) &= 0 \\ f_2(x_1 + \eta_x, y_1 + \eta_y) &= 0 \end{aligned}$$

A first-order Taylor expansion yields

$$\begin{aligned} f_1(x_1, y_1) + \eta_x \left(\left.\frac{\partial f_1}{\partial x}\right|_{(x_1,y_1)}\right) + \eta_y \left(\left.\frac{\partial f_1}{\partial y}\right|_{(x_1,y_1)}\right) &\approx 0 \\ f_2(x_1, y_1) + \eta_x \left(\left.\frac{\partial f_2}{\partial x}\right|_{(x_1,y_1)}\right) + \eta_y \left(\left.\frac{\partial f_2}{\partial y}\right|_{(x_1,y_1)}\right) &\approx 0 \end{aligned}$$

which may be written as

$$\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}_{(x_1,y_1)} \begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \approx \begin{bmatrix} -f_1(x_1,y_1) \\ -f_2(x_1,y_1) \end{bmatrix}$$

so that

$$\begin{bmatrix} \eta_x \\ \eta_y \end{bmatrix} \approx \left(\begin{bmatrix} \frac{\partial f_1}{\partial x} & \frac{\partial f_1}{\partial y} \\ \frac{\partial f_2}{\partial x} & \frac{\partial f_2}{\partial y} \end{bmatrix}_{(x_1,y_1)} \right)^{-1} \begin{bmatrix} -f_1(x_1,y_1) \\ -f_2(x_1,y_1) \end{bmatrix}$$

The square matrix here is known as a *Jacobian matrix*, and is evaluated at the point (x_1, y_1).

Chapter 4

SYSTEMS OF LINEAR EQUATIONS

4.1 Introduction

In this chapter we consider the numerical solution of a system of linear equations. The numerical method we describe is an iterative one, known as the *Jacobi method*.

4.2 Solvability

Consider m equations in n unknowns

$$\begin{aligned} a_{11}x_1 + \ a_{12}x_2 + \cdots + \ a_{1n}x_n &= b_1 \\ a_{21}x_1 + \ a_{22}x_2 + \cdots + \ a_{2n}x_n &= b_2 \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m \end{aligned}$$

which have the matrix representation

$$\mathsf{A}\boldsymbol{x} = \boldsymbol{b}. \tag{4.1}$$

From linear algebra the following is known regarding the solution of (4.1):

(a) Generally, there are no solutions if $m > n$.

(b) Generally, there are infinitely many solutions if $m < n$.

(c) If $m = n$ then A is a square matrix and possibly solvable.

4.3 Cramer's rule

For $m = n$ we define $D = \det \mathsf{A}$. Cramer's rule differentiates between three cases:

(a) If $D = 0$ and $\boldsymbol{b} \neq \boldsymbol{0}$ then, in general, there are no solutions.

(b) If $\boldsymbol{b} = \boldsymbol{0}$ and $D \neq 0$ then there is only the trivial solution $\boldsymbol{x} = \boldsymbol{0}$. Thus, a necessary condition for nontrivial solutions, if $\boldsymbol{b} = \boldsymbol{0}$, is that $D = 0$.

(c) The general case is when $D \neq 0$ and $\boldsymbol{b} \neq \boldsymbol{0}$. From Cramer's rule it follows that if D_j is the determinant obtained when the jth column of A is replaced by $\boldsymbol{b}$, then the solution is given by

$$x_j = \frac{D_j}{D} \qquad j = 1, 2, \ldots, n. \tag{4.2}$$

Determining (4.2) is equivalent to inverting the matrix equation (4.1), i.e.

$$\boldsymbol{x} = \mathsf{A}^{-1}\boldsymbol{b}.$$

In practice (4.2) is not used because of the large amount of computational effort required: it can be shown that $(n-1)n!$ multiplications and $n!-1$ additions are needed, which is approximately $nn!$ arithmetical operations in all.

4.4 The Jacobi method

If the diagonal elements a_{ii} of A are all nonzero, then we may implement an iterative technique to solve (4.1)), known as the *Jacobi method*.

Let

$$\mathsf{A} = \mathsf{D} + \mathsf{L} + \mathsf{U}$$

where D is the matrix of diagonal entries of A, L is the lower triangular part of A, and U is the upper triangular part of A. This gives

$$\begin{aligned}\mathsf{A}\boldsymbol{x} &= (\mathsf{D}+\mathsf{L}+\mathsf{U})\boldsymbol{x}\\ &= \mathsf{D}\boldsymbol{x} + (\mathsf{L}+\mathsf{U})\boldsymbol{x} = \boldsymbol{b}\end{aligned}$$

so that

$$\boldsymbol{x} = \mathsf{D}^{-1}\big(\boldsymbol{b} - (\mathsf{L}+\mathsf{U})\boldsymbol{x}\big) \tag{4.3}$$

provided that $a_{ii} \neq 0$. Note that the computation of D^{-1} is straightforward; since D is a diagonal matrix, its inverse is obtained simply by taking the reciprocal of its elements (this requires no more than n arithmetical operations). Equation (4.3) suggests the iteration scheme

$$\boldsymbol{x}^{(k)} = \mathsf{D}^{-1}\big(\boldsymbol{b} - (\mathsf{L}+\mathsf{U})\boldsymbol{x}^{(k-1)}\big), \qquad k = 0, 1, 2, \ldots \tag{4.4}$$

where k denotes the iteration count. Clearly, the implementation of this method requires an initial guess $\boldsymbol{x}^{(0)}$. It is interesting to note that (4.4)) is a form of fixed-point iteration, as seen in the previous chapter, although here it is of a multivariable nature (those variables being the components x_i of $\boldsymbol{x}$).

It can be shown that *convergence* to the exact solution $\boldsymbol{x}$ is guaranteed provided

$$|a_{ii}| > \sum_{\substack{j=1\\ i\neq j}}^{n} |a_{ij}| \tag{4.5}$$

for $i = 1, 2, \ldots, n$. In other words, on each row, the magnitude of the diagonal element must exceed the sum of the magnitudes of all the other elements on that row, and this must hold for all rows in A. An indication of the quality of the approximate solution $\boldsymbol{x}^{(k)}$ may be determined by computing the *residual*

$$\boldsymbol{r}^{(k)} = \mathsf{A}\boldsymbol{x}^{(k)} - \boldsymbol{b}$$

and the iteration process is stopped when the magnitude of $\boldsymbol{r}^{(k)}$ is less than some imposed tolerance.

It can be shown that the Jacobi method requires about n^2 arithmetical operations per iteration; in comparison with a computational implementation of Cramer's rule we have, as measure of relative efficiency,

$$\frac{Mn^2}{n \times n!} = \frac{Mn^2}{nn(n-1)!} = \frac{M}{(n-1)!}$$

where M is the number of Jacobi iterations. Clearly, if n is large we would expect the Jacobi method to be more efficient.

Example 4.1. Consider

$$\mathsf{A} = \begin{bmatrix} 5 & 3 & 1 \\ -2 & 3 & 0 \\ 6 & -1 & 8 \end{bmatrix} = \underbrace{\begin{bmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 8 \end{bmatrix}}_{\mathsf{D}} + \underbrace{\begin{bmatrix} 0 & 3 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathsf{U}} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ -2 & 0 & 0 \\ 6 & -1 & 0 \end{bmatrix}}_{\mathsf{L}}$$

$$\boldsymbol{b} = \begin{bmatrix} 3 \\ 0 \\ -7 \end{bmatrix}$$

Clearly, A satisfies (4.5) so that the Jacobi method may be used. Starting with

$$\boldsymbol{x}^{(0)} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

we find, after 40 iterations,

$$\boldsymbol{x}^{(40)} = \begin{bmatrix} 0.61184210526407 \\ 0.40789473683653 \\ -1.28289473683631 \end{bmatrix} \qquad \boldsymbol{r}^{(40)} = \begin{bmatrix} -0.06 \\ -0.18 \\ 0.57 \end{bmatrix} \times 10^{-10}$$

with $\left|\boldsymbol{r}^{(40)}\right| = 6.1 \times 10^{-11}$. The true solution is

$$\boldsymbol{x} = \begin{bmatrix} 0.61184210526316 \\ 0.40789473684210 \\ -1.28289473684211 \end{bmatrix}$$

and we see that the difference between each of these entries and those in $\boldsymbol{x}^{(40)}$ is less than 10^{-11}.

We note that there are other types of iterative methods for solving linear systems, such as the *Gauss-Seidel* method and *successive over-relaxtion* (SOR), but they are similar in spirit to the Jacobi method, which is the simplest of the three. Like the Jacobi method, these other methods also require that the diagonal entries of A must all be nonzero. Furthermore, for large n, all these iterative methods become relatively more efficient (with respect to arithmetic computation) than direct inversion.

Chapter 5

APPROXIMATION METHODS

5.1 Introduction

In this chapter we investigate two related problems: the approximation of given functions by other, simpler functions, and the fitting of known functions to given data. In the cases studied here, we approximate a continuous function by means of a polynomial.

5.2 Polynomial interpolation

Consider a function $y(x)$ given in the form of the coordinates (x_i, y_i), $i = 0, 1, \ldots, n$. We may approximate the function with a *polynomial* $p_n(x)$ of degree n that is exactly equal to $y(x)$ at the $n+1$ given points

$$y_i = p_n(x_i) = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_n x_i^n. \tag{5.1}$$

We note that there are $n+1$ coefficients a_i as well as $n+1$ points (x_i, y_i). The polynomial in (5.1) is known as an *interpolating polynomial.*

There are $n+1$ unknowns a_i that must be determined from the $n+1$ linear equations

$$\begin{aligned} a_0 + a_1 x_0 + \cdots + a_n x_0^n &= p(x_0) = y_0 \\ a_0 + a_1 x_1 + \cdots + a_n x_1^n &= p(x_1) = y_1 \\ &\vdots \\ a_0 + a_1 x_n + \cdots + a_n x_n^n &= p(x_n) = y_n \end{aligned}$$

which can be written compactly in matrix form as

$$\begin{pmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_0 & \cdots & x_0^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

If $\det\left(x_j^i\right) \neq 0$, these equations have a unique solution. In practice it is difficult and/or inefficient to solve these equations directly, and so we investigate easier and/or faster methods. Firstly, though, we state a few points of general importance.

We may rightly ask whether this interpolating polynomial is unique. Assume that it is not, and thus there are two interpolating polynomials

$$\begin{aligned} p_n(x) &= a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \\ q_n(x) &= b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n \end{aligned}$$

Then

$$\begin{aligned} Q(x) &= p_n(x) - q_n(x) \\ &= (a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)x^2 + \cdots + (a_n - b_n)x^n \end{aligned}$$

is a polynomial of degree n. But $Q(x) = 0$ at the $n+1$ points $\{x_0, x_1, x_2, \ldots, x_n\}$. However, since $Q(x)$ is of degree n, it may only have n roots. This contradiction is resolved only if $Q(x) = 0$, which implies $p_n(x) = q_n(x)$. We conclude that the interpolating polynomial is unique.

Error analysis. We obtain an estimate of the error made when using the polynomial $p_n(x)$ instead of the function $y(x)$. We know that $y(x) = p_n(x)$ at $x_i, i = 0, 1, 2, \ldots, n$. Now consider

$$F(x) = y(x) - p_n(x) - C\prod_{i=0}^{n}(x - x_i). \tag{5.2}$$

Clearly, $F(x) = 0$ at each of the nodes $\{x_0, x_1, \ldots, x_n\}$. Consider any point $x_{n+1} \notin \{x_0, x_1, \ldots, x_n\}$. We choose C such that $F(x_{n+1}) = 0$, which gives

$$C = \frac{y(x_{n+1}) - p_n(x_{n+1})}{\prod_{i=0}^{n}(x_{n+1} - x_i)}. \tag{5.3}$$

Assume $y(x)$, and hence $F(x)$, is continuous. From the constructions (5.2) and (5.3) it follows that $F(x) = 0$ at the $n+2$ nodes $\{x_0, x_1, \ldots, x_n, x_{n+1}\}$. Hence, by the Generalized Rolle's Theorem, there exists a point ξ such that

$$F^{(n+1)}(\xi) = 0.$$

If we differentiate (5.2) we obtain

$$\begin{aligned} 0 &= F^{(n+1)}(\xi) \\ &= y^{(n+1)}(\xi) - 0 - C(n+1)! \end{aligned}$$

and so

$$C = \frac{y^{(n+1)}(\xi)}{(n+1)!},$$

which yields, using (5.3),

$$y(x_{n+1}) - p_n(x_{n+1}) = \frac{y^{(n+1)}(\xi)}{(n+1)!}\prod_{i=0}^{n}(x_{n+1} - x_i).$$

But x_{n+1} was chosen arbitrarily, so that this error expression

$$y(x) - p_n(x) = \frac{y^{(n+1)}(\xi(x))}{(n+1)!}\prod_{i=0}^{n}(x - x_i). \tag{5.4}$$

is true for all x. It is easily verified that (5.4) is also true at $\{x_0, x_1, \ldots, x_n\}$.

5.3 Lagrange's method

We now describe a method developed by Lagrange that is a shortcut for determining the coefficients a_i in (5.1).

For each of the points $(x_i, y_i),\ i = 0, 1, \ldots, n$, we construct an nth degree polynomial that is equal to zero at each of the other points $x_0, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$

$$L_i(x) = A(x - x_0)(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n). \tag{5.5}$$

Furthermore, we demand that this polynomial has the value 1 when $x = x_i$

$$1 = A(x_i - x_0)(x_i - x_1)\cdots(x_i - x_i - 1)(x_i - x_{i+1})\cdots(x_i - x_n)$$

from which we obtain

$$A = \frac{1}{(x_i - x_0)(x_i - x_1)\cdots(x_i - x_{i-1})(x_i - x_{i+1})\cdots(x_i - x_n)}$$

so that

$$L_i(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{i-1})(x - x_{i+1})\cdots(x - x_n)}{(x_i - x_0)(x_i - x_1)\cdots(x_i - x_i - 1)(x_i - x_{i+1})\cdots(x_i - x_n)}.$$

The nth degree polynomial through the points (x_i, y_i) is then given by

$$p_n(x) = \sum_{i=0}^{n} y_i L_i(x). \tag{5.6}$$

where the $L_i(x)$ are known as the *Lagrange coefficient polynomials*. We may test if this is the required polynomial by substituting $x = x_k$ into equation (5.6)

$$\begin{aligned} p_n(x_k) &= \sum_{i=0}^{n} y_i L_i(x_k) \\ &= y_0 L_0(x_k) + y_1 L_1(x_k) + \cdots + y_k L_k(x_k) + \cdots + y_n L_n(x_k) \\ &= 0 + 0 + \cdots + y_k(1) + \cdots + 0 \\ &= y_k. \end{aligned}$$

Example 5.1. Approximate the sine function with a polynomial that is equal to $\sin x$ at $x = 0, \frac{\pi}{4}, \frac{\pi}{2}$. Determine the true error at $x = \frac{\pi}{6}$ and compare it to an easily calculable upper bound of the error.

The coordinates of the three points that the interpolating polynomial must pass through are shown the following table:

i	x_i	$y_i = \sin x_i$
0	0	0
1	$\frac{\pi}{4}$	$\frac{1}{\sqrt{2}}$
2	$\frac{\pi}{2}$	1

We first determine the polynomials defined in (5.5)

$$\begin{aligned} L_0(x) &= \frac{\left(x - \frac{\pi}{4}\right)\left(x - \frac{\pi}{2}\right)}{\left(0 - \frac{\pi}{4}\right)\left(0 - \frac{\pi}{2}\right)} = \frac{8}{\pi^2}\left(x^2 - \frac{3\pi}{4}x + \frac{\pi^2}{8}\right) \\ L_1(x) &= \frac{(x - 0)\left(x - \frac{\pi}{2}\right)}{\left(\frac{\pi}{4} - 0\right)\left(\frac{\pi}{4} - \frac{\pi}{2}\right)} = -\frac{16}{\pi^2}\left(x^2 - \frac{\pi}{2}x\right) \\ L_2(x) &= \frac{(x - 0)\left(x - \frac{\pi}{4}\right)}{\left(\frac{\pi}{2} - 0\right)\left(\frac{\pi}{2} - \frac{\pi}{4}\right)} = \frac{8}{\pi^2}\left(x^2 - \frac{\pi}{4}x\right) \end{aligned}$$

The interpolating polynomial is given by

$$\begin{aligned} p_2(x) &= y_0 L_0(x) + y_1 L_1(x) + y_2 L_2(x) \\ &= 0 + \frac{1}{\sqrt{2}} L_1 + (1) L_2 \\ &= -\frac{8\left(\sqrt{2} - 1\right)}{\pi^2}x^2 + \frac{2\left(2\sqrt{2} - 1\right)}{\pi}x \end{aligned}$$

We determine the approximation error at $x = \frac{\pi}{6}$

$$\left|\sin\left(\frac{\pi}{6}\right) - p_2\left(\frac{\pi}{6}\right)\right| = |0.5 - 0.517| = 0.017$$

It is interesting to compare this error with the upper limit (5.4)

$$\begin{aligned}\left|\sin\left(\frac{\pi}{6}\right) - p_2\left(\frac{\pi}{6}\right)\right| &\leq \frac{\max_{[0,\pi/2]}|\cos x|}{3!}\left|\prod_{i=0}^{2}\left(\frac{\pi}{2}\right)\right| \\ &= \frac{1}{6}\left|\left(\frac{\pi}{6}-0\right)\left(\frac{\pi}{6}-\frac{\pi}{4}\right)\left(\frac{\pi}{6}-\frac{\pi}{2}\right)\right| \\ &= 0.024\end{aligned}$$

We see that using (5.4) does indeed give, in this case, an upper bound for the magnitude of the approximation error.

5.4 Least-squares curve fitting

A function $y = y(x)$ may be approximated by an nth degree polynomial that passes through each data point, as in §5.2 and §5.3 . However, if we believe that the relationship $y(x)$ is in fact linear, then it would make more sense to find the "best" straight line that approximates the function. We need a criterion that allows us to determine the fitting function such that its deviation from the given points is minimized. The deviation Δ_i at point x_i is the difference between the fit $f(x)$ and the actual function $y(x)$, i.e.

$$\Delta_i = f(x_i) - y_i.$$

We examine a few optimization criteria, given n data points.

Minimization of $\sum_i \Delta_i$: Since errors may be both positive and negative, a positive error and negative error summed will give a sum that is less in magnitude than either error. Furthermore, since $-1+1 = -2+2 = -3+3 = \cdots = 0$ it is clear that the sum cannot give a unique minimum. This means that such a fitting criterion cannot allow the fitting function to be uniquely determined.

Minimization of $\sum_i |\Delta_i|$: When we have an error range (y_-, y_+), $y_- < y_+$ around the y coordinate of a data point (x, y), then the straight line fit $f(x) = mx + c$ passing between (x, y_-) and (x, y_+) such that $y_- \leq f(x) \leq y_+$ yields

$$|f(x) - y_-| + |f(x) - y_+| \leq |y_+ - f(x) + f(x) - y_-| = |y_+ - y_-|$$

which is independent of m and c. Again, the fitting function cannot be determined uniquely.

Minimization of $\sum_i \Delta_i^2$: Let $f(x)$ be a function fitting the data. Suppose we have two data points with the same x coordinates but with different y coordinates, namely (x, y_1) and (x, y_2). We have that

$$S = \sum_i \Delta_i^2 = (f(x) - y_2)^2 + (f(x) - y_1)^2 = e^2 + (e + d)^2,$$

where $e = f(x) - y_2$ and $d = y_2 + y_1$. Then it follows that

$$\frac{\mathrm{d}S}{\mathrm{d}e} = 2e + 2(e + d) = 2(2e + d)$$

and

$$\frac{\mathrm{d}^2 S}{\mathrm{d}e^2} = 4 > 0.$$

Therefore S will be a *minimum* if $e = \frac{d}{2}$. This means that the value of $f(x)$ is the mean of the two y coordinates; a result which is both unique and intuitively acceptable and from mathematical statistics this norm is known to be the correct choice. The function $f(x)$ is therefore chosen so that

$$S = \sum_{i=0}^{n} [f(x_i) - y_i]^2 \tag{5.7}$$

is a minimum. We will see that, in the case of polynomial fitting, the norm (5.7) gives a unique result.

5.5 Least-squares polynomial fitting

A wide variety of functions may be used in least-squares curve fitting. We will leave the fitting of trigonometric, exponential and logarithmic functions etc. to the student as self-study, and discuss only the use of polynomials as fitting functions.

We consider the case where $f(x)$ in (5.7) is an mth degree polynomial $p_m(x)$, with $m \leq n$. Then $p_m(x)$ must be chosen such that

$$S = \sum_{i=0}^{n} [p_m(x_i) - y_i]^2 \tag{5.8}$$

is a *minimum*. Since

$$p_m(x) = a_0 + a_1 x + \cdots + a_m x^m \tag{5.9}$$

we must demand that

$$\frac{\partial S}{\partial a_k} = 0 \qquad \text{for } k = 0, 1, \ldots, m. \tag{5.10}$$

From (5.8) we then have

$$\frac{\partial S}{\partial a_k} = 2 \sum_{i=0}^{n} [p_m(x_i) - y_i] \frac{\partial p_m}{\partial a_k}(x_i) \qquad \text{for } k = 0, 1, \ldots, m.$$

From (5.9) we have that

$$\frac{\partial p_m}{\partial a_k}(x_i) = x_i^k$$

and so

$$\frac{\partial S}{\partial a_k} = 2 \sum_{i=0}^{n} [p_m(x_i) - y_i] x_i^k \qquad \text{for } k = 0, 1, \ldots, m. \tag{5.11}$$

Furthermore,

$$\frac{\partial^2 S}{\partial a_k^2} = 2 \sum_{i=0}^{n} x_i^k x_i^k \geq 0 \qquad \text{for } k = 0, 1, \ldots, m,$$

so that the requirement in (5.10) does indeed give a minimum for S. This value is obtained from (5.10) and (5.11):

$$\sum_{i=0}^{n} p_m(x_i) x_i^k = \sum_{i=0}^{n} y_i x_i^k \qquad k = 0, 1, \ldots, m. \tag{5.12}$$

The system in (5.12) consists of $m + 1$ equations in the $m + 1$ unknowns a_k; the a_k are thus determined uniquely.

From (5.9) we obtain a more explicit form for (5.12):

$$\begin{aligned}
a_0(n+1) + a_1 \sum_i x_i + \cdots + a_m \sum_i x_i^m &= \sum_i y_i && (k = 0) \\
a_0 \sum_i x_i + a_1 \sum_i x_i^2 + \cdots + a_m \sum_i x_i^{m+1} &= \sum_i x_i y_i && (k = 1) \\
&\vdots \\
a_0 \sum_i x_i^m + a_1 \sum_i x_i^{m+1} + \cdots + a_m \sum_i x_i^{2m} &= \sum_i x_i^m y_i && (k = m)
\end{aligned}$$

To fit a straight line, for example, we have $m = 1$ and

$$p_1(x) = a_0 + a_1 x$$

and the coefficients a_k are obtained from

$$\begin{aligned}
a_0(n+1) + a_1 \sum_i x_i &= \sum_i y_i \\
a_0 \sum_i x_i + a_1 \sum_i x_i^2 &= \sum_i x_i y_i
\end{aligned} \tag{5.13}$$

Variance A question arises regarding the order of the fitting polynomial. If we increase the degree of the polynomial to n (number of data points $= n+1$) then the fitting polynomial becomes an interpolating polynomial, which has zero deviation (since an interpolating polynomial passes through each point). How may we measure the quality of the fit for various degrees? Mathematical statistics tells us that we choose the degree for which the variation σ^2 shows a minimum, where

$$\sigma^2 = \frac{\sum_i \Delta_i^2}{n-m}.$$

Example 5.2. Fit a straight line to the data points in the following table:

i	x_i	y_i
0	1	2.04
1	2	4.12
2	3	5.64
3	4	7.18
4	5	9.20
5	6	12.04

Using the data in the table, we find

$$n+1 = 6$$
$$\sum_i x_i = 21$$
$$\sum_i y_i = 40.22$$
$$\sum_i x_i^2 = 91$$
$$\sum_i x_i y_i = 174.16$$

Substitution into equation (5.13) gives the two simultaneous equations

$$6a_0 + 21a_1 = 40.22$$
$$21a_0 + 91a_1 = 174.16$$

which are solved to give

$$a_0 = 0.0253$$
$$a_1 = 1.9080$$

The fit is thus

$$p_1(x) = 1.9080x + 0.0253.$$

Example 5.3. In the following table the coordinates of six data points are given. Fit polynomials of degrees 1 to 4 to this data, and decide which is the best fit.

x	0	0.8	1.4	2.1	2.7	3.4
y	0.015	0.644	1.926	4.442	7.274	11.621

The fitting polynomials, and variance for each, are given in the following table:

Degree	Fitting polynomial	σ^2
1	$p_1(x) = -1.601 + 3.416x$	2.1
2	$p_2(x) = 0.018 - 0.043x + 1.016x^2$	0.0010
3	$p_3(x) = 0.015 - 0.023x + 0.999x^2 + 0.003x^3$	0.0014
4	$p_4(x) = 0.017 - 0.113x + 1.142x^2 - 0.064x^3 + 0.010x^4$	0.0026

It is clear that the variance for all those with degree of two or higher are of the same order of magnitude, and that the optimal fit is the one with degree two.

5.6 Approximation with Chebyshev polynomials

One disadvantage of polynomial approximation is related to the fact that polynomial maxima and minima are spread unevenly on any interval: On $[-1, 1]$, $\left|x^k\right|$ has maxima at -1 and 1 and a minimum at 0. We now seek related functions that have evenly spaced maxima and minima, and for which the maxima and minima over a given interval are as small as possible. Possible candidates are, for example, the cosine functions: $\cos\theta, \cos 2\theta, \cos 3\theta, \ldots$.

5.6.1 Definition

The Chebyshev polynomial of the nth degree is defined by

$$T_n(x) = \cos(n \arccos x). \tag{5.14}$$

Here n is an integer and only the cases with $n \geq 0$ need to be studied, since it follows from the definition that $T_n = T_{-n}$.

We note that $T_n(x)$ is defined on $[-1, 1]$ only, due to the presence of the arccos function in (5.14). From the definition it follows

$$T_0(x) = 1 \qquad \text{and} \qquad T_1(x) = x.$$

Higher order Chebyshev polynomials are generated using a recursion formula, given by

$$T_{m+n}(x) + T_{m-n}(x) = 2T_m(x)T_n(x).$$

For the particular case $n = 1$ it follows that

$$T_{m+1}(x) = 2xT_m(x) - T_{m-1}(x) \tag{5.15}$$

and we find, for example,

$$\begin{aligned} T_2(x) &= 2xT_1(x) - T_0(x) \\ &= 2x^2 - 1 \\ T_3(x) &= 2xT_2(x) - T_1(x) \\ &= 4x^3 - 3x \end{aligned} \tag{5.16}$$

The relationships in (5.15) and (5.16) may be inverted to give powers of x in terms of T_n:

$$\begin{aligned} 1 &= T_0 \\ x &= T_1 \\ x^2 &= \frac{1}{2}(T_2 + 1) = \frac{T_2 + T_0}{2} \\ x^3 &= \frac{1}{4}(T_3 + 3x) = \frac{T_3 + 3T_1}{4} \end{aligned} \tag{5.17}$$

Zero points: From definition (5.14) it follows that

$$T_n(\xi) = 0 \qquad \text{for} \qquad n \arccos\, \xi = (2r+1)\frac{\pi}{2},$$

where r is an *integer*, and thus the zero points of $T_n(x)$ occur at

$$\xi_r = \cos\left(\frac{2r+1}{2n}\pi\right), \qquad r = 0, 1, \ldots, n-1. \tag{5.18}$$

We note that T_n has n zero points.

Extreme points: From definition (5.14) it follows that

$$T_n(x) = \pm 1 \qquad \text{for} \qquad n \arccos x = r\pi,$$

where r is an *integer* and thus we have the extreme points

$$x_r = \cos\left(\frac{r}{n}\pi\right), \qquad r = 0, 1, \ldots, n, \tag{5.19}$$

of T_n. We note that T_n has $n+1$ extreme points and $T_n(x_r) = (-1)^r$.

Orthogonality: Consider the integral

$$I_{mn} = \int_{-1}^{1} \frac{T_m(x)T_n(x)}{\sqrt{1-x^2}}\,\mathrm{d}x.$$

If we make the substitution

$$\begin{aligned} x &= \cos\theta \\ \mathrm{d}x &= -\sin\theta\,\mathrm{d}\theta \\ \sqrt{1-x^2} &= \sin\theta \end{aligned}$$

the integral becomes

$$\begin{aligned} I_{mn} &= -\int_{\pi}^{0} \frac{\cos m\theta \cos n\theta}{\sin\theta}\sin\theta\,\mathrm{d}\theta \\ &= \int_{0}^{\pi} \cos m\theta \cos n\theta\,\mathrm{d}\theta \\ &= \frac{1}{2}\int_{0}^{\pi} \big[\cos\big((m+n)\theta\big) + \cos\big((m-n)\theta\big)\big]\,\mathrm{d}\theta \end{aligned}$$

There are three distinct possibilities as far as the choice of m and n is concerned.

1. If $m = n = 0$, then
$$I_{mn} = \frac{1}{2}\int_{0}^{\pi} (1+1)\,\mathrm{d}\theta = \pi.$$
2. If $m = n \neq 0$, then
$$I_{mn} = \frac{1}{2}\int_{0}^{\pi} [\cos(2n\theta) + 1]\,\mathrm{d}\theta = \frac{\pi}{2}.$$
3. If $m \neq n$, then
$$I_{mn} = \frac{1}{2}\left[\frac{\sin(m+n)\theta}{m+n} + \frac{\sin(m-n)\theta}{m-n}\right]_0^{\pi} = 0.$$

In summary,

$$\int_{-1}^{1} \frac{T_m(x)T_n(x)}{\sqrt{1-x^2}}\,\mathrm{d}x = \begin{cases} \pi & m = n = 0 \\ \frac{\pi}{2} & m = n \neq 0 \\ 0 & m \neq n \end{cases} \tag{5.20}$$

Thus, the Chebyshev polynomials are said to be orthogonal with respect to the weight function $\frac{1}{\sqrt{1-x^2}}$ on the interval $[-1, 1]$.

5.6.2 Minimal property

From equations (5.16) it should be clear that the coefficient of x^n in $T_n(x)$ is equal to 2^{n-1}. The importance of Chebyshev polynomials arises from the following remarkable result:

Theorem 5.1. Consider the set $\mathcal{P}_n$ of all polynomials of degree n and with the coefficient of x^n equal to 1, on the interval $[-1, 1]$. If $p_n(x) \in \mathcal{P}_n$, then

$$\alpha_n = \max_{x\in[-1,1]} |p_n(x)|$$

is a minimum if

$$p_n(x) = \frac{1}{2^{n-1}} T_n(x).$$

Proof. Assume that there is a polynomial $p_n(x)$ with the coefficient of x^n equal to 1, that is $p_n(x) \in \mathcal{P}_n$, and for which

$$\alpha_n = \max_{x\in[-1,1]} |p_n(x)| < \left| \frac{T_n(x)}{2^{n-1}} \right| \leq \frac{1}{2^{n-1}}$$

everywhere on $[-1, 1]$. The last inequality follows from definition (5.14) where $|T_n(x)| \leq 1$ and so

$$\left| \frac{1}{2^{n-1}} T_n(x) \right| \leq \frac{1}{2^{n-1}} \qquad \text{on } [-1, 1].$$

For the extreme points x_r, $r = 0, 1, \ldots, n$, from (5.19) we have

$$\begin{aligned} 2^{-n+1}T_n(x_0) - p_n(x_0) &> 0 \\ 2^{-n+1}T_n(x_1) - p_n(x_1) &< 0 \\ &\vdots \\ 2^{-n+1}T_n(x_n) - p_n(x_n) &> 0 \end{aligned}$$

for n even (the inequalities reverse for n odd). The function

$$F(x) = 2^{-n+1}T_n(x) - p_n(x)$$

thus changes sign n times on $[-1, 1]$ and so must have n roots. But, by construction, $F(x)$ is a polynomial of degree $n - 1$. Thus, the assumption regarding the existence of p_n is thus incorrect. Therefore, if $p_n(x) \in \mathcal{P}_n$, then

$$\alpha_n \geq \frac{T_n(x)}{2^{n-1}}$$

with equality only if α_n is a minimum. Q.E.D.

We may use theorem 5.1 to reduce the error in polynomial interpolation. If $p_n(x)$ is an interpolating polynomial that approximates a function $y(x)$ defined on $[-1, 1]$, recall that the error for polynomial interpolation is given by

$$y(x) - p_n(x) = \frac{y^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n} (x - x_i).$$

It follows that

$$\max_{x\in[-1,1]} |y(x) - p_n(x)| \leq \frac{1}{(n+1)!} \max_{x\in[-1,1]} \left| y^{(n+1)}(x) \right| \max_{x\in[-1,1]} \left| \prod_{i=0}^{n} (x - x_i) \right|.$$

Clearly, $\prod_{i=0}^{n}(x-x_i)$ is an element of $\mathcal{P}_n$ and therefore

$$\max_{x\in[-1,1]}\left|\prod_{j=0}^{n}(x-x_j)\right| \geq \frac{1}{2^{n-1}}.$$

One notices that the nodes x_i are the roots of the polynomial $\prod_{i=0}^{n}(x-x_j)$, and we may write

$$\frac{T_n(x)}{2^{n-1}} = \prod_{i=0}^{n}(x-\bar{x}_i)$$

where $\bar{x}_i$ are the zeroes of $T_n(x)$. We can thus better our interpolation if we replace the nodes x_i with the Chebyshev nodes $\bar{x}_i$.

We are not restricted to the interval $[-1,1]$ and can apply this minimisation on an interval $[a,b]$ by the linear change of variable

$$x_j = \frac{\left[(a+b)+(b-a)\bar{x}_j\right]}{2}$$

for $j = 0, 1, \ldots, n-1$.

Another application of theorem 5.1 is reducing the degree of a known approximation polynomial. Let $p_n(x) = a_0 + a_1x + \cdots + a_{n-1}x^{n-1} + a_nx^n$ be a nth degree interpolating polynomial approximating the function $y(x)$. Consider now a $(n-1)$th degree polynomial $p_{n-1}(x) = b_0 + b_1x + \cdots + b_{n-1}x^{n-1}$. We want $\max_{x\in[-1,1]}|p_n(x) - p_{n-1}(x)|$ to be as small as possible. Note that

$$\frac{p_n(x)-p_{n-1}(x)}{a_n} = \left(\frac{a_0-b_0}{a_n}\right) + \left(\frac{a_1-b_1}{a_n}\right)x + \cdots + \left(\frac{a_{n-1}-b_{n-1}}{a_n}\right)x^{n-1} + x^n$$

is indeed a monic polynomial. Since

$$\max_{x\in[-1,1]}\left|\frac{p_n(x)-p_{n-1}(x)}{a_n}\right| \geq \max_{x\in[-1,1]}\left|\frac{T_n(x)}{2^{n-1}}\right|$$

and the minimum will be obtained when we have equality, that is

$$\frac{p_n(x)-p_{n-1}(x)}{a_n} = \frac{T_n(x)}{2^{n-1}},$$

we can thus choose

$$p_{n-1}(x) = p_n(x) - \frac{a_n}{2^{n-1}}T_n(x).$$

5.6.3 Expansion of a function in terms of Chebyshev polynomials

We investigate the possibility of approximating a given function using an nth order expansion in terms of Chebyshev polynomials, i.e.

$$f(x) \approx \frac{1}{2}c_0T_0(x) + c_1T_1(x) + c_2T_2(x) + \cdots + c_nT_n(x). \tag{5.21}$$

There are two ways to obtain such an expansion, and we will illustrate each by means of an example.

Method 1: "Economization" of a power series. If the powers of x in a known series expansion are expressed in terms of Chebyshev polynomials and the series is truncated after the term $c_{n-1}T_{n-1}(x)$, then the resulting error will essentially be given by $c_nT_n(x)$ and so will oscillate between c_n and $-c_n$.

Example 5.4. Economize the Taylor series for the exponential function

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \frac{x^5}{5!} + \cdots .$$

We use (5.17) to write the powers of x in terms of T_n, i.e.

$$e^x = T_0 + T_1 + \frac{1}{2}\left(\frac{T_0 + T_2}{2}\right) + \frac{1}{6}\left(\frac{3T_1 + T_3}{4}\right) + \frac{1}{24}\left(\frac{3T_0 + 4T_2 + T_4}{8}\right) + \frac{1}{120}\left(\frac{10T_1 + 5T_3 + T_5}{16}\right) + \cdots$$

Next we collect terms in T_n for the first six terms, and truncate after T_3 to obtain

$$e^x \approx \frac{81}{64}T_0 + \frac{217}{192}T_1 + \frac{13}{48}T_2 + \frac{17}{384}T_3.$$

Finally use (5.16) to write the T_n in terms of powers of x:

$$e^x \approx \frac{191}{192} + \frac{383}{384}x + \frac{13}{24}x^2 + \frac{17}{96}x^3 = p(x).$$

If we truncate the Taylor series after the 3rd-order term, we obtain

$$e^x \approx \frac{192}{192} + \frac{384}{384}x + \frac{12}{24}x^2 + \frac{16}{96}x^3 = q(x).$$

It is clear that $p(x)$ and $q(x)$ are almost identical, but it is the subtle difference between the two that significantly affects the quality of the approximations. The error for these two approximations on $[-1, 1]$ is shown in the following table.

$\boldsymbol{x}$	$\lvert e^{\boldsymbol{x}} - \boldsymbol{p}(\boldsymbol{x})\rvert$	$\lvert e^{\boldsymbol{x}} - \boldsymbol{q}(\boldsymbol{x})\rvert$
0	0.005	0
0.05	0.005	10^{-7}
0.1	0.005	10^{-6}
0.2	0.004	0.0001
0.5	0.002	0.003
0.8	0.005	0.02
1	0.007	0.05

It is clear that even though the Taylor series is better for small values of x, the Chebyshev series seems to "spread" the error evenly over the whole interval.

Method 2: Direct application of the orthogonality property. We multiply (5.21) by $\frac{T_k(x)}{\sqrt{1-x^2}}$ and integrate over $[-1, 1]$. From (5.20) it follows that

$$\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}\,\mathrm{d}x = 0 + 0 + \cdots + \frac{\pi}{2}c_k + \cdots + 0$$

and so

$$c_k = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}\,\mathrm{d}x.$$

The substitution $x = \cos\theta$ together with the definition (5.14) gives a more compact form for the integral

$$c_k = \frac{2}{\pi}\int_0^{\pi} f(\cos\theta)\cos(k\theta)\,\mathrm{d}\theta$$

An obvious disadvantage of this method is that sometimes difficult integrals may have to be determined.

Example 5.5. Expand $f(x) = \arcsin x$ in terms of Chebyshev polynomials.

We find the coefficients by

$$\begin{aligned} c_k &= \frac{2}{\pi}\int_0^\pi \arcsin(\cos\theta)\cos(k\theta)\,\mathrm{d}\theta \\ &= \frac{2}{\pi}\int_0^\pi \arcsin\left(\sin\left(\frac{\pi}{2}-\theta\right)\right)\cos(k\theta)\,\mathrm{d}\theta \\ &= \frac{2}{\pi}\int_0^\pi \left(\frac{\pi}{2}-\theta\right)\cos(k\theta)\,\mathrm{d}\theta \end{aligned}$$

We identify two cases:

1. If $k = 0$, then
$$c_0 = \frac{2}{\pi}\int_0^\pi \left(\frac{\pi}{2}-\theta\right)\,\mathrm{d}\theta = \frac{2}{\pi}\left[\frac{\pi}{2}\theta - \frac{\theta^2}{2}\right]_0^\pi = 0.$$

2. If $k \neq 0$, then
$$\begin{aligned} c_k &= \left[\frac{\sin(k\theta)}{k}\right]_0^\pi - \frac{2}{\pi}\left(\left[\frac{\theta\sin(k\theta)}{k}\right]_0^\pi - \int_0^\pi \frac{\sin(k\theta)}{k}\,\mathrm{d}\theta\right) \\ &= -\frac{2}{\pi}\left[\frac{\cos(k\theta)}{k^2}\right]_0^\pi = \frac{2}{\pi k^2}\left[1-(-1)^k\right]. \end{aligned}$$

   It follows that $c_k = 0$ when k is even and $c_k = \frac{4}{\pi k^2}$ when k is odd.

We thus have

$$\begin{aligned} \arcsin x &= \frac{4}{\pi}\sum_{k=0}^{\infty}\frac{1}{(2k+1)^2}T_{2k+1}(x) \\ &= \frac{4}{\pi}\left[T_1(x) + \frac{T_3(x)}{9} + \frac{T_5(x)}{25} + \cdots\right]. \end{aligned}$$

If we truncated this series after j terms (i.e. when $k = j-1$), the last term in the series is

$$\frac{4T_{2j-1}(x)}{\pi(2j-1)^2}.$$

We know that $\max|T_{2j-1}| = 1$, so the residual term R of the series, i.e. the sum of all the terms for $k \geq j$, satisfies

$$\begin{aligned} |R| &= \frac{4}{\pi}\sum_{k=j}^{\infty}\frac{1}{(2k+1)^2}\left|T_{2k+1}(x)\right| \\ &\leq \frac{4}{\pi}\sum_{k=j}^{\infty}\frac{1}{(2k+1)^2} \\ &\leq \frac{4}{\pi}\left|\int_j^\infty \frac{1}{(2x+1)^2}\,\mathrm{d}x\right| \\ &= \frac{2}{(2j+1)\pi}. \end{aligned}$$

Hence, if a tolerance of ϵ was imposed, we would have

$$\begin{aligned} |R| &\leq \epsilon \\ \frac{2}{(2j+1)\pi} &\leq \epsilon \\ \therefore\ j &\geq \frac{1}{\pi\epsilon} - \frac{1}{2} \end{aligned}$$

and so we see that if ϵ is very small, then j must be very large for the desired tolerance to be satisfied.

Chapter 6

NUMERICAL DIFFERENTIATION

6.1 Introduction

Suppose we wish to estimate the derivative of a function $f(x)$ at some point x, given only discrete data points $(x_i, f(x_i))$. One approach would be to determine an interpolating or fitting polynomial, and then differentiate that polynomial analytically. However, it is also possible to estimate the derivative using a direct numerical method.

6.2 First derivative

Consider the Taylor expansions

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) - \cdots \tag{6.1}$$

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + \cdots \tag{6.2}$$

Subtracting (6.1) from (6.2) yields

$$f(x+h) - f(x-h) = 2hf'(x) + \frac{2h^3}{3!}f'''(x) + \cdots \tag{6.3}$$

and we truncate the infinite series on the right of the equality to give the approximation

$$f(x+h) - f(x-h) \approx 2hf'(x)$$

which we can rewrite as

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \tag{6.4}$$

This is the *finite-difference* representation of the first derivative of $f(x)$ at x.

Note: The truncated terms $\frac{h^3}{3!}f'''(x) + \cdots$ from (6.3) constitute the approximation error, i.e.

$$\left| f'(x) - \frac{f(x+h) - f(x-h)}{2h} \right| = \left| \frac{2h^2}{3!}f'''(x) + \cdots \right|$$

The error would tend toward zero as $h \to 0$ and (6.4) would become exact. Since we are working with Taylor expansions, we may replace the error by the Lagrange estimate of the error

$$\left| \frac{2h^2}{3!}f'''(x) + \frac{2h^4}{5!}f^{(5)}(x) + \cdots \right| \leq \left| \frac{2h^2}{3!}f'''(\xi) \right|$$

where $x < \xi < x + h$. Due to the function being unknown, we cannot determine its higher-order derivatives analytically and it would be difficult to determine an upper bound on the Lagrange estimate of the error and therefore we analyse the behaviour of h as it is changed. If $0 < h < 1$, then $\left|a_0h^2 + a_1h^4 + a_2h^6 + \cdots\right| \leq M\left|h^2\right|$ for some positive constant $M \in \mathbb{R}$. We introduce the notation $O(\cdot)$ to indicate this boundedness; the approximation error above can then be rewritten as

$$\left|\frac{2h^2}{3!}f'''(x) + \frac{2h^4}{5!}f^{(5)}(x) + \cdots\right| \in O(h^2).$$

The behaviour of the error pending changes in the step size can then be illustrated by the following example. Suppose that the approximation error is given by $e(h) = 10h^2$ and thus $e(h) \in O(h^2)$ by the previous statements. If $h = 0.1$, then $e\,(0.1) = 0.1$. If the step size were to be halved $h^* = h/2 = 0.05$, then the error becomes $e(0.05) = 0.025$; halving the error further yields $e(0.025) = 0.00625$. Therefore, as the step size is decreased the error will also decrease proportional to h^2. Since the error in (6.4) is of order h^2, we require that h be reasonably small in order for (6.4) to be reasonably accurate.

6.3 Second derivative

If we add (6.1) and (6.2) we obtain

$$f(x-h) + f(x+h) = 2f(x) + h^2f''(x) + O(h^4)$$

which yields

$$f''(x) = \frac{f(x+h) + f(x-h) - 2f(x)}{h^2} \tag{6.5}$$

with an error of $O(h^2)$. This is the *finite-difference* representation of the second derivative of $f(x)$ at x. Since the error is $O(h^2)$ we would expect that, if h is small enough for (6.4) to be accurate, then (6.5) will also be accurate. Both (6.4) and (6.5) are known as *central-difference formulae*.

Note: If we use the notation $x_{i-1} = x - h$, $x_i = x$, $x_{i+1} = x + h$ we have

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{x_{i+1} - x_{i-1}}. \tag{6.6}$$

Using the subscripted notation we have

$$f''(x_i) = 4\left(\frac{f(x_{i+1}) + f(x_{i-1}) - 2f(x_i)}{(x_{i+1} - x_{i-1})^2}\right). \tag{6.7}$$

We can simplify the notation in (6.6) and (6.7) more by letting $y_i = f(x_i)$, $y_{i+1} = f(x_{i+1})$, $y_{i-1} = f(x_{i-1})$, etc. Then (6.6) becomes

$$y_i' = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}} = \frac{y_{i+1} - y_{i-1}}{2h}$$

and (6.7) becomes

$$y_i'' = \frac{y_{i+1} + y_{i-1} - 2y_i}{h^2}.$$

Note: The approximation formula for $f'(x_i)$ is dependent on the values of $f(x)$ at the neighbouring points x_{i-1} and x_{i+1} and that the approximation formula for $f''(x_i)$ is dependent on the values of $f(x)$ at x_{i-1}, x_i and x_{i+1}. Since the approximation error is proportional to h^2, we see that as h is reduced so the error is reduced. The converse is also true: if h is large then the error will be large. Numerical differentiation must thus be used with caution, particularly if h is large (the discrete data points are far apart).

6.4 Higher-order derivatives

There are higher-order expressions that may be derived. We will not go into the details of their derivation (the principles are the same as for the expressions above), but we will state them for completeness.

Central-difference, $O(h^2)$

$$
\begin{aligned}
y_i' &= \frac{y_{i+1} - y_{i-1}}{2h} \\
y_i'' &= \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} \\
y_i''' &= \frac{y_{i+2} - 2y_{i+1} + 2y_{i-1} - y_{i-2}}{2h^3} \\
y_i^{(4)} &= \frac{y_{i+2} - 4y_{i+1} + 6y_i - 4y_{i-1} + y_{i-2}}{h^4}
\end{aligned}
$$

Central-difference, $O(h^4)$

$$
\begin{aligned}
y_i' &= \frac{-y_{i+2} + 8y_{i+1} - 8y_{i-1} + y_{i-2}}{12h} \\
y_i'' &= \frac{-y_{i+2} + 16y_{i+1} - 30y_i + 16y_{i-1} - y_{i-2}}{12h^2} \\
y_i''' &= \frac{-y_{i+3} + 8y_{i+2} - 13y_{i+1} + 13y_{i-1} - 8y_{i-2} + y_{i-3}}{8h^3} \\
y_i^{(4)} &= \frac{-y_{i+3} + 12y_{i+2} - 39y_{i+1} + 56y_i - 39y_{i-1} + 12y_{i-2} - y_{i-3}}{6h^4}
\end{aligned}
$$

Forward-difference, $O(h)$

$$
\begin{aligned}
y_i' &= \frac{y_{i+1} - y_i}{h} \\
y_i'' &= \frac{y_{i+2} - 2y_{i+1} + y_i}{h^2} \\
y_i''' &= \frac{y_{i+3} - 3y_{i+2} + 3y_{i+1} - y_i}{h^3} \\
y_i^{(4)} &= \frac{y_{i+4} - 4y_{i+3} + 6y_{i+2} - 4y_{i+1} + y_i}{h^4}
\end{aligned}
$$

Forward-difference, $O(h^2)$

$$
\begin{aligned}
y_i' &= \frac{-y_{i+2} + 4y_{i+1} - 3y_i}{2h} \\
y_i'' &= \frac{-y_{i+3} + 4y_{i+2} - 5y_{i+1} + 2y_i}{h^2} \\
y_i''' &= \frac{-3y_{i+4} + 14y_{i+3} - 24y_{i+2} + 18y_{i+1} - 5y_i}{2h^3} \\
y_i^{(4)} &= \frac{-2y_{i+5} + 11y_{i+4} - 24y_{i+3} + 26y_{i+2} - 14y_{i+1} + 3y_i}{h^4}
\end{aligned}
$$

Backward-difference, $O(h)$

$$
\begin{aligned}
y_i' &= \frac{-y_{i-1} + y_i}{h} \\
y_i'' &= \frac{y_{i-2} - 2y_{i-1} + y_i}{h^2} \\
y_i''' &= \frac{-y_{i-3} + 3y_{i-2} - 3y_{i-1} + y_i}{h^3} \\
y_i^{(4)} &= \frac{y_{i-4} - 4y_{i-3} + 6y_{i-2} - 4y_{i-1} + y_i}{h^4}
\end{aligned}
$$

Backward-difference, $O(h^2)$

$$y_i' = \frac{y_{i-2} - 4y_{i-1} + 3y_i}{2h}$$
$$y_i'' = \frac{-y_{i-3} + 4y_{i-2} - 5y_{i-1} + 2y_i}{h^2}$$
$$y_i''' = \frac{3y_{i-4} - 14y_{i-3} + 24y_{i-2} - 18y_{i-1} + 5y_i}{2h^3}$$
$$y_i^{(4)} = \frac{-2y_{i-5} + 11y_{i-4} - 24y_{i-3} + 26y_{i-2} - 14y_{i-1} + 3y_i}{h^4}$$

The forward-difference and backward-difference formulae should only be used when data points to the left or right of x_i are not available. The central-difference expressions generally give better results.

Example 6.1. The following table shows discrete data points corresponding to the function $\sin x$ on the interval $[0, \frac{\pi}{2}]$. We have used $h = \frac{\pi}{20}$.

i	x_i	$\sin x_i$
1	0	0
2	0.15707	0.15643
3	0.31415	0.30901
4	0.47123	0.45399
5	0.62831	0.58778
6	0.78539	0.70710
7	0.94247	0.80901
8	1.09955	0.89100
9	1.25663	0.98768
10	1.41371	0.98768
11	1.57079	1

In the following table we show the various derivatives calculated using the various expressions given above, at the point $x = \frac{\pi}{4}$.

	$y'\left(\frac{\pi}{4}\right)$	$y''\left(\frac{\pi}{4}\right)$	$y'''\left(\frac{\pi}{4}\right)$	$y^{(4)}\left(\frac{\pi}{4}\right)$
Exact	0.70710	−0.70710	−0.70710	0.70710
CD $O(h^2)$	0.70420	−0.70565	−0.70275	0.70420
CD $O(h^4)$	0.70709	−0.70710	−0.70708	0.70709
FD $O(h)$	0.64878	−0.80735	−0.52088	0.88734
FD $O(h^2)$	0.71219	−0.72553	−0.72996	0.76774
BD $O(h)$	0.75962	−0.58657	−0.85001	0.45212
BD $O(h^2)$	0.71355	−0.72009	−0.74348	0.74088

The central-difference expressions give better results, and the $O(h^4)$ results have an error of better than 10^{-4}. Also, the results for the $O(h^2)$ forward-difference and backward difference expressions are considerably better than those for the corresponding $O(h)$ expressions.

Chapter 7

NUMERICAL INTEGRATION

7.1 Introduction

Definite integrals may be determined *analytically* by the following methods:

(a) Direct integration.

(b) By means of substitution.

(c) Integration by parts.

(d) Using a recursion formula.

(e) Integration in the complex plane.

(f) Series expansion of the integrand.

If none of these methods can be used, then a *numerical method* may be implemented. In this chapter, we consider two techniques based on polynomial interpolation. We also consider the approximation error associated with these techniques, to the extent that we are able to control the accuracy of the approximation.

7.2 Newton-Cotes formulae

The Newton-Cotes formulae for numerical integration, also called interpolatory quadrature, are derived by using *interpolatory polynomial approximations of the integrand*.

Suppose a function $y = f(x)$ is known everywhere on $[x_0, x_n]$. The interval $[x_0, x_n]$ is now divided into n equally sized subintervals, each of length h where

$$h = \frac{x_n - x_0}{n}.$$

The function $y = f(x)$ is approximated by the interpolating polynomial $p_n(x)$ that passes through the points (x_k, y_k), where

$$x_k = x_0 + kh$$
$$y_k = f(x_k)$$

and $k = 0, 1, \ldots, n$. The integral of $p_n(x)$ over $[x_0, x_n]$ is taken as an approximation of $\int_{x_0}^{x_n} f(x)\,\mathrm{d}x$.

Recall from (5.6) we have

$$p_n(x) = \sum_{k=0}^{n} y_k L_k(x)$$

where

$$L_k(x) = \frac{(x - x_0)(x - x_1)\cdots(x - x_{k-1})(x - x_{k+1})\cdots(x - x_n)}{(x_k - x_0)(x_k - x_1)\cdots(x_k - x_{k-1})(x_k - x_{k+1})\cdots(x_k - x_n)}.$$

Since $x_k = x_0 + kh$, it is natural to make the substitution $x = x_0 + sh$, where s is a continuous variable:

$$L_k(s) = \frac{sh(sh-h)\cdots\big[sh-(k-1)h\big]\big[sh-(k+1)h\big]\cdots(sh-nh)}{kh(kh-h)\cdots\big[kh-(k-1)h\big]\big[kh-(k+1)h\big]\cdots(kh-nh)}$$

Both the numerator and denominator are divided by h^n

$$L_k(s) = \frac{s(s-1)\cdots(s-k+1)(s-k-1)\cdots(s-n)}{k(k-1)\cdots(1)(-1)\cdots(k-n)}$$

From the substitution we also have $\mathrm{d}x = h\,\mathrm{d}s$, and so

$$\int_{x_0}^{x_n} p_n(x)\,\mathrm{d}x = \int_0^n \left(\sum_{k=0}^{n} y_k L_k(s)\right) h\,\mathrm{d}s.$$

Since integration is a linear operation the order of integration and summation may be swapped, so that

$$\int_{x_0}^{x_n} p_n(x)\,\mathrm{d}x = (nh)\frac{1}{n}\sum_{k=0}^{n} y_k \int_0^n L_k(s)\,\mathrm{d}s$$

and hence

$$\int_{x_0}^{x_n} f(x)\,\mathrm{d}x \approx \int_{x_0}^{x_n} p_n(x)\,\mathrm{d}x = (x_n - x_0)\sum_{k=0}^{n} C_k^n y_k \tag{7.1}$$

where

$$C_k^n = \frac{1}{n}\int_0^n L_k(s)\,\mathrm{d}s$$

are the so-called *quadrature weights*, sometimes known as the *Cotes numbers*. We will study the *trapezium rule* ($n = 1$) and *Simpson's rule* ($n = 2$).

7.3 Trapezium rule

For $n = 1$ (*linear interpolation*) there are two Cotes numbers

$$C_0^1 = \int_0^1 L_0\,\mathrm{d}s = \int_0^1 \frac{s-1}{0-1}\,\mathrm{d}s = \frac{1}{2}$$
$$C_1^1 = \int_0^1 L_1\,\mathrm{d}s = \int_0^1 \frac{s-0}{1-0}\,\mathrm{d}s = \frac{1}{2}$$

From (7.1) we have

$$\begin{aligned}\int_{x_0}^{x_1} f(x)\,\mathrm{d}x &\approx \frac{1}{2}(x_1 - x_0)(y_0 + y_1)\\ &= \frac{1}{2}(x_1 - x_0)\big[f(x_0) + f(x_1)\big].\end{aligned} \tag{7.2}$$

This is simply the area of the trapezium formed by the interpolation polynomial, the x-axis and the x-intercepts on $[x_0, x_1]$.

If $f(x)$ is nonlinear we might expect the approximation error to be large, particularly if the interval of integration is large. Therefore, to evaluate the integral $\int_a^b f(x)\,\mathrm{d}x$ in a manner which allows the error to be controlled, we subdivide the interval $[a, b]$ into N subintervals, each of length

$$h = \frac{b-a}{N} = \frac{x_N - x_0}{N} \tag{7.3}$$

and the linear approximation (7.2) is performed on *each* of these subintervals. With

$$x_k = x_0 + kh$$
$$y_k = f(x_k)$$

for $k = 0, 1, \ldots, N$, we have

$$\begin{aligned}\int_a^b f(x)\,\mathrm{d}x &\approx \frac{1}{2}h(y_0 + y_1) + \frac{1}{2}h(y_1 + y_2) + \cdots + \frac{1}{2}h(y_{N-1} + y_N) \\ &= \frac{h}{2}(y_0 + 2y_1 + 2y_2 + \cdots + 2y_{N-1} + y_N) \end{aligned} \tag{7.4}$$

This is known as the *composite trapezium rule*. Equation (7.4) may also be written as

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}\left(y_0 + y_N + 2\sum_{j=1}^{N-1} y_j\right).$$

Approximation error: In order to study the approximation error in the composite trapezium rule, we first obtain an estimate for the error on the first subinterval in (7.4)

$$\Delta_1 = \int_a^{a+h} f(x)\,\mathrm{d}x - \frac{h}{2}\big[f(a) + f(a+h)\big]$$

A Taylor series expansion of $f(x)$ is made about $x = a$:

$$\begin{aligned}\Delta_1 &= \int_a^{a+h}\left[f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \cdots\right]\mathrm{d}x \\ &\quad - \frac{h}{2}\big[f(a) + f(a+h)\big] \\ &= [x]_a^{a+h} f(a) + \left[\frac{(x-a)^2}{2}\right]_a^{a+h} f'(a) + \left[\frac{(x-a)^3}{6}\right]_a^{a+h} f''(a) + \cdots \\ &\quad - \frac{h}{2}\big[f(a) + f(a+h)\big] \\ &= hf(a) + \frac{h^2}{2}f'(a) + \frac{h^3}{6}f''(a) + \cdots \\ &\quad - \frac{h}{2}\left[2f(a) + hf'(a) + \frac{h^2}{2}f''(a) + \cdots\right]\end{aligned}$$

To lowest order in h we have

$$\Delta_1 \approx -\frac{h^3}{12}f''(a).$$

For the total approximation error we have

$$\begin{aligned}|\Delta| &\le |\Delta_1| + |\Delta_2| + \cdots + |\Delta_N| \\ &\approx \frac{h^3}{12}\big(|f''(a)| + |f''(a+h)| + \cdots + |f''(a+(N-1)h)|\big) \\ &\le \frac{h^3}{12}NM\end{aligned}$$

where

$$M = \max_{x\in[a,b]} |f''(x)|\,.$$

Thus, we obtain an upper bound for the approximation error

$$|\Delta| \le \frac{h^3}{12}\left(\frac{b-a}{h}\right)M = \frac{h^2\,(b-a)\,M}{12}. \tag{7.5}$$

Since the computing time is determined by N, it follows from (7.3) that it is wise to make h as large as possible, for a given accuracy. The important term in (7.5) is the power of h. We see that the trapezium rule is a practical computational method, since $\Delta = O(h^2)$.

Example 7.1. Determine

$$I = \int_0^2 e^x \, \mathrm{d}x$$

correct to 4 decimal places.

From (7.5) we have that

$$|\Delta| = \frac{h^2}{12}(2-0)M \leq 10^{-4}$$

where

$$M = \max_{x\in[0,2]} \left| \frac{\mathrm{d}^2}{\mathrm{d}x^2} e^x \right| = e^2.$$

Thus, we obtain an upper bound for the size of each subinterval

$$h \leq \left(\frac{6 \times 10^{-4}}{e^2} \right)^{\frac{1}{2}} = 0.0090$$

and hence a lower bound for the number of subintervals

$$N \geq \frac{2}{h} = 222.22,$$

and so we choose $N = 223$, $h = \frac{2}{223}$. From (7.4) we have

$$\begin{aligned} I_{\text{trap}} &= \frac{1}{2}\left(\frac{2}{223}\right)\left(e^0 + 2e^{\frac{2}{223}} + 2e^{\frac{4}{223}} + \cdots + 2e^{\frac{444}{223}} + e^2\right) \\ &= 6.38910 \end{aligned}$$

Analytically, we find

$$I = e^2 - 1 = 6.38906$$

and we see that the approximation error is of the correct magnitude

$$|\Delta| = |I - I_{\text{trap}}| = 4 \times 10^{-5}.$$

7.4 Simpson's rule

For $n = 2$ (*quadratic interpolation*) there are three Cotes numbers:

$$\begin{aligned} C_0^2 &= \frac{1}{2}\int_0^2 \frac{(s-1)(s-2)}{(0-1)(0-2)}\,\mathrm{d}s = \frac{1}{6} \\ C_1^2 &= \frac{1}{2}\int_0^2 \frac{(s-0)(s-2)}{(1-0)(1-2)}\,\mathrm{d}s = \frac{4}{6} \\ C_2^2 &= \frac{1}{2}\int_0^2 \frac{(s-0)(s-1)}{(2-0)(2-1)}\,\mathrm{d}s = \frac{1}{6} \end{aligned}$$

Equation (7.1) gives the following approximation for the integral:

$$\int_{x_0}^{x_2} f(x)\,\mathrm{d}x \approx (x_2 - x_0)\left(\frac{1}{6}y_0 + \frac{4}{6}y_1 + \frac{1}{6}y_2\right) \tag{7.6}$$

Again, to determine $\int_a^b f(x)\,\mathrm{d}x$ with error control, it is necessary to subdivide $[a, b]$ into subintervals, and in this case, an even number $2N$. The size of each subinterval is

$$h = \frac{b-a}{2N}.$$

The quadratic approximation is performed on each pair of subintervals. With

$$x_k = x_0 + kh$$
$$y_k = f(x_k)$$

and $k = 0, 1, \ldots, 2N$, it follows from (7.6) that

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{2h}{6}\big[(y_0 + 4y_1 + y_2) + (y_2 + 4y_3 + y_4) + \cdots + (y_{2N-2} + 4y_{2N-1} + y_{2N})\big]. \tag{7.7}$$

This is known as the *composite Simpson's rule*. Equation (7.7) may also be written as

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{3}\left(y_0 + y_{2N} + 4y_1 + \sum_{k=1}^{N-1}(2y_{2k} + 4y_{2k+1})\right).$$

Approximation error: Similar to the procedure followed for the composite trapezium rule, we first find the approximation error on the first two subintervals:

$$\Delta_1 = \int_a^{a+2h} f(x)\,\mathrm{d}x - \frac{h}{3}\big[f(a) + 4f(a+h) + f(a+2h)\big]$$

We perform a Taylor expansion of the integrand, $f(a+h)$ and $f(a+2h)$:

$$\begin{aligned}
\Delta_1 = \int_a^{a+2h} &\left[f(a) + (x-a)f'(a) + \frac{(x-a)^2}{2!}f''(a) + \frac{(x-a)^3}{3!}f'''(a)\right.\\
&\left. + \frac{(x-a)^4}{4!}f^{(4)}(a) + \cdots\right]\,\mathrm{d}x - \frac{h}{3}\left[f(a) + 4\left(f(a) + hf'(a) + \frac{h^2}{2!}f''(a)\right.\right.\\
&\left. + \frac{h^3}{3!}f'''(a) + \frac{h^4}{4!}f^{(4)}(a) + \cdots\right) + \left(f(a) + 2hf'(a) + \frac{(2h)^2}{2!}f''(a)\right.\\
&\left.\left. + \frac{(2h)^3}{3!}f'''(a) + \frac{(2h)^4}{4!}f^{(4)}(a) + \cdots\right)\right]\\
= &\left(2hf(a) + 2h^2f'(a) + \frac{4h^3}{3}f''(a) + \frac{2h^4}{3}f'''(a) + \frac{4h^5}{15}f^{(4)}(a) + \cdots\right)\\
&- \frac{h}{3}\left(6f(a) + 6hf'(a) + 4h^2f''(a) + 2h^3f'''(a) + \frac{5h^4}{6}f^{(4)}(a) + \cdots\right)
\end{aligned}$$

The terms of order h to h^4 cancel identically, and we have to lowest order in h

$$\Delta_1 = -\frac{h^5}{90}f^{(4)}(a).$$

Thus, for the total approximation error on $[a, b]$, we have

$$\begin{aligned}
|\Delta| &\le |\Delta_1| + |\Delta_2| + \cdots + |\Delta_{2N}|\\
&= \frac{h^5}{90}\big(\left|f^{(4)}(a)\right| + \left|f^{(4)}(a+2h)\right| + \cdots + \left|f^{(4)}(a+(2N-2)h)\right|\big)\\
&\le \frac{h^5}{90}NK
\end{aligned}$$

where

$$K = \max_{x\in[a,b]}\left|f^{(4)}(x)\right|. \tag{7.8}$$

Since $hN = \frac{b-a}{2}$, it follows that

$$|\Delta| \le \frac{h^4(b-a)K}{180}. \tag{7.9}$$

We see that

$$\Delta = O(h^4)$$

and Simpson's method is clearly "faster" than the trapezium method, i.e. it will require fewer subintervals to achieve the same accuracy.

Example 7.2. Determine

$$I = \int_0^2 e^x \, \mathrm{d}x$$

correct to 4 decimal places.

From (7.8) we have

$$K = \max_{x \in [0,2]} e^x = e^2$$

and from (7.9) we have

$$h \leq \left(\frac{180\,|\Delta|}{K(b-a)} \right)^{\frac{1}{4}} = \left(\frac{180 \times 10^{-4}}{2e^2} \right)^{\frac{1}{4}} = 0.187$$

and so

$$2N \geq \frac{b-a}{h} = 10.7$$
$$\therefore \quad N \geq 5.35$$

We choose $N = 6$, $h = \frac{2}{12} = \frac{1}{6}$. Equation (7.7) then gives

$$\begin{aligned} \int_0^2 e^x \, \mathrm{d}x &\approx \frac{1}{18} \left[e^0 + e^2 + 4 \left(e^{\frac{1}{6}} + e^{\frac{3}{6}} + \cdots + e^{\frac{11}{6}} \right) + 2 \left(e^{\frac{2}{6}} + e^{\frac{4}{6}} + \cdots + e^{\frac{10}{6}} \right) \right] \\ &= 6.38908 \end{aligned}$$

We compare the Simpson approximation with the exact value of the integral

$$|\Delta| = |I - I_{\text{Simpson}}| = 2.7 \times 10^{-5}$$

We see that the error is as expected $\left(\leq 10^{-4}\right)$. We also see that Simpson's method only requires 13 evaluations of $f(x)$, whereas the Trapezium rule required 223 such evaluations.

7.5 An analytical complication

If $\int_a^b f(x)\,\mathrm{d}x$ exists but the integrand $f(x)$ does not exist somewhere on $[a, b]$, then the numerical evaluation of the integral requires particular care. We consider this case by way of an example.

Example 7.3. Determine

$$I = \int_0^1 x^{-\frac{1}{2}} \cos x \, \mathrm{d}x.$$

The integrand $f(x) = \frac{\cos x}{\sqrt{x}}$ does not exist at $x = 0$, and so the integral cannot be determined using either the Trapezium rule or Simpson's rule. However, the singularity at $x = 0$ is of the form $x^{-\frac{1}{2}}$ and the integral does indeed exist. To facilitate numerical evaluation we first integrate by parts:

$$I = \left[2x^{\frac{1}{2}} \cos x \right]_0^1 + \int_0^1 2x^{\frac{1}{2}} \sin x \, \mathrm{d}x = 2\cos 1 + \; I_1$$

where the integrand in

$$I_1 = \int_0^1 2x^{\frac{1}{2}} \sin x \, \mathrm{d}x$$

exists everywhere on $[0, 1]$ and so I_1 may be determined numerically.

Chapter 8

NUMERICAL METHODS FOR ORDINARY DIFFERENTIAL EQUATIONS

8.1 Introduction

Ordinary differential equations (ODEs) arise in the modelling of systems in the physical, engineering, biological, behavioural, economic and other sciences. The solution of such equations is a very important part of applied mathematics, and the analytical solution of such equations is an active and fruitful area of research. In practice, however, most ODEs are not solvable by analytical means, and so numerical methods must be used. In this chapter, we focus our attention on initial-value problems (IVPs) of the form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y), \qquad y(x_0) = y_0. \tag{8.1}$$

As an example, we describe a case that arises in biological research. An ecologist determines the following facts regarding an isolated field mouse population:

(a) The birth rate is proportional to the population N.

(b) For a constant food supply the mortality rate is proportional to $N^{1.7}$.

(c) The proportionality constant k varies with the food supply and is given as a function of time t in following table.

t (months)	1	2	3	4	5	6	7	8
k (months$^{-1} \times 10^{-4}$)	36	11	1	4	13	28	43	56

For $N = N(t)$ it follows from (a)-(c) that

$$\frac{dN}{dt} = aN - k(t)N^{1.7}. \tag{8.2}$$

Of course, this equation may only be solved uniquely if N is known at some specific time t_0. In other words, we need an initial value

$$N(t_0) = N_0.$$

It should be clear that (8.2) cannot be solved analytically and so a numerical method must be used to obtain N for any given value of t.

8.2 One-step methods

To solve (8.1) on a large interval, we subdivide the interval into n subintervals, each of length h (usually referred to as the *step size*), and then make the approximation

$$y(x_{m+1}) \approx y_{m+1} = y_m + hF(x_m, y_m), \qquad m = 0, 1, \ldots, n, \tag{8.3}$$

consecutively on each subinterval. We use a notation here that will appear throughout this chapter—the exact value of y at x_m is denoted $y(x_m)$ and the approximate value of y at x_m is denoted y_m. Note also that $y(x_0) = y_0$ because the initial value is known exactly.

In (8.3) the expression

$$y_{m+1} = y_m + hF(x_m, y_m) \tag{8.4}$$

is known as an *explicit one-step method*, where the function $F(x, y)$ is characteristic of the particular method.

In the next three sections we will describe various one-step methods, and their characteristic functions $F(x, y)$.

8.3 Euler's method

The solution $y = y(x)$ of (8.1) may be represented as a curve in the xy- plane. The Euler method uses a simple geometrical insight—over a small interval any curve may be approximated by a straight line. The function $y(x)$ is approximated by a straight line on the interval $[x_0, x_0 + h]$. The line equals $y(x_0)$ at x_0 and its slope is equal to that of $y(x)$ at x_0.

For small h the y-value y_1 of the tangent line at x_1 will, we assume, be a good *approximation* to that of the curve $y = y(x)$. From (8.1) we have

$$\frac{\Delta y}{h} = \left.\frac{\mathrm{d}y}{\mathrm{d}x}\right|_{(x_0,y_0)} = f(x_0, y_0)$$

and so

$$\Delta y = y_1 - y_0 = hf(x_0, y_0).$$

We thus have the approximation

$$y(x_1) \approx y_1 = y_0 + hf(x_0, y_0).$$

More generally, on the $(m + 1)$th subinterval we have

$$y_{m+1} = y_m + hF(x_m, y_m) \tag{8.5}$$

where

$$F(x_m, y_m) = f(x_m, y_m).$$

8.4 The modified Euler method

In principle, it would seem that greater accuracy could be achieved if we used the trapezium rule to integrate. We multiply (8.1) by $\mathrm{d}x$ to obtain

$$\mathrm{d}y = f(x, y)\,\mathrm{d}x$$

which we integrate and apply the trapezium rule to the RHS:

$$\begin{aligned}
\int_{y_0}^{y(x_1)} \mathrm{d}y &= \int_{x_0}^{x_1} f(x, y)\,\mathrm{d}x \\
y(x_1) - y_0 &= \int_{x_0}^{x_0+h} f(x, y)\,\mathrm{d}x \\
&\approx \frac{h}{2}\big[f(x_0, y_0) + f(x_1, y_1)\big]
\end{aligned}$$

We now have the approximation

$$y(x_1) \approx y_1 = y_0 + \frac{h}{2}\big[f(x_0, y_0) + f(x_1, y_1)\big].$$

The *unknown* y_1 appears on both sides of this equation and may, of course, be found numerically (using Newton's method, for example). In practice, however, we find that it is sufficient to make an approximation for y_1 on the RHS using the Euler method (8.5)

$$y_1^* = y_0 + hf(x_0, y_0)$$

The equation thus becomes

$$y(x_1) \approx y_1 = y_0 + \frac{h}{2}\big[f(x_0, y_0) + f(x_1, y_1^*)\big].$$

For the $(m+1)$th subinterval we have

$$\begin{aligned} y_{m+1}^* &= y_m + hf(x_m, y_m) \\ y_{m+1} &= y_m + \frac{h}{2}\big[f(x_m, y_m) + f(x_{m+1}, y_{m+1}^*)\big] \end{aligned} \tag{8.6}$$

This is the *modified Euler method.* The quantity y_{m+1}^* is sometimes referred to as the *predictor*, and y_{m+1} as the *corrector.*

The method may be written in the same form as (8.4) by letting

$$F(x_m, y_m) = \frac{f(x_m, y_m) + f(x_m + h, y_m + hf(x_m, y_m))}{2}.$$

and thus we obtain

$$y_{m+1} = y_m + h\left(\frac{f(x_m, y_m) + f(x_m + h, y_m + hf(x_m, y_m))}{2}\right).$$

8.5 The Runge-Kutta methods

In a Runge-Kutta method of the nth order, the function $y(x)$ in (8.1) is, in principle, written as a Taylor series of order n. We describe here two cases that are generally used.

8.5.1 The second-order Runge-Kutta method (RK2)

Consider the following attempt to approximate $y(x_1)$:

$$\begin{aligned} k_1 &= hf(x_0, y_0) \\ k_2 &= hf(x_0 + \alpha h, y_0 + \beta k_1) \\ y_1 &= y_0 + ak_1 + bk_2 \end{aligned} \tag{8.7}$$

We note that the term k_1 represents the *Euler approximation.* The term bk_2 may thus be regarded as an attempt to improve this approximation. The *four* unknowns a, b, α and β are obtained by requiring that (8.7) agrees as closely as possible with a Taylor expansion of 2nd-order:

$$\begin{aligned} y_1 &\approx y(x_1) \\ &= y(x_0 + h) \\ &= y(x_0) + hy'(x_0) + \frac{h^2}{2!}y''(x_0) + \cdots \\ &= y(x_0) + hf(x_0, y_0) + \frac{h^2}{2}\big[f_x + f_y f\big]_{(x_0, y_0)} + \cdots \end{aligned} \tag{8.8}$$

where we have used

$$y''(x) = \frac{\mathrm{d}f(x,y)}{\mathrm{d}x} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}x} = f_x + f_y f.$$

Equation (8.7) is also expanded to 2nd-order, using the multivariable Taylor series:

$$\begin{aligned} y_1 &= y_0 + ahf(x_0,y_0) + bhf(x_0+\alpha h, y_0+\beta k_1) \\ &= y_0 + ahf(x_0,y_0) + bh\big[f(x_0,y_0+\beta k_1) + \alpha h f_x(x_0,y_0+\beta k_1) + \cdots\big] \\ &= y_0 + ahf(x_0,y_0) + bh\Big[\big(f(x_0,y_0) + \beta k_1 f_y(x_0,y_0) + \cdots\big) \\ &\quad + \alpha h\big(f_x(x_0,y_0) + \beta k_1 f_{xy}(x_0,y_0) + \cdots\big)\Big] \\ &= y_0 + (a+b)hf(x_0,y_0) + h^2\big[\beta b f_y f + \alpha b f_x\big] + \cdots \end{aligned} \tag{8.9}$$

Equating (8.8) and (8.9)), term for term, yields

$$a+b=1 \qquad \alpha b = \frac{1}{2} \qquad \beta b = \frac{1}{2}$$

Here, we have three equations in four unknowns; one unknown may thus be chosen. A convenient choice is, for example, $b = \frac{1}{2}$ which gives

$$a = \frac{1}{2} \qquad \alpha = \beta = 1.$$

For the $(m+1)$th interval we now have

$$\begin{aligned} k_1 &= hf(x_m, y_m) \\ k_2 &= hf(x_m + h, y_m + k_1) \\ y_{m+1} &= y_m + \frac{1}{2}\big(k_1 + k_2\big) \\ &= y_m + hF(x_m, y_m) \end{aligned}$$

where

$$F(x_m,y_m) = \frac{f(x_m,y_m) + f(x_m+h, y_m + hf(x_m,y_m))}{2}.$$

We note that the 2nd-order Runge-Kutta method derived here is, in fact, identical to the modified Euler method derived previously. A different choice for b would have yielded a different method. For example, the choice $b = \frac{1}{3}$ gives $a = \frac{2}{3}$, $\alpha = \beta = \frac{3}{2}$.

8.5.2 The fourth-order Runge-Kutta method (RK4)

Here, we have

$$\begin{aligned} k_1 &= hf(x_0,y_0) \\ k_2 &= hf(x_0+ph, y_0+\alpha k_1) \\ k_3 &= hf(x_0+qh, y_0+\beta k_1 + \gamma k_2) \\ k_4 &= hf(x_0+rh, y_0+\mu k_1 + \nu k_2 + \lambda k_3) \\ y_1 &= y_0 + ak_1 + bk_2 + ck_3 + dk_4 \end{aligned}$$

There are 13 unknowns in these equations. Expanding $y(x_1) = y(x_0+h)$ and y_1 in Taylor series to 4th-order, and equating term for term, as done for RK2, we find 11 equations. By choosing

$$p = q = \frac{1}{2}$$

we obtain

$$\begin{aligned}\beta &= \mu = \nu = 0\\ \alpha &= \gamma = \frac{1}{2}\\ \lambda &= r = 1\\ a &= d = \frac{1}{6}\\ b &= c = \frac{1}{3}\end{aligned}$$

On the $(m+1)$th interval we have, for the 4th-order Runge-Kutta method (RK4)

$$\begin{aligned}k_1 &= hf(x_m, y_m)\\ k_2 &= hf\left(x_m + \frac{h}{2}, y_m + \frac{1}{2}k_1\right)\\ k_3 &= hf\left(x_m + \frac{h}{2}, y_m + \frac{1}{2}k_2\right)\\ k_4 &= hf(x_m + h, y_m + k_3)\\ y_{m+1} &= y_m + \frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4\\ &= y_m + hF(x_m, y_m)\end{aligned}$$

where

$$hF(x_m, y_m) = \frac{1}{6}k_1 + \frac{2}{6}k_2 + \frac{2}{6}k_3 + \frac{1}{6}k_4.$$

8.6 Approximation error in one-step methods

Recall that an explicit one-step method for solving an IVP has the form

$$y_{m+1} = y_m + hF(x_m, y_m).$$

As shown, the Euler method, the modified Euler method, RK2 and RK4 are all explicit one-step methods.

Define the *local error* at x_m in an explicit one-step method as by

$$\varepsilon_m = y(x_{m-1}) + hF\big(x_{m-1}, y(x_{m-1})\big) - y(x_m).$$

Note the use of the exact value $y(x_m)$ in this definition. The *global error* at x_m in an explicit one-step method is given by

$$\Delta_m = y_m - y(x_m)$$

which is simply the difference between the approximate value y_m and the exact value $y(x_m)$.

We seek to investigate the propagation of errors in an explicit one-step method. With $y(x_0) = y_0$ (because the initial value is known exactly), we have

$$\begin{aligned}\Delta_1 &= y_1 - y(x_1)\\ &= y_0 + hF(x_0, y_0) - y(x_1)\\ &= \varepsilon_1\end{aligned}$$

and

$$\begin{aligned}\Delta_2 &= y_2 - y(x_2)\\ &= \big[y_1 + hF(x_1, y_1)\big] - y(x_2)\\ &= \big[y(x_1) + \Delta_1\big] + hF\big(x_1, y(x_1) + \Delta_1\big) - y(x_2)\\ &= \big(y(x_1) + \Delta_1\big) + h\Big[F\big(x_1, y(x_1)\big) + \Delta_1 F_y(x_1, \xi_1)\Big] - y(x_2)\\ &= \big[y(x_1) + hF(x_1, y(x_1)) - y(x_2)\big] + \Delta_1\big[1 + hF_y(x_1, \xi_1)\big]\\ &= \varepsilon_2 + \alpha_1\varepsilon_1\end{aligned}$$

where

$$\alpha_1 = 1 + hF_y(x_1, \xi_1).$$

Repeating this process, we obtain

$$\Delta_3 = \varepsilon_3 + \alpha_2\varepsilon_2 + \alpha_2\alpha_1\varepsilon_1$$

and

$$\Delta_4 = \varepsilon_4 + \alpha_3\varepsilon_3 + \alpha_3\alpha_2\varepsilon_2 + \alpha_3\alpha_2\alpha_1\varepsilon_1,$$

and in general,

$$\Delta_n = \sum_{j=1}^{n}\left(\prod_{k=j}^{n-1}\alpha_k\right)\varepsilon_j$$

where

$$\alpha_k = 1 + hF_y(x_k, \xi_k)$$

with $y(x_k) < \xi_k < y(x_k) + \Delta_k$.

If $|hF_y(x_k, \xi_k)|$ is small then $\alpha_k \approx 1$, and so

$$\Delta_n \approx \sum_{j=1}^{n}\varepsilon_j,$$

which is simply an accumulation of local errors. However, this is generally not expected to be the case, particularly if $F_y(x_k, \xi_k)$ has large magnitude. Note also, if the α's have magnitude larger than one, then the term in ε_1 could make the most significant contribution to the global error. If the local error has the form

$$\varepsilon_j = E_j h^{r+1}$$

where E_j is an appropriate coefficient, then the global error Δ_n is

$$\Delta_n = \sum_{j=1}^{n}\beta_j h^{r+1} = \left(\frac{1}{n}\sum_{j=1}^{n}\beta_j\right)(nh)h^r = \bar{\beta}(x_n - x_0)h^r$$

where

$$\beta_j = \left(\prod_{k=j}^{n-1}\alpha_k\right)E_j \qquad \text{and} \qquad \bar{\beta} = \frac{1}{n}\sum_{j=1}^{n}\beta_j$$

and we have used $nh = x_n - x_0$. Note that Δ_n is $O\left(h^r\right)$.

Order of local errors: For Euler's method we have

$$y(x_{m+1}) = y(x_m) + hf(x_m, y(x_m)) + O(h^2)$$

and so

$$\varepsilon_{m+1} = y(x_m) + hf(x_m, y(x_m)) - y(x_{m+1}) = O(h^2)$$

which gives $r = 1$. For the modified Euler method and RK2 we have

$$y(x_{m+1}) = y(x_m) + hF(x_m, y(x_m)) + O(h^3)$$

due to the fact that $y(x_m)+hF(x_m, y(x_m))$ is, by construction, equivalent to a second-order Taylor series (the $O(h^3)$ term represents the residual term). Hence, we have

$$\varepsilon_{m+1} = y(x_m) + hF(x_m, y(x_m)) - y(x_{m+1}) = O(h^3)$$

and so $r = 2$. Similarly, we have $r = 4$ for RK4.

Example 8.1. In the following table we compare the methods derived above for finding $y(1)$ from

$$\frac{\mathrm{d}y}{\mathrm{d}x} = x + y$$

with $y(0) = 0$. The stepsize h used in each calculation is indicated. The exact solution at $y(1)$ is 0.71828.

Method	$\boldsymbol{h}$	$\boldsymbol{y(1)}$	$\boldsymbol{\lvert\Delta\rvert}$ at $\boldsymbol{x = 1}$
Euler	$\frac{1}{8}$	0.56578	0.15
Modified Euler	$\frac{1}{4}$	0.69486	0.02
RK4	$\frac{1}{2}$	0.71735	0.0009
RK4	$\frac{1}{4}$	0.71821	0.00007

Clearly, the higher the order of the method, the more accurate the result. Note that even with a smaller stepsize, Euler's method is less accurate than the others.