Numerical Analysis

MATH-6375

Dambaru Bhatta
Professor of Mathematics

School of Mathematical & Statistical Sciences
The University of Texas-Rio Grande Valley

Contents

6. Numerical Integration
    - Newton-Cotes Formulas
    - Composite Numerical Integration
    - Gaussian Quadrature
    - Adaptive Quadrature

7. Initial Value Problems (IVP)
    - Euler's Method
    - Taylor Methods
    - Runge-Kutta Methods
    - Systems and Higher Order ODEs

8. Boundary Value Problems (BVP)
    - Classification
    - Shooting Method
    - Finite Difference Method

Chapter 1

Fundamentals

1.1 Introduction

Why Numerical Methods?

Numerical methods involves

Numerical methods provide approximations to the various problems which do not have analytical (exact) solutions.

While trying to solve real world problems, first we construct a mathematical model and then try to obtain a solution for this model. If we have an analytical solution, we do not worry about error from approximation. But it is rare to have analytical solutions for real applications. Since most of the problems arising from real applications do not have analytical or exact solutions, we need to consider approximating the solutions by some numerical techniques.

Algorithm: An algorithm consists of a sequence of steps to solve a specified mathematical problem.

Sources of Errors

- Error in modeling
  To represent a physical system, we simplify some assumptions to model which may lead to wrong solutions/conclusions.

- Error in input data
  Most of the time, computational models needs some input data. If

Figure 1.1: Solving Real Application

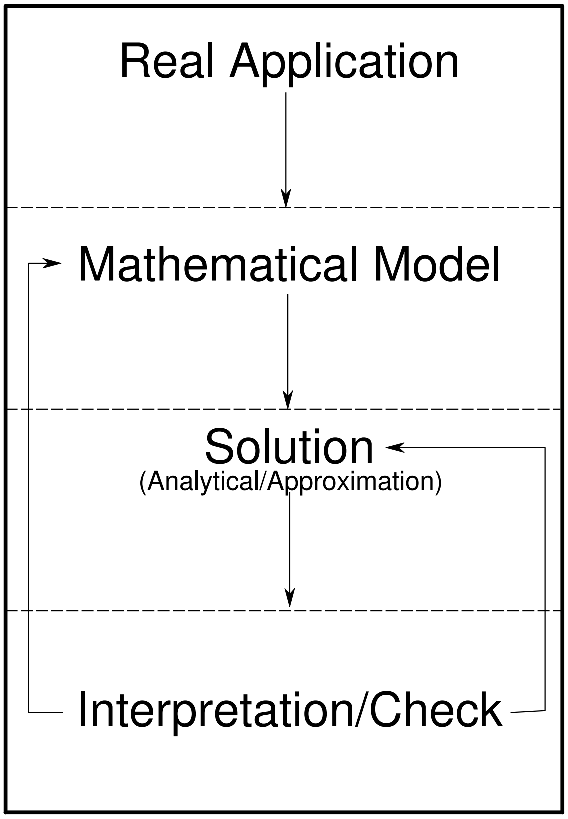
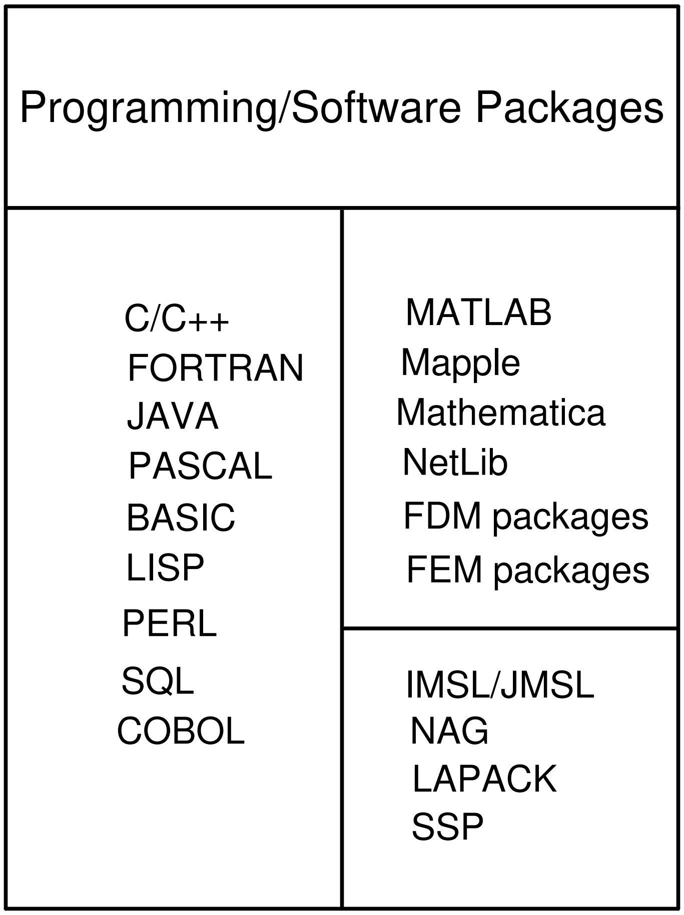


Figure 1.2: Computer Programs/Computing Utilities



input data contains error, output will be erroneous.

- Programming error/wrong selection of procedure or technique
  Programmers are human being. It is natural that they may make mistakes while programming. Peer-view programming technique is very useful to avoid this type of mistake. Also choice of solution method may lead to wrong direction.

- Machine error
  Error created by machine is also a source of error. Machine can not represent all real numbers. Each machine uses some precision to represent numbers. Because of this, while computing, often we encounter overflow/underflow/rounding etc.

- Truncation error
  When we are dealing with integration involving an infinite limit or dealing with infinite series, we need to truncate it so that we can use numerical technique. This truncation is a major source of error.

Numerical Analysis:

Numerical Analysis is used to develop and analyze numerical methods for solving problems appearing in many fields. Overall goal of numerical analysis is the design and analysis of techniques to give approximate but accurate solutions to difficult problems. Numerical analysis is the area that produces, analyzes, and implements algorithms for solving numerically the problems of continuous mathematics originated from real-world applications. These problems arise throughout the natural sciences, engineering, social sciences, medicine, and business. Growth in power and accessibility of digital computers has led to an increasing use of complex mathematical models in numerical analysis of escalating sophistication needed to solve more accurately.

1.2 Evaluating a polynomial

Goal of this course to discuss methods of solving mathematical problems with computers. Fundamental operations of arithmetic are addition and multiplication. Polynomials are basic building blocks for many computational techniques.

Example

Consider

$$P(x) \quad = \quad 2x^4 + 4x^3 + 5x^2 + 7x + 1$$

Evaluate this polynomial at $x = \frac{1}{3}$.

Method 1:

$$P\left(\frac{1}{3}\right) \quad = \quad 2 * \frac{1}{3} * \frac{1}{3} * \frac{1}{3} * \frac{1}{3} \; + \; 4 * \frac{1}{3} * \frac{1}{3} * \frac{1}{3} \; + \; 5 * \frac{1}{3} * \frac{1}{3} \; + \; 7 * \frac{1}{3} \; + \; 1$$

Here we use 10 multiplications and 4 additions.

Method 2:

In this method the powers of $x = \frac{1}{3}$ are stored and used to evaluate the polynomial.

$$\begin{aligned} \frac{1}{3} * \frac{1}{3} &= \left(\frac{1}{3}\right)^2 \\ \left(\frac{1}{3}\right)^2 * \frac{1}{3} &= \left(\frac{1}{3}\right)^3 \\ \left(\frac{1}{3}\right)^3 * \frac{1}{3} &= \left(\frac{1}{3}\right)^4 \end{aligned}$$

Thus we can write

$$P\left(\frac{1}{3}\right) \quad = \quad 2 * \left(\frac{1}{3}\right)^4 \; + \; 4 * \left(\frac{1}{3}\right)^3 \; + \; 5 * \left(\frac{1}{3}\right)^2 \; + \; 7 * \frac{1}{3} \; + \; 1$$

which uses 7 multiplications (3 for powers and other 4 with coefficients) and 4 additions.

Method 3:

Nested Multiplication: Efficient Evaluation of Polynomials:

Now

$$\begin{aligned} P(x) &= 1 + 7x + 5x^2 + 4x^3 + 2x^4 \\ &= 1 + x\left(7 + 5x + 4x^2 + 2x^3\right) \\ &= 1 + x(7 + x(5 + x(4 + 2x))) \\ &= 1 + x * (7 + x * (5 + x * (4 + 2 * x))) \end{aligned}$$

so that we have

$$P\left(\frac{1}{3}\right) \;=\; 1+\frac{1}{3}*(7+\frac{1}{3}*(5+\frac{1}{3}*(4+2*\frac{1}{3})))$$

Here we have 4 multiplications and 4 additions.
First evaluate $2*\frac{1}{3}$ and then addition with 4, get $4+\frac{2}{3}=\frac{14}{3}$.
Next evaluate $\frac{1}{3}*\frac{14}{3}=\frac{14}{9}$, then addition with 5, and obtain $5+\frac{14}{9}=\frac{59}{9}$.
Then evaluate $\frac{1}{3}*\frac{59}{9}=\frac{59}{27}$, then addition with 7, and obtain $7+\frac{59}{27}=\frac{248}{27}$.
Next evaluate $\frac{1}{3}*\frac{248}{27}=\frac{248}{81}$, then addition with 1, and obtain $1+\frac{248}{81}=\frac{329}{81}$.
This is known as Horner's method.
Also,

$$\begin{aligned} P(x) &= 2x^4+4x^3+5x^2+7x+1 \\ &= \left(2x^3+4x^2+5x+7\right)x+1 \\ &= \left(\left(2x^2+4x+5\right)x+7\right)x+1 \\ &= (((2x+4)\,x+5)\,x+7)\,x+1 \\ &= (((2*x+4)*x+5)*x+7)*x+1 \end{aligned}$$

Why important?
Solving many problems, such as ordinary differential equation (ODE) or partial differential equation (PDE) by finite difference method (FEM) or finite element method (FDM), one may need to evaluate a polynomial more than million times (considering space nodes and smaller time steps)

To evaluate a polynomial efficiently we group the terms in a nested multiplication. Let

$$P(x) \;=\; a_0+a_1x+a_2x^2+..........+a_{n-1}x^{n-1}+a_nx^n$$

be a polynomial of degree n. Nested multiplication gives

$$P(x) \;=\; a_0+x(a_1+x(a_2+..........+x(a_{n-1}+xa_n)....))$$

i.e.,

$$P(x) \;=\; \sum_{i=0}^{n} a_i x^i = \sum_{i=0}^{n}\left(a_i\prod_{j=1}^{i} x\right)$$

A general n degree polynomial can be evaluated in n multiplications and n additions.

Exercise

1. Rewrite the following polynomials in nested form. Evaluate with and without nested forms at $x = 1/3$.
   (a) $P(x) = 6x^4 + x^3 + 5x^2 + x + 1$

   (b) $P(x) = -3x^4 + 4x^3 + 5x^2 - 5x + 1$

   (c) $P(x) = 2x^4 + x^3 - x^2 + 1$

2. Evaluate $P(x) = x^6 - 4x^4 + 2x^2 + 1$ at $x = 1/2$ by considering $P(x)$ as a polynomial in x^2 and using nested multiplication.

3. Explain how to evaluate the polynomial for a given input x, using as few operations as possible. How many multiplications and additions are required (here a, b, c, d, e are nonzero constant coefficients)?
   (a) $P(x) = a + bx^5 + cx^{10} + dx^{15}$
   (b) $P(x) = ax^7 + bx^{12} + cx^{17} + dx^{22} + ex^{27}$

1.3 Computer Arithmetic:

In general when we think of a number, we mean a number in Decimal system. In decimal system we use 10 as the base. The number system used in computer arithmetic is Binary number system, not decimal number system. There are few other number systems available: Octal and Hexadecimal systems. These two are used rarely. If we use a base, say β, then the numbers represented in the βsystem look like the following:

$$(a_n a_{n-1} ......a_2 a_1 a_0 .b_1 b_2 b_3 ....)_\beta = \sum_{k=0}^{n} a_k \beta^k + \sum_{k=1}^{\infty} b_k \beta^{-k}$$

The separator between the integer and fractional part is called the radix point, since decimal point is reserved for base-10 system.

Example:

Consider the number 341.57 with $\beta = 10$. Then

$$341.57 = 3 \times 10^2 + 4 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1} + 7 \times 10^{-2}$$

Here $a_0 = 1$, $a_1 = 4$, $a_2 = 3$, $b_1 = 5$ and $b_2 = 7$.

Binary number system uses only two numbers, 0 (zero) and 1, to represent all numbers. Decimal system uses 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

0, 1, 2, 3, 4, 5, 6, 7 are used in Octal system whereas 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F are used in Hexadecimal system.

1.3.1 Binary Numbers

In binary system, only 0 and 1 are used. Each is called a binary digit (in short, bit). bit means 0 or 1. Binary numbers are expressed as

$$.......b_2 b_1 b_0 .b_{-1} b_{-2}......$$

This binary number is equivalent to

$$....b_2 2^2 + b_1 2^1 + b_0 2^0 + b_{-1} 2^{-1} + b_{-2} 2^{-2}....$$

in decimal system. To distinguish a binary number, we will use subscript 2, for example $(101)_2$ which is 5 in decimal and $(111)_2 = 7$.

Table 1.1: Binary and Decimal Numbers

Binary	Decimal
$(101)_2$	5
$(11111)_2$	15
$(0.1)_2$	1/2
$(0.11)_2$	3/4

1.3.1.1 Conversion from Binary to Decimal

To convert a binary number to a decimal, we consider by separating into integer and fractional parts. For integer part, we add up positive powers of 2 and for fractional part, we add up negative powers of 2.

Example 1:

Consider $(10110)_2$.

Decimal equivalent is $1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0 = 16+4+2 = 22$, i.e.,

$(10110)_2 = 22$.

Example 2:

Consider $(0.11)_2$.

Decimal equivalent is $1 \times 2^{-1} + 1 \times 2^{-2} = \frac{1}{2} + \frac{1}{4} = \frac{3}{4}$

Example 3:

Consider $(1011.1010)_2$.

For integer part, decimal equivalent is $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 8 + 2 + 1 = 11$.

For fractional part, decimal equivalent is $1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} = \frac{1}{2} + \frac{1}{8} = \frac{5}{8} = 0.625$.

Thus we have $(1011.1010)_2 = 11\frac{5}{8} = 11.625$.

Example 4:

Consider $(0.\overline{1011})_2$.

Let $x = (0.\overline{1011})_2$. Multiply x by 2^4 which shifts 4 places to the left in binary (as $0.\overline{1011} = 0.1011\overline{1011}$), we have

$$\begin{aligned} 2^4 x &= 1011.\overline{1011} \\ x &= 0000.\overline{1011} \end{aligned}$$

Subtracting,

$$\left(2^4 - 1\right) x = (1011)_2 = 11$$

Hence, we obtain $15x = 11$, or, $x = \frac{11}{15}$, i.e., $(0.\overline{1011})_2 = \frac{11}{15}$.

Example 5:

Consider $(0.10\overline{101})_2$.

Let $x = (0.10\overline{101})_2$. Multiply by 2^2 shifts 2 places to the left, we have $y = 2^2x = 10.\overline{101}$ and let $z = .\overline{101} = .101\overline{101}$, then we can write

$$\begin{aligned} 2^3 z &= 101.\overline{101} \\ z &= 000.\overline{101} \end{aligned}$$

which gives us $(2^3 - 1)\, z = (101)_2$, i.e., $7z = 5$,i.e., $z = \frac{5}{7}$. Now $y = 2 + \frac{5}{7} = 19/7$,and finally we obtain $x = \frac{y}{2^2} = \frac{19}{28}$.

1.3.1.2 Conversion from Decimal to Binary

To convert a decimal number to its binary equivalent, we consider integer and fractional parts separately. For integer part, we divide the decimal number by 2 successively and recording the remainders. For fractional part, we reverse the process, i.e., multiply by 2 successively and record the integer parts.

Example 6:

Consider the decimal number 26.

$$\begin{aligned}26 \div 2 &= 13R0\\ 13 \div 2 &= 6R1\\ 6 \div 2 &= 3R0\\ 3 \div 2 &= 1R1\\ 1 \div 2 &= 0R1\end{aligned}$$

Hence $26 = (11010)_2$. Check: $(11010)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 16 + 8 + 2 = 26$.

Example 7:

Consider the decimal number 0.75

Here we obtain

$$\begin{aligned}0.75 \times 2 &= 1.5 = 0.5 + 1\\ 0.5 \times 2 &= 1.0 = 0 + 1\end{aligned}$$

Thus $0.75 = (.11)_2$.

Example 8:

Consider the decimal number $\frac{2}{5} = 0.4$.

Here we obtain

$$\begin{aligned}0.4 \times 2 &= 0.8 = 0.8 + 0\\ 0.8 \times 2 &= 1.6 = 0.6 + 1\\ 0.6 \times 2 &= 1.2 = 0.2 + 1\\ 0.2 \times 2 &= 0.4 = 0.4 + 0\\ 0.4 \times 2 &= 0.8 = 0.8 + 0\\ 0.8 \times 2 &= 1.6 = 0.6 + 1\\ &\vdots\end{aligned}$$

Hence $\frac{2}{5} = 0.4 = (.\overline{0110})_2$

Example 9:

Consider 53.7

For integer part,

$$\begin{aligned}
53 \div 2 &= 26R1 \\
26 \div 2 &= 13R0 \\
13 \div 2 &= 6R1 \\
6 \div 2 &= 3R0 \\
3 \div 2 &= 1R1 \\
1 \div 2 &= 0R1
\end{aligned}$$

so that we have $53 = (110101)_2$

For fractional part,

$$\begin{aligned}
0.7 \times 2 &= 1.4 = 0.4 + 1 \\
0.4 \times 2 &= 0.8 = 0.8 + 0 \\
0.8 \times 2 &= 1.6 = 0.6 + 1 \\
0.6 \times 2 &= 1.2 = 0.2 + 1 \\
0.2 \times 2 &= 0.4 = 0.4 + 0 \\
0.4 \times 2 &= 0.8 = 0.8 + 0 \\
&\vdots
\end{aligned}$$

yielding $0.7 = (.1\overline{0110})_2$

Thus we have $53.7 = (110101.1\overline{0110})_2$.

1.3.1.3 Addition and Multiplication of Binary Numbers

Additions and multiplications are respectively defined as

$$\begin{array}{cccccc}
0 & + & 0 & = & 0 \\
1 & + & 0 & = & 1 \\
0 & + & 1 & = & 1 \\
1 & + & 1 & = & 10
\end{array} \quad \text{and} \quad \begin{array}{ccccc}
0 & \times & 0 & = & 0 \\
1 & \times & 0 & = & 0 \\
0 & \times & 1 & = & 0 \\
1 & \times & 1 & = & 1
\end{array}$$

Example 10

Simplify

$$\begin{array}{cr} & 110\\ + & 101\\ + & 11\\ & - - --\\ = & 1110 \end{array} \qquad \text{Check} \qquad \begin{array}{cr} & 6\\ + & 5\\ + & 3\\ & - - -\\ = & 14 \end{array}$$

Example 11

Simplify

$$\begin{array}{cr} & 110\\ \times & 11\\ & - - --\\ = & 10010 \end{array} \qquad \text{Check} \qquad \begin{array}{cr} & 6\\ \times & 3\\ & - - -\\ = & 18 \end{array}$$

Exercise

1. Convert the following numbers to decimal numbers
   (a) $(1101)_8$, (b) $(507)_8$, (c) $(2A3C)_{16}$ (d) $(11.101)_2$ (e) $(32.41)_8$

2. Convert the following numbers to binary number
   (a) 792, (b) 95.8, (c) 1011.11 (d) $(63)_8$

3. Complete the following binary operations and check your answers by converting to decimals.

$$\text{(a)} \; \begin{array}{cr} & 110\\ + & 111\\ & - - -- \end{array} \qquad \text{(b)} \; \begin{array}{cr} & 1011\\ + & 101\\ & - - -- \end{array} \qquad \text{(c)} \; \begin{array}{cr} & 101\\ \times & 11\\ & - - -- \end{array} \qquad \text{(d)} \; \begin{array}{cr} & 111\\ + & 11\\ + & 100\\ & - - -- \end{array}$$

Table 1.2: Distribution of bits among three precisions

precision	sign	exponent (M)	mantissa (N)	total
single	1	8	23	32
double	1	11	52	64
long double	1	15	64	80

1.3.2 Floating Point Representation of Real Numbers

Here we present a model for computer arithmetic of floating numbers. This IEEE (Institute of Electrical and Electronics Engineers) Floating Point Standard consists of a set of binary representation of real numbers. A floating point number consists of three parts, the sign, a mantissa and an exponent. There are three commonly used level of precision for floating point numbers: single precision, double precision and extended precision. Number of bits allocated are as follows:

The form of a normalized IEEE floating point number is

$$\pm 1.bbbbb......b \times 2^p$$

where each of the N b's is 0 or 1, and p is an M-bit binary number representing the exponent. Normalization means that the leading (leftmost) bit must be 1. For single precision $M = 8$ and $N = 23$. For double precision $M = 11$ and $N = 52$.

Example 1

Consider the decimal number 9.

Since $9 = (1001)_2$, Normalized binary representation is given by 1.001×2^3. Shift of 3 bits is equivalent to multiplication by 2^3.

The double precision number 1 is

$+1.\boxed{0000000000000000000000000000000000000000000000000000}\times 2^0$

where mantissa is presented in 52 boxes. The next floating number greater than 1 is

$+1.\boxed{0000000000000000000000000000000000000000000000000001}\times 2^0$

or $1 + 2^{-52}$.

The distance between 1 and the smallest floating point number greater than 1 is called machine epsilon, denoted by ϵ_{mach}. For IEEE double precision floating point standard

$$\epsilon_{mach} \;=\; 2^{-52} = 2.2 \times 10^{-16}$$

and for single precision machine epsilon is $\epsilon_{mach} = 2^{-23}$

1.3.3 Truncating to a finite number of bits

Consider the decimal number $9.4 = (1001.\overline{0110})_2$ which a representation

$+1.\boxed{0010110011001100110011001100110011001100110011001100}1100110\ldots\times 2^3.$

How do we fit the infinite bits in finite bits?

There are two kinds of procedure used to truncate an infinite number of bits to a finite number of bits. Doing so we make some error. One is known as Chopping and other is called Rounding.

1.3.3.1 Chopping and Rounding

In chopping, the bits fall off the end are simply thrown away; in single precision, beyond 23rd bit to the right of the radix point; in double precision, beyond 52nd bit to the right of the radix point.

Rounding corresponds adding 1 to the last bit of mantissa if the next bit is 1 and do nothing if the next bit is 0.

Consider the decimal number 9.4. We have shown earlier that it is equivalent to

$+1.\boxed{0010110011001100110011001100110011001100110011001100}11001100\ldots\times 2^3.$

In Chopping, we obtain 9.4 as

$+1.\boxed{0010110011001100110011001100110011001100110011001100}\times 2^3.$

Rounding yields 9.4 as

$+1.\boxed{0010110011001100110011001100110011001100110011001101}\times 2^3.$

1.3.4 Notation fl(x)

Rounding IEEE double precision floating point number associated with x is denoted by $\text{fl}(x)$. We obtain 9.4 as

$+1.\boxed{\text{52 bits here}}110011001100\ldots\times 2^3.$

To obtain fl(9.4) we have ignored the bits appearing right to the box above and add 1 to the 52nd bit. Infinite tail is given by

$.\overline{1100} \times 2^{-52} \times 2^3 = .\overline{0110} \times 2^{-51} \times 2^3 = .4 \times 2^{-48}$.

Addition of 1 to the 52nd is done by adding

$2^{-52} \times 2^3 = 2^{-49}$ in the rounding step. Thus we have

$$\begin{aligned} \text{fl}(9.4) &= 9.4 + 2^{-49} - 0.4 \times 2^{-48} \\ &= 9.4 + (1 - 0.8) \times 2^{-49} \\ &= 9.4 + 0.2 \times 2^{-49} \end{aligned}$$

Hence storing 9.4 we see that rounding error is 0.2×2^{-49}. Thus $\text{fl}(9.4) \neq 9.4$, but they are close.

1.3.5 Underflow and Overflow

While doing computations if we encounter a number which lies between the zero and the smallest number that can be represented using particular precision, then it is known as the underflow. On the other hand, if the computed number falls beyond the largest number that can be represented for that precision, then we say that an overflow has occurred.

Most of the compilers do not complain about underflow and the number produced by underflow is assumed to be zero. If an overflow has occurred, compiler sends a message and program is aborted. **NaN** stands for Not a Number. NaN occurs when something like $\frac{0}{0}$, ∞ are encountered. When NaN is encountered, the programmer needs to take care of it by checking his/her program.

As a conclusion, when a sensitive computation is done using computer programming, it is always better to use the higher precision available.

1.3.6 Absolute and Relative Errors

Let x^* be an approximation of the exact quantity x. Absolute and relative errors are defined as

$$\text{Absolute error} \quad = \quad |x^* - x|$$

$$\text{Relative error} \quad = \quad \frac{|x^* - x|}{|x|}$$

Example 1

If $x = 0.3000 \times 10^1$ and $x^* = 0.3100 \times 10^1$, the absolute error is 0.1 and the relative error is $0.00\overline{3} \times 10^1$.

If $x = 0.3000 \times 10^{-3}$ and $x^* = 0.3100 \times 10^{-3}$, the absolute error is 0.1×10^{-4} and the relative error is $0.00\overline{3} \times 10^1$.

If $x = 0.3000 \times 10^4$ and $x^* = 0.3100 \times 10^4$, the absolute error is 0.1×10^3 and the relative error is $0.00\overline{3} \times 10^1$.

It shows that absolute errors vary widely. As a measure of accuracy, the relative error may be more meaningful since it takes into consideration the size of the value.

Relative Rounding Error

If we write $x^* = \text{fl}(x)$, we obtain the relative rounding error as

$$\text{Relative rounding error} \quad = \quad \frac{|\text{fl}(x) - x|}{|x|} \leq \frac{1}{2}\epsilon_{mach}$$

Example 2

Consider $x = 9.4$, then the relative rounding error is

$$\frac{|\text{fl}(9.4) - 9.4|}{|9.4|} = \frac{0.2 \times 2^{-49}}{9.4} = \frac{2 \times 2^3}{94} \times 2^{-52} = \frac{8}{47} \times 2^{-52} \quad < \quad \frac{1}{2}\epsilon_{mach}$$

1.3.7 Loss of Significance

Consider the quadratic equation

$$x^2 + 8^{13}x - 4 \quad = \quad 0 \qquad (1.1)$$

We know that roots of the quadratic equation

$$ax^2 + bx + c \quad = \quad 0 \tag{1.2}$$

are given by

$$x \quad = \quad \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1.3}$$

So for (1.1), we have

$$x \quad = \quad \frac{-8^{13} \pm \sqrt{8^{26} - 4(-4)}}{2}$$

Considering the negative sign in front of the radical, we obtain one root as

$$x_1 \quad = \quad \frac{-8^{13} - \sqrt{8^{26} + 16}}{2}$$

The positive sign in front of the radical yields the other root as

$$x_2 \quad = \quad \frac{-8^{13} + \sqrt{8^{26} + 16}}{2} = 0.0$$

Let us analyze the case of x_2 :

$$\begin{aligned} x_2 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\ &= \frac{\left(-b + \sqrt{b^2 - 4ac}\right)\left(b + \sqrt{b^2 - 4ac}\right)}{2a\left(b + \sqrt{b^2 - 4ac}\right)} \end{aligned}$$

i.e.,

$$x_2 \quad = \quad \frac{-2c}{b + \sqrt{b^2 - 4ac}} \tag{1.4}$$

Using the formula (1.4), we obtain x_2 as

$$x_2 \quad = \quad 1.1 \times 10^{-11}$$

Evaluation of the solutions of the formula (1.3) needs careful consideration when b is very large compared with a and/or c. If $4|ac| \ll b^2$, then b and $\sqrt{b^2 - 4ac}$ are nearly equal. In that case, loss of significance is encountered while computing one of the roots.

In the above scenario if b is positive, the roots are given by

$$x_1 = \frac{-b - \sqrt{b^2 - 4ac}}{2a} \qquad \text{and} \qquad x_2 = \frac{-2c}{b + \sqrt{b^2 - 4ac}}$$

If b is negative (and $4|ac| \ll b^2$) , the roots are computed by

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \qquad \text{and} \qquad x_2 = \frac{2c}{-b + \sqrt{b^2 - 4ac}}$$

Exercise

1. Find the absolute and relative errors for the following
   (a) $x = 5.410, \quad x^* = 5.411$
   (b) $x = 5410, \quad x^* = 5411$
   (c) $x = 54.10, \quad x^* = 54.11$

2. Find the relative rounding error for the following numbers
   (a) $x = 5.4$ (b) $x = 9.6$

3. Write the fl(x)in binary format using rounding in double precision for the following decimal numbers:
   (a) $x = \frac{1}{4}$ (b) $x = 9.5$ (c) $x = \frac{1}{3}$

1.4 Matrix Algebra

Let us consider a system of m-linear equations in n unknowns $x_1, x_2, \ldots.., x_n$ for given a_{ij} and b_i, $i = 1, 2, 3, \ldots.., m$, $j = 1, 2, 3, \ldots.., n$, of the form

$$\begin{array}{rcl}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots. \quad \ldots. + a_{1n}x_n & = & b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots. \quad \ldots. + a_{2n}x_n & = & b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots. \quad \ldots. + a_{3n}x_n & = & b_3 \\
\vdots & \vdots & \vdots \\
a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \ldots. \quad \ldots. + a_{mn}x_n & = & b_m
\end{array} \tag{1.5}$$

The above system of m equations in n unknowns (1.5) can be expressed in matrix notation as

$$A\mathbf{x} = \mathbf{b} \tag{1.6}$$

where the matrices A, $\mathbf{x}$ and $\mathbf{b}$ are respectively given by

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{bmatrix}$$

In short, we write $A = [a_{ij}]_{m \times n}$ where a_{ij} is the (i, j)th entry. In a matrix, horizontal entries represent **rows** and vertical entries represent **columns**. A matrix with only one row is called a row vector. A matrix with only one column is known as a column vector or simply a vector. A matrix is called a **square matrix** of order n if the number of rows = the number of columns $= n$. The **transpose** of an $m \times n$ matrix A is obtained by exchanging the rows and columns and it is denoted by A^T. A matrix is called **symmetric** if

$$A^T = A$$

A square matrix is called an **Identity** matrix if all the diagonal entries are 1 and non-diagonal entries are 0, i.e., $a_{ii} = 1$ and $a_{ij} = 0$, $i \neq j$. It is denoted by I.

Diagonal Matrix

A matrix is called diagonal if all non-diagonal elements are zero, i.e., $a_{ij} = 0$ for $i \neq j$. Identity matrix is a diagonal matrix.

Diagonally Dominant Matrix

A matrix is said to be diagonally dominant if for every row of the matrix, the magnitude of the diagonal entry in a row is larger than or equal to the sum of the magnitudes of all the other (non-diagonal) entries in that row, i.e., the matrix A is diagonally dominant if it satisfies the following property:

$$|a_{ii}| \quad \geq \textstyle\sum_{j \neq i} \quad |a_{ij}| \qquad for\ all\ i$$

The matrix

$$A = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 7 & 5 \\ -2 & 3 & -6 \end{bmatrix}$$

is diagonally dominant.

Lower and Upper Triangular Matrices

The matrix $A = [a_{ij}]_{m \times n}$ is called a lower triangular matrix if $a_{ij} = 0$ for $i < j$. A is called an upper triangular matrix if $a_{ij} = 0$ for $i > j$.

Sparse Matrix

A matrix is called sparse if many of the entries are known to be zero.

Example 1

Consider the matrices

$$A = \begin{bmatrix} 2 & -4 & 9 \\ 3 & 7 & 5 \\ -1 & 3 & 6 \\ 5 & 2 & 1 \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} 3 \\ 1 \\ -2 \end{bmatrix}$$

Here A and A^T are 4×3 and 3×4 matrices respectively. $\mathbf{x}^T = [3\ 1\ -2]$ is 1×3, i.e., a row vector. Also

$$A^T = \begin{bmatrix} 2 & 3 & -1 & 5 \\ -4 & 7 & 3 & 2 \\ 9 & 5 & 6 & 1 \end{bmatrix}$$

Addition and Multiplication:

Let $A = [a_{ij}]_{m \times n}$ and $B = [b_{ij}]_{m \times n}$ be two $m \times n$ matrices. Then $A + B = C = [c_{ij}]_{m \times n}$ where $c_{ij} = a_{ij} + b_{ij}$. For a scalar λ, λA is defined as $[\lambda a_{ij}]_{m \times n}$.

Let $A = [a_{ij}]_{m \times n}$ be an $m \times n$ matrix and $B = [b_{jk}]_{n \times p}$ be an $n \times p$ matrix. Then the product $AB = C = [c_{ik}]_{m \times p}$ is an $m \times p$ matrix where c_{ik} is defined by

$$c_{ik} = \sum_{j=1}^{n} a_{ij} b_{jk} \qquad i = 1, ..., m;\ k = 1, ..., p.$$

In general, matrix multiplication is not commutative, i.e.,

$$AB \neq BA$$

For any matrices A, B and C such that all the indicated sums and products exist, then we have

$$A(BC) = (AB)C, \qquad A(B + C) = AB + AC, \qquad (A + B)C = AC + BC$$

Example 2

Consider the matrices

$$A = \begin{bmatrix} 2 & 0 & 9 \\ 3 & 7 & 5 \\ 5 & 2 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 2 & 1 \\ -1 & 0 & 4 \\ 3 & 1 & 2 \end{bmatrix}$$

Here

$$A + B = \begin{bmatrix} 2+5 & 0+2 & 9+1 \\ 3-1 & 7+0 & 5+4 \\ 5+3 & 2+1 & -1+2 \end{bmatrix} = \begin{bmatrix} 7 & 2 & 10 \\ 2 & 7 & 9 \\ 8 & 3 & 1 \end{bmatrix} \qquad 5A = \begin{bmatrix} 10 & 0 & 45 \\ 15 & 35 & 25 \\ 25 & 10 & -5 \end{bmatrix}$$

Example 3

Consider the matrices

$$A = \begin{bmatrix} 2 & 0 & 9 \\ 3 & 7 & 5 \\ 5 & 2 & -1 \end{bmatrix} \qquad B = \begin{bmatrix} 5 & 2 \\ -1 & 0 \\ 3 & 1 \end{bmatrix}$$

Here

$$\begin{aligned} AB &= \begin{bmatrix} 2 & 0 & 9 \\ 3 & 7 & 5 \\ 5 & 2 & -1 \end{bmatrix} \begin{bmatrix} 5 & 2 \\ -1 & 0 \\ 3 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 2 \times 5 + 0 \times (-1) + 9 \times 3 & 2 \times 2 + 0 \times 0 + 9 \times 1 \\ 3 \times 5 + 7 \times (-1) + 5 \times 3 & 3 \times 2 + 7 \times 0 + 5 \times 1 \\ 5 \times 5 + 2 \times (-1) + (-1) \times 3 & 5 \times 2 + 2 \times 0 + (-1) \times 1 \end{bmatrix} \\ &= \begin{bmatrix} 37 & 13 \\ 23 & 11 \\ 20 & 9 \end{bmatrix} \end{aligned}$$

Inverse of a matrix

Let A be a square matrix of order n. A is called **invertible** or **non-singular** if there exists a square matrix B such that

$$AB = BA = I$$

and the inverse of A is denoted by A^{-1}. If a matrix does not have an inverse, it is called a **singular** matrix.

Example 4

Consider the matrices

$$A = \begin{bmatrix} 4 & 6 \\ 1 & 2 \end{bmatrix} \qquad B = \begin{bmatrix} 1 & -3 \\ -\frac{1}{2} & 2 \end{bmatrix}$$

Here

$$AB = \begin{bmatrix} 4 & 6 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & -3 \\ -\frac{1}{2} & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

and

$$BA = \begin{bmatrix} 1 & -3 \\ -\frac{1}{2} & 2 \end{bmatrix} \begin{bmatrix} 4 & 6 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Thus we have $B = A^{-1}$. Also $B^{-1} = A$.

Determinant of a Matrix:

Here it goes....

Result:

Let A and B be two square matrices of order n. Then

$$(AB)^{-1} = B^{-1}A^{-1}$$

The matrix A is non-singular if and only if $|A| \neq 0$.

Elementary Matrix:

Elementary (row ??) operations are of three types

- Interchanging of two rows $R_i \leftrightarrow R_j$
- Multiplying by a nonzero number $\lambda R_i \leftrightarrow R_i$
- Adding to a row a multiple of another row $R_i + \lambda R_j \leftrightarrow R_i$

An elementary matrix is defined as an $n \times n$ matrix that is obtained by an elementary operation on an $n \times n$ identity matrix.

If a matrix is invertible, after applying a series of elementary row operations on it, we can reduce it to I i.e.,

$$E_m E_{m-1} \ldots\ldots E_2 E_1 A = I$$

Then we have

$$A^{-1} = E_m E_{m-1} \ldots\ldots E_2 E_1$$

Example 5

Consider

$$A \;=\; \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 4 & 7 \end{bmatrix}$$

Obtain the inverse of A by using elementary operations.

Consider $I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

Applying $R_2 + (-1)R_1 \to R_2$, we have

$$E_1 \;=\; \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

so that we have

$$E_1A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 2 & 4 & 7 \end{bmatrix} \qquad E_1I = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{by } R_2 + (-1)R_1 \to R_2$$

$$E_2E_1A = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_2E_1I = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \qquad \text{by } R_3 + (-2)R_1 \to R_3$$

$$E_3E_2E_1A = \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_3E_2E_1I = \begin{bmatrix} 3 & -2 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \qquad \text{by } R_1 + (-2)R_2 \to R_1$$

$$E_4E_3E_2E_1A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_4E_3E_2E_1I = \begin{bmatrix} 9 & -2 & -3 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} = A^{-1}$$

by $R_1 + (-3)R_3 \to R_1$

with

$$E_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \qquad E_3 = \begin{bmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad E_4 = \begin{bmatrix} 1 & 0 & -3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Thus we have

$$A^{-1} = \begin{bmatrix} 9 & -2 & -3 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

Check that $AA^{-1} = I = A^{-1}A$.

Another way to find the inverse of a square matrix A is to use the adjoint of A.

$$A^{-1} = \frac{1}{|A|}\,\text{Adj}(A)$$

where

$$\text{Adj}(A) = [c_{ji}]_{n\times n} \quad \text{with} \quad c_{ij} = (-1)^{i+j} \begin{vmatrix} a_{11} & a_{12} & \vdots & a_{1\,j-1} & a_{1,\,j+1} & \vdots & a_{1n} \\ a_{21} & a_{22} & \vdots & a_{2\,j-1} & a_{2,\,j+1} & \vdots & a_{2n} \\ \cdots & \cdots & & & & & \cdots \\ & & & & & & \\ a_{i-1\,1} & a_{i-1\,2} & \vdots & a_{i-1\,j-1} & a_{i-1,\,j+1} & \vdots & a_{i-1,n} \\ a_{i+1\,1} & a_{i+1\,2} & \vdots & a_{i+1\,j-1} & a_{i+1,\,j+1} & \vdots & a_{i+1,n} \\ \cdots & \cdots & & & & & \cdots \\ a_{n1} & a_{n2} & \vdots & a_{n\,j-1} & a_{n,\,j+1} & \vdots & a_{nn} \end{vmatrix}_{(n-1)\times(n-1)}$$

Example 6

Consider the matrix

$$A = \begin{bmatrix} 3 & 2 & -4 \\ -4 & 0 & 2 \\ -1 & 1 & 5 \end{bmatrix}$$

Here we obtain

$$c_{11} = +\begin{vmatrix} 0 & 2 \\ 1 & 5 \end{vmatrix} = -2 \qquad c_{12} = -\begin{vmatrix} -4 & 2 \\ -1 & 5 \end{vmatrix} = 18 \qquad c_{13} = +\begin{vmatrix} -4 & 0 \\ -1 & 1 \end{vmatrix} = -4$$

$$c_{21} = -\begin{vmatrix} 2 & -4 \\ 1 & 5 \end{vmatrix} = -14 \quad c_{22} = +\begin{vmatrix} 3 & -4 \\ -1 & 5 \end{vmatrix} = 11 \quad c_{23} = -\begin{vmatrix} 3 & 2 \\ -1 & 1 \end{vmatrix} = -5$$

$$c_{31} = +\begin{vmatrix} 2 & -4 \\ 0 & 2 \end{vmatrix} = 4 \qquad c_{32} = -\begin{vmatrix} 3 & -4 \\ -4 & 2 \end{vmatrix} = 10 \quad c_{33} = +\begin{vmatrix} 3 & 2 \\ -4 & 0 \end{vmatrix} = 8$$

so that

$$\mathrm{Adj}(A) \quad = \quad \begin{bmatrix} -2 & -14 & 4 \\ 18 & 11 & 10 \\ -4 & -5 & 8 \end{bmatrix}$$

and

$$\begin{aligned} |A| &= 3\begin{vmatrix} 0 & 2 \\ 1 & 5 \end{vmatrix} - 2\begin{vmatrix} -4 & 2 \\ -1 & 5 \end{vmatrix} + (-4)\begin{vmatrix} -4 & 0 \\ -1 & 1 \end{vmatrix} \\ &= 3(-2) - 2(-18) - 4(-4) \\ &= -6 + 36 + 16 = 46 \end{aligned}$$

yielding

$$A^{-1} \quad = \quad \begin{bmatrix} -\frac{1}{23} & -\frac{7}{23} & \frac{2}{23} \\ \frac{9}{23} & \frac{11}{46} & \frac{5}{23} \\ -\frac{2}{23} & -\frac{5}{46} & \frac{4}{23} \end{bmatrix}$$

Properties of non-singular matrix

For an $n \times n$ matrix A, the following properties are equivalent:

- Inverse of A exists, i.e., A is non-singular
- Determinant of A is nonzero
- Rows of A form a basis for $\mathbb{R}^n$
- Columns of A form a basis for $\mathbb{R}^n$
- As a map from $\mathbb{R}^n$ to $\mathbb{R}^n$, A is injective (one to one)
- As a map from $\mathbb{R}^n$ to $\mathbb{R}^n$, A is surjective (onto)
- Equation $A\mathbf{x} = \mathbf{0}$ implies $\mathbf{x} = \mathbf{0}$
- For each $\mathbf{b} \in \mathbb{R}^n$, there exists exactly one $\mathbf{x} \in \mathbb{R}^n$ such that $A\mathbf{x} = \mathbf{b}$.
- A can be expressed as a product of elementary matrices
- 0 is not an eigenvalue of A

Positive definite matrix

A matrix A is positive definite if $\mathbf{x}^T A\mathbf{x} > 0$ for every nonzero vector $\mathbf{x}$. For example for

$$A = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

we have

$$\begin{aligned} \mathbf{x}^T A\mathbf{x} &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} x_1 - x_2 \\ x_1 + x_2 \end{bmatrix} \\ &= x_1(x_1 - x_2) + x_2(x_1 + x_2) = x_1^2 + x_2^2 > 0 \end{aligned}$$

for all $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \neq 0$. Also $\mathbf{x}^T A\mathbf{x}$ is called a quadratic form.

Eigenvalues of a Matrix

Let A be an $n \times n$ matrix. Then λ is called an eigenvalue of A if there exists some nonzero vector $\mathbf{x}$ such that

$$A\mathbf{x} = \lambda\mathbf{x} \tag{1.7}$$

Result

Eigenvalues of a square matrix A can be obtained by solving the characteristic equation

$$|\lambda I - A| = 0 \tag{1.8}$$

If A is positive definite and symmetric, then its eigenvalues are real and positive.

Example 7

Find the eigenvalues of the matrix A where A is given by

$$A = \begin{bmatrix} 3 & 2 \\ 1 & 2 \end{bmatrix}$$

Here the characteristic equation is given by

$$\begin{vmatrix} \lambda - 3 & -2 \\ -1 & \lambda - 2 \end{vmatrix} = 0$$

This yields us

$$\begin{aligned} (\lambda - 3)(\lambda - 2) - 2 &= 0 \\ \lambda^2 - 5\lambda + 4 &= 0 \\ (\lambda - 4)(\lambda - 1) &= 0 \end{aligned}$$

Thus the eigenvalues are given by $\{4, 1\}$.

Exercise

1. Let $A = [a_{ij}]_{m\times n} \quad B = [b_{jk}]_{n\times p} C = [c_{ik}]_{p\times r}$. Prove that $A(BC) = (AB)C$

2. Consider the following matrix:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 4 & 7 \end{bmatrix}$$

Find its inverse by using the adjoint definition. What is the value of the determinant of the original matrix.

3. Consider the following matrix:

$$A = \begin{bmatrix} -1 & 2 & -3 \\ 2 & 1 & 0 \\ 4 & -2 & 5 \end{bmatrix}$$

Find its inverse by using the elementary matrix operations. What are the elementary matrices you have used to obtain the inverse.

4. Find the eigenvalues of A where

$$\text{(a)} \ A = \begin{bmatrix} 3 & 1 & 1 \\ 2 & 4 & 2 \\ 1 & 1 & 3 \end{bmatrix} \qquad \text{and} \qquad \text{(b)} \ A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

1.5 Vector and Matrix Norms

Vector and matrix norms are very useful in error analysis.

Vector Norms

Let us consider a vector space V.

A vector norm is a function $\| \ \|$ from V to the set of non-negative real numbers such that

- $\|\mathbf{x}\| \geq 0$ for any $\mathbf{x} \in V$
- $\|\mathbf{x}\| = 0$ if and only if $\mathbf{x} = \mathbf{0}$
- $\|\alpha \mathbf{x}\| = |\alpha| \, \|\mathbf{x}\|$ for scale α and $\mathbf{x} \in V$
- $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ for any $\mathbf{x}$, $\mathbf{y} \in V$

Last property is called triangle inequality.

Let $\mathbf{x}$ be vector in $\mathbb{R}^n$, i.e.,

$$\mathbf{x} \quad = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \quad (x_1, \, x_2, \ldots\ldots\ldots, x_n)^T$$

Then $\mathbb{R}^+ \cup \{0\}$ denotes the set of non-negative real numbers. The most common norm used is the Euclidean L_2-norm.

L_2**-norm** is defined by

$$\|\mathbf{x}\|_2 \quad = \quad \left(x_1^2 + x_2^2 + \ldots\ldots + x_n^2\right)^{\frac{1}{2}} = \left(\sum_{i=1}^{n} x_i^2\right)^{\frac{1}{2}}$$

Another simple norm is L_∞-norm.

L_∞**-norm** is defined by

$$\|\mathbf{x}\|_\infty \quad = \quad \max_{1 \leq i \leq n} |x_i|$$

A third norm often used is L_1-norm.

L_1**-norm** is defined by

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

In general, L_p**-norm** is defined by

$$\|\mathbf{x}\|_p = (|x_1|^p + |x_2|^p + ... + |x_n|^p)^{\frac{1}{p}} = \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}$$

Example 1

Consider the following vectors in $\mathbb{R}^3$.

(a) $\mathbf{x} = (1,2,3)^T$ (b) $\mathbf{x} = (0,2,3)^T$ (c) $\mathbf{x} = (0,0,3)^T$

Find $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$ and $\|\mathbf{x}\|_\infty$ for the above three vectors.

(a) Here $x_1 = 1$, $x_2 = 2$ and $x_3 = 3$ so that we have

$$\begin{aligned}\|\mathbf{x}\|_1 &= \sum_{i=1}^{3} |x_i| = 6\\ \|\mathbf{x}\|_2 &= \left(x_1^2 + x_2^2 + x_3^2\right)^{\frac{1}{2}} = \sqrt{1^2 + 2^2 + 3^2} = \sqrt{14}\\ \|\mathbf{x}\|_\infty &= \max_{1\le i\le 3} |x_i| = 3\end{aligned}$$

(b) For this case, $x_1 = 0$, $x_2 = 2$ and $x_3 = 3$

$$\begin{aligned}\|\mathbf{x}\|_1 &= \sum_{i=1}^{3} |x_i| = 5\\ \|\mathbf{x}\|_2 &= \left(x_1^2 + x_2^2 + x_3^2\right)^{\frac{1}{2}} = \sqrt{0^2 + 2^2 + 3^2} = \sqrt{13}\\ \|\mathbf{x}\|_\infty &= \max_{1\le i\le 3} |x_i| = 3\end{aligned}$$

(c) Here $x_1 = 0$, $x_2 = 0$ and $x_3 = 3$ so we obtain

$$\begin{aligned}\|\mathbf{x}\|_1 &= \sum_{i=1}^{3} |x_i| = 3\\ \|\mathbf{x}\|_2 &= \left(x_1^2 + x_2^2 + x_3^2\right)^{\frac{1}{2}} = \sqrt{0^2 + 0^2 + 3^2} = 3\\ \|\mathbf{x}\|_\infty &= \max_{1\le i\le 3} |x_i| = 3\end{aligned}$$

Example 2

Consider the vector space $\mathbb{R}^2$. Let us consider the following subsets of $\mathbb{R}^2$

$$\left\{\mathbf{x} : \ \|\mathbf{x}\| \leq 1, \quad \mathbf{x} = (x_1,\, x_2)^T \in \mathbb{R}^2\right\}$$

with L_1, L_2 and L_∞ norms.

Here we have

$$\begin{aligned} \|\mathbf{x}\|_1 &= \sum_{i=1}^{2} |x_i| = |x_1| + |x_2| \leq 1 \\ \|\mathbf{x}\|_2 &= \sqrt{x_1^2 + x_2^2} \leq 1 \\ \|\mathbf{x}\|_\infty &= \max_{1\leq i\leq 2} |x_i| \leq 1 \end{aligned}$$

Example 3

Prove the following property

$$||\mathbf{x} + \mathbf{y}||_\infty \ \leq \ ||\mathbf{x}||_\infty + ||\mathbf{y}||_\infty$$

Proof:

$$\begin{aligned} ||\mathbf{x} + \mathbf{y}||_\infty &= \max_{1\leq i\leq n} |x_i + y_i| \\ &\leq \max_{1\leq i\leq n} (|x_i| + |y_i|) \end{aligned}$$

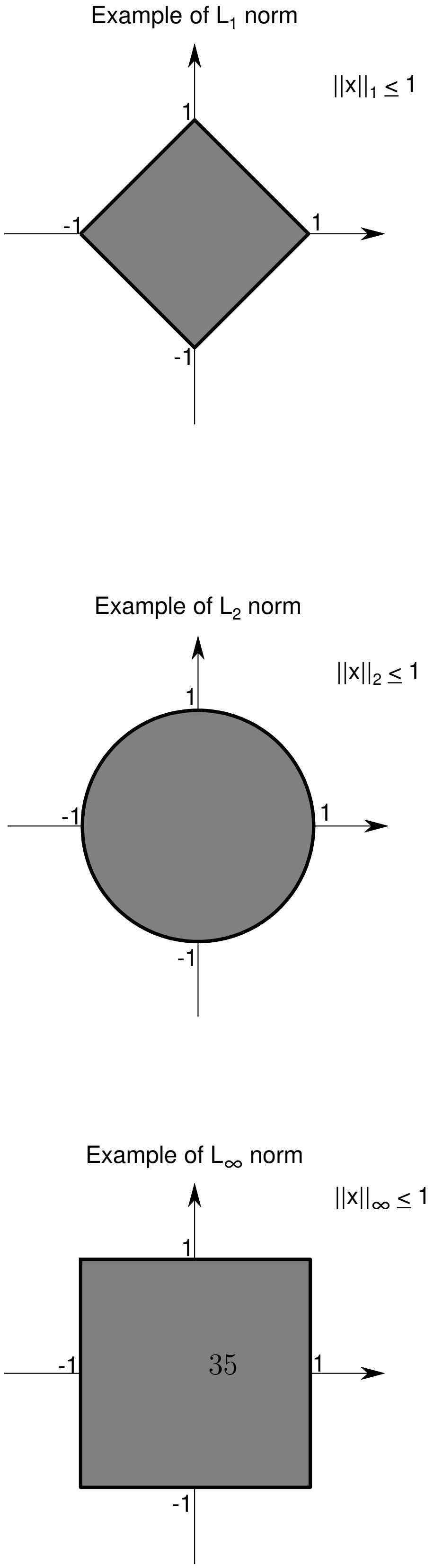
Thus we have

$$\begin{aligned} |\mathbf{x} + \mathbf{y}||_\infty &\leq \max_{1\leq i\leq n} |x_i| + \max_{1\leq i\leq n} |y_i| \\ &= ||\mathbf{x}||_\infty + ||\mathbf{y}||_\infty \end{aligned}$$

Matrix Norms

Let V be the set of all $n \times n$ matrices. A matrix norm on V is a real-valued function, $\|\ \|$ defined on V such that

Figure 1.3: Various norms in $\mathbb{R}^2$



Example of L_1 norm

Example of L_2 norm

Example of L_∞ norm

- $\|A\| \geq 0$ for any $A \in V$
- $\|A\| = 0$ if and only if $A = O$
- $\|\alpha A\| = |\alpha| \, \|A\|$ for scalar α and $A \in V$
- $\|A + B\| \leq \|A\| + \|B\|$ for any $A, B \in V$
- $\|AB\| \leq \|A\| \, \|B\|$ for any $A, B \in V$

Result:

If $\| \ \|$ is a vector norm on $\mathbb{R}^n$, then

$$\|A\| = \max_{\|x\|=1} \|A\mathbf{x}\|$$

is a matrix norm.

Also $\|A\mathbf{x}\| \leq \|A\| \, . \, \|\mathbf{x}\|$ for any vector $\mathbf{x}$.

Various matrix norms can be associated with various vector norms, i.e., every vector norm can produce an associated matrix norm. Although matrix norms are obtained different ways, the norms we consider are those obtained from L_∞ and L_2 norms.

L_∞ **matrix norm:**

If $A = (a_{ij})$ is an $n \times n$ matrix, then

$$\begin{aligned} \|A\|_\infty &= \max_{\|x\|_\infty = 1} \|A\mathbf{x}\|_\infty \\ &= \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}| = \text{maximum absolute row sum} \end{aligned}$$

L_2 **matrix norm:**

If $A = (a_{ij})$ is an $n \times n$ matrix, then

$$\|A\|_2 = \max_{1 \leq i \leq n} |\sigma_i|$$

where σ_i represents a singular value of A, i.e., σ_i^2 is an eigenvalue of A^TA. The largest eigenvalue of A^TA is called the spectral radius of A^TA and denoted by $\rho\left(A^TA\right)$. Thus we have

$$\|A\|_2 = \sqrt{\rho\left(A^TA\right)}$$

L_1 **matrix norm:**

If $A = (a_{ij})$ is an $n \times n$ matrix, then

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}| = \text{maximum absolute column sum}$$

Condition Number

Let A be an $n \times n$ square matrix. Then the condition number of A, denoted by $\kappa(A)$ is defined as

$$\kappa(A) = \|A\|_\infty \,\mathbf{.}\, \left\|A^{-1}\right\|_\infty$$

Condition number of a singular matrix is infinite.

Results:

(i) If $\mathbf{A}$ is invertible show that $\kappa(\mathbf{A}) \geq 1$
(ii) $\kappa(\mathbf{A}) \geq \frac{\|\mathbf{A}\|\,\|\mathbf{x}\|}{\|\mathbf{b}\|}$ if $\mathbf{Ax} = \mathbf{b}$.

Proof:

(i) If $\mathbf{A}$ is invertible, we have $\mathbf{AA}^{-1} = \mathbf{I}$. Thus,

$$1 = \|\mathbf{I}\| = \left\|\mathbf{AA}^{-1}\right\| \leq \|\mathbf{A}\|\left\|\mathbf{A}^{-1}\right\| = \kappa(\mathbf{A})$$

(ii) $\mathbf{Ax} = \mathbf{b} \Rightarrow \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$. Taking norm on both sides, we have

$$\|\mathbf{x}\| = \left\|\mathbf{A}^{-1}\mathbf{b}\right\| \leq \left\|\mathbf{A}^{-1}\right\|\|\mathbf{b}\|$$

Now, multiplying by $\|\mathbf{A}\|$ on both side,

$$\|\mathbf{A}\| \, \|\mathbf{x}\| \leq \|\mathbf{A}\| \left\|\mathbf{A}^{-1}\right\| \|\mathbf{b}\| = \kappa\left(\mathbf{A}\right) \|\mathbf{b}\|$$

yielding

$$\kappa\left(\mathbf{A}\right) \geq \frac{\|\mathbf{A}\| \, \|\mathbf{x}\|}{\|\mathbf{b}\|}.$$

Example 4

Consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 3 \\ 2 & 4 & 7 \end{bmatrix}$$

Find L_∞ and L_1 norms of A. Using L_∞ norm, find $\kappa\left(A\right).$

Here for L_∞ norm, we have

$$\|A\|_\infty = \max_{1\leq i\leq n} \sum_{j=1}^{n} |a_{ij}| = 2+4+7 = 13$$

Here for L_1 norm, we have

$$\|A\|_1 = \max_{1\leq j\leq n} \sum_{i=1}^{n} |a_{ij}| = 3+3+7 = 13$$

Condition number $\kappa\left(A\right)$ of A is given by

$$\kappa\left(A\right) = \|A\|_\infty \centerdot \left\|A^{-1}\right\|_\infty$$

Also A^{-1} can be obtained as

$$A^{-1} = \begin{bmatrix} 9 & -2 & -3 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

Hence

$$\kappa\left(A\right) = \|A\|_\infty \centerdot \left\|A^{-1}\right\|_\infty = 13 \times 14 = 182$$

Why Condition Number is Important:

Ex:

Consider

$$\begin{aligned} 46x + 45y &= 91 \\ 45x + 44y &= 89 \end{aligned}$$

Here solutions of $Ax = b$ are $x = 1,\ y = 1$. As $|A| = -1$, inverse of A

$$A^{-1} = \begin{bmatrix} -44 & 45 \\ 45 & -46 \end{bmatrix}$$

This yields $||A||_\infty = 91 = ||A^{-1}||_\infty$ so that $\kappa(A) = 8281$ which is much larger compared to 1. If we change the input $b = (90.9,\ 89.1)^T$, then solutions are $x = 9.9,\ y = -8.1$. Small change in input makes a large change in output.

Exercise

1. Find $\|\mathbf{x}\|_1$, $\|\mathbf{x}\|_2$ and $\|\mathbf{x}\|_\infty$ for the following vectors

   (a) $\mathbf{x} = \begin{bmatrix} 5 \\ -2 \\ 3 \end{bmatrix}$, (b) $\mathbf{x} = \begin{bmatrix} -7 \\ 0 \\ 4 \end{bmatrix}$

2. Prove the following properties

   (a) $\|\alpha\mathbf{x}\|_\infty = |\alpha|\,\|\mathbf{x}\|_\infty$ (b) $||\mathbf{x}+\mathbf{y}||_1 \leq ||\mathbf{x}||_1 + ||\mathbf{y}||_1$ (c) $||\mathbf{x}+\mathbf{y}||_2 \leq ||\mathbf{x}||_2 + ||\mathbf{y}||_2$

3. Find $\|A\|_\infty$, $\|A\|_1$ and $\kappa(A)$ for the following matrices

   (a) $A = \begin{bmatrix} 5 & 6 & -9 \\ 1 & 2 & 3 \\ 0 & 7 & 2 \end{bmatrix}$ (b) $A = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{bmatrix}$

1.6 Review of Calculus

In numerical analysis, some results from calculus are frequently used. For example, Intermediate Value Theorem, Mean Value Theorem will be used to solve equations. Taylor's Theorem is important in understanding error estimation, interpolation and it contributes towards the solution of differential equations.

1.6.1 Intermediate Value Theorem

Let f be continuous function on the interval $[a,\, b]$ and M be any number between $f(a)$ and $f(b)$ where $f(a) \neq f(b)$. Then there exists a number $c \in (a, b)$ such that $f(c) = M$.

Example 1

Show that $f(x) = x^2 - 3$ on the interval $[1, 3]$ must take on values 0 and 1.

Here $f(1) = -2$ and $f(3) = 6$. f is continuous on $[1, 3]$. Hence f must take values between -2 and 6.

We see that $f(\sqrt{3}) = 0$ and $\sqrt{3} \in (1, 3)$. Also $f(2) = 1$ and $2 \in (1,\, 3)$.

1.6.2 Continuous Limit

Let f be continuous function in a neighborhood of x_0, and assume $\lim_{n\to\infty} x_n = x_0$. Then

$$\lim_{n\to\infty} f(x_n) \quad = f(\lim_{n\to\infty} x_n) = \quad f(x_0)$$

1.6.3 Mean Value Theorem (MVT)

Let f be continuously differentiable function on the interval $[a, b]$. Then there is a number c in (a, b) such that

$$f'(c) \quad = \quad \frac{f(b) - f(a)}{b - a}$$

Example 2

Illustrate MVT for $f(x) = x^3 - x, \ \ a = 0, \ b = 2.$

Here f is a polynomial, it is continuously differentiable in $[0, 2]$. Since $f(0) = 0, \ f(2) = 6$ and $f'(x) = 3x^2 - 1$, we have $3c^2 - 1 = 3$, i.e., $c = \pm\frac{2}{\sqrt{3}}$. c must lie in $(0, 2)$ so $c = \frac{2}{\sqrt{3}}$.

1.6.3.1 Rolle's Theorem

Let f be continuously differentiable function on the interval $[a, b]$ and $f(a) = f(b)$. Then there is a number c in (a, b) such that

$$f'(c) \quad = \quad 0$$

Rolle's theorem is a special case of MVT with $f(a) = f(b)$.

1.6.4 Taylor's Theorem with Remainder

Let x and x_0 be real numbers and let f be $(n+1)$ times continuously differentiable on the interval between x and x_0. Then there exists a number c between x and x_0 such that

$$f(x) \quad = \quad P_n(x) + R_n(x)$$

where

$$\begin{aligned} P_n(x) &= f(x_0) + (x - x_0)\, f'(x_0) + \frac{(x - x_0)^2}{2!} f''(x_0) + \frac{(x - x_0)^3}{3!} f'''(x_0) + \ldots. \\ &\qquad \ldots\ldots\ldots\ldots + \frac{(x - x_0)^{n-1}}{(n-1)!} f^{(n-1)}(x_0) + \frac{(x - x_0)^n}{n!} f^{(n)}(x_0) \\ R_n(x) &= \frac{(x - x_0)^{n+1}}{(n+1)!} f^{(n+1)}(c) \end{aligned}$$

Taylor's Series

Here are some

$$\begin{aligned}
e^x &= 1+x+\frac{x^2}{2!}+\frac{x^3}{3!}+\ldots\ldots\ldots \\
\cos x &= 1-\frac{x^2}{2!}+\frac{x^4}{4!}-\ldots\ldots\ldots \\
\sin x &= x-\frac{x^3}{3!}+\frac{x^5}{5!}-\ldots\ldots\ldots \\
\tan^{-1} x &= x-\frac{x^3}{3}+\frac{x^5}{5}-\ldots\ldots\ldots \\
\ln(1+x) &= x-\frac{x^2}{2}+\frac{x^3}{3}-\ldots\ldots\ldots \\
\ln(1-x) &= -x-\frac{x^2}{2}-\frac{x^3}{3}-\ldots\ldots\ldots \\
\cosh x &= 1+\frac{x^2}{2!}+\frac{x^4}{4!}+\ldots\ldots\ldots \\
\sinh x &= x+\frac{x^3}{3!}+\frac{x^5}{5!}+\ldots\ldots\ldots
\end{aligned}$$

1.6.5 Mean Value Theorem for Integrals

Let f be a continuous function on $[a, b]$,and let g be an integrable function that does not change sign on $[a, b]$. Then there exists a number c between a and b such that

$$\int\limits_a^b f(x)g(x)\,dx = f(c)\int\limits_a^b g(x)\,dx$$

1.6.6 Taylor's Theorem in Two Variables

Let $(x,\, y)$ and $(x+h,\, y+k)$ be two points in the rectangle $[a,\, b]\times[c,\, d]\subseteq\mathcal{R}^2$. Let f be $(n+1)$ times continuously differentiable in $[a,\, b]\times[c,\, d]$ .Then

$$f(x+h,\, y+k) = P_n(x,\, y)+R_n(h,\, k)$$

where

$$
\begin{aligned}
P_n(x,\, y) &= \sum_{j=0}^{n} \frac{1}{j!} \left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y} \right)^{j} f(x, y) \\
R_n(h,\, k) &= \frac{1}{(n+1)!} \left(h\frac{\partial}{\partial x} + k\frac{\partial}{\partial y} \right)^{n+1} f^{(n+1)}(x + \theta h,\, y + \theta k)
\end{aligned}
$$

1.7 Convergence

In lot of algorithms (iterative in nature used later), s sequence of approximations, $\{x_n\}$, will be generated that converge to the desired solution, x^*. If several techniques are available to solve a specific problem, the technique whose sequence converges more rapidly than other techniques is preferred. To help in comparing competeing methods, two quantitative measaures of convergence speed are presented below.

Rate of Convergence:

Definition: The sequence $\{x_n\}$ converges to the value x^* provided

$$\lim_{n\to\infty} x_n = x^*$$

or, equivalently,

$$\lim_{n\to\infty} |x_n - x^*| = 0.$$

The value to which the sequence converges, x^*, is called the limit of the sequence. If the limit $\lim_{n\to\infty} x_n$ does not exist, the sequence is said to be divergent.

Definition: Let $\{x_n\}$ be a sequence that converges to a number x^*. If there is a sequence $\{\lambda_n\}$ that converges to zero and a positive constant β (independent of n), such that

$$|x_n - x^*| \leq \beta |\lambda_n|$$

for all sufficiently large n, then $\{x_n\}$ is said to converge to $x^*$with **rate of convergence $\mathcal{O}(\lambda_n)$ (Big-O notation).** When $\{x_n\}$ is said to converge to $x^*$with rate of convergence $\mathcal{O}(\lambda_n)$, it is written as

$$x_n = x^* + \mathcal{O}(\lambda_n).$$

For example, a sequence with rate of convergence $\mathcal{O}\left(\frac{1}{n}\right)$ converges slower than one with rate of convergence $\mathcal{O}\left(\frac{1}{n^6}\right)$, which converges more slowly than a sequence with rate of convergence $\mathcal{O}\left(\frac{1}{2^n}\right)$.

Example: Comparing Rate of Convergence

Consider the sequences

$$\left\{\frac{n+4}{n+9}\right\} \quad \text{and} \quad \left\{\frac{3^n+4}{3^n+9}\right\}$$

n	$\frac{n+4}{n+9}$	$\frac{3^n+4}{3^n+9}$
1	0.500000	0.583333
2	0.545455	0.722222
3	0.583333	0.861111
4	0.615385	0.944444
5	0.642857	0.980159
6	0.666667	0.993225
7	0.687500	0.997723
8	0.705882	0.999239
9	0.722222	0.999746
10	0.736842	0.999915
11	0.750000	0.999972
12	0.761905	0.999991
13	0.772727	0.999997
14	0.782609	0.999999
15	0.791667	1.000000

Since

$$\lim_{n\to\infty}\left\{\frac{n+4}{n+9}\right\}=1 \quad \text{and} \quad \lim_{n\to\infty}\left\{\frac{3^n+4}{3^n+9}\right\}=1$$

it means that both sequences converge to the limit 1. Although both have the same limit, it is seen from the table that the sequence $\left\{\frac{3^n+4}{3^n+9}\right\}$ approaches 1 much faster that the sequence $\left\{\frac{n+4}{n+9}\right\}$.

To obtain rate of convergence, we have

$$\left|\frac{n+4}{n+9}-1\right| = \frac{5}{n+9} < \frac{5}{n}$$

So, we have $\beta = 5$ and $\lambda_n = \frac{1}{n}$. Thus the sequence $\left\{\frac{n+4}{n+9}\right\}$ converges to 1 with rate of convergemce $\mathcal{O}\left(\frac{1}{n}\right)$. Here $\lim_{n\to\infty} \lambda_n \to 0$.

For the sequence $\left\{\frac{3^n+4}{3^n+9}\right\}$, we get

$$\left|\frac{3^n+4}{3^n+9}-1\right| = \frac{5}{3^n+9} < \frac{5}{3^n}$$

So, we have $\beta = 5$ and $\lambda_n = \frac{1}{3^n}$. Thus the sequence $\left\{\frac{3^n+4}{3^n+9}\right\}$ converges to 1 with rate of convergemce $\mathcal{O}\left(\frac{1}{3^n}\right)$. Obviously, $\lambda_n = \frac{1}{3^n}$ approaches zero faster that $\lambda_n = \frac{1}{n}$ as $n \to \infty$.

Rate of Convergence for a Function:

Definition: Let f be a function defined on the interval $(a,\, b)$ that contains $x = 0$ and suppose $\lim_{x\to 0} f(x) = L$. If there exists a function g for which $\lim_{x\to 0} g(x) = 0$ and a positive constant C such that

$$|f(x) - L| \leq C\,|g(x)|$$

for all sufficiently small values of x, then $f(x)$ is said to converge to L with rate of convergence $\mathcal{O}\left(g(x)\right)$.

Example

Consider the function

$$f(x) \quad = \quad \frac{\sin x - x}{x^3}$$

What is the limit of f as $x \to 0$? At what rate does f converge to this limit?

Using Taylor's thorem, we have

$$\sin x \quad = \quad x - \frac{x^3}{3!} + \frac{x^5}{5!}\cos\xi$$

for some ξ between 0 and x. Hence,

$$\frac{\sin x - x}{x^3} \quad = \quad -\frac{1}{6} + \frac{1}{120}\, x^2 \cos\xi$$

Now, we have

$$\left| \frac{\sin x - x}{x^3} + \frac{1}{6} \right| \quad = \quad \frac{1}{120}\, x^2 \cos\xi \leq \frac{1}{120}\, x^2,$$

it follows that $\lim_{x\to 0} f(x) = -\frac{1}{6}$ and the rate of convergence is $\mathcal{O}\left(x^2\right)$.

Order of Convergence:

Order of convergence offers a different measure of convergence speed that rate of convergence. While rate of convergence provides individually the terms in the sequence of error values $e_n = |x_n - x^*|$, order of convergence provides the connection between consecutive error values, determining the efficiency with which each iteration diminishes the estimated error.

Let $\{x_n\}$ be a sequence that converges to x^* with $x_n \neq x^*$ for all n. If there exist two positive constants α and β such that

$$\lim_{n\to\infty} \frac{|x_{n+1} - x^*|}{|x_n - x^*|^{\alpha}} = \lim_{n\to\infty} \frac{e_{n+1}}{e_n^{\alpha}} \quad = \quad \beta,$$

then $\{x_n\}$ converges to x^* of order α with asysmptotic error β. Here $e_n = |x_n - x^*|$, the error at nth iteration.

Notes:

If a sequence $\{x_n\}$ converges to x^* of order α, the error satisfies the asymptotic relation $e_{n+1} \approx \beta\, e_n^{\alpha}$. An iterative method is said to of order α if the sequence it generates converges of order α. In general, a sequence with a higher value of α converges faster than a sequence with a lower value of α.

- **Linearly Convergent:** If $\alpha = 1$, then the sequence is said to converge linearly.

- **Quadratically Convergent:** If $\alpha = 2$, then the sequence is said to converge quadratically.

- **Superlinear Convergent:** If $1 < \alpha < 2$, then the convergence is superlinear. Superlinear convergence is better than linear convergence and not as good as quadratic convergence.

- Noninteger values for α are possible.

If $\alpha = 1$, the sequence of error values satisfies

$$|x_n - x^*| = e_n \approx \beta\, e_{n-1} \approx \beta^2\, e_{n-2} \approx \beta^3\, e_{n-3} \approx .... \approx \beta^n\, e_0$$

Hence, a linearly convergent (of order 1) sequence converges with rate of convergence $\mathcal{O}\left(\beta^n\right)$.

Follwing table demonstrates the differences for order of convergence (Linear and Quadratic). Taking $\beta = 0.6$ and $e_0 = 1.0$, we have

Error	*Linear*	*Quadratic*	*Cubic*
e_1	$6.00000e-01$	$6.00000e-01$	$6.00000e-01$
e_2	$3.60000e-01$	$2.16000e-01$	$1.29600e-01$
e_3	$2.16000e-01$	$2.79936e-02$	$1.30607e-03$
e_4	$1.29600e-01$	$4.70185e-04$	$1.33675e-09$
e_5	$7.77600e-02$	$1.32644e-07$	$1.43318e-27$
e_6	$4.66560e-02$	$1.05567e-14$	$1.76626e-81$
e_7	$2.79936e-02$	$6.68665e-29$	$3.30611e-243$

There is a dramatic difference even between linear and quadratic methods. The linear method would take 127 iterations to achieve the accuracy attained by the quadratic method in just 7 iterations. Even the more modest accuarcy achieved by the quadratic method in 6 iterations would take the linear method 63 iterations.

Unless each iteration of the quadratic method requires significantly more work than each iteration of the linear method.

On the other hand, there is only a slight difference (2 or 3 iterations) between quadratic and cubic methods. In practice, the extra work needed to achieve cubic convergence would not be justified.

Big O and Little o Notations

Consider standard ways of comparing two sequences of two functions.

Let $\{x_n\}$ and $\{y_n\}$ be two different sequences.

We write

$$x_n = \mathcal{O}(y_n)$$

if there are constants N and C such that

$$|x_n| \leq C\,|y_n|$$

when $n \geq N$.

We write

$$x_n = o(y_n)$$

if

$$\lim_{n\to\infty} \frac{x_n}{y_n} = 0$$

These two give a coarse method of comparing two sequences. They are frequently used when both sequences converge to 0. Thus, if $x_n \to 0$, $y_n \to 0$, and $x_n = \mathcal{O}(y_n)$, then x_n converges to 0 at least as rapidly as y_n does. If $x_n = o(y_n)$, then x_n converges to 0 more rapidly than y_n does.

Examples:

$$\begin{aligned}
\frac{n+1}{n^2} &= \mathcal{O}\left(\frac{1}{n}\right)\\
\frac{5}{n} + e^{-n} &= \mathcal{O}\left(\frac{1}{n}\right)\\
e^{-n} &= o\left(\frac{1}{n^2}\right)\\
\frac{1}{n\,\ln n} &= o\left(\frac{1}{n}\right)\\
\frac{1}{n} &= o\left(\frac{1}{\ln n}\right)
\end{aligned}$$

$\mathcal{O}\left(\frac{1}{n}\right)$ is a slow convergence, but $\mathcal{O}\left(\frac{1}{n!}\right)$ is a fast convergence.

Notation introduced is also used for functions other than sequences. For example, we have

$$\sin x \;=\; x - \frac{x^3}{3!} + \mathcal{O}\left(x^5\right) \qquad (x \to 0)$$

This means that there exist a neighborhood of 0 and a constant C, such that on that neighborhood

$$\left|\sin x - x + \frac{x^3}{3!}\right| \;\leq\; C\left|x^5\right|.$$

An equation of the form

$$f(x) \;=\; \mathcal{O}\left(g(x)\right) \qquad (x \to \infty)$$

means that constants x_0 and C exist so that $|f(x)| \leq C\,|g(x)|$ whenever $x \geq x_0$.

Example

$$\sqrt{x^2+1} \;=\; \mathcal{O}\left(x\right) \qquad (x \to \infty)$$

since $\sqrt{x^2+1} \leq 2x$ when $x \geq 1$.

In using notation $f(x) = \mathcal{O}\left(g(x)\right)$ or $f(x) = o\left(g(x)\right)$, it is important that we state what point of convergence is intended. For example, $x^{-2} = o\left(x^{-1}\right)$ as $x \to \infty$. This relation is reversed at 0: $x^{-1} = o\left(x^{-2}\right)$ as $x \to 0$.

In general,

$$f(x) \;=\; \mathcal{O}\left(g(x)\right) \qquad (x \to x^*)$$

Here, C is constant and a neighborhood of x^*such that $|f(x)| \leq C\,|g(x)|$ in the neighborhood.

$$f(x) \;=\; o\left(g(x)\right) \qquad (x \to x^*)$$

means $\lim_{x \to x^*} \frac{f(x)}{g(x)} = 0$.

Multiplicity:

A root $x^*$of the equation $f(x) = 0$ is called a root of multiplicity m if f can be expressed in the form

$$f(x) = (x - x^*)^m \, q(x)$$

where $lim_{x \to x^*} q(x) \neq 0$. A root of multiplicity one is called a SIMPLE root.

Example 1:

The polynomial $x^6+2x^5-8x^4-14x^3+11x^2+28x+12$ has a root of multiplicity 3 at $x = -1$, a root of multiplicity 2 at $x = 2$, and a simple root at $x = -3$ as

$$\begin{aligned} & x^6 + 2x^5 - 8x^4 - 14x^3 + 11x^2 + 28x + 12 \\ = \quad & (x+1)^3 (x-2)^2 (x+3) \, . \end{aligned}$$

Example 2:

The polynomial $x^8 - 22x^6 + 32x^5 + 69x^4 - 104x^3 - 104x^2 + 96x + 80$ has a root of multiplicity 4 at $x = 2$, a root of multiplicity 3 at $x = -1$, and a simple root at $x = -5$ as

$$\begin{aligned} & x^8 - 22x^6 + 32x^5 + 69x^4 - 104x^3 - 104x^2 + 96x + 80 \\ = \quad & (x-2)^4 (x+1)^3 (x+5) \, . \end{aligned}$$

For non obvious functions (non-polynomial functions), what do we do? Consider

$$f(x) = \ln \left(\frac{1+x}{1-x} \right) - 2x$$

Obviously, $f(0) = 0$. So the equation has a root at $x = 0$. What is the multiplicity of this root?

The following theorem is helpful in this respect:

Theorem:

Let f be a continuous function with m continuous derivatives. The equation $f(x) = 0$ has a root of multiplicity m at $x = x^*$if and only if

$$f\ (x^*) = f'\,(x^*) = f''\,(x^*) = ... = f^{\,(m-1)}\,(x^*) = 0, \text{ but } f^{\,(m)}\,(x^*) \neq 0.$$

For the example earlier, for the function

$$f(x) = \ln\left(\frac{1+x}{1-x}\right) - 2x$$

we have

$$\begin{aligned} f'\,(x) &= \frac{1}{1+x} + \frac{1}{1-x} - 2 = \frac{2x^2}{1-x^2} \\ f''\,(x) &= -\frac{1}{(1+x)^2} + \frac{1}{(1-x)^2} = \frac{4x}{(1-x^2)^2} \\ f'''\,(x) &= \frac{2}{(1+x)^3} + \frac{2}{(1-x)^3} = \frac{4(1+3x^2)}{(1-x^2)^3} \end{aligned}$$

so

$$f\ (0) = f'\,(0) = f''\,(0) = 0, \text{ but } f^{\,'''})\,(0) = 4 \neq 0.$$

Chapter 2

Solving Equations

One very important basic problem we encounter in scientific computing is Equation Solving. We will try to locate solutions x of the equation $f(x) = 0$. Why do we need to know more than one method to solve equations. Choice of the method will depend on the cost of evaluating the function and its derivative. Here we introduce methods such as the Bisection Method, Fixed Point Iteration, Newton's Method, Secant Method and Muller's Method. We discuss their computational complexity and rates of convergence.

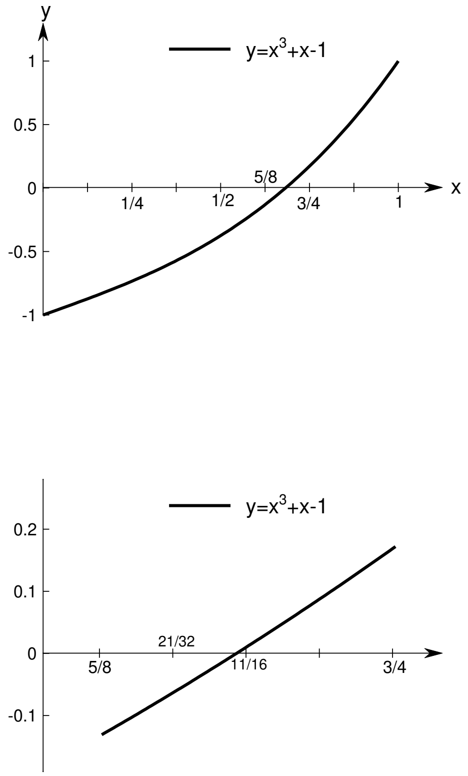
Definitions:

Root: A function $f(x)$ has a root at $x = x^*$ if $f(x^*) = 0$. Also x^* is called a solution of $f(x) = 0$. For example, if $f(x) = x^2 - 3x - 10$, then $f(x) = 0$ has two roots (or $f(x)$ has two zeros) at $x = 5$ and $x = -2$ since $f(5) = 5^2 - 15 - 10 = 25 - 25 = 0$ and $f(-2) = (-2)^2 - 3(-2) - 10 = 4 + 6 - 10 = 0$.

2.1 The Bisection Method

This method works like looking up a name or word in a phone book or dictionary. First open the book by dividing it in two equal parts. Then check if the name/word is on the right side or left side. If the name/word

Figure 2.1: Bisection Method



is on right side, consider the right side and repeat the previous step. If not, consider the left side and repeat the process.

Theorem:

Consider a function $f(x)$ which is continuous on $[a,\, b]$ and satisfies $f(a)f(b) < 0$. Then there is a number $c \in (a, b)$ such that $f(c) = 0$.

This is a corollary of the Intermediate Value Theorem.

Following graph describes the Bisection Method.

Algorithm for Bisection Method:

```
    Given an interval $[a,\, b]$ such that $f(a)f(b) < 0$
    while $(b - a)/2 > TOL$
```

```
        $c = \frac{a+b}{2}$
        if $f(c) = 0$, stop, end
        if $f(a)f(c) < 0$
            $b = c$
        else
            $a = c$
        end
    end
    Approximate root is $\frac{a+b}{2}$.
```

Example 1

Use the Intermediate Value Theorem to find an interval of length one that contains a root of the equation.

(a) $x^3 = 9$ (b) $3x^3 + x^2 = x + 5$

(a) Here we consider $f(x) = x^3 - 9$ so that we have $f(2) = -1$ and $f(3) = 18$. Since $f(2)f(3) < 0$, there is root between $x = 2$ and $x = 3$.

(b) Here $f(x) = 3x^3 + x^2 - x - 5$. Thus we have $f(1) = -2$ and $f(2) = 21$. Since $f(1)f(2) < 0$, there is root between $x = 1$ and $x = 2$.

Example 2

Consider the function $f(x) = x^3 + x - 1$. Using the Bisection Method find a root of f on the interval $[0,\ 1]$.

Here we start with $a = 0,\ b = 1$. Since $f(0)f(1) < 0$, there is a root on $(0, 1)$.

See table 2.1.

Hence the approximate root is $\frac{\frac{349}{512} + \frac{699}{1024}}{2} \approx 0.6821$

2.1.1 Convergence Analysis

Under what circumstances will the sequence of approximations generated by the bisection method converge to a root of $f(x) = 0$? When the sequence does converge, what is the speed of convergence? The follwoing theorem answers some of these.

Table 2.1: Bisection Method

a	$c = \frac{a+b}{2}$	b	$f(a)$	$f(c)$	$f(b)$	new interval
0	$\frac{1}{2}$	1	-	-	$+$	$\left(\frac{1}{2},\ 1\right)$
$\frac{1}{2}$	$\frac{3}{4}$	1	-	$+$	$+$	$\left(\frac{1}{2},\ \frac{3}{4}\right)$
$\frac{1}{2}$	$\frac{5}{8}$	$\frac{3}{4}$	-	-	$+$	$\left(\frac{5}{8},\ \frac{3}{4}\right)$
$\frac{5}{8}$	$\frac{11}{16}$	$\frac{3}{4}$	-	$+$	$+$	$\left(\frac{5}{8},\ \frac{11}{16}\right)$
$\frac{5}{8}$	$\frac{21}{32}$	$\frac{11}{16}$	-	-	$+$	$\left(\frac{21}{32},\ \frac{11}{16}\right)$
$\frac{21}{32}$	$\frac{43}{64}$	$\frac{11}{16}$	-	-	$+$	$\left(\frac{43}{64},\ \frac{11}{16}\right)$
$\frac{43}{64}$	$\frac{87}{128}$	$\frac{11}{16}$	-	-	$+$	$\left(\frac{87}{128},\ \frac{11}{16}\right)$
$\frac{87}{128}$	$\frac{175}{256}$	$\frac{11}{16}$	-	$+$	$+$	$\left(\frac{87}{128},\ \frac{175}{256}\right)$
$\frac{87}{128}$	$\frac{349}{512}$	$\frac{175}{256}$	-	-	$+$	$\left(\frac{349}{512},\ \frac{175}{256}\right)$
$\frac{349}{512}$	$\frac{699}{1024}$	$\frac{175}{256}$	-	$+$	$+$	$\left(\frac{349}{512},\ \frac{699}{1024}\right)$

Theorem:

Let f be a continuous function on the closed interval $[a,\, b]$, and suppose that $f(a)f(b) < 0$. The bisection method generates a sequence of approximations $\{x_n\}_{n=1}^{\infty}$ which converges to a root $x^* \in (a,\, b)$ with the property

$$|\, x_n - x^*| \leq \frac{b-a}{2^n}$$

Proof:

Let x^*be a root of the equation $f(x) = 0$ in $[a,\, b]$ (or x^*be a zero of $f(x)$ in $[a,\, b]$). Let $[a_n\, b_n]$ be the interval at step n and x_n is the mid-point of this interval.

Step $-$ 1 For $[a_1,\, b_1]$, length of the interval $= b_1 - a_1 = b - a$

$$\text{midpoint} = x_1 = \frac{a_1 + b_1}{2}$$

Step $-$ 2 For $[a_2,\, b_2]$, length of the interval $= b_2 - a_2 = \dfrac{1}{2}\,(b_1 - a_1) = \dfrac{b-a}{2}$

$$\text{midpoint} = x_2 = \frac{a_2 + b_2}{2}$$

Step $-$ 3 For $[a_3,\, b_3]$, length of the interval $= b_3 - a_3 = \dfrac{b_2 - a_2}{2} = \dfrac{b-a}{2^2}$

$$\text{midpoint} = x_3 = \frac{a_3 + b_3}{2}$$

$\vdots$

Step $- n$ For $[a_n,\, b_n]$, length of the interval $= b_n - a_n = \dfrac{b_{n-1} - a_{n-1}}{2} = \dfrac{b-a}{2^{n-1}}$

$$\text{midpoint} = x_n = \frac{a_n + b_n}{2}$$

Now, for each $n \geq 1$, we have

$$b_n - a_n = \frac{1}{2^{n-1}}\,(b-a) \qquad \text{and} \qquad x^* \in (a_n,\, b_n)\,.$$

Since $x_n = \frac{a_n + b_n}{2}$ for all $n \geq 1$, it follows that

$$|\, x_n - x^*| \leq \frac{1}{2}\,(b_n - a_n) = \frac{b-a}{2^n}$$

Thus, the sequence $\{x_n\}_{n=1}^{\infty}$ converges to x^*as $n \to \infty$ (since $\frac{1}{2^n} \to 0$ as $n \to \infty$), in other words

$$\lim_{n\to\infty} x_n = x^*.$$

Thus, the rate of convergence $O\left(\frac{1}{2^n}\right)$.

Notes:

- Bisection method states that it converges to a root of f , not the root of f. The condition $f(a)f(b) < 0$ implies different signs at the endpoints of the interval, which guarantees the existence of a root, not the uquiness. There may be more than one root on the interval and there is no way to know , a priori, to which root the equence will converge, but it will converge to one of them.

- Since $|\,x_n - x^*|$ is the absolute error in the approximation x_n, the expression on the right hand side of the inequality at the end of the thoerem is referred to as a theretical error bound. The error at any stage of the iteration can never be larger than this quantity. Working with problems for which the analytical solution is known and verufying that a theretical error bound is staisfied is a powerful tool for eliminating human errors in develpoing computer codes.

- The requirement that an interval $[a\,b]$ be found such that $f(a)f(b) < 0$ inplies that the bisection method cannot be used to locate roots of even multiplicity. For such roots, the sign of the function does not change on either side of the root. This restriction is common to all simple enclosure techniques and is not peculiar to the bisection method.

2.1.2 The number of iterations n for a given tolerance parameter ϵ:

Suppose the error tolerance is prescribed as ϵ, . i.e.

$$|\,x_n - x^*| < \epsilon$$

We have shown that

$$|\,x_n - x^*| \leq \frac{b-a}{2^n}$$

Hence, we can write

$$\frac{b-a}{2^n} < \epsilon$$

which gives $2^n\epsilon > (b-a)$.

Taking logarithm on both sides, we get

$$\begin{aligned} \ln\epsilon + n\ln 2 &> \ln(b-a) \\ n &> \frac{\ln(b-a) - \ln\epsilon}{\ln 2} \end{aligned}$$

Example:

Determine the number of iterations necessary to solve $f(x) = xxx$ in the interval [1, 2] with accuracy 10^{-3}.

2.1.3 Stopping Criteria

Let x_n denote the approximation at step n. If the tolerance parameter is ϵ_1, a small number, we can set the stopping criteria as

$$|x_{n+1} - x_n| \;\; < \;\; \epsilon_1$$

Also,

We have shown that

$$|\,x_n - x^*| \leq \frac{b-a}{2^n}$$

Similarly, we can write

$$|\,x_{n+1} - x^*| \leq \frac{b-a}{2^{n+1}}$$

Now,

$$x_{n+1} - x_n = x_{n+1} - x^* + x^* - x_n$$

yields

$$\begin{aligned}|x_{n+1} - x_n| &= |x_{n+1} - x^* + x^* - x_n| \\ &\leq |x_{n+1} - x^*| + |x_n - x^*|\end{aligned}$$

Thus,

$$\begin{aligned}|x_{n+1} - x_n| &\leq \frac{b-a}{2^{n+1}} + \frac{b-a}{2^n} = \frac{b-a}{2^n}\left[\frac{1}{2} + 1\right] \\ &< \frac{b-a}{2^n} \times 2 = \frac{b-a}{2^{n-1}} = \epsilon_1\end{aligned}$$

2.1.4 Efficiency and Accuracy

How fast and how accurate Bisection Method is?

In Bisection method, we start with the interval $[a,\, b]$. Let $[a_n, b_n]$ be the interval after n steps. This interval is of length $\frac{b-a}{2^{n-1}}$. We choose the midpoint $x_n = \frac{a_n+b_n}{2}$ as our approximatio for the solution x^* which is half the interval length. Thus after n steps of the Bisection method, we find

$$\text{Solution Error} \;\; = \;\; e_n = |x_n - x^*| \leq \frac{b-a}{2^n}$$

Function evaluations $= n+1$ if starting at $a_1 = a$ and $b_1 = b$. (If starting at $a_0 = a$ and $b_0 = b$, function evaluations $= n+2$). Thus the efficiency and accuracy of the Bisection Method depends on the efficiency and accuracy in function evaluation.

Converegence is linear. Worst scenario is $|x_n - x^*| = \frac{b-a}{2^n}$, so

$$\frac{e_{n+1}}{e_n} = \frac{|x_{n+1} - x^*|}{|x_n - x^*|} = \frac{\frac{|b-a|}{2^{n+1}}}{\frac{|b-a|}{2^n}} = \tfrac{1}{2}$$

which indicate that order of convergence is 1. Also $|e_{n+1}| \sim \frac{1}{2}|e_n|$

Exercise

1. Consider the equation $x^3 - 9 = 0$ and the interval $[2, 3]$. Apply (manually, i.e., without programming) three steps to obtain an approximate root using Bisection Method.

2. Approximate the root of $3x^3 + x^2 - x - 5 = 0$ in the interval $[1, 2]$ to five correct decimal places by writing a computer program using any programming language. (Use Bisection Method).

2.2 Fixed-Point Iteration (FPI)

Definition:

A number x^* is a fixed point of a function g if $g(x^*) = x^*$. A fixed point is a point where the line $y = x$ crosses the graph of $y = g(x)$. Here we will find the solutions to fixed-point problems and discuss the relation between the root-finding problems and the fixed-point problems.

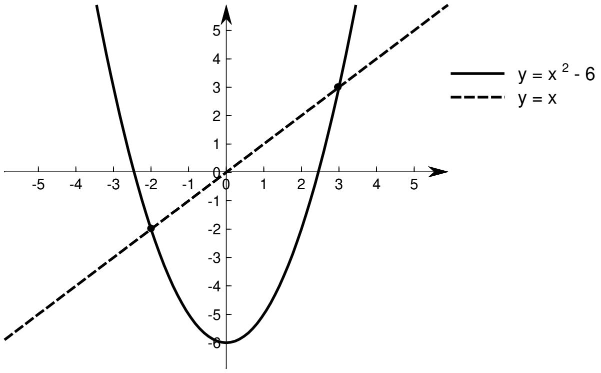
Example 1

Consider the function $g(x) = x^2 - 6$ in $-4 \leq x \leq 4$. Here g has two fixed points at $x^* = -2$ and $x^* = 3$ as

$$g(-2) = (-2)^2 - 6 = -2 \qquad \text{and} \qquad g(3) = 3^2 - 6 = 3.$$

See figure 2.2

Figure 2.2: Fixed-point example 1



Example 1 is equivalent to finding the root of $f(x) = g(x) - x = x^2 - x - 6$.

Example 2

Consider the function $g(x) = 2 - 2x$. Here g has a fixed point at $x^* = \frac{2}{3}$ as $g\left(\frac{2}{3}\right) = 2 - 2 \times \frac{2}{3} = \frac{2}{3}$.

Example 3

Consider the function $g(x) = 2^{1-x}$. Here g has a fixed point at $x^* = 1$ as $g(1) = 2^{1-1} = 1$.

See figure 2.3

Every equation $f(x) = 0$ can be turned into a fixed-point problem $g(x) = x$, not only by one way but in many different ways.

Example 4

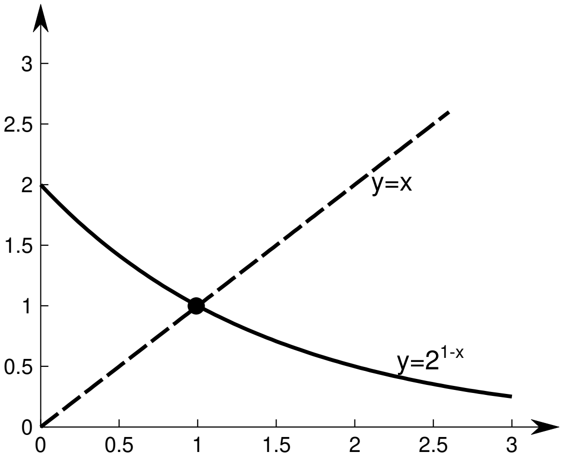
Consider the Example 2 of Bisection Method, i.e., finding the roots of $f(x) = x^3 + x - 1$ in $0 \leq x \leq 1$.

Now

$$x^3 + x - 1 \quad = \quad 0$$

can be expressed as

Figure 2.3: Fixed-point example 3



Choice 1:

$$\begin{aligned} x^3 + x - 1 &= 0 \\ 1 - x^3 &= x \\ g(x) &= x \end{aligned}$$

by defining $g(x) = 1 - x^3$.

Choice 2:

$$\begin{aligned} x^3 + x - 1 &= 0 \\ 1 - x &= x^3 \\ \sqrt[3]{1 - x} &= x \\ g(x) &= x \end{aligned}$$

by defining $g(x) = \sqrt[3]{1 - x}$.

Choice 3:

$$\begin{aligned}
x^3 + x - 1 &= 0 \\
2x^3 + 1 &= 3x^3 + x \\
2x^3 + 1 &= x(3x^2 + 1) \\
\frac{2x^3 + 1}{3x^2 + 1} &= x \\
g(x) &= x
\end{aligned}$$

by defining $g(x) = \frac{2x^3+1}{3x^2+1}$.

Theorem:

(a) Existence: If g is continuous on $[a,\, b]$ and $a \leq g(x) \leq b$ for all $x \in [a,\, b]$, then g has a fixed point in $[a,\, b]$.

(b) Uniqueness: If g is differentiable on (a, b) and there exists a positive constant $k < 1$ such that

$$|g'(x)| \leq k < 1, \qquad \text{for all} \quad x \in (a,\, b),$$

then the fixed point in $[a,\, b]$ is unique.

Proof:

(a) Existence:

Let g be continuous on $[a,\, b]$ and $g : [a, b] \to [a, b]$. We define $h(x) = g(x) - x$. Since $g(x)$ and x are continuous, $h(x)$ is continuous in $[a, b]$. By construction, roots of h are the fixed points of g.

Since

$$\min_{x \in [a,b]} g(x) \geq a \qquad \text{and} \qquad \max_{x \in [a,b]} g(x) \leq b$$

we have

$$h(a) = g(a) - a \geq 0 \qquad \text{and} \qquad h(b) \leq 0.$$

If either $h(a) = 0$ or $h(b) = 0$, then we have obtained a root of h which is a fixed point of g and we are done. If neither $h(a) = 0$ nor $h(b) = 0$, then $h(b) < 0 < h(a)$. Since h is continuous on $[a, b]$, by Intermediate Value Theorem, there is a $\xi \in [a, b]$ such that $h(\xi) = 0$ implying $g(\xi) = \xi$.

(b) Uniqueness:

We prove the uniqueness by contradiction. Assume that g has two distinct fixed points $\xi_1$and $\xi_2$on the interval $[a, b]$. Thus we have $g\left(\xi_1\right) = \xi_1$ and $g\left(\xi_2\right) = \xi_2$. Now we can write

$$\begin{aligned} \left|\xi_1 - \xi_2\right| &= \left|g\left(\xi_1\right) - g\left(\xi_2\right)\right| \\ &= \left|\left(\xi_1 - \xi_2\right) g'\left(\xi\right)\right| \qquad \text{using MVT} \\ &= \left|\left(\xi_1 - \xi_2\right)\right| \left|g'\left(\xi\right)\right| \\ &\leq k \left|\left(\xi_1 - \xi_2\right)\right| \\ &< \left|\left(\xi_1 - \xi_2\right)\right| \end{aligned}$$

which is a contradiction. Hence $\xi_1 = \xi_2$, i.e., uniqueness is proved.

2.2.1 Cobweb Diagram (Geometric Interpretation of FPI)

Fixed-point is the point where the graphs $y = g(x)$ and $y = x$ intersect. Various steps of FPI can be obtained by drawing vertical line segments to the function $y = g(x)$ and horizontal line segments to the line $y = x$. We start with our initial guess x_0 on the x-axis and draw a vertical line to $g(x)$, then draw a horizontal line to $y = x$, and continue this process.

Consider $g(x) = 2^{1-x}$. Following figure explains how Cobweb diagram is drawn.

See figure 2.4.

Fixed Point Iteration:

$x_0 =$ initial guess.

$x_{n+1} = g(x_n)$ for $n = 0, 1, 2, ...$

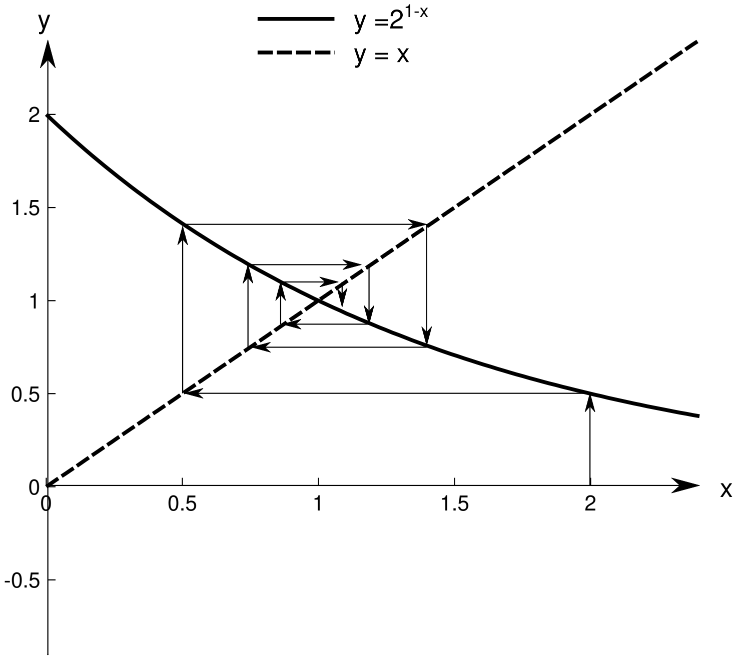
The sequence $\{x_n\}_{n=0}^{\infty}$ may or may not converge as the number of steps goes to ∞. If g is continuous and x_n converges to a number x^*, then we have

$$g\left(x^*\right) = g\left(\lim_{n\to\infty} x_n\right) = \lim_{n\to\infty} g\left(x_n\right) = \lim_{n\to\infty} x_{n+1} = x^*.$$

2.2.2 Stopping Criteria

Let x_n denote the approximation at step n. If the tolerance parameter is ϵ, a small number, we can set the stopping criteria as

Figure 2.4: FPI Cobweb example



$$|x_{n+1} - x_n| \quad < \quad \epsilon$$

2.2.3 Convergence

When does FPI converge?

See figure 2.5 here. We took $x_0 = 0.55$.

Definition:

Linear Convergence

Let e_n represent the error in the n^{th} step of an iterative method. If the following condition

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} \quad = \quad L_c < 1$$

is satisfied, the method is said to have linear convergence with rate L_c.

Local Convergence

An iterative method is locally convergent to x^* if it converges to x^* when initial guess is sufficiently close to x^*.

Theorem:

If g is continuously differentiable, $g(x^*) = x^*$, and $L_c = |g'(x^*)| < 1$, then FPI converges linearly with rate L_c to the fixed point x^* for initial guess sufficiently close to x^*, i.e., it is locally convergent at the rate L_c.

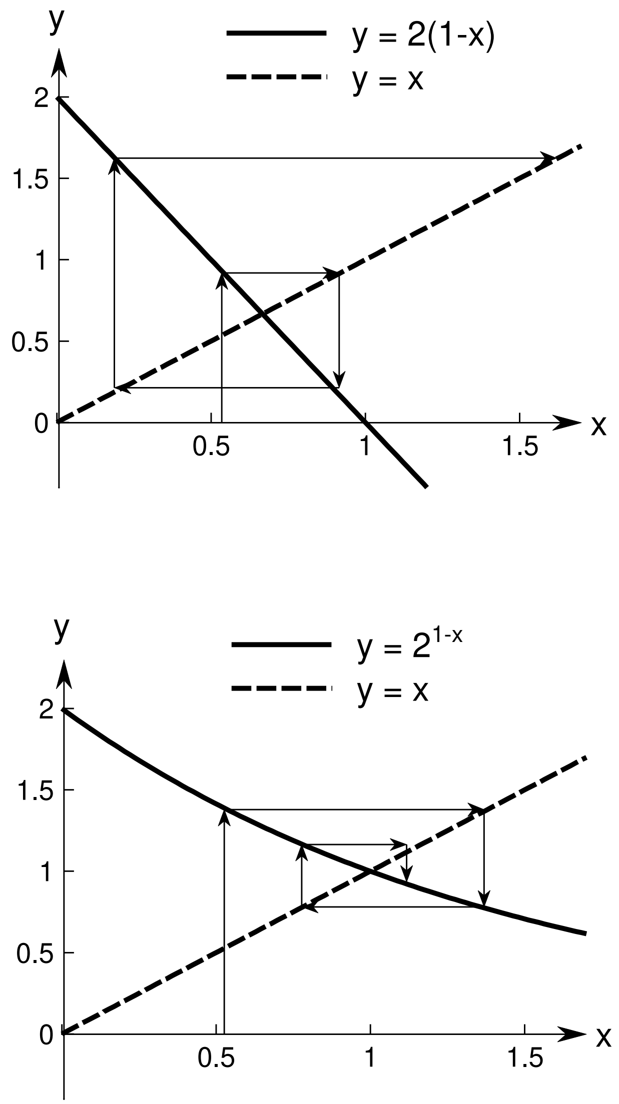
Proof:

Let x_n denote the n^{th} iteration. Using Mean Value Theorem we can find a number ξ_n between x^* and x_n such that

$$g'(\xi_n) \quad = \tfrac{g(x_n)-g(x^*)}{x_n-x^*} \quad = \frac{x_{n+1} - x^*}{x_n - x^*}$$

If we define $e_n = |x_n - x^*|$, we obtain

Figure 2.5: FPI Convergence



$$|g'(\xi_n)| = \frac{e_{n+1}}{e_n}$$

or,

$$e_{n+1} = |g'(\xi_n)|\, e_n$$

Since ξ_n lies between x^* and x_n, it converges to x^* as x_n does. Thus we obtain

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = \lim_{n\to\infty} |g'(\xi_n)| = |g'(x^*)| = L_c < 1.$$

Note:

The Bisection Method converges linearly. FPI is locally convergent, and when it converges, it is a linear convergence. Both of these methods require one function evaluation per step. The uncertainty is cut into half per step by the bisection method, whereas it is approximately $L_c = |g'(x^*)|$ for Fixed-point iteration. Depending on whether L_c is larger or smaller than $\frac{1}{2}$, FPI may converge slower or faster than bisection.

The condition $L_c = |g'(x^*)| < 1$ is sufficient for convergence, not necessary. For example, consider $f(x) = x^3 + x^2 - 3x - 3$, (By Brian Bradie Book), Construct

$$g(x) = x - \frac{x^3 + x^2 - 3x - 3}{3x^2 + 2x - 3} = x - \frac{f(x)}{f'(x)}$$

we have $|g'(x)| = 8$, but it conveges very rapidly.

---

exercise

1. Use the theorem on convergence (page 56) to determine whether the FPI of $g(x)$ is locally convergent to the given fixed point x^*. Also draw the Cobweb diagrams for each.

   (a) $g(x) = 2^{1-x}$, $x^* = 1$, (b) $g(x) = \dfrac{x^4 - 10}{3}$, $x^* = 2$

   (c) $g(x) = \sin x + x$, $x^* = 0$.

2. Consider $f(x) = 2x^2 + x - 1$. Find the roots of $f(x) = 0$ by factoring. Construct two functions for $g(x)$, say, $g_1(x)$ and $g_2(x)$. One by isolating x and the other isolating by x^2 term and then solving. Find the fixed points for these two functions (use four iterations for $g_1(x)$ and six iterations for $g_2(x)$.) Check their local convergence.

3. Using FPI algorithm and initial guess 1, approximate the solution of $e^x + \sin x - 4 = 0$ to eight correct decimal places. Use any programming language/tool. In how many iterations this accuracy is achieved?

2.3 Newton's Method

Newton's Method usually converges much faster than the methods mentioned earlier (which are linearly convergent).

We know that the slope of the tangent at $x = x_0$ to the curve $y = f(x)$ is the derivative $f'(x_0)$. Using the point slope form, the equation of the tangent can be expressed as

$$y - f(x_0) = f'(x_0)(x - x_0)$$

x-intercept of this line is obtained by substituting $y = 0$ as

$$\begin{aligned} f'(x_0)(x - x_0) &= -f(x_0) \\ x - x_0 &= -\frac{f(x_0)}{f'(x_0)} \\ x &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

Assumption above is that $f'(x_0) \neq 0$.

See figure 2.6

This x-intercept is the first approximation x_1. Similar process is used to obtain the second approximation x_2 starting from x_1. Thus we have

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

This process is continued to obtain other iterations.

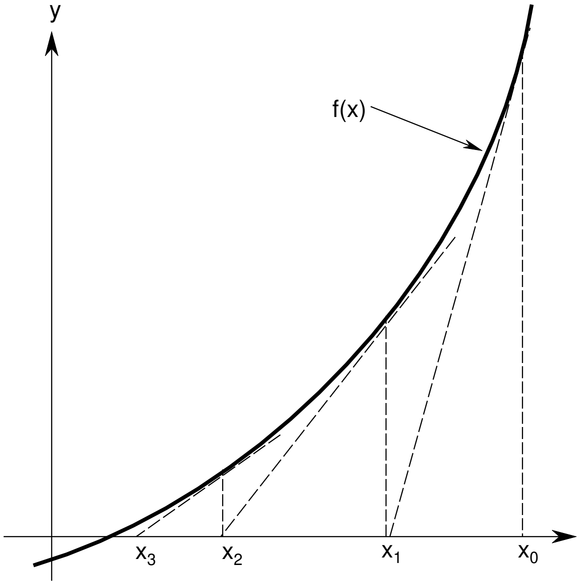
Algorithm for Newton's Method:

$$\begin{aligned} x_0 &= \text{initial guess} \\ x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} \qquad \text{for } n = 0, 1, 2, ... \end{aligned}$$

Example 1

Consider the function $f(x) = x^2 - 6$. Does it has a zero in [1, 3]? Find the first few iterations to approximate a root or zero of $f(x)$ using Newton's Method and $x_0 = 2$.

Figure 2.6: Newton's Method



We start as our initial guess $x_0 = 2$. Here $f'(x) = 2x$.

$$\begin{aligned} x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} = 2 - \frac{f(2)}{f'(2)} \\ &= 2 - \frac{-2}{4} = 2 + 1/2 = 5/2 = 2.5 \end{aligned}$$

$$\begin{aligned} x_2 &= x_1 - \frac{f(x_1)}{f'(x_1)} = 2.5 - \frac{f(2.5)}{f'(2.5)} \\ &= 2.5 - \frac{0.25}{5} = 2.5 - 0.05 = 2.45 \end{aligned}$$

Example 2

Find the first few iterations to approximate a root or zero of $f(x) = 4x^3 - 2x^2 + 3$ using Newton's Method and $x_0 = -1$.

We start as our initial guess $x_0 = -1$. Here $f'(x) = 12x^2 - 4x$.

Please see the extra sheets with a program and its results.

Definition: Quadratic Convergence

Let e_n represent the error in the n^{th} step of an iterative method. If the following condition

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^2} \;=\; C < \infty$$

is satisfied, the method is said to have quadratic convergence where C is a constant.

Theorem:

Let f be twice continuously differentiable, $f\left(x^*\right) = 0$, and $f'\left(x^*\right) \neq 0$, (i.e., f'' is continuous and x^* is a simple zero of f). Then Newton's method is locally and quadratically convergent to x^*. The error e_n in the n^{th} step satisfies

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^2} \;=\; C$$

where

$$C \;=\; \left|\frac{f''\left(x^*\right)}{2f'\left(x^*\right)}\right|.$$

Proof:

Local Convergence: Newton's Method is a particular case of FPI where

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

Differentiating we obtain

$$\begin{aligned} g'(x) &= 1 - \frac{\left(f'(x)\right)^2 - f(x)f''(x)}{\left(f'(x)\right)^2} \\ &= \frac{f(x)f''(x)}{\left(f'(x)\right)^2} \end{aligned}$$

Since $g'\left(x^*\right) = 0$, Newton's Method is locally convergent.

Quadratically convergent: Let x_n and e_n denote the approximate solution and the error in the n^{th} iteration respectively. By Taylor's Theorem, we write

$$f(x^*) = f(x_n) + (x^* - x_n) f'(x_n) + \frac{(x^* - x_n)^2}{2} f''(\xi_n)$$

where ξ_n is a number between x^* and x_n. Since x^* is the root, i.e., $f(x^*) = 0$. Also use of Newton's Method yields

$$\begin{aligned} (x_n - x^*) - \frac{f(x_n)}{f'(x_n)} &= \frac{(x^* - x_n)^2}{2} \frac{f''(\xi_n)}{f'(x_n)} \\ x_n - \frac{f(x_n)}{f'(x_n)} - x^* &= \frac{e_n^2}{2} \frac{f''(\xi_n)}{f'(x_n)} \\ x_{n+1} - x^* &= \frac{e_n^2}{2} \frac{f''(\xi_n)}{f'(x_n)} \end{aligned}$$

i.e.,

$$e_{n+1} = e_n^2 \left| \frac{f''(\xi_n)}{2f'(x_n)} \right|$$

Since ξ_n lies between x^* and x_n, it converges to x^* as x_n does, hence,

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^2} = \left| \frac{f''(x^*)}{2f'(x^*)} \right| = C.$$

Exercise

1. Using Newton's Method and initial guess x_0, find the iterations x_1 and x_2 of the roots for the following equations (without computer programming, but you can use a calculator):

   (a) $x^3 + x^2 - 1 = 0, \quad x_0 = 1$ (b) $x^3 + x - 2 = 0, \quad x_0 = 0$

2. Write the implementation of the algorithm for Newton's Method and using ϵ as stopping criteria.

3. Using Newton's Method and initial guess 1, approximate the root of $e^x + \sin x - 4 = 0$ to eight correct decimal places. Use any programming language/tool. In how many iterations this accuracy is achieved? Do you see any difference in the number of iterations with the problem 3 in FPI exercise?

2.4 Secant Method

The Secant Method is similar to Newton's method. In this method we use the difference quotient instead of the derivative. Geometrically speaking, we use a secant line instead of a tangent line. We know that difference quotient

$$\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

approximates the derivative at x_n. In $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, replacing the derivative by the difference quotient, we have

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \tag{2.1}$$

Secant Method:

$$\begin{aligned} x_0,\, x_1 &= \text{initial guesses} \\ x_{n+1} &= x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \qquad \text{for } n = 1, 2, 3, ... \end{aligned}$$

FPI and Newton's Method needs one initial guess, but Secant Method requires two initial guesses.

Example 1

Find the first few iterations to approximate a root of $f(x) = 4x^3 - 2x^2 + 3$ with $x_0 = -1$ and $x_1 = 0$ using Secant Method.

Here goes the extra sheets with a program and results.

==

Superlinear convergent: By Taylor's Theorem (as $e_n = x_n - x^*$, i.e., $x_n = x^* + e_n$)

$$f(x_n) = f(x^*) + e_n f'(x^*) + \frac{1}{2} e_n^2 f''(x^*) + O(e_n^3)$$

Since $f(x^*) = 0$, we have

$$\frac{f\left(x_n\right)}{e_n} = f'\left(x^*\right)+\frac{1}{2}e_n f''\left(x^*\right)+O\left(e_n^2\right)$$

Changing the index to $n-1$, we obtain

$$\frac{f\left(x_{n-1}\right)}{e_{n-1}} = f'\left(x^*\right)+\frac{1}{2}e_{n-1} f''\left(x^*\right)+O\left(e_{n-1}^2\right)$$

Subtraction of the last two equations yields

$$\frac{f\left(x_n\right)}{e_n}-\frac{f\left(x_{n-1}\right)}{e_{n-1}} \approx \frac{1}{2}\left(e_n-e_{n-1}\right)f''\left(x^*\right)$$

Since $x_n - x_{n-1} = x_n - x^* - \left(x_{n-1}-x^*\right) = e_n - e_{n-1}$, we obtain

$$\begin{aligned}
\frac{\frac{f(x_n)}{e_n}-\frac{f(x_{n-1})}{e_{n-1}}}{x_n-x_{n-1}} &\approx \frac{1}{2}f''\left(x^*\right)\\
\frac{f\left(x_n\right)e_{n-1}-f\left(x_{n-1}\right)e_n}{e_n e_{n-1}\left(x_n-x_{n-1}\right)} &\approx \frac{1}{2}f''\left(x^*\right)\\
\frac{1}{e_n e_{n-1}}\times\frac{f\left(x_n\right)e_{n-1}-f\left(x_{n-1}\right)e_n}{f\left(x_n\right)-f\left(x_{n-1}\right)}\times\frac{f\left(x_n\right)-f\left(x_{n-1}\right)}{x_n-x_{n-1}} &\approx \frac{1}{2}f''\left(x^*\right)\\
\frac{f'(x^*)}{e_n e_{n-1}}\times\frac{f\left(x_n\right)e_{n-1}-f\left(x_{n-1}\right)e_n}{f\left(x_n\right)-f\left(x_{n-1}\right)} &\approx \frac{1}{2}f''\left(x^*\right)
\end{aligned}$$

Now

$$\begin{aligned}
e_{n+1} &= x_{n+1}-x^*\\
&= x_n-\frac{f\left(x_n\right)\left(x_n-x_{n-1}\right)}{f\left(x_n\right)-f\left(x_{n-1}\right)}-x^*\\
&= e_n-\frac{f\left(x_n\right)\left(x_n-x_{n-1}\right)}{f\left(x_n\right)-f\left(x_{n-1}\right)} \quad \text{since } x_n-x^*=e_n\\
&= e_n-\frac{f\left(x_n\right)\left(e_n-e_{n-1}\right)}{f\left(x_n\right)-f\left(x_{n-1}\right)} \quad \text{since } x_n-x_{n-1}=e_n-e_{n-1}\\
&= \frac{f\left(x_n\right)e_{n-1}-f\left(x_{n-1}\right)e_n}{f\left(x_n\right)-f\left(x_{n-1}\right)}
\end{aligned}$$

Using the previous result, we can write

$$\frac{e_{n+1}f'(x^*)}{e_n e_{n-1}} \approx \frac{1}{2}f''(x^*)$$

or,

$$e_{n+1} \approx \frac{1}{2}\frac{f''(x^*)}{f'(x^*)}e_n e_{n-1} = C\, e_n e_{n-1}. \tag{2.2}$$

This is similar to the expression we obtained in the analysis of Newton's method.

Order of Convergence

We consider

$$\lim_{n\to\infty}\frac{e_{n+1}}{e_n^\alpha} = A$$

Asymptotic relation is given by

$$e_{n+1} \approx Ae_n^\alpha$$

where A is a positive constant and $n \to \infty$. Substituting $n-1$ for n, we obtain

$$e_n \approx Ae_{n-1}^\alpha$$

which gives us $e_{n-1} = \left(\frac{e_n}{A}\right)^{1/\alpha}$. Now from $e_{n+1} = C\, e_n e_{n-1}$, we have

$$Ae_n^\alpha \approx Ce_n\left(\frac{e_n}{A}\right)^{1/\alpha}$$

Comparing the powers of e_n on both sides, we obtain $\alpha = 1 + \frac{1}{\alpha}$, i.e., $1 - \alpha + \frac{1}{\alpha} = 0$. Quadratic equation for is α $\alpha^2 - \alpha - 1 = 0$. Thus, $\alpha = \frac{1\pm\sqrt{1+4}}{2}$.

Taking the positive root, we obtain $\alpha = \frac{1+\sqrt{5}}{2}$ which is equivalent to 1.618. Since $1 < \alpha < 2$, secant method's convergence is superlinear which is better than linear. The rapidity of convergence of the secant method is not as good as Newton's method but it is better than the bisection method.

Exercise

1. Using Secant Method and initial guesses $x_0 = 1$ and $x_1 = 2$, find x_2 and x_3 of the roots for the following equations (without computer programming, but use a calculator):

   (a) $x^3 - 2x - 2 = 0$ (b) $e^x + x - 7 = 0$

2. Use Taylor expansions to $f(x+h)$ and $f(x+k)$ to approximate $f'(x)$, i.e.,

$$f'(x) \approx \frac{k^2 f(x+h) + (h^2 - k^2) f(x) - h^2 f(x+k)}{(k-h)hk}$$

3. Write the implementation of the algorithm for Secant Method and using ϵ as stopping criteria.

4. Using Secant Method and initial guesses $x_0 = 1$ and $x_1 = 2$, approximate the root of
   $e^x + \sin x - 4 = 0$ to eight correct decimal places. Use any programming language/tool. In how many iterations this accuracy is achieved?

5. Let $f(x) = x^2 - 6$. Use the Secant method with $x_0 = 3$ and $x_1 = 2$ to find x_3.

Chapter 3

Systems of Equations

3.1 Cramer's Rule

Consider the following n linear equations in n unknowns.

$$\begin{array}{rcl}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... \quad .... + a_{1n}x_n & = & b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... \quad .... + a_{2n}x_n & = & b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + ... \quad .... + a_{3n}x_n & = & b_3 \\
\vdots \qquad\qquad\qquad & \vdots & \vdots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + ... \quad .... + a_{nn}x_n & = & b_n
\end{array} \tag{3.1}$$

Corresponding Matrix form is $A\mathbf{x} = \mathbf{b}$ where

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}$$

Using Cramer's rule we can write the solution of the system as

$$x_i = \frac{|A_i|}{|A|}, \quad i = 1, 2, ..., n$$

where

$$|A| = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,i-1} & a_{1,i} & a_{1,i+1} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,i-1} & a_{2,i} & a_{2,i+1} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & & \ddots & & \vdots & \vdots \\ & & & & & \ddots & & \\ \vdots & \vdots & \vdots & & & & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,i-1} & a_{n,i} & a_{n,i+1} & \cdots & a_{n,n} \end{vmatrix}$$

and

$$|A_i| = \begin{vmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,i-1} & b_1 & a_{1,i+1} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,i-1} & b_2 & a_{2,i+1} & \cdots & a_{2,n} \\ \vdots & \vdots & \vdots & & \ddots & & \vdots & \vdots \\ & & & & & \ddots & & \\ \vdots & \vdots & \vdots & & & & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,i-1} & b_n & a_{n,i+1} & \cdots & a_{n,n} \end{vmatrix}$$

Example 1

Solve the following system using Cramer's rule:

$$\begin{aligned} 2x_1 + 3x_2 &= 4 \\ 3x_1 + 2x_2 &= 1 \end{aligned}$$

Solution is given by $\{-1, 2\}$.

Example 2

$$\begin{aligned} x_1 - x_2 + x_3 &= 3 \\ 2x_1 + x_2 - x_3 &= 0 \\ 3x_1 + 2x_2 + 2x_3 &= 15 \end{aligned}$$

Here

$$\begin{aligned} |A| &= 12 \\ |A_1| &= 12 \\ |A_2| &= 24 \\ |A_3| &= 48 \end{aligned}$$

Thus the solution is $\{1,\, 2,\, 4\}$.

Exercise

1. Compute the determinant of the matrix

$$\begin{bmatrix} 1 & 2 & 0 \\ 2 & 1 & -1 \\ 3 & 1 & 1 \end{bmatrix}$$

2. Find all values of α such that the following matrix is singular

$$\begin{bmatrix} 1 & -1 & \alpha \\ 2 & 2 & 1 \\ 0 & \alpha & -\frac{3}{2} \end{bmatrix}$$

3. Use Cramer's rule to solve

$$\begin{array}{ccc} 2x_1 + 3x_2 - x_3 & = & 4 \\ x_1 - 2x_2 + x_3 & = & 6 \\ x_1 - 12x_2 + 5x_3 & = & 10 \end{array}$$

3.2 Gaussian Elimination

3.2.1 Basic Gaussian Elimination

This method consists of two parts: elimination and back substitution.

Example 1

Let us consider an example first

$$\begin{aligned} x_1 + x_2 + x_3 &= 6 \\ 2x_1 - 2x_2 - x_3 &= -5 \\ 3x_1 + x_2 - x_3 &= 2 \end{aligned}$$

This system in matrix form becomes

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & -2 & -1 \\ 3 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -5 \\ 2 \end{bmatrix}$$

Elimination:

The element a_{11} is called the pivot element and the first row is known as the pivot row in this step. Carrying out $R_2 - 2R_1 \to R_2$ to eliminate a_{21}, we obtain

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -4 & -3 \\ 3 & 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -17 \\ 2 \end{bmatrix}$$

Operation $R_3 - 3R_1 \to R_3$ to eliminate a_{31} yields

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -4 & -3 \\ 0 & -2 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -17 \\ -16 \end{bmatrix}$$

Now we have $a_{22} = -4$ as the pivot element and second row as the pivot row. Applying $R_3 - \frac{R_2}{2} \to R_3$ to eliminate a_{32}, we have

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & -4 & -3 \\ 0 & 0 & -\frac{5}{2} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 6 \\ -17 \\ -\frac{15}{2} \end{bmatrix}$$

This yields

$$\begin{aligned} x_1 + x_2 + x_3 &= 6 \\ -4x_2 - 3x_3 &= -17 \\ -\tfrac{5}{2}x_3 &= -\tfrac{15}{2} \end{aligned}$$

Back Substitution:

Now we solve the new system backward, i.e., solving

$$-\frac{5}{2}x_3 = -\frac{15}{2}$$

we obtain $x_3 = 3$. Solving

$$-4x_2 - 3x_3 = -17$$

we have $x_2 = 2$. Substituting x_2 and x_3 in

$$x_1 + x_2 + x_3 = 6$$

we get $x_1 = 1$. Hence the solution of the original system is $\{1,\ 2,\ 3\}$.

Example 2

Using Gaussian elimination, solve

$$\begin{aligned} 6x_1 + 4x_2 + 2x_3 &= 12 \\ 3x_1 - 2x_2 - x_3 &= 0 \\ 3x_1 + 4x_2 + x_3 &= 8 \end{aligned}$$

i.e.,

$$A\mathbf{x} = \mathbf{b} \qquad \Longrightarrow \begin{bmatrix} 6 & 4 & 2 \\ 3 & -2 & -1 \\ 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 0 \\ 8 \end{bmatrix}$$

Elimination part

One way to combine the coefficient matrix A and the right hand vector $\mathbf{b}$ into an augmented matrix as

$$\begin{bmatrix} 6 & 4 & 2 & | & 12 \\ 3 & -2 & -1 & | & 0 \\ 3 & 4 & 1 & | & 8 \end{bmatrix}$$

We operate on this matrix as follows

$$\begin{aligned}
\begin{bmatrix} 6 & 4 & 2 & | & 12 \\ 3 & -2 & -1 & | & 0 \\ 3 & 4 & 1 & | & 8 \end{bmatrix} & \sim \begin{bmatrix} 6 & 4 & 2 & | & 12 \\ 0 & -4 & -2 & | & -6 \\ 3 & 4 & 1 & | & 8 \end{bmatrix} & \qquad R_2 - \frac{1}{2}R_1 \to R_2 \\
& \sim \begin{bmatrix} 6 & 4 & 2 & | & 12 \\ 0 & -4 & -2 & | & -6 \\ 0 & 2 & 0 & | & 2 \end{bmatrix} & \qquad R_3 - \frac{1}{2}R_1 \to R_3 \\
& \sim \begin{bmatrix} 6 & 4 & 2 & | & 12 \\ 0 & -4 & -2 & | & -6 \\ 0 & 0 & -1 & | & -1 \end{bmatrix} & \qquad R_3 + \frac{1}{2}R_2 \to R_3
\end{aligned}$$

which is equivalent to

$$\begin{aligned}
6x_1 + 4x_2 + 2x_3 &= 12 \\
-4x_2 - 2x_3 &= -6 \\
-x_3 &= -1
\end{aligned}$$

Backward substitution part

Backward substitution yields

$$\begin{aligned}
x_3 &= 1 \\
x_2 &= (6 - 2 \times 1)/4 = 1 \\
x_1 &= (12 - 4 \times 1 - 2 \times 1)/6 = 1
\end{aligned}$$

Thus we obtain $x_1 = 1$, $x_2 = 1$ and $x_3 = 1$.

Let us consider the following system of linear equations

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... \quad .... + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + ... \quad .... + a_{2n}x_n & = & b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + ... \quad .... + a_{3n}x_n & = & b_3 \\ \vdots \qquad\qquad\qquad \vdots & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + ... \quad .... + a_{nn}x_n & = & b_n \end{array} \tag{3.2}$$

Matrix form is given by

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{3.3}$$

Here A is the coefficient matrix.

To eliminate a_{21} we perform $R_2 - \frac{a_{21}}{a_{11}}R_1 \to R_2$. Here we assume that $a_{11} \neq 0$. Thus we have

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} - \frac{a_{21}}{a_{11}}a_{12} & a_{23} - \frac{a_{21}}{a_{11}}a_{13} & \cdots & a_{2n} - \frac{a_{21}}{a_{11}}a_{1n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 - \frac{a_{21}}{a_{11}}b_1 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

In general, to eliminate a_{i1} we perform $R_i - \frac{a_{i1}}{a_{11}}R_1 \to R_i$ which yields

$$\begin{array}{cccccc|c} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & & b_1 \\ \vdots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & a_{i2} - \frac{a_{i1}}{a_{11}}a_{12} & a_{i3} - \frac{a_{i1}}{a_{11}}a_{13} & \cdots & a_{in} - \frac{a_{i1}}{a_{11}}a_{1n} & & b_i - \frac{a_{i1}}{a_{11}}b_1 \end{array}$$

After completion of the elimination, reduce matrix form becomes

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

which can be expressed as

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + ... \quad .... + a_{1n}x_n &= b_1 \\ a_{22}x_2 + a_{23}x_3 + ... \quad .... + a_{2n}x_n &= b_2 \\ \vdots \quad & \\ a_{n-1\,n-1}x_{n-1+}a_{n-1\,n}x_n &= b_{n-1} \\ a_{nn}x_n &= b_n \end{aligned} \tag{3.4}$$

Back substitution yields

$$\begin{aligned} x_n &= \tfrac{b_n}{a_{nn}} \\ x_{n-1} &= \tfrac{b_{n-1} - a_{n-1\,n}x_n}{a_{n-1\,n-1}} \\ \vdots & \\ x_2 &= \tfrac{b_2 - a_{23}x_3 - .... - a_{2n}x_n}{a_{22}} \\ x_1 &= \tfrac{b_1 - a_{12}x_2 - .... - a_{1n}x_n}{a_{11}} \end{aligned} \tag{3.5}$$

In general,

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{ij}x_j}{a_{ii}}$$

Algorithm for elimination

```
for $k = 1$ to $n - 1$
    for $i = k + 1$ to $n$
        mult $= \frac{a_{ik}}{a_{kk}}$
        for $j = k + 1$ to $n$
            $a_{ij} = a_{ij} - \text{mult} * a_{kj}$
            end $j$ loop
            $b_i = b_i - \text{mult} * b_k$
        end $i$ loop
    end $k$ loop
```

Algorithm for back substitution

```
    $x_n = b_n/a_{n,n}$
     for $i = n-1$ to $1:\ -1$
        for $j = i{+}1$ to $n$
            $b_i = b_i - a_{ij} * x_j$
        end $j$ loop
        $x_i = b_i/a_{ii}$
    end $i$ loop
```

Results

$$\begin{aligned}
1+2+3+\ldots\ldots\ldots\ldots\ldots+n &= \frac{n(n+1)}{2} \\
1^2+2^2+3^2+\ldots\ldots\ldots\ldots\ldots+n^2 &= \frac{n(n+1)(2n+1)}{6} \\
1^3+2^3+3^3+\ldots\ldots\ldots\ldots\ldots+n^3 &= \left\{\frac{n(n+1)}{2}\right\}^2
\end{aligned}$$

3.2.2 Operation Counts

Here we present the operation counts for elimination and back substitution parts. In the elimination step, we put zeros in the lower triangle.

In Elimination Step : Total operation count:

$$\begin{aligned}
&\sum_{k=1}^{n-1}\left[\sum_{i=k+1}^{n}\left\{1+\left(\sum_{j=k+1}^{n}2\right)+2\right\}\right]\\
&\quad=\sum_{k=1}^{n-1}\left[\sum_{i=k+1}^{n}\left\{3+2\left(n-k-1+1\right)\right\}\right]\\
&\quad=\sum_{k=1}^{n-1}\left[\sum_{i=k+1}^{n}\left\{2n+3-2k\right\}\right]\\
&\quad=\sum_{k=1}^{n-1}\left[\left(2n+3-2k\right)\left(n-k-1+1\right)\right]\\
&\quad=\sum_{k=1}^{n-1}\left[2(n-k)^2+3(n-k)\right]\\
&\quad=\sum_{k=1}^{n-1}\left[2n^2-4nk+2k^2+3n-3k\right]\\
&\quad=\sum_{k=1}^{n-1}\left[2n^2+3n\right]+2\sum_{k=1}^{n-1}k^2-(4n+3)\sum_{k=1}^{n-1}k\\
&\quad=\left(2n^2+3n\right)(n-1)+2\frac{(n-1)n(2n-1)}{6}-(4n+3)\frac{(n-1)n}{2}\\
&\quad=2n^3+n^2-3n+\frac{(n^2-n)(2n-1)}{3}-\frac{(4n+3)(n^2-n)}{2}\\
&\quad=2n^3+n^2-3n+\frac{2}{3}n^3-n^2+\frac{n}{3}-2n^3+\frac{n^2}{2}+\frac{3n}{2}\\
&\quad=\frac{2n^3}{3}+\frac{n^2}{2}-\frac{7n}{6}
\end{aligned}$$

Table 3.1: Number of Operations

n	Operations in Elimination	Operations in Back Subs	Total Operations
5	90	25	115
10	705	100	805
50	84525	2500	87025
100	671550	10000	681550

In Back Substitution Step: Total operation count:

$$
\begin{aligned}
1+\sum_{i=1}^{n-1}&\left[\left(\sum_{j=i+1}^{n}2\right)+1\right]\\
&=1+\sum_{i=1}^{n-1}\left[2\left(n-i-1+1\right)+1\right]\\
&=1+\sum_{i=1}^{n-1}\left[2n+1-2i\right]\\
&=1+(2n+1)(n-1)-2\sum_{i=1}^{n-1}i\\
&=2n^2-n-2\frac{(n-1)(n)}{2}\\
&=2n^2-n-n^2+n\\
&=n^2
\end{aligned}
$$

This total operation count (elimination + back sub) is $\frac{2n^3}{3}+\frac{3n^2}{2}-\frac{7n}{6}$
$O(n^3)$ dominates: If $n=100$,then $n^3=1000000$,but $n^2=10000$.

3.2.3 Pivoting (Partial and Complete)

What happens if $a_{ii} = 0$? We will discuss this later.

Exercise

1. Using Gaussian Elimination solve the following systems:

   (a) $\begin{aligned} 4x_1 + x_2 + 2x_3 &= 9 \\ 2x_1 + 4x_2 - x_3 &= -5 \\ x_1 + x_2 - 3x_3 &= -9 \end{aligned}$ (b) $\begin{aligned} 2x - 2y - z &= -2 \\ 4x + y - 2z &= 1 \\ -2x + y - z &= -3 \end{aligned}$

2. Using back substitution solve the following systems:

   (a) $\begin{aligned} 3x_1 - 4x_2 + 5x_3 &= 2 \\ 3x_2 - 4x_3 &= -1 \\ 5x_3 &= 5 \end{aligned}$ (b) $\begin{aligned} x - 2y + z &= -2 \\ 4y - 3z &= 1 \\ -3z &= 3 \end{aligned}$

3. Write the implementation of the Gaussian Elimination algorithm for elimination and back substitution in your favorite programming language.

4. If the approximate operation count by using Gaussian elimination to solve n equations in n unknowns is $2n^3/3$, how much longer it takes to estimate if n is tripled?

3.3 LU Factorization

Consider matrix equation

$$A\mathbf{x} = \mathbf{b} \tag{3.6}$$

where the matrices A, $\mathbf{x}$ and $\mathbf{b}$ are respectively given by

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}$$

If A is diagonal , then it is easy to solve the system. In this case we have

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{3.7}$$

so that obtain $a_{ii}x_i = b_i$ for $i = 1, ..., n$. Thus solution of the system (3.7) is given by

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1/a_{11} \\ b_2/a_{22} \\ b_3/a_{33} \\ \vdots \\ b_n/a_{33} \end{bmatrix} \tag{3.8}$$

If $a_{ii} = 0$ for some i, what happens?

If $a_{ii} = 0$ and $b_i = 0$ for some i, then x_i can be any number. If $a_{ii} = 0$ and $b_i \neq 0$ for some i, the system does not have any solution.

If A is upper triangular, we use back substitution to solve the system as we did in the Gaussian elimination. Thus

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} \\ 0 & 0 & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{3.9}$$

yields the following algorithm

```
    for $i = n$ to $1 : \ -1$
        $x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij}x_j\right) / a_{ii}$
    end $i$
```

If A is lower triangular, we use forward substitution to solve the system. Thus

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ a_{21} & a_{22} & 0 & \cdots & 0 \\ a_{31} & a_{32} & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix} \tag{3.10}$$

yields

$$\begin{aligned} a_{11}x_1 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2 \\ \vdots \\ a_{n-1\,1}x_1 + a_{n-1\,2}x_2 + ... \quad .... + a_{n-1\,n-1}x_{n-1} &= b_{n-1} \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + ... \quad .... + a_{n-1\,n-1}x_{n-1} + a_{nn}x_n &= b_n \end{aligned}$$

Using forward substitution, we obtain

$$\begin{aligned} & x_1 \quad = \tfrac{b_1}{a_{11}} \\ & x_2 = \tfrac{b_2 - a_{21}x_1}{a_{22}} \\ & \vdots \\ & x_{n-1} \quad = \tfrac{b_{n-1} - a_{n-1\,1}x_1 - a_{n-1\,2}x_2 - .... - a_{n-1\,n-2}x_{n-2}}{a_{n-1\,n-1}} \\ & x_n \quad = \tfrac{b_1 - a_{n1}x_1 - a_{n2}x_2 - a_{n3}x_3 - .... - a_{n-1\,n-1}x_{n-1}}{a_{nn}} \end{aligned} \tag{3.11}$$

which yields the following algorithm

```
    for $i = 1$ to $n$
        $x_i = \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j\right)/a_{ii}$
    end do $i$ loop
```

In LU factorization, we decompose A into two matrices, one is lower triangular L and the other one is upper triangular U, i.e., we write

$$A = LU$$

where L is lower triangular and U is upper triangular.

Example 1

Consider

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Here we write

$$\begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Also

$$\begin{bmatrix} 1 & 0 \\ 3 & -2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Example 2

Consider

$$A = \begin{bmatrix} 6 & 4 & 2 \\ 3 & -2 & -1 \\ 3 & 4 & 1 \end{bmatrix}$$

Here we can express

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix} \begin{bmatrix} 6 & 4 & 2 \\ 0 & -4 & -2 \\ 0 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 6 & 4 & 2 \\ 3 & -2 & -1 \\ 3 & 4 & 1 \end{bmatrix}$$

To solve

$$A\mathbf{x} = \mathbf{b} \quad \text{for} \quad \mathbf{x} \tag{3.12}$$

we solve the following two matrix equations

$$L\mathbf{y} = \mathbf{b} \quad \text{for} \quad \mathbf{y} \tag{3.13}$$

$$U\mathbf{x} = \mathbf{y} \quad \text{for} \quad \mathbf{x} \tag{3.14}$$

where

$$A = LU \tag{3.15}$$

Example 3

Using LU factorization, solve

$$\begin{aligned} 6x_1 + 4x_2 + 2x_3 &= 12 \\ 3x_1 - 2x_2 - x_3 &= 0 \\ 3x_1 + 4x_2 + x_3 &= 8 \end{aligned}$$

i.e.,

$$A\mathbf{x} = \mathbf{b} \qquad \Longrightarrow \begin{bmatrix} 6 & 4 & 2 \\ 3 & -2 & -1 \\ 3 & 4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 0 \\ 8 \end{bmatrix}$$

First we need to find L and U such that $LU = A$.

$$\begin{aligned}
\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & 4 & 2 \\ 3 & -2 & -1 \\ 3 & 4 & 1 \end{bmatrix} &\sim \begin{bmatrix} 1 & 0 & 0 \\ \left(\frac{1}{2}\right) & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & 4 & 2 \\ 0 & -4 & -2 \\ 3 & 4 & 1 \end{bmatrix} \qquad R_2 - \frac{1}{2}R_1 \to R_2 \\
&\sim \begin{bmatrix} 1 & 0 & 0 \\ \left(\frac{1}{2}\right) & 1 & 0 \\ \left(\frac{1}{2}\right) & 0 & 1 \end{bmatrix}\begin{bmatrix} 6 & 4 & 2 \\ 0 & -4 & -2 \\ 0 & 2 & 0 \end{bmatrix} \qquad R_3 - \frac{1}{2}R_1 \to R_3 \\
&\sim \begin{bmatrix} 1 & 0 & 0 \\ \left(\frac{1}{2}\right) & 1 & 0 \\ \left(\frac{1}{2}\right) & \left(-\frac{1}{2}\right) & 1 \end{bmatrix}\begin{bmatrix} 6 & 4 & 2 \\ 0 & -4 & -2 \\ 0 & 0 & -1 \end{bmatrix} \qquad R_3 + \frac{1}{2}R_2 \to R_3
\end{aligned}$$

Thus we first solve

$$L\mathbf{y} = \mathbf{b}$$

i.e.,

$$\begin{bmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 12 \\ 0 \\ 8 \end{bmatrix}$$

which yields

$$\begin{aligned}
y_1 &= 12 \\
y_2 &= 0 - \frac{1}{2} \times 12 = -6 \\
y_3 &= 8 - \frac{1}{2} \times 12 + \frac{1}{2} \times (-6) = -1
\end{aligned}$$

Then we solve

$$U\mathbf{x} = \mathbf{y}$$

i.e.,

$$\begin{bmatrix} 6 & 4 & 2 \\ 0 & -4 & -2 \\ 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 12 \\ -6 \\ -1 \end{bmatrix}$$

yielding

$$\begin{aligned}
x_3 &= 1 \\
x_2 &= \left(6 - 2 \times 1\right)/4 = 1 \\
x_1 &= \left(12 - 4 \times 1 - 2 \times 1\right)/6 = 1
\end{aligned}$$

Thus we obtain $x_1 = 1$, $x_2 = 1$ and $x_3 = 1$.

Example 4

Find the LU decomposition of the following matrix:

$$A = \begin{bmatrix} 4 & 0 & 1 & 1 \\ 3 & 1 & 3 & 1 \\ 0 & 1 & 2 & 0 \\ 3 & 2 & 4 & 1 \end{bmatrix}$$

Here we store the multipliers by enclosing in parentheses. These entries in reduced matrix form the lower triangular matrix with diagonal entries all 1. Entries in the upper triangle form the upper triangular matrix.

$$\begin{bmatrix} 4 & 0 & 1 & 1 \\ 3 & 1 & 3 & 1 \\ 0 & 1 & 2 & 0 \\ 3 & 2 & 4 & 1 \end{bmatrix} \sim \begin{bmatrix} 4 & 0 & 1 & 1 \\ \left(\frac{3}{4}\right) & 1 & \frac{9}{4} & \frac{1}{4} \\ 0 & 1 & 2 & 0 \\ 3 & 2 & 4 & 1 \end{bmatrix} \qquad R_2 - \frac{3}{4}R_1 \to R_2$$

$$\sim \begin{bmatrix} 4 & 0 & 1 & 1 \\ \left(\frac{3}{4}\right) & 1 & \frac{9}{4} & \frac{1}{4} \\ (0) & 1 & 2 & 0 \\ 3 & 2 & 4 & 1 \end{bmatrix} \qquad \text{already zero entry}$$

$$\sim \begin{bmatrix} 4 & 0 & 1 & 1 \\ \left(\frac{3}{4}\right) & 1 & \frac{9}{4} & \frac{1}{4} \\ (0) & 1 & 2 & 0 \\ \left(\frac{3}{4}\right) & 2 & \frac{13}{4} & \frac{1}{4} \end{bmatrix} \qquad R_4 - \frac{3}{4}R_1 \to R_4$$

$$\sim \begin{bmatrix} 4 & 0 & 1 & 1 \\ \left(\frac{3}{4}\right) & 1 & \frac{9}{4} & \frac{1}{4} \\ (0) & (1) & -\frac{1}{4} & -\frac{1}{4} \\ \left(\frac{3}{4}\right) & 2 & \frac{13}{4} & \frac{1}{4} \end{bmatrix} \qquad R_3 - R_2 \to R_3$$

$$\sim \begin{bmatrix} 4 & 0 & 1 & 1 \\ \left(\frac{3}{4}\right) & 1 & \frac{9}{4} & \frac{1}{4} \\ (0) & (1) & -\frac{1}{4} & -\frac{1}{4} \\ \left(\frac{3}{4}\right) & (2) & -\frac{5}{4} & -\frac{1}{4} \end{bmatrix} \qquad R_4 - 2R_2 \to R_4$$

$$\sim \begin{bmatrix} 4 & 0 & 1 & 1 \\ \left(\frac{3}{4}\right) & 1 & \frac{9}{4} & \frac{1}{4} \\ (0) & (1) & -\frac{1}{4} & -\frac{1}{4} \\ \left(\frac{3}{4}\right) & (2) & (5) & 1 \end{bmatrix} \qquad R_4 - 5R_3 \to R_4$$

Thus we have

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \frac{3}{4} & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ \frac{3}{4} & 2 & 5 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 4 & 0 & 1 & 1 \\ 0 & 1 & \frac{9}{4} & \frac{1}{4} \\ 0 & 0 & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exercise

1. Solve the following linear systems:

$$(a) \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & 1 \\ 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ -1 \\ 1 \end{bmatrix}$$

$$(b) \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & -3 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 8 \end{bmatrix}$$

2. Find LU factorization of the given matrices. Check your results by matrix multiplication.

$$(a) \begin{bmatrix} 3 & 1 & 2 \\ 6 & 3 & 4 \\ 3 & 1 & 5 \end{bmatrix} \qquad (b) \begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix}$$

3. Using LU factorization to solve the following system

$$\begin{aligned} x_1 + 2x_2 + 3x_3 &= 1 \\ 4x_1 + 5x_2 + 6x_3 &= 2 \\ 7x_1 + 8x_2 + 9x_3 &= 3 \end{aligned}$$

3.4 Pivoting

Partial, Complete and Scaled Pivoting

If a matrix is diagonally dominant, then Gaussian elimination without pivoting works. What happens if it is not diagonally dominant? What happens if $a_{ii} = 0$ in Gaussian Elimination or LU Factorization. We will discuss this now.

Diagonals are called pivots or pivot elements. The process of interchanging rows to avoid a zero diagonal element is called pivoting. **Partial pivoting** consists of comparing the pivot element with only entries in the pivot column below the current diagonal. In **Complete pivoting**, we examine not only on the current column but also on all subsequent columns. Complete pivoting is more stable but more expensive to implement. **Scaled pivoting** consists of scaling the rows before pivoting. Here we present the implementation of partial pivoting.

3.4.1 Partial Pivoting in Gaussian Elimination

In partial pivoting before we start elimination, we compare the element in the first column and select the mth row such that

$$|a_{m1}| \quad \geq |a_{i1}| \qquad \text{for all } 1 \leq i \leq n$$

and then exchange the first row and the mth row. For the second pivot, we start with the current a_{22} and examine all elements directly below this column and select the mth row such that

$$|a_{m2}| \quad \geq |a_{i2}| \qquad \text{for all } 2 \leq i \leq n$$

and if $m \neq 2$, then exchange the second row and the mth row. If $|a_{22}|$ is largest, no exchange is made. First row is not effected by this process. For the kth pivot element we examine the $n - k + 1$ entries in the subcolumn below and including the diagonal entry and the row with largest entry in that column is selected as the pivot row. This procedure is continued for each column during elimination.

Example 1

Apply Gaussian Elimination with partial pivoting to solve the following system

$$\begin{aligned} x_1 + x_2 &= 3 \\ 3x_1 - 4x_2 &= 2 \end{aligned}$$

Here the augumented matrix is given by

$$\begin{bmatrix} 1 & 1 & | & 3 \\ 3 & -4 & | & 2 \end{bmatrix}$$

Here $a_{11} = 1$ and $a_{21} = 3$ so that we have $|a_{21}| > |a_{11}|$. Exchanging the first and second rows, we obtain

$$\begin{bmatrix} 3 & -4 & | & 2 \\ 1 & 1 & | & 3 \end{bmatrix}$$

Now $R_2 - \frac{1}{3}R_1 \to R_2$ yields

$$\begin{bmatrix} 3 & -4 & | & 2 \\ 0 & \frac{7}{3} & | & \frac{7}{3} \end{bmatrix}$$

which can be expressed as

$$\begin{aligned} 3x_1 - 4x_2 &= 2 \\ \frac{7}{3}x_2 &= \frac{7}{3} \end{aligned}$$

Now using back substitution, we obtain

$$\begin{aligned} x_2 &= 1 \\ x_1 &= \frac{1}{3}(2 + 4 \times 1) = 2 \end{aligned}$$

Thus the solution is $\{1,\ 2\}$.

Example 2

Apply Gaussian Elimination with partial pivoting to solve the following system

$$A\mathbf{x} \;=\; \mathbf{b}$$

where

$$A = \begin{bmatrix} 1 & -1 & 3 \\ -1 & 0 & -2 \\ 2 & 2 & 4 \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} -3 \\ 1 \\ 0 \end{bmatrix}$$

Here the augumented matrix is can be transformed as

$$\begin{bmatrix} 1 & -1 & 3 & | & -3 \\ -1 & 0 & -2 & | & 1 \\ 2 & 2 & 4 & | & 0 \end{bmatrix} \sim$$

$$\begin{aligned}
\begin{bmatrix} 1 & -1 & 3 & | & -3 \\ -1 & 0 & -2 & | & 1 \\ 2 & 2 & 4 & | & 0 \end{bmatrix} &\sim \begin{bmatrix} 2 & 2 & 4 & | & 0 \\ -1 & 0 & -2 & | & 1 \\ 1 & -1 & 3 & | & -3 \end{bmatrix} & R_1 \leftrightarrow R_3 \\
&\sim \begin{bmatrix} 2 & 2 & 4 & | & 0 \\ 0 & 1 & 0 & | & 1 \\ 1 & -1 & 3 & | & -3 \end{bmatrix} & R_2 + \frac{1}{2}R_1 \to R_2 \\
&\sim \begin{bmatrix} 2 & 2 & 4 & | & 0 \\ 0 & 1 & 0 & | & 1 \\ 0 & -2 & 1 & | & -3 \end{bmatrix} & R_3 - \frac{1}{2}R_1 \to R_3 \\
&\sim \begin{bmatrix} 2 & 2 & 4 & | & 0 \\ 0 & -2 & 1 & | & -3 \\ 0 & 1 & 0 & | & 1 \end{bmatrix} & R_2 \leftrightarrow R_3 \\
&\sim \begin{bmatrix} 2 & 2 & 4 & | & 0 \\ 0 & -2 & 1 & | & -3 \\ 0 & 0 & \frac{1}{2} & | & -\frac{1}{2} \end{bmatrix} & R_3 + \frac{1}{2}R_2 \to R_3
\end{aligned}$$

which is equivalent to

$$\begin{array}{lcl} 2x_1 + 2x_2 + 4x_3 & = & 0 \\ \quad -2x_2 + x_3 & = & -3 \\ \qquad \frac{1}{2}x_3 & = & -\frac{1}{2} \end{array}$$

Back substitution yields the solution as $\{1,\ 1,\ -1\}$.

3.4.2 Partial Pivoting in $PA = LU$ Factorization

Let us first define permutation matrix. A **permutation matrix** is an $n \times n$ matrix consists of a single 1 in each row and column and all other entries are zeros. Identity matrix is a permutation matrix. Following are few examples of permutation matrices

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Example 3

Multiplying a matrix A by permutation matrix P on the left side, we swap those associated rows of A.

(a) Consider

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & 3 & -1 \\ -4 & 2 & -5 \\ 3 & 6 & 2 \end{bmatrix}$$

Here we have

$$PA \ = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ -4 & 2 & -5 \\ 3 & 6 & 2 \end{bmatrix} = \begin{bmatrix} -4 & 2 & -5 \\ 2 & 3 & -1 \\ 3 & 6 & 2 \end{bmatrix}$$

(b) Consider

$$P = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \qquad A = \begin{bmatrix} 2 & 3 & -1 \\ -4 & 2 & -5 \\ 3 & 6 & 2 \end{bmatrix}$$

Here we have

$$PA = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 & -1 \\ -4 & 2 & -5 \\ 3 & 6 & 2 \end{bmatrix} = \begin{bmatrix} -4 & 2 & -5 \\ 3 & 6 & 2 \\ 2 & 3 & -1 \end{bmatrix}$$

To solve

$$A\mathbf{x} = \mathbf{b} \tag{3.16}$$

we multiply the above equation by a permutation matrix P both sides on left and using $PA = LU$, we can write

$$PA\mathbf{x} = P\mathbf{b} \tag{3.17}$$

i.e.,

$$LU\mathbf{x} = P\mathbf{b} \tag{3.18}$$

Thus we have

$$L\mathbf{y} = P\mathbf{b} \quad \text{for} \quad \mathbf{y} \tag{3.19}$$

$$U\mathbf{x} = \mathbf{y} \quad \text{for} \quad \mathbf{x} \tag{3.20}$$

Example 4

Apply $PA = LU$ factorization with partial pivoting to solve the following system

$$\begin{aligned} 2x_1 + 3x_2 &= 4 \\ 3x_1 + 2x_2 &= 1 \end{aligned}$$

i.e.,

$$A\mathbf{x} = \mathbf{b} \quad \Longrightarrow \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \end{bmatrix}$$

We want to find L and U such that $PA = LU$. Here our coefficient matrix is

$$A = \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix}$$

we choose P to be $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ and hence we obtain PA as $\begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$. Now we have

$$\begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix} \sim \begin{bmatrix} 3 & 2 \\ \left(\frac{2}{3}\right) & \frac{5}{3} \end{bmatrix} \qquad R_2 - \frac{2}{3}R_1 \to R_2$$

This yields $PA = LU$ as

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 \\ \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 0 & \frac{5}{3} \end{bmatrix}$$

Now we first solve

$$L\mathbf{y} = P\mathbf{b}$$

i.e.,

$$\begin{bmatrix} 1 & 0 \\ \frac{2}{3} & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

to obtain $y_1 = 1$ and $y_2 = \frac{10}{3}$. Next we solve

$$U\mathbf{x} = \mathbf{y}$$

i.e.,

$$\begin{bmatrix} 3 & 2 \\ 0 & \frac{5}{3} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{10}{3} \end{bmatrix}$$

giving us

$$\begin{aligned} 3x_1 + 2x_2 &= 1 \\ \frac{5}{3}x_2 &= \frac{10}{3} \end{aligned}$$

which yields the solution as

$$\begin{aligned} x_2 &= 2 \\ x_1 &= \frac{1}{3}(1 - 2 \times 2) = -1. \end{aligned}$$

Exercise

1. Using Gaussian Elimination with partial pivoting solve the following system

$$\begin{aligned} x_1 + 2x_2 + 2x_3 &= 1 \\ 4x_1 + 4x_2 + 12x_3 &= 12 \\ 4x_1 + 8x_2 + 12x_3 &= 8 \end{aligned}$$

2. Using partial pivoting find the $PA = LU$ factorization of the following matrices

$$\text{(a)} \quad \begin{bmatrix} 2 & 4 \\ 1 & 3 \end{bmatrix} \qquad \text{(b)} \quad \begin{bmatrix} 1 & 5 \\ 5 & 12 \end{bmatrix}$$

3. Using partial pivoting find the $PA = LU$ factorization of the following matrices

$$\text{(a)} \quad \begin{bmatrix} 1 & 2 & -1 \\ 1 & 2 & 3 \\ 2 & -1 & 4 \end{bmatrix} \qquad \text{(b)} \quad \begin{bmatrix} 0 & 1 & 3 \\ 2 & 1 & 1 \\ -1 & -1 & 2 \end{bmatrix}$$

3.5 Error Estimate and Condition Number

Consider the linear system $\mathbf{Ax} = \mathbf{b}$. If $\tilde{\mathbf{x}}$ denotes an approximate solution of that system, then $\mathbf{r} = \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b}$ is called the **residual**.

The error is given by

$$\mathbf{e} \;=\; \tilde{\mathbf{x}} - \mathbf{x}.$$

When $\mathbf{r} = \mathbf{0}$, then we have exact solution $\mathbf{x}$.,i.e., $\mathbf{e} = \mathbf{0}$. It seems if $\mathbf{r}$ is small, $\mathbf{e}$ is small. But this is not the case.

Example:

Consider the linear system:

$$\begin{bmatrix} -1.01 & 2.01 \\ 1 & -2 \end{bmatrix} \mathbf{x} \;=\; \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Here, $\mathbf{x} = [1,\ 1]^T$ is its exact solution. The vector $\tilde{\mathbf{x}} = [-1,\ 0]^T$ is a poor approximation to $\mathbf{x}$.

$$\mathbf{e} = \begin{bmatrix} -2 \\ -1 \end{bmatrix} \quad \Rightarrow \quad \|\mathbf{e}\|_\infty = 2$$

Now,

$$\mathbf{r} \;=\; \mathbf{A}\tilde{\mathbf{x}} - \mathbf{b} = \begin{bmatrix} -1.01 & 2.01 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \end{bmatrix} - \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}$$

Hence, $\|\mathbf{r}\| = 0.01$.Error is 200 times larger that the residual. Norm of the coefiicient matrix and its inverse (condition number) play an important role on the reliability of the resudual as a presiction of error.

Also, we have

$$\mathbf{A}^{-1} \;=\; \frac{1}{|\mathbf{A}|} \begin{bmatrix} -1.01 & 2.01 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} -200 & -201 \\ 100 & 101 \end{bmatrix}$$

$$\|\mathbf{A}\|_\infty = |1.01| + |2.01| = 3.02 \qquad \|\mathbf{A}\|_\infty = 401$$

so that $\kappa(\mathbf{A}) = 401 \times 3.02 = 1211.02$.

3.6 Error Estimatess-2

Backward and Forward Errors

Consider the linear system $\mathbf{Ax} = \mathbf{b}$. If $\mathbf{x}^*$ denotes an approximate solution of that system, then $\mathbf{r} = \mathbf{b} - \mathbf{Ax}^*$ is called the **residual**. The backward and forward errors are defined as

$$\begin{aligned} \text{Backward error} &= \|\mathbf{r}\|_\infty = \|\mathbf{b} - A\mathbf{x}^*\|_\infty \\ \text{Forward error} &= \|\mathbf{x} - \mathbf{x}^*\|_\infty \end{aligned}$$

and their relative errors are

$$\begin{aligned} \text{Relative backward error} &= \frac{\|\mathbf{r}\|_\infty}{\|\mathbf{b}\|_\infty} \\ \text{Relative forward error} &= \frac{\|\mathbf{x} - \mathbf{x}^*\|_\infty}{\|\mathbf{x}\|_\infty} \end{aligned}$$

Error Magnification Factor

This error is defined as the ratio of relative forward error to relative backward error, i.e.,

$$\text{Error magnification factor} = \frac{\|\mathbf{x} - \mathbf{x}^*\|_\infty / \|\mathbf{x}\|_\infty}{\|\mathbf{r}\|_\infty / \|\mathbf{b}\|_\infty}$$

Definitions

A system of equations is called **ill conditioned** if small changes in the input lead to large changes in the solution. Ill conditioned system has a large condition number. A system of equations is called **well conditioned** if small changes in the input lead to small changes in the solution. The condition number of the coefficient matrix is some kind measure of conditioning.

Theorem

The condition number of A, $\kappa(A)$, is the maximum possible error magnification factor, i.e., $\kappa(A)$ is an upper bound for all error magnification factors, i.e.,

$$\frac{\|\mathbf{x}-\mathbf{x}^*\|_\infty / \|\mathbf{x}\|_\infty}{\|\mathbf{r}\|_\infty / \|\mathbf{b}\|_\infty} \leq \kappa(A)$$

Proof

Let $\mathbf{x}^*$ be an approximate solution of the linear system

$$A\mathbf{x} = \mathbf{b}$$

This yields

$$\|\mathbf{b}\|_\infty = \|A\mathbf{x}\|_\infty \leq \|A\|_\infty . \|\mathbf{x}\|_\infty \tag{3.21}$$

Also we have

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}^* = A\mathbf{x} - A\mathbf{x}^* = A(\mathbf{x}-\mathbf{x}^*)$$

i.e., $\mathbf{x}-\mathbf{x}^* = A^{-1}\mathbf{r}$ so that we can write

$$\|\mathbf{x}-\mathbf{x}^*\|_\infty \leq \left\|A^{-1}\right\|_\infty . \|\mathbf{r}\|_\infty \tag{3.22}$$

From equations (3.21) and (3.22), we obtain

$$\|\mathbf{x}-\mathbf{x}^*\|_\infty . \|\mathbf{b}\|_\infty \leq \|A\|_\infty . \|\mathbf{x}\|_\infty . \left\|A^{-1}\right\|_\infty . \|\mathbf{r}\|_\infty$$

which gives us

$$\frac{\|\mathbf{x}-\mathbf{x}^*\|_\infty . \|\mathbf{b}\|_\infty}{\|\mathbf{x}\|_\infty . \|\mathbf{r}\|_\infty} \leq \|A\|_\infty . \left\|A^{-1}\right\|_\infty = \kappa(A)$$

i.e.,

$$\text{Error magnification factor} = \frac{\|\mathbf{x}-\mathbf{x}^*\|_\infty / \|\mathbf{x}\|_\infty}{\|\mathbf{r}\|_\infty / \|\mathbf{b}\|_\infty} \leq \kappa(A)$$

Example 1

Consider the linear system $A\mathbf{x} = \mathbf{b}$ where A and $\mathbf{b}$ are given by

$$A = \begin{bmatrix} 1 & 1 \\ 1.001 & 1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 2.001 \end{bmatrix}$$

Find the forward and backward errors for approximate solution $\mathbf{x}^* = \begin{bmatrix} -1 \\ 3.001 \end{bmatrix}$ using the solution $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ obtained from Gaussian elimination. Also find the error magnification factor and the condition number, $\kappa(A)$.

Here we use Gaussian elimination first

$$\begin{aligned} \begin{bmatrix} 1 & 1 \\ 1.001 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 2 \\ 2.001 \end{bmatrix} \\ \begin{bmatrix} 1 & 1 \\ 0 & -0.001 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} &= \begin{bmatrix} 2 \\ -0.001 \end{bmatrix} \qquad R_2 - 1.001R_1 \to R_2 \end{aligned}$$

which yields

$$\begin{aligned} x_1 + x_2 &= 2 \\ -0.001x_2 &= -0.001 \end{aligned}$$

Thus we obtain $x_2 = 1$ and $x_1 = 1$. Hence

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Now we compute the residual vector $\mathbf{r}$

$$\begin{aligned} \mathbf{r} &= \mathbf{b} - A\mathbf{x}^* = \begin{bmatrix} 2 \\ 2.001 \end{bmatrix} - \begin{bmatrix} 1 & 1 \\ 1.001 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 3.001 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 2.001 \end{bmatrix} - \begin{bmatrix} 2.001 \\ 2 \end{bmatrix} = \begin{bmatrix} -0.001 \\ 0.001 \end{bmatrix} \end{aligned}$$

Hence the backward error is $\|\mathbf{r}\|_\infty = 0.001$. Also $\|\mathbf{b}\|_\infty = 2.001$. This implies that

$$\text{Relative backward error} \quad = \quad \frac{0.001}{2.001} \approx 0.0005$$

Now we compute the forward error.

$$\mathbf{x} - \mathbf{x}^* \quad = \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 3.001 \end{bmatrix} = \begin{bmatrix} 2 \\ -2.001 \end{bmatrix}$$

Hence the forward error is $\|\mathbf{x} - \mathbf{x}^*\|_\infty = 2.001$. Also $\|\mathbf{x}\|_\infty = 1$. This implies that

$$\text{Relative forward error} \quad = \quad \frac{2.001}{1} = 2.001$$

Thus we have

$$\text{Error magnification factor} \quad = \quad \frac{2.001}{0.001/2.001} = 4004.001$$

To compute $\kappa(A)$,we calculate $\|A\|_\infty$ and $\|A^{-1}\|_\infty$ for the matrix $A = \begin{bmatrix} 1 & 1 \\ 1.001 & 1 \end{bmatrix}$.

$$\begin{aligned} \|A\|_\infty \quad &= \quad \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{ij}| = \text{maximum absolute row sum} \\ &= \quad 2.001 \end{aligned}$$

A^{-1} is given by

$$\begin{aligned} A^{-1} \quad &= \quad \frac{1}{|A|} \operatorname{Adj}(A) \\ &= \quad \begin{bmatrix} -1000 & 1000 \\ 1001 & -1000 \end{bmatrix} \end{aligned}$$

yielding

$$\left\|A^{-1}\right\|_\infty \quad = \quad 2001$$

Thus

$$\kappa(A) \quad = \quad \|A\|_\infty . \left\|A^{-1}\right\|_\infty = 2.001 \times 2001 = 4004.001$$

Exercise

1. Compare the solutions of the following two systems

$$\begin{aligned} x_1 - x_2 &= 1 \\ x_1 - 1.00001x_2 &= 0 \end{aligned} \qquad \begin{aligned} x_1 - x_2 &= 1 \\ x_1 - 0.9999x_2 &= 0 \end{aligned}$$

2. Consider the following system

$$\begin{aligned} x_1 - x_2 &= 5 \\ cx_1 - x_2 &= 4 \end{aligned}$$

   Find the condition numbers of the coefficient matrix when $|c| > 1$ and $|c| < 1$.

3. Consider the linear system $A\mathbf{x} = \mathbf{b}$ where A and $\mathbf{b}$ are given by

$$A = \begin{bmatrix} 1 & 1 \\ 1.001 & 1 \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} 2 \\ 2.001 \end{bmatrix}$$

   Find the forward and backward errors, the error magnification factor and the condition number, $\kappa(A)$ for approximate solution $\mathbf{x}^* = \begin{bmatrix} -1 \\ 2.999 \end{bmatrix}$.

4. (a) Find the condition number of the coefficient matrix for the following system

$$\begin{bmatrix} 1 & 1 \\ 1+\epsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2+\epsilon \end{bmatrix}$$

   as a function of ϵ.
   (b) Find the error magnification factor for the approximate solution $x^* = (-1, 3+\epsilon)^T$.

3.7 Iterative Methods

The methods discussed in this chapter are direct methods to solve linear equations. Direct methods yield the exact solution after finite number of steps. Although we loose some precision because of computing errors. Condition number is a measurement for this loss of precision. Root finding methods are iterative in nature as we have seen it before. Iterative methods are also used to solve linear systems of equations. Various iterative methods we will discuss here are

- Jacobi Method
- Gauss-Seidel Method
- SOR

3.7.1 Jacobi Method

Consider the following linear system

$$\begin{aligned} 8x_1 - x_2 &= 6 \\ 2x_1 + 5x_2 &= 5 \end{aligned}$$

Solving the first equation for x_1 and the second equation for x_2, we obtain

$$\begin{aligned} x_1 &= \frac{1}{8}[6 + x_2] \\ x_2 &= \frac{1}{5}[5 - 2x_1] \end{aligned}$$

Here we obtain an approximate solution vector $\mathbf{x}^{(k+1)}$ from the previous approximate solution vector $\mathbf{x}^{(k)}$. We start with an initial guess $\mathbf{x}^{(0)} = \left(x_1^{(0)}, x_2^{(0)}\right)^T$ to obtain $\mathbf{x}^{(1)}$ and continue this process till the stopping criteria or convergence criteria is achieved.

Thus we have

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{8}\left[6 + x_2^{(k)}\right] \\ x_2^{(k+1)} &= \frac{1}{5}\left[5 - 2x_1^{(k)}\right] \end{aligned}$$

Using the above relationships and an initial guess $\mathbf{x}^{(0)} = (0,\, 0)^T$, we obtain a sequence of solution vectors as

k	$x_1^{(k)}$	$x_2^{(k)}$
0	0	0
1	$\frac{3}{4}$	1
2	$\frac{7}{8}$	$\frac{7}{10}$
3	$\vdots$	$\vdots$

Let us consider the following system

$$\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \ldots. \quad \ldots. + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \ldots. \quad \ldots. + a_{2n}x_n & = & b_2 \\ \vdots \qquad\qquad\qquad \vdots & \vdots & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \ldots. \quad \ldots. + a_{nn}x_n & = & b_n \end{array}$$

Matrix form is given by

$$A\mathbf{x} \;=\; \mathbf{b} \tag{3.23}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Solving the ith equation for the variable x_i $(i = 1, 2, \ldots.n)$, we can express x_i for $i = 1, 2, \ldots.., n$ as the following

$$x_i \;=\; \frac{1}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j \right]$$

Using the above results and introducing the iteration concept, we write $\mathbf{x}^{(k+1)}$ in terms of $\mathbf{x}^{(k)}$ for $k = 0, 1, 2, ...$ as

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j^{(k)} \right] \tag{3.24}$$

In the Jacobi method, components of $\mathbf{x}^{(k+1)}$ can be computed in any order, i.e., it can be computed simultaneously. Thus the Jacobi method is also known as **Simultaneous Relaxation.**

Algorithm for Jacobi Iteration

$$\begin{aligned} x_i^{(0)} &= \text{initial vector} \\ x_i^{(k+1)} &= \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right] \end{aligned}$$

Another way we can express Jacobi iteration is to use matrix notation.

Matrix Splitting

Let D denote the main diagonal of the matrix A. Suppose L and U represent the strict lower triangle and strict upper triangle of A respectively. Strict means diagonal elements in lower/upper triangular matrix are zeros. Then we can express as

$$A = D + L + U$$

For example

$$\begin{bmatrix} 9 & 8 & 7 \\ 6 & 5 & 4 \\ 3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 3 & 2 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 8 & 7 \\ 0 & 0 & 4 \\ 0 & 0 & 0 \end{bmatrix}$$

Splitting A into D, L and U, we have

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ (D+L+U)\,\mathbf{x} &= \mathbf{b} \\ D\mathbf{x} &= \mathbf{b}-(L+U)\,\mathbf{x} \\ \mathbf{x} &= D^{-1}\,(\mathbf{b}-(L+U)\,\mathbf{x}) \end{aligned}$$

Algorithm for Jacobi Iteration in matrix form

$$\begin{aligned} \mathbf{x}^{(0)} &= \text{initial vector} \\ \mathbf{x}^{(k+1)} &= D^{-1}\left(\mathbf{b}-(L+U)\,\mathbf{x}^{(k)}\right) \end{aligned}$$

Result

If D is a diagonal matrix given by

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

then its inverse D^{-1}can be obtained as

$$D^{-1} = \begin{bmatrix} 1/a_{11} & 0 & \cdots & 0 \\ 0 & 1/a_{22} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & 1/a_{nn} \end{bmatrix}$$

Example 1

Solve the following linear system using Jacobi method

$$\begin{aligned} 3x_1+x_2 &= 5 \\ x_1+2x_2 &= 5 \end{aligned}$$

Here we can write

$$\begin{aligned} x_1 &= \frac{5 - x_2}{3} \\ x_2 &= \frac{5 - x_1}{2} \end{aligned}$$

Please see below.

Example 2

Solve the following linear system using Jacobi method

$$\begin{aligned} x_1 + 2x_2 &= 5 \\ 3x_1 + x_2 &= 5 \end{aligned}$$

Here we can write

$$\begin{aligned} x_1 &= 5 - 2x_2 \\ x_2 &= 5 - 3x_1 \end{aligned}$$

```
Example 1
x0 = {0, 0}; A = {{3, 1}, {1,2}}; b = {5, 5};
Initial Approximation: 0.0 0.0
------------------------------------------------------------
iter #      x1                   x2
1      1.666666666667       2.500000000000
2      0.833333333333       1.666666666667
3       1.111111111111       2.083333333333
4       0.972222222222       1.944444444444
5       1.018518518519       2.013888888889
6       0.995370370370       1.990740740741
7       1.003086419753       2.002314814815
8       0.999228395062       1.998456790123
9       1.000514403292       2.000385802469
10       0.999871399177       1.999742798354
11       1.000085733882       2.000064300412
12       0.999978566529       1.999957133059
```

Example 2

x0 = {0, 0}; A = {{1,2}, {3,1}}; b = {5, 5};

Initial Approximation: 0.0 0.0

----

iter	x_1	x_2
1	5	5
2	-5	-10
3	25	20
4	-35	-70
5	145	110
6	-215	-430
7	865	650
8	-1,295	-2,590
9	5,185	3,890
10	-7,775	-15,550

Remarks:

From these two examples we see that Jacobi method always does not converge. To converge, the coefficient matrix A needs to be **strictly diagonally dominant**. A matrix $A = [a_{ij}]$ is strictly diagonally dominant if

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

Example 3

Solve the following linear system using Jacobi method

$$\begin{aligned} 3x_1 + x_2 + x_3 &= 1 \\ 3x_1 + 6x_2 + 2x_3 &= 0 \\ 3x_1 + 3x_2 + 7x_3 &= 4 \end{aligned}$$

3.7.2 Gauss-Seidel Method

Gauss-Seidel method is a modified version of Jacobi iteration method. In Gauss-Seidel method, new information is used as soon as it is available. Jacobi iteration gives

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right]$$

Gauss-Seidel iteration is obtained by modifying the second expression in the above equation using the latest solutions available as follows

$$x_i^{(k+1)} = \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right]$$

In Gauss-Seidel method, we need to compute $\mathbf{x}^{(k+1)}$ in succession. Thus the Gauss-Seidel method is also known as **Successive Relaxation** method.

Algorithm for Gauss-Seidel Iteration

$$\begin{aligned} x_i^{(0)} &= \text{initial vector} \\ x_i^{(k+1)} &= \frac{1}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right] \end{aligned}$$

Algorithm for Gauss-Seidel in matrix form

$$\begin{aligned} \mathbf{x}^{(0)} &= \text{initial vector} \\ \mathbf{x}^{(k+1)} &= D^{-1}\left(\mathbf{b} - L\mathbf{x}^{(k+1)} - U\mathbf{x}^{(k)}\right) \end{aligned}$$

Example 4

Solve the following linear system using Gauss-seidel method

$$\begin{aligned} 9x_1 - 3x_2 + x_3 &= 2 \\ -2x_1 + 7x_2 + 3x_3 &= 9 \\ x_1 - 4x_2 - 8x_3 &= 1 \end{aligned}$$

Example 5

Solve the following linear system using Gauss-seidel method

$$\begin{aligned} 3x_1 + x_2 + x_3 &= 1 \\ 3x_1 + 6x_2 + 2x_3 &= 0 \\ 3x_1 + 3x_2 + 7x_3 &= 4 \end{aligned}$$

3.7.3 Successive Over Relaxation (SOR)

This method is modified version of Gauss-Seidel method by computing $x_i^{(k+1)}$ as a weighted average of $x_i^{(k)}$. This is achieved as follows

$$x_i^{(k+1)} = (1-w)\,x_i^{(k)} + \frac{w}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij}x_j^{(k)}\right]$$

If $w = 1$, it reduces to the Gauss-Seidel method. This method is called successive under relaxation for $0 < w < 1$ and can be used to obtain convergent solution for some systems that are not convergent by the Gauss-Seidel method. For $1 < w < 2$, this method is called as successive over relaxation and can be used to accelerate the convergence of the systems that are convergent by the Gauss-Seidel method.

Algorithm for SOR Iteration

$$\begin{aligned} x_i^{(0)} &= \text{initial vector} \\ x_i^{(k+1)} &= (1-w)\,x_i^{(k)} + \frac{w}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right] \end{aligned}$$

SOR in matrix form

$$\begin{aligned} A\mathbf{x} &= \mathbf{b} \\ wA\mathbf{x} &= w\mathbf{b} \\ (wD + wL + wU)\,\mathbf{x} &= w\mathbf{b} \\ (wL + D)\,\mathbf{x} &= w\mathbf{b} - wU\mathbf{x} + (1-w)\,D\mathbf{x} \\ \mathbf{x} &= (wL+D)^{-1}\left[w\mathbf{b} - wU\mathbf{x} + (1-w)\,D\mathbf{x}\right] \end{aligned}$$

Algorithm

$$\begin{aligned} \mathbf{x}^{(0)} &= \text{initial vector} \\ \mathbf{x}^{(k+1)} &= (wL+D)^{-1}\left[w\mathbf{b} - wU\mathbf{x}^{(k)} + (1-w)\,D\mathbf{x}^{(k)}\right] \end{aligned}$$

Example 6

Solve the following linear system using SOR method

$$\begin{aligned} 9x_1 - 3x_2 + x_3 &= 2 \\ -2x_1 + 7x_2 + 3x_3 &= 9 \\ x_1 - 4x_2 - 8x_3 &= 1 \end{aligned}$$

3.7.4 Convergence and Stopping Tolerance Criteria

When does Jacobi method yield a solution? If Jacobi method gives us a solution of $A\mathbf{x} = \mathbf{b}$, why do we need Gauss-Seidel and/or SOR?

If the matrix A is strictly diagonally dominant, then the Jacobi and Gauss-Seidel methods yield a convergent solution. For any coefficient matrix with no zero diagonal element, SOR converges for $0 < w < 2$ and diverges for $w < 0$ and $w \geq 2$. Best performance for SOR is obtained when $w \in (1, 2)$. If $w = 1$, SOR becomes Gauss-Seidel method. Typically SOR converges faster than Gauss-Seidel does. Convergence in Gauss-Seidel is faster than the convergence in Jacobi iteration method. But the above may not valid for some specific cases. Components for the solution vector can be obtained simultaneously in Jacobi method whereas solutions in Gauss-Seidel and SOR are obtained successively only. Also if A is a SPD (symmetric positive definite) matrix, then the Jacobi and Gauss-Seidel methods converge. For a SPD matrix A and $0 < w < 2$, the SOR method converges for any choice of initial approximation $\mathbf{x}^{(0)}$.

Tolerance Parameter

For a given small number $\epsilon > 0$, the criteria to stop an iterative process is $\left\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right\|_\infty < \epsilon$. User can provide ϵ to be 10^{-5} or 10^{-6} etc as a parameter to the program. Number of iterations can also be used together with the ϵ-criteria as the stopping criteria.

3.7.5 Conjugate Gradient Method

Some iterative methods uses different approach to solve a linear system than using splitting the coefficinet matrix and converting to a fixed point problem. One of these classes is based on the equivalence between the solution of a linear system and the minimization of an associated quadratic functional. Congjugate gradient method belongs to such class and is used to large sparse systems. Title CGM comes from ehat his method does: sliding down the slopes of a quadratic paraboloid in n dimensions. The gradient aprt means that it is finding the direction of fgastest decline by using Calculus, and conjugate means, not quite that its individual steps are orthogonal to one anothert, but that at leasat the residuals are.

Reference: [MR Hestenes and E Steifel, 'Conjugate Gradient Methods in Optimization', J. Research National Bureau of Standards, 49, pp. 409-436, (1952)].

Minimizing a Quadratic Functional to Solve a Linear System

Let $\mathbf{A}$ be a sysmmetric and positive definite $n \times n$ matrix. We define $f : \mathbb{R}^n \to \mathbb{R}$ such that

$$f(\mathbf{x}) \quad = \quad \frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x} - \mathbf{b}^T\mathbf{x} + c$$

Since $\mathbf{A}$ is positive definite, f behaves like an upward opening parabola and has a unique global minimizer which implies that there exists a unique vector $\mathbf{x}^*$ such $f(\mathbf{x}^*) < f(\mathbf{x})$ that for all $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x}^* \neq \mathbf{x}$. Now, we have

$$\nabla f \quad = \quad \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_1}, \cdots\cdots, \frac{\partial f}{\partial x_1},\right]^T$$

The gradient of f, ∇f has to zero at $\mathbf{x}^*$ the location of the minimum values of f. Writing $f(\mathbf{x})$ in terms of the elements of $\mathbf{A}$ and the components of $\mathbf{x}$, we have

$$f(\mathbf{x}) \quad = \quad \frac{1}{2}\sum_{i=1}^{n} x_i \left(\sum_{j=1}^{n} a_{ij}x_j\right) - \sum_{j=1}^{n} b_j x_j + c.$$

Now differentiating w.r.t. k,we have

$$\begin{aligned}
\frac{\partial f}{\partial x_k} &= \frac{1}{2}\left(\sum_{i=1}^{n} x_i a_{ik} + \sum_{j=1}^{n} a_{kj} x_j\right) - b_k \\
&= \frac{1}{2}\left(\sum_{i=1}^{n} a_{ki} x_i + \sum_{j=1}^{n} a_{kj} x_j\right) - b_k \\
&= \sum_{j=1}^{n} a_{kj} x_j - b_k = (\mathbf{A}\mathbf{x} - \mathbf{b})_k\,,
\end{aligned}$$

due to symmetry of $\mathbf{A}$. Hence,

$$\nabla f = \mathbf{A}\mathbf{x} - \mathbf{b}$$

and locating the minimum of f is same as solving

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

How do we obtain (approximate) the global minimizer of f ? We start with an initial approximation $\mathbf{x}^{(0)}$ and generate a sequence $\left\{\mathbf{x}^{(n)}\right\}$ such that

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha_n \mathbf{d}^{(n)}$$

where α_n is called the step size and $\mathbf{d}^{(n)}$ is the search direction. Different minimizer method uses different choices of step size and search direction. Here we obtain those for the conjugate gradient method.

Step Size:

Assuming that we know the approximate minimizer $\mathbf{x}^{(n)}$ and search direction $\mathbf{d}^{(n)}$, in the following, we obtain the step size $\alpha^{(n)}$. We select α_n so that $f(\mathbf{x})$ is minimized along the line $\mathbf{x} = \mathbf{x}^{(n)} + \alpha\mathbf{d}^{(n)}$. Now Putting $\mathbf{x} = \mathbf{x}^{(n)} + \alpha\mathbf{d}^{(n)}$

in $f(\mathbf{x})$, we get

$$\begin{aligned}
f\left(\mathbf{x}^{(n)}+\alpha\mathbf{d}^{(n)}\right) &= \frac{1}{2}\left(\mathbf{x}^{(n)}+\alpha\mathbf{d}^{(n)}\right)^T\mathbf{A}\left(\mathbf{x}^{(n)}+\alpha\mathbf{d}^{(n)}\right)-\mathbf{b}^T\left(\mathbf{x}^{(n)}+\alpha\mathbf{d}^{(n)}\right)+c \\
&= \frac{1}{2}\mathbf{x}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}+\frac{1}{2}\alpha\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}+\frac{1}{2}\alpha\mathbf{x}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)} \\
&\qquad +\frac{1}{2}\alpha^2\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}-\mathbf{b}^T\mathbf{x}^{(n)}-\mathbf{b}^T\alpha\mathbf{d}^{(n)}+c \\
&= \left[\frac{1}{2}\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}\right]\alpha^2+\left[\frac{1}{2}\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}+\frac{1}{2}\mathbf{x}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}-\mathbf{b}^T\mathbf{d}^{(n)}\right]\alpha \\
&\qquad \frac{1}{2}\mathbf{x}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}-\mathbf{b}^T\mathbf{x}^{(n)}+c
\end{aligned}$$

Since $\mathbf{A}$ is symmetric and for any two vectors $\mathbf{t}$ and $\mathbf{s}$, $\mathbf{t}^T\mathbf{s}=\mathbf{s}^T\mathbf{t}$, we can write

$$\mathbf{x}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}=\mathbf{x}^{(n)^T}\mathbf{A}^T\mathbf{d}^{(n)}=\left(\mathbf{A}\mathbf{x}^{(n)}\right)^T\mathbf{d}^{(n)}=\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}$$

and $\mathbf{b}^T\mathbf{d}^{(n)}=\left(\mathbf{d}^{(n)}\right)^T\mathbf{b}$. Thus,

$$\begin{aligned}
f\left(\mathbf{x}^{(n)}+\alpha\mathbf{d}^{(n)}\right) &= \left[\frac{1}{2}\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}\right]\alpha^2+\left[\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}-\left(\mathbf{d}^{(n)}\right)^T\mathbf{b}\right]\alpha+c \\
&= \left[\frac{1}{2}\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}\right]\alpha^2-\left[\mathbf{d}^{(n)^T}\mathbf{r}^{(n)}\right]\alpha+c_1
\end{aligned}$$

where $c_1=\frac{1}{2}\mathbf{x}^{(n)^T}\mathbf{A}\mathbf{x}^{(n)}-\mathbf{b}^T\mathbf{x}^{(n)}+c$ and $\mathbf{r}^{(n)}=\mathbf{b}-\mathbf{A}\mathbf{x}^{(n)}$, the residual associated with approximation $\mathbf{x}^{(n)}$. Since $\mathbf{A}$ is positive definite, we have $\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}>0$ for $\mathbf{d}^{(n)}\neq 0$. So, $f\left(\mathbf{x}^{(n)}+\alpha\mathbf{d}^{(n)}\right)$ is an upward opening parabola in α implying that it achieves its minimum value when

$$\frac{\partial f}{\partial\alpha} = \mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}\alpha-\mathbf{d}^{(n)^T}\mathbf{r}^{(n)}=0.$$

So, we obtain

$$\alpha_n = \frac{\mathbf{d}^{(n)^T}\mathbf{r}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}.$$

Search Direction:

Search directions are based on the gradient of f. We see that

$$\nabla f\left(\mathbf{x}^{(n)}\right) = \mathbf{A}\mathbf{x}^{(n)} - \mathbf{b}$$

as $\nabla f = \mathbf{A}\mathbf{x} - \mathbf{b}$. We take $\mathbf{r}^{(n)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(n)}$. If $\mathbf{x}^{(0)} = \mathbf{0}$, then $\mathbf{r}^{(0)} = \mathbf{b}$.

For the first iteration, we choose search direction as

$$\mathbf{d}^{(0)} = \mathbf{r}^{(0)}$$

i.e., we start from $\mathbf{x}^{(0)}$ and travel in opposite direction of the gradient which guarantees that this is the direction of maximum decrease in the value of f. Other subsequent search directions are determined by

$$\mathbf{d}^{(n+1)} = \mathbf{r}^{(n+1)} + \lambda_n \mathbf{d}^{(n)}, \quad n \geq 0.$$

Here, λ_n is chosen so that the search direction $\mathbf{d}^{(n+1)}$ is A-conjugate to the direction $\mathbf{d}^{(n)}$.

A-Conjugate:

For a symmetric matrix $\mathbf{A}$, two vectors $\mathbf{t}$ and $\mathbf{s}$, are called A-Conjugate if

$$\mathbf{t}^T\mathbf{A}\mathbf{s} = 0.$$

Now, we compute

$$\begin{aligned}\mathbf{d}^{(n+1)^T}\mathbf{A}\mathbf{d}^{(n)} &= \left(\mathbf{r}^{(n+1)} + \lambda_n \mathbf{d}^{(n)}\right)^T \mathbf{A}\mathbf{d}^{(n)} \\ &= \mathbf{r}^{(n+1)^T}\mathbf{A}\mathbf{d}^{(n)} + \lambda_n \mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)} = 0\end{aligned}$$

Thus, we have

$$\lambda_n = -\frac{\mathbf{r}^{(n+1)^T}\mathbf{A}\mathbf{d}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}.$$

Algorithm for Conjugate Gradient Method:

```
$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$
$\mathbf{d}^{(0)} = \mathbf{r}^{(0)}$
for  $n = 0, 1, 2, .......$
```

```
        $\alpha_n = \frac{\mathbf{d}^{(n)^T}\mathbf{r}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}$
        $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha_n \mathbf{d}^{(n)}$
        $\mathbf{r}^{(n+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(n+1)}$
        if $\sqrt{\mathbf{r}^{(n+1)^T}\mathbf{r}^{(n+1)}} <$ TOL, PRINT $\mathbf{x}^{(n+1)}$
        $\lambda_n = -\frac{\mathbf{r}^{(n+1)^T}\mathbf{A}\mathbf{d}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}$
        $\mathbf{d}^{(n+1)} = \mathbf{r}^{(n+1)} + \lambda_n \mathbf{d}^{(n)}$
    end for n
```

Efficient Conjugate Gradient Method:

We calculate $\mathbf{r}^{(n+1)}$ from $\mathbf{r}^{(n)}$. We have

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha_n \mathbf{d}^{(n)}$$

so we can write

$$\mathbf{r}^{(n+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(n+1)} = \mathbf{r}^{(n+1)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(n)} - \alpha_n \mathbf{A}\mathbf{d}^{(n)}$$

Hence,

$$\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \alpha_n \mathbf{A}\mathbf{d}^{(n)}$$

Also,

$$\begin{aligned}
\mathbf{d}^{(n)^T}\mathbf{r}^{(n+1)} &= \mathbf{d}^{(n)^T}\mathbf{r}^{(n)} - \alpha_n \mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)} \\
&= \mathbf{d}^{(n)^T}\mathbf{r}^{(n)} - \frac{\mathbf{d}^{(n)^T}\mathbf{r}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)} \\
&= \mathbf{d}^{(n)^T}\mathbf{r}^{(n)} - \mathbf{d}^{(n)^T}\mathbf{r}^{(n)} = 0
\end{aligned}$$

Thus, the previous search direction and the new residual vector are orthogonal. Choice of λ_n to make the search direction A-Conjugate yields that the residual vectors are orthogonal to one another, i.e.,

$$\mathbf{r}^{(n)^T}\mathbf{r}^{(m)} = 0 \qquad n \neq m.$$

Now $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \alpha_n \mathbf{A}\mathbf{d}^{(n)}$ allows us to write

$$\mathbf{A}\mathbf{d}^{(n)} = \frac{1}{\alpha_n}\left[\mathbf{r}^{(n)} - \mathbf{r}^{(n+1)}\right] \tag{3.25}$$

Premultiplying by $\mathbf{d}^{(n)^T}$, we obtain

$$\begin{aligned}
\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)} &= \mathbf{d}^{(n)^T}\frac{1}{\alpha_n}\left[\mathbf{r}^{(n)} - \mathbf{r}^{(n+1)}\right] \\
&= \frac{1}{\alpha_n}\mathbf{d}^{(n)^T}\mathbf{r}^{(n)} \\
&= \frac{1}{\alpha_n}\left[\mathbf{r}^{(n)} + \lambda_{n-1}\mathbf{d}^{(n-1)}\right]^T\mathbf{r}^{(n)} \\
& \quad \frac{1}{\alpha_n}\mathbf{r}^{(n)^T}\mathbf{r}^{(n)}
\end{aligned}$$

Thus,

$$\alpha_n = \frac{\mathbf{r}^{(n)^T}\mathbf{r}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}$$

Also, from (3.25)

$$\begin{aligned}
\mathbf{r}^{(n+1)^T}\mathbf{A}\mathbf{d}^{(n)} &= \mathbf{r}^{(n+1)^T}\frac{1}{\alpha_n}\left[\mathbf{r}^{(n)} - \mathbf{r}^{(n+1)}\right] \\
&= -\frac{1}{\alpha_n}\mathbf{r}^{(n+1)^T}\mathbf{r}^{(n+1)}
\end{aligned}$$

So we obtain

$$\lambda_n = -\frac{\mathbf{r}^{(n+1)^T}\mathbf{A}\mathbf{d}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}} = \frac{\mathbf{r}^{(n+1)^T}\mathbf{r}^{(n+1)}}{\alpha_n\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}} = \frac{\mathbf{r}^{(n+1)^T}\mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)^T}\mathbf{r}^{(n)}}.$$

Efficient Algorithm for Conjugate Gradient Method:

```
$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$
$\mathbf{d}^{(0)} = \mathbf{r}^{(0)}$
$\beta_0 = \mathbf{r}^{(0)^T}\mathbf{r}^{(0)}$
for  $n = 0, 1, 2, .......$
    $\mathbf{v} = \mathbf{A}\mathbf{d}^{(n)}$
    $\alpha_n = \frac{\beta_n}{\mathbf{d}^{(n)^T}\mathbf{v}}$
    $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \alpha_n\mathbf{d}^{(n)}$
    $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \alpha_n\mathbf{v}$
    set $\beta_{n+1} = \mathbf{r}^{(n+1)^T}\mathbf{r}^{(n+1)}$
    if $\sqrt{\beta_{n+1}} <$ TOL, PRINT $\mathbf{x}^{(n+1)}$
    $\lambda_n = \frac{\beta_{n+1}}{\beta_n}$
    $\mathbf{d}^{(n+1)} = \mathbf{r}^{(n+1)} + \lambda_n\mathbf{d}^{(n)}$
end for n
```

Why CGM preferred/not preferred over Gaussian Elimination (GE)? Why CGM treated as iterative method?

CGM and Gaussian Elimination are direct methods. They yield the theoretically correct solution in a finite number of steps.

CGM is somewhat simpler to implement than the implementation of Gaussian elimination. There are no row operations to worry about, no triple loop as in GE. Operation count contributes to the answer why CGM not preferred over GE. Moving through loop requires one matrix-vector product $\mathbf{Ad}^{(n)}$ and several dot products. The matrix-vector product alone requires n^2 multiplications for each step (along with about the same number of additions), for a total of n^3 multiplications after n steps. Compared with the count $n^3/3$ for Gaussian elimination, which is tree times more expensive.

This scenario changes if $\mathbf{A}$ is sparse. If n is too large then $n^3/3$ operations (needed for GE) may not be feasible. Although GE must be run to to completion to give a solution, CGM gives an approximation at each step. The Euclidean length of the residual decreases on each step, i.e., $\mathbf{Ax}^{(n)}$ gets closer to $\mathbf{b}$ on each step. Thus, monitoring the residual $\mathbf{r}^{(n)}$, an approximate solution can be found to avoid completing all n steps. In this sense, CGM is considered as an iterative method.

Preconditioning

Here we discuss why do we need proconditioning and how does it work. CGM fell out of favor due to its susceptibility to accumulation of round-off errors for ill-condotioned matrix $\mathbf{A}$. Its performance on ill-conditioned matrices is inferior to Gaussian elimination with partial pivoting. Using preconditioning essentially changes the problem to one of solving a better-conditioned matrix system, after which CGM is applied.

The main idea underlying any preconditioning process for iterative solvers is to modify the ill-conditioned system

$$\mathbf{Ax} \;=\; \mathbf{b}$$

so that we find an equivalent system

$$\widehat{\mathbf{A}}\widehat{\mathbf{x}} \;=\; \widehat{\mathbf{b}}$$

for which the iterative method converged faster. A classical approach is to use a non-singular matrix $\mathbf{P}$, so that we can write

$$\mathbf{P}^{-1}\mathbf{Ax} \;=\; \mathbf{P}^{-1}\mathbf{b}.$$

The preconditioner $\mathbf{P}$ is chosen such that the matrix $\widehat{\mathbf{A}} = \mathbf{P}^{-1}\mathbf{A}$ is better conditioned for the CGM, or has better clustered eigenvalues for the GMRES method.

The number of iterations required for the convergence of conjugate gradient algorithm to converge is proportional to $\sqrt{\kappa(\mathbf{A})}$. So, for poorly conditioned matrices, convergence will be very slow. We want to choose $\mathbf{P}$ such that a faster convergence is obtained, i.e., $\kappa\left(\widehat{\mathbf{A}}\right) < \kappa(\mathbf{A})$.

It order to retain symmetry and positive definiteness of $\hat{\mathbf{A}}$ we take

$$\mathbf{P}^{-1} \;=\; \mathbf{MM}^T$$

where $\mathbf{M}$ is a nonsingular matrix of order n. Now,

$$\begin{aligned}
\mathbf{Ax} = \mathbf{b} \;&\Longleftrightarrow\; \mathbf{P^{-1}Ax} = \mathbf{P^{-1}b} \\
&\Longleftrightarrow\; \mathbf{M}^T\mathbf{Ax} = \mathbf{M}^T\mathbf{b} \\
&\Longleftrightarrow\; \mathbf{M}^T\mathbf{A}(\mathbf{MM^{-1}})\mathbf{x} = \mathbf{M}^T\mathbf{b} \\
&\Longleftrightarrow\; (\mathbf{M}^T\mathbf{AM})(\mathbf{M}^{-1}\mathbf{x}) = \mathbf{M}^T\mathbf{b} \\
&\Longleftrightarrow\; \widehat{\mathbf{A}}\widehat{\mathbf{x}} = \widehat{\mathbf{b}}
\end{aligned}$$

where

$$\widehat{\mathbf{A}} = \mathbf{M}^T\mathbf{A}\mathbf{M}, \qquad \widehat{\mathbf{x}} = \mathbf{M}^{\mathbf{-1}}\mathbf{x}, \qquad \widehat{\mathbf{b}} = \mathbf{M}^T\mathbf{b}.$$

The symmetric positive definite matrix $\mathbf{P}$ is called splitting matrix or preconditioner. It can be shown that $\widehat{\mathbf{A}}$ is symmetric positive definite. One can use CGM with these hatted quantities. But to make more efficient, it is better to incorporate the preconditioning into the iteration.

Let's start with our new residual

$$\begin{aligned}
\widehat{\mathbf{r}}^{(n)} &= \hat{\mathbf{b}} - \widehat{\mathbf{A}}\widehat{\mathbf{x}}^{(n)} = \mathbf{M}^T\mathbf{b} - (\mathbf{M}^T\mathbf{A}\mathbf{M})(\mathbf{M}^{\mathbf{-1}}\mathbf{x}^{(n)}) \\
&= \mathbf{M}^T\mathbf{b} - \mathbf{M}^T\mathbf{A}\mathbf{x}^{(n)} = \mathbf{M}^T\left(\mathbf{b} - \mathbf{A}\mathbf{x}^{(n)}\right) \\
&= \mathbf{M}^T\mathbf{r}^{(n)}
\end{aligned}$$

We define the following:

$$\begin{aligned}
\widehat{\mathbf{d}}^{(n)} &= \mathbf{M}^{-1}\mathbf{d}^{(n)} \\
\widetilde{\mathbf{r}}^{(n)} &= \mathbf{P}^{-1}\mathbf{r}^{(n)}
\end{aligned}$$

Here we consider how to transform the CG algorithm (for the hatted quantities).

$$\widehat{\mathbf{x}}^{(0)} = \mathbf{M}^{-1}\mathbf{x}^{(0)}$$

with $\mathbf{x}^{(0)}$ as initial solution and

$$\widehat{\mathbf{r}}^{(0)} = \mathbf{M}^T\mathbf{r}^{(0)}$$

where $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$.

Now, initial search direction becomes

$$\begin{aligned}
\widehat{\mathbf{d}}^{(0)} = \widehat{\mathbf{r}}^{(0)} &\iff \mathbf{M}^{-1}\mathbf{d}^{(0)} = \mathbf{M}^T\mathbf{r}^{(0)} \\
&\iff \mathbf{d}^{(0)} = \mathbf{M}\mathbf{M}^T\mathbf{r}^{(0)} = \mathbf{P}^{-1}\mathbf{r}^{(0)} \\
&\iff \mathbf{d}^{(0)} = \widetilde{\mathbf{r}}^{(0)}
\end{aligned}$$

The step size $\widehat{\alpha}_n$ transformation is given by

$$\begin{aligned}
\widehat{\alpha}_n &= \frac{\widehat{\mathbf{r}}^{(n)^T}\widehat{\mathbf{r}}^{(n)}}{\widehat{\mathbf{d}}^{(n)^T}\widehat{\mathbf{A}}\widehat{\mathbf{d}}^{(n)}} = \frac{\left(\mathbf{M}^T\mathbf{r}^{(n)}\right)^T\left(\mathbf{M}^T\mathbf{r}^{(n)}\right)}{\left(\mathbf{M}^{-1}\mathbf{d}^{(n)}\right)^T\left(\mathbf{M}^T\mathbf{A}\mathbf{M}\right)\left(\mathbf{M}^{-1}\mathbf{d}^{(n)}\right)} \\
&= \frac{\mathbf{r}^{(n)^T}\mathbf{M}\mathbf{M}^T\mathbf{r}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{M}^{-T}\mathbf{M}^T\mathbf{A}\mathbf{M}\mathbf{M}^{-1}\mathbf{d}^{(n)}} \\
&= \frac{\mathbf{r}^{(n)^T}\mathbf{P}^{-1}\mathbf{r}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}} \\
&= \frac{\mathbf{r}^{(n)^T}\widetilde{\mathbf{r}}^{(n)}}{\mathbf{d}^{(n)^T}\mathbf{A}\mathbf{d}^{(n)}}
\end{aligned}$$

The approximate solution is modified as

$$\begin{aligned}
\widehat{\mathbf{x}}^{(n+1)} = \widehat{\mathbf{x}}^{(n)} + \widehat{\alpha}_n\widehat{\mathbf{d}}^{(n)} &\iff \mathbf{M}^{-1}\mathbf{x}^{(n+1)} = \mathbf{M}^{-1}\mathbf{x}^{(n)} + \widehat{\alpha}_n\mathbf{M}^{-1}\mathbf{d}^{(n)} \\
&\iff \mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \widehat{\alpha}_n\mathbf{d}^{(n)}
\end{aligned}$$

The residuals are updated as follows

$$\begin{aligned}
\widehat{\mathbf{r}}^{(n+1)} = \widehat{\mathbf{r}}^{(n)} - \widehat{\alpha}_n\widehat{\mathbf{A}}\widehat{\mathbf{d}}^{(n)} &\iff \mathbf{M}^T\mathbf{r}^{(n+1)} = \mathbf{M}^T\mathbf{r}^{(n)} - \widehat{\alpha}_n\left(\mathbf{M}^T\mathbf{A}\mathbf{M}\right)\left(\mathbf{M}^{-1}\mathbf{d}^{(n)}\right) \\
&\iff \mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \widehat{\alpha}_n\mathbf{A}\mathbf{d}^{(n)}
\end{aligned}$$

The gradient correction factor $\widehat{\lambda}_n$ is given by

$$\begin{aligned}
\widehat{\lambda}_n &= \frac{\widehat{\mathbf{r}}^{(n+1)^T}\widehat{\mathbf{r}}^{(n+1)}}{\widehat{\mathbf{r}}^{(n)^T}\widehat{\mathbf{r}}^{(n)}} \\
&= \frac{\left(\mathbf{M}^T\mathbf{r}^{(n+1)}\right)^T\left(\mathbf{M}^T\mathbf{r}^{(n+1)}\right)}{\left(\mathbf{M}^T\mathbf{r}^{(n)}\right)^T\left(\mathbf{M}^T\mathbf{r}^{(n)}\right)} \\
&= \frac{\mathbf{r}^{(n+1)^T}\mathbf{M}\mathbf{M}^T\mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)^T}\mathbf{M}\mathbf{M}^T\mathbf{r}^{(n)}} = \frac{\mathbf{r}^{(n+1)^T}\mathbf{P}^{-1}\mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)^T}\mathbf{P}^{-1}\mathbf{r}^{(n)}} \\
&= \frac{\mathbf{r}^{(n+1)^T}\widetilde{\mathbf{r}}^{(n+1)}}{\mathbf{r}^{(n)^T}\widetilde{\mathbf{r}}^{(n)}}
\end{aligned}$$

So, the search direction becomes

$$\begin{aligned}
\widehat{\mathbf{d}}^{(n+1)} = \widehat{\mathbf{r}}^{(n+1)} + \widehat{\lambda}_n\widehat{\mathbf{d}}^{(n)} &\iff \mathbf{M}^{-1}\mathbf{d}^{(n+1)} = \mathbf{M}^T\mathbf{r}^{(n+1)} + \widehat{\lambda}_n\mathbf{M}^{-1}\mathbf{d}^{(n)} \\
&\iff \mathbf{d}^{(n+1)} = \mathbf{P}^{-1}\mathbf{r}^{(n+1)} + \widehat{\lambda}_n\mathbf{d}^{(n)} \\
&\iff \mathbf{d}^{(n+1)} = \widetilde{\mathbf{r}}^{(n+1)} + \widehat{\lambda}_n\mathbf{d}^{(n)}
\end{aligned}$$

Algorithm for Preconditioned Conjugate Gradient Method:

```
$\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$
solve $\mathbf{P}\widetilde{\mathbf{r}}^{(0)} = \mathbf{r}^{(0)}$ for $\widetilde{\mathbf{r}}^{(0)}$
$\mathbf{d}^{(0)} = \widetilde{\mathbf{r}}^{(0)}$
$\beta_0 = \mathbf{r}^{(0)^T}\widetilde{\mathbf{r}}^{(0)}$
for $n = 0, 1, 2, .......$
    $\mathbf{v} = \mathbf{A}\mathbf{d}^{(n)}$
    $\widehat{\alpha}_n = \frac{\beta_n}{\mathbf{d}^{(n)^T}\mathbf{v}}$
    $\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \widehat{\alpha}_n\mathbf{d}^{(n)}$
    $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} - \widehat{\alpha}_n\mathbf{v}$
    solve $\mathbf{P}\widetilde{\mathbf{r}}^{(\mathrm{n+1})} = \mathbf{r}^{(\mathrm{n+1})}$ for $\widetilde{\mathbf{r}}^{(n+1)}$
    set $\beta_{n+1} = \mathbf{r}^{(n+1)^T}\widetilde{\mathbf{r}}^{(n+1)}$
    if $\sqrt{\beta_{n+1}} <$ TOL, PRINT $\mathbf{x}^{(n+1)}$
    $\widehat{\lambda}_n = \frac{\beta_{n+1}}{\beta_n}$
    $\mathbf{d}^{(n+1)} = \widetilde{\mathbf{r}}^{(n+1)} + \widehat{\lambda}_n\mathbf{d}^{(n)}$
end for n
```

Exercise

1. Compute the first two steps of the Jacobi Method with starting vector as zero vector (leave your answer in fraction form).

   (a) $\begin{bmatrix} 3 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$, (b) $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$

2. Compute the first two steps of the Gauss-Seidel Method with starting vector as zero vector (leave your answer in fraction form).

   (a) $\begin{bmatrix} 3 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$, (b) $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$

3. Compute the first two steps of the SOR Method with starting vector as zero vector and $\omega = 1.5$ (leave your answer in fraction form).

   (a) $\begin{bmatrix} 3 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \end{bmatrix}$, (b) $\begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \end{bmatrix}$

Chapter 4

Interpolation and Polynomial Approximation

Interpolation is the process of estimating the values of a function at some intermediate values for the independent variable using a tabular values of the independent and dependent variables.

Theorem (Weiestrass Approximation Theorem)

If f is a continuous function on the closed interval $[a,\ b]$ and $\epsilon > 0$, then there exists a polynomial p such that

$$\|f-p\|_{\infty} \;=\; \max_{x\in[a,b]} |f(x)-p(x)| < \epsilon$$

Theorem on Polynomial Interpolation

Let $(x_0, y_0)\,, (x_1, y_1)\,, ...\ \ ,(x_n, y_n)$ be $(n+1)$ points with distinct x_i. Then there exists a unique polynomial of degree n such that $p\,(x_i) = y_i$ for $i = 0, 1, ..., n$.

4.1 Lagrange Interpolation

Consider two data points (x_0, y_0) and $(x_1, y_1)\,.$ We want to approximate a function f such that $y_0 = f\,(x_0)$ and $y_1 = f\,(x_1)\,.$ Let us define two functions

$$L_0(x) = \frac{x - x_1}{x_0 - x_1} \qquad \text{and} \qquad L_1(x) = \frac{x - x_0}{x_1 - x_0}$$

such that

$$L_0(x) + L_1(x) \quad = \quad 1 \quad \text{for all } x \in [a, b]$$

If we write a first degree polynomial (linear)

$$p(x) \quad = \quad L_0(x)y_0 + L_1(x)y_1$$

we see that

$$L_0\left(x_0\right) = 1,\ L_0\left(x_1\right) = 0 \qquad \text{and} \qquad L_1\left(x_0\right) = 0,\ L_1\left(x_1\right) = 1$$

yielding

$$p(x_0) \quad = \quad L_0(x_0)y_0 + L_1(x_0)y_1 = 1.y_0 + 0.y_1 = y_0$$

and

$$p(x_1) \quad = \quad L_0(x_1)y_0 + L_1(x_1)y_1 = 0.y_0 + 1.y_1 = y_1$$

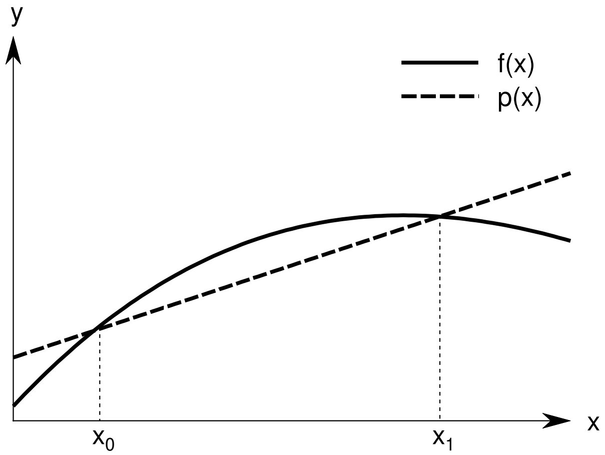
Thus $p(x)$ is the linear approximation of the function $f(x)$ passing through (x_0, y_0) and (x_1, y_1). The functions $L_0(x)$ and $L_1(x)$ are called Lagrange interpolation functions. This allows us to write

$$\begin{aligned} p(x) &= L_0(x)y_0 + L_1(x)y_1 \\ &= \frac{x - x_1}{x_0 - x_1}y_0 + \frac{x - x_0}{x_1 - x_0}y_1 \\ &= \sum_{i=0}^{1} L_i\left(x\right)y_i \end{aligned}$$

Figure 4.1 shows an example of linear interpolation polynomial $p(x)$ to approximate a function $f(x)$.

Considering three data points (x_0, y_0) ,(x_1, y_1) and (x_2, y_2), we want to approximate a function f satisfying $y_0 = f\left(x_0\right),\ y_1 = f\left(x_1\right)$ and $y_2 = f\left(x_2\right)$.

Figure 4.1: Example of linear interpolation



Assuming a second degree polynomial of the form $p(x) = a + bx + cx^2$ which is equivalent to

$$p(x) \quad = \quad a_0\,(x - x_1)\,(x - x_2) + a_1\,(x - x_0)\,(x - x_2) + a_2\,(x - x_0)\,(x - x_1)$$

it can be shown that

$$p(x) \quad = \quad L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2$$

where

$$L_0(x) = \frac{(x - x_1)\,(x - x_2)}{(x_0 - x_1)\,(x_0 - x_2)}, \quad L_1(x) = \frac{(x - x_0)\,(x - x_2)}{(x_1 - x_0)\,(x_1 - x_2)}, \quad L_2(x) = \frac{(x - x_0)\,(x - x_1)}{(x_2 - x_0)\,(x_2 - x_1)}$$

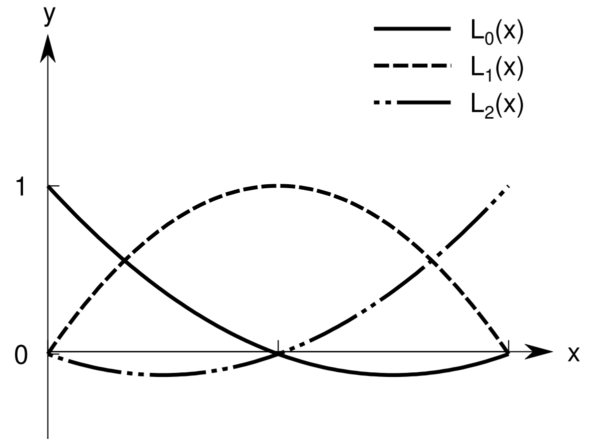
with

$$L_i\,(x_k) = 1 \text{ for } i = k \qquad \text{and} \qquad L_i\,(x_k) = 0 \text{ for } i \neq k.$$

Also we have the following property

$$\sum_{i=0}^{2} L_i(x) \quad = \quad 1 \quad \text{for all } x \in [a, b]$$

Figure 4.2: Example of the quadratic Lagrange interpolation functions



Here we see that p passes through the given three data points, i.e.,

$$p\left(x_0\right)=y_0, \quad p\left(x_1\right)=y_1 \quad \text{and} \quad p\left(x_2\right)=y_2$$

and

$$L_i(x) \;=\; \prod_{k=0,\,k\neq i}^{2} \frac{(x-x_k)}{(x_i-x_k)} \qquad\qquad i=0,1,2$$

Figure 4.2 shows the quadratic Lagrange interpolation functions $L_0(x)$, $L_1(x)$ and $L_2(x)$.

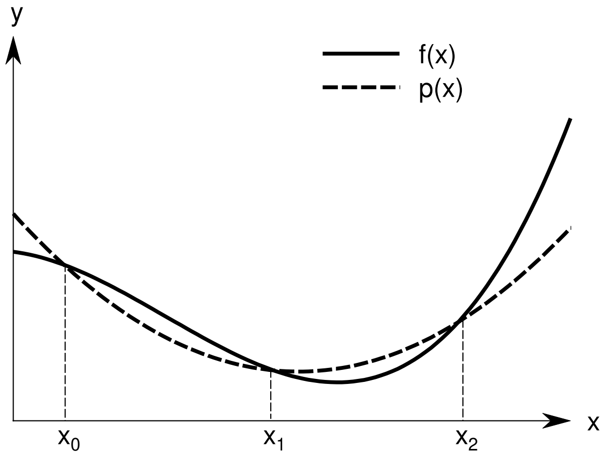
Finally, we have

$$p(x) \;=\; \sum_{i=0}^{2} y_i L_i(x)=\sum_{i=0}^{2} y_i \prod_{k=0,\,k\neq i}^{2} \frac{(x-x_k)}{(x_i-x_k)}$$

Figure 4.3 shows an example of quadratic interpolation polynomial $p(x)$ to approximate a function $f(x)$.

Now we generalize the above results to $n+1$ points $\left(x_0,y_0\right),\left(x_1,y_1\right),..........$, $\left(x_n,y_n\right)$ such that $y_i=f\left(x_i\right),\ i=0,1,....,n$. The Lagrange interpolation

Figure 4.3: Example of quadratic interpolation



functions $L_i(x)$ in this case can be written as

$$\begin{aligned} L_i(x) &= \frac{(x-x_0)\,(x-x_1)\,.........\,(x-x_{i-1})\,(x-x_{i+1})\,.......\,(x-x_n)}{(x_i-x_0)\,(x_i-x_1)\,.........\,(x_i-x_{i-1})\,(x_i-x_{i+1})\,.......\,(x_i-x_n)} \\ &= \prod_{k=0,\,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)} \end{aligned}$$

Also we have the following property

$$\sum_{i=0}^{n} L_i(x) = 1 \quad \text{for all } x \in [a,b]$$

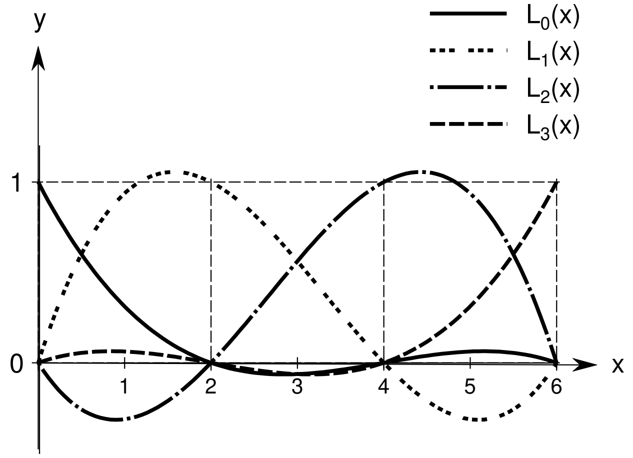
with

$$L_i\,(x_k) = 1 \text{ for } i = k \qquad \text{and} \qquad L_i\,(x_k) = 0 \text{ for } i \neq k.$$

The $(n+1)$ degree polynomial $p(x)$ which approximates $f(x)$ is given by

$$p(x) = \sum_{i=0}^{n} \left[y_i \prod_{k=0,\,k\neq i}^{n} \frac{(x-x_k)}{(x_i-x_k)} \right] = \sum_{i=0}^{n} y_i L_i(x)$$

Figure 4.4: Example of Lagrange interpolation functions



Example 1

Find the Lagrange interpolation functions for the data points $(0,-3)$, $(2,2)$ $(4,-1)$ and $(6,3)$. Then write the interpolating polynomial $p(x)$ and evaluate $p(3)$.

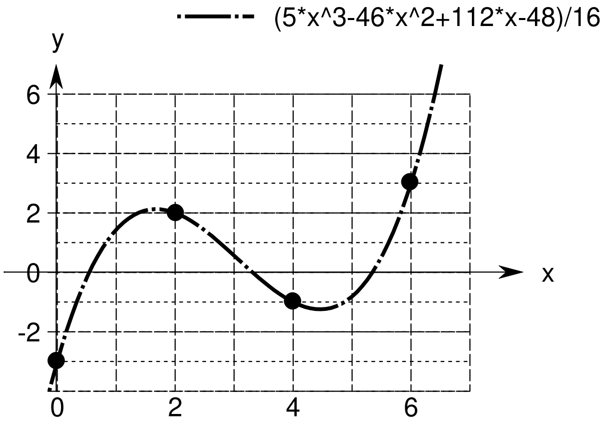
Here we have $x_0 = 0$, $x_1 = 2$, $x_2 = 4$, $x_3 = 6$ yielding the Lagrange interpolation functions as

$$\begin{aligned}
L_0(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} = \frac{(x-2)(x-4)(x-6)}{(-2)(-4)(-6)} = -\frac{x^3-12x^2+44x-48}{48} \\
L_1(x) &= \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} = \frac{(x-0)(x-4)(x-6)}{(2)(-2)(-4)} = \frac{x^3-10x^2+24x}{16} \\
L_2(x) &= \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} = \frac{(x-0)(x-2)(x-6)}{(4)(2)(-2)} = -\frac{x^3-8x^2+12x}{16} \\
L_3(x) &= \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} = \frac{(x-0)(x-2)(x-4)}{(6)(4)(2)} = \frac{x^3-6x^2+8x}{48}
\end{aligned}$$

Figure 4.4 shows four Lagrange interpolation functions used to interpolate data for four given points.

Interpolating third-degree polynomial is given by

Figure 4.5: Example of Lagrange interpolation given four data points



$$
\begin{aligned}
p(x) &= L_0(x)y_0 + L_1(x)y_1 + L_2(x)y_2 + L_3(x)y_3 \\
&= \frac{x^3 - 12x^2 + 44x - 48}{16} + \frac{2x^3 - 20x^2 + 48x}{16} + \frac{x^3 - 8x^2 + 12x}{16} + \frac{x^3 - 6x^2 + 8x}{16} \\
&= \frac{1}{16}\left(5x^3 - 46x^2 + 112x - 48\right)
\end{aligned}
$$

and

$$
\begin{aligned}
p(3) &= \frac{1}{16}\left(5 \times 3^3 - 46 \times 3^2 + 112 \times 3 - 48\right) \\
&= \frac{9}{16} = 0.563
\end{aligned}
$$

Also as we expected

$$p(0) = -3,\ p(2) = 2,\ p(4) = -1 \quad \text{and} \quad p(6) = 3.$$

Figure 4.5 shows an example of cubic interpolation polynomial passing through four given points.

Limitations

Some of the limitations of the Lagrange interpolation polynomial are

- Change in the number of data points requires new derivation of the polynomial
- Computational effort is more for a single interpolation
- To interpolate at another point, computational results at a previous point are not used so no saving in computational effort.
- Error estimation is not easy.
- Other interpolation methods are less computationally complex.

4.2 Newton's Divided Difference

Newton's divided difference presents a simple way to obtain the interpolating polynomials. This method overcomes the limitations of the Lagrange interpolation method. First we derive the lower order interpolations formulas. Then we proceed to derive the general nth-order interpolation formula.

4.2.1 Linear Interpolation

Let us consider two data points $(x_0, f(x_0))$ and $(x_1, f(x_1))$. Considering a linear approximation

$$p(x) = a_0 + a_1 (x - x_0)$$

Substituting x_0 and x_1 for x respectively, we have

$$a_0 = p(x_0) = f(x_0) = f[x_0]$$

and $f(x_1) = p(x_1) = f(x_0) + a_1 (x_1 - x_0)$

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = f[x_0, x_1]$$

Here $f[x_0]$ and $f[x_0, x_1]$ denote the zeroth divided difference and first divided difference respectively.

Thus we obtain

$$p(x) = f[x_0] + f[x_0, x_1](x - x_0)$$

4.2.2 Quadratic Interpolation

Let us consider three data points $(x_0, f(x_0))$, $(x_1, f(x_1))$ and $(x_2, f(x_2))$. Considering a quadratic polynomial

$$p(x) = a_0 + a_1 (x - x_0) + a_2 (x - x_0)(x - x_1)$$

Substituting x_0, x_1 and x_2 for x respectively, we have

$$a_0 = p(x_0) = f(x_0) = f[x_0]$$

and $f(x_1) = p(x_1) = f(x_0) + a_1(x_1 - x_0)$

$$a_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = f[x_0,\, x_1]$$

and finally $f(x_2) = p(x_2) = f(x_0) + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1)$

$$\begin{aligned} a_2 &= \frac{f(x_2) - f(x_0) - a_1(x_2 - x_0)}{(x_2 - x_0)(x_2 - x_1)} \\ &= \frac{1}{x_2 - x_0}\left\{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}\right\} \\ &= \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} \\ &= f[x_0,\, x_1,\, x_2] \end{aligned}$$

Here $f[x_0,\, x_1,\, x_2]$ represents the second divided difference.

Thus we obtain

$$p(x) = f[x_0] + f[x_0,\, x_1](x - x_0) + f[x_0,\, x_1,\, x_2](x - x_0)(x - x_1)$$

4.2.3 nth-order Interpolation

Now we consider $(n+1)$ data points $(x_0, f(x_0))\,,\ (x_1, f(x_1))\ ......\ (x_n, f(x_n))\,.$ Let us consider an nth-order polynomial

$$\begin{aligned} p(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + ... \\ &\quad ... + a_n(x - x_0)(x - x_1)\,.........\,(x - x_{n-1}) \end{aligned}$$

Proceeding as before, we obtain

$$\begin{aligned} p(x) &= f[x_0] + f[x_0,\, x_1](x - x_0) + f[x_0,\, x_1,\, x_2](x - x_0)(x - x_1) + ... \\ &\quad .... + f[x_0,\, x_1, ..., x_n](x - x_0)(x - x_1)\,.......\,(x - x_{n-1}) \\ &= \sum_{k=0}^{n}\left[f[x_0,\, x_1, ..., x_k]\left(\prod_{i=0}^{k-1}(x - x_i)\right)\right] \end{aligned}$$

Table 4.1: Divided Difference

x	Zeroth-order	First-order	Second-order	Third-order
x_0	$f\left[x_0\right]$	$f\left[x_0, x_1\right] = \frac{f[x_1]-f[x_0]}{x_1-x_0}$		
x_1	$f\left[x_1\right]$	$f\left[x_1, x_2\right] = \frac{f[x_2]-f[x_1]}{x_2-x_1}$	$f\left[x_0, x_1, x_2\right] = \frac{f[x_1,x_2]-f[x_0,x_1]}{x_2-x_0}$	
x_2	$f\left[x_2\right]$	$f\left[x_2, x_3\right] = \frac{f[x_3]-f[x_2]}{x_3-x_2}$	$f\left[x_1, x_2, x_3\right] = \frac{f[x_2,x_3]-f[x_1,x_2]}{x_3-x_1}$	$f\left[x_0, x_1, x_2, x_3\right] = \frac{f[x_1,x_2,x_3]-f[x_0,x_1,x_2]}{x_3-x_0}$
x_3	$f\left[x_3\right]$	$f\left[x_3, x_4\right] = \frac{f[x_4]-f[x_3]}{x_4-x_3}$	$f\left[x_2, x_3, x_4\right] = \frac{f[x_3,x_4]-f[x_2,x_3]}{x_4-x_2}$	$f\left[x_1, x_2, x_3, x_4\right] = \frac{f[x_2,x_3,x_4]-f[x_1,x_2,x_3]}{x_4-x_1}$
x_4	$f\left[x_4\right]$			

Example

Using the following four data points $(0,-3)$, $(2,2)$ $(4,-1)$ and $(6,3)$, and divided difference, find the interpolating polynomial $p(x)$ and evaluate $p(3)$.

Here we have $x_0 = 0$, $x_1 = 2$, $x_2 = 4$, $x_3 = 6$.

Thus the polynomial is given by

$$\begin{aligned} p(x) &= f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \\ &\quad + f[x_0, x_1, x_2, x_3](x - x_0)(x - x_1)(x - x_2) \\ &= -3 + \frac{5x}{2} - x(x-2) + \frac{15}{48}x(x-2)(x-4) \\ &= \frac{1}{48}\left(-144 + 120x - 48x^2 + 96x + 15x^3 - 90x^2 + 120x\right) \\ &= \frac{1}{48}\left(15x^3 - 138x^2 + 336x - 144\right) \\ &= \frac{1}{16}\left(5x^3 - 46x^2 + 112x - 48\right) \end{aligned}$$

It is observed that we obtain the same polynomial as in Lagrange interpolation. But here the computational effort is relatively less and easy to maintain.

Results

If f is continuous on $[x_0,\, x_1]$ and differentiable on $(x_0,\, x_1)$, then by Mean Value Theorem there exists a $\xi \in (x_0,\, x_1)$ such that

$$f'(\xi) = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

This yields that $f[x_0,\, x_1] = f'(\xi)$.

Theorem

Let $x_0, x_1, ..., x_n$ be $(n+1)$ distinct points in $[a,\, b]$ and $f \in C^n[a,\, b]$. Then there exists a $\xi \in (a,\, b)$ such that

$$f[x_0,\, x_1, ..., x_n] = \frac{f^{(n)}(\xi)}{n!}$$

Table 4.2: Example for Divided Difference

x	Zeroth-order	First-order	Second-order	Third-order
$x_0 = 0$	$f\left[x_0\right] = -3$			
		$f\left[x_0, x_1\right] = \frac{2+3}{2-0} = \frac{5}{2}$		
$x_1 = 2$	$f\left[x_1\right] = 2$		$f\left[x_0, x_1, x_2\right] = \frac{-4}{4} = -1$	
		$f\left[x_1, x_2\right] = \frac{-1-2}{4-2} = -\frac{3}{2}$		$f\left[x_0, x_1, x_2, x_3\right] = \frac{15}{48}$
$x_2 = 4$	$f\left[x_2\right] = -1$		$f\left[x_1, x_2, x_3\right] = \frac{7}{8}$	
		$f\left[x_2, x_3\right] = \frac{3+1}{6-4} = 2$		
$x_3 = 6$	$f\left[x_3\right] = 3$			

Proof

Let us define

$$g(x) = f(x) - p(x)$$

Since $f(x_i) = p(x_i)$, $i = 0, 1, ..., n$, we have

$$g(x_i) = 0 \qquad i = 0, 1, ..., n$$

i.e., the function g has $n+1$ distinct zeros in $[a, b]$. Applying the Generalized Rolle's Theorem , there exists a $\xi \in (a, b)$ such that $g^{(n)}(\xi) = 0$. [Since g vanishes at $n+1$ points, $x_0, x_1, ..., x_n$, By Rolle's Theorem, g' has n distinct zeros in $[a, b]$. Similarly, g'' has $n-1$ distinct zeros in $[a, b]$. So on, continuing we get $g^{(n)}$ has at least zero (say ξ) in $[a, b]$.]

This yields

$$p^{(n)}(\xi) = f^{(n)}(\xi)$$

Since $p(x)$ is a polynomial of degree n with leading coefficient $f[x_0, x_1, ..., x_n]$,we have

$$p^{(n)}(\xi) = n! \quad f[x_0, x_1, ..., x_n]$$

Hence we obtain

$$f[x_0, x_1, ..., x_n] = \frac{f^{(n)}(\xi)}{n!}$$

Divided Difference with Uniformly Spaced Data

Newton's divided difference formula can be obtained in much simpler form if x_0, x_1, ..., x_n are equally spaced. Let us define the uniform length as $h = x_{i+1} - x_i$ for $i = 0, 1, ..., n-1$ and let $x = x_0 + rh$. Then we can write

$$x - x_i = (r - i)h$$

Now the nth-order divided difference polynomial becomes

$$
\begin{aligned}
p(x) = p\left(x_0 + rh\right) &= f\left[x_0\right] + \sum_{k=1}^{n} f\left[x_0,\, x_1, ..., x_k\right]\left(x - x_0\right)\left(x - x_1\right) .... \left(x - x_{k-1}\right) \\
&= f\left[x_0\right] + \sum_{k=1}^{n} r(r-1)........(r-k+1)h^k\, f\left[x_0,\, x_1, ..., x_k\right] \\
&= f\left[x_0\right] + \sum_{k=1}^{n} \begin{pmatrix} r \\ k \end{pmatrix} k!h^k\, f\left[x_0,\, x_1, ..., x_k\right]
\end{aligned}
$$

where

$$
\begin{pmatrix} r \\ k \end{pmatrix} = \frac{r(r-1)........(r-k+1)}{k!} = \frac{r!}{k!(r-k)!}
$$

Newton's Forward Divided Difference Formula

Introducing the symbol $\triangle$ for forward difference, we can express

$$
\begin{aligned}
f\left[x_0,\, x_1\right] &= \frac{f\left(x_1\right) - f\left(x_0\right)}{x_1 - x_0} = \frac{\triangle f\left(x_0\right)}{h} \\
f\left[x_0,\, x_1,\, x_2\right] &= \frac{1}{x_2 - x_0}\left\{\frac{f\left(x_2\right) - f\left(x_1\right)}{h} - \frac{f\left(x_1\right) - f\left(x_0\right)}{h}\right\} \\
&= \frac{f\left(x_2\right) - 2f\left(x_1\right) + f\left(x_0\right)}{2h^2} = \frac{\triangle^2 f\left(x_0\right)}{2!h^2}
\end{aligned}
$$

and in general

$$
f\left[x_0,\, x_1, ..., x_k\right] = \frac{\triangle^k f\left(x_0\right)}{k!h^k}
$$

so that we have

$$
p(x) = p\left(x_0 + rh\right) = f\left(x_0\right) + \sum_{k=1}^{n} \begin{pmatrix} r \\ k \end{pmatrix} \triangle^k f\left(x_0\right)
$$

Table 4.3: Example for Forward Divided Difference

x	Zeroth-order	First-order	Second-order	Third-order
$x_0 = 0$	$f\left(x_0\right) = -3$			
		$\triangle f\left(x_0\right) = 5$		
$x_1 = 2$	$f\left(x_0\right) = 2$		$\triangle^2 f\left(x_0\right) = -8$	
		$\triangle f\left(x_1\right) = -3$		$\triangle^3 f\left(x_0\right) = 15$
$x_2 = 4$	$f\left(x_0\right) = -1$		$\triangle^2 f\left(x_1\right) = 7$	
		$\triangle f\left(x_2\right) = 4$		
$x_3 = 6$	$f\left(x_0\right) = 3$			

Example 2

Using the following four data points $(0,-3)$, $(2,2)$ $(4,-1)$ and $(6,3)$, and forward divided difference, evaluate $p(3)$.

Here $x=3=x_0+rh \Longrightarrow r=3/2$. Thus $p(3)$ is given by

$$\begin{aligned}
p(3) &= f\left(x_0\right)+\sum_{k=1}^{3}\left(\begin{array}{c} r \\ k \end{array}\right) \triangle^k f\left(x_0\right) \\
&= -3+\left(\frac{3}{2}\right)(5)+\left(\frac{3}{2}\right)\left(\frac{1}{2}\right)\left(\frac{1}{2!}\right)(-8)+\left(\frac{3}{2}\right)\left(\frac{1}{2}\right)\left(-\frac{1}{2}\right)\left(\frac{1}{3!}\right)(15) \\
&= -3+\frac{15}{2}-3-\frac{15}{16}=\frac{9}{16}
\end{aligned}$$

4.3 Interpolation Error

Theorem

Let x_0, $x_1, \ldots\ldots, x_n$ be $(n+1)$ distinct points in $[a,b]$ and $f \in C^{(n+1)}[a,b]$. Then for each $x \in [a,b]$, there exists a $\xi = \xi(x)$ in (a,b) such that

$$\begin{aligned} f(x) - p(x) &= \frac{f^{(n+1)}(\xi)}{(n+1)!}(x-x_0)(x-x_1)\ldots\ldots\ldots(x-x_n) \\ &= \frac{f^{(n+1)}(\xi)}{(n+1)!}\prod_{k=0}^{n}(x-x_k) \end{aligned}$$

Proof

Let us define for any $t \in [a,b]$, $x \neq x_k$

$$g(t) = f(t) - p(t) - U(t)\left[f(x) - p(x)\right]$$

with $U(t) = \frac{u(t)}{u(x)}$ where $u(t) = \prod_{k=0}^{n}(t-x_k)$. So $U(x_k) = 0$ and $g(x) = 0$.

We have $f(x_k) = p(x_k)$ for $k = 0, 1, 2, \quad, n$.

Since $g(x_k) = 0$ for $0 \leq k \leq n$ and $g(x) = 0$, this implies that there are $(n+2)$ points (at x_0, $x_1, \ldots\ldots, x_n$ and x) where g is zero. Using the generalized Rolle's theorem, there exists a $\xi \in [a,b]$ such that $g^{(n+1)}(\xi) = 0$. Now differentiations yield

$$g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - p^{(n+1)}(\xi) - U^{(n+1)}(\xi)\left[f(x) - p(x)\right]$$

Also p is a polynomial of degree at most n, so $p^{(n+1)}(\xi) = 0$. Thus we have

$$f^{(n+1)}(\xi) = U^{(n+1)}(\xi)\left[f(x) - p(x)\right]$$

Now

$$U^{(n+1)}(t) = \left(\frac{u(t)}{u(x)}\right)^{(n+1)} = \frac{(u(t))^{(n+1)}}{u(x)} = \frac{\left(\prod_{k=0}^{n}(t-x_k)\right)^{(n+1)}}{\prod_{k=0}^{n}(x-x_k)}$$

$u(t)$ is a polynomial of independent variable t of order $(n+1)$. So its $(n+1)$ ‘ derivative is $(n+1)!$.

Thus we have

$$f^{(n+1)}(\xi) = \frac{(n+1)!}{u(x)} [f(x) - p(x)]$$

which gives us

$$\begin{aligned} f(x) - p(x) &= \frac{f^{(n+1)}(\xi)}{(n+1)!} u(x) \\ &= \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{k=0}^{n} (x - x_k) \end{aligned}$$

Error from the Lagrange polynomial is similar to that for Taylor's polynomial

$$\frac{f^{(n+1)}(\xi)}{(n+1)!} (x - x_0)^{n+1}$$

Only difference is that Lagrange polynomial uses $(n+1)$ distinct points $x_0, ..., x_n$ to evaluate $\prod_{k=0}^{n} (x - x_k)$ instead of $(x - x_0)^{n+1}$.

4.4 Chebyshev Interpolation

Definition

For a given non-negative integer n, the Chebyshev Polynomial T_n is defined by

$$T_n(x) \;=\; \cos\left(n \cos^{-1} x\right)$$

for $x \in [-1, 1]$.

It can be shown that $T_n(x)$ is a polynomial of degree n. Also we will prove that

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \end{aligned}$$

and the recurrence relation

$$T_{n+1}(x) \;=\; 2xT_n(x) - T_{n-1}(x)$$

Using $n = 0$ and $n = 1$ in the definition respectively, we obtain

$$\begin{aligned} T_0(x) &= \cos\left(0.\cos^{-1} x\right) = \cos 0 = 1 \\ T_1(x) &= \cos\left(1.\cos^{-1} x\right) = x \end{aligned}$$

To prove the recurrence relation, we put $\theta = \cos^{-1} x$ so that we have $\cos\theta = x$. This yields $T_n(\cos\theta) = \cos n\theta$.

Thus we can write

$$T_{n+1}(\cos\theta) \;=\; \cos(n+1)\theta = \cos n\theta \cos\theta - \sin n\theta \sin\theta$$

and

$$T_{n-1}(\cos\theta) \;=\; \cos(n-1)\theta = \cos n\theta \cos\theta + \sin n\theta \sin\theta$$

Adding the last two equations we obtain

$$\begin{aligned} T_{n+1}(\cos\theta) + T_{n-1}(\cos\theta) &= 2\cos n\theta\cos\theta \\ T_{n+1}(x) + T_{n-1}(x) &= 2xT_n(x) \qquad \text{as } \cos\theta = x \text{ and } \cos n\theta = T_n(\cos\theta) \end{aligned}$$

which yields

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Thus, $T_2(x) = 2xT_1(x) - T_0(x) = 2x^2 - 1$, $T_3(x) = 2xT_2(x) - T_1(x) = 4x^3 - 3x$ and so on.

The first seven polynomials are given below

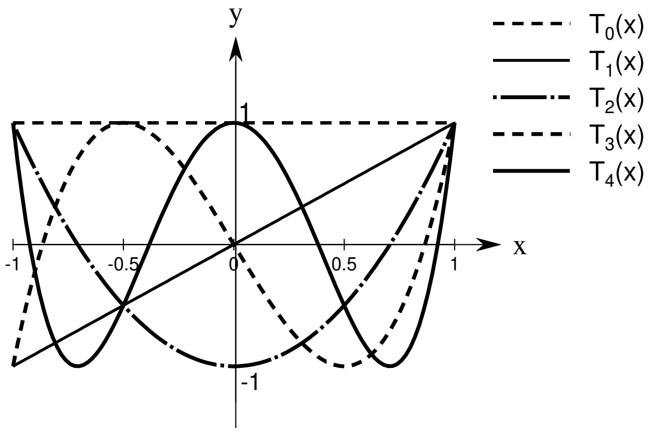
$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \\ T_5(x) &= 16x^5 - 20x^3 + 5x \\ T_6(x) &= 32x^6 - 48x^4 + 18x^2 - 1 \\ T_7(x) &= 64x^7 - 112x^5 + 56x^3 - 7x \end{aligned}$$

Some Properties of Chebyshev's Polynomial

- $T_n(x)$ is odd if n is odd and it is even if n is even.
- $T_n(0) = (-1)^{n/2}$ if n is even and $T_n(0) = 0$ if n is odd.

Chebyshev's polynomials can be used to express a polynomial. Various powers of x in terms of Chebyshev's polynomials can be obtained as

Figure 4.6: First Few Chebyshev's Polynomials



$$
\begin{aligned}
1 &= T_0(x) \\
x &= T_1(x) \\
x^2 &= \left[T_0(x) + T_2(x)\right]/2 \\
x^3 &= \left[3T_1(x) + T_3(x)\right]/4 \\
x^4 &= \left[3T_0(x) + 4T_2(x) + T_4(x)\right]/8 \\
x^5 &= \left[10T_1(x) + 5T_3(x) + T_5(x)\right]/16 \\
x^6 &= \left[10T_0(x) + 15T_2(x) + 6T_4(x) + T_6(x)\right]/32 \\
x^7 &= \left[35T_1(x) + 21T_3(x) + 7T_5(x) + T_7(x)\right]/64
\end{aligned}
$$

Example

Consider the function $f(x) = 16x^5 - 3x^4 + 5x^2 - 7x + 2$. Express it in terms of Chebyshev's polynomials.

Substituting various powers of x, we obtain

$$
\begin{aligned}
f(x) &= 10T_1(x) + 5T_3(x) + T_5(x) - \frac{9T_0(x) + 12T_2(x) + 3T_4(x)}{8} \\
&\quad + \frac{5T_0(x) + 5T_2(x)}{2} - 7T_1(x) + 2T_0(x) \\
&= T_5(x) - \frac{3}{8}T_4(x) + 5T_3(x) + T_2(x) + 3T_1(x) + \frac{27}{8}T_0(x)
\end{aligned}
$$

Property

Show that Chebyshev's polynomial $y = T_n(x)$ satisfies the following differential equation

$$\left(1 - x^2\right)\frac{d^2y}{dx^2} - x\frac{dy}{dx} + n^2y \quad = \quad 0$$

Proof

Here $y = T_n(x) = \cos(n\cos^{-1}x)$. Let $\theta = \cos^{-1}x$, then $\cos\theta = x$. This yields $y = T_n(x) = \cos n\theta$.

So we have

$$y' = \frac{dy}{dx} \quad = \quad \frac{dy}{d\theta}/\frac{dx}{d\theta} = (-n\sin n\theta)/(-\sin\theta) = \frac{n\sin n\theta}{\sin\theta}$$

Differential wrt x once again

$$\begin{aligned} \frac{d^2y}{dx^2} &= \frac{d}{dx}\left(\frac{dy}{dx}\right) = \frac{d}{d\theta}\left(\frac{dy}{dx}\right)/\left(\frac{dx}{d\theta}\right) \\ &= \frac{\frac{n^2\cos n\theta\sin\theta - n\sin n\theta\cos\theta}{\sin^2\theta}}{-\sin\theta} \\ &= \frac{-n^2\cos n\theta + \frac{n\sin n\theta}{\sin\theta}\cos\theta}{\sin^2\theta} \\ &= -\frac{n^2y}{1-x^2} + \frac{xy'}{1-x^2} \end{aligned}$$

Hence

$$\left(1 - x^2\right)\frac{d^2y}{dx^2} - x\frac{dy}{dx} + n^2y \quad = \quad 0$$

Property

Show that Chebyshev's polynomial $y = T_n(x)$ satisfies the orthogonal property under integration over $[-1, 1]$ with respect to the weight function $w(x) = \frac{1}{\sqrt{1-x^2}}$. That is, show that

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

Proof

Here $T_n(x) = \cos(n \cos^{-1} x)$. Let $\theta = \cos^{-1} x$, then $\cos\theta = x$. This yields $T_n(x) = \cos n\theta$. Also

$$\cos m\theta \cos n\theta \;=\; \frac{1}{2}\left[\cos(m+n)\theta + \cos(m-n)\theta\right]$$

Now show the result.

Chebyshev Expansion

Now we will approximate a function $f(x)$ using the zeros of $T_n(x)$. Let $x_0, x_1, ..., x_{n-1}$ be n zeros. Then

$$f(x) \;\approx\; \sum_{k=0}^{n-1} C_k T_k(x)$$

Using orthogonal property,

$$C_k \;=\; \begin{cases} \frac{1}{n}\sum_{i=0}^{n-1} f(x_i) T_0(x_i) & k = 0 \\ \\ \frac{2}{n}\sum_{i=0}^{n-1} f(x_i) T_k(x_i) & k > 0 \end{cases}$$

Example

Find a polynomial of degree two using the zeros of $T_3(x)$ for the function $f(x) = e^x$ on $[-1,\, 1]$.

Here we use $T_3(x) = 4x^3 - 3x$. Zeros of T_3 are $x_0 = -\frac{\sqrt{3}}{2}$, $x_1 = 0$ and $x_2 = \frac{\sqrt{3}}{2}$. Now

$$\begin{aligned} e^x &\approx \sum_{k=0}^{2} C_k T_k(x) = C_0 T_0(x) + C_1 T_1(x) + C_2 T_2(x) \\ &= C_0 + C_1 x + C_2\left(2x^2 - 1\right)\big) \end{aligned}$$

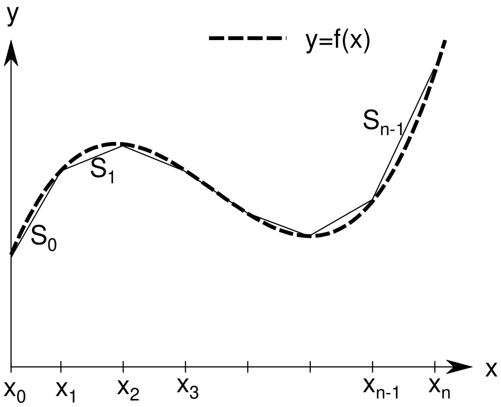
where

$$\begin{aligned}
C_0 &= \frac{1}{3}\sum_{k=0}^{2} e^{x_k} T_0(x_k) = \frac{1}{3}\left[e^{-\sqrt{3}/2} + 1 + e^{\sqrt{3}/2}\right] \approx 0.8982 \\
C_1 &= \frac{2}{3}\sum_{k=0}^{2} e^{x_k} T_1(x_k) = \frac{2}{3}\left[e^{-\sqrt{3}/2}\left(-\frac{\sqrt{3}}{2}\right) + 0 + e^{\sqrt{3}/2}\left(\frac{\sqrt{3}}{2}\right)\right] \approx 1.1298 \\
C_2 &= \frac{2}{3}\sum_{k=0}^{2} e^{x_k} T_2(x_k) = \frac{2}{3}\left[e^{-\sqrt{3}/2}\left(2 \times \frac{3}{4} - 1\right) - 1 + e^{\sqrt{3}/2}\left(2 \times \frac{3}{4} - 1\right)\right] \approx 0.2660
\end{aligned}$$

Hence

$$\begin{aligned}
e^x &\approx C_0T_0(x) + C_1T_1(x) + C_2T_2(x) \\
&= 0.8982 + 1.1298x + 0.2660\left(2x^2 - 1)\right) \\
&= 0.5320x^2 + 1.1298x + 0.6322
\end{aligned}$$

Figure 4.7: Piecewise linear interpolation of $f(x)$



4.5 Cubic Spline

Piecewise Polynomial Approximations

Consider a function $y = f(x)$ defined in $[a, b]$ with $(n+1)$ points $a = x_0 < x_1 < .... < x_n = b$. If we approximate f by some polynomial in each sub interval $[x_i, x_{i+1}]$ for $i = 0, 1, ..., n-1$, then this type of approximation is called piecewise polynomial approximation. Here $(n+1)$ data points, $(x_i,\, y_i = f\,(x_i))$, are called knots. These piecewise polynomials can be linear, quadratic, cubic and so on. Figure 4.7 shows piecewise linear interpolation of a function.

Linear Spline

The linear spline approximates a function by line segments drawn between two consecutive knots. Considering two neighboring knots $(x_i,\, y_i)$ and $(x_{i+1},\, y_{i+1})$, we obtain the line joining these two knots for $x_i \leq x \leq x_{i+1}$ as

$$S(x) = S_i(x) \quad = \quad y_i + \frac{y_{i+1} - y_i}{x_{i+1} - x_i}\,(x - x_i) \qquad i = 0, 1, ..., n-1 \quad (4.1)$$

Here $S(x)$ denotes the approximation. This can be expressed as

$$S(x) = \left\{ \begin{array}{cc} S_0(x) = y_0 + \frac{y_1-y_0}{x_1-x_0}(x-x_0) & x \in [x_0, x_1] \\ S_1(x) = y_1 + \frac{y_2-y_1}{x_2-x_1}(x-x_1) & x \in [x_1, x_2] \\ \vdots & \\ S_{n-1}(x) = y_{n-1} + \frac{y_n-y_{n-1}}{x_n-x_{n-1}}(x-x_{n-1}) & x \in [x_{n-1}, x_n] \end{array} \right\} \quad (4.2)$$

The resulting interpolant is obviously continuous, but not smooth (differentiable). It is nice to have a smooth interpolant.

Cubic Spline

Piecewise polynomial of low order is called a spline. Although linear and quadratic splines are used, most commonly used one is cubic spline. Now we introduce cubic spline interpolant.

A **Cubic Spline Interpolant** of $f(x)$ is a function $S(x)$ that satisfies the following

- $S(x)$ is a cubic polynomial $S_i(x)$ on the sub interval $[x_i, x_{i+1}]$, $i = 0, 1, ..., n-1$, i.e., $S(x)$ takes the form

  $$S(x) = S_i(x) \quad = \quad a_i + b_i(x-x_i) + c_i(x-x_i)^2 + d_i(x-x_i)^3 \quad (4.3)$$

  i.e.,

  $$S(x) = \left\{ \begin{array}{cc} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ & \vdots \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{array} \right\}$$

- S interpolates f at $x_0, x_1, ..., x_n$, i.e., $S_i(x_i) = f(x_i)$ and $S_i(x_{i+1}) = f(x_{i+1})$, $i = 0, 1, 2, ..., n-1$.

- S is continuous on $[a, b]$, i.e., $S_i(x_i) = S_{i-1}(x_i)$, $i = 1, 2, ..., n-1$.

- S' is continuous on $[a, b]$, i.e., $S_i'(x_i) = S_{i-1}'(x_i)$, $i = 1, 2, ..., n-1$.

- S'' is continuous on $[a, b]$, i.e., $S_i''(x_i) = S_{i-1}''(x_i)$, $i = 1, 2, ..., n-1$.

In one cubic polynomial there are 4 unknown coefficients a_i, b_i, c_i and d_i; and there are n piecewise cubic polynomials. Thus total number of unknowns are $4n$. Since S, S', S'' are continuous at the $(n-1)$ interior knots, we have $3(n-1)$ equations. Also we know that S interpolates f at the $(n+1)$ points $x_0, x_1, ..., x_n$, we obtain $n+1$equations. This information leads to the following

- Total number of unknowns $= 4n$
- Total number of equations $= 3(n-1) + n + 1 = 4n - 2$
- To determine the interpolating functions, we need 2 more equations.

Three different types of constraint or boundary conditions are used to obtain two more equations. These are

1. Natural or free boundary conditions: $S''(x_0) = 0,\ S''(x_n) = 0;$
2. Clamped or complete boundary conditions: $S'(x_0) = f'(x_0),\ S'(x_n) = f'(x_n);$
3. Not-a-knot boundary conditions: S''' is continuous at $x = x_1$ and $x = x_{n-1}$.

Natural boundary conditions are equivalent to $S_0''(x_0) = 0,\ S_{n-1}''(x_n) = 0$. Clamped conditions yield $S_0'(x_0) = f'(x_0),\ S_{n-1}'(x_n) = f'(x_n)$. Not-a-knot boundary conditions become $S_0'''(x_1) = S_1'''(x_1)$ and $S_{n-2}'''(x_{n-1}) = S_{n-1}'''(x_{n-1})$.

Determination of the unknown coefficients

Let us define h_i as the length of the sub interval $[x_i, x_{i+1}]$, i.e., $h_i = x_{i+1} - x_i,\ i = 0, 1, ..., n-1$. The unknowns a_i are determined from

$$S_i(x_i) = a_i = f(x_i), \quad i = 0, 1, ..., n \tag{4.4}$$

i.e., a_i's are known now.
Continuity of the spline yields

$$a_{i+1} = a_i + b_i h_i + c_i h_i^2 + d_i h_i^3, \ \ i = 0, 1, ..., n-2 \tag{4.5}$$

Continuity of the spline derivative yields

$$b_{i+1} = b_i + 2c_ih_i + 3d_ih_i^2, \quad i = 0, 1, ..., n-2 \tag{4.6}$$

Continuity of the spline second derivative yields

$$c_{i+1} = c_i + 3d_ih_i, \quad i = 0, 1, ..., n-2 \tag{4.7}$$

From the condition (4.7) we can express d_i in terms of c_i as follows

$$d_i = \frac{c_{i+1} - c_i}{3h_i}$$

From the condition (4.6) we can write

$$\begin{aligned} b_{i+1} &= b_i + 2c_ih_i + 3h_i^2\frac{c_{i+1} - c_i}{3h_i} \\ &= b_i + h_i\left(c_{i+1} + c_i\right) \end{aligned} \tag{4.8}$$

From the continuity condition (4.5) we obtain

$$\begin{aligned} a_{i+1} &= a_i + b_ih_i + c_ih_i^2 + d_ih_i^3 \\ &= a_i + b_ih_i + \frac{h_i^2}{3}\left(2c_i + c_{i+1}\right) \end{aligned} \tag{4.9}$$

Solving the equation (4.9) for b_i we get

$$b_i = \frac{a_{i+1} - a_i}{h_i} - \frac{h_i}{3}\left(2c_i + c_{i+1}\right) \tag{4.10}$$

This allows us to write

$$b_{i-1} = \frac{a_i - a_{i-1}}{h_{i-1}} - \frac{h_{i-1}}{3}\left(2c_{i-1} + c_i\right) \tag{4.11}$$

Reducing the index by one and using the equation (4.8), we obtain

$$h_{i-1}c_{i-1} + 2\left(h_{i-1} + h_i\right)c_i + h_ic_{i+1} = 3\left[\frac{a_{i+1} - a_i}{h_i} - \frac{a_i - a_{i-1}}{h_{i-1}}\right] \tag{4.12}$$

where $i = 1, 2, ..., n-1$.Since a_i and h_i's are known, we solve the equation (4.12) for c_i which forms a tridiagonal system. The equations for $i = 0$ and $i = n$ depend on the boundary conditions.

Cubic Spline with Natural Boundary Conditions

Here we have $S''(x_0) = 0,\ S''(x_n) = 0$. The first condition yields

$$2c_0 + 6d_0(x_0 - x_0) \quad = \quad 0$$

i.e., $c_0 = 0$. The second condition gives us $c_n = 0$. Thus the equations given by (4.12) can be expressed as a linear system

$$A\mathbf{c} \quad = \quad \mathbf{r} \tag{4.13}$$

where

$$A \quad = \left[\begin{array}{cccccccc} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \cdots & 0 & 0 & \\ \vdots & & & & \vdots & & \vdots & 0 \\ & & & & \cdots & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{array} \right]$$

and $\mathbf{r}$ and $\mathbf{c}$ are given by

$$\mathbf{r} = 3 \left[\begin{array}{c} 0 \\ \frac{a_2 - a_1}{h_1} - \frac{a_1 - a_0}{h_0} \\ \frac{a_3 - a_2}{h_2} - \frac{a_2 - a_1}{h_1} \\ \vdots \\ \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{a_{n-1} - a_{n-2}}{h_{n-2}} \\ 0 \end{array} \right] \qquad \text{and} \qquad \mathbf{c} = \left[\begin{array}{c} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{array} \right]$$

System (4.13) is equivalent to solving

$$\left[\begin{array}{cccccc} 2(h_0 + h_1) & h_1 & 0 & \cdots & 0 & 0 \\ h_1 & 2(h_1 + h_2) & h_2 & \cdots & 0 & 0 \\ & & & \vdots & & \\ & & & \cdots & h_{n-2} & 2(h_{n-2} + h_{n-1}) \end{array} \right] \left[\begin{array}{c} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \end{array} \right]$$

$$= 3 \left[\begin{array}{c} \frac{a_2 - a_1}{h_1} - \frac{a_1 - a_0}{h_0} \\ \frac{a_3 - a_2}{h_2} - \frac{a_2 - a_1}{h_1} \\ \vdots \\ \frac{a_n - a_{n-1}}{h_{n-1}} - \frac{a_{n-1} - a_{n-2}}{h_{n-2}} \end{array} \right]$$

with $c_0 = 0 = c_n$.

Example 1

Find the natural cubic spline through $(1, 2)$, $(2, -3)$ and $(3, 4)$.

Here $x_0 = 1,\ x_1 = 2,\ x_2 = 3$ and $y_0 = 2 = a_0,\ y_1 = -3 = a_1$ and $y_2 = 4 = a_2$. Also $h_0 = 1 = h_1$

System becomes

$$\begin{bmatrix} 1 & 0 & 0 \\ 1 & 4 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = 3 \begin{bmatrix} 0 \\ 7-(-5) \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 36 \\ 0 \end{bmatrix}$$

yielding $c_0 = 0$, $c_1 = 9$, $c_2 = 0$. Also $d_0 = \frac{c_1 - c_0}{3h_0} = 3$, $d_1 = \frac{c_2 - c_1}{3h_1} = -3$. Finally

$b_0 = \frac{a_1 - a_0}{h_0} - \frac{h_0}{3}(2c_0 + c_1) = -5 - 3 = -8$ and $b_1 = \frac{a_2 - a_1}{h_1} - \frac{h_1}{3}(2c_1 + c_2) = 7 - 6 = 1$

Thus

$$S(x) = S_i(x) = a_i + b_i (x - x_i) + c_i (x - x_i)^2 + d_i (x - x_i)^3$$

which gives us

$$\begin{aligned} S_0(x) &= a_0 + b_0 (x - x_0) + c_0 (x - x_0)^2 + d_0 (x - x_0)^3 \\ &= 2 - 8(x-1) + 3(x-1)^3 \qquad \text{on} \quad [1, 2] \end{aligned}$$

and

$$\begin{aligned} S_1(x) &= a_1 + b_1 (x - x_1) + c_1 (x - x_0)^2 + d_1 (x - x_1)^3 \\ &= -3 + (x-2) + 9(x-2)^2 - 3(x-2)^3 \qquad \text{on} \quad [2, 3] \end{aligned}$$

Exercise

1. Find a polynomial using Lagrange interpolation that passes through $(0, -2)$, $(2, 1)$, $(4, 4)$.

2. Consider $f(x) = \sqrt{1+x}$ and $x_0 = 0$, $x_1 = 0.6$, $x_2 = 0.9$.Construct interpolation polynomials of degree at most one and at most two to approximate $f(0.45)$ and find the absolute error.

3. Using divided difference (and making a table) find the interpolating polynomial that passes through $(0,1)\,,(2,2)$, $(3,4)$.

4. Express the function $f(x)=4-7x+2x^2+5x^3$ in terms of Chebyshev polynomials.

5. Find the natural $(S''(x_0)=0,\ S''(x_n)=0)$ cubic spline through (0, 3), $(1,-2)$ and (2, 1).

Chapter 5

Numerical Differentiation

5.1 Finite Difference Formulas

Approximation of the First Order Derivative

The derivative of a function $f(x)$ at x^* is defined as

$$f'(x^*) = \lim_{h\to 0} \frac{f(x^*+h) - f(x^*)}{h} \tag{5.1}$$

or,

$$f'(x^*) = \lim_{x\to x^*} \frac{f(x) - f(x^*)}{x - x^*}$$

If f is twice continuously differentiable, Taylor's Theorem gives us

$$f(x^*+h) = f(x^*) + hf'(x^*) + \frac{h^2}{2} f''(\xi) \tag{5.2}$$

where $x^* < \xi < x^* + h$. Equation (5.2) allows us to write

$$f'(x^*) = \frac{f(x^*+h) - f(x^*)}{h} - \frac{h}{2} f''(\xi) \tag{5.3}$$

Forward Difference Formula

From (5.3), we can approximate $f'(x^*)$ as

$$f'(x^*) \approx \frac{f(x^*+h)-f(x^*)}{h} \tag{5.4}$$

by considering $\frac{h}{2}f''(\xi)$ as the error term which of order h. This is known as the forward difference formula.

Backward Difference Formula

Taylor's Theorem gives us

$$f(x^*-h) = f(x^*)-hf'(x^*)+\frac{h^2}{2}f''(\xi) \tag{5.5}$$

where $x^*-h<\xi<x^*$. Thus we have

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

To approximate $f'(x^*)$, backward difference formula can be obtained as

$$f'(x^*) \approx \frac{f(x^*)-f(x^*-h)}{h} \tag{5.6}$$

Here the error is $O(h)$.

Central Difference Formula

Using Taylor's Theorem we can write

$$f(x^*+h) = f(x^*)+hf'(x^*)+\frac{h^2}{2}f''(x^*)+\frac{h^3}{6}f'''(\xi_1)$$

and

$$f(x^*-h) = f(x^*)-hf'(x^*)+\frac{h^2}{2}f''(x^*)-\frac{h^3}{6}f'''(\xi_2)$$

Subtracting and dividing by two we get

$$\frac{f(x^*+h)-f(x^*-h)}{2} = hf'(x^*)+\frac{h^3}{12}\left[f'''(\xi_1)+f'''(\xi_2)\right] \quad (5.7)$$

Using the Intermediate Value Theorem for some ξ between ξ_1 and ξ_2, we have $f'''(\xi)=\frac{f'''(\xi_1)+f'''(\xi_2)}{2}$. Result (5.7) yields

$$f'(x^*) = \frac{f(x^*+h)-f(x^*-h)}{2h}-\frac{h^2}{6}f'''(\xi)$$

Ignoring the last (error) term, the approximation is given by

$$f'(x^*) \approx \frac{f(x^*+h)-f(x^*-h)}{2h} \quad (5.8)$$

This is known as the **central difference formula** to approximate first order derivative of f. Here the error is $O(h^2)$. In general, central difference approximation is superior to forward or backward difference approximations. But sometime central difference can not be used. For example, if x^* is the first point there is no point to the left of x^* which is used in the central difference formula.

Approximations of the Higher Order Derivatives

Approximations for the second order derivative is obtained using the following Taylor's expansions

$$\begin{aligned} f(x^*+h) &= f(x^*)+hf'(x^*)+\frac{h^2}{2}f''(x^*)+\frac{h^3}{6}f'''(x^*)+\frac{h^4}{24}f^{(4)}(\xi_1) \\ f(x^*-h) &= f(x^*)-hf'(x^*)+\frac{h^2}{2}f''(x^*)-\frac{h^3}{6}f'''(x^*)+\frac{h^4}{24}f^{(4)}(\xi_2) \end{aligned}$$

where $x^*-h<\xi_2<x^*<\xi_1<x^*+h$. Adding these two equations, we find $f''(x^*)$ as

$$f''(x^*) = \frac{f(x^*+h)-2f(x^*)+f(x^*-h)}{h^2}-\frac{h^2}{24}\left[f^{(4)}(\xi_1)+f^{(4)}(\xi_2)\right] \quad (5.9)$$

Suppose that $f^{(4)}$ is continuous on $[x^* - h,\, x^* + h]$, there exists a number ξ between ξ_1 and ξ_2 and hence in $(x^* - h,\, x^* + h)$ such that

$$f^{(4)}(\xi) \;=\; \frac{f^{(4)}(\xi_1) + f^{(4)}(\xi_2)}{2}$$

Thus (5.9) becomes

$$f''(x^*) \;=\; \frac{f(x^* + h) - 2f(x^*) + f(x^* - h)}{h^2} - \frac{h^2}{12} f^{(4)}(\xi) \qquad (5.10)$$

This approximation is of order $O(h^2)$. Thus the second derivative is approximated as

$$f''(x^*) \;\approx\; \frac{f(x^* + h) - 2f(x^*) + f(x^* - h)}{h^2} \qquad (5.11)$$

5.2 Differentiation Using Interpolating Polynomials

Polynomial Interpolation can also used to differentiate a function. If we have $(n+1)$ points $(x_0, f(x_0)), (x_1, f(x_1))$....... $(x_n, f(x_n))$. The Lagrange interpolation functions $L_i(x),\;\; i = 0, 1, ..., n,$ in this case can be written as

$$\begin{aligned} L_i(x) \;&=\; \frac{(x - x_0)(x - x_1) \ldots\ldots\ldots (x - x_{i-1})(x - x_{i+1}) \ldots\ldots. (x - x_n)}{(x_i - x_0)(x_i - x_1) \ldots\ldots\ldots (x_i - x_{i-1})(x_i - x_{i+1}) \ldots\ldots. (x_i - x_n)} \\ &=\; \prod_{k=0,\, k \neq i}^{n} \frac{(x - x_k)}{(x_i - x_k)} \end{aligned} \qquad (5.12)$$

with the following properties

$$\sum_{i=0}^{n} L_i(x) \;=\; 1 \quad \text{for all } x \in [a, b]$$

and

$$L_i(x_k) = 1 \text{ for } i = k \qquad \text{and} \qquad L_i(x_k) = 0 \text{ for } i \neq k.$$

The n degree polynomial $p(x)$ which approximates $f(x)$ is given by

$$p(x) = \sum_{i=0}^{n} f(x_i) L_i(x) \tag{5.13}$$

and including the error term we can write $f(x)$ which was shown in a previous theorem as

$$\begin{aligned} f(x) &= p(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^{n} (x - x_k) \\ &= \sum_{i=0}^{n} f(x_i) L_i(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} u(x) \end{aligned} \tag{5.14}$$

where $\xi_x = \xi(x)$ and $u(x) = \prod_{k=0}^{n} (x - x_k)$. Differentiating (5.14), we obtain

$$f'(x) = \sum_{i=0}^{n} f(x_i) L_i'(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} u'(x) + \frac{u(x)}{(n+1)!} \frac{d}{dx} f^{(n+1)}(\xi_x)$$

At $x = x_j$, $u(x_j) = 0$, thus the above differentiation becomes

$$f'(x_j) = \sum_{i=0}^{n} f(x_i) L_i'(x_j) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} u'(x_j)$$

which includes the error term. In this case, differentiation formula becomes

$$f'(x) \approx \sum_{i=0}^{n} f(x_i) L_i'(x) \tag{5.15}$$

5.2.1 Errors

Following table (Table 5.1) presents the results of Forward Difference (FD) and Central Difference (CD) with their errors. We took $f(x) = e^x$ and $x = 1.2$.

Table 5.1: Error in Numerical Differentiation

h	Forward Difference (FD)	|Error in FD|	Central Difference (CD)	|Error in CD|
1×10^{-1}	3.491797448826972	0.171680526090425	3.325653218364058	0.005536295627510
1×10^{-2}	3.336772981247637	0.016656058511090	3.320172258295262	0.000055335558714
1×10^{-3}	3.321777534688763	0.001660611952215	3.320117476088845	0.000000553352298
1×10^{-4}	3.320282934118345	0.000166011381798	3.320116928271498	0.000000005534951
1×10^{-5}	3.320133523398992	0.000016600662445	3.320116922811421	0.000000000074874
1×10^{-6}	3.320118582728070	0.000001659991522	3.320116922500559	0.000000000235989
1×10^{-7}	3.320117092364681	0.000000169628134	3.320116925831227	0.000000003094680
1×10^{-8}	3.320116936933458	0.000000014196910	3.320116914728997	0.000000008007550
1×10^{-9}	3.320117514249430	0.000000591512883	3.320117292204825	0.000000369468278
1×10^{-10}	3.320117514249430	0.000000591512883	3.320117514249430	0.000000591512883
1×10^{-11}	3.320144159602021	0.000027236865474	3.320121955141529	0.000005032404982
1×10^{-12}	3.320455022048916	0.000338099312369	3.320232977443991	0.000116054707444

5.3 Numerical Differentiation of Partial Derivatives

Finite difference approach is used to differentiate a function $f(x)$ of a single independent variable x. For a function of two or more independent variables, we can also use finite difference for partial derivatives. Here we consider a function $f(x, y)$ of two independent variables x and y. Similar procedure can be used for three or more independent variables.

Difference Formulas for Partial Derivatives w.r.t. x

The following schemes can be used to evaluate partial derivative $\frac{\partial f}{\partial x}$ at a point (x^*, y^*) :

$$\left.\frac{\partial f}{\partial x}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^* + h,\, y^*\right) - f\left(x^*,\, y^*\right)}{h} \tag{5.16}$$

$$\left.\frac{\partial f}{\partial x}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^*,\, y^*\right) - f\left(x^* - h,\, y^*\right)}{h} \tag{5.17}$$

$$\left.\frac{\partial f}{\partial x}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^* + h,\, y^*\right) - f\left(x^* - h,\, y^*\right)}{2h} \tag{5.18}$$

Difference Formulas for Partial Derivatives w.r.t. y

$$\left.\frac{\partial f}{\partial y}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^*,\, y^* + k\right) - f\left(x^*,\, y^*\right)}{k} \tag{5.19}$$

$$\left.\frac{\partial f}{\partial y}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^*,\, y^*\right) - f\left(x^*,\, y^* - k\right)}{k} \tag{5.20}$$

$$\left.\frac{\partial f}{\partial y}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^*,\, y^* + k\right) - f\left(x^*,\, y^* - k\right)}{2k} \tag{5.21}$$

Second Order Partial Derivatives

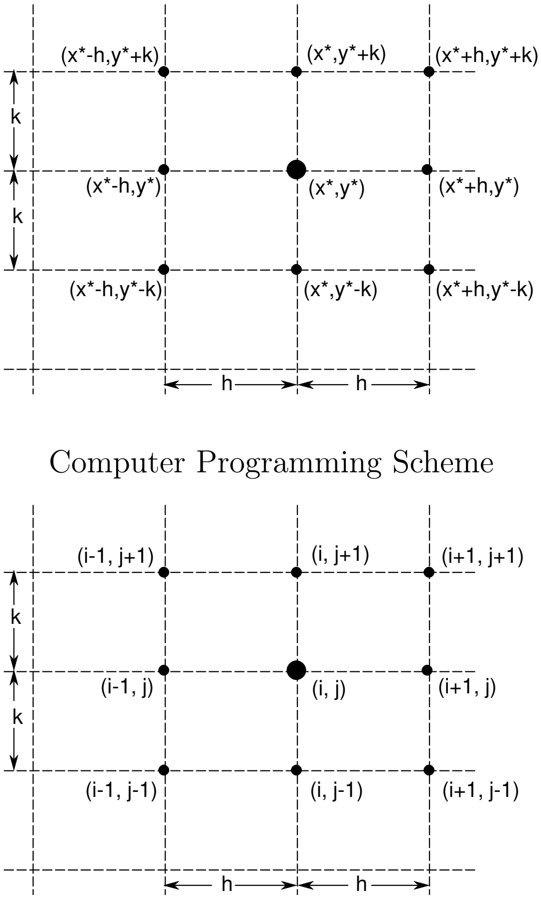
$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^* + h,\, y^*\right) - 2f\left(x^*,\, y^*\right) + f\left(x^* - h,\, y^*\right)}{h^2} \tag{5.22}$$

$$\left.\frac{\partial^2 f}{\partial y^2}\right|_{(x^*,\,y^*)} \approx \frac{f\left(x^*,\, y^* + k\right) - 2f\left(x^*,\, y^*\right) + f\left(x^*,\, y^* - k\right)}{k^2} \tag{5.23}$$

and

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{(x^*,\, y^*)} \approx \frac{1}{4hk}\left[f\left(x^*+h,\, y^*+k\right)-f\left(x^*+h,\, y^*-k\right)\right.$$
$$\left.-\left(x^*-h,\, y^*+k\right)+f\left(x^*-h,\, y^*-k\right)\right] \qquad (5.24)$$

Figure 5.1: Numerical Scheme for Partial Derivatives



Computer Programming Scheme

Exercise

1. Use the two-point forward difference formula with $h = 0.1$ to approximate the derivative of $f(x) = \frac{1}{x}$ at $x = 2$. What is the difference between this approximation and the exact derivative.

2. Show that the second order formula for fourth derivative is

$$f^{(4)}(x) \quad = \quad \frac{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)}{h^4} + O\left(h^2\right)$$

3. Using the following table, determine the approximate values of $f'(x)$ and $f''(x)$ using different order Lagrange interpolation polynomials at $x = 2.0$.

i	0	1	2	3
x_i	2.0	2.2	2.6	3.1
$f\left(x_i\right)$	7.3890	9.0250	13.4637	22.1979

Compare your results with exact function $f(x) = e^x$.

Chapter 6

Numerical Integration

In many applications we need to evaluate definite integrals of continuous functions over intervals, i.e., we want to compute the definite integral of a continuous $f(x)$ over the interval $[a,\, b]$

$$\int_a^b f(x)\, dx$$

Fundamental Theorem of Calculus (FTC)

1. Let $f(x)$ be a continuous function on $[a, b]$. Then

$$\int_a^b f(x)\, dx \quad = \quad F(b) - F(a) \tag{6.1}$$

   where $F(x)$ is the anti-derivative of f, i.e., $F'(x) = f(x)$.

2. Let $f(x)$ be a continuous function on $[a, b]$ and $g(x) = \int_a^x f(t)\, dt$ where $a \leq x \leq b$. Then

$$g'(x) \quad = \quad f(x)$$

Using Fundamental Theorem (1) of Calculus , we can obtain exact values of the definite integrals. But, in practice, lot of times the functions appearing in the definite integrals are so complicated that it is very difficult to find anti derivatives. So the above result (6.1) can not be used. Hence we need to use some numerical technique to evaluate the definite integrals. Here we will use polynomial interpolation to approximate a function $f(x)$.

Consider the set of distinct nodes $x_0, x_1, ..., x_n$ of $[a, b]$. Then we use Lagrange polynomials to express $f(x)$ as

$$f(x) \quad = \quad p(x) + \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^{n} (x - x_k) \tag{6.2}$$

where $\xi_x \in (a, b)$ and

$$p(x) \quad = \quad \sum_{i=0}^{n} f(x_i) L_i(x)$$

so that integration yields us

$$\int_a^b f(x)dx \quad = \quad \sum_{i=0}^{n} f(x_i) \int_a^b L_i(x)dx + \int_a^b \frac{f^{(n+1)}(\xi_x)}{(n+1)!} \prod_{k=0}^{n} (x - x_k) dx$$

Thus we have

$$\int_a^b f(x)dx \quad = \quad \sum_{i=0}^{n} w_i f(x_i) + \frac{1}{(n+1)!} \int_a^b \prod_{k=0}^{n} (x - x_k) f^{(n+1)}(\xi_x) dx (6.3)$$

where

$$w_i \quad = \quad \int_a^b L_i(x)dx$$

The quadrature formula becomes

$$\int_a^b f(x)dx \quad \approx \quad \sum_{i=0}^{n} w_i f(x_i) \tag{6.4}$$

with the error given by

$$E(f) = \frac{1}{(n+1)!} \int_a^b \prod_{k=0}^{n} (x - x_k) f^{(n+1)}(\xi_x)\, dx$$

There are two types of techniques used to evaluate definite integral (6.4): Newton-Cotes Formulas and Gaussian Quadrature. If the nodes are equally spaced, the formula presented in (6.4) is known as **Newton-Cotes formula**.Other technique is the **Gaussian Quadrature.**

6.1 Newton-Cotes Formulas

Now we discuss Newton-Cotes Formulas in detail. These include Trapezoid Rule, various Simpson's Rules and Midpoint Rule.

6.1.1 Trapezoid Rule

If we consider $n = 1$ in (6.2) and the nodes are $x_0 = a,\ x_1 = b$, we obtain one of simplest rule to evaluate definite integrals. In this case

$$L_0(x) = \frac{x_1 - x}{x_1 - x_0} = \frac{b - x}{b - a} \quad \text{and} \quad L_1(x) = \frac{x - x_0}{x_1 - x_0} = \frac{x - a}{b - a}$$

yielding

$$w_0 = \int_a^b L_0(x)dx = \frac{b-a}{2} \quad \text{and} \quad w_1 = \int_a^b L_1(x)dx = \frac{b-a}{2}$$

Hence the formula becomes

$$\begin{aligned} \int_a^b f(x)dx &\approx \sum_{i=0}^{1} w_i f(x_i) \\ &= \frac{b-a}{2}\left[f(a) + f(b)\right] \\ &= \frac{h}{2}\left[f(x_0) + f(x_1)\right] \qquad (6.5) \end{aligned}$$

where $h = x_1 - x_0 = b - a$.

MVT for integrals:

Let $f(x)$ be a continuous function on $[a,\, b]$ and let $g(x)$ be an integrable function that does not change sign on $[a,\, b]$. Then there exists a $\xi \in (a,\, b)$ such that

$$\int_a^b f(x)g(x)dx \;=\; f(\xi)\int_a^b g(x)dx$$

Error in Trapezoidal Rule

Let f be twice continuously differentiable on $[x_0, x_1]$. Then the error in Trapezoidal rule is given by

$$E(f) \;=\; \int\limits_a^b f(x)dx - \frac{h}{2}\left[f\left(x_0\right) + f\left(x_1\right)\right] = \int\limits_{x_0}^{x_1} \prod_{k=0}^{1}\left(x - x_k\right)\frac{f''\left(\xi_x\right)}{2}dx$$

and for some $\xi_x \in \left(x_0, x_1\right)$.

This error can be computed as

$$\begin{aligned}
E(f) &= \frac{1}{2}\int\limits_{x_0}^{x_1}\prod_{k=0}^{1}\left(x - x_k\right)f''\left(\xi_x\right)dx \\
&= \frac{f''\left(\xi\right)}{2}\int\limits_{x_0}^{x_1}\left(x - x_0\right)\left(x - x_1\right)dx \qquad for\ some\ \xi \in \left(x_0,\, x_1\right)\ using\ MVT\ for\ integrals \\
&= \frac{f''\left(\xi\right)}{2}\left[\frac{x^3}{3} - \frac{x^2}{2}\left(x_0 + x_1\right) + x_0x_1x\right]_{x_0}^{x_1} \\
&= \frac{f''\left(\xi\right)}{12}\left[2\left(x_1^3 - x_0^3\right) - 3\left(x_1^2 - x_0^2\right)\left(x_0 + x_1\right) + 6x_0x_1\left(x_1 - x_0\right)\right] \\
&= \frac{f''\left(\xi\right)}{12}\left(x_1 - x_0\right)\left[2x_1^2 + 2x_1x_0 + 2x_0^2 - 3x_1^2 - 6x_1x_0 - 3x_0^2 + 6x_0x_1\right] \\
&= \frac{f''\left(\xi\right)}{12}\left(x_1 - x_0\right)\left[-x_1^2 + 2x_1x_0 - x_0^2\right] \\
&= -\frac{f''\left(\xi\right)}{12}\left(x_1 - x_0\right)^3 \\
&= -\frac{1}{12}h^3f''\left(\xi\right) \qquad (6.6)
\end{aligned}$$

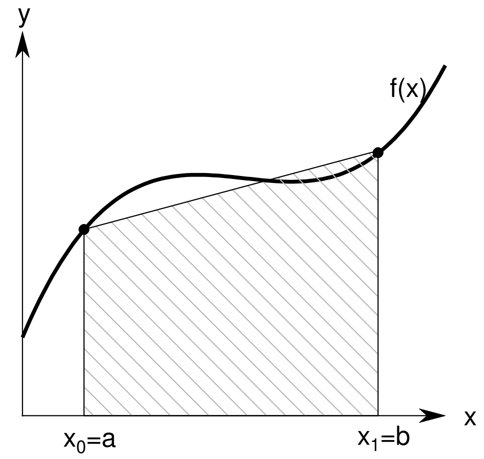
Another way to calculate the error:

Substitute $x - x_0 = v$. So we have $dx = dv$, $x = x_0 \implies v = 0$ and $x = x_1 \implies v = x_1 - x_0 = h$. So the error becomes

$$\begin{aligned}
E(f) &= \frac{1}{2}\int\limits_{x_0}^{x_1} \prod_{k=0}^{1} (x - x_k)\, f''(\xi_x)\, dx \\
&= \frac{f''(\xi)}{2}\int\limits_{x_0}^{x_1} (x - x_0)(x - x_1)\, dx \qquad for\ some\ \xi \in (x_0,\, x_1)\ using\ MVT\ for\ integrals \\
&= \frac{f''(\xi)}{2}\int_0^h v(v-h)dv = \frac{f''(\xi)}{2}\left[\frac{v^3}{3} - \frac{hv^2}{2}\right]_{v=0}^{h} \\
&= -\frac{1}{12}h^3 f''(\xi)
\end{aligned}$$

Figure 5.2 shows the Trapezoid rule.

Figure 6.1: Trapezoid Rule to Evaluate Definite Integral



6.1.2 Simpson's 1/3 Rule

Here we consider $n = 2$ in (6.2) and the nodes are $x_0 = a,\ x_1$ and $x_2 = b$, we obtain Simpson's rule to evaluate definite integrals. Thus we have the three Lagrange polynomials as

$$L_0\left(x\right) = \frac{\left(x - x_1\right)\left(x - x_2\right)}{\left(x_0 - x_1\right)\left(x_0 - x_2\right)}, \quad L_1\left(x\right) = \frac{\left(x - x_0\right)\left(x - x_2\right)}{\left(x_1 - x_0\right)\left(x_1 - x_2\right)}, \qquad L_2\left(x\right) = \frac{\left(x - x_0\right)\left(x - x_1\right)}{\left(x_2 - x_0\right)\left(x_2 - x_0\right)}$$

and to compute definite integral, we need to evaluate

$$\int\limits_{x_0}^{x_2} f(x)dx \ \approx \ \sum_{i=0}^{2} w_i\, f\left(x_i\right)$$

i.e.,

$$w_0 = \int\limits_{x_0}^{x_2} L_0(x)dx, \quad w_1 = \int\limits_{x_0}^{x_2} L_1(x)dx \quad \text{and} \quad w_2 = \int\limits_{x_0}^{x_2} L_2(x)dx$$

Computation of w_0 :

$$w_0 \ = \ \int_{x_0}^{x_2} \frac{\left(x - x_1\right)\left(x - x_2\right)}{\left(x_0 - x_1\right)\left(x_0 - x_2\right)}\, dx$$

Substitute $x - x_1 = v$. So we have $dx = dv,\ \ x = x_0 \implies v = x_0 - x_1 = -h$ and $x = x_2 \implies v = x_2 - x_1 = h$. Also we know that $(x_2 - x_0)/2 = h$. This yields

$$\begin{aligned} w_0 \ &= \ \int_{-h}^{h} \frac{v(v-h)}{(-h)\left(-2h\right)}\, dv = \frac{1}{2h^2}\int_{-h}^{h}(v^2 - hv)dv \\ &= \ \frac{1}{2h^2}\left[\frac{v^3}{3} - \frac{hv^2}{2}\right]_{-h}^{h} = \frac{1}{2h^2}.\frac{2h^3}{3} = \frac{h}{3} \end{aligned}$$

Similarly, it can be shown that

$$w_1 = \frac{4h}{3} \qquad \textit{and} \qquad w_2 = \frac{h}{3}$$

Hence the formula becomes

$$\begin{aligned}\int_a^b f(x)dx &\approx \sum_{i=0}^{2} w_i f\left(x_i\right) \\ &= \frac{h}{3}\left[f\left(x_0\right)+4f\left(x_1\right)+f\left(x_2\right)\right] \qquad (6.7)\end{aligned}$$

where $h = x_1 - x_0 = x_2 - x_1 = (b-a)/2$.

Error in Simpson's 1/3 Rule

Let f be thrice continuously differentiable on $[x_0, x_2]$. Then the error in Simpson's 1/3-rd rule gives

$$E(f) = \int_a^b f(x)dx - \frac{h}{3}\left[f\left(x_0\right)+4f\left(x_1\right)+f\left(x_2\right)\right] = \int_{x_0}^{x_2} \frac{f^{(3)}\left(\xi_x\right)}{6}\prod_{k=0}^{2}\left(x-x_k\right)dx$$

for some for some $\xi_x \in \left(x_0, x_2\right)$. This provides only an $O(h^4)$ error term involving $f^{(3)}$. Following another approach, we can have higher-order term involving $f^{(4)}$. Using Taylor polynomial of degree 3 about $x_1$as follows, we have for each $x \in [x_0,\, x_2]$, there exists a number $\xi_x \in (x_0,\, x_2)$ such that

$$f(x) = f(x_1) + \left(x-x_1\right)f'(x_1) + \frac{\left(x-x_1\right)^2}{2}f''(x_1) + \frac{\left(x-x_1\right)^3}{6}f'''(x_1) + + \frac{\left(x-x_1\right)^4}{24}f^{(4)}(\xi_x)$$

where (Burden and Faires)

$$\begin{aligned}E(f) &= \frac{1}{3!}\int_{x_0}^{x_2}\prod_{k=0}^{2}\left(x-x_k\right)f'''\left(\xi_x\right)dx \\ &= -\frac{1}{90}h^5 f^{(4)}\left(\xi\right). \qquad (6.8)\end{aligned}$$

Figure 6.2: Simpson's Rule to Evaluate Definite Integral

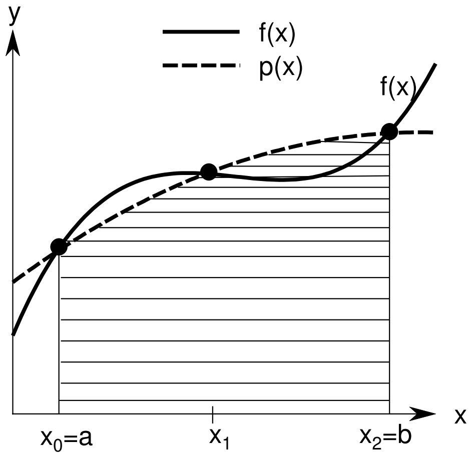


Figure 5.3 shows the Simpson's rule.
Simpson's (one-third) rule ($n = 2$) is given by

$$\int_{x_0}^{x_2} f(x)dx \quad = \quad \frac{h}{3}\left[f\left(x_0\right)+4f\left(x_1\right)+f\left(x_2\right)\right]-\frac{h^5}{90}f^{(4)}\left(\xi\right)$$

where $h = \frac{x_2 - x_0}{2} = \frac{b-a}{2}$ and $x_0 < \xi < x_2$.

6.1.3 Simpson's Three-Eights Rule

Here we consider $n = 3$ in (6.2) and the nodes are $x_0 = a,\ x_1,\ x_2$ and $x_3 = b$, we obtain Simpson's 3/8 th rule to evaluate definite integrals. This rule can be expressed including the error term as

$$\int_{x_0}^{x_3} f(x)dx \quad = \quad \frac{3h}{8}\left[f\left(x_0\right)+3f\left(x_1\right)+3f\left(x_2\right)+f\left(x_3\right)\right]-\frac{3h^5}{80}f^{(4)}\left(\xi\right)$$

where $h = \frac{b-a}{3}$ and $x_0 < \xi < x_3$.

6.1.4 Midpoint Rule

This is also known as the Open Newton-Cotes method. Consider the interval $[x_0,\, x_1]$. Let $h = \frac{x_1 - x_0}{2}$. Then we have

$$\int\limits_{x_0}^{x_1} f(x)dx \quad = 2hf\left(x_m\right) \quad + \frac{h^3}{3} f''\left(\xi\right)$$

where $x_m = x_0 + h$ which is the midpoint of the interval and $x_0 < \xi < x_1$.

6.2 Composite Numerical Integration

A composite rule is a rule constructed from an integration rule for a single interval to its sub-intervals. Consider the sub intervals of $[a, b]$ as

$$a = x_0 < x_1 < x_2 < .... < x_n = b$$

6.2.1 Composite Trapezoid Rule

Using the Trapezoid rule to each interval, we can write the composite Trapezoid rule as

$$\begin{aligned}\int\limits_a^b f(x)dx &= \sum_{i=1}^{n} \int\limits_{x_{i-1}}^{x_i} f(x)dx \\ &= \frac{1}{2}\sum_{i=1}^{n} \left(x_i - x_{i-1}\right)\left[f\left(x_{i-1}\right) + f\left(x_i\right)\right] - \sum_{i=1}^{n} \frac{f''\left(\xi_i\right)}{12}\left(x_i - x_{i-1}\right)^3\end{aligned}$$

where $x_{i-1} < \xi_i < x_i$. If x_i's are spaced uniformly, we can define
$h = x_i - x_{i-1} = \frac{b-a}{n}$ and $x_i = x_{i-1} + h = a + ih$, then the composite Trapezoid rule can be expressed as

$$\begin{aligned}\int\limits_a^b f(x)dx &\approx \frac{h}{2}\sum_{i=1}^{n}\left[f\left(x_{i-1}\right) + f\left(x_i\right)\right] \\ &= \frac{h}{2}\left[f(a) + 2\sum_{i=1}^{n-1} f(a + ih) + f(b)\right]\end{aligned}$$

The error term in this composite rule becomes

$$-\frac{h^3}{12}\sum_{i=1}^{n} f''\left(\xi_i\right)$$

If $f \in C^2\left[a,b\right]$, then by Extreme value Theorem there exist two numbers $c_1, c_2 \in [a,b]$ such that

$$f''\left(c_1\right) = \min_{a\leq x\leq b} f''\left(x\right) \quad \text{and} \quad f''\left(c_2\right) = \max_{a\leq x\leq b} f''\left(x\right)$$

which gives us, for each i

$$f''\left(c_1\right) \quad \leq f''\left(\xi_i\right) \leq \quad f''\left(c_2\right)$$

Adding for all sub intervals, we get

$$nf''\left(c_1\right) \leq \sum_{i=1}^{n} \quad f''\left(\xi_i\right) \leq \quad nf''\left(c_2\right)$$

or,

$$f''\left(c_1\right) \quad \leq \tfrac{1}{n}\textstyle\sum_{i=1}^{n} f''\left(\xi_i\right) \leq \quad f''\left(c_2\right)$$

Now by the Intermediate Value Theorem, we conclude that there exists a ξ between a and b such that $f''\left(\xi\right) = \frac{1}{n}\sum_{i=1}^{n} f''\left(\xi_i\right)$ so that the error term is given by

$$-\frac{h^3}{12}\sum_{i=1}^{n} f''\left(\xi_i\right) \quad = \quad -\frac{nh^3}{12}f''\left(\xi\right) = -\frac{(b-a)h^2}{12}f''\left(\xi\right).$$

6.2.2 Composite Simpson's Rule

Since the Simpson's rule divides the interval $[a,b]$ into two parts, to use composite rule n must be even, say, $n = 2m$. If we define $h = \frac{b-a}{n} = \frac{b-a}{2m}$. we can write $x_i = a + ih,\ 0 \leq i \leq 2m$. Now we apply the Simpson's rule over the intervals $[x_{2i-2}, x_{2i}]$ for $i = 1, 2, ..., m$ to obtain

$$\begin{aligned}\int\limits_a^b f(x)dx &= \sum_{i=1}^{m} \frac{x_{2i}-x_{2i-2}}{6}\left[f\left(x_{2i-2}\right)+4f\left(x_{2i-1}\right)+f\left(x_{2i}\right)\right] \\ &\quad -\sum_{i=1}^{m} \frac{\left(x_{2i}-x_{2i-2}\right)^5}{90\times 2^5} f^{(4)}\left(\xi_i\right) \\ &= \frac{h}{3}\left[f\left(x_0\right)+4\sum_{i=1}^{m} f\left(x_{2i-1}\right)+2\sum_{i=1}^{m-1} f\left(x_{2i}\right)+f\left(x_{2m}\right)\right]-\frac{h^5}{90}\sum_{i=1}^{m} f^{(4)}\left(\xi_i\right)\end{aligned}$$

or,

$$\int\limits_a^b f(x)dx \approx \frac{h}{3}\left[f\left(x_0\right)+4\sum_{i=1}^{m} f\left(x_{2i-1}\right)+2\sum_{i=1}^{m-1} f\left(x_{2i}\right)+f\left(x_{2m}\right)\right]$$

and the error term becomes

$$-\frac{mh^5}{90} f^{(4)}\left(\xi\right) = -\frac{(b-a)h^4}{180} f^{(4)}\left(\xi\right)$$

where $\xi \in (a,b)$.

6.3 Gaussian Quadrature

The quadrature methods mentioned in earlier sections evaluate the definite integrals using polynomial interpolation at uniformly spaced points. The assumption of equally spaced sub intervals is convenient to obtain those composite integration rules, but it can adversely affect the accuracy of the results. For example, consider the following situation described in this figure. The Trapezoid rule uses the end points to join the line to integrate, but this may not be a better approximation. Other line obtained by some two other points may give us a better estimate. Figure 6.3 below illustrates this point.

Transformation

The integral $\int_a^b f(x)dx$ over an interval $[a,b]$ can be transformed to an integral over $[-1,1]$ by using the change of variables as follows.

We define a relationship between x and t as

$$x = \frac{1}{2}\left[(b-a)t+b+a\right]$$

or,

$$t = \frac{2x-b-a}{b-a}$$

we have $dx = \frac{b-a}{2}dt$, so that we can write

$$\int\limits_a^b f(x)dx = \tfrac{b-a}{2} \int\limits_{-1}^{1} f\left(\frac{(b-a)t+b+a}{2}\right)dt$$

Gaussian Quadrature formula

Formula is

$$\int\limits_{-1}^{1} f(x)dx \approx \sum_{i=0}^{n} c_i\, f\left(x_i\right) \tag{6.9}$$

where $c_i = \int_{-1}^{1} L_i(x)dx,\ i = 1, 2, ..., n.$

Example 5

Using Gaussian Quadrature, evaluate $\int_2^5 \ln x\, dx$

Here we have

$$\int\limits_2^5 \ln x\, dx = \frac{3}{2}\int\limits_{-1}^{1} \ln\left(\frac{3t+7}{2}\right)dt$$

Please complete this example using $n = 2$ and $n = 3$.

Table 6.1: Coefficients for Gaussian Quadrature

n	x_i	coefficients, c_i
2	$-\sqrt{\frac{1}{3}} = -0.57735$ $\sqrt{\frac{1}{3}} = 0.57735$	1 1
3	$-\sqrt{\frac{3}{5}} = -0.77459$ 0 $\sqrt{\frac{3}{5}} = 0.77459$	$\frac{5}{9} = 0.\overline{5}$ $\frac{8}{9} = 0.\overline{8}$ $\frac{5}{9} = 0.\overline{5}$
4	$-\sqrt{\frac{15+2\sqrt{30}}{35}}$ $-\sqrt{\frac{15-2\sqrt{30}}{35}}$ $\sqrt{\frac{15-2\sqrt{30}}{35}}$ $\sqrt{\frac{15+2\sqrt{30}}{35}}$	$\frac{90-5\sqrt{30}}{180}$ $\frac{90+5\sqrt{30}}{180}$ $\frac{90+5\sqrt{30}}{180}$ $\frac{90-5\sqrt{30}}{180}$

Derivation of Gaussian Points and Weights

$$\int_{-1}^{1} f(x)\,dx \quad \approx \quad \sum_{i=1}^{n} c_i f(x_i) = c_1 f(x_1) + c_2 f(x_2) + ... + c_n f(x_n) \tag{6.10}$$

Two point ($n = 2$) : Choose (c_1, c_2, x_1, x_2) such that the method yields exact integral for $f(x) = 1$, x, x^2, x^3.

$$\int_{-1}^{1} f(x)\,dx \quad = \quad c_1 f(x_1) + c_2 f(x_2) \tag{6.11}$$

Here we have four equations for four unknowns (c_1, c_2, x_1, x_2) , namely,

$$f(x) = 1 \Rightarrow \int_{-1}^{1} f(x)\,dx = 2 \quad \Longrightarrow \quad c_1 + c_2 = 2 \tag{6.12}$$

$$f(x) = x \Rightarrow \int_{-1}^{1} f(x)\,dx = 0 \quad \Longrightarrow \quad c_1 x_1 + c_2 x_2 = 0 \tag{6.13}$$

$$f(x) = x^2 \Rightarrow \int_{-1}^{1} f(x)\,dx = \frac{2}{3} \quad \Longrightarrow \quad c_1 x_1^2 + c_2 x_2^2 = \frac{2}{3} \tag{6.14}$$

$$f(x) = x^3 \Rightarrow \int_{-1}^{1} f(x)\,dx = 0 \quad \Longrightarrow \quad c_1 x_1^3 + c_2 x_2^3 = 0 \tag{6.15}$$

From (6.13) and (6.15), we have

$$\left(\frac{x_2}{x_1}\right) = -\left(\frac{c_1}{c_2}\right) \quad \textit{and} \quad \left(\frac{x_2}{x_1}\right)^3 = -\left(\frac{c_1}{c_2}\right)$$

which yield

$$\left(-\frac{c_1}{c_2}\right)^3 = -\left(\frac{c_1}{c_2}\right) \;\Rightarrow\; \left(\frac{c_1}{c_2}\right)^2 = 1 \tag{6.16}$$

From (6.12) and (6.16), we obtain $c_1 = 1 = c_2$. Putting these values , from (6.13) and (6.14), we can write

$$x_1 + x_2 = 0 \quad \textit{and} \quad x_1^2 + x_2^2 = \frac{2}{3}$$

yielding $2x_2^2 = \frac{2}{3}$, or, $x_2 = \frac{1}{\sqrt{3}}$ and $x_1 = -\frac{1}{\sqrt{3}}$. Here, we assume $x_1 < x_2$.

Example 7

6.4 Adaptive Quadrature

The composite formulas for approximation definite integrals use equally spaced nodes. These methods are not suitable for a function with large variation in one region and small variation in another region. For a region with large variation, smaller step size ie required than the for those with small variation. Adaptive quadrature methods take acre of this problem. These methods adapt the step size depending the functional variation. These methods, in general, also provide approximations that are within a given specified tolerance.

Here we discuss an adaptive quadrature technique that can be used not only to reduce approximation error, but also to predict an error estimate for the approximation which does not rely on knowledge of higher derivatives of the function. This particular method is based on Composite Simpson's rule.

We want to approximate $\int_a^b f(x)\,dx$ within a specified tolerance ϵ. Applying Simpson's rule with $n = 2$, i.e., with $h = (b-a)/2$,, we obtain

$$\int_a^b f(x)\,dx \;=\; \frac{h}{3}\left[f(a) + 4f(a+h) + f(b)\right] - \frac{h^5}{90} f^{(4)}(\xi_1)$$

for some $\xi_1 \in (a,\, b)$. Let us write

$$Q(a, b) \;=\; \frac{h}{3}\left[f(a) + 4f(a+h) + f(b)\right]$$

Now, we apply Composite Simpson's rule with $n = 4$, i.e., with step size $h_1 = h/2 = (b-a)/4$, we obtain

$$\int_a^b f(x)\,dx = \frac{h}{6}\left[f(a) + 4f\left(a + \frac{h}{2}\right) + 2f\left(a+h\right) + 4f\left(a + \frac{3h}{2}\right) + f(b)\right] - \left(\frac{h}{2}\right)^4 \frac{b-a}{180} f^{(4)}(\xi_2) \qquad (6.17)$$

for some $\xi_2 \in (a,\, b)$. Using the following notations,

$$Q\left(a,\, \frac{a+b}{2}\right) \;=\; \frac{h}{6}\left[f(a) + 4f\left(a + \frac{h}{2}\right) + f\left(a+h\right)\right]$$

and

$$Q\left(\frac{a+b}{2},\, b\right) \;=\; \frac{h}{6}\left[f(a+h) + 4f\left(a + \frac{3h}{2}\right) + f\left(b\right)\right]$$

we have, from (6.17),

$$\int_a^b f(x)\,dx \;=\; Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\xi_2) \quad (6.18)$$

Error estimation is obtained assuming that $f^{(4)}(\xi_1) \approx f^{(4)}(\xi_2) \approx f^{(4)}(\xi)$. So we have

$$\begin{aligned} &Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right) - \frac{1}{16}\frac{h^5}{90} f^{(4)}(\xi) \\ &\qquad \approx Q\left(a,\, \frac{a+b}{2}\right) - Q\left(\frac{a+b}{2},\, b\right) Q(a,\, b) - \frac{h^5}{90} f^{(4)}(\xi) \end{aligned}$$

i.e.,

$$\frac{h^5}{90} f^{(4)}(\xi) \approx \frac{16}{15}\left[Q(a,\, b) - Q\left(a,\, \frac{a+b}{2}\right) - Q\left(\frac{a+b}{2},\, b\right)\right]$$

Use of (6.18) yields

$$\begin{aligned} &\left|\int_a^b f(x)\,dx - \left\{Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right)\right\}\right| \\ &\qquad \approx \frac{1}{15}\left|Q(a,\, b) - \left\{Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right)\right\}\right| \end{aligned}$$

This means that $Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right)$ approximates $\int_a^b f(x)\,dx$ about 15 times better than it does with $Q(a,\, b)$. So, if

$$\left|Q(a,\, b) - \left\{Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right)\right\}\right| < 15\epsilon, \qquad (6.19)$$

we expect to get

$$\left|\int_a^b f(x)\,dx - \left\{Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right)\right\}\right| < \epsilon, \qquad (6.20)$$

and

$$Q\left(a,\, \frac{a+b}{2}\right) + Q\left(\frac{a+b}{2},\, b\right)$$

yields a sufficiently accurate approximation of the integral $\int_a^b f(x)\,dx$.

Example 8

Check the error accuracy given in (6.19) and (6.20) by considering the integral

$$\int_0^{\pi/2} \sin x \, dx$$

Solution:

Here the exact solution is 1. We have

$$Q\left(0, \frac{\pi}{2}\right) = \frac{\pi/4}{3}\left[\sin 0 + 4\sin\frac{\pi}{4} + \sin\frac{\pi}{2}\right] = \frac{\pi}{12}\left[2\sqrt{2}+1\right] = 1.0022798775$$

and

$$\begin{aligned} Q\left(0, \frac{\pi}{4}\right) + Q\left(\frac{\pi}{4}, \frac{\pi}{2}\right) &= \frac{\pi/8}{3}\left[\sin 0 + 4\sin\frac{\pi}{8} + 2\sin\frac{\pi}{4} + 4\sin\frac{3\pi}{8} + \sin\frac{\pi}{2}\right] \\ &= 1.000134585 \end{aligned}$$

Hence,

$$\left|Q\left(0, \frac{\pi}{2}\right) - \left(0, \frac{\pi}{4}\right) - Q\left(\frac{\pi}{4}, \frac{\pi}{2}\right)\right| = 0.0021452925$$

The estimate for the error obtained using $Q\left(0, \frac{\pi}{4}\right) + Q\left(\frac{\pi}{4}, \frac{\pi}{2}\right)$ to approximate $\int_0^{\pi/2} \sin x \, dx$ is

$$\frac{1}{15}\left|Q\left(0, \frac{\pi}{2}\right) - \left(0, \frac{\pi}{4}\right) - Q\left(\frac{\pi}{4}, \frac{\pi}{2}\right)\right| = 0.0001430195$$

which closely approximates the actual error

$$\left|\int_0^{\pi/2} \sin x \, dx - \left(0, \frac{\pi}{4}\right) - Q\left(\frac{\pi}{4}, \frac{\pi}{2}\right)\right| = |1 - 1.000134585| = 0.000134585$$

even though $f^{(4)}(\sin x) = \sin x$ has significant variation in the interval $\left(0, \frac{\pi}{2}\right)$.

Exercise

1. Approximate the integrals using Trapezoid rule:

$$\text{(a)} \int_1^{1.5} x^2 \ln x dx \qquad \text{(b)} \int_e^{e+1} \frac{1}{x \ln x} dx$$

2. Derive Simpson's Rule with error term by using

$$\int_{x_0}^{x_2} f(x)dx = a_0 f\left(x_0\right) + a_1 f\left(x_1\right) + a_2 f\left(x_2\right) + k f^{(4)}\left(\xi\right)$$

   Find $a_0,\ a_1,\ a_2$ from the fact that Simpson's rule is exact for $f(x) = x^n$ when $n = 1,\ 2,\ 3$. Then find k by applying the integration formula with $f(x) = x^4$. Here $x_1 = x_0 + h$ and $x_2 = x_1 + h = x_0 + 2h$.

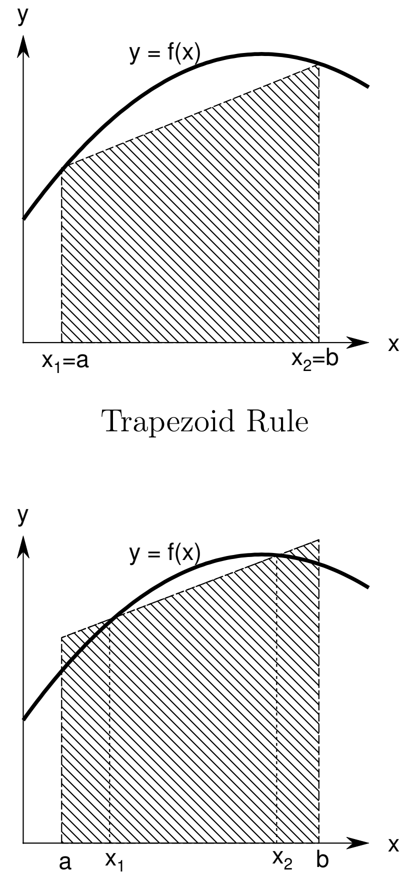
3. Apply the composite Trapezoid Rule with $m = 1,\ 2$ to approximate $\int_0^1 x^2 dx$. Compute the error by comparing with the exact value.

4. Apply the composite Simpson's Rule with $m = 1,\ 2$ to approximate $\int_0^1 x^2 dx$. Compute the error by comparing with the exact value.

5. Approximate $\int_{-1}^{1} \left(x^3 + 2x\right) dx$ using $n = 2$ Gaussian Quadrature. Compare your result with the exact value.

6. Approximate $\int_0^4 \left(t\, e^{2t}\right) dt$ using $n = 2$ Gaussian Quadrature. Compare your result with the exact value.

Figure 6.3: Comparison of Trapezoid and Gaussian Quadrature



Trapezoid Rule

Gaussian Quadrature

Chapter 7

Initial Value Problems (IVP)

An equation involving one or more derivatives is called a **differential equation**. Differential equations are obtained when we try to model various scenarios arising from different practical applications. If the dependent variable is a function of a single independent variable, then the equation is known as an **ordinary differential equation (ODE).** When the dependent variable is a function of two or more independent variables, then the equation is referred as a **partial differential equation (PDE).**

Here we will consider only ordinary differential equations. The **order** of an ordinary differential equation is defined to be the order of the highest derivative present in that equation. An nth-order ordinary differential equation can be expressed as

$$F\left(t, y, \frac{dy}{dt}, ..., \frac{d^n y}{dt^n}\right) = 0$$

It is said to be **linear** if F is a linear function in $y, \frac{dy}{dt}, ..., \frac{d^n y}{dt^n}$, otherwise it is **nonlinear**. A linear ordinary differential equation of nth order can be written in the form

$$a_n(t)\frac{d^n y}{dt^n} + a_{n-1}(t)\frac{d^{n-1} y}{dt^{n-1}} + ... + a_1(t)\frac{dy}{dt} + a_0(t)y = f(t) \quad (7.1)$$

where $a_i(t)$ and $f(t)$ are known functions of the independent variable t. This equation (7.1) is also called as the nth-order linear ordinary differential equation with **variable coefficients**. If the coefficients, $a_i(t),\ i = 0, 1, ..., n$ are constant, then the ODE presented in (7.1) is referred to as the nth-order linear ordinary differential equation with **constant coefficients**. If $f(t) = 0$, then the equation is said to be **homogeneous**, otherwise it is referred to as **non-homogeneous**.

Example

1. Consider

$$5t\frac{d^3y}{dt^3} + t^2\frac{dy}{dt} + 2t = 0$$

which a third-order linear homogeneous ODE with variable coefficients.

2. Consider

$$5\frac{d^2y}{dt^2} + \left(\frac{dy}{dt}\right)^3 + 2t = \sin t$$

which a second-order nonlinear non homogeneous ODE with constant coefficients.

3. Consider

$$5t\frac{d^2y}{dt^2} + y^2\frac{dy}{dt} + 2t = \sin t$$

which a second-order nonlinear ODE.

Initial Value Problem and Boundary Value problem

To solve an ordinary differential equation, we need to do integration. For example, consider the following ODE

$$\frac{dy}{dt} - 12t = 5$$

If we solve the above ODE, we obtain

$$y(t) = 6t^2 + 5t + C$$

where C is an arbitrary constant.

This solution is not unique, for each C, we obtain a different solution $y(t)$. When we are interested in a particular solution, we need to specify a condition at the initial value of t, say

$$y(1) \;=\; 8$$

then we have unique solution

$$y(t) \;=\; 6t^2 + 5t - 3$$

This type of problem is called **Initial Value Problem.** Thus the initial value problem

$$\frac{dy}{dt} - 12t = 5 \qquad\qquad y(1) = 8$$

has the solution $y(t) = 6t^2 + 5t - 3$.

Consider the following second order linear homogeneous differential equation with constant coefficients

$$\frac{d^2y}{dx^2} + y = 0 \qquad\qquad 0 \leq x \leq \frac{\pi}{2}$$

Here solution can be obtained as

$$y(x) \;=\; A\cos x + B\sin x$$

where A and B are arbitrary constants. If we impose two conditions at the boundaries, i.e., at $x = 0$ and $x = \pi/2$, say, $y(0) = 1$ and $y\left(\frac{\pi}{2}\right) = 2$, we obtain the solution as $y(x) = \cos x + 2\sin x$.

This type of problem is known as the **Boundary Value Problem.** Thus the following boundary value problem

$$\frac{d^2y}{dx^2} + y = 0 \qquad\qquad 0 \leq x \leq \frac{\pi}{2}$$

$$y(0) = 1 \qquad \text{and} \qquad y\left(\frac{\pi}{2}\right) = 2$$

has the solution

$$y(x) \;=\; \cos x + 2\sin x.$$

But most of the time, ordinary or partial differential equations we encounter in various applications are not as simple as the above one. In many situations, we obtain complicated differential equations which can not be solved analytically. We need to solve those equations numerically.

Numerical Methods to Solve Initial Value Problems

Now we discuss different numerical methods to solve initial value problems. Various methods presented here are **Euler's Method**, **Taylor Method**s, **and Runge-Kutta Method**s.

7.1 Euler's Method

Consider the following first-order initial value problem (IVP)

$$\begin{aligned} \frac{dy}{dt} = y'(t) = f(t, y), \qquad & a \leq t \leq b \\ y(a) = \rho_0 \quad & \end{aligned} \tag{7.2}$$

where ρ_0 is a constant. We want to approximate $y(t)$ numerically, i.e., we want to find a numerical approximation z such that $z \approx y$ where $y(t)$ is the exact solution the above IVP. We consider equally spaced $(n+1)$ points $a = t_0 < t_1 < ..... < t_{n-1} < t_n = b$ such that step size h and t_i are given by

$$h = \frac{b-a}{n} \qquad \text{and} \qquad t_i = a + ih, \; i = 0, 1, ..., n$$

and we will obtain discretized values of z at t_i i.e., z_i. This means z_i represents the approximation $y\left(t_i\right) = y_i$. Assuming that $y(t)$ has two continuous derivatives and using Taylor series expansion about $t = t_i$, we can write

$$y(t) \quad = \quad y\left(t_i\right) + \left(t - t_i\right) y'\left(t_i\right) + \frac{\left(t - t_i\right)^2}{2} y''\left(\xi\right)$$

where ξ lies between t and t_i. Evaluating at $t = t_{i+1}$, above result can be expressed as

$$y\left(t_{i+1}\right) \quad = \quad y\left(t_i\right) + hf\left(t_i, y_i\right) + \frac{h^2}{2} y''\left(\xi\right)$$

or,

$$y_{i+1} = y_i + hf\left(t_i, y_i\right) + \frac{h^2}{2} y''\left(\xi\right) \tag{7.3}$$

Now using approximation, we have

$$z_{i+1} = z_i + hf\left(t_i, z_i\right) \qquad i = 0, 1, 2, ..., n-1 \tag{7.4}$$

Subtraction of (7.3) and (7.4) yields the local truncation error as

$$e_{i+1} = \left|z_{i+1} - y_{i+1}\right| = \frac{h^2}{2}\left|y''\left(\xi\right)\right| \tag{7.5}$$

Let M be an upper bound for y'' on $[a, b]$. Then local truncation error satisfies

$$e_i \leq \frac{Mh^2}{2}$$

Another way to derive Euler's method is to integrate both sides of (7.2) from $t = t_i$ to $t = t_{i+1}$ to yield

$$y\left(t_{i+1}\right) - y\left(t_i\right) = \int_{t_i}^{t_{i+1}} f(t, y)\, dt$$

Using the left end point approximation, we get the same result as in equation (7.4).

Algorithm for Euler's Method

Here is the algorithm for Euler's method.

```
$z_0 = \rho_0$
do $i = 0, 1, ....., n-1$
    $z_{i+1} = z_i + hf\left(t_i, z_i\right)$
end $i$ loop
```

Example 1

Consider the following IVP

$$y' = y - t^2 + 1, \qquad 0 \le t \le 2, \quad y(0) = \tfrac{1}{2}$$

Use Euler's method to solve numerically this IVP with $n = 10$.

7.2 Taylor Methods

It is easy and straightforward to derive and implement Euler's method, but it lacks accuracy as the error is only $O(h)$. Assuming that $y(t)$ has $(n+1)$ continuous derivatives and using Taylor series expansion about $t = t_i$, we can write

$$\begin{aligned} y(t) &= y\left(t_i\right) + \left(t - t_i\right) y'\left(t_i\right) + \frac{\left(t-t_i\right)^2}{2} y''\left(t_i\right) + \ldots\ldots.. + \frac{\left(t-t_i\right)^n}{n!} y^{(n)}\left(t_i\right) \\ &\quad + \frac{\left(t-t_i\right)^{n+1}}{(n+1)!} y^{(n+1)}\left(\xi\right) \end{aligned}$$

where ξ lies between t and t_i. Evaluating at $t = t_{i+1}$, above result can be expressed as

$$\begin{aligned} y(t_{i+1}) &= y\left(t_i\right) + h y'\left(t_i\right) + \frac{h^2}{2} y''\left(t_i\right) + \ldots\ldots. \\ &\qquad \ldots.. + \frac{h^n}{n!} y^{(n)}\left(t_i\right) + \frac{h^{n+1}}{(n+1)!} y^{(n+1)}\left(\xi\right) \end{aligned} \tag{7.6}$$

Now we look into $y''\left(t_i\right)$. (Recall that if $w = w(u, v)$ and $u = u(t)$, $v = v(t)$ then $\frac{dw}{dt} = \frac{\partial w}{\partial u}\frac{du}{dt} + \frac{\partial w}{\partial v}\frac{dv}{dt}$). Using total derivative we obtain

$$y''\left(t_i\right) = \frac{d}{dt} f(t, y)|_{t=t_i} = \left(\frac{\partial f}{\partial t} + \frac{\partial f}{\partial y}\frac{dy}{dt}\right)_{t=t_i} = \left(\frac{\partial f}{\partial t} + f\,\frac{\partial f}{\partial y}\right)_{t=t_i} \tag{7.7}$$

Substituting into (7.6), we can derive the approximations as

$$z_{i+1} = z_i + h f\left(t_i, z_i\right) + \frac{h^2}{2}\frac{d}{dt} f\left(t_i, z_i\right) + \frac{h^3}{6}\frac{d^2}{dt^2} f\left(t_i, z_i\right) + \ldots\ldots + \frac{h^n}{n!}\frac{d^{n-1}}{dt^{n-1}} f\left(t_i, z_i\right) \tag{7.8}$$

Local truncation error in Taylor method

$$e_{i+1} = \left|z_{i+1} - y_{i+1}\right| = \frac{h^{n+1}}{(n+1)!}\left|y^{(n+1)}(\xi)\right| \tag{7.9}$$

The first-order Taylor method is $z_{i+1} = z_i + hf(t_i, z_i)$ which is Euler's method.

The second-order Taylor method can be written as

$$z_{i+1} = z_i + hf(t_i, z_i) + \frac{h^2}{2}\frac{d}{dt}f(t_i, z_i) \tag{7.10}$$

The second-order Taylor method requires two function evaluations per time step.

Algorithm for the Second-Order Taylor Method

Here is the algorithm for Euler's method.

```
$z_0 = \rho_0$
do $i = 0, 1, \ldots.., n-1$
    $z_{i+1} = z_i + hf(t_i, z_i) + \frac{h^2}{2}\frac{d}{dt}f(t_i, z_i)$
end $i$ loop
```

The fourth-order Taylor method becomes

$$z_{i+1} = z_i + hf(t_i, z_i) + \frac{h^2}{2}\frac{d}{dt}f(t_i, z_i) + \frac{h^3}{6}\frac{d^2}{dt^2}f(t_i, z_i) + \frac{h^4}{24}\frac{d^3}{dt^3}f(t_i, z_i) \tag{7.11}$$

This fourth-order Taylor method requires four function evaluations per time step. Euler's method needs one function evaluation per time step.

Example 2

Consider the following IVP

$$y' = y - t^2 + 1, \qquad 0 \le t \le 2, \quad y(0) = \tfrac{1}{2}$$

Solve numerically this IVP using Taylor's method of order two and four.

7.3 Runge-Kutta Methods

Runge-Kutta Methods offer a series of ODE solvers of various orders named after Carl Runge and Wilhelm Kutta. These methods overcome the disadvantage of Taylor methods which require the computation of the right hand side function f. Runge-Kutta Methods use the values of f directly without computing its derivatives. Taylor series in two variables is

$$f\left(x+h,y+k\right) \;=\; \sum_{j=0}^{\infty}\frac{1}{j!}\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)^{j} f(x,y)$$

Truncated Taylor series can be written as

$$\begin{aligned}
f\left(x+h,y+k\right) &= f(x,y)+\sum_{j=1}^{n-1}\frac{1}{j!}\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)^{j} f(x,y)+\frac{1}{n!}\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)^{n} f(\xi,\eta)\\
&= f(x,y)+\left(h\frac{\partial f}{\partial x}+k\frac{\partial f}{\partial y}\right)+\frac{1}{2!}\left(h^2\frac{\partial^2 f}{\partial x^2}+2hk\frac{\partial^2 f}{\partial x\partial y}+k^2\frac{\partial^2 f}{\partial y^2}\right)\\
&\quad +\frac{1}{3!}\left(h^3\frac{\partial^3 f}{\partial x^3}+3h^2k\frac{\partial^2 f}{\partial x^2\partial y}+3hk^2\frac{\partial^2 f}{\partial x\partial y^2}+k^3\frac{\partial^3 f}{\partial y^3}\right)+...\\
&\quad ....+\frac{1}{n!}\left(h\frac{\partial}{\partial x}+k\frac{\partial}{\partial y}\right)^{n} f(\xi,\eta) \qquad (7.12)
\end{aligned}$$

where (ξ,η) lies on the line segment obtained by joining $(x,y)$and $(x+h,y+k)$.

We can treat $f(x+h,y)$ and $f(x,y+k)$as special cases of (7.12).

Consider the following first-order initial value problem (IVP)

$$\begin{aligned}
\frac{dy}{dt} = y'(t) = f(t,y), &\qquad a \le t \le b \qquad (7.13)\\
y(a) = \rho_0 &
\end{aligned}$$

where ρ_0 is a constant. There are various versions of Runge-Kutta methods. We will discuss some of the common methods in the following sections.

Second-Order Runge-Kutta Method

Expanding $y(t)$ about h by using Taylor series and truncating, we obtain

$$y(t+h) \quad = \quad y\left(t\right)+hy'\left(t\right)+\frac{h^2}{2}y''\left(t\right)+O\left(h^3\right) \tag{7.14}$$

Using the total derivative as mention in the previous section, $y(t+h)$ becomes

$$\begin{aligned} y(t+h) &= y(t)+hf+\frac{h^2}{2}\left[\frac{\partial f}{\partial t}+f\,\frac{\partial f}{\partial y}\right]+O\left(h^3\right) \\ &= y(t)+\frac{1}{2}hf+\frac{1}{2}h\left[f+h\frac{\partial f}{\partial t}+hf\,\frac{\partial f}{\partial y}\right]+O\left(h^3\right) \quad (7.15) \end{aligned}$$

Since $f(t+h,\, y+hf)=f+h\frac{\partial f}{\partial t}+hf\,\frac{\partial f}{\partial y}+O\left(h^2\right)$ (by using Taylor series in two variables). Thus, we have

$$y(t+h)=y\left(t\right)+\frac{1}{2}hf(t,y)+\frac{1}{2}h\left[f\left(t+h,\, y+hf(t,y)\right)\right]+O\left(h^3\right) \tag{7.16}$$

Ignoring higher order terms and at time step $t_{i+1}=t_i+h$, we obtain

$$y\left(t_{i+1}\right) \quad \approx \quad z_{i+1}=z_i+\frac{h}{2}\left(K_1+K_2\right) \tag{7.17}$$

with

$$\begin{aligned} K_1 &= f(t_i,\, z_i) \\ K_2 &= f\left(t_i+h,\, z_i+hK_1\right) \end{aligned}$$

General formula for second-order Runge-Kutta method can be expressed as

$$y(t+h)=y(t)+c_1hf(t,y)+c_2hf\left(t+\alpha h,\, y+\beta hf(t,y)\right)+O\left(h^3\right) \tag{7.18}$$

where c_1, c_2, α and β are parameters.

Equation (7.18) is same as

$$y(t+h) \quad = \quad y(t)+c_1hf+c_2h\left[f+\alpha h\frac{\partial f}{\partial t}+\beta hf\frac{\partial f}{\partial x}\right]+O\left(h^3\right) \tag{7.19}$$

Thus the equivalent system is

$$y(t_{i+1}) \approx z_{i+1} = z_i + \sum_{i=1}^{2} c_i h K_i \tag{7.20}$$

with

$$\begin{aligned} K_1 &= f(t_i,\, z_i) \\ K_2 &= f\left(t_i + \alpha h,\, z_i + \beta h K_1\right) \end{aligned}$$

Comparing (7.15) and (7.19) we can derive the following conditions:

$$\begin{aligned} c_1 + c_2 &= 1 \\ c_2\alpha &= \frac{1}{2} \\ c_2\beta &= \frac{1}{2} \end{aligned} \tag{7.21}$$

One obvious choice is $c_1 = \frac{1}{2} = c_2$and $\alpha = 1 = \beta$. Putting these values in (7.20), we obtain (7.17). The second-order Runge-Kutta method is accurate to the second-order term and local truncation error is $O\left(h^3\right)$.

Modified Euler Method

Another set of parameters can be used is $c_1 = 0,\ c_2 = 1$ and $\alpha = \frac{1}{2} = \beta$. These parameters yield

$$z_{i+1} = z_i + hK_2 \tag{7.22}$$

where

$$\begin{aligned} K_1 &= f(t_i,\, z_i) \\ K_2 &= f\left(t_i + \frac{h}{2},\, z_i + \frac{hK_1}{2}\right) \end{aligned}$$

Third-Order Runge-Kutta Method

The third-order Runge-Kutta method is accurate to the third-order term and local truncation error is $O\left(h^4\right)$. The most common version of the third-order Runge-Kutta is given by

$$z_{i+1} = z_i + \frac{h}{6}\left[K_1 + 4K_2 + K_3\right] \tag{7.23}$$

where

$$\begin{aligned} K_1 &= f(t_i,\, z_i) \\ K_2 &= f\left(t_i + \frac{h}{2},\, z_i + \frac{hK_1}{2}\right) \\ K_3 &= f\left(t_i + h,\, z_i - hK_1 + 2hK_2\right) \end{aligned}$$

Fourth-Order Runge-Kutta Method (RK4)

The fourth-order Runge-Kutta method is accurate to the fourth-order term and local truncation error is $O\left(h^5\right)$. Formula for the classical fourth-order Runge-Kutta Method is

$$z_{i+1} = z_i + \frac{h}{6}\left[K_1 + 2K_2 + 2K_3 + K_4\right] \tag{7.24}$$

where

$$\begin{aligned} K_1 &= f(t_i,\, z_i) \\ K_2 &= f\left(t_i + \frac{h}{2},\, z_i + \frac{hK_1}{2}\right) \\ K_3 &= f\left(t_i + \frac{h}{2},\, z_i + \frac{hK_2}{2}\right) \\ K_4 &= f\left(t_i + h,\, z_i + hK_3\right) \end{aligned}$$

The method mentioned in equation (7.24) is the most popular version among all the Runge-Kutta methods.

Algorithm for the Fourth-Order Runge-Kutta Method (RK4)

Here is the algorithm for RK4.

```
$t = a$
$z = \rho_0$
$h = \frac{b-a}{N}$
print $0,\ t,\ z$
do $i = 1, ....., N$
    $K_1 = f\,(t,\ z)$
    $K_2 = f\left(t + \frac{h}{2},\ z + \frac{h}{2}K_1\right)$
    $K_3 = f\left(t + \frac{h}{2},\ z + \frac{h}{2}K_2\right)$
    $K_4 = f\,(t + h,\ z + hK_3)$
    $z = z + h\,(K_1 + 2K_2 + 2K_3 + K_4)\,/6$
    $t = a + ih$
    print $i,\ t,\ z$
end $i$ loop
```

Example 3

Consider the following IVP

$$y' = y - t^2 + 1, \qquad 0 \le t \le 2, \quad y(0) = \tfrac{1}{2}$$

Solve numerically this IVP using RK2 and RK4.

7.4 Systems and Higher-Order Ordinary Differential Equations

A system of n first-order ordinary differential equations

$$\left.\begin{array}{c} \frac{dy_1}{dt} = f_1\left(t, y_1, y_{2,} \ldots\ldots, y_n\right) \\ \frac{dy_2}{dt} = f_2\left(t, y_1, y_{2,} \ldots\ldots, y_n\right) \\ \vdots \\ \frac{dy_n}{dt} = f_n\left(t, y_1, y_{2,} \ldots\ldots, y_n\right) \end{array}\right\} \tag{7.25}$$

Consider an nth-order ordinary differential equation

$$\frac{d^n y}{dt^n} \quad = \quad F\left(t, y, \frac{dy}{dt}, \ldots\ldots\ldots\ldots\ldots, \frac{d^{n-1}y}{dt^{n-1}}\right) \tag{7.26}$$

Introducing the new variables $y_1, y_2, \ldots.., y_n$ such that

$$y_1 = y,\ y_2 = \frac{dy}{dt},\ y_3 = \frac{d^2y}{dt^2},\ \ldots\ldots.., y_{n-1} = \frac{d^{n-2}y}{dt^{n-2}},\ y_n = \frac{d^{n-1}y}{dt^{n-1}}$$

we obtain the following system of first-order ordinary differential equations

$$\begin{array}{ccc} \frac{dy_1}{dt} & = & \frac{dy}{dt} = y_2 \\ \frac{dy_2}{dt} & = & \frac{d^2y}{dt^2} = y_3 \\ \vdots & & \vdots \\ \frac{dy_{n-1}}{dt} & = & \frac{d^{n-1}y}{dt^{n-1}} = y_n \\ \frac{dy_n}{dt} & = & \frac{d^n y}{dt^n} = F \end{array}$$

Thus we obtain a system similar to (7.25) as presented below.

$$\left.\begin{array}{c} \frac{dy_1}{dt} = y_2 = f_1\left(t, y_1, y_{2,} \ldots\ldots, y_n\right) \\ \frac{dy_2}{dt} = y_3 = f_2\left(t, y_1, y_{2,} \ldots\ldots, y_n\right) \\ \vdots \\ \frac{dy_n}{dt} = F = f_n\left(t, y_1, y_{2,} \ldots\ldots, y_n\right) \end{array}\right\}$$

Conclusion here is that solving a higher-order ordinary differential equation is equivalent to solving a system of ordinary differential equations.

Example 4

Convert the following ODE

$$y^{(iv)} \;=\; 5yy'' - 2\left(y'\right)^3 + 3\cos t \tag{7.27}$$

into a system of first-order ODEs.

We set $y_1 = y$ and construct new variables such that

$$\begin{aligned} y_2 &= y' \\ y_3 &= y'' \\ y_4 &= y''' \end{aligned}$$

Then we obtain (by differentiating once)

$$\begin{aligned} y_1' &= y' = y_2 \\ y_2' &= y'' = y_3 \\ y_3' &= y''' = y_4 \\ y_4' &= y^{(iv)} = 5y_1y_3 - 2y_2^3 + 3\cos t \end{aligned} \tag{7.28}$$

Thus the solution $y(t)$ of the fourth-order ODE (7.27) can be obtained by solving the system (7.28) for $y_1(t)$, $y_2(t)$, $y_2(t)$ and $y_4(t)$.

Example 5

Convert the following higher-order system with independent variable t and dependent variables x and y:

$$\left.\begin{array}{c} \left(x'\right)^2 + x''y + 4x' = 3x + y^3 \\ xy'' + x'y - 2y' = xy^2 \end{array}\right\} \tag{7.29}$$

into a system of first-order ODEs.

We set new variables z_i such that $z_1 = x$, $z_2 = x'$, $z_3 = y$ and $z_4 = y'$. Then we obtain a first-order ODE system

$$\begin{aligned} z_1' &= x' = z_2 \\ z_2' &= x'' = \left(3z_1 + z_3^3 - z_2^2 - 4z_2\right)/z_3 \\ z_3' &= y' = z_4 \\ z_4' &= y'' = \left(z_1z_3^2 - z_2z_3 + 2z - 4\right)/z_1 \end{aligned} \tag{7.30}$$

So we will concentrate our discussion towards solving a system of ordinary differential equations in the following sections.

Vector Notation

It is very convenient to use vector notation to represent a system of ordinary differential equations. Consider the following nth-order system in the interval

$$\begin{aligned}
\frac{dy_1}{dt} &= f_1\left(t, y_1, y_2, \ldots\ldots, y_n\right) \\
\frac{dy_2}{dt} &= f_2\left(t, y_1, y_2, \ldots\ldots, y_n\right) \\
\vdots \quad & \quad \vdots \\
\frac{dy_n}{dt} &= f_n\left(t, y_1, y_2, \ldots\ldots, y_n\right)
\end{aligned} \tag{7.31}$$

with initial conditions

$$y_1(a) = \rho_1, y_2(a) = \rho_2, \quad , y_n(a) = \rho_n.$$

In vector notation, the above system (7.31) becomes

$$\frac{d\mathbf{Y}}{dt} = \mathbf{F}\left(t, \mathbf{Y}\right) \qquad \text{with} \qquad \mathbf{Y}\left(a\right) = \mathbf{A} \tag{7.32}$$

where

$$\frac{d\mathbf{Y}}{dt} = \mathbf{Y}' = \begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \\ \vdots \\ \frac{dy_n}{dt} \end{bmatrix} \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \qquad \mathbf{A} = \begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \end{bmatrix}$$

Taylor Series Method

Truncated Taylor series to each component $y_j,\ j = 1, 2, \ldots\ldots, n$ of $\mathbf{Y}$ yields

$$y_j(t+h) = y_j\left(t\right) + hy_j'\left(t\right) + \frac{h^2}{2}y_j''\left(t\right) + \frac{h^3}{3}y_j'''\left(t\right) + \ldots\ldots.. + \frac{h^n}{n!}y_j^{(n)}\left(t\right)$$

Thus, in vector notation, we have

$$\mathbf{Y}(t+h) = \mathbf{Y}\left(t\right) + h\mathbf{Y}'\left(t\right) + \frac{h^2}{2}\mathbf{Y}''\left(t\right) + \frac{h^3}{3}\mathbf{Y}'''\left(t\right) + \ldots\ldots.. + \frac{h^n}{n!}\mathbf{Y}^{(n)}\left(t\right)$$

The methods mentioned in section (7.13) can be used to solve a system of ordinary system of equations.

$$z_{i+1} = z_i + hf\left(t_i, z_i\right) + \frac{h^2}{2}\frac{d}{dt}f\left(t_i, z_i\right) + \frac{h^3}{6}\frac{d^2}{dt^2}f\left(t_i, z_i\right) + ... + \frac{h^n}{n!}\frac{d^{n-1}}{dt^{n-1}}f\left(t_i, z_i\right) \quad (7.33)$$

Fourth-Order Runge-Kutta Method

The fourth-order Runge-Kutta method is accurate to the fourth-order term and local truncation error is $O\left(h^5\right)$. Formula for the classical fourth-order Runge-Kutta Method is

$$\mathbf{Y}(t+h) \quad = \quad \mathbf{Y}(t) + \frac{h}{6}\left[\mathbf{K}_1 + 2\mathbf{K}_2 + 2\mathbf{K}_3 + \mathbf{K}_4\right] \qquad (7.34)$$

where

$$\begin{aligned} \mathbf{K}_1 &= \mathbf{F}\left(t,\, \mathbf{Y}\right) \\ \mathbf{K}_2 &= \mathbf{F}\left(t + \frac{h}{2},\, \mathbf{Y} + \frac{h\mathbf{K}_1}{2}\right) \\ \mathbf{K}_3 &= \mathbf{F}\left(t + \frac{h}{2},\, \mathbf{Y} + \frac{h\mathbf{K}_2}{2}\right) \\ \mathbf{K}_4 &= \mathbf{F}\left(t + h,\, \mathbf{Y} + h\mathbf{K}_3\right) \end{aligned}$$

The method mentioned in equation (7.24) is the most popular version among all the Runge-Kutta methods.

Fourth-Order Runge-Kutta (RK4) Algorithm to Solve a System of ODE

Here is the algorithm for RK4 method. Here M is the length of the vectors and N is the number of time steps..

```
$t = a$
$h = \frac{b-a}{N}$
do $j = 1, ....., M$
        $z_{,j} = \rho_j$
end $j$ loop
 print 0, $t$, $z_1, z_2, ......, z_M$
do $i = 1, ....., N$
```

```
        do $j = 1, ....., M$
            $K_{1,j} = f_j\left(t,\ z_1, z_2, ....., z_M\right)$
        end $j$ loop
        do $j = 1, ....., M$
            $K_{2,j} = f_j\left(t + \frac{h}{2},\ z_1 + \frac{h}{2}K_{1,1}, z_2 + \frac{h}{2}K_{1,2}, ....., z_M + \frac{h}{2}K_{1,M}\right)$
        end $j$ loop
        do $j = 1, ....., M$
            $K_{3,j} = f_j\left(t + \frac{h}{2},\ z_1 + \frac{h}{2}K_{2,1}, z_2 + \frac{h}{2}K_{2,2}, ....., z_M + \frac{h}{2}K_{2,M}\right)$
        end $j$ loop
        do $j = 1, ....., M$
            $K_{4,j} = f_j\left(t + h,\ z_1 + hK_{3,1}, z_2 + hK_{3,2}, ....., z_M + hK_{3,M}\right)$
        end $j$ loop
        do $j = 1, ....., M$
            $z_j = z_j + h\left(K_{1,j} + 2K_{2,j} + 2K_{3,j} + K_{4,j}\right)/6$
        end $j$ loop
        $t = a + ih$
        print $i,\ t,\ z_1, z_2, ......., z_M$
    end $i$ loop
```

Example

RK4

Exercise

1. Use separation of variables to find the solutions of the IVP with $y(0) = 1$ and the following differential equation

$$\text{(a)}\ \ y' = \frac{t^3}{y^2}, \qquad \text{(b)}\ \ y' = 2(t+1)y, \qquad \text{(c)}\ \ y' = \frac{1}{y^2}$$

2. Apply Euler's Method with step size $h = \frac{1}{4}$ to the following IVP on the interval [0, 1]

$$y' = \frac{t^3}{y^2} \qquad \text{with} \quad y(0) = 1.$$

Compare your results with exact solutions.

3. Apply Euler's Method with step size $h = \frac{1}{2}$ to the following IVP on the interval [0, 1]

$$y' = te^{3t} - 2y \qquad \text{with} \quad y(0) = 0.$$

Compare your results with exact solutions.

4. Compute $y(0.1)$ by solving the following IVP

$$y' = -ty^2 \qquad \text{with} \quad y(0) = 2.$$

with one step of the Taylor series method of order 2.

5. Apply Taylor's Method of Order 2 with step size $h = \frac{1}{2}$ to the following IVP on the interval [0, 1]

$$y' = te^{3t} - 2y \qquad \text{with} \quad y(0) = 0.$$

Compare your results with exact solutions.

6. Apply Taylor's Method of Order 4 with step size $h = \frac{1}{2}$ to the following IVP on the interval [0, 1]

$$y' = te^{3t} - 2y \qquad \text{with} \quad y(0) = 0.$$

Compare your results with exact solutions.

7. Apply Euler's Modified Method with step size $h = \frac{1}{2}$ to the following IVP on the interval [0, 1]

$$y' = te^{3t} - 2y \qquad \text{with} \quad y(0) = 0.$$

Compare your results with exact solutions.

8. Apply Fourth Order Runge-Kutta to the following IVP on the interval [0, 1]

$$y' = \frac{t^3}{y^2} \qquad \text{with} \quad y(0) = 1.$$

List the z_i, $i = 0, 1, ..., 4$ and find the error at $t = 1$ by comparing with the exact solutions.

9. Apply RK4 Method with step size $h = \frac{1}{2}$ to the following IVP on the interval [0, 1]

$$y' = te^{3t} - 2y \qquad \text{with} \quad y(0) = 0.$$

Compare your results with exact solutions.

10. Use the RK Method with $h = 0.25$ to approximate the solutions of the IVP

$$y' = 1 + \frac{y}{t} \qquad 1 \leq t \leq 2, \quad \text{with} \quad y(1) = 2.$$

Compare your result with the exact solution $y(t) = t \ln t + 2t$.

11. Apply Euler's Method with step size $h = \frac{1}{4}$ to the IVP system on [0, 1]

$$\begin{aligned} y_1' &= -y_1 - y_2, & y_1(0) &= 1 \\ y_2' &= y_1 - y_2, & y_2(0) &= 0 \end{aligned}$$

Find the analytical solutions and compare your numerical results with analytical results.

12. Apply RK4 Method with step size $h = \frac{1}{2}$ to the IVP system on [0, 1]

$$\begin{aligned} y_1' &= y_2, & y_1(0) &= 1 \\ y_2' &= -y_1 - 2e^t + 1, & y_2(0) &= 0 \\ y_3' &= -y_1 - e^t + 1, & y_3(0) &= 1 \end{aligned}$$

Find the analytical solutions and compare your numerical results with analytical results.

13. Apply Euler's Method to approximate the solution of the following second order ODE with $h = 0.1$

$$y'' - 2y' + y = te^t - 1, \qquad 0 \leq t \leq 1, \quad y(0) = y'(0) = 0.$$

Find the analytical solutions and compare your numerical results with analytical results.

14. Apply RK4 Method to approximate the solution of the following second order ODE with $h = 0.1$

$$y'' - 2y' + y = te^t - 1, \qquad 0 \leq t \leq 1, \quad y(0) = y'(0) = 0.$$

Compare your numerical results with analytical results (from Q# 13).

15. Apply RK4 Method to approximate the solution of the following third order ODE with $h = 0.2$

$$y''' + y'' - 4y' - 4 = 0, \qquad 0 \leq t \leq 2, \quad y(0) = 3,\ y'(0) = -1,\ y''(0) = 9.$$

Find the analytical solutions and compare your numerical results with analytical results.

Chapter 8

Boundary Value Problems (BVP)

Here we consider two point boundary value problems. Let us consider the following second-order ordinary differential equation

$$y'' = f\,(x, y, y') \qquad\qquad a \leq x \leq b \tag{8.1}$$

$$\begin{aligned} \alpha_1 y(a) + \alpha_2 y'(a) &= \alpha_3 \\ \beta_1 y(b) + \beta_2 y'(b) &= \beta_3 \end{aligned} \tag{8.2}$$

where $y = y(x)$ and $y' = \frac{dy}{dx},\ y'' = \frac{d^2y}{dx^2}$. Here $\alpha_i,\ \beta_i,\ i = 1, 2, 3$ are constants.

8.1 Classification of Boundary Value Problems

When $f\,(x, y, y')$ can be expressed as

$$f\,(x, y, y') \quad = \quad p(x)y' + q(x)y + r(x)$$

then the equation (8.1) is called **linear**, otherwise it is **nonlinear**.

Dirichlet Boundary Condition

A boundary condition which specifies the value of the dependent variable at a boundary, i.e., a boundary condition of the type

$$y(a) = \alpha \quad \text{or} \quad y(b) = \beta \tag{8.3}$$

is known as the Dirichlet boundary condition.

Neumann Boundary Condition

A boundary condition which specifies the value of the derivative of the dependent variable at a boundary, i.e., a boundary condition of the type

$$y'(a) = \alpha \quad \text{or} \quad y'(b) = \beta \tag{8.4}$$

is known as the Neumann boundary condition.

Robin or Mixed Boundary Condition

A boundary condition which specifies the value of the the linear combination of the dependent variable and its derivative at a boundary, i.e., a boundary condition of the type

$$\alpha_1 y(a) + \alpha_2 y'(a) = \alpha_3 \quad \text{or} \quad \beta_1 y(b) + \beta_2 y'(b) = \beta_3 \tag{8.5}$$

is known as the Robin or Mixed boundary condition.

8.2 The Shooting Method

Consider a two-point boundary value problem

$$\begin{aligned} y'' = f\left(x, y, y'\right) \qquad & x \in [a, b] \\ y(a) = \alpha, \quad y(b) = \beta & \end{aligned} \tag{8.6}$$

Since we are familiar with the methods to solve an initial value problem, to solve a boundary value problem, we can consider a corresponding initial value problem and then march the solution to the other end of the boundary and check with the other given condition at that boundary. Assuming an initial value $y'(a)$, solve the IVP and check the final value $y(b)$. If $y(b) = \beta$, we are done. Otherwise, the value of $y'(a)$ is modified and repeat this process hoping to achieve the final value to match with β. That is why this method is called a shooting method. This procedure transform the boundary value problem to a root finding problem. For nonlinear problem the above approach will be used.

8.2.1 Linear Problems

For linear problem, the above procedure is slightly different in the sense that we can obtain the solution in some direct way. For linear non-homogeneous BVP, the solution can be written as a linear combination of the particular solution and a solution to the corresponding homogeneous BVP.

Theorem

Consider the following two point linear boundary value problem (BVP) with Dirichlet boundary condition

$$\begin{aligned} y'' = p(x)y' + q(x)y + r(x) \qquad & x \in [a, b] \\ y(a) = \alpha, \quad y(b) = \beta & \end{aligned} \tag{8.7}$$

If $p(x)$, $q(x)$ and $r(x)$ are continuous on $[a, b]$, and $q(x) > 0$ on $[a, b]$, then the above BVP has a unique solution.

The BVP in (8.7) is solved by considering two initial value problems (IVPs) as follows:

First IVP:

$$\begin{array}{c} y_1'' = p(x)y_1' + q(x)y_1 + r(x) \qquad x \in [a,b] \\ y_1(a) = \alpha, \quad y_1'(a) = 0 \end{array} \tag{8.8}$$

Second IVP:

$$\begin{array}{c} y_2'' = p(x)y_2' + q(x)y_2 \qquad x \in [a,b] \\ y_2(a) = 0, \quad y_2'(a) = 1 \end{array} \tag{8.9}$$

Now we consider a linear combination of the solution the first IVP and the solution of the second IVP as follows

$$y(x) \;=\; y_1(x) + Ay_2(x) \tag{8.10}$$

and we determine the constant A from the boundary condition $y(b) = \beta$. Thus we get

$$A \;=\; \frac{y(b) - y_1(b)}{y_2(b)} = \frac{\beta - y_1(b)}{y_2(b)} \tag{8.11}$$

Hence the solution of (8.7) is given by

$$y(x) \;=\; y_1(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2(x) \tag{8.12}$$

where $y_1(x)$ and $y_2(x)$ are solutions of the first IVP (8.8) and the second IVP (8.9) respectively.

Now we show that $y(x)$ presented in (8.12) satisfies our original differential equation and the boundary conditions (8.7).

First let us check the differential equation. Differentiating (8.12) we obtain

$$\begin{aligned} y'(x) &= y_1'(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2'(x) \\ y''(x) &= y_1''(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2''(x) \end{aligned}$$

Now (8.8) and (8.9) yield

$$
\begin{aligned}
y''(x) &= y_1''(x) + \frac{\beta - y_1(b)}{y_2(b)} y_2''(x) \\
&= p(x)y_1' + q(x)y_1 + r(x) + \frac{\beta - y_1(b)}{y_2(b)} \left[p(x)y_2' + q(x)y_2\right] \\
&= p(x)\left[y_1' + \frac{\beta - y_1(b)}{y_2(b)} y_2'\right] + q(x)\left[y_1 + \frac{\beta - y_1(b)}{y_2(b)} y_2\right] + r(x) \\
&= p(x)y' + q(x)y + r(x)
\end{aligned}
$$

Now we check the boundary conditions. At $x = a$,we have

$$
\begin{aligned}
y(a) &= y_1(a) + \frac{\beta - y_1(b)}{y_2(b)} y_2(a) \\
&= \alpha + \frac{\beta - y_1(b)}{y_2(b)}.0 = \alpha
\end{aligned}
$$

At $x = b$, we get

$$
\begin{aligned}
y(b) &= y_1(b) + \frac{\beta - y_1(b)}{y_2(b)} y_2(b) \\
&= y_1(b) + \beta - y_1(b) = \beta
\end{aligned}
$$

Thus the linear combination of the solutions of the two IVPs given by (8.8) and (8.9) satisfies our original BVP (8.7).

Other boundary conditions

Consider a more general boundary condition (like Robin condition) at $x = b$

$$
\begin{gathered}
y'' = p(x)y' + q(x)y + r(x) \qquad x \in [a, b] \\
y(a) = \alpha, \quad \beta_1 y(b) + \beta_2 y'(b) = \beta_3
\end{gathered}
\tag{8.13}
$$

To solve the above problem, we use the same IVPs mentioned in the previous section because the condition at $x = a$ remains the same. We obtain a solution as a linear combination of the solutions of these two IVPs which can be written as

$$
y(x) \;=\; y_1(x) + Ay_2(x)
$$

and at $x = b$, we have

$$y(b) = y_1(b) + Ay_2(b) \qquad \text{and} \qquad y'(b) = y_1'(b) + Ay_2'(b) \tag{8.14}$$

Now we are given that at $x = b$,

$$\beta_1 y(b) + \beta_2 y'(b) \;=\; \beta_3 \tag{8.15}$$

i.e.,

$$\beta_1 \left[y_1(b) + Ay_2(b)\right] + \beta_2 \left[y_1'(b) + Ay_2'(b)\right] \;=\; \beta_3$$

which yields us

$$A \;=\; \frac{\beta_3 - \beta_1 y_1(b) - \beta_2 y_1'(b)}{\beta_1 y_2(b) + \beta_2 y_2'(b)} \tag{8.16}$$

Thus the solution of (8.13) becomes

$$y(x) \;=\; y_1(x) + \frac{\beta_3 - \beta_1 y_1(b) - \beta_2 y_1'(b)}{\beta_1 y_2(b) + \beta_2 y_2'(b)} y_2(x) \tag{8.17}$$

If we put $\beta_2 = 0$ and $\frac{\beta_3}{\beta_1} = \beta$, we obtain the solution of (8.7).

If we put $\beta_1 = 0$ and and $\frac{\beta_3}{\beta_2} = \beta$, the BVP becomes

$$\begin{array}{c} y'' = p(x)y' + q(x)y + r(x) \qquad x \in [a, b] \\ y(a) = \alpha, \quad y'(b) = \beta \end{array} \tag{8.18}$$

and the constant A is given by

$$A \;=\; \frac{\beta - y_1'(b)}{y_2'(b)} \tag{8.19}$$

Thus the solution is

$$y(x) \;=\; y_1(x) + \frac{\beta - y_1'(b)}{y_2'(b)} y_2(x) \tag{8.20}$$

Neumann condition at $x = a$

To solve the following BVP with Neumann condition at $x = a$, we consider two initial value problems as mentioned below.

First IVP:

$$\begin{gathered} y_1'' = p(x)y_1' + q(x)y_1 + r(x) \qquad x \in [a,b] \\ y_1(a) = 0, \quad y_1'(a) = \alpha \end{gathered} \tag{8.21}$$

Second IVP:

$$\begin{gathered} y_2'' = p(x)y_2' + q(x)y_2 \qquad x \in [a,b] \\ y_2(a) = 1, \quad y_2'(a) = 0 \end{gathered} \tag{8.22}$$

Linear combination of the solutions of (8.21) and (8.22), i.e., $y(x) = y_1(x) + Ay_2(x)$ will be a solution of the BVP. The value of the constant A is given by (8.11), (8.19) or (8.16) depending on the boundary condition at $x = b$ is a Dirichlet condition, a Neumann condition, or a Robin condition.

Example

Consider BVP

$$y'' = -\frac{2}{x}y' + \frac{2}{x^2}y + \frac{\sin(\ln x)}{x^2}, \qquad 1 \le x \le 2, \quad y(1) = 1, \quad y(2) = 2$$

Exact solution is

$$y(x) = Ax + \frac{B}{x} - \frac{3}{10}\sin(\ln x) - \frac{1}{10}\cos(\ln x)$$

where $B = \frac{1}{70}\left[8 - 12\sin(\ln 2) - 4\cos(\ln 2)\right] \simeq -0.03920701$ and $A = \frac{11}{10} - B \simeq 1.13920701$.

Use Linear Shooting method with $h = 0.1$.

8.2.2 Nonlinear Problems

Consider the following nonlinear second-order BVP

$$\begin{gathered} y'' = f\left(x, y, y'\right) \qquad x \in [a,b] \\ y(a) = \alpha, \quad y(b) = \beta \end{gathered} \tag{8.23}$$

To solve this nonlinear BVP, we consider initial value problems of type

$$\begin{array}{c} y'' = f\left(x, y, y'\right) \qquad x \in [a, b] \\ y(a) = \alpha, \quad y'(a) = z \end{array} \tag{8.24}$$

We obtain a sequence of solutions of these IVPs involving a parameter z. We choose the parameters $z = z_k$ such that

$$\lim_{n\to\infty} y\left(b,\, z_n\right) \;=\; y(b) = \beta$$

where $y(x)$ is the solution of (8.23) and $y\left(x, z_n\right)$ is the solution of (8.24).

We start with the parameter z_0 and solve the IVP to obtain the solution $y\left(b,\, z_0\right)$. If $y\left(b,\, z_0\right)$ is not close to β, we modify the parameter it to z_1, z_2 so on, until $y\left(b,\, z_n\right)$ is close to β. To determine the next z, we find a zero of $y(b, z) - \beta$, i.e., find a root of

$$y(b,\, z) - \beta \;=\; 0$$

which is a root finding problem for a nonlinear equation. Thus Secant or Newton's methods can be used to achieve this goal.

8.2.2.1 Nonlinear Shooting using Secant Method

Here we assume two initial approximations z_0 and z_1 and the Secant method allows us to write

$$z_{n+1} \;=\; z_n + \frac{z_n - z_{n-1}}{y\left(b, z_n\right) - y\left(b, z_{n-1}\right)}\left[\beta - y\left(b, z_n\right)\right] \tag{8.25}$$

Then the result $y\left(b,\, z_{n+1}\right)$ is compared with β.

8.2.2.2 Nonlinear Shooting using Newton's Method

Here we need only one initial approximation z_0. In this case, using Newton's method we can write

$$z_{n+1} \;=\; z_n + \frac{\beta - y\left(b, z_n\right)}{\left.\frac{dy(b,z)}{dz}\right]_{z=z_n}}$$

Since the explicit form of $y(b,z)$ is not known, evaluation of $\left.\frac{dy(b,z)}{dz}\right]_{z=z_n}$ is not easy. To use Newton's method we follow the procedure mentioned below by considering two IVPs.

First initial value problem is

$$\begin{gathered} y''(x,z) = f\left(x, y(x,z), y'(x,z)\right) \qquad x \in [a,b] \\ y(a,z) = \alpha, \quad y'(a,z) = z \end{gathered} \tag{8.26}$$

Second initial value problem is

$$\begin{gathered} w''(x,z) = \frac{\partial f(x,y,y')}{\partial y} w(x,z) + \frac{\partial f(x,y,y')}{\partial y'} w'(x,z) \qquad x \in [a,b] \\ y(a,z) = 0, \quad y'(a,z) = 1 \end{gathered} \tag{8.27}$$

The sequence $\{z_n\}$ is obtained by using

$$z_{n+1} \quad = \quad z_n + \frac{\beta - y\left(b, z_n\right)}{w\left(b, z_n\right)}$$

8.3 Finite Difference Method

Nonlinear second-order two point boundary value problem is given by

$$y'' = f\left(x, y, y'\right) \qquad a \le x \le b \tag{8.28}$$

$$\begin{aligned} \alpha_1 y(a) + \alpha_2 y'(a) = \alpha_3 \\ \beta_1 y(b) + \beta_2 y'(b) = \beta_3 \end{aligned} \tag{8.29}$$

8.3.1 Linear Problems

The linear counterpart of the BVP presented in (8.28) with Dirichlet boundary condition is

$$\begin{aligned} y'' = p(x)y' + q(x)y + r(x) \qquad x \in [a, b] \\ y(a) = \alpha, \quad y(b) = \beta \end{aligned} \tag{8.30}$$

Consider a partition of $[a, b]$ with the nodes $x_0, x_1, ..., x_{N-1}, x_N$ at equal spacing such that

$$a = x_0 < x_1 < x_2 < .... < x_{N-1} < x_N = b$$

with uniform spacing $h = \frac{b-a}{N}$ and $x_i = a + ih$. The nodes x_0 and x_N are called boundary nodes and $x_1, x_2, ..., x_{N-1}$ are known as interior nodes. Let $y(x)$ denote the exact value of the solution of the BVP (8.30), and $y_i = y\left(x_i\right)$ denote the exact value at $x = x_i$. Also we use z_i to represent the finite difference approximation to y_i.

Let us evaluate the differential equation at each interior node x_i, i.e.,

$$\left[y'' = p(x)y' + q(x)y + r(x\right]_{x=x_i} \qquad 1 \le i \le N-1$$

Now using second-order central difference scheme, we have

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + O\left(h^2\right) = p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i + r_i + O\left(h^2\right) \tag{8.31}$$

Considering the approximations z_i, truncating the higher-order terms we derive for $1 \le i \le N-1$

$$\frac{z_{i+1} - 2z_i + z_{i-1}}{h^2} = p_i \frac{z_{i+1} - z_{i-1}}{2h} + q_i z_i + r_i \tag{8.32}$$

Thus we have $(N-1)$ equations with $(N+1)$ approximations, namely, $z_0, z_1, ..., z_{N-1}, z_N$. So we need two more equations. These two extra equations can be derived from the boundary conditions at the exterior nodes, $x_0 = a$, $x_N = b$ and we will use $z_0 = \alpha$ and $z_N = \beta$. Thus we will need to find $(N-1)$ approximations $z_1, z_{2,.}......., z_{N-1}$. The equation (8.32) yields

$$z_{i+1} - 2z_i + z_{i-1} - \frac{h}{2} p_i \left[z_{i+1} - z_{i-1} \right] - h^2 q_i z_i \quad = \quad h^2 r_i$$

which can be written as (for $i = 1, 2, ..., N-1$)

$$\left[-\frac{h}{2} p_i - 1 \right] z_{i-1} + \left[h^2 q_i + 2 \right] z_i + \left[\frac{h}{2} p_i - 1 \right] z_{i+1} \quad = \quad -h^2 r_i \qquad (8.33)$$

Putting $i = 1$ and $i = N-1$ respectively in (8.33), we have

$$\begin{aligned} \left[-\frac{h}{2} p_1 - 1 \right] z_0 + \left[h^2 q_1 + 2 \right] z_1 + \left[\frac{h}{2} p_1 - 1 \right] z_2 \quad &= \quad -h^2 r_1 \\ \left[-\frac{h}{2} p_{N-1} - 1 \right] z_{N-2} + \left[h^2 q_{N-1} + 2 \right] z_{N-1} + \left[\frac{h}{2} p_{N-1} - 1 \right] z_N \quad &= \quad -h^2 r_{N-1} \end{aligned}$$

Since $z_0 = \alpha$ and $z_N = \beta$, from the two above equations we observe that

$$\begin{aligned} \left[h^2 q_1 + 2 \right] z_1 + \left[\frac{h}{2} p_1 - 1 \right] z_2 \quad &= \quad -h^2 r_1 + \left[\frac{h}{2} p_1 + 1 \right] \alpha \\ \left[-\frac{h}{2} p_{N-1} - 1 \right] z_{N-2} + \left[h^2 q_{N-1} + 2 \right] z_{N-1} \quad &= \quad -h^2 r_{N-1} + \left[1 - \frac{h}{2} p_{N-1} \right] \beta \end{aligned}$$

Finally we need to solve a system of equations for $z_1, z_2, ..., z_{N-1}$which in matrix form

$$A\mathbf{z} \quad = \quad \mathbf{b} \qquad (8.34)$$

where

$$A = \begin{bmatrix} d_1 & u_1 & 0 & \cdots & \cdots & \cdots & 0 \\ l_2 & d_2 & u_2 & 0 & & & \vdots \\ 0 & l_3 & d_3 & u_3 & & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & 0 & l_{N-3} & d_{N-3} & u_{N-3} & 0 \\ \vdots & & & 0 & l_{N-2} & d_{N-2} & u_{N-2} \\ 0 & \cdots & \cdots & \cdots & 0 & l_{N-1} & d_{N-1} \end{bmatrix} \tag{8.35}$$

with

$$l_i = -\frac{h}{2}p_i - 1, \qquad d_i = h^2 q_i + 2, \qquad u_i = \frac{h}{2}p_i - 1 \tag{8.36}$$

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \vdots \\ \\ z_{N-3} \\ z_{N-2} \\ z_{N-1} \end{bmatrix} \qquad \text{and} \qquad \mathbf{b} = \begin{bmatrix} -h^2 r_1 + \left(\frac{h}{2}p_1 + 1\right)\alpha \\ -h^2 r_2 \\ -h^2 r_3 \\ \vdots \\ -h^2 r_{N-3} \\ -h^2 r_{N-2} \\ -h^2 r_{N-1} + \left(1 - \frac{h}{2}p_{N-1}\right)\beta \end{bmatrix} \tag{8.37}$$

Other Boundary Conditions

Now we consider the second-order linear two point boundary value problem

$$y'' = p(x)y' + q(x)y + r(x) \qquad a \le x \le b \tag{8.38}$$

$$\begin{aligned} \alpha_1 y(a) + \alpha_2 y'(a) &= \alpha_3 \\ \beta_1 y(b) + \beta_2 y'(b) &= \beta_3 \end{aligned} \tag{8.39}$$

Exercise

1. Consider the boundary value problem

   $$y'' = y' + 2y + \cos x, \quad 0 \le x \le \frac{\pi}{2}, \quad y(0) = -0.3,\ y(\pi/2) = -0.1.$$

Use the Linear Shooting method to approximate the solution. Find the exact solution (analytical) solution. Compare your numerical result with the actual (exact) solution. First use $h = \frac{\pi}{4}$ and then use $h = \frac{\pi}{8}$.

2. Consider the BVP

$$y'' = -3y' + 2y + 2x + 3, \quad 0 \le x \le 1, \quad y(0) = 2,\ y(1) = 1; \quad h = 0.1.$$

Use the Linear Shooting method to approximate the solution. Find the exact solution solution. Compare your numerical result with the exact solution.

3. Consider the following nonlinear BVP

$$y'' = y^3 - yy', \quad 1 \le x \le 2, \quad y(1) = \frac{1}{2},\ y(2) = \frac{1}{3}; \quad h = 0.1.$$

This BVP has analytical solution $y(x) = (1+x)^{-1}$. Use the Nonlinear Shooting method with TOL$=10^{-4}$ to approximate the solution. Compare your numerical result with the exact solution.

4. Consider the boundary value problem

$$y'' = y' + 2y + \cos x, \quad 0 \le x \le \frac{\pi}{2}, \quad y(0) = -0.3,\ y(\pi/2) = -0.1.$$

Use the Linear Finite-Difference method to approximate the solution. Compare your numerical result with the actual (exact) solution. First use $h = \frac{\pi}{4}$ and then use $h = \frac{\pi}{8}$.