Contents

Chapter 1

Introduction

Numerical analysis is concerned with the development and analysis of methods for the numerical solution of practical problems. Traditionally, these methods have been mainly used to solve problems in the physical sciences and engineering. However, they are finding increasing relevance in a much broader range of subjects including economics and business studies.

The first stage in the solution of a particular problem is the formulation of a mathematical model. Mathematical symbols are introduced to represent the variables involved and physical (or economic) principles are applied to derive equations which describe the behavior of these variables. Unfortunately, it is often impossible to find the exact solution of the resulting mathematical problem using standard techniques. In fact, there are very few problems for which an analytical solution can be determined. For example, there are formulas for solving quadratic, cubic and quartic polynomial equations, but no such formula exists for polynomial equations of degree greater than four or even for a simple equation such as

$$x = \cos(x)$$

Similarly, we can certainly evaluate the integral

$$A = \int_a^b e^x dx$$

as $e^a - e^b$, but we cannot find the exact value of

$$A = \int_a^b e^{x^2} dx$$

since no function exists which differentiates to e^{x^2}. Even when an analytical solution can be found it may be of more theoretical than practical use. For example, if the solution of a differential equation

$$y'' = f(x, y, y')$$

is expressed as an infinite sum of Bessel functions, then it is most unsuitable for calculating the numerical value of y corresponding to some numerical value of x.

1.1 Errors

Computations generally yield approximations as their output. This output may be an approximation to a true solution of an equation, or an approximation of a true value of some quantity. Errors are commonly measured in one of two ways: absolute error and relative error as the following definition.

Definition 1. *If x_A is an approximation to x, the **error** is defined as*

$$err(x_A) = x_T - x_A \tag{1.1}$$

*The **absolute error** is defined as*

$$Aerr(x_A) = |err(x_A)| = |x_T - x_A| \tag{1.2}$$

*And the **relative error** is given by*

$$rel(x_A) = \frac{Absolute\ error}{True\ value} = \frac{|x_T - x_A|}{x_T}, \qquad x_T \neq 0 \tag{1.3}$$

Note that if the true value happens to be zero, $x = 0$, the relative error is regarded as undefined. The relative error is generally of more significance than the absolute error.

Example 1.1. *Let* $x_T = \frac{19}{7} \approx 2.714285$ *and* $x_A = 2.718281$. *Then*

$$err\,(x_A) = x_T - x_A = \frac{19}{7} - 2.718281 \approx -0.003996$$

$$Aerr\,(x_A) = |err\,(x_A)| \approx 0.003996$$

$$rel\,(x_A) = \frac{Aerr\,(x_A)}{x_T} = \frac{0.003996}{2.718281} \approx 0.00147$$

Example 1.2. *Consider the following table*

x_T	x_A	*Absolute Error*	*Relative Error*
1	*0.99*	*0.01*	*0.01*
1	*1.1*	*0.01*	*0.01*
–1.5	*–1.2*	*0.3*	*0.2*
100	*99.99*	*0.01*	*0.0001*
100	*99*	*1*	*0.01*

Example 1.3. *Consider two different computations. In the first one, an estimate* $x_A = 0.003$ *is obtained for the true value* $x_T = 0.004$. *In the second one,* $y_A = 1238$ *for* $y_T = 1258$. *Therefore, the absolute errors are*

$$Aerr(x_A) = |x_T - x_A| = 0.001, \qquad Aerr(y_A) = |y_T - y_A| = 20$$

The corresponding relative errors are

$$rel(x_A) = \frac{Aerr(x_A)}{x_T} = \frac{0.001}{0.004} = 0.25,$$

$$rel(y_A) = \frac{Aerr(y_A)}{y_T} = \frac{20}{1258} = 0.0159$$

We notice that the absolute errors of 0.001 *and* 20 *can be rather misleading, judging by their magnitudes. In other words, the fact that* 0.001 *is much smaller than* 20 *does not make the first error a smaller error relative to its corresponding computation. In fact, looking at the relative errors, we see that* 0.001 *is associated with a* 25% *error, while* 20 *corresponds to* 1.59% *error, much smaller than the first. Because they convey a more specific type of information, relative errors are considered more significant than absolute errors.*

1.2 Computational and Errors

Numerical methods are procedures that allow for efficient solution of a mathematically formulated problem in a finite number of steps to within an arbitrary precision. Computers are needed in most cases. A very important issue here is the errors caused in computations.
A numerical algorithm consists of a sequence of arithmetic and logical operations which produces an approximate solution to within any prescribed accuracy. There are often several different algorithms for the solution of any one problem. The particular algorithm chosen depends on the context from which the problem is taken. In economics, for example, it may be that only the general behavior of a variable is required, in which case a simple, low accuracy method which uses only a few calculations is appropriate. On the other hand, in precision engineering, it may be essential to use a complex, highly accurate method, regardless of the total amount of computational effort involved.
Once a numerical algorithm has been selected, a computer

program is usually written for its implementation. The program is run to obtain numerical results, although this may not be the end of the story. The computed solution could indicate that the original mathematical model needs modifying with a corresponding change in both the numerical algorithm and the program.
Although the solution of 'real problems' by numerical techniques involves the use of a digital computer or calculator, Determination of the eigenvalues of large matrices, for example, did not become a realistic proposition until computers became available because of the amount of computation involved. Nowadays any numerical technique can at least be demonstrated on a microcomputer, although there are some problems that can only be solved using the speed and storage capacity of much larger machines.

There exist three possible sources of error:

1. **Errors in the formulation of the problem** to be solved.
   (a) Errors in the mathematical model. For example, when simplifying assumptions are made in the derivation of the mathematical model of a physical system. (Simplifications).
   (b) Error in input data. (Measurements).
2. **Approximation errors**
   (a) Discretization error.
   (b) Convergence error in iterative methods.
   (c) Discretization/convergence errors may be estimated by an analysis of the method used.
3. **Roundoff errors**: This error is caused by the computer representation of numbers.

(a) Roundoff errors arise everywhere in numerical computation because of the finite precision arithmetic.

(b) Roundoff errors behave quite unorganized.

4. **Truncation error**: Whenever an expression is approximated by some type of a mathematical method. For example, suppose we use the Maclaurin series representation of the sine function:

$$\sin\alpha = \sum_{n=odd}^{\infty} \frac{(-1)^{\frac{(n-1)}{2}}}{n!}\alpha^n = \alpha - \frac{1}{3!}\alpha^3 + \frac{1}{5!}\alpha^5 - \cdots + \frac{(-1)^{\frac{(m-1)}{2}}}{3!}\alpha^m + E_m$$

where E_m is the tail end of the expansion, neglected in the process, and known as the truncation error.

1.3 ERRORS AND STABILITY

The majority of numerical methods involve a large number of calculations which are best performed on a computer or calculator. Unfortunately, such machines are incapable of working to infinite precision and so small errors occur in nearly every arithmetic operation. Even an apparently simple number such as $2/3$ cannot be represented exactly on a computer. This number has a non-terminating decimal expansion

$$0.66666666666666\cdots$$

and if, for example, the machine uses ten-digit arithmetic, then it is stored as

$$0.666\,666\,666\,7$$

(In fact, computers use binary arithmetic. However, since the substance of the argument is the same in either case, we restrict our attention to decimal arithmetic for simplicity).

The difference between the exact and stored values is called the rounding error which, for this example, is

$$-0.000\,000\,000\,033\,33...$$

Suppose that for a given real number α the digits after the decimal point are

$$d_1 d_2 \cdots d_n d_{n+1} \cdots$$

To round α to n decimal places (abbreviated to nD) we proceed as follows. If $d_{n+1} < 5$, then α is rounded down; all digits after the nth place are removed. If $d_{n+1} \geq 5$, then α is rounded up; d_n is increased by one and all digits after the nth place are removed. It should be clear that in either case the magnitude of the rounding error does not exceed 0.5×10^{-n}.

In most situations the introduction of rounding errors into the calculations does not significantly affect the final results. However, in certain cases it can lead to a serious loss of accuracy so that computed results are very different from those obtained using exact arithmetic. The term instability is used to describe this phenomenon.

There are two fundamental types of instability in numerical analysis - **inherent** and **induced**. The first of these is a fault of the problem, the second of the method of solution.

Definition 2. *A problem is said to be **inherently unstable** (or **ill - conditioned**) if small changes in the data of the problem cause large changes in its solution.*

This concept is important for two reasons. Firstly, the data may be given as a set of readings from an analogue device such as a thermometer or voltmeter and as such cannot be measured exactly. If the problem is ill-conditioned then any numerical results, irrespective of the method used to

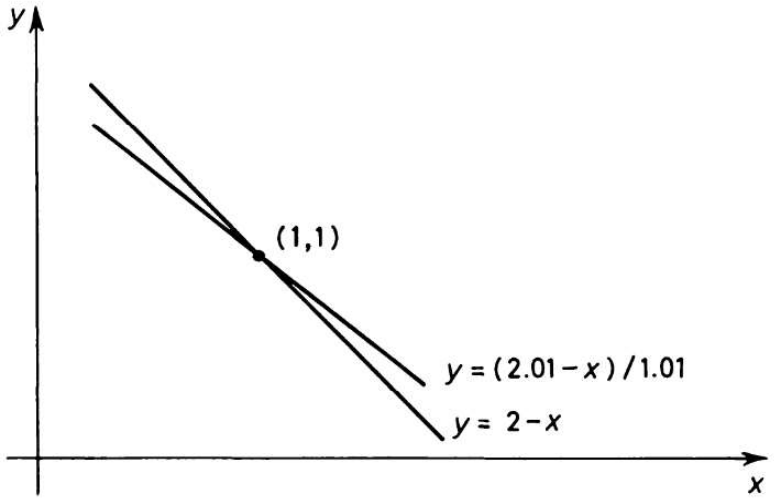

Figure 1.1: sketche of example 1.4

obtain them, will be highly inaccurate and may be worthless. The second reason is that even if the data is exact it will not necessarily be stored exactly on a computer. Consequently, the problem which the computer is attempting to solve may differ slightly from the one originally posed. This does not usually matter, but if the problem is ill-conditioned then the computed results may differ wildly from those expected.

Example 1.4. *Consider the simultaneous linear equations*

$$x + y = 2$$
$$x + 1.01y = 2.01$$

which have solution $x = y = 1$*. If the number* 2.01 *is changed to* 2.02*, the corresponding solution is* $x = 0$*,* $y = 2$*. We see that a* 0.5% *change in the data produces a* 100% *change in the solution. It is instructive to give a geometrical interpretation of this result. The solution of the system is the point of intersection of the two lines* $y = 2 - x$ *and* $y = (2.01 - x)/1.01$*. These lines are sketched in figure 1.1. It is clear that the point of*

intersection is sensitive to small movements in either of these lines since they are nearly parallel. In fact, if the coefficient of y in the second equation is 1.00, the two lines are exactly parallel and the system has no solution. This is fairly typical of ill-conditioned problems. They are often close to 'critical' problems which either possess infinitely many solutions or no solution whatsoever.

Example 1.5. *Consider the initial value problem*

$$y'' - 10y' - 11y = 0; \quad y(0) = 1, \quad y'(0) = -1$$

defined on $x \geq 0$. The corresponding auxiliary equation has roots -1 and 11, so the general solution of the differential equation is

$$y = Ae^{-x} + Be^{11x}$$

for arbitrary constants A and B. The particular solution which satisfies the given initial conditions is

$$y = e^{-x}$$

Now suppose that the initial conditions are replaced by

$$y(0) = 1 + \delta, \qquad y'(0) = -1 + \epsilon$$

for some small numbers δ and ϵ. The particular solution satisfying these conditions is

$$y = \left(1 + \frac{11\delta}{12} - \frac{\epsilon}{12}\right) e^{-x} + \left(\frac{\delta}{12} + \frac{\epsilon}{12}\right) e^{11x}$$

and the change in the solution is therefore

$$\left(\frac{11\delta}{12} - \frac{\epsilon}{12}\right) e^{-x} + \left(\frac{\delta}{12} + \frac{\epsilon}{12}\right) e^{11x}$$

The term $\dfrac{(\delta + \epsilon)e^{11x}}{12}$ is large compared with e^{-x} for $x > 0$, indicating that this problem is ill-conditioned.

To inherent stability depends on the size of the solution to the original problem as well as on the size of any changes in the data. Under these circumstances, one would say that the problem is ill-conditioned.

We now consider a different type of instability which is a consequence of the method of solution rather than the problem itself.

Definition 3. *A method is said to suffer from* ***induced instability*** *if small errors present at one stage of the method lead to bad effect in subsequent stages to such final results are totally inaccurate.*

Nearly all numerical methods involve a repetitive sequence of calculations and so it is inevitable that small individual rounding errors accumulate as they proceed. However, the actual growth of these errors can occur in different ways. If, after n steps of the method, the total rounding error is approximately $C\,n\,\epsilon$, where C is a positive constant and ϵ is the size of a typical rounding error, then the growth in rounding errors is usually acceptable. For example, if $C = 1$ and $\epsilon = 10^{-11}$, it takes about 50000 steps before the sixth decimal place is affected. On the other hand, if the total rounding error is approximately $Ca^n\epsilon$ or $Cn!\,\epsilon$, for some number $a > 1$, then the growth in rounding errors is usually unacceptable. For example, in the first case, if $C = 1$, $\epsilon = 10^{-11}$ and $a = 10$, it only takes about five steps before the sixth decimal place is affected. The second case is illustrated by the following example.

Example 1.6. *Many successful algorithms are available for calculating individual real roots of polynomial equations of the form*

$$p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_0 = 0$$

Some of these are described later. An attractive idea would be to use these methods to estimate one of the real roots, α say, and then to divide $P_n(x)$ by $x - \alpha$ to produce a polynomial of degree $n - 1$ which contains the remaining roots. This process can then be repeated until all of the roots have been located. This is usually referred to as the ***method of deflation****. If α were an exact root of $P_n(x) = 0$, then the remaining $n - 1$ roots would, of course, be the zeros of the deflated polynomial of degree $n - 1$. However, in practice α might only be an approximate root and in this case the zeros of the deflated polynomial can be very different from those of $P_n(x)$. For example, consider the cubic*

$$p_3(x) = x^3 - 13x^2 + 32x - 20 = (x - 1)(x - 2)(x - 10)$$

and suppose that an estimate of its largest zero is taken as 10.1. If we divide $p_3(x)$ by $x - 10.1$, the quotient is $x^2 - 2.9x + 2.71$ which has zeros $1.45 \pm 0.78i$. Clearly an error of 0.1 in the largest zero of $p_3(x)$ has induced a large error into the calculation of the remaining zeros.
It is interesting to note that if we divide $p_3(x)$ by $x - 1.1$, the corresponding quadratic has zeros 1.9 and 10.0 which are perfectly acceptable. The deflation process can be applied successfully provided that certain precautions are taken. In particular, the roots should be eliminated in increasing order of magnitude.

Of the two types of instability discussed, that of inherent instability is the most serious. Induced instability is a fault of the method and can be avoided either by modifying the existing method, as we did for some examples given in this section, or by using a completely different solution procedure. Inherent instability, however, is a fault of the problem so there is relatively little that we can do about it. The extent to which this property is potentially disastrous depends

not only on the degree of ill-conditioning involved but also on the context from which the problem is taken.

1.4 Taylor Series Expansions

Ever wondered

- How a pocket calculator can give you the value of sine (or cos, or cot) of any angle ?.
- How it can give you the square root (or cube root, or 4th root) of any positive number ?.
- How it can find the logarithm of any (positive) number you give it ?.

Does a calculator store every answer that every human may ever ask it ?. Actually, no. The pocket calculator just remembers special polynomials and substitutes whatever you give it into that polynomial. It keeps substituting into terms of that polynomial until it reaches the required number of decimal places. It then displays the answer on the screen.

A **polynomial function** of degree n is of the form:

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_0 \tag{1.4}$$

where $a_n \neq 0$ and n is a positive integer, called the $degree$ of the polynomial.

Example 1.7.

$$f(x) = x^4 - x^3 - 19x^2 + 5 \tag{1.5}$$

is a polynomial function of degree 4.

Given a infinitely differentiable function $f : \Re \to \Re$, defined in a region near the value $x = a$, then its **Taylor series expanded** around a is

$$f(x) = f(a) + f'(a)(x-a) + f''(a)\frac{(x-a)^2}{2!} + f'''(a)\frac{(x-a)^3}{3!} + \cdots + f^{(n)}(a)\frac{(x-a)^n}{n!} + \cdots \tag{1.6}$$

We can write this more conveniently using summation notation as:

$$f(x) \approx \sum_{n=0}^{\infty} \frac{f^{(n)}(a)\,(x-a)^n}{n!} \tag{1.7}$$

By Taylor series we can find a polynomial that gives us a good approximation to some function in the region near $x = a$, we need to find the first, second, third (and so on) derivatives of the function and substitute the value of a. Then we need to multiply those values by corresponding powers of $(x - a)$, giving us the **Taylor Series expansion** of the function $f(x)$ about $x = a$.

Conditions

In order to find such a series, some conditions have to be in place:

- The function $f(x)$ has to be infinitely differentiable (that is, we can find each of the first derivative, second derivative, third derivative, and so on forever).
- The function $f(x)$ has to be defined in a region near the value $(x = a)$.

Let's see what a Taylor Series is all about with an example.

Example 1.8. *Find the Taylor Expansion of* $f(x) = \ln x$ *near* $x = 10$.

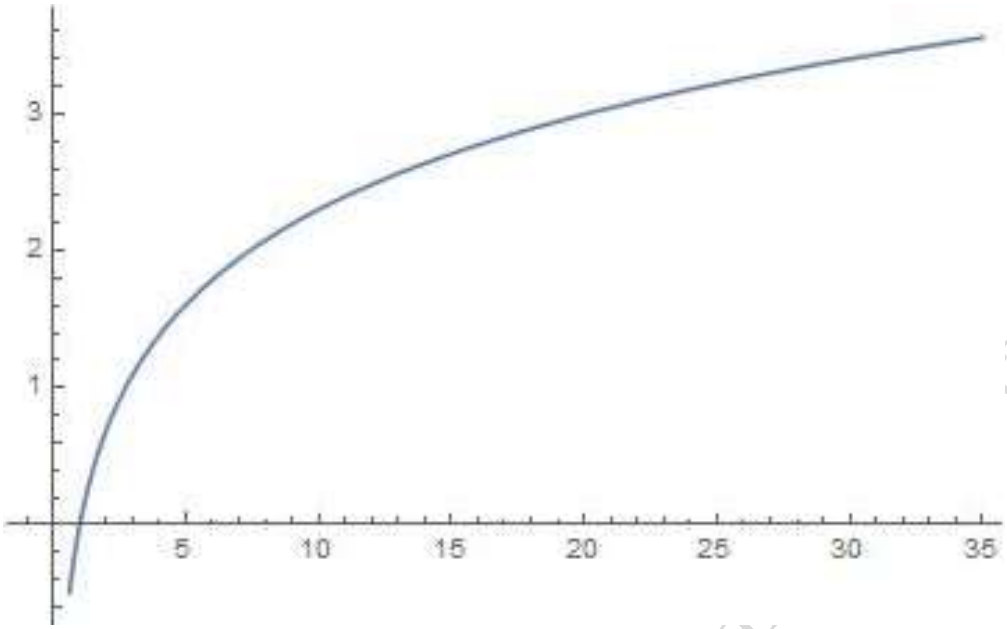

Figure 1.2: Graph of $f(x) = \ln(x)$

Our aim is to find a good polynomial approximation to the curve in the region near $x = 10$. *We need to use the Taylor Series with* $a = 10$. *The first term in the Taylor Series is* $f(a)$. *In this example,*

$$f(a) = f(10) = \ln(10) = 2.302585093.$$

Now for the derivatives; Recall the derivative of $\ln x$ *for* $x = 10$. *So*

$$f'(x) = \ln'(x) = \tfrac{1}{x} \qquad f'(10) = \ln'(x) = \tfrac{1}{10} = 0.1.$$

$$f''(x) = \ln''(x) = \tfrac{-1}{x^2} \qquad f''(10) = \ln''(x) = \tfrac{-1}{10^2} = -0.01.$$

$$f'''(x) = \ln'''(x) = \tfrac{2}{x^3} \qquad f'''(10) = \ln'''(10) = \tfrac{2}{10^3} = 0.002.$$

$$f^{iv}(x) = \ln^{iv}(x) = \tfrac{-6}{x^4} \qquad f^{iv}(10) = \ln^{iv}(10) = \tfrac{-6}{10^4} = -0.0006.$$

You can see that we could continue forever. This function is

infinitely differentiable. Now to substitute these values into the Taylor Series:

$$f(x) \approx f(a) + f'(a)(x-a) + f''(a)\frac{(x-a)^2}{2!} + f'''(a)\frac{(x-a)^3}{3!} + \cdots + f^{(n)}(a)\frac{(x-a)^n}{n!} + \cdots$$

We have

$$\ln(x) \approx \ln(10) + \ln'(10)(x-10) + \ln''(10)\frac{(x-10)^2}{2!} + \ln'''(10)\frac{(x-10)^3}{3!} + \cdots + \ln^{(n)}(10)\frac{(x-10)^n}{n!} + \cdots$$

$$\ln(x) \approx 2.302585093 + 0.1(x-10) + \frac{-0.01}{2!}(x-10)^2 + \frac{2\times 0.001}{3!}(x-10)^3 + \frac{-6\times 0.0001}{4!}(x-10)^4 + \cdots$$

Expanding this all out and collecting like terms, we obtain the polynomial which approximates $ln(x)$*:*

$$\ln(x) \approx 0.21925 + 0.4x - 0.03x^2 + 0.00133x^3 - 0.000025x^4 + \cdots$$

This is the approximating polynomial that we were looking for. We see from the graph that our polynomial (Dashed) is a good approximation for the graph of the natural logarithm function (Thick) in the region near $x = 10$*. Notice that the graph is not so good as we get further away from* $x = 10$*. The regions near* $x = 0$ *and* $x = 20$ *are showing some divergence (see figure 1.3).*
Let's zoom out some more and observe what happens with the approximation (see figure **??***).*

Clearly, it is no longer a good approximation for values of x *less than* 3 *or greater than* 20*. How do we get a better approximation ?. We would need to take more terms of the polynomial.*

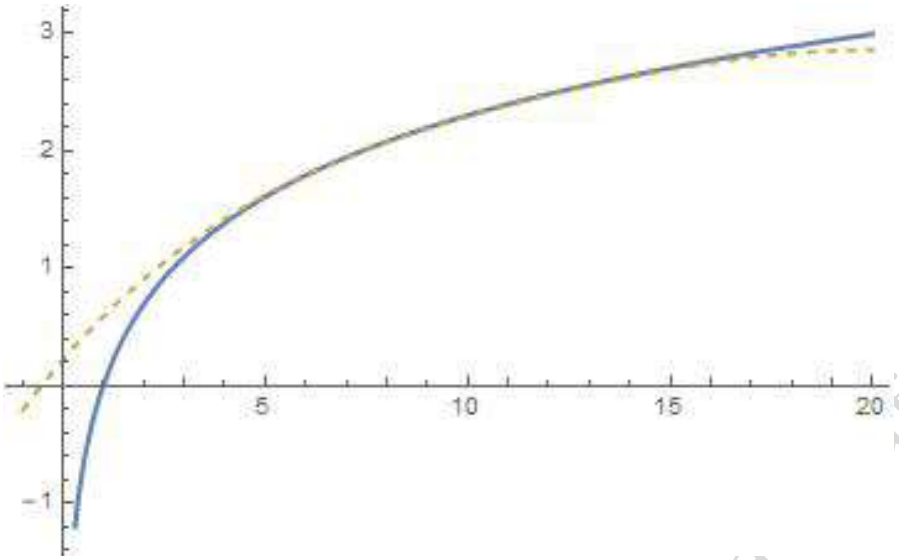

Figure 1.3: Graph of the approximating polynomial, and $f(x) = \ln(x)$

Home Work:
by the same procedure we can find the Taylor series of $\log x$ near $x = 1$

$$\log x = \sum_{n=1}^{\infty} \frac{(-1)^{n+1}\,(x-1)^n}{n} = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \cdots$$

1.5 Maclaurin Series

Maclaurin Series is a particular case of Taylor Series, in the region near $x = 0$. Such a polynomial is called the Maclaurin Series.
The infinite series expansion for $f(x)$ about $x = 0$ becomes:

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + \cdots + f^{(n)}(0)\frac{x^n}{n!} + \cdots$$

We can write this using summation notation as:

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(0)\,x^n}{n!} \tag{1.8}$$

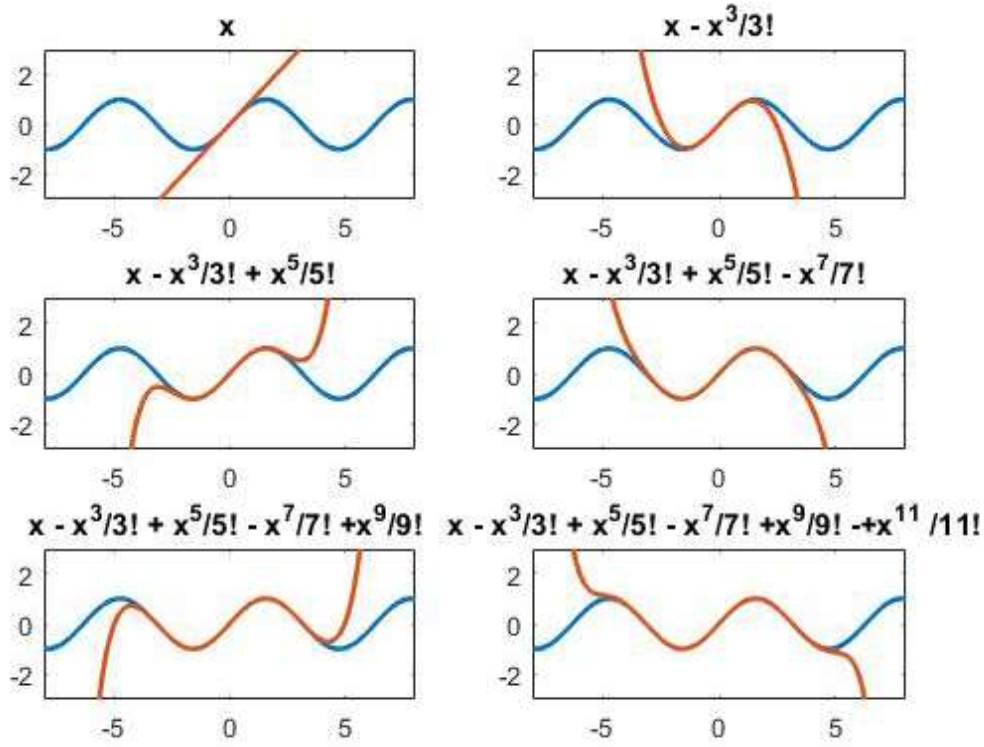

Figure 1.4: Graph of $f(x) = \sin(x)$ and different orders of Maclaurin series

Example 1.9. *Find the Maclaurin Series expansion for* $f(x) = \sin x$.

We need to find the first, second, third, etc derivatives and evaluate them at $x = 0$. *Starting with:*

$f(x) = \sin(x) \qquad f(0) = \sin(0) = 0$

Now for the derivatives:

$$f'(x) = \cos(x) \qquad f'(0) = \cos(0) = 1.$$

$$f''(x) = -\sin(x) \qquad f''(0) = -\sin(0) = 0.$$

$$f'''(x) = -\cos(x) \qquad f'''(0) = -\cos(0) = -1.$$

$$f^{iv}(x) = \sin(x) \qquad f^{iv}(0) = \sin(0) = 0.$$

We observe that this pattern will continue forever. Now to substitute the values of these derivatives into the Maclaurin Series:

$$f(x) = f(0) + f'(0)x + f''(0)\frac{x^2}{2!} + f'''(0)\frac{x^3}{3!} + \cdots + f^{(n)}(0)\frac{x^n}{n!} + \cdots$$

we have

$$\sin(x) = \sin(0) + \sin'(0)x + \sin''(0)\frac{x^2}{2!} + \sin'''(0)\frac{x^3}{3!} + \cdots + \sin^{(n)}(0)\frac{x^n}{n!} + \cdots$$

This gives us:

$$\begin{aligned}\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \cdots \\ &= \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} x^{2n+1}\end{aligned}$$

Matlab Code 1.10. *Taylor and Maclaurin series*

```
clc
clear
close
x1 = -3*pi:pi/100:3*pi;
y1 = sin(x1);
y2=@(x) x;
y3=@(x) x - x.^3 /factorial(3);
y4=@(x) x - x.^3 /factorial(3)+ x.^5 /factorial
  (5);
y5=@(x) x - x.^3 /factorial(3)+ x.^5 /factorial
  (5)- x.^7 /factorial(7) ;
y6=@(x) x - x.^3 /factorial(3)+ x.^5 /factorial
  (5)- x.^7 /factorial(7)+x.^9 /factorial(9) ;
```

```
y7=@(x) x  - x.^3 /factorial(3)+ x.^5 /factorial
    (5)- x.^7 /factorial(7)+x.^9 /factorial(9)-x
    .^11 /factorial(11) ;

subplot(3,2,1)
plot(x1,y1, x1, y2(x1),'LineWidth',2)
axis([-8 8 -3 3])
title('x ')

subplot(3,2,2)
plot(x1,y1, x1, y3(x1),'LineWidth',2)
axis([-8 8 -3 3])
title('x - x^3/3!')

subplot(3,2,3)
plot(x1,y1, x1, y4(x1),'LineWidth',2)
axis([-8 8 -3 3])

title('x - x^3/3! + x^5/5!')

subplot(3,2,4)
plot(x1,y1, x1, y5(x1),'LineWidth',2)
axis([-8 8 -3 3])
title('x - x^3/3! + x^5/5! - x^7/7! ')

subplot(3,2,5)
plot(x1,y1, x1, y6(x1),'LineWidth',2)
axis([-8 8 -3 3])
title('x - x^3/3! + x^5/5! - x^7/7! +x^9/9! ')

subplot(3,2,6)
plot(x1,y1, x1, y7(x1),'LineWidth',2)
axis([-8 8 -3 3])
```

```
42 title('x - x^3/3! + x^5/5! - x^7/7! +x^9/9! -+x
     ^{11} /11! ')
```

Home Work:
Use the same procedure as in previous example 1.9 to check the following Maclaurin series:

$$\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n = 1 + x + x^2 + x^3 + \cdots \quad \text{(when } -1 < x < 1\text{)}$$

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

$$\cos x = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} \quad = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots$$

Chapter 2

Solutions of Equations in One Variable

One of the fundamental problems of mathematics is that of solving equations of the form

$$f(x) = 0 \tag{2.1}$$

where f is a real valued function of a real variable x. Any number α satisfying equation (2.1) is called a **root** of the equation or a zero of f.

Most equations arising in practice are non-linear and are rarely of a form which allows the roots to be determined exactly. Consequently, numerical techniques must be used to find them.

Graphically, a solution, or a root, of Equation (2.1) refers to the point of intersection of $f(x)$ and the $x-$axis. Therefore, depending on the nature of the curve of $f(x)$ in relation to the $x-$axis, Equation (2.1) may have a unique solution, multiple solutions, or no solution. A root of an equation can sometimes be determined analytically resulting in an exact solution. For instance, the equation $e^{2x} - 3 = 0$ can be solved analytically to obtain a unique solution $x = \frac{1}{2}\ln 3$. In most situations, however, this is not possible and the root(s) must be found using a numerical procedure.

2.1 Bisection Technique

This technique based on the Intermediate Value Theorem. Suppose f is a continuous function defined on the interval $[a, b]$, with $f(a)$ and $f(b)$ of opposite sign. The Intermediate Value Theorem implies that a number p exists in (a, b) with $f(p) = 0$. The method calls for a repeated halving of subintervals of $[a, b]$ and, at each step, locating the half containing p. To begin, set $a_1 = a$ and $b_1 = b$, and let p_1 be the midpoint of $[a, b]$; that is,

$$p_1 = a_1 + \frac{b_1 - a_1}{2} = \frac{a_1 + b_1}{2}$$

1. If $f(p_1) = 0$, then $p = p_1$, and we are done.
2. If $f(p_1) \neq 0$, then $f(p_1)$ has the same sign as either $f(a_1)$ or $f(b_1)$.
   - If $f(p_1)$ and $f(a_1)$ have the same sign, $p \in (p_1, b_1)$. Set $a_2 = p_1$ and $b_2 = b_1$.
   - If $f(p_1)$ and $f(a_1)$ have opposite signs, $p \in (a_1, p_1)$. Set $a_2 = a_1$ and $b_2 = p_1$.

Then reapply the process to the interval $[a_2, b_2]$. See Figure 2.1.

We can select a tolerance $\epsilon > 0$ and generate $p_1, p_2, \cdots, p_N$ until one of the following conditions is met:

- $|p_N - p_{N-1}| < \epsilon$,
- $\frac{|p_N - p_{N-1}|}{|p_N|} < \epsilon, \quad p_N \neq 0, \quad$ or
- $f(p_N) < \epsilon$,

When using a computer to generate approximations, it is good practice to set an upper bound on the number of iterations. This eliminates the possibility of entering an infinite

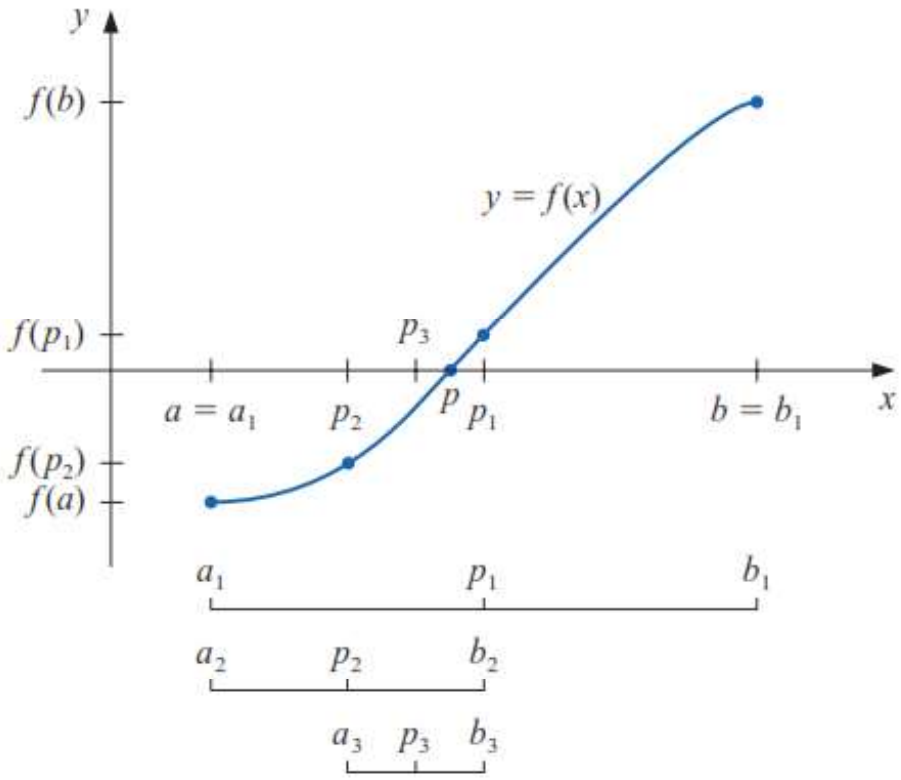

Figure 2.1: Produces of Bisection Technique

loop, a situation that can arise when the sequence diverges (and also when the program is incorrectly coded).

Example 2.1. *The function* $f(x) = x^3 + 4x^2 - 10$ *has a root in* $[1, 2]$*, because* $f(1) = -5$ *and and* $f(2) = 14$ *the Intermediate Value Theorem ensures that this continuous function has a root in* $[1, 2]$*.*
Using Bisection method with the Matlab code to determine an approximation to the root.

Matlab Code 2.2. *Bisection method*

```
% ********************************************************
% ************  bisection method  ************
% ****  to find a root of the function f(x)  ***
```

```
% ***********************************************
clc
clear
close all
f=@(x) x.^3+4*x.^2-10 ;
% f=@(x) (x+1)^2*exp(x^2-2)-1;
a=1;
b=2;
c=(a+b)/2;
e=0.00001;
k=1;
fprintf('      k          a           b          f(c)
      \n');
fprintf('    ___  ___________  ___________
   ___________ \n');

while abs(f(c)) > e
c=(a+b)/2;
if f(c)*f(a)<0
      b=c;
    else
      a=c;
 end
fprintf('%6.f  %10.8f  %10.8f  %10.8f \n', k,a,b
   ,f(c));
k=k+1;
end
fprintf('  The approximated root is c= %10.10f
 \n', c);
```

The result as the following table:

```
      k          a           b          f(c)
    ___  ___________  ___________  ___________
```

```
      1   1.00000000   1.50000000   2.37500000
      2   1.25000000   1.50000000  -1.79687500
      3   1.25000000   1.37500000   0.16210938
      4   1.31250000   1.37500000  -0.84838867
      5   1.34375000   1.37500000  -0.35098267
      6   1.35937500   1.37500000  -0.09640884
      7   1.35937500   1.36718750   0.03235579
      8   1.36328125   1.36718750  -0.03214997
      9   1.36328125   1.36523438   0.00007202
     10   1.36425781   1.36523438  -0.01604669
     11   1.36474609   1.36523438  -0.00798926
     12   1.36499023   1.36523438  -0.00395910
     13   1.36511230   1.36523438  -0.00194366
     14   1.36517334   1.36523438  -0.00093585
     15   1.36520386   1.36523438  -0.00043192
     16   1.36521912   1.36523438  -0.00017995
     17   1.36522675   1.36523438  -0.00005396
     18   1.36522675   1.36523056   0.00000903
  The approximated root is c= 1.3652305603
>>
```

Example 2.3. *The function* $f(x) = (x+1)^2 e^{(x^2-2)} - 1$ *has a root in* $[0, 1]$ *because* $f(0) < 0$ *and* $f(1) > 0$*. Use Bisection method to find the approximate root with* $\epsilon = 0.00001$*.*

2.2 MaTlab built-In Function fzero

The fzero function in MATLAB finds the roots of $f(x) = 0$ for a real function $f(x)$. FZERO Scalar nonlinear zero finding. $X = FZERO(FUN, X_0)$ tries to find a zero of the function FUN near X_0, if X_0 is a scalar.
For example 2.1 use the following Matlab code:

```
clc
clear
fun = @(x) x.^3+4*x.^2-10; % function
x0 = 1; % initial point
x = fzero(fun,x0)
```

the resulte is:
$x = 1.365230013414097$

Theorem 2.4. *Suppose that $f \in C[a,b]$ and $f(a)f(b) < 0$. The Bisection method generates a sequence $\{p_n\}_{n=1}^{\infty}$ approximating a zero p of f with*

$$|p_n - p| < \frac{b-a}{2^n}, \qquad n \geq 1$$

Proof. For each $n \geq 1$, we have

$$b_1 - a_1 = \frac{1}{2}(b-a), \quad and \quad p_1 \in (a_1, b_1)$$

$$b_2 - a_2 = \frac{1}{2}\left[\frac{1}{2}(b-a)\right] = \frac{1}{2^2}(b-a), \quad and \quad p_2 \in (a_2, b_2)$$

$$b_3 - a_3 = \frac{1}{2}(b_2 - a_2) = \frac{1}{2^3}(b-a), \quad and \quad p_3 \in (a_3, b_3)$$

and so for the n step we can get

$$b_n - a_n = \frac{1}{2^n}(b-a), \quad and \quad p_n \in (a_n, b_n)$$

Since $p_n \in (a_n, b_n)$ and $|(a_n, b_n)| = b_n - a_n$ for all $n \geq 1$, it follows that

$$|p_n - p| < b_n - a_n = \frac{b-a}{2^n}$$

the sequence $\{p_n\}_{n=1}^{\infty}$ converges to p with rate of convergence of order $\frac{1}{2^n}$; that is

$$p_n = p + O\left(\frac{1}{2^n}\right)$$

□

It is important to realize that Theorem 2.4 gives only a bound for approximation error and that this bound might be quite conservative. For example, this bound applied to the problem in Example 2.1 ensures only that

$$|p - p_9| < \frac{2-1}{2^9} = 0.001953125 \approx 2 \times 10^{-3}$$

but the actual error is much smaller:

$$\begin{aligned} |p - p_9| &\leq |1.365230013414097 - 1.365234375| \\ &\approx -0.000004361585903 \\ &\approx 4.4 \times 10^{-6} \end{aligned}$$

Example 2.5. *Determine the number of iterations necessary to solve* $f(x) = x^3 + 4x^2 - 10 = 0$ *with accuracy* 10^{-3} *using* $a_1 = 1$ *and* $b_1 = 2$.

Solution: *We we will use logarithms to find an integer* N *that satisfies*

$$\begin{aligned} |p - p_n| &< 2^{-N}(b_1 - a_1) \\ &= 2^{-N}(2-1) \\ &= 2^{-N} < 10^{-3} \end{aligned}$$

One can use logarithms to any base, but we will use base-10 *logarithms because the tolerance is given as a power of* 10. *Since* $2^{-N} < 10^{-3}$ *implies that* $\log_{10} 2^{-N} < \log_{10} 10^{-3} = -3$, *we have*

$$-N \log_{10} 2 < -3 \qquad \textit{and} \qquad N > \frac{3}{\log_{10} 2} \approx 9.96$$

Hence, 10 iterations will ensure an approximation accurate to within 10^{-3}.

2.3 EXERCISE

1. Use the Bisection method to find p_3 for $f(x) = \sqrt{x} - \cos x$ on $[0, 1]$.

2. Let $f(x) = 3(x+1)(x-\frac{1}{2})(x-1)$ Use the Bisection method on the intervals $[-2, 1.5]$ and $[-1.25, 2.5]$ to find p_3.

3. Use the Bisection method on the solutions accurate to within 10^{-2} for $f(x) = x^3 - 7x^2 + 14x - 6 = 0$ on each intervals: $[0, 1]$, $[1, 3.2]$ and $[3.2, 4]$.

4. Find an approximation to $\sqrt{3}$ correct to within 10^{-4} using the Bisection Algorithm. Hint: Consider $f(x) = x^2 - 3$.

2.4 Fixed-Point Iteration

A fixed point for a function is a number at which the value of the function does not change when the function is applied.

Definition 4. *The number p is a fixed point for a given function g if $g(p) = p$.*

Suppose that the equation $f(x) = 0$ can be rearranged as

$$x = g(x) \tag{2.2}$$

Any solution of this equation is called a fixed point of g. An obvious iteration to try for the calculation of fixed points is

$$x_{n+1} = g(x_n) \qquad n = 0, 1, 2, \cdots \tag{2.3}$$

The value of x_0 is chosen arbitrarily and the hope is that the sequence $x_0, x_1, x_2, \cdots$ converges to a number α which will automatically satisfy equation (2.2).
Moreover, since equation (2.2) is a rearrangement of (2.1), α is guaranteed to be a zero of f.
In general, there are many different ways of rearranging $f(x) = 0$ in the form (2.2). However, only some of these are likely to give rise to successful iterations, as the following example demonstrates.

Example 2.6. *Consider the quadratic equation*

$$x^2 - 2x - 8 = 0$$

with roots -2 and 4. Three possible rearrangements of this equation are

(a) $x_{n+1} = \sqrt{2x_n + 8}$

(b) $x_{n+1} = \frac{2x_n + 8}{x}$

(c) $x_{n+1} = \frac{x_n^2 - 8}{2}$

Numerical results for the corresponding iterations, starting with $x_0 = 5$, are given in Matlab code 2.7 with the Table.

Matlab Code 2.7. *Fixed Point Iteration*

```
1
2  clc
3  clear
4  close all
5
6  xa =5; % Initial value of root
7  xb =5;
8  xc =5;
9  fprintf('      k        Xa           Xb           Xc
       \n');
10 fprintf('    ___  __________  __________
   __________ \n');
11
12 for k=1:1:6
13  xa=sqrt(2*xa+8);
14  xb =(2*xb +8)/xb;
15  xc =(xc^2-8)/2;
16 fprintf('%6.f  %10.8f  %10.8f  %10.8f \n', k, xa
      , xb , xc);
17 end
```

The result as the following table:

k	Xa	Xb	Xc
1	4.24264069	3.60000000	8.50000000
2	4.06020706	4.22222222	32.12500000
3	4.01502355	3.89473684	512.0078125

```
6         4   4.00375413   4.05405405   131072.0000
7         5   4.00093842   3.97333333   8589934592.0
8         6   4.00023460   4.01342282   3.6893e+19
9 >>
```

Consider that the sequence converges for (a) and (b), but diverges for (c).
This example highlights the need for a mathematical analysis of the method. Sufficient conditions for the convergence of the fixed point iteration are given in the following (without proof) theorem.

Theorem 2.8. *If g' exists on an interval $I = [\alpha - A, \alpha + A]$ containing the starting value x_0 and fixed point α, then x_n converges to α provided*

$$|g'(x)| < 1 \quad on \quad I$$

We can now explain the results of Example 2.6

(a) If $g(x) = (2x+8)^{\frac{1}{2}}$ then $g'(x) = (2x+8)^{-1/2}$ Theorem 2.8 guarantees convergence to the positive root $\alpha = 4$, because $|g'(x)| < 1$ on the interval $I = [3, 5] = [\alpha - 1, \alpha + 1]$ containing the starting value $x_0 = 5$. which is in agreement with the results of column Xa in the Table.

(b) If $g(x) = \frac{(2x+8)}{x}$ then $g'(x) = \frac{-8}{x^2}$ Theorem 2.8 guarantees convergence to the positive root $\alpha = 4$, because $|g'(x)| < 1$ as (a), which is in agreement with the results of column Xb in the Table.

(c) If $g(x) = \frac{(x^2-8)}{2}$ then $g'(x) = x$ Theorem 2.8 cannot be used to guarantee convergence, which is in agreement with the results of column Xc in the Table.

Example 2.9. *Find the approximate solution for the equation*

$$f(x) = x^4 - x - 10 = 0$$

by fixed point iteration method starting with $x_0 = 1.5$ *with* $|x_n - x_{n-1}| < 0.009$

Solution

The function $f(x)$ has a root in the interval $(1, 2)$, **Why ?**, rearrange the equation as

$$x_{n+1} = g(x_n) = \sqrt{x_n + 10}$$

then

$$g'(x) = \frac{(x+10)^{\frac{-3}{4}}}{4}$$

Achieving the condition

$$|g'(x)| \leq 0.04139 \qquad on \quad (1, 2)$$

then we get the solution sequence $\{1.5, 1.8415, 1.85503, 1.8556, \cdots\}$. consider that $|1.85503 - 1.8556| = 0.00057 < 0.009$.

2.5 EXERCISE

1. Use an appropriate fixed point iteration to find the root of

   (a) $x - \cos x = 0$

   (b) $x^2 + \ln x = 0$

   starting in each case with $x_0 = 1$. Stop when $|x_{n+1} - x_n| < 0.5 \times 10^{-2}$.

2. Find the first nine terms of the sequence generated by $x_{n+1} = e^{-x_n}$ starting with $x_0 = 1$.

2.6 Newton-Raphson method

Newton-Raphson method is one of the most popular techniques for finding roots of non-linear equations.

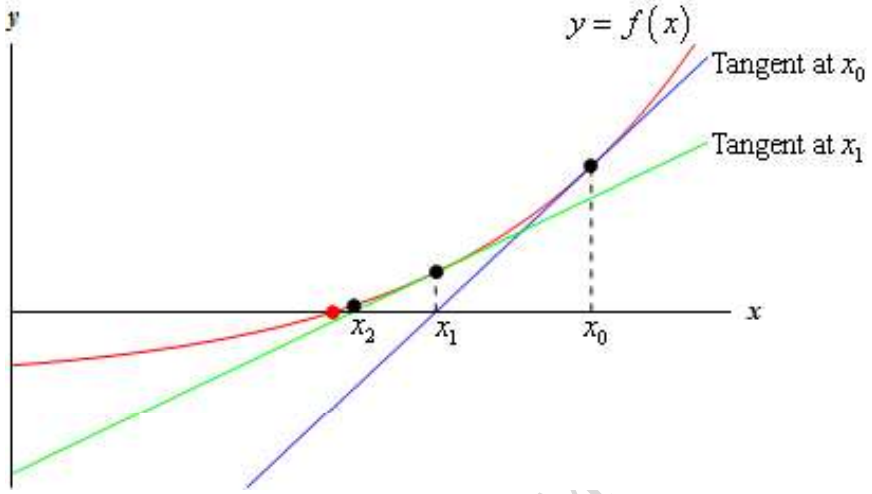

Figure 2.2: sketch of the Newton Raphson method

Derivative Newton-Raphson method:

Now Suppose that x_0 is a known approximation to a root of the function $y = f(x)$, as shown in Fig. 2.2.
The next approximation, x_2 is taken to be the point where tangent graph of $y = f(x)$ at $x = x_0$ intersects the $x-$axis.
From Taylor series we have

$$f(x_1) = f(x_0) + f'(x_0)(x_1 - x_0) + f''(x_0)\frac{(x_1 - x_0)^2}{2!} + f'''(x_0)\frac{(x_1 - x_0)^3}{3!} + \cdots + f^{(n)}(a)\frac{(x_1 - x_0)^n}{n!} + \cdots$$

consider x_1 as a root and take only the first two terms as an approximation:

$$\begin{aligned} 0 &= f(x_0) + f'(x_0)(x_1 - x_0) \\ (x_1 - x_0) &= -\frac{f(x_0)}{f'(x_0)} \\ x_1 &= x_0 - \frac{f(x_0)}{f'(x_0)} \end{aligned}$$

So, we can find the new approximation x_1. Now we can repeat the whole process to find an even better approximation.

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

we will arrive at the following formula.

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad n = 0, 1, 2, \cdots \tag{2.4}$$

Note that when $f'(x_n) = 0$ the calculation of x_{n+1} fails. This is because the tangent at x_n is horizontal.

Example 2.10. *Newton's method for calculating the zeros of*

$$f(x) = e^x - x - 2$$

is given by

$$\begin{aligned} x_{n+1} &= x_n - \frac{e^{x_n} - x_n - 2}{e^{x_n} - 1} \\ &= \frac{e^{x_n}(x_n - 1) + 2}{e^{x_n} - 1} \end{aligned}$$

The graph of f, sketched in Fig. 2.3, shows that it has two zeros. It is clear from this graph that x_n converges to the negative root if $x_0 < 0$ and to the positive root if $x_0 > 0$, and that it breaks down if $x_0 = 0$. The results obtained with $x_0 = -10$ and $x_0 = 10$ are listed in next table.

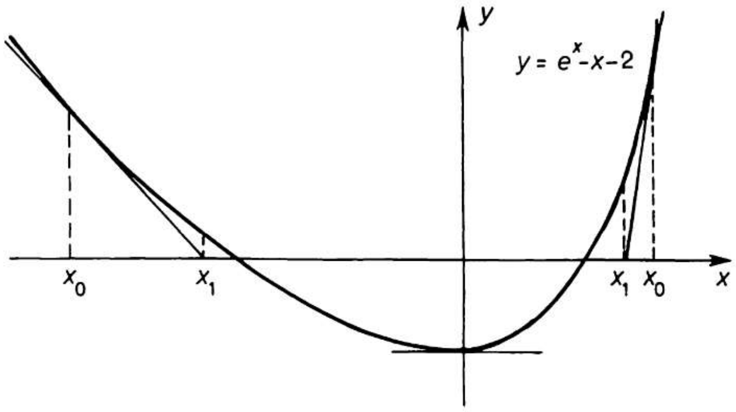

Figure 2.3: sketch of the Newton Raphson method for example 2.10

Matlab Code 2.11. *Newton Raphson method*

```
% ********  Newton Raphson method  ************
% ****  to find a root of the function f(x) ***
clc
clear
close all
f=@(x) exp(x)-x-2 ; % the function f(x)
fp=@(x) exp(x)-1 ; % the derivative f'(x) of f(x
    )
xa=-10; % Initial value of first root
xb=10;  % Initial value of second root
r = 'failure';
fprintf('    k          Xa                Xb   \n');
fprintf('    ___     ___________      ___________   \n'
    );
fprintf('%6.f     %10.8f     %10.8f \n', 0, xa ,
    xb );
for k=1:1:14
      if fp(xa)==0; r
      return
      elseif fp(xb)==0; r
```

```
        return
        end
    xa=xa-f(xa)/fp(xa);
    xb=xb-f(xb)/fp(xb);
    fprintf('%6.f      %10.8f      %10.8f \n', k, xa  ,
        xb  );
end
```

The result as the following table:

```
>>
```

k	Xa	Xb
1	−1.99959138	9.00049942
2	−1.84347236	8.00173312
3	−1.84140606	7.00474864
13	−1.84140566	1.14619325
14	−1.84140566	1.14619322

Sufficient conditions for the convergence of Newton's method are given in the following theorem.

Theorem 2.12. *If f'' is continuous on an interval $[\alpha - A, \alpha + A]$, then x_n converges to α provided $f'(\alpha) \neq 0$ and x_0 is sufficiently close to α.*

Proof. Comparison of equation

$$x_{n+1} = g(x_n) \qquad n = 0, 1, 2, \cdots$$

and the equation

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

shows that Newton's method is a fixed point iteration with

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

By the quotient rule,

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

let $x = \alpha$ then

$$g'(\alpha) = \frac{f(\alpha)f''(\alpha)}{(f'(\alpha))^2}$$

This implies that $g'(\alpha) = 0$, because $f(\alpha) = 0$ and $f'(\alpha) \neq 0$. Hence by the continuity of f'', there exists an interval $I = [\alpha - \delta, \alpha + \delta]$, for some $\delta > 0$, on which $|g'(x)| < 1$. Theorem 2.8 then guarantees convergence provided $x_0 \in I$, i.e. provided x_0 is sufficiently close to α. □

2.7 EXERCISE

1. Use Newton's method to find the roots of

   (a) $x - \cos x = 0$

   (b) $x^2 + \ln x = 0$

   (b) $x^3 + 4x^2 + 4x + 3 = 0$

   starting in each case with $x_0 = 1$. Stop when $|x_{n+1} - x_n| < 10^{-6}$.

2. Find the roots of $x^2 - 3x - 7$ using Newton's method with $\epsilon = 10^{-4}$ or maximum 20 iterations.

2.8 System of Non Linear Equations

Consider a system of m nonlinear equations with m unknowns

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_m) &= 0 \\ f_2(x_1, x_2, \cdots, x_m) &= 0 \\ \vdots \qquad\qquad & \vdots \qquad\qquad \vdots \\ f_m(x_1, x_2, \cdots, x_m) &= 0 \end{aligned}$$

where each $fi(i = 1, 2, ..., m)$ is a real valued function of m real variables. we shall only consider the generalization of Newton's method. In order to motivate the general case, consider a system of two non linear simultaneous equations in two unknowns given by

$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned} \tag{2.5}$$

Geometrically, the roots of this system are the points in the (x, y) plane where the curves defined by f and g intersect. For example, the curves represented by

$$\begin{aligned} f(x, y) = x^2 + y^2 - 4 &= 0 \\ g(x, y) = 2x - y^2 &= 0 \end{aligned}$$

are shown in Fig. 2.4. The roots of this system are then (α_1, β_1) and (α_2, β_2). Suppose that (α_n, β_n) is an approximation to a root (α, β). Writing $\alpha = (\alpha - x_n)$ and $\beta = y_n + (\beta - y_n)$ we can use Taylor's theorem for functions of two variables to deduce that

$$\begin{aligned} 0 &= f[\alpha, \beta] \\ &= f[x_n + (\alpha - x_n), y_n + (\beta - y_n)] \\ &= f(x_n, y_n) + (\alpha - x_n) f_x(x_n, y_n) + (\beta - y_n) f_y(x_n, y_n) + \cdots \end{aligned}$$

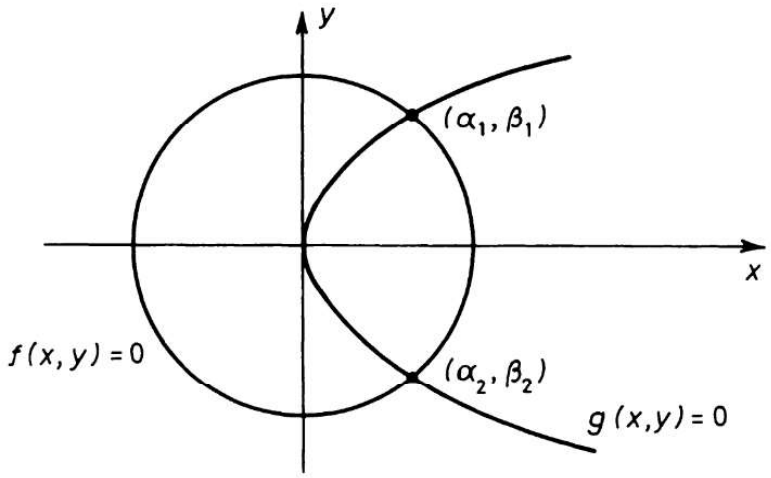

Figure 2.4: sketche of example 2.13

and

$$\begin{aligned} 0 &= g[\alpha, \beta] \\ &= g[x_n + (\alpha - x_n), y_n + (\beta - y_n)] \\ &= g(x_n, y_n) + (\alpha - x_n) g_x(x_n, y_n) + (\beta - y_n) g_y(x_n, y_n) + \cdots \end{aligned}$$

The notation f_x, f_y is used as an abbreviation for $\dfrac{\partial f}{\partial x}$, $\dfrac{\partial f}{\partial y}$, etc. If (x_n, y_n) is sufficiently close to (α, β) then higher order terms may be neglected to obtain

$$\begin{aligned} 0 &= f(x_n, y_n) + (\alpha - x_n) f_x(x_n, y_n) + (\beta - y_n) f_y(x_n, y_n) \\ 0 &= g(x_n, y_n) + (\alpha - x_n) g_x(x_n, y_n) + (\beta - y_n) g_y(x_n, y_n) \end{aligned} \qquad (2.6)$$

This represents a system of two linear algebraic equations for α and β. Of course, since higher order terms are omitted in the derivation of these equations, their solution (α, β) is no longer an exact root of equation (2.5). However, it will usually be a better approximation than (x_n, y_n), so replacing (α, β) by (x_{n+1}, y_{n+1}) in equation (2.6) gives the iterative

scheme

$$0 = f(x_n, y_n) + (x_{n+1} - x_n)f_x(x_n, y_n) + (y_{n+1} - y_n)f_y(x_n, y_n)$$
$$0 = g(x_n, y_n) + (x_{n+1} - x_n)g_x(x_n, y_n) + (y_{n+1} - y_n)g_y(x_n, y_n)$$

Or rewritten as:

$$\begin{aligned}(x_{n+1} - x_n)f_x(x_n, y_n) + (y_{n+1} - y_n)f_y(x_n, y_n) &= -f(x_n, y_n)\\ (x_{n+1} - x_n)g_x(x_n, y_n) + (y_{n+1} - y_n)g_y(x_n, y_n) &= -g(x_n, y_n)\end{aligned} \tag{2.7}$$

At a starting approximation (x_0, y_0), the functions f, f_x, f_y, g, g_x and g_y are evaluated. The linear equations are then solved for (x_1, y_1) and the whole process is repeated until convergence is obtained.
In matrix notation, equation (2.7) may be written as

$$\begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix} \begin{pmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \end{pmatrix} = - \begin{pmatrix} f \\ g \end{pmatrix}$$

where f, g and their partial derivatives are evaluated at (x_n, y_n). Hence

$$\begin{pmatrix} x_{n+1} - x_n \\ y_{n+1} - y_n \end{pmatrix} = - \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}^{-1} \begin{pmatrix} f \\ g \end{pmatrix}$$

Or rewritten as

$$\begin{pmatrix} x_{n+1} \\ y_{n+1} \end{pmatrix} = \begin{pmatrix} x_n \\ y_n \end{pmatrix} - \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}^{-1} \begin{pmatrix} f \\ g \end{pmatrix} \tag{2.8}$$

The matrix

$$J = \begin{pmatrix} f_x & f_y \\ g_x & g_y \end{pmatrix}$$

is called the Jacobian matrix. If the inverse of Jacobian matrix does not exist, then the method fails. Comparison of equations (2.4) and (2.8) shows that the above procedure is indeed an extension of Newton's method in one variable,

where division by f' generalizes to pre-multiplication by J^{-1}. For a larger system of equations it is convenient to use vector notation.

Note: for a 2×2 matrix the inverse is

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1} = \frac{1}{ad - dc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} \tag{2.9}$$

Example 2.13. *As an illustration of the above, consider the solution of*

$$f(x, y) = x^2 + y^2 - 4 = 0$$
$$g(x, y) = 2x - y^2 = 0$$

starting with $x_0 = y_0 = 1$. *In this case*

$$f = x^2 + y^2 - 4, \qquad f_x = 2x, \qquad f_y = 2y$$
$$g = 2x - y^2, \qquad g_x = 2, \qquad g_y = -2y$$

At the point $(1, 1)$, *equations* (2.7) *are given by*

$$2(x_1 - 1) + 2(y_1 - 1) = 2$$
$$2(x_1 - 1) - 2(y_1 - 1) = -1$$

which have solution $x_1 = 1.25$ *and* $y_1 = 1.75$. *This and further steps of the method are listed in the following Table.*

The following Matlab code is for example 2.13:

Matlab Code 2.14.

```
% ***************************************************
% *******  find a root of a System ***********
% **   of Two nonlinear  equations f and g    **
% ***************************************************
clc
clear
```

```
close all
% Define the functions f and g
% and their partial derivative
f=@(x,y) x^2+y^2-4 ; % the function f(x,y)
g=@(x,y) 2*x-y^2 ;   % the function g(x,y)
fx=@(x,y) 2*x;       % partial derivative of f
   to x
fy=@(x,y) 2*y;       % partial derivative of f
   to y
gx=@(x,y) 2 ;        % partial derivative of f
   to x
gy=@(x,y) -2*y;      % partial derivative of f
   to y
a=1; b=1;  % Initial root value
 fprintf(' n        Xn              Yn  \n')
for k=1:1:5
 X=[a;b];
 xn(k)=a;yn(k)=b;
 F=[f(a,b);g(a,b)];
 J=[fx(a,b),fy(a,b);gx(a,b),gy(a,b)]; % the
    Jacobian matrix
 X=X-inv(J)*F;
 a=X(1);
 b=X(2);
 fprintf('%2.0f     %2.6f        %2.6f \n', k ,a,b)
end
```

The result as the following table:

n	Xn	Yn
1	1.250000	1.750000
2	1.236111	1.581349
3	1.236068	1.572329
4	1.236068	1.572303

```
6   5       1.236068        1.572303
7  >>
```

2.9 EXERCISE

1. The system

$$3x^2 + y^2 + 9x - y - 12 = 0$$
$$x^2 + 36y^2 - 36 = 0$$

has exactly four roots. Find these roots starting with $(1, 1)$, $(1, -1)$, $(-4, 1)$ and $(-4, -1)$. Stop when successive iterates differ by less than 10^{-7}.

2. The system

$$4x^3 + y - 6 = 0$$
$$x^2 y - 1 = 0$$

has exactly three roots. Find these roots starting with $(1, 1)$, $(0.5, 5)$ and $(-1, 5)$. Stop when successive iterates differ by less than 10^{-7}.

3. Determine the series expansion about zero (at least first three nonzero terms) for the functions e^{-x^2}, $\frac{1}{2+x}$, $e^{\cos x}$, $\sin(\cos x)$, $(\cos x)^2(\sin x)$.

2.10 Fixed Point for System of Non Linear Equations

We now generalize fixed-point iteration to the problem of solving a system of m nonlinear equations in m unknowns

$$\begin{aligned} f_1(x_1, x_2, \cdots, x_m) &= 0 \\ f_2(x_1, x_2, \cdots, x_m) &= 0 \\ \vdots \qquad\qquad & \vdots \qquad\qquad \vdots \\ f_m(x_1, x_2, \cdots, x_m) &= 0 \end{aligned}$$

We can define fixed-point iteration for solving a system of nonlinear equations. First, we transform this system of equations into an equivalent system of the form

$$\begin{aligned} x_1 &= g_1(x_1, x_2, \cdots, x_m) \\ x_2 &= g_2(x_1, x_2, \cdots, x_m) \\ \vdots \qquad\qquad & \vdots \qquad\qquad \vdots \\ x_m &= g_m(x_1, x_2, \cdots, x_m) \end{aligned}$$

Then, we compute subsequent iterates by

$$\begin{aligned} x_1^{n+1} &= g_1(x_1^n, x_2^n, \cdots, x_m^n) \\ x_2^{n+1} &= g_2(x_1^n, x_2^n, \cdots, x_m^n) \\ \vdots \qquad\qquad & \vdots \qquad\qquad \vdots \\ x_3^{n+1} &= g_m(x_1^n, x_2^n, \cdots, x_m^n) \end{aligned}$$

For simplicity, consider a system of two non linear simultaneous equations in two unknowns given by

$$\begin{aligned} f(x, y) &= 0 \\ g(x, y) &= 0 \end{aligned} \tag{2.10}$$

to solve this system by fixed iteration method, transform this system of equations into an equivalent system of the form

$$\begin{aligned} x &= F(x,y) \\ y &= G(x,y) \end{aligned} \tag{2.11}$$

compute subsequent iterates by

$$\begin{aligned} x_{n+1} &= F(x_n, y_n) \\ y_{n+1} &= G(x_n, y_n) \end{aligned} \tag{2.12}$$

The convergent condition for this subsequent is

$$\begin{aligned} |F_x| + |F_y| &< 1 \\ |G_x| + |G_y| &< 1 \end{aligned}$$

Example 2.15. *consider the solution of*

$$\begin{aligned} f(x,y) &= x^3 + y^3 - 6x + 3 = 0 \\ g(x,y) &= x^3 - y^3 - 6y + 2 = 0 \end{aligned}$$

starting with $x_0 = y_0 = 0.5$*. In this case*

$$\begin{aligned} x = F(x,y) &= \frac{x^3 + y^3 + 3}{6} \\ y = G(x,y) &= \frac{x^3 - y^3 + 2}{6} \\ F_x = \frac{x^2}{2} \qquad & F_y = \frac{y^2}{2} \\ G_x = \frac{x^2}{2} \qquad & G_y = \frac{-y^2}{2} \end{aligned}$$

Now consider that at the point $(0.5, 0.5)$ *we have*

$$\begin{aligned} |F_x| + |F_y| &= |\frac{x_0^2}{2}| + |\frac{y_0^2}{2}| \\ &= \frac{(0.5)^2}{2} + \frac{(0.5)^2}{2} = 0.25 < 1 \\ |G_x| + |G_y| &= |\frac{x_0^2}{2}| + |\frac{-y_0^2}{2}| \\ &= \frac{(0.5)^2}{2} + \frac{(0.5)^2}{2} = 0.25 < 1 \end{aligned}$$

so, the convergence condition is satisfied at the point $(0.5, 0.5)$. *then*

$$\begin{aligned} x_1 &= \frac{x_0^3 + y_0^3 + 3}{6} = 0.5417 \\ y_1 &= \frac{x_0^3 - y_0^3 + 2}{6} = 0.3390 \end{aligned}$$

by the same procedure we have:

$$\begin{aligned} x_2 &= 0.5330 \qquad y_2 = 0.3520 \\ x_3 &= 0.5325 \qquad y_3 = 0.3512 \end{aligned}$$

and so on.

The following Matlab code is for example 2.15:

Matlab Code 2.16.

```
% ****************************************
% *******  find a root of a System **********
% **  of Two nonlinear  equations f and g   **
% ******        By Fixed Point Method        **
% ****************************************
clc
clear
close all
```

```
% Define the functions f and g
% and their partial derivative
f=@(x,y) (x^3+y^3+3)/6 ; % the function f(x,y)
g=@(x,y) (x^3-y^3+2)/6 ;   % the function g(x,y)
fx=@(x,y) x*x*0.5;         % partial derivative of
    f to x
fy=@(x,y) y*y*0.5;         % partial derivative of
    f to y
gx=@(x,y) x*x*0.5 ;          % partial derivative
   of f to x
gy=@(x,y) -y*y*0.5;        % partial derivative of
    f to y
a=0.5; b=0.5;  % Initial root value
 fprintf(' n        Xn              Yn  \n')
 fprintf('%2.0f     %2.8f       %2.8f \n', 0 ,a,b)
for k=1:1:8
    w1=abs(fx(a,b)+fy(a,b));
    w2=abs(gx(a,b)+gy(a,b));
     if w1 > 1 ; break ; end
     if w2 > 1 ; break ; end
    a=f(a,b);
    b=g(a,b) ;
   fprintf('%2.0f     %2.8f       %2.8f \n', k ,a,
     b)
end
```

The result as the following table:

n	Xn	Yn
0	0.50000000	0.50000000
1	0.54166667	0.33898775
2	0.53298008	0.35207474
3	0.53250741	0.35122633
4	0.53238788	0.35126185

```
 5     0.53237312      0.35125757
 6     0.53237077      0.35125750
 7     0.53237043      0.35125745
 8     0.53237038      0.35125745
>>
```

2.11 EXERCISE

1. solve problems 1 and 2 from exercise 2.9 by the fixed point method.

2. solve the system

$$x = \sin y$$
$$y = \cos x$$

using Newton method and the fixed point method with $(x_0, y_0) = (1, 1)$.

Chapter 3

Linear Algebraic Equations

Many important problems in science and engineering require the solution of systems of simultaneous linear equations of the form

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 \\ \vdots \quad \vdots \quad \vdots \quad \vdots & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n \end{aligned} \tag{3.1}$$

Where the coefficients a_{ij} and the right hand sides b_i are given numbers, and the quantities x_i are the unknowns which need to be determined. In matrix notation this system can be written as

$$A\,X = b \tag{3.2}$$

where $A = (a_{ij})$, $b = (b_i)$ and $x = (x_i)$. We shall assume that the $n \times n$ matrix A is non-singular (i.e. that the determinant of A is non-zero) so that equation (3.2) has a unique solution.

There are two classes of method for solving systems of this type. **Direct methods** find the solution in a finite number of steps, or **iterative methods** start with an arbitrary first approximation to x and then improve this estimate in an infinite but convergent sequence of steps.

3.1 Gauss elimination

Gauss elimination is used to solve a system of linear equations by transforming it to an upper triangular system (i.e. one in which all of the coefficients below the leading diagonal are zero) using elementary row operations. The solution of the upper triangular system is then found using back substitution.
We shall describe the method in detail for the general example of 3×3 system

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3 \end{aligned}$$

In matrix notation this system can be written as

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

STEP 1

The first step eliminates the variable x_1 from the second and third equations. This can be done by subtracting multiples $m_{21} = \frac{a_{21}}{a_{11}}$ and $m_{31} = \frac{a_{31}}{a_{11}}$ of row 1 from rows 2 and 3, respectively, producing the equivalent system

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & a_{32}^{(2)} & a_{33}^{(2)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(2)} \end{pmatrix}$$

where $a_{ij}^{(2)} = a_{ij} - m_{ij}a_{1j}$ and $b^{(2)} = b_i - m_{i1}b_1$ $(i, j = 2, 3)$.

STEP 2

The second step eliminates the variable x_2 from the third equation. This can be done by subtracting a multiple $m_{32} = \frac{a_{32}^{(2)}}{a_{22}^{(2)}}$ from row 2 and 3, producing the equivalent upper triangular system

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} \\ 0 & 0 & a_{33}^{(3)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2^{(2)} \\ b_3^{(3)} \end{pmatrix}$$

where $a_{33}^{(3)} = a_{33}^{(2)} - m_{32}a_{23}^{(2)}$ and $b_3^{(2)} = b_3^{(2)} - m_{32}b_2^{(2)}$.
Since these row operations are reversible, the original system and the upper triangular system have the same solution. The upper triangular system is solved using back substitution. The last equation implies that

$$x_3 = \frac{b_3^{(3)}}{a_{33}^{(3)}}$$

This number can then be substituted into the second equation and the value of x_2 obtained from

$$x_2 = \frac{b_2^{(2)} - a_{23}^{(2)}x_3}{a_{22}^{(2)}}$$

Finally, the known values of x_2 and x_3 can be substituted into the first equation and the value of x_1 obtained from

$$x_1 = \frac{b_1 - a_{12}x_2 - a_{13}x_3}{a_{11}}$$

It is clear from previous equations that the algorithm fails if any of the quantities $a_{jj}^{(j)}$ are zero, since these numbers are used as the denominators both in the multipliers m_{ij}

and in the back substitution equations. These numbers are usually referred to as pivots. Elimination also produces poor results if any of the multipliers are greater than one in modulus. It is possible to prevent these difficulties by using row interchanges. At step j, the elements in column j which are on or below the diagonal are scanned. The row containing the element of largest modulus is called the pivotal row. Row j is then interchanged (if necessary) with the pivotal row.
It can, of course, happen that all of the numbers $a_{jj}^{(j)}$, $a_{j+1.j}^{(j)}$, $\cdots$, $a_{nj}^{(j)}$ are exactly zero, in which case the coefficient matrix does not have full rank and the system fails to possess a unique solution.

Example 3.1. *To illustrate the effect of partial pivoting, consider the solution of*

$$\begin{pmatrix} 0.61 & 1.23 & 1.72 \\ 1.02 & 2.15 & -5.51 \\ -4.34 & 11.2 & -4.25 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.792 \\ 12 \\ 16.3 \end{pmatrix}$$

using three significant figure arithmetic with rounding. This models the more realistic case of solving a large system of equations on a computer capable of working to, say, ten significant figure accuracy. Without partial pivoting we proceed as follows:
***Step 1**: The multipliers are* $m_{21} = \frac{1.02}{0.61} = 1.67$ *and* $m_{31} = \frac{-4.34}{0.61} = -7.11$, *which give*

$$\begin{pmatrix} 0.61 & 1.23 & 1.72 \\ 0 & 0.10 & -8.38 \\ 0 & 20.0 & 7.95 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.792 \\ 10.7 \\ 21.9 \end{pmatrix}$$

Step 2 *The multiplier is* $m_{32} = \frac{20}{0.1} = 200$*, which gives*

$$\begin{pmatrix} 0.61 & 1.23 & 1.72 \\ 0 & 0.10 & -8.38 \\ 0 & 0 & 1690 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0.792 \\ 10.7 \\ -2120 \end{pmatrix}$$

Solving by back substitution, we obtain
$x_3 = -1.25 \qquad x_2 = 2 \qquad x_1 = 0.790$
With partial pivoting we proceed as follows:
Step 1*: Since* $|-4.34| > |0.610|$ *and* $|1.02|$*, rows* 1 *and* 3 *are interchanged to get*

$$\begin{pmatrix} -4.34 & 11.2 & -4.25 \\ 1.02 & 2.15 & -5.51 \\ 0.61 & 1.23 & 1.72 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16.3 \\ 12 \\ 0.792 \end{pmatrix}$$

The multiplier is $m_{21} = \frac{1.02}{-4.34} = -0.235$ *and* $m_{31} = \frac{0.610}{-4.34} = -0.141$ *which gives*

$$\begin{pmatrix} -4.34 & 11.2 & -4.25 \\ 0 & 4.78 & -6.51 \\ 0 & 2.81 & 1.12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16.3 \\ 15.8 \\ 3.09 \end{pmatrix}$$

Step 2 *Since* $|4.78| > |2.81|$*, no further interchanged are needed and* $m_{32} = \frac{2.81}{4.78} = 5.88$*, which gives*

$$\begin{pmatrix} -4.34 & 11.2 & -4.25 \\ 0 & 4.78 & -6.51 \\ 0 & 0 & 4.95 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 16.3 \\ 15.8 \\ -6.20 \end{pmatrix}$$

Solving by back substitution, we obtain
$x_3 = -1.25 \qquad x_2 = 1.60 \qquad x_1 = 0.1.59$
By substituting these values into the original system of equations it is easy to verify that the result obtained with partial pivoting is a reasonably accurate solution. (In fact, the exact solution, rounded to three significant figures, is given by

$x_3 = -1.26$, $x_2 = 1.60$ *and* $x_1 = 1.61$ *) However, the values obtained without partial pivoting are totally unacceptable; the value of* x_1 *is not even correct to one significant figure.*

3.2 EXERCISE

Solve the following systems of linear equations using Gauss elimination (i) without pivoting (ii) with partial pivoting.

1.

$$\begin{aligned} 0.005x_1 + x_2 + x_3 &= 2 \\ x_1 + 2x_2 + x_3 &= 4 \\ -3x_1 - x_2 + 6x_3 &= 2 \end{aligned}$$

2.

$$\begin{aligned} x_1 - x_2 + 2x_3 &= 5 \\ 2x_1 - 2x_2 + x_3 &= 1 \\ 30x_1 - 2x_2 + 7x_3 &= 20 \end{aligned}$$

3.

$$\begin{aligned} 1.19x_1 + 2.37x_2 - 7.31x_3 + 1.75x_4 &= 2.78 \\ 2.15x_1 - 9.76x_2 + 1.54x_3 - 2.08x_4 &= 6.27 \\ 10.7x_1 - 1.11x_2 + 3.78x_3 + 4.49x_4 &= 9.03 \\ 2.17x_1 + 3.58x_2 + 1.70x_3 + 9.33x_4 &= 5.00 \end{aligned}$$

3.3 Gauss Jordan Method

The following row operations produce an equivalent system, i.e., a system with the same solution as the original one.

1. Interchange any two rows.
2. Multiply each element of a row by a nonzero constant.
3. Replace a row by the sum of itself and a constant multiple of another row of the matrix.

Convention: For these row operations, we will use the following notations:

- $R_i \longleftrightarrow Rj$ means: Interchange row i and row j.
- αR_i means: Replace row i with α times row i.
- $R_i + \alpha R_j$ means: Replace row i with the sum of row i and α times row j.

The Gauss-Jordan elimination method to solve a system of linear equations is described in the following steps.

1. Write the extended matrix of the system.
2. Use row operations to transform the extended matrix to have following properties:
   (a) The rows (if any) consisting entirely of zeros are grouped together at the bottom of the matrix.
   (b) In each row that does not consist entirely of zeros, the leftmost nonzero element is a 1 (called a leading 1 or a pivot).
   (c) Each column that contains a leading 1 has zeros in all other entries.
   (d) The leading 1 in any row is to the left of any leading 1's in the rows below it.
3. Stop process in step 2 if you obtain a row whose elements are all zeros except the last one on the right. In that case, the system is inconsistent and has no solutions. Otherwise, finish step 2 and read the solutions of the system from the final matrix.

Example 3.2. *Solve the following system of equations using the Gauss Jordan elimination method.*

$$\begin{aligned} x + y + z &= 5 \\ 2x + 3y + 5z &= 8 \\ 4x + 5z &= 2 \end{aligned}$$

Solution: *The extended matrix of the system is the following.*

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 5 \\ 2 & 3 & 5 & 8 \\ 4 & 0 & 5 & 2 \end{array}\right]$$

use the row operations as following:

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 5 \\ 2 & 3 & 5 & 8 \\ 4 & 0 & 5 & 2 \end{array}\right] \xrightarrow[R_3 = R_3 - 4R_1]{R_2 = R_2 - 2R_1} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 5 \\ 0 & 1 & 3 & -2 \\ 0 & -4 & 1 & -18 \end{array}\right]$$

$$\xrightarrow[R_3 = \frac{1}{13}R_3]{R_3 = R_3 + 4R_2} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 5 \\ 0 & 1 & 3 & -2 \\ 0 & 0 & 1 & -2 \end{array}\right]$$

$$\xrightarrow[R_1 = R_1 - R_2]{\begin{array}{c} R_2 = R_2 - 3R_3 \\ R_1 = R_1 - 3R_3 \end{array}} \left[\begin{array}{ccc|c} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & -2 \end{array}\right]$$

From this final matrix, we can read the solution of the system. It is

$$x = 3, \qquad y = 4, \qquad z = -2$$

Example 3.3. *Solve the following system of equations using the Gauss Jordan elimination method.*

$$\begin{aligned} x + 2y - 3z &= 2 \\ 6x + 3y - 9z &= 6 \\ 7x + 14y - 21z &= 13 \end{aligned}$$

Solution: *The extended matrix of the system is the following.*

$$\left[\begin{array}{ccc|c} 1 & 2 & -3 & 2 \\ 6 & 3 & -9 & 6 \\ 7 & 14 & -21 & 13 \end{array}\right]$$

use the row operations as following:

$$\left[\begin{array}{ccc|c} 1 & 2 & -3 & 2 \\ 6 & 3 & -9 & 6 \\ 7 & 14 & -21 & 13 \end{array}\right] \xrightarrow[R_3 = R_3 - 7R_1]{R_2 = R_2 - 6R_1} \left[\begin{array}{ccc|c} 1 & 1 & -3 & 2 \\ 0 & -9 & 9 & -6 \\ 0 & 0 & 0 & -1 \end{array}\right]$$

We obtain a row whose elements are all zeros except the last one on the right. Therefore, we conclude that the system of equations is inconsistent, i.e., it has no solutions.

Example 3.4. *Solve the following system of equations using the Gauss Jordan elimination method.*

$$\begin{aligned} 4y + z &= 2 \\ 2x + 6y - 2z &= 3 \\ 4x + 8y - 5z &= 4 \end{aligned}$$

Solution: *The extended matrix of the system is the following.*

$$\left[\begin{array}{ccc|c} 0 & 4 & 1 & 2 \\ 2 & 6 & -2 & 3 \\ 4 & 8 & -5 & 4 \end{array}\right]$$

use the row operations as following:

$$\left[\begin{array}{ccc|c} 0 & 4 & 1 & 2 \\ 2 & 6 & -2 & 3 \\ 4 & 8 & -5 & 4 \end{array}\right] \xrightarrow[R_3 = R_3 - 2R_1]{R_1 \longleftrightarrow R_2} \left[\begin{array}{ccc|c} 2 & 6 & -2 & 3 \\ 0 & 4 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

$$\xrightarrow[R_1 = \frac{1}{2}R_1]{\substack{R_2 = \frac{1}{4}R_2 \\ R_1 = R_1 - 6R_2}} \left[\begin{array}{ccc|c} 1 & 0 & \frac{-7}{4} & 0 \\ 0 & 1 & \frac{1}{4} & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{array}\right]$$

We can stop This because the form of the last matrix. It corresponds to the following system.

$$x - \frac{7}{4}z = 0$$
$$y + \frac{1}{4}z = \frac{1}{2}$$

We can express the solutions of this system as

$$x = \frac{7}{4}z \qquad y = -\frac{1}{4}z + \frac{1}{2}$$

*Since there is no specific value for z, it can be chosen arbitrarily. This means that there are **infinitely many** solutions for this system. We can represent all the solutions by using a parameter t as follows.*

$$x = \frac{7}{4}t \qquad y = -\frac{1}{4}t + \frac{1}{2} \qquad z = t$$

Any value of the parameter t gives us a solution of the system. For example:
$t = 4$ gives the solution $(x, y, z) = (7, \frac{-1}{2}, 4)$
$t = -2$ gives the solution $(x, y, z) = (\frac{-7}{2}, 1, -2)$

For **Gauss elimination method** we can use the following Matlab code:

Matlab Code 3.5. *Gauss method*

```
% ***************************************
% ****  Solve a system of linear equation ****
% **          by Gauss elimination method        **
% ***************************************
clc
clear
close all
```

```
a = [3  4 -2  2  2
     4  9 -3  5  8
    -2 -3  7  6  10
     1  4  6  7  2];
[m,n]=size(a);
% m = Number of Rows
% n = Number of Colomns
for j=1:m-1
    for z=2:m
        % Pivoting
        if a(j,j)==0
            t=a(j,:);
            a(j,:)=a(z,:);
            a(z,:)=t;
        end
    end
    for i=j+1:m
        a(i,:)=a(i,:)-a(j,:)*(a(i,j)/a(j,j));
    end
end
x=zeros(1,m);
%  Back Substitution
for s=m:-1:1
    c=0;
    for k=2:m
        c=c+a(s,k)*x(k);
    end
    x(s)=(a(s,n)-c)/a(s,s);
end
% Display the results
disp('Gauss elimination method:');
a
x'
```

The result as the following:

```
Gauss elimination method:

a =

    3.0000    4.0000   -2.0000    2.0000
       2.0000
         0    3.6667   -0.3333    2.3333
            5.3333
         0         0    5.6364    7.5455
            11.8182
         0         0         0   -4.6129
            -17.0323


ans =

   -2.1538
   -1.1538
   -2.8462
    3.6923

>>
```

For **Gauss Jordan elimination method** we can use the following Matlab code:

Matlab Code 3.6. *Gauss Jordan method*

```
% ***************************************
% ****  Solve a system of linear equation ****
% **   by Gauss Jordan elimination method     **
% ***************************************
clc
clear
```

```
close all
a = [3  4 -2  2  2
     4  9 -3  5  8
    -2 -3  7  6 10
     1  4  6  7  2];
[m,n]=size(a);
% m = Number of Rows
% n = Number of Colomns

for j=1:m-1
    % Pivoting
    for z=2:m
        if a(j,j)==0
            t=a(1,:);
            a(1,:)=a(z,:);
            a(z,:)=t;
        end
    end
    for i=j+1:m
        a(i,:)=a(i,:)-a(j,:)*(a(i,j)/a(j,j));
    end
end

for j=m:-1:2
    for i=j-1:-1:1
        a(i,:)=a(i,:)-a(j,:)*(a(i,j)/a(j,j));
    end
end

for s=1:m
    a(s,:)=a(s,:)/a(s,s);
    x(s)=a(s,n);
end
```

```
% Display the results
disp('Gauss–Jordan method:');
a
x'
```

The result as the following:

```
Gauss–Jordan method:

a =

    1.0000         0         0         0
   −2.1538
         0    1.0000         0         0
              −1.1538
         0         0    1.0000         0
              −2.8462
         0         0         0    1.0000
               3.6923


ans =

   −2.1538
   −1.1538
   −2.8462
    3.6923

>>
```


3.4 EXERCISE

1. solve exercise 3.2 by Gauss Jordan Method

2. Solve the following system of equations using the Gauss Jordan elimination method.

$$\begin{aligned} x + y + 2z &= 1 \\ 2x + -y + w &= -2 \\ x - y - z - 2w &= 4 \\ 2x - y + 2z - w &= 0 \end{aligned}$$

3.5 Matrix Inverse using Gauss-Jordan method

Given a matrix A of order $(n \times n)$, its inverse A^{-1} is the matrix with the property that $AA^{-1} = I = A^{-1}A$, Note the following identities

1. $(A^{-1})^{-1} = A$
2. $(A^T)^{-1} = (A^{-1})^T$
3. $(AB)^{-1} = B^{-1}A^{-1}$

Moreover, A is invertible, then the solution to the system of linear equations $AX = b$ can be written as $X = A^{-1}b$. We can

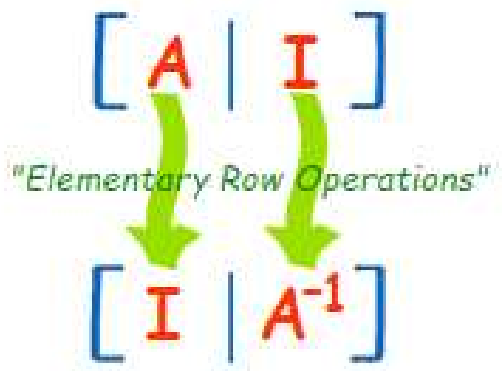

Figure 3.1: digram of find the inverse of a matrix using elementary row operations

use Gauss Jordan method To obtain the inverse of a $n \times n$ matrix A as following:

1. Create the partitioned matrix $(A|I)$, where I is the identity matrix.
2. use Gauss Jordan Elimination steps on partitioned matrix.

3. If done correctly (A have an inverse), the resulting partitioned matrix will take the form $\left(I|A^{-1}\right)$.

4. Double check your work by making sure that $AA^{-1} = I$.

Below is a demonstration of this process:

Example 3.7. *Find inverse of the matrix* $A = \begin{bmatrix} 3 & 2 & 0 \\ 1 & -1 & 0 \\ 0 & 5 & 1 \end{bmatrix}$ *using Gauss-Jordan method.*

Solution: *The partitioned matrix of the system is the following.*

$$\left[\begin{array}{ccc|ccc} 3 & 2 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 5 & 1 & 0 & 0 & 1 \end{array}\right]$$

use the row operations as following:

$$\left[\begin{array}{ccc|ccc} 3 & 2 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 5 & 1 & 0 & 0 & 1 \end{array}\right] \xrightarrow{R_1 \longleftrightarrow R_2} \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 1 & 0 \\ 3 & 2 & 0 & 1 & 0 & 0 \\ 0 & 5 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{R_2 = R_2 - 3R_1} \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 5 & 0 & 1 & -3 & 0 \\ 0 & 5 & 1 & 0 & 0 & 1 \end{array}\right]$$

$$\xrightarrow{R_3 = R_3 - R_2} \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 5 & 0 & 1 & -3 & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 \end{array}\right]$$

$$\xrightarrow{R_2 = \frac{1}{5}R_2} \left[\begin{array}{ccc|ccc} 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & \frac{1}{5} & \frac{-3}{5} & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 \end{array}\right]$$

$$\xrightarrow{R_1 = R_1 + R_2} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{1}{5} & \frac{2}{5} & 0 \\ 0 & 1 & 0 & \frac{1}{5} & \frac{-3}{5} & 0 \\ 0 & 0 & 1 & -1 & 3 & 1 \end{array}\right]$$

Now we have

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

check the solution ($AA^{-1} = I$).

3.6 Cramer's Rule

Cramer's rule begins with the clever observation

$$\begin{vmatrix} x_1 & 0 & 0 \\ x_2 & 1 & 0 \\ x_3 & 0 & 1 \end{vmatrix} = x_1$$

That is to say, if you replace the first column of the identity matrix with the vector $\mathbf{x} = \begin{pmatrix} x_1, x_2, x_3 \end{pmatrix}^T$, the determinant is x_1. Now, we've illustrated this for the 3×3 case and for column one. In general, if you replace the ith column of an $n \times n$ identity matrix with a vector $\mathbf{x}$, the determinant of the matrix you get will be x_i, the ith component of $\mathbf{x}$.
Note that if $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \; b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}. \text{ and } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}.$$

then

$$\begin{pmatrix} \\ A \\ \\ \end{pmatrix} \begin{pmatrix} x_1 & 0 & 0 \\ x_2 & 1 & 0 \\ x_3 & 0 & 1 \end{pmatrix} = \begin{pmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{pmatrix}.$$

Take determinants of both sides then we get

$$det(A)x_1 = det(B_1)$$

where B_1 is the matrix we get when we replace column 1 of A by the vector **b**. So,

$$x_1 = \frac{det(B_1)}{det(A)}.$$

In general

$$x_i = \frac{det(B_i)}{det(A)},$$

where B_i is the matrix we get by replacing column i of A with **b**.

Example 3.8. *Use Cramer's rule to solve for the the linear system:*

$$\begin{aligned} 2x_1 + x_2 - 5x_3 + x_4 &= 8 \\ x_1 - 3x_2 - 6x_4 &= 9 \\ 2x_2 - x_3 + 2x_4 &= -5 \\ x_1 + 4x_2 - 7x_3 + x_4 &= 0 \end{aligned}$$

***Solution**: write the system in matrix notation $AX = b$, then we have*

$$A = \begin{pmatrix} 2 & 1 & -5 & 1 \\ 1 & -3 & 0 & -6 \\ 0 & 2 & -1 & 2 \\ 1 & 4 & -7 & 6 \end{pmatrix} \text{ and } \quad b = \begin{pmatrix} 8 \\ 9 \\ -5 \\ 0 \end{pmatrix}.$$

Now we need to calculate $det(A), det(B_1), det(B_2), det(B_3), det(B_4)$*:*

$$A = \begin{pmatrix} 2 & 1 & -5 & 1 \\ 1 & -3 & 0 & -6 \\ 0 & 2 & -1 & 2 \\ 1 & 4 & -7 & 6 \end{pmatrix} \xrightarrow{then} det(A) = 27 \neq 0$$

$$B_1 = \begin{pmatrix} 8 & 1 & -5 & 1 \\ 9 & -3 & 0 & -6 \\ -5 & 2 & -1 & 2 \\ 0 & 4 & -7 & 6 \end{pmatrix} \xrightarrow{then} det(B_1) = 81$$

$$B_2 = \begin{pmatrix} 2 & 8 & -5 & 1 \\ 1 & 9 & 0 & -6 \\ 0 & -5 & -1 & 2 \\ 1 & 0 & -7 & 6 \end{pmatrix} \xrightarrow{then} det(B_2) = -108$$

$$B_3 = \begin{pmatrix} 2 & 1 & 8 & 1 \\ 1 & -3 & 9 & -6 \\ 0 & 2 & -5 & 2 \\ 1 & 4 & 0 & 6 \end{pmatrix} \xrightarrow{then} det(B_3) = -27$$

$$B_4 = \begin{pmatrix} 2 & 1 & -5 & 8 \\ 1 & -3 & 0 & 9 \\ 0 & 2 & -1 & -5 \\ 1 & 4 & -7 & 0 \end{pmatrix} \xrightarrow{then} det(B_4) = 27$$

This lead to:

$$x_1 = \frac{det(B_1)}{det(A)} = \frac{81}{27} = 3$$

$$x_2 = \frac{det(B_2)}{det(A)} = \frac{-108}{27} = -4$$

$$x_3 = \frac{det(B_3)}{det(A)} = \frac{-27}{27} = -1$$

$$x_4 = \frac{det(B_4)}{det(A)} = \frac{27}{27} = 1$$

3.7 EXERCISE

1. Solve problems in exercise 3.2 and exercise 3.4 using Cramer's rule.

2. Use Cramer's rule to solve for the vector $X = [x_1, x_2, x_3]^t$:

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

3.8 Iterative Methods: Jacobi and Gauss-Seidel

Jacobi's method is the easiest iterative method for solving a system of linear equations. Given a general set of n equations and n unknowns ($A_{n\times n}\mathbf{x}_{n\times 1} = \mathbf{b}_{n\times 1}$), where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}, b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \text{ and } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

If the diagonal elements are non-zero, each equation is rewritten for the corresponding unknown, that is, the first equation is rewritten with x_1 on the left hand side, the second equation is rewritten with x_2 on the left hand side and so on as follows

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}\left(b_1 - \sum_{j=2}^{n} a_{1j}x_j\right) \\ x_2 &= \frac{1}{a_{22}}\left(b_2 - \sum_{j=1;j\neq 2}^{n} a_{2j}x_j\right) \\ &\ldots \quad \ldots \quad \ldots \quad \ldots \\ x_n &= \frac{1}{a_{nn}}\left(b_n - \sum_{j=1}^{n-1} a_{nj}x_j\right) \end{aligned} \tag{3.3}$$

This suggests an iterative method by

$$x_1^{k+1} = \frac{1}{a_{11}}\left(b_1 - \sum_{j=2}^{n} a_{1j}x_j^k\right)$$
$$x_2^{k+1} = \frac{1}{a_{22}}\left(b_2 - \sum_{j=1;j\neq 2}^{n} a_{2j}x_j^k\right)$$
$$\cdots \quad \cdots \quad \cdots \quad \cdots$$
$$x_n^{k+1} = \frac{1}{a_{nn}}\left(b_n - \sum_{j=1}^{n-1} a_{nj}x_j^k\right)$$

where x^k means the value of kth iteration for unknown x with $k = 1, 2, 3, \cdots$, and $\mathbf{x}(0) = (x_1^0, x_2^0, \cdots, x_n^0)$ is an initial guess vector.
This is so called **Jacobi**'s method.

Example 3.9. *Apply the Jacobi method to solve*

$$5x_1 - 2x_2 + 3x_3 = 12$$
$$-3x_1 + 9x_2 + x_3 = 14$$
$$2x_1 - x_2 - 7x_3 = -12$$

Choose the initial guess $\boldsymbol{x}^{(0)} = (0, 0, 0)$.
***Solution**: To begin, rewrite the system*

$$x_1^{k+1} = \frac{1}{5}(12 + 2x_2^k - 3x_3^k)$$
$$x_2^{k+1} = \frac{1}{9}(14 + 3x_1^k - x_3^k)$$
$$x_3^{k+1} = \frac{-1}{7}(-12 - 2x_1^k + x_2^k)$$

the approximation is

k	x_1	x_2	x_3
0	0	0	0
1	2.40000000	1.55555556	1.71428571
2	1.99365079	2.16507937	2.17777778
3	1.95936508	1.97813051	1.97460317
4	...	...	...

Example 3.10. *Now for the same previous example but with changing the order of equations:*

$$\begin{aligned} -3x_1 + 9x_2 + x_3 &= 14 \\ 2x_1 - x_2 - 7x_3 &= -12 \\ 5x_1 - 2x_2 + 3x_3 &= 12 \end{aligned}$$

Applying Jacobi method and rewrite the system

$$\begin{aligned} x_1^{k+1} &= \frac{-1}{3}(14 - 9x_2^k - x_3^k) \\ x_2^{k+1} &= -(-12 - 2x_1^k + 7x_3^k) \\ x_3^{k+1} &= \frac{1}{3}(12 - 5x_1^k + 2x_2^k) \end{aligned}$$

Choose the same initial guess $\boldsymbol{x}^{(0)} = (0, 0, 0)$*, the approximation is*

k	x_1	x_2	x_3
0	0	0	0
1	-4.66666667	12.00000000	4.00000000
2	32.66666667	-25.33333333	19.77777778
3	-74.07407407	-61.11111111	-67.33333333
6	...	...	...

and this is divergence.?

Theorem 3.11. *The convergence condition (for any iterative method) is when the matrix* A *is diagonally dominant.*